Administrator's Guide for ArcSight Platform 22.1.2

June, 2022

Legal Notices

Micro Focus
The Lawn
22-30 Old Bath Road
Newbury, Berkshire RG14 1QN
UK

https://www.microfocus.com

Copyright Notice

Trademark Notices

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number
- Document Release Date, which changes each time the document is updated
- Software Release Date, which indicates the release date of this version of the software

To check for recent updates or to verify that you are using the most recent edition of a document, go to:

https://www.microfocus.com/support-and-services/documentation

About this PDF Version of Online Help

This document is a PDF version of the online help. This PDF file is provided so you can easily print multiple topics from the help information or read the online help in PDF format. Because this content was originally created to be viewed as online help in a web browser, some topics may not be formatted properly. Some interactive topics may not be present in this PDF version. Those topics can be successfully printed from within the online help.

Contents

About this Guide

Wednesday, September 7, 2022

This Administrator's Guide contains installation, user, and deployment guidance for the ArcSight software products and components that you deploy in the containerized platform. You can access the additional documents from the Micro Focus Product Documentation website.

Intended Audience

This guide provides information for IT administrators who are responsible for managing the ArcSight software and its environment.

Additional Documentation

ArcSight Platform documentation library includes the following resources:

- *Administrator's Guide to ArcSight Platform* 22.1.2, which provides concepts, use cases, and contextual help for the Dashboard and user management of the Fusion layer in ArcSight Platform.
- *Technical Requirements for ArcSight Platform 22.1.2*, which provides information about the hardware and software requirements for installing ArcSight Platform and the deployed capabilities.
- *ArcSight Platform 22.1.2 Release Notes*, which provides information about the latest release.

For the most recent version of this guide and other ArcSight documentation resources, visit the documentation site for ArcSight.

Contact Information

We want to hear your comments and suggestions about this book and the other documentation included with this product. You can use the comment on this topic link at the bottom of each page of the online documentation, or send an email to Documentation-Feedback@microfocus.com.

For specific product issues, contact Micro Focus Customer Care.

Chapter 1: Introducing ArcSight Platform

ArcSight Platform (the Platform) enables you to deploy a combination of security, user, and entity solutions into a single cluster within the Container Deployment Foundation (CDF) environment. With CDF, you can add and remove product capabilities, as well as manage the workload across the installed nodes.

The Platform enables you to visualize, identify, and analyze potential threats by incorporating intelligence from the multiple layers of security sources that might be installed in your security environment.

These product capabilities might include the following:

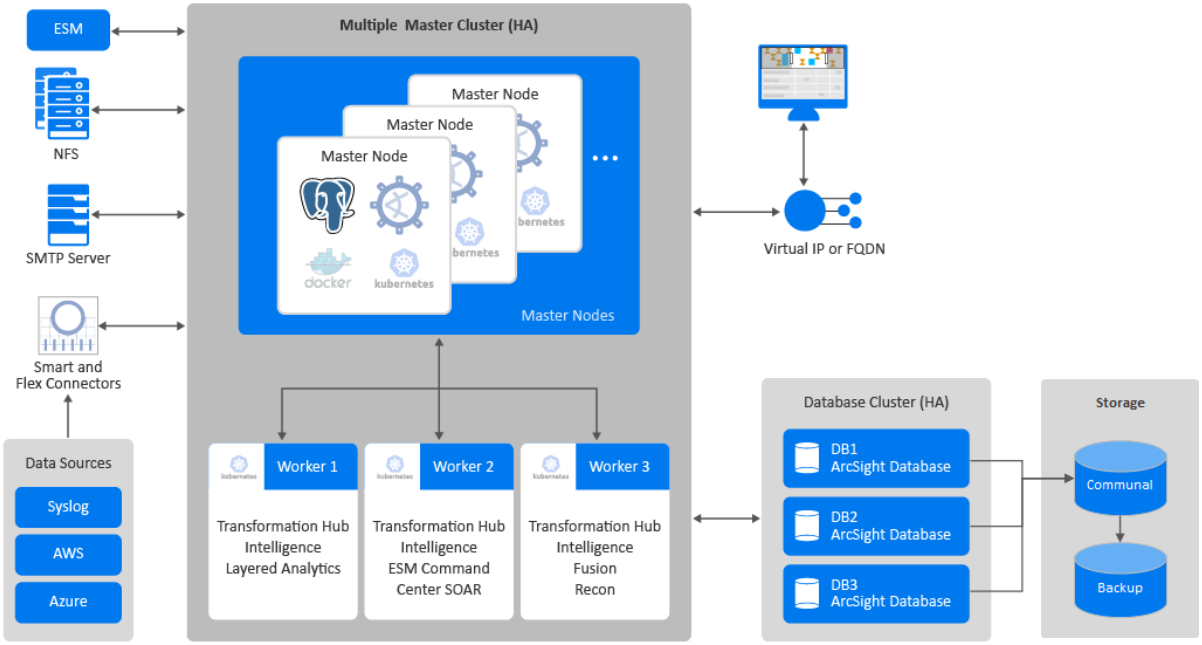
- Real-time event monitoring and correlation with data from ArcSight Enterprise Security Manager (ESM)
- Analyzing end-user behavior with ArcSight Intelligence
- Performing deep-dive investigations with ArcSight Recon
- Responding to and mitigating cyber attacks with ArcSight SOAR
- Coordinating and managing data streams with Transformation Hub

The Platform's SSO function ensures that users can navigate among the features in the Platform or launch applications from the Platform without having to log in for each product solution.

Understanding the Platform Architecture

The Platform includes three primary elements:

- The underlying CDF infrastructure
- The capabilities you deploy into the infrastructure
- The functions and applications that support the deployed capabilities



The following sections describe these three elements of the Platform architecture.

Although you can also deploy NetIQ Identity Intelligence in this CDF-based environment, this *Administrator's Guide* does not provide instructions for deploying or managing that capability. For more information, see the Administrator's Guide to NetIQ Identity Intelligence.

Understanding the CDF Infrastructure

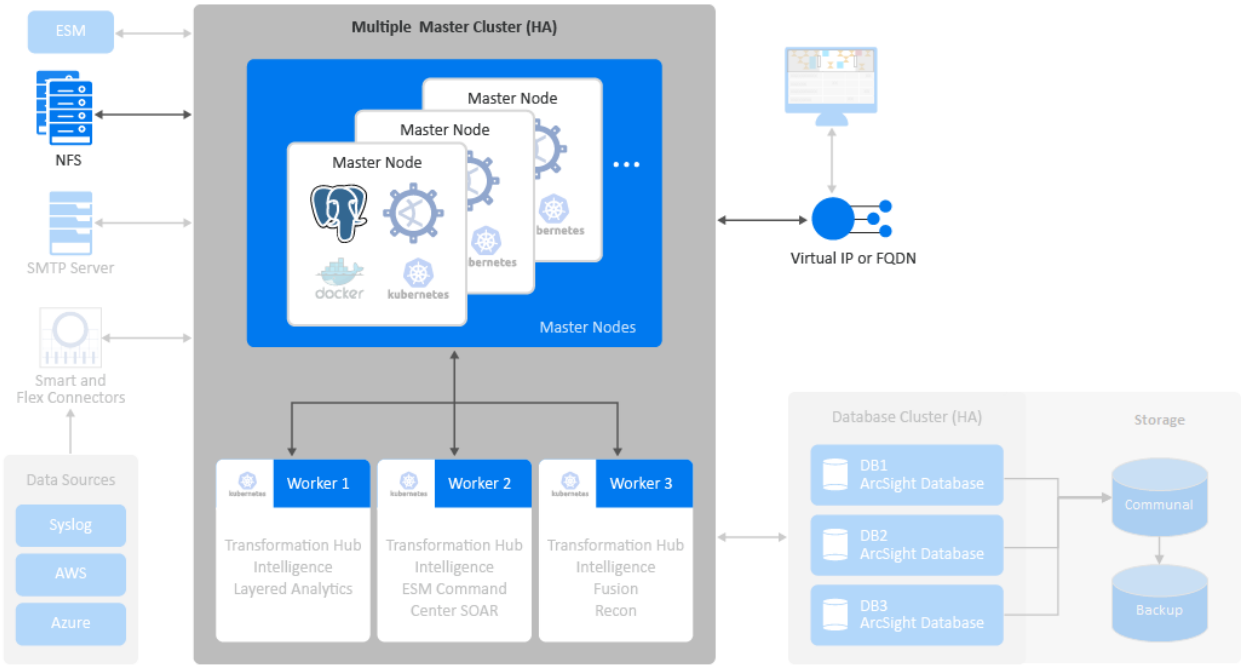
The Platform runs in the Container Deployment Foundation (CDF) infrastructure, which incorporates container management functions from Kubernetes and Docker. This containerized environment enables you to swiftly install and manage an integrated solution of ArcSight products in a single interface. The CDF has both a "CDF Installer" on page 21 and a browser-based "CDF Management Portal" on page 21.

We provide two ways of using the installer function:

- An assisted process using the ArcSight Installation Tool
- A manual process

You will also need to install additional software and components to support your security solution. Your ArcSight environment might include the containerized capabilities, which are distributed across multiple host systems, plus servers for databases and the supporting products.

The number of hosts you need depends on several factors, such as the need for high availability and the size of workloads based on events per second.



The CDF architecture requires several components:

- "CDF Installer" on the next page
- "CDF Management Portal" on the next page
- "Kubernetes and Docker" on the next page
- "Master Nodes" on the next page
- "Network File System" on page 22
- "Worker Nodes" on page 22
- "Virtual IP Address" on page 22

CDF Installer

You use the CDF installer for installing, configuring, and upgrading the CDF infrastructure. When using the ArcSight Installation Tool, the CDF installer is executed automatically in an embedded manner so that you need not use the CDF installer directly.

CDF Management Portal

The Management Portal enables you to manage and reconfigure your deployed environment after the installation process is complete. You can add or remove deployed capabilities and worker nodes, as well as manage license keys.

During installation, you specify the credentials for the administrator of the Management Portal. This administrator is not the same as the admin user that you are prompted to create the first time that you log in to the Platform after installation.

When you upgrade the Platform, you use the Management Portal to upgrade the deployed capabilities.

Kubernetes and Docker

Kubernetes automates deployment, scaling, maintenance, and management of the containerized capabilities across the cluster of host systems. Applications running in Kubernetes are defined as pods, which group containerized components. Kubernetes clusters use Docker containers as the pod components.

A **pod** consists of one or more containers that are guaranteed to be co-located on the host server and can share resources. Each pod in Kubernetes is assigned a unique IP address within the cluster, allowing applications to use ports without the risk of conflict.

Persistent services for a pod can be defined as a volume, such as a local disk directory or a network disk, and exposed by Kubernetes to the containers in the pod to use. A cluster relies on a Network File System (NFS) as its shared persistent storage. The clusters require master and worker nodes. For more information about the Platform pods, see Understanding Labels and Pods.

Master Nodes

The master nodes control the Kubernetes cluster, manage the workload on the worker nodes, and direct communication across the system. You should deploy three master nodes to ensure high availability. However, you can use the Platform with a single master node.

Network File System

The Network File System (NFS) stores some of the persistent data generated by Transformation Hub, Intelligence, and Fusion. This data includes component configuration data for ArcSight Platform based capabilities unrelated to event data.

Worker Nodes

Worker nodes run the application components and perform the work in the Kubernetes cluster. For all highly available configurations, we recommend deploying a minimum of three dedicated worker nodes.

You can add and remove worker nodes from the cluster as needed. Scaling the cluster to perform more work requires additional worker nodes, all of which are managed by the master nodes. The workload assigned to each node depends on the labels assigned to them during deployment or reconfiguration after deployment.

Virtual IP Address

CDF supports high availability (HA) through load balancers and the Keepalived service. You can configure either external load balancers or Keepalived for high availability. If you have configured a virtual IP for a multi-master installation, the HA virtual IP address you defined bonds to one of the three master nodes.

If a master node fails, the virtual IP address is assigned to an active master node. This setup helps to provide high availability for the cluster.

When you configure a connection to the cluster, configure the connection to use the virtual IP so that it benefits from the HA capability. One exception to this recommendation is when you are configuring a connection to Transformation Hub's Kafka, in which case you can achieve better performance by configuring the Kafka connection to connect directly to the list of worker nodes where Kafka is deployed.

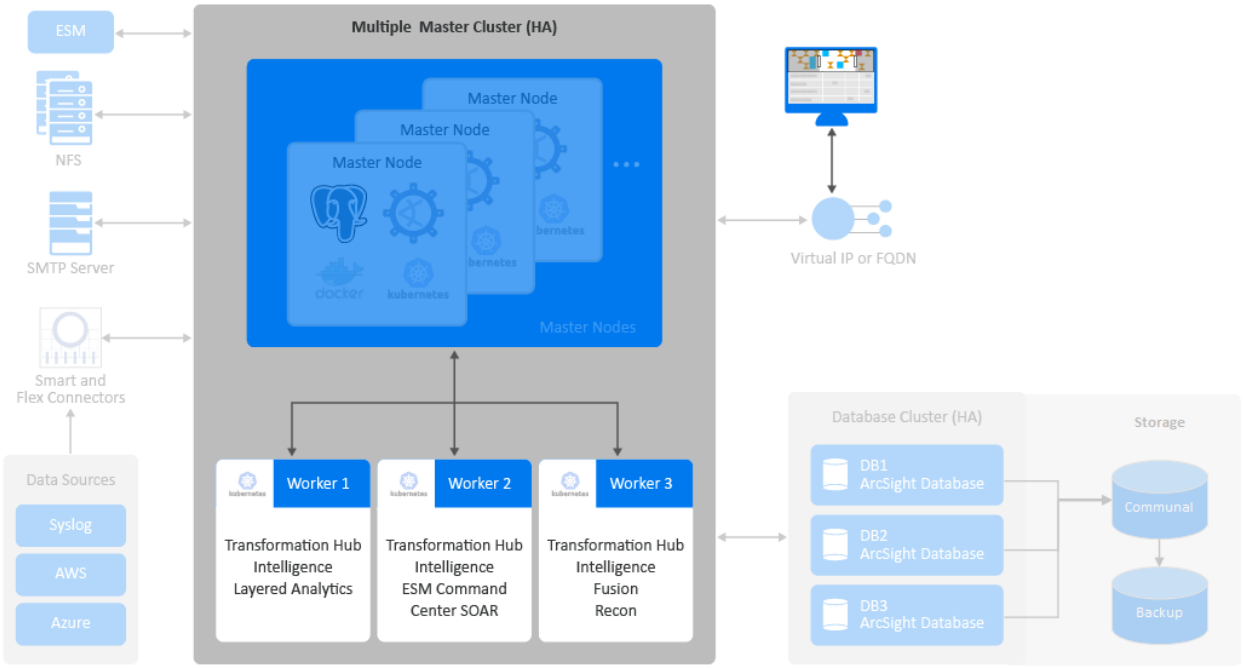
Deciding on the Capabilities to Deploy

The Platform infrastructure enables you to deploy a combination of container-based **capabilities**, which represent licensed products and functions that shape your ArcSight environment. Each release of the Platform supports a specific set of capabilities that you can deploy.

To perform appropriately, some capabilities that you deploy depend on the presence of additional capabilities. For example, most capabilities need the Fusion capability because it provides the user management functions in the Platform.

> The capabilities that can be deployed in the Platform are designed to automatically integrate with each other when deployed to the same cluster. You must deploy capabilities to the same cluster for them to operate in an integrated manner.

For a complete security, user, and entity solution, you might also need to integrate software and components that are not deployed within the Platform. For example, your solution might need a database for data storage and Micro Focus ArcSight SmartConnectors for data collection from various data sources.



You can deploy the following capabilities in the Platform:

- "ESM Command Center" on the next page
- "Fusion" on the next page
- "Intelligence" on page 25
- "Layered Analytics" on page 25
- "Recon" on page 25
- "Transformation Hub " on page 26

For more shared capabilities, see "Understanding Labels and Pods" on page 578

ESM Command Center

ArcSight Command Center for Enterprise Security Manager (ESM Command Center) is a licensed product that provides widgets and dashboards that you can customize in the Dashboard feature for detecting threats to your enterprise. If you deploy "Intelligence" on the next page and "Layered Analytics" on the next page in the same cluster as ESM Command Center, certain widgets will combine data from ESM and Intelligence to provide you greater insight into events and entity behavior.

With Transformation Hub deployed in the same cluster, ESM can receive event data for dashboarding and further correlation.

This capability requires Fusion to be deployed in the same cluster.

Fusion

All capabilities require that Fusion be deployed in the same cluster.

To ensure a unified solution experience, Fusion provides the common elements needed for the products that you deploy in the Platform environment: user management, the Dashboard, SOAR, ArcMC, and other core services. Fusion services also support single sign-on (SSO) configuration across the capabilities, high-capacity data management, and a search engine.

Fusion enables you to add users and groups, as well as manage their roles and permissions. The *My Profile* section of user management enables users to set their preferences for features like Search. The **Dashboard** enables you to visualize, identify, and analyze potential threats by incorporating intelligence from the multiple layers of security sources that might be installed in your security environment.

Fusion ArcMC, the containerized version of ArcSight Management Center (ArcMC), serves as a centralized management interface to help you effectively administer and monitor Transformation Hub and the SmartConnectors. Fusion ArcMC communicates with the Platform by connecting to the virtual IP address or fully qualified domain name (FQDN) assigned to the primary master node in the cluster. You can configure Fusion ArcMC to manage another instance of Fusion ArcMC or to manage the standalone ArcMC product. Note that, although a standalone instance of ArcMC can manage other standalone instances of ArcMC, it cannot manage Fusion ArcMC.

Fusion includes **ArcSight SOAR**, a licensed Security Orchestration, Automation and Response Platform that combines orchestration of both technology and people, automation, and incident management into a seamless experience. You can connect the dots between people, process, and technology in SecOps with various and diverse forms of automation, analyst augmentation, and collaborative investigation and response. With 120+ integrations from different vendors,

SOAR provides a single pane of glass for security operations and speeds up the incident response process. To use SOAR's capabilities, you must deploy the ESM, "Intelligence" below, or Recon capability, or any combination of these capabilities.

Intelligence

ArcSight Intelligence provides a market-leading analytics platform, using unsupervised online machine learning to identify unknown threats like insider threats or targeted outside attacks such as APTs. These types of threats simply cannot be identified by searching for a known “bad signature.” Unsupervised machine learning gives threat hunters a high-quality set of leads to help them identify these elusive threats.

The analytics platform in ArcSight Intelligence uses:

- ArcSight SmartConnectors
- Supporting Active Directory/Authentication data
- Web proxy data
- Additional data sources

In addition, you can use FlexConnectors to pull ArcSight Intelligence analytical results and push them into ESM for higher accuracy correlation rules that leverage unsupervised learning anomalies, as well as correlation rule filtering using top risky entity lists.

If you deploy "ESM Command Center" on the previous page and "Layered Analytics" below in the same cluster as the ArcSight Intelligence capability, certain widgets will combine data from ESM and ArcSight Intelligence to provide you greater insight into events and entity behavior.

This capability requires the Fusion and Transformation Hub capabilities to be deployed in the same cluster, and the ArcSight Database.

Layered Analytics

Layered Analytics blends the analytics results from the "ESM Command Center" on the previous page and "Intelligence" above capabilities, thus providing multiple layers of useful data that can lead to actionable insights.

This capability requires the ESM Command Center and Intelligence capabilities.

Recon

ArcSight Recon is a licensed product that enables you to search, analyze, and visualize machine-generated data gathered from web sites, applications, sensors, and devices that make up your monitored network. Recon indexes the events from your data source so that you can view and search them.

The intuitive search language makes it easy to formulate queries. You can use the large set of dashboards and reports available in the Reports Portal to monitor and identify vulnerabilities and threats in your enterprise.

Recon integrates with "Transformation Hub " below for processing raw events. Recon also can integrate with ESM to receive alerts and start the investigation process.

This capability requires the Fusion and Transformation Hub capabilities to be deployed in the same cluster, and the ArcSight Database.

Transformation Hub

Transformation Hub is a licensed product that lets you take advantage of scalable, high-throughput, multi-broker clusters for publishing and subscribing to event data. It coordinates and manages data streams, which enables your environment to scale, and opens events to third-party data solutions. Moreover, to reduce the computational overhead and workload on a syslog SmartConnector infrastructure, you can make use of Connectors in Transformation Hub (CTH) instead.

Transformation Hub ingests, enriches, normalizes, then routes event data from data producers to connections between existing data lakes, analytics platforms, and other security technologies and the multiple systems within the Security Operations Center (SOC).

Transformation Hub can seamlessly broker data from any source and to any destination. Its architecture is based on Apache Kafka and it supports native Hadoop Distributed File System (HDFS) capabilities, enabling both the ArcSight Logger and ArcSight Recon technologies to push to HDFS for long-term, low-cost storage.

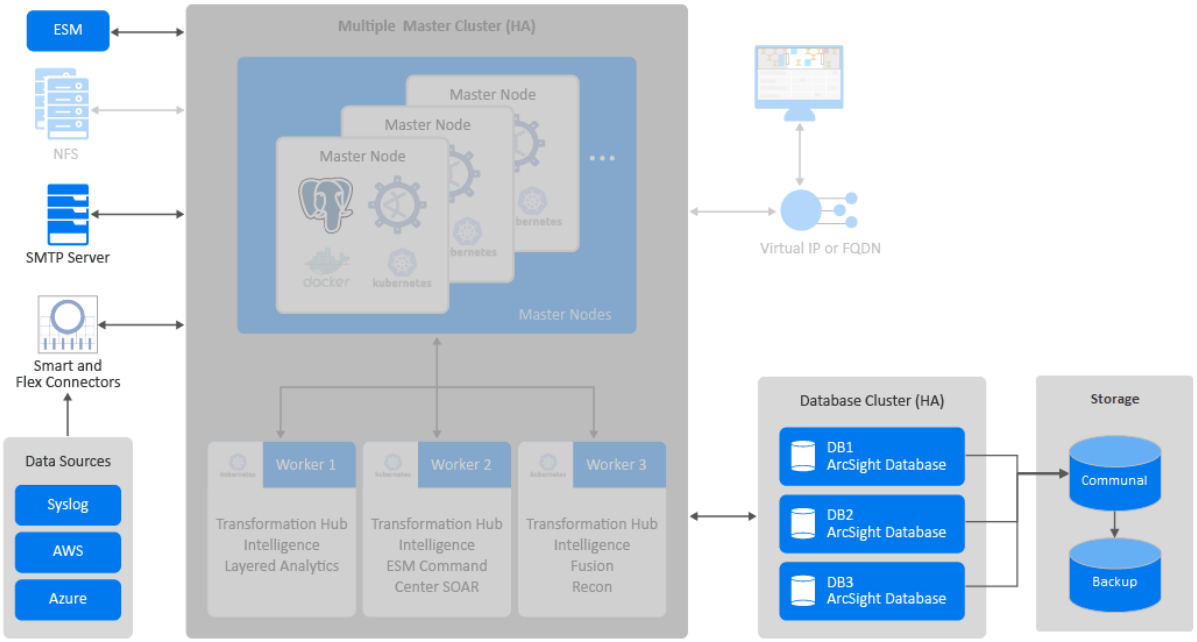
This architecture reduces the overall ArcSight infrastructure footprint, scales event ingestion using built-in capabilities, and greatly simplifies upgrades to newer Transformation Hub releases.

It also positions the platform to support an analytics streaming plug-in framework, supporting automated machine learning and artificial intelligence engines for data source onboarding, event enrichment, and detection and attribution of entities and actors.

This capability requires Fusion to be deployed in the same cluster.

Understanding Related Components

The capabilities you deploy in the Platform depend on functions and applications installed in your environment. For example, Transformation Hub consumes data from a wide variety of collectors and connectors before passing that content to ESM and other products. Recon and Intelligence need the ArcSight Database to store their data.



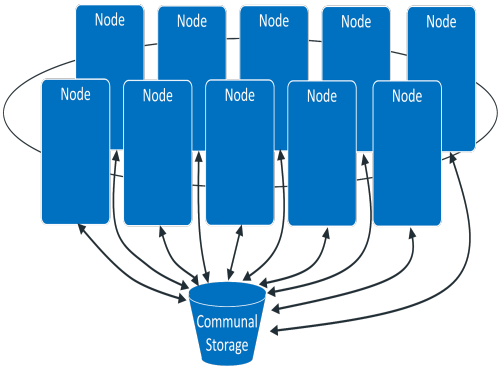
- "ArcSight Database" below
- "Data Sources" on the next page
- "Enterprise Security Manager" on the next page
- "SMTP Server" on page 29

ArcSight Database

The ArcSight Database stores all collected events and provides event searches and analysis capabilities.

The database keeps the primary copy of your data in **Communal Storage**, and the local cache serves as the secondary copy. Communal storage is the database's centralized storage location, shared among the database nodes. This means that adding and removing nodes does not redistribute the primary copy. Communal storage is based on an object store, such as Amazon's S3 service in the cloud or an S3 compatible object store in an on-premises deployment.The database relies on the object store to maintain the durable copy of the data.

Within communal storage, data is divided into portions called *shards*. Shards are how the database divides the data among the nodes. Nodes subscribe to particular shards, with subscriptions balanced among the nodes. When loading or querying data, each node is responsible for the data in the shards that it subscribes to.



This shared storage model enables elasticity, meaning it is both time and cost effective to adapt the cluster resources to fit the usage pattern of the cluster. If a node goes down, other nodes are not impacted because of shared storage. Node restarts are fast and no recovery is needed. Thus, you do not need to keep track of and load/unload long- term retention event data explicitly. The ArcSight Database can bring them to the cache on demand automatically then move data out when not in use. To expand communal storage, you can purchase additional storage devices rather than purchasing additional CPU and memory.

Data Sources

The deployed capabilities incorporate data from a variety of sources.

- **SmartConnectors** collect events from supported data sources, normalize those events, then send them to the Transformation Hub's Kafka cluster.
  - When collecting data and sending it to Transformation Hub, the SmartConnector normalizes the values (such as severity, priority, and time zone) into the common format and normalizes the data structure into the common schema.
  - Next, the connectors filter and aggregate events to reduce the volume of events sent to the system.
  - You need to install and maintain connectors separately.
  - You can subscribe to the data Transformation Hub manages.
- Third-party collectors and connectors also provide data to the deployed capabilities.

Enterprise Security Manager

ArcSight Enterprise Security Manager (ESM) operates outside of the Platform CDF environment, but integrates with capabilities that operate within the Platform environment. For example, ESM shares SSO, event processing, and event search behavior with the Platform.

You can deploy the ESM Command Center capability to the Platform CDF environment to provide a more seamless user experience with other capabilities that integrate with the

Platform Fusion capability, such as Intelligence and SOAR. When deployed in this manner, ESM Command Center integrates with ESM operating outside of the Platform CDF environment.

SMTP Server

The SMTP server enables the Platform to send notification messages to users. For example, when you create new users, you need the SMTP server to notify the users about their account and how to change their passwords.

Chapter 2: Planning to Install and Deploy

This section describes the installation and deployment options, considerations, and caveats that you need to know for a successful deployment.

Checklist: Planning to Deploy the Platform

Use the following checklist to install and configure the Platform infrastructure. Perform the tasks in the listed order.

	Task	See
❑	1. Learn about the software and components that you need to install, deploy, and configure.	Deciding on the Capabilities to Deploy Understanding the CDF Infrastructure Understanding Related Components
❑	2. Decide how you want to configure your Platform environment, and ensure that the computers on which you are installing the Platform components meet the specified requirements.	ArcSight Platform Technical Requirements
❑	3. Review the knowledge and individuals needed to perform the installation processes.	Identifying the Installation Team
❑	4. Review the considerations for creating the Platform infrastructure.	Reviewing the Considerations and Best Practices
❑	5. Understand the security modes and their prerequisites needed for establishing communication between the infrastructure components.	Understanding Secure Communication Among Components
❑	6. If your connectors do not send raw event data, review the fields that the database indexes for Search.	Indexing Event Fields Before Installing the Database
❑	7. Decide whether to use the ArcSight Installation Tool (on-premises only) or the manual process.	Choosing Your Installation Method

Identifying Your Installation Team

Your installation will require specific administration skills, and coordination with corporate IT departments, including the following:

- Linux operating system administration (including applying OS updates; configuring networks, firewalls, ports, and user access; and performing additional tasks)
- Familiarity with editing configuration files
- Running commands and scripts on one or more operating systems
- Familiarity with Micro Focus components
- Familiarity with Kafka processing and configuration

Your installation team will need the following roles and responsibilities to properly configure the infrastructure environment.

Role	Responsibility
Application admin	The person in this role must ensure successful execution of the entire installation including verification and post-installation tasks. This person must have a good understanding of the entire installation process, request support from other appropriate roles as needed, and complete the installation once the environment is ready for installation.
IT admin	The person in this role prepares physical or virtual machines as requested by the application administrator.
Network admin	The person in this role manages network-related configuration for your organization. This person needs to perform network configuration tasks as requested by the application administrator.
Storage admin	The person in this role plans and deploys all types of storage for your organization. This person needs to set up one or more NFS servers required by the CDF installation.
Cloud Technology admin	The person in this role demonstrates an understanding of the cloud key concepts and their relevant terminology. As such, they manage cloud-related configurations for your organization.

Reviewing the Considerations and Best Practices

Before starting the installation process, there are several decisions to be made to plan and prepare your infrastructure. Below are the considerations you need to consider, as well as an outline of steps you to follow during this planning and preparation process. We will explain details in later sections of this guide.

Consideration	Best Practices
Host Systems	• Your host systems must meet or exceed the technical requirements for CPU cores, memory, and disk storage capacity, and anticipated requirements for end-to-end events processing throughput. In addition, with insufficient resources available on a host, the installation process may fail. Consult the ArcSight Technical Requirements for guidance. • Provision cluster (master and worker node) host systems and operating environments, including OS, storage, network, and Virtual IP (VIP) if needed for high availability (HA). Note the IP addresses and FQDNs of these systems for use during product deployment. • You can install the cluster using a `sudo` USER with sufficient privileges, or, alternatively, you can install it using the `root` USERID. • Master and worker nodes can be deployed on virtual machines. However, since most of the processing occurs on worker nodes, if possible, you should deploy worker nodes on physical servers. • When using virtual environments, please ensure: ◦ Resources are reserved and not shared. ◦ The UUID and MAC addresses are static and do not change after a reboot or a VM move. Dynamic IP addresses will cause the Kubernetes cluster to fail. • All master and worker nodes must be installed in the same subnet. • If a master and worker are sharing a node, follow the higher-capacity worker node sizing guidelines. Microfocus does not recommend this configuration for production Transformation Hub environments.

Consideration	Best Practices
High Availability	• For high availability (HA) of master nodes on a multi-master installation, you must create a Virtual IP (VIP) which will be shared by all master nodes. Prior to installation, a VIP must not respond when pinged. • All master nodes should use the same hardware configuration, and all worker nodes should use the same hardware configuration (which is likely to be different from that of the master nodes). • For HA, exactly three master nodes, at least three worker nodes, and at least three database nodes should be used so that if one of each node type fails, the remaining nodes can continue to operate the system without downtime. This is the configuration illustrated in the diagram. You can use fewer nodes of each node type. However, this configuration will result in that node type not being HA. • For HA, use an NFS server that is separate from the Kubenetes cluster nodes, and has HA capabilities, so there is not a single point of failure. For example, this could be 2 NFS servers (active/passive) configured with replication with a Virtual IP managed between them which CDF is configured to use to connect to the NFS server. An example of configuring the 2 NFS servers in replication mode is described here. ***Note:*** *Link opens an external site.* • For master nodes, only 1 or 3 master nodes are allowed. • If you deploy a single master node, failure of the single master node could cause you to lose the ability to manage the entire cluster until you recover the single master node. In some extreme scenarios, failure of the single master node could cause the entire cluster to become unrecoverable, requiring a complete reinstall and reconfiguration. When using only a single master node, the system will be more reliable if you also host the NFS server on the same master node. • When the installer is configured to create more than one database node, the database fault tolerance will be set to one. This means the data in the database will be replicated so that one database node can fail and the system will continue to operate properly. Database storage utilization will double as a result of the data replication. In a failure scenario, the failed node should urgently be restored before there is a chance of another node failure, which will shut down the database to avoid additional problems. • If you configure the installer to create only a single database node, the database fault tolerance is set to zero because there is only a single node. Therefore, no other node will continue during a failure, and no data replication will occur in this scenario.
Storage	• Create or use an existing NFS storage environment with sufficient capacity for the throughput needed. Guidelines are provided below. • Determine the size and total throughput requirements of your environment using total EPS. For example, if there are 50K EPS inbound, and 100K EPS consumed, then the size would be 150K EPS. • Data compression is performed on the producer side (for example, in a Smart Connector).

Consideration	Best Practices
Scaling	• Adding more worker nodes is typically more effective than installing bigger and faster hardware because individual workloads on worker nodes are usually relatively small and some of them work better when there are fewer different workloads on the same node. Using more worker nodes also enables you to perform maintenance on your cluster nodes with minimal impact to your production environment. Adding more nodes also helps with predicting costs due to new hardware. • Unlike worker nodes, for the database it is typically more effective to use bigger and faster hardware than to increase the number of database nodes because the database technology can fully utilize larger hardware and this decreases the need for coordination between database nodes. With that said, for HA it is important to deploy enough database nodes to be resilient in case of a database node failure or individual node downtime for maintenance.
Network	• Although event data containing IPv6 content is supported, the cluster infrastructure is not supported on IPv6-only systems.
Cloud	All SmartConnector or Collector remote connections depend on the Operating System random number pool (entropy pool) to generate private keys for secure communication. For a cloud environment, you might need to increase the entropy pool beyond the lower limit of 3290 to ensure uninterrupted communication. For more information see, "SmartConnector or Collector Remote Connections Failing Due to Low Entropy" in the Installation Guide for ArcSight SmartConnectors.
Security	• Determine a security mode (FIPS, TLS, Client Authentication) for communication between components. • "Determining a Security Mode Between Components" on page 49
Performance	• If SmartConnector is configured to send events to Transformation Hub in CEF format and the events are being stored in ArcSight Database, consider the potential performance effects of the CEF-to-Avro data transformation, and allow a 20% increase in CPU utilization. This will generally have a large impact only with very high EPS (250K+) rates. Consider configuring the SmartConnector to use the Avro event format instead, which avoids the need for this transformation.
Downloads and Licensing	• Ensure that you have access to the Micro Focus software download location. You will download installation packages to the Initial Master Node in the cluster. • Ensure that you have a valid Micro Focus license key for the software being installed.
Installing with Enterprise Security Manager	If you want to install the Platform and the ESM server in the same environment, specify during the Platform intallation a CDF API Server Port that does not use the same port as the ESM server (default 8443). For example, when using the Platform Install tool, the `example-install-config-esm_cmd_center-single-node.yaml` sets the master-api-ssl-port to port 7443.

Understanding Object Storage Options for the ArcSight Database

When installing the database on-premises, you can choose to use any one of several S3-compatible object store technologies. To choose a technology compatible with the ArcSight Database system, see Eon On-Premises Storage.

Not all S3 compatible object store technologies provide the same performance or capabilities. Research is important to determine the ideal solution for your needs. We've found that MinIO works well in the testing we've performed in our labs, but other solutions might work better for your environment. To help you get an idea of how to configure MinIO for use with ArcSight Database, see the example in "'Configuring the Database for MinIO Storage (Examples Only)" on page 711."

Understanding Firewall Ports for the ArcSight Platform

This section lists the ports that must be open for the elements that make up the ArcSight Platform:

- Underlying Container Deployment Foundation (CDF) infrastructure components
- Capabilities that you deploy into the CDF infrastructure
- Components that support the deployed capabilities

Firewall Ports for CDF Infrastructure Components

The following tables list the ports that must be open for the CDF infrastructure components:

- CDF Vault
- CDF Management Portal
- Kubernetes
- Network File System (NFS)

In most cases, the firewalls for these components are host-based. These components are not likely to have network-based firewalls between them.

In most cases, you do not need to take action to configure the firewalls for these ports.

CDF Vault

Ports (TCP)	Node	Description
8200	Master	Used by the `itom-vault` service, which provides a secured configuration store All cluster nodes should be able to access this port for the client connection.
8201	Master	Used by the `itom-vault` service, which provides a secured configuration store Web clients must be able to access this port for peer member connections.

CDF Management Portal

Ports (TCP)	Node	Description
3000	Master	Used only for accessing the CDF Management Portal during CDF installation from a web browser Web clients must be able to access this port during the installation of CDF. After installation, web clients use port 5443 to access the CDF Management Portal.
5443	Master	Used for accessing the CDF Management Portal post CDF deployment from a web browser Web clients must be able to access this port for administration and management of CDF.
5444	Master	Used for accessing the CDF Management Portal post CDF deployment from a web browser, when using two-way (mutual) SSL authentication Web clients must be able to access this port for administration and management of CDF, when using two-way (mutual) SSL authentication.

Kubernetes

Ports (TCP)	Node	Description
2380	Master	Used by the `etcd` component, which provides a distributed configuration database All the master nodes should be able to access this port for the `etcd` cluster communication.

4001	Master	Used by the `etcd` component, which provides a distributed configuration database All cluster nodes should be able to access this port for the client connection.
5000	Master	Used by the `kube-registry` component, which handles the management of container image delivery All cluster nodes should be able to access this port to communicate with the local container registry.
7443	Master	*(Conditional)* Used by the Kubernetes API server when you perform one of the following methods of installation: • Use the provided scripts • Install manually and on the same node as ESM All cluster nodes should be able to access this port for internal communication.
8443	Master	*(Conditional)* Used by the Kubernetes API server when you manually install and the installation is not on the same node as ESM All cluster nodes should be able to access this port for internal communication.
8472	All nodes	*Uses UDP protocol* Used by the Flannel service component, which manages the internal cluster networking All cluster nodes should be able to access this port for internal communication.
10250	All nodes	Used by the Kubelet service, which functions as a local node agent that watches pod specifications through the Kubernetes API server All cluster nodes should be able to access this port for internal communications and worker node Kubelet API for exec and logs.
10251	All nodes	Used by the `Kube-scheduler` component that watches for any new pod with no assigned node and assigns a node to the pod All cluster nodes should be able to access this port for internal communication.
10252	All nodes	Used by the `kube-controller-manager` component that runs controller processes which regulate the state of the cluster All cluster nodes should be able to access this port for internal communication.
10256	All nodes	Used by the `Kube-proxy` component, which is a network proxy that runs on each node, for exposing the services on each node All cluster nodes should be able to access this port for internal communication.

Network File System (NFS)

Ports (TCP)	Node	Description
111	NFS server	Used by the `portmapper` service All cluster nodes should be able to access this port.
2049	NFS server	Used by the `nfsd` daemon All cluster nodes should be able to access this port. **Note:** This port must be open even during a single-node deployment.
20048	NFS server	Used by the `mountd` daemon All cluster nodes should be able to access this port.

Firewall Ports for Deployed Capabilities

The following tables list the ports that must be available when you deploy the associated capability into the CDF infrastructure:

- ArcMC
- Intelligence
- SOAR
- Transformation Hub

In most cases, you do not need to take action to configure the firewalls for these ports.

ArcMC

Ports	Direction	Description
32080, 9000	Inbound	Used for Transformation Hub and ArcMC communication

Intelligence

Ports	Node	Direction	Description
TCP 30820	Worker (HDFS Namenode)	Inbound	Used for the database to connect to HDFS during Analytics processing
TCP 30070	Worker (HDFS Namenode)	Inbound	Used for the Hadoop Monitoring Dashboard (optional)
TCP 30010	Worker (HDFS Datanodes)	Inbound	Used for communication between the HDFS Namenode and the HDFS Datanodes
TCP 30210	Worker (HDFS Datanodes)	Inbound	Used by the database to establish secure communication with HDFS during Analytics processing

SOAR

The SOAR cluster listens on the following ports on all Kubernetes master and worker nodes, but Micro Focus recommends that you only use the ports on the master virtual IP.

Port	Description
32200	Data from ESM
32201	Data from QRadar
32202	Data from McAfee

Transformation Hub

Ports (TCP)	Direction	Description
2181	Inbound	Used by ZooKeeper as an inbound port
9092	Inbound	Used by Kafka during non-SSL communication
9093	Inbound	Used by Kafka when TLS is enabled
32080	Outbound	Used by Transformation Hub to send data to ArcMC
32081	Outbound	Used by Schema Registry to send data to Avro consumers.
443	Inbound	Used by ArcMC
9000	Inbound	Used by ArcMC
9999, 10000	Inbound	Used by the Transformation Hub Kafka Manager to monitor Kafka
39001, 39050	Outbound	Used by ArcMC to communicate with Connectors in Transformation Hub

Firewall Ports for Supporting Components

The following tables list the ports that must be available for supporting components:

- Database
- SmartConnectors

Database

The database requires several ports to be open on the local network. Micro Focus does not recommend placing a firewall between nodes (all nodes should be behind a firewall), but if you must use a firewall between nodes, ensure that the following ports are available:

Ports	Description
TCP 22	Required for the Administration Tools and Management Console Cluster installation wizard
TCP 5433	Used by database clients, such as vsql, ODBC, JDBC, and so on
TCP 5434	Used for Intra-cluster and inter-cluster communication
UDP 5433	Used for database spread monitoring
TCP 5438	Used as Management Console-to-node and node-to-node (agent) communication port
TCP 5450	Used to connect to Management Console from a web browser and allows communication from nodes to the Management Console application/web server
TCP 4803	Used for client connections
UDP 4803	Used for daemon to daemon connections
UDP 4804	Used for daemon to daemon connections
UDP 6543	Used to monitor daemon connections

SmartConnectors

If you have SmartConnectors that are deployed logically far away in the network with firewalls in between, those intermediate firewalls will need to permit traffic on port 9092 (for non-SSL traffic) and 9093 (for SSL traffic).

Port	Direction	Description
• 1515 (Raw TCP) • 1999 (TLS)	Inbound	Used by SmartConnector to receive events
• 9092 (Non-SSL) • 9093 (SSL)	Outbound	Used by SmartConnector to send data to Transformation Hub

Understanding Security Modes

ArcSight consists of multiple different products, services, and infrastructure components. The different products and services communicate with each other and they communicate with any administrative applications. You must secure the different communication channels between the different components to stop security breaches and protect your data. ArcSight enables you to deploy the different products, services, and infrastructure components using your selected security mode. You must understand these possible different modes and how to secure communication between the different components.

IMPORTANT: You must deploy all of the components using the same security mode. If you do not, the communication between the components does not work. If you want to change a

security mode for a component, you must uninstall it and reinstall with the correct security mode for communication to function correctly.

ArcSight supports several security modes of communication between the infrastructure components and the ArcSight products or services. You must understand these different security modes so that you know how to install the different components. All components must have the same security mode. If you want to change a security modes of a component, you have to uninstall and reinstall the component. Use the information about the different security modes to plan your deployment of ArcSight and the infrastructure components.

The supported security modes are:

- **Allow Plain Text:** All communication, between the components, occurs as plain text. ArcSight recommends that you never use this mode in a production environment. Using plain text causes security issues and data breaches.
- **FIPS-compliant TLS Settings:** All communication, between the components, meets the Federal Information Processing Standard (FIPS) established by the United States. For more information, see Using ArcSight Platform and Products in FIPS Mode
- **TLS Client Authentication:** Allows secure communication between components that do not utilize client user name and password authentication, such as producers and consumers connecting to Transformation Hub. With TLS Client Authentication enabled, the client and the server authenticate each other to ensure that both parties involved in the communication are trusted.
- **TLS:** Allows secure communication between the components that have been configured with public key infrastructure certificates to be able to communicate over the transport layer security (TLS). For more information, see "Understanding the Components of Secure Communication" below.

Understanding the Components of Secure Communication

ArcSight uses common industry standards to secure communication such as X.509 certificate, public key infrastructure (PKI), and transport layer security (TLS). You must decide which security mode that you will use during the deployment of the ArcSight products and the infrastructure products.

This section contains the following topics:

Understanding Public Key Infrastructure and TLS Components

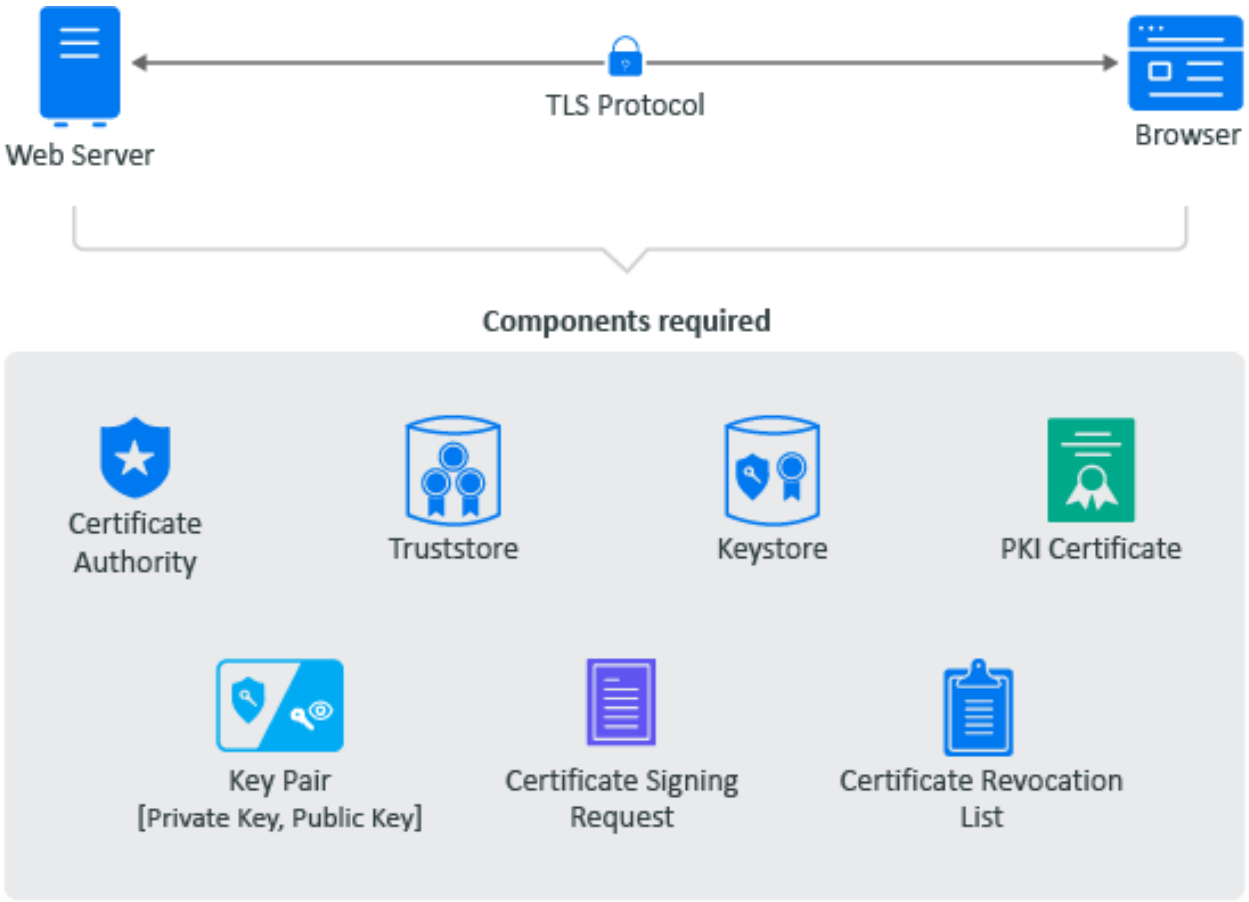
In order to configure secure communication between the ArcSight products, infrastructure products, and any administrative tools requires that you have a good understanding of the components that enable the secure communication to occur. ArcSight uses the industry standards of X.509 certificates, public key infrastructure (PKI), and transport layer security (TLS). This section provides a basic introductions to these components. For more detailed information, see (links open an external site):

- Internet X.509 Public Key Infrastructure Certificates and Certificate Revocation List (CRL) Profile
- The Transport Layer Security (TLS) Protocol Version 1.2
- The Transport Layer Security (TLS) Protocol Version 1.3

You mus secure the communication channels between servers and clients to protect your data and stop security breaches from happening in your environment. The following graphic depicts the different components required for secure communication using certificates, PKI, TLS, and tools to manage the keys.

The secure communication occurs between a server and a client. An example server is a web server and a client could be a browser. The following items are the terms that you need to

understand to create a secure connection between a server and a client.



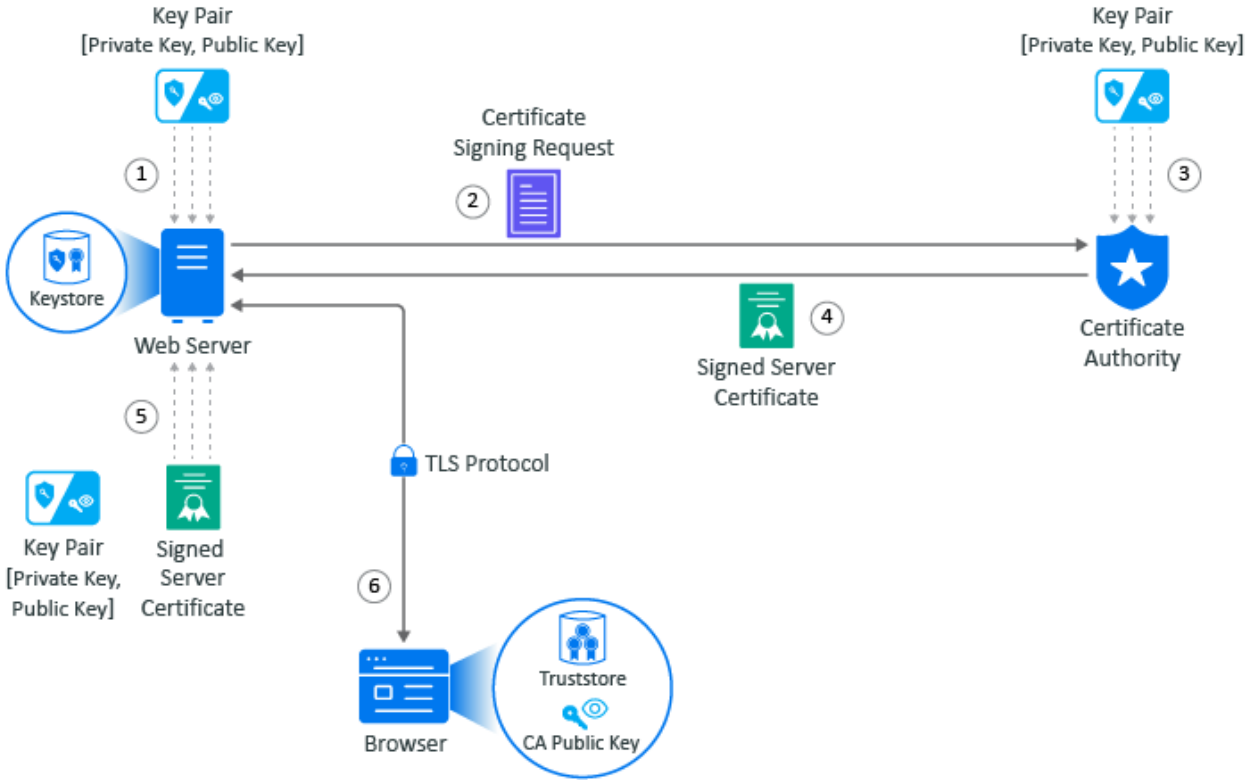
- **Certificate Authority (CA)**: It is an entity that issues digital certificates. A certificate authority (CA) acts as a trusted third party—trusted both by the subject (owner) of the certificate and by the party relying upon the certificate. There are two different types of CAs:
  - **Well-known:** A certificate authority that provides server certificates signed by well-known CAs such as IdenTrust or DigiCert.
  - **Self-signed:** A certificate authority that other products such as openSSL, eDirectory, and Active Directory that contain a certificate authority. You can create self-signed certificates through the certificate authorities in these other products to use in test environments. A security recommendation is to use a well-known CA to issue certificates in productions environments.
- **Public Key Infrastructure (PKI) Certificates:** Digital certificates that the CA issues that prove ownership of the certificate. The CA can issue certificates for users, applications, or devices. The PKI certificates contain the following information:
  - Version number
  - Unique serial number
  - CA digital signature and algorithm used
  - Validity period

  - Certificate Usage
  - Subject name, URL, email address
  - Public and private keys (sometimes it is only the public key)
- **Key Pair:** Consists of a private key and public key that work together to encrypt and decrypt messages. PKI is based on the fact that everyone will trust any communication encrypted with a public key or trust any certificate signed by a private key.
  - Private Key: A cryptographic key that you use it to decrypt any communication encrypted by the public key. Only the private key of the key pair can decrypt the communication encrypted with the corresponding public key. You keep the private key private and do not share it.
  - Public Key: A cryptographic key that you use to encrypt communications to keep the communication secure. Only someone with the private key can decrypt the communications. You share the public key so that anyone with access to the public key can verify that any communication signed with this public key is really from the sending source.
- **Certificate Revocation List:** A list that the CA creates and manage that contains a list of unique serial numbers that it has revoked. The CA uses the certificate revocation list (CLR) to denied requests from any user, application, or device that contain a serial number on the CLR.
- **Certificate Signing Request:** A message sent from an applicant to the CA to apply for a PKI certificate. Usually the certificate signing request (CSR) contains a copy of the public key of the applicant making the request, identifying information such as a domain name, and a digital signature.
- **KeyStore:** A secure Java repository that stores the private key and identity certificate for the server in the trust relationship. The information is stored encrypted on the server with a KeyStore password that you set and manage. Use either the keytool or keytoolgui tools to set and manage the KeyStore passwords.
- **TrustStore:** A secure Java repository that stores the certificates signed by a CA in a secure repository on the client. The information is stored and encrypted on the client with a TrustStore password that you set and manage. Use either the keytool or keytoolgui tools to set and manage the TrustStore passwords.
- **TransportLayerSecurityProtocol:** A secure protocol created by all of the components defined in this section. It allows the server and client to communicate securely by using certificates and key pairs to prove identity on the server and client.

Example: Establishing Secure Communication for a Web Server

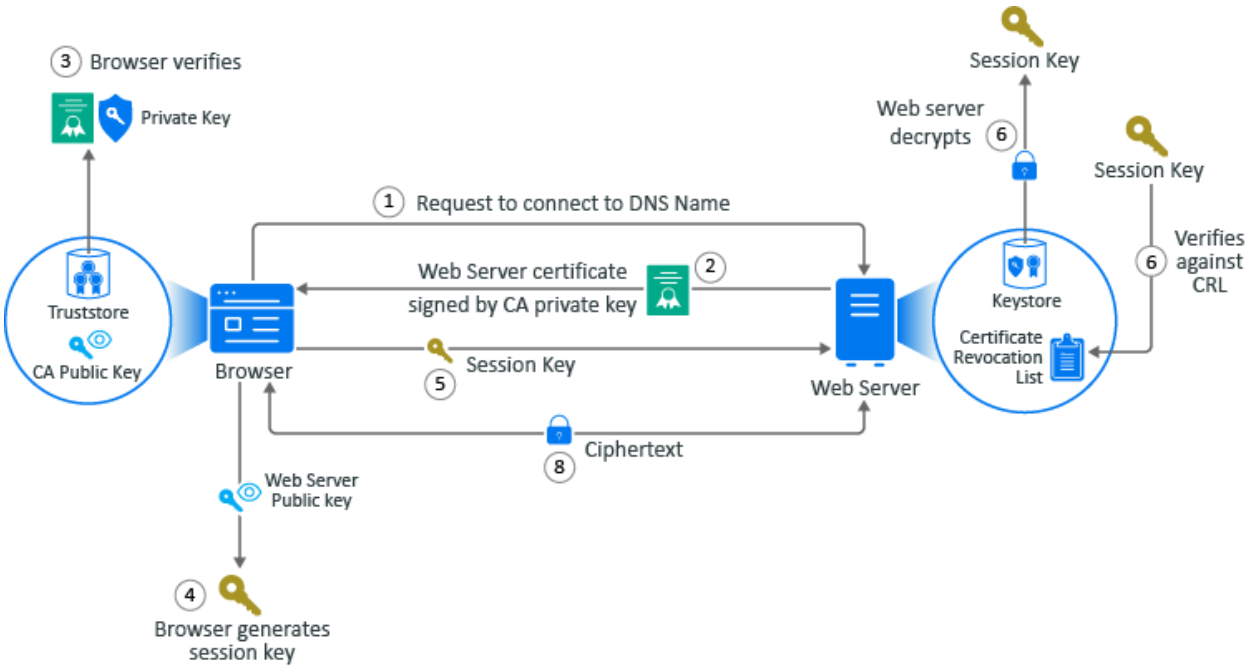
When you install a web server, communication is not secure by default. If the communication is secure, it is usually using a self-signed certificate. The following example shows how the web

server obtains a server certificate signed by a well-known certificate authority (CA) to use in establishing secure communications with any client. In the example, Adam the administrator requests a signed server certificate from the well-known CA and uses the certificate to establish secure communications with a client that is a web application.



1. Adam generates a key pair on the web server using keytool. Adam use the key pair to create a certificate signing request (CSR) using keytool. The CSR contains the fully qualified DNS name of the server, the key pair, and other such information to help identity the web server.
2. Adam sends the CSR that contains the web server's information to a well-known CA such as DigCert.
3. The CA uses the CSR to generate a server certificate for the web server. The CA uses its private key to sign the certificate. The server certificate contains the key pair and the web server's information included in the CSR. The CA signs the certificate with its private key.
4. The CA sends the signed web server certificate back to Adam.
5. Adam imports the signed web server certificate into the web server and the web server's certificate and private key are stored in the KeyStore on the web server.
6. When a browser access the web server, the web server sends a certificate signed by the private key of the CA to the browser. The browser has a copy of the CA's public key in its TrustStore and uses the public key to decrypt the signature of the CA. Now, the browser will trust any communication coming from this web server.

Example: Secure Handshake for the Client



This example show how the secure handshake occurs between a client and a server so that they can create their own secure communication channel that no other entities can use or access. In this example, Adam the administrator logs into the administration console that is a web application. Every action (except for Adam entering the URL in the web browser) happens automatically between the browser and the web server. No user interaction is required.

1. Adam adds a the URL into the browser. The browser sends a request to connect to the fully qualified DNS names of the web server.
2. The web server sends a copy of its server certificate that has been signed by the private key of a well-known CA.
3. The browser access the public key of the well-known CA that is stored in the browser's TrustStore. The browser uses the public key of the well-known CA to decrypt the signature on the web server's certificate to verify that the certificate is valid.
4. The browser generates a session key using the public key in the web server's certificate.
5. The browser sends the newly generated session key back to the web server.
6. The web server uses its private key stored in the KeyStore to decrypt the session key.
7. The web server verifies that the session key is not on the certificate revocation list (CLR). At this point the secure handshake between the browser and web server is established.
8. The web server encrypts the data using the session key and sends the data back in ciphertext to the browser. The browser uses the session key to decrypt the data and then

uses the session key to encrypt data and then it sends the data back in ciphertext. This secure communication continues until the session ends.

.

Using Arcsight Platform and Products in FIPS Mode

The **Federal Information Processing Standard (FIPS)** comprises a set of rules and regulations defined by the United States government that specify the security requirements for data processing and communication between the components.

- Understanding FIPS 140 Security Requirements
- Enabling FIPS Mode for ArcSight Platform Components

For a more thorough understanding of FIPS, official FIPS documentation (FIPS PUBS) is available online.

Understanding FIPS 140 Security Requirements

FIPS 140 is one of the standards of FIPS that governs the use of encryption and cryptographic services. FIPS 140 defines security rules and regulations for cryptographic modules to keep sensitive information secure.

According to the **Federal Information Security Management Act (FISMA)**, all the United States government agencies, United States government contractors, and third parties working for the federal agencies must adhere to the FIPS 140 standard.

For testing cryptographic modules, the two revised editions of FIPS 140 are given below:

FIPS Publication	Standard
FIPS 140-2	Includes changes in technology and standards defined by other standards bodies. Includes modifications based on comments from vendors, laboratories, and user communities.
FIPS 140-3	Aligns with standards defined by the **International Organization for Standardization (ISO)** and the **International Electrotechnical Commission (IEC)**

Enabling FIPS Mode for ArcSight Platform Components

Most components in the ArcSight Platform architecture can operate in the FIPS 140 mode: this includes all of the components that directly handle event data from edge ingestion, to storage in the database, to retrieval from the database supports FIPS 140 Mode. FIPS 140 mode is active by default for some components and cannot be disabled. ArcSight Platform establishes a secure communication between its components using FIPS-validated cryptographic modules.

The table below describes the component level FIPS 140 support:

Component	Sub-components that support FIPS mode	Enabling FIPS mode
ArcSight Management Center (ArcMC)	fusion-arcmc-web-app	• Always enabled
Database	All	• See the Setting FIPS Mode on the Database Server section.
Enterprise Security Manager (ESM)	All	• Enabled by default on fresh installations. • See the Configuring the Deployed Capabilities section. • See the *User's Guide for ESM*. • Complete the process described in the *Administrator's Guide to ESM*.
Fusion	All	• Always enabled
Intelligence	All	• Always enabled
Layered Analytics	All	• Always enabled
Recon	All	• Always enabled

Component	Sub-components that support FIPS mode	Enabling FIPS mode
SmartConnectors	All	• See the SmartConnectors section

SOAR	`soar-web-app`
	`soar-message-broker`
	`soar-jms-migration`
		For the sub-components listed that support FIPS mode, FIPS mode is always enabled.

Transformation Hub	`th-kafka`
	`th-kafka-manager`
	`th-schemaregistry`
	`th-routing-processor`
	`th-c2av-processor`
	`th-web-service`
	`th-cth`
	`th-c2av-processor-esm`
	`th-enrichment-processor`
		• For the sub-components listed that support FIPS mode, FIPS mode can be enabled during deployment.
		• When using the ArcSight Platform Installer tool, add the property `th-init-fips: true` to the `suite > config-params` section of your installation configuration `yaml` file. For example:
		`suite:`
		`  products: [fusion, esm, soar, transformationhub]`
		`  config-params:`
		`  th-init-fips: true`
		• When performing the installation manually, configure the `Transformation Hub > Connections use FIPS encryption` option as described in the Configuring the Deployed Capabilities section.

Components that can not operate in the FIPS 140 mode use strong industry standard encryption to establish secure communication. However, our objective is to increase the coverage of components that can operate in the FIPS 140 mode.

For more information about each of the pods listed above, see Understanding Labels and Pods.

For information on creating a RedHat operating system with FIPS enabled, see "10.2 Federal Information Processing Standard (FIPS)" on the RedHat Customer Portal. **Note:** This link opens an external site.

Determining a Security Mode Between Components

You must determine a security mode for communication between your infrastructure components. The security mode of connected producers and consumers must be the same across all components.

The secure communication described applies only in the context of the components that relate to the Micro Focus container-based application you are using, which is specified in that application's documentation.

When possible, configure the Micro Focus components with the security mode you intend to use *before* connecting them to additional ArcSight Platform products.

To enhance security, you can configure TLS Client Authentication between components that do not utilize client username and password authentication, such as producers and consumers connecting to Transformation Hub. With TLS Client Authentication enabled, the client and the server authenticate each other to ensure that both parties involved in the communication are trusted.

Changing the Allow Plain Text, TLS Client Authentication, or FIPS-compliant TLS settings after the deployment will necessitate system downtime.

Micro Focus product documentation for ArcSight products in the table is available from the Micro Focus support community.

Unless otherwise indicated in the table below, the ArcSight Platform and the capabilities that deploy to it, communicate with each other using TLS with authentication performed in a manner appropriate for the component.

Product	Preparations Needed	TCP Ports	Supported Security Modes
Standalone ArcMC	• Be sure to use v2.9.5 or later. • Install ArcMC before the Platform installation.	• 443 • 32080	• TLS • FIPS-Compliant TLS • TLS Client Authentication
SmartConnectors and Collectors	• You can install and run SmartConnectors and ArcMC onboard connectors before you install the Platform. Or, you can install them after you deploy the Platform. • FIPS mode setup is not supported between SmartConnector v7.5 and the Platform. • Only TLS and TLS Client Authentication are supported. • FIPS mode *is* supported between Connectors v7.6 and later and the Platform.	• 9092 (Plain Text) • 9093 (TLS)	• TLS • FIPS-Compliant TLS (SC 7.6+ only) • TLS Client Authentication • Plain text

ArcSight ESM	• You can install and run ESM before you install the Platform. • You can change compact mode ESM from TLS to FIPS-compliant TLS after you install ESM. • Changing compact mode ESM from FIPS-compliant TLS to TLS requires reinstalling ESM. • In distributed mode, the ESM security mode is set at installation time. Any change between TLS and FIPS-compliant TLS requires reinstalling ESM.	• 9092 (Plain Text) • 8443	• TLS • FIPS-Compliant TLS • TLS Client Authentication
ArcSight Logger	• You can install and run Logger before you install the Platform.	• 9092 (Plain Text) • 9093 (TLS)	• TLS • FIPS-Compliant TLS • TLS Client Authentication • Plain text
ArcSight Database	• You install the ArcSight Database before the Platform.	• 9092 (Plain Text) • 9093 (TLS)	• Plain text • TLS • TLS Client Authentication • FIPS-Compliant TLS
NFS Server	• For optimal security, secure all NFS settings to allow only required hosts to connect to the NFS server.	• 2049	• Plain text
Web Browser	• By default, TLS is enabled.	• 443 • 5443 • 3000	• TLS
ArcSight Intelligence (HDFS)	Secure HDFS for Intelligence.	• 30070 (HDFS master) • 30820 (HDFS master) • 30010 (HDFS datanode) • 30210 (HDFS datanode)	• TLS • FIPS-Compliant TLS • TLS Client Authentication • Plain Text

Understanding Kubernetes Network Subnets

Kubernetes automates the deployment of its management services and the pods associated with deployed capabilities to master and worker nodes. As part of this process, it allocates a unique IP address to each service and pod.

In order to do so, Kubernetes must be provided with a reserved range of private network IP addresses for its services (service-CIDR parameter, default is 172.17.17.0/24) and a separate reserved range of private network IP addresses for pods (pod-CIDR parameter, default is 172.16.0.0/16).

The two IP ranges must not overlap, must not be allocated to other systems in the network, and are provided to Kubernetes at install time by specifying a network subnet in Classless Inter-Domain Routing (CIDR) format. CIDR notation includes an IP address, a slash ('/') character, and a network prefix (a decimal number).

The minimum useful network prefix is /24 and the maximum useful network prefix is /8. The default value is 172.16.0.0/16. For example:

```
POD_CIDR=172.16.0.0/16
```

The pod-CIDR IP range must contain an adequate number of IP addresses to accommodate the functions of all of the pods deployed to the cluster. Each node in the cluster is allocated a segment of the pod-CIDR IP range for use by the pods that are deployed to that node as determined by the pod-CIDR-subnetlen parameter.

The default value for pod-cidr-subnetlen is automatically computed depending on the value of pod-CIDR, as described below. The default value of pod-CIDR-subnetlen is expected to be adequate. However, if for some unexpected reason you find that pods on nodes run out of available IP addresses, you can set the pod-CIDR-subnetlen parameter to a value that makes more IP addresses available to each node.

POD_CIDR Prefix	POD_CIDR_SUBNETLEN defaults	POD_CIDR_SUBNETLEN allowed values
/8 to /21	/24	/(POD_CIDR prefix + 3) to /27
/22 to /24	/(POD_CIDR prefix + 3)	/(POD_CIDR prefix + 3) to /27

Smaller prefix values indicate a larger number of available addresses. The minimum useful network prefix is /27 and the maximum useful network prefix is /12. The default value is 172.17.17.0/24.

Choosing Your Installation Method

You can install ArcSight products using one of the two methods below.

- "Using the ArcSight Platform Installer (On-premises Only)" below
- "Deploying a Manual Installation (On-premises or Cloud)" on the next page

Using the ArcSight Platform Installer (On-premises Only)

ArcSight Platform Installer significantly simplifies the installation and deployment experience using automation. The installer has a prerequisites checker that verifies the OS, storage, network, and other settings are appropriate for the desired deployment. You can have the Installer tool adjust the prerequisite settings to ensure a successful deployment.

It is also capable of deploying containerized and database infrastructures in a simple, all-in-one node or in a highly-available multi-node configuration. It requires a minimum set of deployment configuration settings to describe the capabilities to deploy; master, worker, and database node host names; and login IDs and passwords. You can run the tool with this deployment configuration on a single node and it will automatically connect to all the nodes specified in the configuration in order to run the capabilities.

The download package includes a set of example deployment configurations , such as a highly-available deployment configuration of Recon.

Consider the following when deciding if the Platform Installer is a good fit to use in your environment:

- ArcSight Platform Installer has not been tested on a dual-homed network (dual or redundant connections to a single Internet Service Provider), so be careful before using it in this scenario.
- ArcSight Platform Installer is only capable of installing to an on-premise environment.
- ArcSight Platform Installer disables the option to authorize the collection suite usage data.
- Passwordless SSH access will automatically be configured between the master node, where ArcSight Platform Installer is used and all other CDF and Database nodes, so ArcSight Platform Installer can automatically perform tasks on the nodes securely without requiring passwords to be retained.
- The ArcSight Platform Installer assumes that `yum` is already installed and configured on every node. The pre-check fails if it is not.
- The ArcSight Platform Installer is capable of installing using the root user only, not sudo mode.

- If ArcSight Platform Installer is not a good fit for your environment, you can perform the installation manually.
- ArcSight Platform Installer will automatically create a dbadmin account on ArcSight DB hosts, which will have a conflict if those hosts have NIS/LDAP account management enabled on them. In such cases, disable NIS/LDAP on the ArcSight DB hosts if you wish to use the ArcSight Platform Installer. If you cannot disable NIS/LDAP on the ArcSight DB hosts, instead of using the ArcSight Platform Installer, perform the manual installation procedure.

If you have any customizations on the operating system, you might want to prepare your machines with the prerequisites for CDF and Database and perform deployment and post deployment configuration using the ArcSight Platform Installer because, when ArcSight Platform Installer automatically configures the prerequisites, it might overwrite your customizations. Information for preparing your machines manually is availability:
- Preparing Your Environment for Database
- Preparing Your Environment for CDF

Deploying a Manual Installation (On-premises or Cloud)

If you need to deploy to a cloud provider, such as Azure or AWS, or if ArcSight Platform Installer does not meet your needs, you can use a manual installation.

- On-premises (Ensure you prepare your environment before using this manual installation method.)
- Cloud deployment

Chapter 3: Creating an On-premises Deployment

This section discusses the process of preparing for and creating an on-premises deployment.

The installation process validates the infrastructure environment before performing the installation, as well as after the installation has completed.

Checklist: Creating an On-premises Deployment

Use the following checklist to create an on-premises deployment of the Platform infrastructure. This process includes installing the CDF and deploying your chosen capabilities. Perform the tasks in the listed order.

To know your storage options before you install the ArcSight Database, review "Understanding Object Storage Options for the ArcSight Database" on page 35.

	Task	See
❑	1. Complete the planning checklist	Checklist: Planning to Deploy the Platform
❑	2. Review guidance on how EDR and SysOps applications might affect the installation process	Caution for EDR/SysOps Applications
❑	3. (Conditional) Install the ArcSight Database	Installing the Database
❑	4. Prepare your on-premises environment for the CDF	Preparing Your Environment
❑	5. (Conditional) For a guided deployment, use ArcSight Platform Installer	Using ArcSight Platform Installer to Deploy
❑	6. (Conditional) For a manual deployment, install the CDF and the related components, and deploy the capabilities	Performing a Manual Deployment

❑	7. Complete the deployment process	Performing Post-deployment Configuration
❑	8. Integrate your platform into your environment	Integrating the Platform Into Your Environment
❑	9. Get the latest security fixes and enhancements	Upgrading to 22.1.2

Caution for EDR Applications

Some EDR (Endpoint Detection and Response) applications which come configured to the most secure settings by default can impede the operation or prevent the correct installation of the CDF application on a node. Micro Focus is aware of difficulties associated with the following security solutions.

- "Broadcom/Symantec AntiVirus for Linux (AVfL)" below
- "McAfee and Fireeye" on the next page
- "Caution for SysOps Applications" on the next page

Broadcom/Symantec AntiVirus for Linux (AVfL)

The Autoprotect feature of Symantec AVfL **will impact** any new as well as any running installation on a CDF node. It will prevent the creation, persistence and use of files required by Kubernetes to operate. As per the guidance from the vendors Linux experts [*link opens external web site*] it is recommended to turn off Autoprotect where performance and stability are important, as well as to consider full scans while the system is in maintenance mode.

Detection: The issue can be identified by seeing a “5” process using a significant amount of CPU when running the TOP command.

Fix: Run `checkcfg` on each node. If Autoprotect is on, please disable it.

Important: Note that the scanning engine may run as normal.

Recommendation: It is recommended that you exclude the sys, proc and tmp directories from scanning.

If installation is being performed on a system with exclusions, create another temporary (tmp) directory and add it to the exclusion list. When installing CDF, make sure to specify the new temporary directory as part of command-line arguments.

McAfee and Fireeye

An installation or upgrade performed on servers running a McAfee or Fireeye agent may cause issues with pod operations.

Detection: Impact can include disabled access to /etc/passwd.

Recommendation: Disable any McAfee agent on the servers used for installation or upgrade until the installation or upgrade is complete. After completion, you may re-enable the disabled McAfee agents.

Caution for SysOps Applications

Micro Focus recognizes the use of third-party SysOps applications such as CFEngine, Puppet, or Chef to help ensure consistent system configurations to support the operation of various applications across their network. It is critical that prerequisite components as part of the configuration of the system are not removed, reset, or altered while the system is running and during its lifecycle. Automated actions modifying the state of the configuration have often been the sources of issues and failures.

Detection: Very often users may notice that the hosts file has been repopulated with the systems' IP and FQDN.

Recommendation: Please coordinate with your appropriate internal group about the creation of a separate policy for the CDF systems, to ensure the prerequisites implemented persist in a manner where they are not altered or re-added in any way during the continued lifecycle of the system.

If you must check on the running configuration using a template which would replace or reset configuration parameters used by the CDF system, the SysOps job must run after all CDF services have been stopped.

Examples to avoid:

- Re-adding of the IP and FQDN to the hosts file when it should be present.
- Running `sysctl -w` on already persisted parameters while the system is operating.
- Specifying conflicting values for the same values (such as ip_forwarding = 1 and 0)

Preparing Your Environment

The actual installation of container-based applications on properly configured infrastructure, as described later in the product Deployment Guides, is quick and straightforward. The most complex part of the installation process is the preparation of the hosts, storage, and networking infrastructure, which is described in this topic.

The installation process includes several milestones, and each milestone includes several interdependent steps. The installation process validates the infrastructure environment before performing application installation, as well as after the installation has completed.

Before building your environment, ensure that the firewall is running on the CDF nodes.

- "Understanding Object Storage Options for the ArcSight Database" on page 35

Deploying ArcSight Platform and ESM on the Same Server

Micro Focus recommends that you install ArcSight Platform and ESM on different servers because this enables the use of ArcSight Platform high availability and provides the option to deploy additional capabilities in the future. However, if you plan to use only the ESM Command Center capability, you can install it on a single node on the same server where you installed ESM.

When you install the Platform, specify a CDF API Server Port that does not use the same port as the ESM server (default 8443). For more information about ArcSight Platform ports, see the Technical Requirements for ArcSight Platform. For example, when you use the ArcSight Platform installer, the `example-install-config-esm_cmd_center-single-node.yaml` sets the master-api-ssl-port to port 7443.

To deploy on the same server:

1. Install ESM.

   Always install ESM before you install the Platform.

2. Add the ESM https port in iptables using the following commands.
   a. To find your active zones, use the following command:

   ```
   firewall-cmd --get-active-zones
   ```

b. To add the ESM port in iptables, use the following command. By default the port number is 8443.

```
firewall-cmd --zone=public --add-port=<port_number>/tcp --permanent
```

This step enables you to access ESM externally (outside the firewall).

c. To reload the firewall so that the changes are applied, use the following command:

```
firewall-cmd --reload
```

3. Continue with the Platform preparation and deployment.

Configuring Proxy Settings

The cluster should have no access to the Internet and proxy settings (`http_proxy, https_proxy` and `no_proxy`) should not be set. However, if you need an Internet connection and you already specified a proxy server for http and https connection, then you must correctly configure `no_proxy`.

- "No Proxy Definitions " below
- "Proxy Settings Example" on the next page

No Proxy Definitions

If you have the `http_proxy` or `https_proxy` set, then the `no_proxy` definitions must contain at least the following values:

```
no_proxy=localhost, 127.0.0.1, <all Master and Worker cluster node IP
addresses>,<all Master and Worker cluster node FQDNs>,<HA virtual IP
Address>,<FQDN for the HA Virtual IP address>
```

This setting is only needed during installation or the persistent running of the cluster.

Incorrect configuration of proxy settings is a common installation issue. To verify that proxy settings are configured properly, on all master and worker nodes, run the following command and ensure the output corresponds to the recommendations.
`echo $http_proxy, $https_proxy, $no_proxy`

If the firewall is turned off, the install process generates a warning. To avoid this warning, set the CDF Install parameter `--auto-configure-firewall` to true.

You may wish to export the variables in a shell script in order to make these settings permanent.

Proxy Settings Example

Note that there may be no line breaks in your proxy settings.

```
export http_proxy="http://web-proxy.http_example.net:8080"
```

```
export https_proxy="https://web-proxy.http_example.net:8080"
```

```
export no_
proxy="localhost,127.0.0.1,node1.example.com,10.94.235.231,node2.example.com,
10.94.235.232,node3.example.com,10.94.235.233,node3.example.com,10.94.235.233
,node4.example.com,10.94.235.234,node5.swinfra.net,10.94.235.235,node6.swinfr
a.net,10.94.235.236,ha.swinfra.net 10.94.235.200"
```

Configuring DNS Settings

Ensure host name resolution through Domain Name Services (DNS) is working across all nodes in the cluster, including correct forward and reverse DNS lookups.

DNS is a critical service used by Kubernetes and you must configure it in a way that is compatible with Kubernetes so that the ArcSight Platform operates properly. There are certain DNS related issues, such as older glibc package allows maximum number of DNS search records limited to 6, that prevent the Kubernetes cluster to operate properly.

Kubernetes will configure containers to use the */etc/resolv.conf* configuration file of your nodes combined with Kubernetes specific entries. Therefore, for the cluster to operate properly, keep the */etc/resolv.conf* configuration file within the limit of two nameservers and two search records. Refer to the Kubernetes documentation on mitigating this issue with package updates and proper DNS configuration content below.

Host name resolution **must not be** performed through `/etc/hosts` file settings.

- "Understanding the Use of a Fully Qualified Domain Name (FQDN)" below
- "Configuring Secure DNS" on the next page
- "Testing Forward and Reverse DNS Lookup" on the next page
- "Running the Commands" on page 62

Understanding the Use of a Fully Qualified Domain Name (FQDN)

All master and worker nodes must be configured with a Fully Qualified Domain Name (FQDN), and must be in the same subnet. Transformation Hub uses the host system FQDN as its Kafka

`advertised.host.name`.

If the FQDN resolves successfully in the Network Address Translation (NAT) environment, Producers and Consumers will function correctly. If there are network-specific issues resolving FQDN through NAT, DNS will need to be updated to resolve these issues.

Configuration Notes

- `localhost` must **not** resolve to an IPv6 address, for example, "`::1`". The install process expects only IPv4 resolution to IP address 127.0.0.1. Any `::1` reference must be commented out in the `/etc/hosts` file.
- The Initial Master Node host name must not resolve to multiple IPv4 addresses, and this includes lookup in `/etc/hosts`.

Configuring Secure DNS

If Secure DNS is being used in the environment where the product is installed, the DNS must be configured so that the ACL allows connections from all of the following:

- Every machine in the Kubernetes cluster, master and worker nodes.
- The network address range of Kubernetes pods in Classless Inter-Domain Routing (CIDR) format. By default, this is 172.16.0.0/16.
- The network address range of Kubernetes services in Classless Inter-Domain Routing (CIDR) format. By default, this is 172.17.17.0/24.
- If the Database is being used, every machine in the Database cluster.

Testing Forward and Reverse DNS Lookup

Test that the forward and reverse lookup records for all servers were properly configured.

To test the forward lookup, run the commands on every master and worker node in the cluster and on every producer and consumer system, including:

- All master and worker nodes
- All ArcMC, Logger, and ESM hosts

Use the `nslookup` or `host` commands to verify your DNS configuration. (Do not use the `ping` command.) You must run the `nslookup` commands on every server specified in your `/etc/resolv.conf` file. Every server must be able to perform forward and reverse lookup properly and return the exact same results.

If you have a public DNS server specified in your `/etc/resolv.conf` file (such as the Google public DNS servers 8.8.8.8 or 8.8.4.4), you must remove this from your DNS configuration.

Running the Commands

Run the commands as follows. Expected sample output is shown below each command:

```
# hostname
```

For CentOS/RHEL 7.x or later, use `# hostnamectl`

```
mastern.yourcompany.com
```

```
# hostname -s
```

```
mastern.yourcompany.com
```

```
# hostname -f
```

```
mastern.yourcompany.com
```

```
# hostname -d
```

```
mastern.yourcompany.com
```

```
# nslookup mastern.yourcompany.com
```

```
Server:                         192.168.0.53
Address:                     192.168.0.53#53
Address:                        192.168.0.1
Name: mastern.example.com
```

```
# nslookup mastern
```

```
Server:192.168.0.53
Address:                                        192.168.0.53#53
Name:                                        mastern.example.com
Address: 192.168.0.1
```

```
# nslookup 192.168.0.1
```

```
Server:                                                    192.168.0.53
Address:                                                   192.168.0.53#53
1.0.168.192.in-addr.arpa name = mastern.example.com.
```

Creating the NFS Shares

NFS storage is used by all nodes in the Platform Kubernetes cluster to maintain state information about the infrastructure and to store other pertinent data.

- For high availability, the NFS server must run on a highly available device separate from the Kubernetes cluster nodes. This topic provides the information to manually configure the NFS share to be used by the Kubernetes cluster.
- If the service availability requirements of your environment do not require the NFS server to be highly available and if you plan to use ArcSight Platform Installer to automate the installation, you can also automate the configuration of the NFS server. To do so, use the NFS type `new` in the install configuration file, and skip this NFS server manual configuration topic.

For optimal security, secure all NFS settings to allow only required hosts to connect to the NFS server.

- "Understanding NFS Prerequisites" below
- "Checking for Prior Installation" on the next page
- "Installing a Missing Required Package" on the next page
- "Understanding NFS Directory Structure" on page 65
- "Exporting the NFS Configuration" on page 66
- "Testing NFS" on page 68

Understanding NFS Prerequisites

To ensure that your environment meets the prerequisites:

1. On the NFS server, ensure ports 111, 2049, and 20048 are open.
2. Ensure the required packages `rpcbind` and `nfs-utils` are installed and the related services are enabled on the NFS server.
3. Check for prior installation.
4. Install any missing required packages.

5. Enable the required services by running the following commands:

```
systemctl enable rpcbind
systemctl start rpcbind
systemctl enable nfs-server
systemctl start nfs-server
```

6. For the minimum required sizes for each of the NFS installation directories, see the "Network File System" section in the *Technical Requirements for ArcSight Platform 22.1*.

Checking for Prior Installation

To check for prior installation of these packages:

1. Set up the yum repository on your server.
2. Run the following command:

```
yum list installed <package name>
```

This command returns an exit status code where:

- 0 indicates the package is installed
- 1 indicates the package is not installed (does not check whether the package is valid)

Installing a Missing Required Package

To install a missing required package, run the following command:

```
yum -y install <package name>
```

Understanding NFS Directory Structure

To create the NFS directory structure:

1. Log in to the NFS server and create the following.

Item	Name	Specification	Example Command
GROUP	arcsight	GID of 1999	# groupadd -g 1999 arcsight
USER	arcsight	UID of 1999	# useradd -u 1999 -g 1999 -d /opt/arcsight arcsight
NFS root directory	/opt/arcsight-nfs	–	# mkdir -p /opt/arcsight-nfs

If you have previously installed any version of CDF, you must remove all NFS shared directories from the NFS server before you proceed. To do this, run the following command for each directory: `rm -rf <path to shared directory>`

2. For each directory listed in the table below, run the following command to create each NFS shared directory.

```
mkdir -p <path to shared directory>
```

For example:

```
mkdir -p /opt/arcsight-nfs/itom-vol
```

Directory	Mount Point Example
<NFS_root_DIRECTORY>/itom-vol	/opt/arcsight-nfs/itom-vol
<NFS_root_DIRECTORY>/db-single-vol	/opt/arcsight-nfs/db-single-vol
<NFS_root_DIRECTORY>/db-backup-vol	/opt/arcsight-nfs/db-backup-vol
<NFS_root_DIRECTORY>/itom-logging-vol	/opt/arcsight-nfs/itom-logging-vol
<NFS_root_DIRECTORY>/arcsight-volume	/opt/arcsight-nfs/arcsight-volume

3. The permission setting of each parent directory and each subdirectory must be recursively set. If it is not, run the following command to update the permissions:

```
chmod -R <path to shared directory>
```

For example:

```
chmod -R 755 /opt/arcsight-nfs
```

4. Set the ownership in this structure to UID 1999 and GID 1999. Change the directory to `/opt`, then run the following command:

```
chown -R 1999:1999 <NFS_root_DIRECTORY>
```

If you use a UID/GID different than 1999/1999, then provide it during the CDF installation in the install script arguments `--system-group-id and --system-user-id`. In addition, if you are using NetApp with NFSv4 configuration, consider applying stickybits to al <NFS_ root_directory> shares with:
`chmod g+w #chmod g+s`

Exporting the NFS Configuration

The `/etc/exports` file on the NFS server must be configured to export each volume in order for the volume to be accessible over the NFS protocol. Every master and worker node in the CDF cluster must be granted access to the NFS volume shares.

For example, this is an `/etc/exports` file for all of the NFS volumes for a cluster with 3 master and 3 worker nodes:

```
/opt/arcsight-nfs/itom-vol yourdomain-masternode1.yourenterprise.net
(rw,sync,anonuid=1999,anongid=1999,all_squash) \
yourdomain-masternode2.yourenterprise.net
(rw,sync,anonuid=1999,anongid=1999,all_squash) \
yourdomain-masternode3.yourenterprise.net
(rw,sync,anonuid=1999,anongid=1999,all_squash) \
yourdomain-workernode1.yourenterprise.net
(rw,sync,anonuid=1999,anongid=1999,all_squash) \
yourdomain-workernode2.yourenterprise.net
(rw,sync,anonuid=1999,anongid=1999,all_squash) \
yourdomain-workernode3.yourenterprise.net
(rw,sync,anonuid=1999,anongid=1999,all_squash)
/opt/arcsight-nfs/db-single-vol yourdomain-masternode1.yourenterprise.net
(rw,sync,anonuid=1999,anongid=1999,all_squash)\
yourdomain-masternode2.yourenterprise.net
(rw,sync,anonuid=1999,anongid=1999,all_squash) \
yourdomain-masternode3.yourenterprise.net
(rw,sync,anonuid=1999,anongid=1999,all_squash) \
yourdomain-workernode1.yourenterprise.net
(rw,sync,anonuid=1999,anongid=1999,all_squash) \
yourdomain-workernode2.yourenterprise.net
(rw,sync,anonuid=1999,anongid=1999,all_squash) \
yourdomain-workernode3.yourenterprise.net
```

```
(rw,sync,anonuid=1999,anongid=1999,all_squash)
/opt/arcsight-nfs/db-backup-vol yourdomain-masternode1.yourenterprise.net
(rw,sync,anonuid=1999,anongid=1999,all_squash) \
yourdomain-masternode2.yourenterprise.net
(rw,sync,anonuid=1999,anongid=1999,all_squash) \
yourdomain-masternode3.yourenterprise.net
(rw,sync,anonuid=1999,anongid=1999,all_squash) \
yourdomain-workernode1.yourenterprise.net
(rw,sync,anonuid=1999,anongid=1999,all_squash) \
yourdomain-workernode2.yourenterprise.net
(rw,sync,anonuid=1999,anongid=1999,all_squash) \
yourdomain-workernode3.yourenterprise.net
(rw,sync,anonuid=1999,anongid=1999,all_squash)
/opt/arcsight-nfs/itom-logging-vol yourdomain-masternode1.yourenterprise.net
(rw,sync,anonuid=1999,anongid=1999,all_squash) \
yourdomain-masternode2.yourenterprise.net
(rw,sync,anonuid=1999,anongid=1999,all_squash) \
yourdomain-masternode3.yourenterprise.net
(rw,sync,anonuid=1999,anongid=1999,all_squash) \
yourdomain-workernode1.yourenterprise.net
(rw,sync,anonuid=1999,anongid=1999,all_squash) \
yourdomain-workernode2.yourenterprise.net
(rw,sync,anonuid=1999,anongid=1999,all_squash) \
yourdomain-workernode3.yourenterprise.net
(rw,sync,anonuid=1999,anongid=1999,all_squash)
/opt/arcsight-nfs/arcsight-volume yourdomain-masternode1.yourenterprise.net
(rw,sync,anonuid=1999,anongid=1999,all_squash) \
yourdomain-masternode2.yourenterprise.net
(rw,sync,anonuid=1999,anongid=1999,all_squash) \
yourdomain-masternode3.yourenterprise.net
(rw,sync,anonuid=1999,anongid=1999,all_squash) \
yourdomain-workernode1.yourenterprise.net
(rw,sync,anonuid=1999,anongid=1999,all_squash) \
yourdomain-workernode2.yourenterprise.net
(rw,sync,anonuid=1999,anongid=1999,all_squash) \
yourdomain-workernode3.yourenterprise.net
(rw,sync,anonuid=1999,anongid=1999,all_squash)
```

1. Save the `/etc/exports` file, then run the following command:

   ```
   exportfs -ra
   ```

2. Synchronize the time on the NFS server and the time on the other servers in the cluster.
3. If you add more NFS shared directories later, you must restart the NFS service.

Testing NFS

Test a mount to the NFS that might be used to determine the supported version.

1. Create a test directory by running the following command:

```
mkdir /mnt/nfstest
```

2. Create a test mount by running the following command:

```
mount -t nfs -o nfsvers=4 192.168.1.15:/opt/arcsight-nfs/arcsight-volume
/mnt/nfstest
```

3. Confirm the command.
4. Remove the mount by running the following command:

```
umount /mnt/nfstest
```

Disabling Swap Space

Disabling of swap space on all master and worker nodes is necessary to evenly distribute resources and not allocate swap space.

This procedure does not apply to database nodes, because the database requires swap space. In the case where the database and Kubernetes master and worker nodes are co-located, such as an all-in-one single node deployment, you must enables swap because it is a hard requirement for the database installation. In such a single-node scenario, Kubernetes will operate properly with swap enabled because pod allocation is only to a single node, so swap does not affect the allocation logic.

To disable swap space:

1. Log on to the node.
2. Run the following command to disable the swap process.

```
# swapoff -a
```

3. Open the `/etc/fstab` file in a supported editor.
4. Comment out the lines that display `swap` as the disk type, then save the file. For example:

```
#/dev/mapper/centos_shcentos72x64-swap swap
```

Downloading the Installation Packages for an On-Premises Deployment

Use this procedure to download the packages for:

- **Installation:** Follow the "Checklist: Creating an On-premises Deployment" on page 55 to ensure a successful installation.
- **Upgrade**: Follow the "Checklist: Upgrading Your Environment" on page 511 to ensure a successful upgrade.

1. Launch a terminal session and log in to the primary master node as `root`.

   If you elect to install as a sudo user, log in to the primary master node as the non-root user.

2. To identify and access the files to download into a directory, see Downloading and Installing the ArcSight Platform Capabilities in the *Release Notes for ArcSight Platform*.
3. Unzip `arcsight-platform-installer-x.x.x.x.zip` into a directory, which we will refer toas *`{unzipped-installer-dir}`*.

   Do not unzip under `/root` or any sub-directory of `/root`..

4. Move the ArcSight Metadata file into the *`{unzipped-installer-dir}`*`/metadata/` directory.

   Do not untar the file. The filename must have the prefix `arcsight-installer-metadata`. Also, do not move signature files as it might cause warnings and errors from the installer script. Therefore, only copy the tar files you need based on what you are deploying.

5. For each ArcSight product to install and upgrade, move the corresponding image tar file into the *`{unzipped-installer-dir}`*`/images/` directory.

   Do not untar the file. Also, do not move signature files as it might cause warnings and errors from the installer script. Therefore, only copy the tar files you need based on what you are deploying.

   For example, if you deploy Transformation Hub, Fusion, and Recon, the image tar filename for each product is as follows:

Transformation Hub	`transformationhub-x.x.x.x.tar`
Fusion	`fusion-x.x.x.x.tar`

Recon	recon-x.x.x.x.tar

Installing with the sudo User Account

If you choose to run the Installer as a sudo (non-root) user, the root user must first grant the sudo user installation permission. The sudo user must have permission to execute scripts under temporary directory /tmp on all master and worker nodes.

There are two distinct file edits that need to be performed: first on the Initial Master Node only, then on all remaining master and worker nodes. These file edits are detailed below. In addition, before installing CDF, the CDF-updateRE.sh script must be modified to install CDF as a sudo user.

- "Editing the sudoers File on the Initial Master Node" below
- "Editing the sudoers File on the Remaining Master and Worker Nodes" on the next page
- "Modifying the cdf-updateRE.sh Script" on page 72
- "Installing the ArcSight Database as a sudo User" on page 73

The arcsight-install tool is not supported in sudo mode. Only manual installations are supported in sudo mode.

Editing the sudoers File on the Initial Master Node

Make the following modifications **only on the Initial Master Node.**

First, log on to the initial master node as the root user. Then, using visudo, edit the /etc/sudoers file and add or modify the following lines.

In the following commands you must ensure there is, at most, a single space character after each comma that delimits parameters. Otherwise, you might receive an error similar to this when you attempt to save the file.
>>> /etc/sudoers: syntax error near line nn <<<

1. Add the following Cmnd_Alias line to the **command aliases** group in the sudoers file.

```
Cmnd_Alias CDFINSTALL = {unzipped-installer-
dir}/installers/cdf/scripts/pre-check.sh, {unzipped-installer-dir}/install, <K8S_
HOME>/uninstall.sh, /usr/bin/kubectl, /usr/bin/docker, /usr/bin/mkdir, /bin/rm,
/bin/su, /bin/chmod, /bin/tar, <K8S_HOME>/scripts/uploadimages.sh, <K8S_
HOME>/scripts/cdf-updateRE.sh, <K8S_HOME>/bin/kube-status.sh, <K8S_
HOME>/bin/kube-stop.sh, <K8S_HOME>/bin/kube-start.sh, <K8S_HOME>/bin/kube-
restart.sh, <K8S_HOME>/bin/env.sh, <K8S_HOME>/bin/kube-common.sh, <K8S_
```

```
HOME>/bin/kubelet-umount-action.sh, /bin/chown
```

For an AWS installation, the cdf- updateRE.sh script has the path: aws-byok-installer/installer/cdf-deployer/scripts/cdf-updateRE.sh

If you are specifying an alternate tmp folder using the --tmp-folder parameter, ensure that you specify the correct path to <tmp path>/scripts/pre-check.sh in the Cmnd_ Alias line.

- Replace the {unzipped-installer-dir} with the directory where you unzipped the installation package. For example, /tmp/arcsight-platform-installer-*<version>*.zip.
- Replace <K8S_HOME> with the value defined from a command line. By default, <K8S_ HOME> is /opt/arcsight/kubernetes.

2. Add the following lines to the **wheel users** group, replacing <username> with your sudo username.

```
%wheel ALL=(ALL) ALL
```

```
cdfuser ALL=NOPASSWD: CDFINSTALL
```

```
                    Defaults: <username> !requiretty
```

```
Defaults: root !requiretty
```

3. Locate the secure_path line in the sudoers file and ensure the following paths are present.

```
Defaults secure_path = /sbin:/bin:/usr/sbin:/usr/bin
```

By doing this, the sudo user can execute the showmount, curl, ifconfig and unzip commands when installing the CDF Installer.

4. Save the file.

Editing the sudoers File on the Remaining Master and Worker Nodes

Make the following modifications only **on the remaining master and worker nodes.**

Log in to each master and worker node. Then, using visudo, edit the /etc/sudoers file and add or modify the following:

In the following commands you must ensure there is, at most, a single space character after each comma that delimits parameters. Otherwise, you might get an error similar to this when you attempt to save the file. `>>> /etc/sudoers: syntax error near line nn <<<`

1. Add the following `Cmnd_Alias` line to the **command aliases** group in the sudoers file.

```
                    Cmnd_Alias CDFINSTALL = /tmp/pre-check.sh, /tmp/ITOM_Suite_
Foundation_Node/install, <K8S_HOME>/uninstall.sh, /usr/bin/kubectl,
/usr/bin/docker, /usr/bin/mkdir, /bin/rm, /bin/su, /bin/chmod, /bin/tar, <K8S_
HOME>/scripts/uploadimages.sh, <K8S_HOME>/scripts/cdf-updateRE.sh, <K8S_
HOME>/bin/kube-status.sh, <K8S_HOME>/bin/kube-stop.sh, <K8S_HOME>/bin/kube-
start.sh, <K8S_HOME>/bin/kube-restart.sh, <K8S_HOME>/bin/env.sh,<K8S_
HOME>/bin/kube-common.sh, <K8S_HOME>/bin/kubelet-umount-action.sh, /bin/chown,
/usr/bin/cp
```

   - Replace `<K8S_HOME>` which will be used from the command line. By default, `<K8S_HOME>` is `/opt/arcsight/kubernetes`.

2. Add the following lines to the **wheel users** group, replacing `<username>` with your sudo username.

```
%wheel ALL=(ALL) ALL
```

```
cdfuser ALL=NOPASSWD: CDFINSTALL
```

```
Defaults: <username> !requiretty
```

```
Defaults: root !requiretty
```

3. Locate the `secure_path` line in the sudoers file and ensure the following paths are present.

```
Defaults secure_path = /sbin:/bin:/usr/sbin:/usr/bin
```

   By doing this, the sudo user can execute the `showmount, curl, ifconfig` and `unzip` commands when installing the CDF Installer.

4. Save the file.
5. Repeat the process for each remaining master and worker node.

Modifying the `cdf-updateRE.sh` Script

In addition to the steps listed above, the following additional step is required for sudo user installation of CDF.

The `cdf-updateRE.sh` script is used in installation and other utility operations in CDF and CDF-based products (such as Transformation Hub). To install CDF as the sudo user, you must modify the script.

1. In the location where you unzip the installer archive, modify the script `{unzipped-installer-dir}/installers/cdf/scripts/cdf-updateRE.sh` file in a text editor as follows.
   - Comment out the line containing the text `exit 1`.
   - Add the following line inside the `if` block.

   ```
   export K8S_HOME=<install directory>
   ```

   For Example:

   ```
   if [[ -z "${K8S_HOME}" ]]; then
   echo "K8S_HOME not set. If running on fresh installation, please use new shell
   session"
   # exit 1
   export K8S_HOME=/opt/arcsight/kubernetes
   fi;
   ```
2. Save the file and then proceed to CDF installation as a sudo user.

Installing the ArcSight Database as a sudo User

To install the ArcSight Database as a non-root user, you must configure the operating system on the database cluster. In this way, the non-root user can run the sudo command.

1. Create the non-root user for all nodes in the cluster.
2. Add the non-root user to the `/etc/sudousers` file on each node:

   ```
   <non_root_userid> ALL=(ALL) ALL
   ```
3. In the `/etc/ssh/sshd_config` file on each node, complete the following steps:
   a. To disable root ssh remote login, change *PermitRootLogin* to `no`:

      ```
      PermitRootLogin no
      ```
   b. To restart sshd, add the following line:

      ```
      systemctl restart sshd
      ```

Using ArcSight Platform Installer

You can use ArcSight Platform Installer to build your environment. ArcSight Platform Installer takes care of the end-to-end installation process, which starts from configuring the prerequisites to completing the post-installation configurations.

Using the Configuration Files

The Platform Installer requires a `.yaml` configuration file to determine which capabilities to deploy on which nodes and how to configure the capabilities. The installation package includes example `.yaml` files with pre-configured scenarios to help you build your configuration file.

The .yaml files are available by default in the `{unzipped-installer-dir}/config` folder. To help you understand the settings that you might want to add, modify, or remove in your chosen `.yaml` file, review the `install-config-doc.yaml`, which is also in the `/config` folder. Do not use the `install-config-doc.yaml` file as your configuration file. Rather, choose one of the example files. Each example has placeholders for your specific environment, such as host names, so you will need to edit the example file before using it. For more information on the examples, see Configuring the Deployed Capabilities.

For example, to deploy ESM Command Center and Transformation Hub in a high-availability environment, start with the file `example-install-config-esm_and_transformation_hub-high_availability.yaml`.

The "suite > config-params" section of the example deployment configuration `.yaml` files include the internal ID of configuration properties that cannot be configured easily after installation. For a description of each property internal ID in the example deployment configuration `.yaml` files, see Configuring the Deployed Capabilities. After installation, you can easily configure most properties (those not in the example deployment configuration .yaml files) using the CDF Management Portal, where descriptions for all properties are supplied as tooltips.

You can start from any of these example files:

In the example files below, SSL, FIPS, and client-auth are all enabled by default.

Configuration	Example File	Deployment Scenario
ArcSight ESM Command Center and Transformation Hub with high availability	example-install-config-esm_and_ transformation_hub-high_availability.yaml	Provides a good starting point if you anticipate your needs will grow since this configuration allows for further scaling when you need it without having to reinstall. Configures all components required by ESM Command Center on a single node, including Fusion and (optionally) SOAR, plus Transformation Hub, across 3 worker and 3 master nodes.
ArcSight ESM Command Center on a single node	example-install-config-esm_cmd_center-single-node.yaml	Installs all components required by ESM Command Center on a single node, including Fusion and (optionally) SOAR.
Intelligence with high availability	example-install-config-intelligence-high_ availability.yaml	Configures all components required by Intelligence including Fusion and Transformation Hub across 3 worker and 3 master nodes. The Database has 3 nodes with data replication enabled (1 original, 1 copy) so that it can tolerate a failure of a single node and remain operational.
Intelligence with high availability on the ArcSight Database	example-install-config-intelligence-scale_ db.yaml	Supports an environment with modest EPS and minimal number of nodes but allows for futher scaling with multiple worker nodes. Configures all components required by Intelligence on a single node, including Fusion and Transformation Hub, across 3 worker nodes and 1 master node. The Database has 3 nodes with data replication enabled (1 original, 1 copy) so that it can tolerate a failure of a single node and remain operational.
Intelligence on a single node	example-install-config-intelligence-single-node.yaml	Configures all components required by Intelligence on a single node, including Fusion and Transformation Hub. The Database has 3 nodes with data replication enabled (1 original, 1 copy) so that it can tolerate a failure of a single node and remain operational.
Intelligence and Recon on a single node	example-install-config-intelligence_and_ recon-single-node.yaml	Configures all components required by Intelligence and Recon on a single node, including Fusion and Transformation Hub. The Database resides on a separate node.

Configuration	Example File	Deployment Scenario
Recon with high availability	example-install-config-recon-high_availability.yaml	Provides a good starting point if you anticipate your needs will grow since this configuration allows for further scaling when you need it without having to reinstall. Configures all components required by Recon, including Fusion, Transformation Hub, and (optionally) SOAR, across 3 worker and 3 master nodes. The Database has 3 nodes with data replication enabled (1 original, 1 copy) so that it can tolerate a failure of a single node and remain operational.
Recon with high availability on the ArcSight Database	example-install-config-recon-scale_db.yaml	Provides a good starting point when you want to scale the Database beyond a single node to handle your workload and storage requirements, but you don't yet wish to invest in high availability for Recon. Configures all components required by Recon on a single node, including Fusion, Transformation Hub, and (optionally) SOAR. The Database has 3 nodes with data replication enabled (1 original, 1 copy) so that it can tolerate a failure of a single node and remain operational.
Recon on a single node	example-install-config-recon-single-node.yaml	Configures all components required by Recon on a single node, including Fusion, Transformation Hub, and (optionally) SOAR. The Database resides on a separate node. For information about FIPS mode on the Database Server, see Enabling FIPS Mode on the Database Server.
Transformation Hub with high availability	example-install-config-transformation_hub_and_fusion-high_availability.yaml	Configures Fusion and Transformation Hub across 3 worker and 3 master nodes.

Understanding the Installation Commands

This table provides information about the installation commands and their purpose.

These instructions use the primary commands with defaults for the most straightforward installation experience. Additional options are available if needed and are explained when you run the command `./arcsight-install --help`.

Script	Purpose
`./arcsight-install -c /opt/my-install-config.yaml --cmd preinstall`	The preinstall command attempts to install automatically any missing operating system package dependencies using the yum command. Therefore, be sure yum is configured on all nodes to automatically be able to download the packages from a package repository. It runs checks on all hosts specified in the install config file and reports if they meet the requirements. It also modifies the configuration of all hosts specified in the install config file so each host meets the required system configuration for the components that will be installed on each host. Not all required system configurations can be handled by this command. The items that must be manually configured will be reported. It also installs or configures NFS as specified in the install config file.
`./arcsight-install -c /opt/my-install-config.yaml --cmd install`	The install command installs or configures the Database, Container Deployment Foundation (CDF) cluster, and ArcSight capabilities as specified in the install config file.
`./arcsight-install -c /opt/my-install-config.yaml --cmd postinstall`	The postinstall command performs the post-installation configurations.

Configuring the System Clock of the Database Nodes

A network time server must be available in your environment. The **chrony** process implements this protocol and it is installed by default on some versions of RHEL and CentOS. Ensure that chrony is installed on every node using. Click here for more information.

CentOS 8.4 only

For all database nodes running CentOS version 8.4, you need to run this command to set the time to UTC:

```
sudo timedatectl set-timezone UTC
```

Using ArcSight Platform Installer to Deploy

ArcSight Platform Installer takes care of the prerequisites, software installations, and post-installation configurations.

Before building your environment, ensure the firewall is running on the CDF nodes.

To copy the metadata file and the images to their corresponding directories, see "Downloading the Installation Packages for an On-Premises Deployment" on page 512.

To use the installer to deploy:

1. Launch a terminal session and log in to the master node as `root`.
2. Change to the following directory:

   ```
   cd {unzipped-installer-dir}/config/
   ```

3. Select an example install config file in the directory that most closely matches the deployment you need.

   There is an explanation at the top of each example file and additional explanations are available in the `{unzipped-installer-dir}/config/` directory. Do not use the `install-config-doc.yaml` file for your deployment, as it is for information purposes only.

4. Make a copy of the selected example file. For example, in these instructions, we will name the copy the following:

   ```
   /opt/my-install-config.yaml
   ```

5. Edit the following file as needed:

   ```
   /opt/my-install-config.yaml
   ```

   Each example install config file explains the minimal changes that must be made before performing the installation with the example file.

   Depending on your specific deployment, you might need to make additional modifications that are not described in the example file. Additional explanations are available in the `{unzipped-installer-dir}/config/install-config-doc.yaml` file.

6. Change to the following directory:

   ```
   {unzipped-installer-dir}
   ```

7. Execute the following command to check all the nodes and deploy all the prerequisites.

```
./arcsight-install -c /opt/my-install-config.yaml --cmd preinstall
```

When you execute the script, the installer prompts you for the username and password you provided for each hostname specified. You need to provide this information only once for each hostname. The installer sets up secure passwordless ssh using certificates so executing commands later is seamless.
Valid password specifications include:
Length: between 8-30
Can contain: letters, digits and special characters
Valid special characters: _ ! % @ &
Valid examples: 9badm1N_X, my6AsW@rd, mypasS_w0?d

8. Execute the command to install the Database, CDF, and ArcSight capabilities.

```
./arcsight-install -c /opt/my-install-config.yaml --cmd install
```

Database

If your install config file specifies to install the Database, the installer displays prompts for:

- Accept License Agreement
- Database admin password
- Database app admin password
- Database search username

Be patient as the Database installation might take time to complete. The Database might need time to create indexes and complete setup tasks. The Database installation might appear to be complete; however, if you start the product before the Database installation is complete, you might experience errors and performance issues.

In the initial 22.1.0 release, the ArcSight Database does not support FIPS mode due to a defect. A fix is already being worked on and will be released soon after 22.1.0. In the mean time, for the database to function properly, you must disable FIPS mode on the database server.

CDF and ArcSight Capabilities

Next, the installer displays prompts for:

- Accept License Agreement (again)

If the installer discovers warnings while running a check of the node hardware configuration, a prompt appears asking you to confirm the warnings and continue.

- CDF admin password

> Be patient as the installation might take time to complete, depending on the number of suite products and cluster nodes being installed. For example, a small cluster might take 40 minutes or more to complete. You can monitor the progress of the installer in the terminal.

9. After the install command completes, run the pod command to check the pod status. Before continuing to the post-installation step, all pods must be in `Running` or `Completed` status.

```
kubectl get pods -A
```

10. View additional cluster status, including logs (as needed).
    a. Log in to the CDF Management Portal using the CDF admin username and password you provided.
    b. Navigate to **Cluster** > **Dashboard**.
    c. In the Kubernetes Dashboard, select **Namespace arcsight-installer-***.
    d. Navigate to pods, then select the pod to inspect.
    e. To view the logs for the pod, click the **View Logs** icon in the upper-right corner of the UI.
    f. In the **Logs from** menu, select a different container to view relevant logs.
11. Execute the following command to perform the post-installation configurations.

```
./arcsight-install -c /opt/my-install-config.yaml --cmd postinstall
```

12. When you run this command, the installer displays the following prompt:

```
Are you sure all arcsight pods are running and you want to continue? (y/N)
```

13. After ensuring that all the ArcSight pods are running, specify **y**.
14. Continue to "Performing Post-deployment Configuration" on page 377.

Updating RE Certificates (optional)

It's optional, but we recommend that you use an RE certificate signed by your Trusted Certificate Authority as part of the installation process. For more information, see:

- Understanding the ArcSight Platform Certificate Authorities
- Using an RE External Communication Certificate Signed by Your Trusted Certificate Authority

Performing a Manual Deployment

This section explains how to set up your deployment architecture for the Platform that runs on-premises, such as on a local network.

Checklist: Manually Installing the Platform Infrastructure

Use the following checklist to install and configure the Platform infrastructure. Perform the tasks in the listed order.

	Task	See...
❑	1. Complete the Planning Checklist.	Checklist: Planning to Deploy the Platform
❑	2. Prepare your on-premises environment for the CDF.	Preparing Your Environment
❑	3. (Conditional) To deploy Intelligence or Recon, install the ArcSight Database.	Installing the Database
❑	4. Install CDF.	Configuring and Running the CDF Installer
❑	5. Deploy the Platform and capabilities.	Deploying ArcSight Platform and Capabilities
❑	6. Configure the database.	Completing Database Setup
❑	7. Complete the deployment process.	Performing Post-deployment Configuration
❑	8. Apply the CDF 2021.05 log4j hotfix.	Applying the CDF 2021.05 log4j Hotfix
❑	9. Get the latest security fixes and enhancements.	Upgrading to 22.1.2

Installing the Database

This section provides information about configuring the database server and installing the ArcSight Database.

Recon and Intelligence are the only capabilities that need the database currently.

Preparing the Database Nodes for Installation

Follow the applicable instructions in the sections below to prepare the database nodes for installation.

- Updating CentOS (conditional)
- Configuring Operating System Settings

Updating CentOS (conditional)

If you are deploying the database with CentOS 8.4 2105, you need to update the distros by running the commands below on all database nodes:

```
sudo dnf --disablerepo '*' --enablerepo=extras swap centos-linux-repos
centos-stream-repos
sudo dnf distro-sync
```

If the *distro repo* above is broken, update the *--enablerepo* repositories from https://www.centos.org/centos-stream.

Configuring Operating System Settings

The database requires that you manually configure several general operating system settings.

1. Provision the server with at least 2 GB of swap space.

   In case the pre-check on swap space fails after provisioned 2 GB on swap, a provision swap with 2.2 GB should solve the problem.

2. Add the following parameters to `/etc/sysctl.conf`.

Parameter	Description
`net.core.somaxconn = 1024`	Increases the number of incoming connections
`net.core.wmem_max = 16777216`	Sets the send socket buffer maximum size in bytes
`net.core.rmem_max = 16777216`	Sets the receive socket buffer maximum size in bytes

`net.core.wmem_default = 262144`	Sets the receive socket buffer default size in bytes
`net.core.rmem_default = 262144`	Controls the default size of receive buffers used by sockets
`net.core.netdev_max_backlog = 100000`	Increase the length of the network interface input queue
`net.ipv4.tcp_mem = 16777216 16777216 16777216`	
`net.ipv4.tcp_wmem = 8192 262144 8388608`	
`net.ipv4.tcp_rmem = 8192 262144 8388608`	
`net.ipv4.udp_mem = 16777216 16777216 16777216`	
`net.ipv4.udp_rmem_min = 16384`	
`net.ipv4.udp_wmem_min = 16384`	
`vm.swappiness = 1`	Defines the amount and frequency at which the kernel copies RAM contents to a swap space For more information, see Check for Swappiness.

3. Add the following parameters to `/etc/rc.local`.

The following commands assume that sdb is the data drive (i.e. /opt), and sda is the operating system/catalog drive.

Parameter	Description
`echo deadline > /sys/block/sdb/queue/scheduler`	Resolve FAIL (S0150)
`/sbin/blockdev --setra 2048 /dev/sdb`	Resolve FAIL (S0020) when database resides on `/dev/sdb`
`echo never > /sys/kernel/mm/transparent_hugepage/enabled`	
`tuned-adm profile throughput-performance`	Resolve WARN (S0140/S0141) (**CentOS only**)
`\nchmod x /etc/rc.d/rc.local`	

4. In `/etc/default/grub`, append line `GRUB_CMDLINE_LINUX` with `intel_idle.max_cstate=0 processor.max_cstate=1`. For example:

```
GRUB_CMDLINE_LINUX="vconsole.keymap=us crashkernel=auto
vconsole.font=latarcyrheb-sun16 rhgb quiet intel_idle.max_cstate=0
processor.max_cstate=1"
```

Execute the following command:

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

5. Run these commands to disable the firewall **WARN (N0010):**

   ```
   systemctl mask firewalld
   systemctl disable firewalld
   systemctl stop firewalld
   ```

   During installation, the database requires that host-based firewalls are disabled on database nodes. After installation, the host-based firewalls can be enabled and the database requires several ports to be open on the local network. We recommend for optimal performance using host-based firewalls between database nodes and a network-based firewall to protect the segment that database cluster is within. However, there is no restriction against using a network-based firewall between database nodes. When using any kind of firewall, ensure that all the database ports are available. For more information, see Firewall Considerations.

6. Set SELinux to permissive mode in `/etc/selinux/config`.

   ```
   SELINUX=permissive
   ```

   For more information, see SELinux Configuration.

7. Run this command to ensure that `rng-tools` packages are installed in all cluster nodes:

   ```
   sudo dnf install rng-tools -y
   ```

8. Set the UTC time for all cluster nodes:

   ```
   sudo timedatectl set-timezone UTC
   ```

   For CentOS 8.4, any changes to the timezone will require a cluster nodes reboot.

9. Reboot the system for your changes to take effect.

Configuring BIOS for Maximum Performance

Depending on your hardware, you might be able to access options to configure power and performance. Configure the system for maximum performance in the BIOS while the system is powering on. For example, for HPE hardware, the following setting is available.

System Configuration > BIOS/Platform Configuration (RBSU) > Power Management > HPE Power Profile > Maximum Performance

Enabling Passwordless Communication

This section describes how to configure passwordless communication from the node1 server to all of the node servers in the cluster. You can perform this procedure as a root or the sudo (non-root) user.

You must repeat the authentication process for all nodes in the cluster.

1. On the node1 server, run the `ssh-keygen` command:

   ```
   ssh-keygen -q -t rsa
   ```

2. Copy the key from node1 to all of the nodes, including node1, using the node IP address:

   ```
   ssh-copy-id -i ~/.ssh/id_rsa.pub <node_IP_address>
   ```

   For example:

   ```
   ssh-copy-id -i ~/.ssh/id_rsa.pub 11.111.111.111
   ```

   The system displays the key fingerprint and requests to authenticate with the node server.
3. Specify the required credentials for the node.
4. The operation is successful when the system displays the following message:

   ```
   Number of key(s) added: 1
   ```

5. To verify successful key installation, run the following command from node1 to the target node to verify that node1 can successfully log in:

   ```
   ssh <node_IP_address>
   ```

Modifying the System Clock

A network time server must be available. `chrony` implements this protocol and is installed by default on some versions of RHEL and CentOS. `chrony` must be installed on every node.

Verify the `chrony` configuration by using the command:

```
chronyc tracking
```

To install chrony, start the chrony daemon, then verify operation with these commands:

```
dnf install chrony
systemctl start chronyd
systemctl enable chronyd
chronyc tracking
```

After modifying the system clock, you must reboot each node by running the reboot command.

Determining FIPS Configuration

To enable or disable FIPS, follow the steps in "Enabling FIPS Mode on the Database Server" on page 709

Configuring and Installing the Database Server

Before installing the database, ensure that you estimate the storage needed for the incoming EPS (event per second) and event size, and also evaluate the retention policy accordingly.

1. On the Database cluster node1 server, create a folder for the database installer.
   For example:

   ```
   mkdir /opt/arcsight-db-tools
   ```

   /opt/arcsight-db-tools should not be under /root or /opt/vertica.

2. From the master node where you performed the Downloading Installation Packages steps, copy the following directory on the Database cluster node1 server:

   {unzipped-installer-dir}/installers/database/db-installer_x.x.x-x.tar.gz file to the /opt/arcsight-db-tools

3. To extract the installer file and place it in the correct directory, run the following commands:

   ```
   cd /opt/arcsight-db-tools
   tar xvfz db-installer_x.x.x.x.tar.gz
   ```

4. Edit the config/db_user.properties file and add all database node IPs to the hosts property.

Property	Description
hosts	A comma separated list of the database servers in IPv4 format (for example, 1.1.1.1,1.1.1.2,1.1.1.3). If it is necessary to construct the cluster, avoid using local loopback (localhost, 127.0.0.1, etc.).

5. Install the database.

```
./db_installer install
```

6. When prompted, create the **database administrator** user.

   The database administrator user account is used during database deployment, configuration, upgrade, and debugging. For security reasons, the platform deployed capabilities will not ask you for the credentials for this user.

```
----------------------------------------
Please specify a username for [ DB Admin ] user:
dbadmin

----------------------------------------
Please specify a password for [ DB Admin ] user:
********
Re-enter password:
********
```

   For a list of options that you can specify when installing the database, see Understanding the Database Installer Options.

7. Specify the shard count. We recommend a shard count of 3 for single-node, or a count of 18 for multi-node to allow for scalability. The prompt options are based on your environment, single-node or multi-node:

   Once the database is installed, this value cannot be changed.

   - Single-node:

```
# ========================================
# STEP 1: Specify Database Shard Count for Eon Mode
Do you plan to keep the database cluster to a single node in the
future?
If yes, the database will be optimized for performance on a single node
by setting the default shard count to 3.
Shard Count [3]:
Shard count cannot be changed after installation.
Confirm shard count [3]?(y/n):y
```

```
Check memory size, 48GB required for single node installation with
shard count > 3.
PASS: Single node installation for shard count: 3
```

- Multi-node:

```
# ======================================
# STEP 1: Specify Database Shard Count for Eon Mode
Recommended shard count for multi node database deployment is 18.
Shard Count [18]:
Shard count cannot be changed after installation.
Confirm shard count [18]?(y/n):y
```

8. Set up the communal storage type for S3 when prompted. For example:

```
# ======================================
# STEP 2: Specify communal storage details
Supported communal storage types -
1) S3
2) Azure Blob Storage
Choose a communal storage type from the above (1/2):1
Specify S3 server:<IP address>
Specify S3 server port (1-65535):9000
Specify S3 server access key:<access_key>
Specify S3 server password:
Specify AWS region (leave empty for MinIO):
Is TLS enabled(y/n):y
Specify S3 bucket for communal storage:<yourS3BucketName>
Specify the folder under bucket for communal storage if
applicable:<newFolderNametoCreate>
```

9. Create the schema.

```
./db_installer create-schema
```

10. When prompted, create the following users:
    - **App admin user:** A regular database user granted elevated permissions for performing operations on the database to manage the database, schema, and resource pools. The credentials for this user will need to be provided later in the CDF Management Portal when you are deploying capabilities.
    - **Search user:** A regular database user with permissions restricted to event search operations. The credentials for this user will need to be provided later in the CDF Management Portal when you are deploying capabilities.

11. Monitor your database cluster status constantly. For more information, see Monitoring the Database.
    - **Database nodes status:** Ensures all nodes are up
    - **Database nodes storage status:** Ensures storage is sufficient

Installing CDF

Checklist: Preparing Your Environment for CDF

Use the following checklist to prepare your ArcSight Platform environment. Perform the tasks in the listed order.

	Task	See...
❑	1. Verify your firewall settings.	Ensuring Your Firewall Settings
❑	2. Verify masquerade settings.	Enabling Masquerade Setting in Firewall
❑	3. Verify the system clock.	Modifying the System Clock
❑	4. Verify password settings.	Checking Password Authentication Settings
❑	5. Verify OS packages are installed.	Ensuring Required OS Packages Are Installed
❑	6. Verify algorithms.	Checking MAC and Cipher Algorithms
❑	7. Set system parameters	Setting System Parameters (Network Bridging)
❑	8. Review example files.	Understanding Example Files
❑	9. Remove libraries.	Removing Libraries
❑	10. Configure settings.	Configuring Elasticsearch Settings

Preparing Your Environment

Checking Your Firewall Settings

Ensure that the `firewalld.service` is enabled and running on all nodes.

```
# systemctl start firewalld
```

```
# systemctl enable firewalld
```

Enabling the Masquerade Setting in the Firewall

You must enable the masquerade setting only when the firewall is enabled.

Run the following command on all master and worker nodes to check whether the masquerade setting is enabled:

```
# firewall-cmd --query-masquerade
```

- If the returned value is `yes`, the masquerade setting is enabled.
- If the returned value is `no`, run the following commands to enable the masquerade setting in the firewall.

```
# firewall-cmd --add-masquerade --permanent
# firewall-cmd --reload
```

Modifying the System Clock

A network time server must be available. chrony implements this protocol and is installed by default on some versions of RHEL and CentOS. chrony must be installed on every node.

Verify the chrony configuration by using the command:

```
# chronyc tracking
```

To install chrony, start the chrony daemon, then verify operation with these commands:

```
# yum install chrony
# systemctl start chronyd
# systemctl enable chronyd
# chronyc tracking
```

Checking Password Authentication Settings

If you use a user name and password authentication for adding cluster nodes during the installation, ensure that the PasswordAuthentication parameter in the /etc/ssh/sshd_config file is set to "yes."

There is no need to check the password authentication setting when you add the cluster nodes using a user name and key authentication.

To ensure the password authentication is enabled, perform the following steps on every master and worker node:

1. Log on to the cluster node.
2. Open the following file:

   ```
   /etc/ssh/sshd_config
   ```

3. Check if the parameter PasswordAuthentication is set to yes. If not, set the parameter to yes as below.

   ```
   PasswordAuthentication yes
   ```

4. Run the following command to restart the sshd service:

   ```
   systemctl restart sshd.service
   ```

Ensuring That Required OS Packages Are Installed

The packages listed in the following table are required on one or more node types, as shown here. These packages are available in the standard `yum` repositories.

Additional Information

- tar is required for tar images. If you do not have tar installed, the following error displays during installation:

  ```
  2020-12-22T20:37:47.380684729-06:00 FATAL The metadata package metadata/arcsight-installer-metadata-22.1.0.16.tar does not have the correct internal structure. Refer to /tmp/install.20201222203742.log file for detail information. If need, please contact system administrator or Micro Focus support.
  ```

- Below are yum example lines including all the required packages for each node type.
  - **Master Nodes**

    ```
    # yum install conntrack-tools container-selinux curl device-mapper-libs httpd-tools java-1.8.0-openjdk openssl libgcrypt libseccomp libtool-libs libtool-ltdl lvm2 net-tools nfs-utils rpcbind socat systemd-libs unzip bind-utils tar
    ```

  - **Worker Nodes**

    ```
    # yum install conntrack-tools container-selinux curl device-mapper-libs httpd-tools libgcrypt openssl libseccomp libtool-libs libtool-ltdl lvm2 net-tools nfs-utils rpcbind socat systemd-libs unzip tar
    ```

  - **NFS**

    ```
    yum install nfs-utils rpcbind
    ```

Package Name	Required by Master Nodes?	Required by Worker Nodes?	Required by NFS Server?
`conntrack-tools`	Yes	Yes	No
`container-selinux` (package version 2.74 or later)	Yes	Yes	No
`curl`	Yes	Yes	No
`device-mapper-libs`	Yes	Yes	No

Package Name	Required by Master Nodes?	Required by Worker Nodes?	Required by NFS Server?
`httpd-tools`	Yes	Yes	No
`java-1.8.0-openjdk`	Yes	No	No
`libgcrypt`	Yes	Yes	No
`libseccomp`	Yes	Yes	No
`libtool-ltdl`	Yes	Yes	No
`lvm2`	Yes	Yes	No
`net-tools`	Yes	Yes	No
`nfs-utils`	Yes	Yes	Yes
`rpcbind`	Yes	Yes	Yes
`socat`	Yes	Yes	No
`systemd-libs` (version >= 219)	Yes	Yes	No
`unzip`	Yes	Yes	No
`bind-utils`	Yes	Yes	No
`openssl`	Yes	Yes	No

If `bash-completion` is not installed as a package on nodes, a warning is shown. However, the `bash-completion` package is not required.

To check for prior installation of any of these packages:

1. Set up the `yum` repository on your server.
2. Run this command:

```
# yum list installed <package name>
```

3. This command returns an exit status code where:

   `0` indicates the package is installed

   `1` indicates the package is not installed (does not check whether the package is valid)

To install a required package:

Run the following command:

```
# yum -y install <package name>
```

Checking MAC and Cipher Algorithms

Ensure that the `/etc/ssh/sshd_config` files on every master and worker nodes are configured with at least one of the following values, which lists all supported algorithms. Add only the algorithms that meet the security policy of your organization.

To verify configurations:

- For MAC algorithms:

```
hmac-sha1,hmac-sha2-256,hmac-sha2-512,hmac-sha1-96
```

- For Cipher algorithms:

```
3des-cbc,aes128-cbc,aes192-cbc,aes256-cbc,aes128-ctr,aes192-ctr,aes256-
ctr,arcfour128,arcfour256,blowfish-cbc
```

For example, you could add the following lines to the `/etc/ssh/sshd_config` files on all master and worker nodes:

```
MACs hmac-sha2-256,hmac-sha2-512
Ciphers aes128-cbc,aes192-cbc,aes256-cbc,aes128-ctr,aes192-ctr,aes256-ctr
```

Setting System Parameters (Network Bridging)

1. Log in to the node.
2. Run the following command:

```
# echo -e "\nnet.bridge.bridge-nf-call-ip6tables=1\nnet.bridge.bridge-nf-call-
iptables=1" >> /etc/sysctl.conf
```

3. Run the following command:

```
echo "br_netfilter" > /etc/modules-load.d/br_netfilter.conf
```

4. Run the following commands:

```
# modprobe br_netfilter && sysctl -p
# echo -e '\nmodprobe br_netfilter && sysctl -p' >> /etc/rc.d/rc.local# chmod +x
/etc/rc.d/rc.local
```

5. Open the following file in a text editor:

```
/etc/sysctl.conf
```

6. (Conditional) If installing on RHEL or CentOS earlier than version 8.1, change the following if the line exists.

```
net.ipv4.tcp_tw_recycle=1 to net.ipv4.tcp_tw_recycle=0
```

7. (Conditional) If installing on RHEL or CentOS 8.1 or later, remove or comment out this line, if it exists.

```
net.ipv4.tcp_tw_recycle=
```

8. Save your changes and close the file.
9. Run this command to apply your updates to the node:

```
# sysctl -p
```

Understanding Example Files

To view example files:

Example `sysctl.conf` file for RedHat/CentOS version 7.x:

```
net.bridge.bridge-nf-call-iptables=1
net.bridge.bridge-nf-call-ip6tables=1
net.ipv4.ip_forward=1
net.ipv4.tcp_tw_recycle=0
kernel.sem=50100 128256000 50100 2560
```

Example `sysctl.conf` file for RedHat/CentOS 8.1 or later:

```
net.bridge.bridge-nf-call-iptables=1
net.bridge.bridge-nf-call-ip6tables=1
net.ipv4.ip_forward=1
kernel.sem=50100 128256000 50100 2560
```

Removing Libraries to Prevent Ingress

You must remove any libraries that will prevent ingress from starting.

1. Run the following command:

```
# yum remove rsh rsh-server vsftpd
```

2. Confirm the removal when prompted.

Configuring Elasticsearch Settings

This procedure applies only when you are deploying the Intelligence capability.

To ensure the Elasticsearch pods run after deployment and the Elasticsearch cluster is accessible:

1. Launch a terminal session and log in to a worker node.
2. Change to the following directory:

```
cd /etc/
```

3. In the `sysctl.conf` file, add the following:

```
vm.max_map_count=262144
```

4. Restart the node:

```
reboot
```

5. Repeat steps 1-4 on all worker nodes.

Configuring and Running the CDF Installer

After the installation packages have been downloaded, validated, and uncompressed in the download folder, you are ready to configure and run the CDF Installer. For a complete list of optional parameters, see CDF Installation CLI Commands.

To configure and run the CDF Installer:

1. Log in to one of the local master nodes where you downloaded and extracted the installation files as the root user. (In this document, the selected master node is referred to as the Initial Master Node. You initiate installations from the Initial Master Node.)

If you choose to install as a sudo user, log in to the master node as the non-root user. Sudo user installation is supported only for manual deployments.

2. Run the CDF Installer on the Initial Master Node with the following commands.

If you choose to install as a sudo user, execute this install command using the sudo command.

In the following commands, the *italicized* Docker parameters are optional, based on your network environment.

```
cd {unzipped-installer-dir}/installers/cdf
./install -m <path_to_a_metadata_file> --k8s-home <path_to_installation_
directory> --nfs-server <your_nfs_server_FQDN or IP Address> --nfs-folder <itom_
volume_folder>
--docker-http-proxy <your_docker_http_proxy_value> --docker-https-proxy <your_
docker_https_proxy_value> --docker-no-proxy <your_docker_no_proxy_value>
--ha-virtual-ip <your_HA_ip> --tmp-folder <your_temp_folder>
```

3. You are prompted for a password, which will be used to log in to the CDF installer portal. For example:

```
cd /opt/arcsight/download/installers/cdf/arcsight-platform-installer-x.x.x.x
./install -m /opt/arcsight/download/arcsight-platform-installer-
xxxxx/metadata/arcsight-installer-metadata-x.x.x.x.tar --k8s-home
/opt/arcsight/kubernetes --docker-http-proxy "http://web-proxy.example.com:8080"
--docker-https-proxy "http://web-proxy.example.com:8080" --docker-no-proxy
"localhost,127.0.0.1,my-vmenv-node1,my-vmenv-
node1.example.com,example.com,216.3.128.12"
--nfs-server yourdomain-nfs.yourenterprise.net --nfs-folder /opt/arcsight-
nfs/itom-vol --ha-virtual-ip 216.3.128.12 --tmp-folder /opt/tmp
```

4. You might need to configure some additional parameters, depending on your organization's OS, network, and storage configurations.

After the CDF Installer is configured and installed, you can use it to deploy one or more capabilities into the cluster.

CDF Manual Installer Script `install` Command Line Arguments

Argument	Description
`--auto-configure-firewall`	Flag to indicate whether to auto configure the firewall rules during node deployment. The allowable values are true or false. The default is true.
`--cluster-name`	Specifies the logical name of the cluster.
`--deployment-log-location`	Specifies the absolute path of the folder for placing the log files from deployments.
`--docker-http-proxy`	Proxy settings for Docker. Specify if accessing the Docker hub or Docker registry requires a proxy. By default, the value will be configured from the `http_proxy` environment variable on your system.
`--docker-https-proxy`	Proxy settings for Docker. Specify if accessing the Docker hub or Docker registry requires a proxy. By default, the value will be configured from `https_proxy` environment variable on your system

Argument	Description
`--docker-no-proxy`	Specifies the IPv4 addresses or FQDs that do not require proxy settings for Docker. By default, the value will be configured from the `no_proxy` environment variable on your system.
`--enable_fips`	This parameter enables suites to enable and disable FIPS. The expected values are true or false. The default is *false*.
`--fail-swap-on`	If 'swapping' is enabled, specifies whether to make the kubelet fail to start. Set to true or false. The default is *true*.
`--flannel-backend-type`	Specifies flannel backend type. Supported values are vxlan and host-gw. The default is host-gw. **Note:** The arcsight-install tool has two prechecks that will validate if Layer 2 connectivity is present between nodes. If not, the tool will recommend to use vxlan for flannel back end type.
`--ha-virtual-ip`	A Virtual IP (VIP) is an IP address that is shared by all master nodes. The VIP is used for the connection redundancy by providing failover for one machine. Should a master node fail, another master node takes over the VIP address and responds to requests sent to the VIP. Mandatory for a Multi-Master cluster; not applicable to a single-master cluster The VIP must be resolved (forward and reverse) to the VIP Fully Qualified Domain Name (FQDN)
`--k8s-home`	Specifies the absolute path of the directory for the installation binaries. By default, the Kubernetes installation directory is /opt/arcsight/kubernetes.
`--keepalived-nopreempt`	Specifies whether to enable nopreempt mode for KeepAlived. The allowable value of this parameter is true or false. The default is true and KeepAlived is started in nopreempt mode.
`--keepalived-virtual-router-id`	Specifies the virtual router ID for KEEPALIVED. This virtual router ID is unique for each cluster under the same network segment. All nodes in the same cluster should use the same value, between 0 and 255. The default is 51.
`--kube-dns-hosts`	Specifies the absolute path of the hosts file used for host name resolution in a non-DNS environment. **Note:** Although this option is supported by the CDF Installer, its use is strongly discouraged to avoid using DNS resolution in production environments, due to hostname resolution issues and the nuances involved in their mitigations.
`--load-balancer-host`	IP address or host name of load balancer used for communication between the master nodes. For a multiple master node cluster, it is required to provide `–load-balancer-host` or `–ha-virtual-ip` arguments.
`--master-api-ssl-port`	Specifies the https port for the Kubernetes (K8S) API server. The default is 8443.

Argument	Description
`--nfs-folder`	Specifies the path to the NFS core volume.
`--nfs-server`	Address of the NFS host.
`--pod-cidr-subnetlen`	Specifies the size of the subnet allocated to each host for pod network addresses.
`--pod-cidr`	Specifies the private network address range for the Kubernetes pods. Default is 172.16.0.0/16. The minimum useful network prefix is /24. The maximum useful network prefix is /8. This must not overlap with any IP ranges assigned to services (see `--service-cidr` parameter below) in Kubernetes. The default is 172.16.0.0/16.
`--registry_orgname`	The organization inside the public Docker registry name where suite images are located. Not mandatory. Select one of the following: • Specify your own organization name (such as your company name). For example: `--registry-orgname=Mycompany.` • Skip this parameter. A default internal registry will be created under the default name HPESWITOM.
`--runtime-home`	Specifies the absolute path for placing Kubernetes runtime data. By default, the runtime data directory is `${K8S_HOME}/data.`
`--service-cidr`	Kubernetes service IP range. Default is 172.30.78.0/24. Must not overlap the POD_CIDR range. Specifies the network address for the Kubernetes services. The minimum useful network prefix is /27 and the maximum network prefix is /12. If SERVICE_CIDR is not specified, then the default value is 172.17.17.0/24. This must not overlap with any IP ranges assigned to nodes for pods. See `--pod-cidr.`
`--skip-check-on-node-lost`	Option used to skip the time synchronization check if the node is lost. The default is true.
`--skip-warning`	Option used to skip the warnings in precheck when installing the Initial master Node. Set to true or false. The default is false.
`--system-group-id`	The group ID exposed on server; default is 1999.
`--system-user-id`	The user ID exposed on server; default is 1999.
`--thinpool-device`	Specifies the path to the Docker devicemapper, which must be in the `/dev/mapper/` directory. For example: `/dev/mapper/docker-thinpool`

Argument	Description
`--tmp-folder`	Specifies the absolute path of the temporary folder for placing temporary files. The default temporary folder is `/tmp`.
`-h, --help`	Displays a help message explaining proper parameter usage
`-m, --metadata`	Specifies the absolute path of the tar.gz suite metadata packages.

Azure CDF install Script Command Line Arguments (Optional)

Argument	Description
`-c, --config`	Absolute path of the configuration json file for silent installation.
`-d, --deployment-name, -n`	Deployment name for suite installation. (Note: -n is to be deprecated in future versions.)
`--backup-vol-size`	Specifies the volume size of pg-backup component. The size must be a plain integer or as a fixed-point integer and the unit must be one of E,P,T,G,M,K,Ei,Pi,Ti,Gi,Mi,Ki; example: 10Gi
`-fg, --feature-gates`	A set of key=value pairs that describe feature gates for alpha/experimental features. The allowable value of this parameter is mapStringBool. Comma-delimited list of strings, each entry format is NameOfFeature=true|false. Options are: • MultipleDeployment=true|false (Alpha - default=false) • Bosun=true|false (Alpha - default=false) • Prometheus=true|false (Alpha - default=false)
`--nfs-server`	Specifies the server for NFS, used to create persistent volume claim 'itom-vol-claim'
`--nfs-folder`	Specifies the folder for NFS, used to create persistent volume claim 'itom-vol-claim'.
`--loadbalancer-info`	Specifies the loadbalancer info. This parameter value formats such as: "KEY1=VALUE1;KEY2=VALUE2;...;KEYn=VALUEn" Example: For gcp: --loadbalancer-info "LOADBALANCERIP=x.x.x.x" For alicloud: --loadbalancer-info "LOADBALANCERID=xxx"
`--logging-vol-size`	Specifies the volume size of fluentd component. The size must be a plain integer or as a fixed-point integer and the unit must be one of E,P,T,G,M,K,Ei,Pi,Ti,Gi,Mi,Ki; example: 10Gi
`-P, password`	Specifies the password for suite administrator which will be created during installation. Wrap the password with single quotes. For example, 'Password@#$!123'.
`--registry-orgname`	Specifies the organization name(namespace) where the suite images are placed. The default name is 'hpeswitom'.
`--registry-ca`	Specifies the path of trusted CA root certificate (bas64 X.509 format) of external registry.
`--registry-password`	Specifies the password for registry.
`--registry-password-file`	Specifies the password file for registry.
`--skip-warning`	Option used to skip the warning(s) in precheck when install.

Argument	Description
`--tmp-folder`	Specifies the absolute path of the temporary folder for placing temporary files. The default temporary folder is '`/tmp`'.
`--db-user`	External suite database user name.
`--db-password`	External suite database password.
`--db-url`	External suite database connection URL.
`--db-crt`	External suite database connection certificate.
`--registry-url`	Specifies the registry for URL.
`--registry-username`	Specifies the username for registry.
`--external-access-host`	Specifies the external access host.
`--cloud-provider`	Specifies the cloud provider when installing CDF on a cloud server. The allowable value of this parameter is 'alicloud', 'gcp' (case-sensitive)

Securing External Communication with the RE Certificate

At the center of the Platform is a Kubernetes cluster where communication occurs between pods within the cluster and with non-containerized ArcSight components outside of the cluster. In order to ensure secure trusted communication between pods within the cluster and components outside of the cluster, encrypted communication with client certificate authentication is configured by default.

- Understanding the ArcSight Platform Certificate Authorities
- Using an RE External Communication Certificate Signed by Your Trusted Certificate Authority

Understanding the ArcSight Platform Certificate Authorities

During installation, three self-signed Certificate Authorities (CA) are created automatically, two for signing certificates used exclusively for pod to pod communication within the cluster (RIC and RID CA), and the other for signing certificates for each pod that performs communication external to the cluster (RE CA). Only pods that perform external communication have a certificate that is signed by the external CA.

External cluster communication occurs not only with ArcSight components, but also with user web browsers and, in some cases, user clients of ArcSight APIs (such as the REST API). By default, when the user connects to the cluster, they will be presented with a certificate that has been signed by the self-signed external CA. Since the external CA is self-signed, the user's connection will not automatically trust the certificate because it will not be verifiable using a certificate chain that is already in the user's trust store.

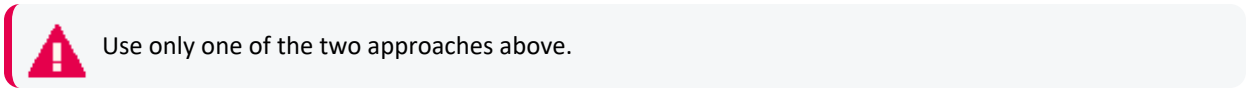
To give users confidence they are connecting to the trusted cluster, we recommend signing the certificates that are presented to the user with a CA that is trusted by the user's trust store. There are two approaches to doing this that are described in the documentation below. These approaches are:

Method 1 - Signing the RE External Communication Certificate with Your Trusted Certificate Authority

This is the recommended approach, because it is theoretically more secure than the other approach, in that, it only involves transferring a CSR and public certificate between systems, which does not put any private secrets at risk.

Method 2 - Importing an Externally Created Intermediate CA

This approach involves creating an Intermediate CA (key and certificate pair) in a system outside of the ArcSight Platform, and then importing it into the ArcSight Platform. While this approach does work, it is theoretically less secure than the other approach, because it involves transferring a CA private key between systems, which potentially exposes it to unintended parties.



Using an RE External Communication Certificate Signed by Your Trusted Certificate Authority

Use only one of the two approaches below. The first one, "Signing the RE External Communication Certificate with Your Trusted Certificate Authority" approach is recommended for the reasons described in Understanding the ArcSight Platform Certificate Authorities.

Method 1 - Signing the RE External Communication Certificate with Your Trusted Certificate Authority

Signing the RE External Communication Certificate with Your Trusted Certificate Authority approach is recommended for the reasons described in Understanding the ArcSight Platform Certificate Authorities.

In order to sign the RE external communication certificate with your trusted CA, you need to (1) create a certificate signing request (CSR) from vault, (2) take it to your organization, (3) sign it, and (4) return the signed CSR and all the public chain-of-certificates used to sign it.

1. Export the following access token dependencies (you can remove these later if not needed):

```
export CDF_APISERVER=$(kubectl get pods -n core -o custom-
columns=":metadata.name"| grep cdf-apiserver)
```

```
export PASSPHRASE=$(kubectl get secret vault-passphrase -n core -o json
2>/dev/null | jq -r '.data.passphrase')
```

```
export ENCRYPTED_ROOT_TOKEN=$(kubectl get secret vault-credential -n core
-o json 2>/dev/null | jq -r '.data."root.token"')
```

```
export VAULT_TOKEN=$(echo ${ENCRYPTED_ROOT_TOKEN} | openssl aes-256-cbc -
md sha256 -a -d -pass pass:"${PASSPHRASE}")
```

2. Ask vault to generate the CSR by running the following command:

   Important: When you execute this command, proceed expeditiously through steps 3 and 4, as your cluster will not be able to issue external certificates while it waits for the CSR to be signed.

   ```
   kubectl exec -it -n core ${CDF_APISERVER} -c cdf-apiserver -- bash -c
   "VAULT_TOKEN=$VAULT_TOKEN vault write -tls-skip-verify -format=json
   RE/intermediate/generate/internal common_name=\"none-MF CDF RE CA on
   <FQDN of ArcSight Platform Virtual IP for HA or single master node>\"
   country=<Country> locality=<Locality> province=<Province>
   organization=<Organization> ou=<Organizational Unit>" | jq -r '.data.csr'
   > /tmp/pki_intermediate.csr
   ```

   Note: The `common_name` in the command above is an example common name. Substitue your own values for the common name to fit your environment. Additionally, your trusted certificate authority might require additional parameters in the CSR besides common_name. Ask your PKI team for what the required CSR parameters are and add the appropriate parameters to the command (similar to how the parameter common_name is specified). The parameter names for the vault command used above are documented at https://www.vaultproject.io/api-docs/secret/pki#generate-intermediate

3. Sign the CSR file with your trusted certificate authority, and save the result into the `intermediate.cert.pem` file.

   Examply only. A basic example is provided below. Your environment will likely be different.

   ```
   openssl ca -keyfile your-rootca-sha256.key -cert your-rootca-sha256.crt -
   config your-openssl-configuration-file -extensions v3_ca -notext -md
   sha256 -in /tmp/pki_intermediate.csr -out intermediate.cert.pem
   ```

> Make sure the `v3_ca` extension is enabled and a new certificate is useable as a certificate authority on its own. Otherwise, you will receive a warning in the next step that given certificates are not marked for CA use.

4. Create an `intermediate.chain.pem` file that includes the combination of the `intermediate.cert.pem`, the public certificate of your trusted certificate authority, and all intermediate public certificates in the chain between them so that `intermediate.chain.pem` includes the full trust chain.

```
cp intermediate.cert.pem intermediate.chain.pem
cat [parent-intermediate1.crt] [parent-intermediate2.crt] [...] your-
rootca-sha256.crt >> intermediate.chain.pem
```

> If you have intermediate certificates between your `intermediate.cert.pem` and your trusted certificate authority, you must add the certificates in the specific order of the sequence of the chain, with the last certificate being the certificate of the root trusted CA.

5. Import the `intermediate.chain.pem` file into the cluster vault:

```
chaincerts=$(cat intermediate.chain.pem) && kubectl exec -it -n core
${CDF_APISERVER} -c cdf-apiserver -- bash -c "VAULT_TOKEN=$VAULT_TOKEN
vault write -tls-skip-verify -format=json RE/intermediate/set-signed
certificate=\"${chaincerts}\""
```

6. Update `ConfigMap RE_ca.crt` by running these commands:

```
reCrtForJson=$(sed -E ':a;N;$!ba;s/\r{0,1}\n/\\n/g'
intermediate.chain.pem) && kubectl patch configmap -n core public-ca-
certificates -p "{\"op\": \"replace\", \"data\":{\"RE_
ca.crt\":\"${reCrtForJson}\"}}"
```

```
ARCSIGHT_NS=$(kubectl get namespaces --no-headers -o custom-
columns=":metadata.name" | grep arcsight-installer)
```

```
if [ -n "$ARCSIGHT_NS" ];then reCrtForJson=$(sed -E ':a;N;$!ba;s/\r
{0,1}\n/\\n/g' intermediate.chain.pem); kubectl patch configmap -n
$ARCSIGHT_NS public-ca-certificates -p "{\"op\": \"replace\", \"data\":
{\"RE_ca.crt\":\"${reCrtForJson}\"}}";fi
```

7. (Conditional) If you already deployed ArcSight Capabilities onto the CDF, update the ArcSight Capabilities to use the updated RE external communication certificate, by following the instructions in Configuring ArcSight Kubernetes Pods to Use the Updated RE External Communication Certificate.

   If you deployed CDF but have not yet deployed any ArcSight Capabilities, you can skip those instructions.

Method 2 - Importing an Externally Created Intermediate CA

This is an alternate approach for signing certificates to connect to the trusted cluster. Before choosing this approach, ensure that your understand the other approach recommended in Understanding the ArcSight Platform Certificate Authorities.

To import an externally created intermediate CA:

1. Obtain an intermediate CA (key and certificate pair) from your trusted certificate authority.
   a. Name the certificate files as follows:
      - key file: `intermediate.key.pem`
      - certificate file: `intermediate.cert.pem`
   b. Obtain the root CA certificate (including chain), and put it in a file named ca.cert.pem.
2. Replace the existing RE CA in the ArcSight Platform with the intermediate CA you obtained in the step above, based on your type of deployment, on-premises **or** cloud.
   a. Change the directory:
      - For an on-premises deployment, run these commands:

        ```
        cd /opt/arcsight/kubernetes/scripts/
        ```

      - For a cloud deployment, run these commands:

        ```
        cd {path to cdf installer}/cdf-deployer/scripts/
        ```

   b. Run the following command to replace the existing RE CA:

      ```
      ./cdf-updateRE.sh write --re-crt=/pathto/intermediate.cert.pem --re-key=/pathto/intermediate.key.pem [--re-ca=/pathto/ca.cert.pem]
      ```

      Note: `--re-ca=/pathto/ca.cert.pem` is the path to the file containing the certificate of CA used to sign re-crt. It is not required when re-crt is self-signed or CA is included in re-crt.

3. (Conditional) If you already deployed ArcSight Capabilities onto CDF, proceed to the next section to update the ArcSight Capabilities to use the updated RE external communication certificate, Configuring ArcSight Kubernetes Pods to Use the Updated RE External Communication Certificate.

   However, if you have only deployed CDF, but have not deployed ArcSight Capabilities yet, you can skip that section.

Deploying ArcSight Platform and Capabilities

Configuring and Deploying the Kubernetes Cluster

After you run the CDF Installer, complete the following steps to deploy your Kubernetes cluster.

To configure and deploy:

1. Browse to the Initial Master Node at:

   ```
   https://{master_FQDN or IP}:3000
   ```

2. Log in using *admin* userid and the password you specified during the platform installation. (This URL appears at the successful completion of the CDF installation shown earlier.)
3. On the Security Risk and Governance - Container Installer page, choose the CDF base product metadata version. Then, click **Next**.
4. On the End User License Agreement page, review the EULA and select the *'I agree…'* check box. You might optionally choose to have suite utilization information passed to Micro Focus. Then, click **Next**.
5. On the Capabilities page, choose the capabilities and products to install, then click **Next**.

   Some capabilities might require other capabilities as prerequisites. Such requirements are noted in the pull-down text associated with the capability. To show additional information associated with the product, click the > (greater than) arrow.

6. On the Database page, ensure the PostgreSQL High Availability box is *cleared*. This database is not used by capabilities in SODP.
7. Click **Next**.
8. On the Deployment Size page, choose a size for your deployment based on your planned implementation.
   - **Small Cluster:** Minimum of one worker node deployed (each node should have 4 cores, 16 GB memory, 50 GB disk)
   - **Medium Cluster:** Minimum of 1 worker node deployed (each node should have 8 cores, 32 GB memory, 100 GB disk)
   - **Large Cluster:** Minimum of 3 worker nodes deployed (each node should have 16 cores, 64 GB memory, 256 GB disk)

The installation will not proceed until the minimal hardware requirements for the deployment are met.

You can configure additional worker nodes, with each running on its own host system, in subsequent steps.

9. Select your appropriate deployment size, then click **Next**.
10. On the Connection page, an external hostname is automatically populated. This is resolved from the Virtual IP (VIP) specified earlier during the install of CDF (`--ha-virtual-ip parameter`), or the master node hostname if the `--ha-virtual-ip` parameter was not specified during CDF installation. Confirm the VIP is correct, then click **Next**.
11. On the Master High Availability page, if high availability (HA) is desired, select **Make master highly available** and add 2 additional master nodes. (CDF requires 3 master nodes to support high availability.) When complete, or if HA is not desired, click **Next**.
12. For High Availability clusters, the installer prompts you to add additional master nodes depending on your selected deployment size. On the Add Master Node page, specify the details of your first master node and then click **Save**. Repeat for any additional master nodes.

    Master node parameters include:

    - **Host:** FQDN (only) of node you are adding.
    - **Ignore Warnings:** If selected, the installer will ignore any warnings that occur during the pre-checks on the server. If deselected, the add node process will stop and a window will display any warning messages. We recommend that you start with Ignore Warnings cleared to view any warnings displayed. You might then evaluate whether to ignore or rectify any warnings, clear the warning dialog, then click Save again with the box selected to avoid stopping.
    - **User Name:** `root` or `sudo` user name.
    - **Verify Mode:** Choose the verification mode as *Password* or *Key-based*, then either specify your password or upload a private key file. If you choose **Key-based**, you must first specify a username, then upload a private key file when connecting the node with a private key file.
    - **Device Type:** Select a device type for the master node from one of the following options.
      - Overlay 2: For production, Overlay 2 is recommended.
      - **Thinpool Device:** (Optional) Specify the Thinpool Device path, that you configured for the master node, if any. For example: `/dev/mapper/docker-thinpool`. You must have already set up the Docker thin pool for all cluster nodes that need to use thinpools.

- **Container data:** Directory location of the container data.
- **flannel IFace:** (optional) Specify the flannel IFace value if the master node has more than one network adapter. This must be a single IPv4 address (or name of the existing interface) and will be used for Docker inter-host communication.

13. On the Add Node page, add the first worker node as required for your deployment by clicking on the **+** (Add) symbol in the box to the right. The current number of nodes is initially shown in red.
14. As you add worker nodes, each Node is then verified for system requirements. The node count progress bar on the Add Node page will progressively show the current number of verified worker nodes you have configured. This progress will continue until the necessary count is met. The progress bar will turn from red to green, which indicates you have reached the minimum number of worker nodes as shown selected in Step 7, above. You might add more Nodes than the minimum number.

    Check the **Allow suite workload to be deployed on the master node** to combine master/worker functionality on the same node (Not recommended for production).

15. On the Add Worker Node dialog, specify the required configuration information for the worker node, then click **Save**. Repeat this process for each of the worker nodes you wish to add.

    Worker node parameters include:

    - **Type:** Default is based on the deployment size you selected earlier, and shows minimum system requirements in terms of CPU, memory, and storage.
    - **Skip Resource Check:** If your worker node does not meet minimum requirements, select **Skip resource check** to bypass minimum node requirement verification. (The progress bar on the **Add Node** page will still show the total of added worker nodes in green, but reflects that the resources of one or more of these have not been verified for minimum requirements.)
    - **Host:** FQDN (only) of node you are adding.

      When adding any worker node for Transformation Hub workload, on the **Add Node** page, **always** use the FQDN to specify the Node. **Do not use the IP address.**

    - **Ignore Warnings:** If selected, the installer will ignore any warnings that occur during the pre-checks on the server. If deselected, the add node process will stop and a window will display any warning messages. You might start with this deselected in order to view any warnings displayed. You may then evaluate whether to ignore or rectify any warnings, then run the deployment again with the box selected to avoid stopping.

- **User Name:** `root` or sudo user name.
- **Verify Mode:** Select a verification credential type: Password or Key-based. Then specify the actual credential.

Once all the required worker nodes have been added, click **Next**.

16. On the File Storage page, configure your NFS volumes.

(For NFS parameter definitions, refer to the section "Configure an NFS Server environment".) For each NFS volume, do the following:

- In **File Server**, specify the IP address or FQDN for the NFS server.
- On the **Exported Path** drop-down, select the appropriate volume.
- Click **Validate**.

> All volumes must validate successfully to continue with the installation.

> A *Self-hosted NFS* refers to the NFS that you prepared when you configured an NFS server environment. Always choose this value for **File System Type.**

The following volumes must be available on your NFS server.

CDF NFS Volume claim	Your NFS volume
`itom-vol-claim`	`{NFS_ROOT_DIRECTORY}/itom-vol`
`db-single-vol`	`{NFS_ROOT_DIRECTORY}/db-single-vol`
`db-backup-vol`	`{NFS_ROOT_DIRECTORY}/db-backup-vol`
`itom-logging-vol`	`{NFS_ROOT_DIRECTORY}/itom-logging-vol`
`arcsight-volume`	`{NFS_ROOT_DIRECTORY}/arcsight-volume`

17. Click **Next**.

> **Warning:** After you click **Next**, the infrastructure implementation will be deployed. *Please ensure that your infrastructure choices are adequate to your needs.* An incorrect or insufficient configuration may require a reinstall of all capabilities.

On the **Confirm** dialog, click **Yes** to start deploying master and worker nodes.

Uploading Images for the Capabilities

The **Check Image Availability** page lists the images that are currently loaded into the local Docker Registry from the originally-downloaded set of images. For an initial install, no images should be uploaded. You will upload the images now.

To upload the images to the local Docker Registry:

1. By this point, the images to be installed have already been downloaded from the Micro Focus software site, validated, and uncompressed. None of the files should require downloading at this point, so on the **Download Images** page, click **Next** to skip this step.

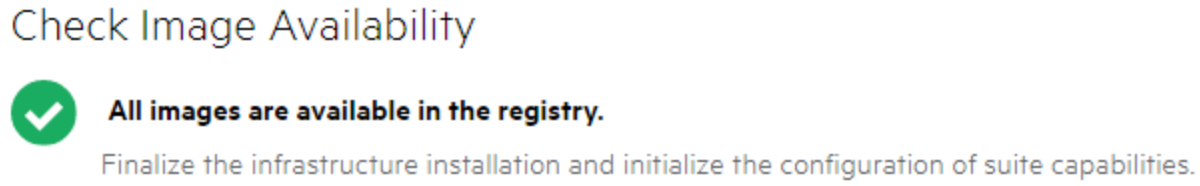
2. Log on to the Initial Master Node in a terminal session as the root or sudo user.
3. Run the following commands to upload the images to the local Docker Registry. Use the `-F <image file>` option on the command line multiple times for each image to upload. Adjust the `-c 2` option up to half of your CPU cores in order to increase the speed of the upload.

   When running the upload images script, you are prompted for the administrator password previously specified in "Configuring and Running the CDF Installer" on page 96.

   ```
   cd ${K8S_HOME}/scripts
   ```

   ```
   ./uploadimages.sh -c 2 \
   -F {unzipped-installer-dir}/images/fusion-x.x.x.x.tar \
   -F {unzipped-installer-dir}/images/recon-x.x.x.x.tar \
   -F {unzipped-installer-dir}/images/transformationhub-x.x.x.x.tar
   ```

4. The pre-deployment validation process verifies all environment prerequisites have been met before deploying.
5. To verify completion of the upload of all images, return to the CDF Management Portal's Check Availability page, and click **Check Image Availability Again**. When the *All images are available in the registry.* message displays, all required component uploads are complete.

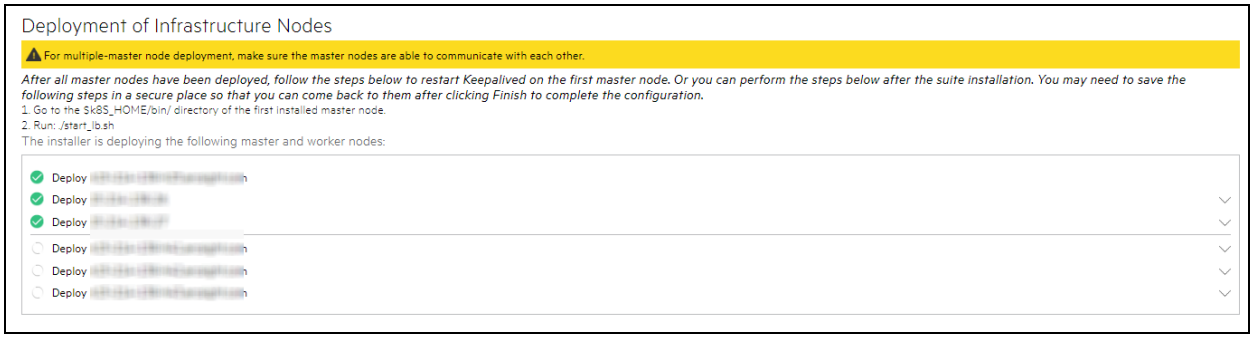


6. After verification, click **Next**.

Deploying Node Infrastructure and Services

Node Infrastructure

After the images are verified and you click **Next**, the node infrastructure is deployed. The **Deployment of Infrastructure Nodes** page will display progress.



Please be patient. Wait for all master and worker nodes to be properly deployed (showing a green check icon). Depending on the speed of your network and node servers, this can take up to 15 minutes to complete.

Clicking the **Retry** button will trigger additional communication with a problematic node, until the button converts to a spinning progress wheel. This indicates that the node deployment process is being started again. Until this occurs, refrain from clicking **Retry** again.

Monitoring Progress: You can monitor deployment progress on a node in the following ways:

- During installation, check the log on the node of interest, in `/tmp/install<timestamp>.log`. Run the command:

```
tail -f <logfilename>
```

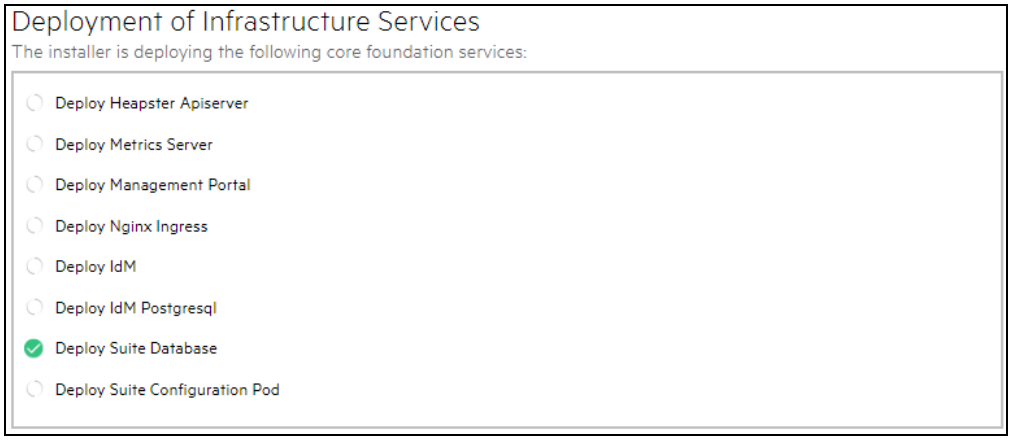
  - After installation has finished, the logs are copied to `${K8S_HOME}/log/scripts/install`
- You can watch the status of deployment pods with the command:

```
kubectl get pods --namespace core -o wide | grep -i cdf-add-node
```

The Initial Master Node is not reflected by its own `cdf-add-node` pod.

Infrastructure Services

Infrastructure services are then deployed. The **Deployment of Infrastructure Services** page shows the deployment progress.



Please be patient. Wait for all services to be properly deployed (showing a green check icon). This can take up to 15 minutes to complete. Should any service show a red icon, then this process might have timed out. If this occurs, click the **Retry** icon to retry the deployment for that service.

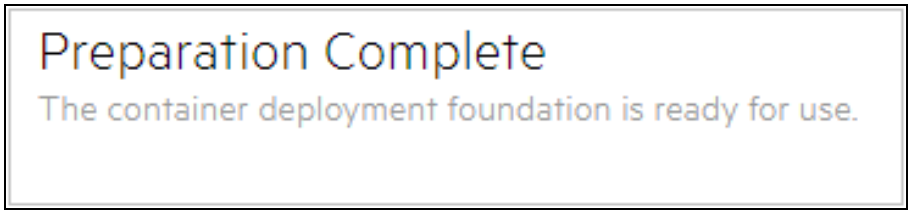
To monitor progress as pods are being deployed, on the Initial Master Node, run the command:

```
watch 'kubectl get pods --all-namespaces'
```

If you try to access the CDF Management Portal Web UI (port 3000) too quickly after this part of the install has finished, you might receive a 'Bad Gateway' error. Allow more time for the web UI to start (3 to 5 minutes) before retrying your login attempt.

After all services show a green check mark, click **Next**.

Once all nodes have been configured, and all services have been started on all nodes, the **Preparation Complete** page will be shown. You are ready to configure product-specific configuration properties.



Click **Next** to configure the products and components of the deployment. You can now deploy ArcSight products.

Configuring the Deployed Capabilities

Refer to System Hardware Sizing and Tuning Guidelines in the *ArcSight Platform 22.1 Technical Requirements* for your workload. It might specify additional settings beyond what is described below.

You are now ready to deploy and then configure your deployed capabilities. The *Pre-Deployment Configuration* page displays to configure the products and capabilities chosen at the start of the installation process. This section explains the process of configuring deployed capabilities on a supported platform for both on-premises and cloud deployments.

Describing Parameters

The following parameters are mentioned in one or more of the example install config files.

Transformation Hub

For the TH yaml, see the following:

Name	Description
routing-processor1-replicas	Specifies the number of Routing Stream Processor Instances to start for the Group 1 Stream Processor. Routing Stream Processors convert incoming CEF events based on predefined rules associated with a unique source Topics. Group numbers are dynamically assigned by Transformation Hub. Tune the number of instances based on throughput requirements.
th-init-noOfTopicPartitions	For newly created Kafka Topics, specifies the number of partitions assigned to each Topic. Default is 6. A Partition is the unit of parallelism in Kafka, enabling write operations on both the Producer and Broker to be performed concurrently. This is a key tuning property.
transform-processor-replicas	Specifies the number of CEF-to-Avro Stream Processor Instances to start. CEF-to-Avro Stream Processors convert incoming CEF events from th-cef topic to Avro format and route these events to th-arcsight-avro topic.
th-init-kafkaRetentionBytesForVertica	Specifies the size, in gigabytes, of the retention log for th-arcsight-avro and mf-event-avro-enriched Topics (Avro primary Topics). Default is 60 GB. This is a key tuning property. This log is associated with Avro processing. It is uncompressed and might require up to 7 times more space than compressed data. When this log size is exceeded, event data will be dropped.
th-init-kafkaRetentionBytes	Specifies the size, in gigabytes, of the retention log for each Kafka Topic. Default is 60 GB. This is a key tuning property. When the retention log exceeds the size limit, event data will be dropped.

enrichment-processor1-replicas	Specifies the number of Enrichment Stream Processor Group Instances to start. Enrichment Stream Processors transform incoming events based on the set of enabled event enrichment features, and route these events to one or more destination Topics. Enrichment examples include adding Global Event IDs and event integrity checking. Tune the number of instances based on throughput requirements.
th-enrichment-processor-group1-source-topic	Specifies the source Topic to be used by the Enrichment Stream Processor Group.
th-enrichment-processor-integrity-enabled	Indicates whether to generate a verification event that accompanies a batch of events for checking the integrity of parsed fields in each event. Recon uses this verification event to check event integrity. Also, specify a value for 'Verification event batch size'.
th-enrichment-processor-integrity-batch-size	Specifies the number of events to be associated with a verification event. A lower value indicates fewer associated events need to be included in the batch for integrity checks; however, it will also result in higher resource consumption by generating more verification events.

Recon

For the Recon yaml, see the following:

Name	Description
interset-elasticsearch-data-instances	Specifies the number of Elasticsearch data processing instances.
interset-elasticsearch-index-replicas-count	Specifies the number of replicas for each Elasticsearch index. 0 means no copy, only use that value when having no HA/Production requirement.
interset-logstash-event-buffering	Specifies the internal queuing model to use for event buffering. Specify memory for legacy in-memory based queuing; persisted for disk-based queuing.
interset-logstash-instances	Specifies the number of Logstash instances.
recon-enable	Indicates whether to explore events in Recon in addition to Intelligence.

Reviewing Settings That Must Be Set During Deployment

This section describes configuration settings that must be set during deployment. Additional settings can be modified after deployment by browsing to the CDF Management Portal.

For more information on a setting, hover over the setting to display the setting tooltip, then set the values accordingly.

The following products require configuration settings to be set during deployment.

- "ArcSight Database" below
- "Transformation Hub" below
- "Fusion" on the next page
- "Intelligence" on page 118

ArcSight Database

If you deployed the ArcSight database and you configure SmartConnectors to use the CEF format when you send events to the Transformation Hub, in the **Transformation Hub** tab, ensure the # of CEF-to-Avro Stream Processor instances to start is set to at least 1 or what is specified in *ArcSight Platform Technical Requirements* for your workload.

On the **Fusion** tab, ensure that you set these configuration settings for your environment:

- Enable Database
- Use SSL for Database Connections

> The SSL configuration requires components to be in a running state before proceeding with the database secured configuration. To apply secure communication to the database, proceed with Completing the Database Setup for AWS S3.

- Database Host

> The host list of the database node's IP, that is node1-IP, node2-IP,..., upto nodeN-IP.

- Database Application Admin User Name
- Database Application Admin User Password
- Search User Name
- Search User Password
- Database Certificate(s)
- Database Host Name(s)

Transformation Hub

If you deployed Transformation Hub, in the **Transformation Hub** tab, ensure the following are set to the number of Kafka worker nodes in your deployment or what is specified in *ArcSight Platform Technical Requirements* for your workload.

- # of Kafka broker nodes in the Kafka cluster (th-kafka-count)
- # of ZooKeeper nodes in the ZooKeeper cluster (th-zookeeper-count)
- # of replicas assigned to each Kafka Topic (th-init-topicReplicationFactor) (This setting must be set to 1 for a single worker deployment, and 2 for a 3-node environment.)

On the **Transformation Hub** tab, configure the following security settings based on how you planned to secure communications as described in the Securing Communication Among Micro Focus Components section.

FIPS and Client-Authentication are available during installation only.

- Allow plain text (non-TLS) connections to Kafka (th-kafka-allow-plaintext)
- Enable FIPS 140-2 Mode (th-init-fips)
- Connection to Kafka uses TLS Client Authentication (th-init-client-auth)
- # of message replicas for the __consumer_offsets Topic (th-init-kafkaOffsetsTopicReplicationFactor)
- Schema Registry nodes in the cluster (th-schema-registry-count)
- # of replicas assigned to each Kafka Topic (th-init-topicReplicationFactor)

If you are deploying ESM, configure your Enrichment Stream Processor Group source Topic according to the scope for which you want to leverage ESM's event enrichment capability. For more information, refer to Enrichment Stream Processors.

Fusion

If you deployed Fusion, on the **Fusion** tab:

- Modify the **Client ID** (sso-client-id) and **Client Secret** (sso-client-secret) to a unique value for your environment.
- If you are deploying Transformation Hub and configured **# of Enrichment Stream Processor Group instances to start** (enrichment-processor1-replicas) with a value greater than zero (default is 2), which means Enrichment Stream Processor will be enabled, the Fusion ArcMC Generator ID Manager must be enabled with a sufficient range of IDs because the Enrichment Stream Processor automatically requests generator IDs from the Fusion ArcMC in the same cluster as needed for its processing. To enable the Fusion ArcMC Generator ID Manager, configure **Enable Generator ID Manager** (arcmc-generator-id-enable) to **True** (default is True) and set the the values of **Generator ID Range Start** (arcmc-generator-id-start) and **Generator ID Range End** (arcmc-generator-id-end) to provide a range of at least 100 between them. A range of 100 should be sufficient for common scenarios with a comfortable buffer, but you could also make the range larger if you have configured a large number of Enrichment Stream Processor instances or other components that use Generator IDs from this Fusion ArcMC instance.
- **Maximum Search Results:** This value specifies number of results that a search can return. Maximum limit is 10 million events.

It is important to choose a range that does not overlap with the Generator ID Manager range configured in any other ArcMC instances in your organization, otherwise different events with duplicate Globally Unique Event IDs could be created.

Intelligence

If you deployed Intelligence, on the Intelligence tab, ensure you set these configuration settings for your environment:

- Number of Database Nodes (interset-vertica-number-of-nodes)

Ensure that you change any password to a unique value for your environment.

- HDFS NameNode (interset-hdfs-namenode)
- Elasticsearch Index Replicas Count (interset-elasticsearch-index-replicas-count)
- H2 Password (interset-h2-password)
- Elasticsearch Password (interset-elasticsearch-password)
- Analytics KeyStore Password (interset-analytics-keystore-password)
- Investigator KeyStore Password (interset-api-keystore-password)
- SearchManager KeyStore Password (searchmanager-api-keystore-password)
- Logstash KeyStore Password (interset-logstash-keystore-password)
- H2 KeyStore Password (interset-h2-keystore-password)

Consider the following:

- If the topic name specified for the Avro Event Topic field is not the default topic, then use Transformation Hub's Avro routing rules using ArcMC 2.96 or later to filter Avro events from the default topic. Create a routing rule with the source topic as mf-event-avro-enriched and destination topic as the topic name you have provided in the Avro Event Topic field. For more information, see Creating a Route.

- For **Analytics Configuration-Spark**, set the values based on the data load. For information about the values for Spark, see System Hardware Sizing and Tuning Guidelines in the *ArcSight Platform 22.1 Technical Requirements* for your workload.
- For the **Data Identifiers to Identify Machine Users** field, if you need to consider only human users for licensing, ensure that you provide appropriate values to identify and filter out the machine users from licensing. For more information, contact Micro Focus Customer Support.

If you are specifying details under the **Hadoop File System (HDFS) Security** section, consider the following:

- If you are enabling Kerberos Authentication, then, before selecting **kerberos** in **Enable Authentication with HDFS Cluster**, ensure you configure the Kerberos Authentication. For more information, see Enabling and Configuring Kerberos Authentication.
- The Kerberos details that you provide in **Kerberos Domain Controller Server**, **Kerberos Domain Controller Admin Server**, **Kerberos Domain Controller Domain**, and **Default Kerberos Domain Controller Realm** will be considered only if you select **kerberos** in **Enable Authentication with HDFS Cluster**. They are not valid if you select **simple**.
- If you are enabling Kerberos Authentication, then you must enable **Enable Secure Data Transfer with HDFS Cluster**.
  If you disable **Enable Secure Data Transfer with HDFS Cluster**, the database and HDFS will use the same communication standard as Intelligence 6.2.

Tuning Your Deployment for Recon or Intelligence

This section describes tuning your deployment for Recon or Intelligence (only).

Updating Event Topic Partition Number

Refer to the ArcSight Platform 22.1 Technical Requirements, section entitled System Hardware Sizing and Tuning Guidelines to determine an appropriate event topic partition number for your workload.

To update the topic partition number from the master node1, run the following commands:

1. Find NAMESPACE ($NS), for th-kafka-0:

   ```
   NS=`kubectl get pods --all-namespaces|grep kafka-0|awk '{print $1}'`
   ```

2. Update the Enrichment Stream Processor source topic (`th-arcsight-avro` or `mf-event-avro-esmfiltered`) and `mf-event-avro-enriched` topic partition numbers to the same for both topics:

   ```
   kubectl exec -n $NS th-kafka-0 -- /usr/bin/kafka-topics --bootstrap-server th-kafka-svc:9092 --alter --topic ENRICHMENT_SP_SOURCE_TOPIC --partitions $number
   ```

3. Use the Kafka Manager to verify that the partition number of the th-cef topic, enrichment stream processor source topic (`th-arcsight-avro` or `mf-event-avro-esmfiltered`) and `mf-event-avro-enriched` topics have been updated to `$number`, where `$number` is the number used to calculate partition size.

4. Update the `th-arcsight-avro` and `mf-event-avro-enriched` topic partition numbers to the same for both topics:

```
kubectl exec -n $NS th-kafka-0 -- /usr/bin/kafka-topics --bootstrap-
server th-kafka-svc:9092 --alter --topic th-arcsight-avro --partitions
$number
```

```
kubectl exec -n $NS th-kafka-0 -- /usr/bin/kafka-topics --bootstrap-
server th-kafka-svc:9092 --alter --topic mf-event-avro-enriched --
partitions $number
```

```
kubectl exec -n $NS th-kafka-0 -- /usr/bin/kafka-topics --bootstrap-
server th-kafka-svc:9092 --alter --topic th-cef --partitions $number
```

   where `$number` is the number used to calculate partition size.

   Standard Kafka topics settings only permit increasing the number of partitions, not decreasing them, so please consider that when performing step 2.

5. Use the Kafka manager to verify that the partition number of th-cef topic, th-arcsight-avro and mf-event-avro-enriched topics have been updated to `$number`.

In case of Recon, partition number == DB nodes # * 12; therefore for a 3 node db cluster, the partition number is 36.

Updating the CDF Hard Eviction Policy

You need to update the Kubernetes hard eviction policy from 15% (default) to 100 GB to maximize disk usage.

To update the CDF Hard Eviction Policy, perform the following steps on each worker node, after deployment has been successfully completed. Please verify the operation is successfully executed on one work node first, then proceed on the next worker node.

The eviction-hard can be defined as either a percentage or a specific amount. The percentage or the specific amount will be determined by the volume storage.

To update the policy:

1. Run the following commands:

```
cp
/usr/lib/systemd/system/kubelet.service
/usr/lib/systemd/system/kubelet.service.orig
```

```
vim /usr/lib/systemd/system/kubelet.service
```

2. In the file, after `ExecStart=/usr/bin/kubelet \`, add the following line:

```
--eviction-
hard=memory.available<100Mi,nodefs.available<100Gi,imagefs.available<2Gi \
```

3. Save your change to the file.
4. To activate the change, run the following command:

```
systemctl daemon-reload and systemctl restart kubelet
```

5. To verify the change, run:

```
systemctl status kubelet
```

No error should be reported.

Labeling On-premises Worker Nodes

Labeling is a means for identifying application processing and qualifying the application as a candidate to run on a specific node. For example, labeling a node with the label `kafka:yes` indicates that a Kafka instance will run on that node. The labels tell Kubernetes the types of workloads that can run on a specific host system.

Immediately following deployment of your chosen capabilities, many of their associated pods will remain in a *Pending* state until you complete the labeling process. For example, the following Transformation Hub pods will be pending: `th-kafka`, `th-zookeeper`, `th-kafka-manager`, `th-web-service`, and `th-schemaregistry`.

When you finish labeling the nodes, Kubernetes immediately schedules and starts the label-dependent containers on the labeled nodes. The starting of services might take 15 minutes or more to complete.

Label the Worker Nodes

You must first define the labels that you want to use, then assign them to each node. Labels are case-sensitive and must include the `:yes` text. To learn which labels apply to your deployed capabilities, see "Understanding Labels and Pods" on page 578.

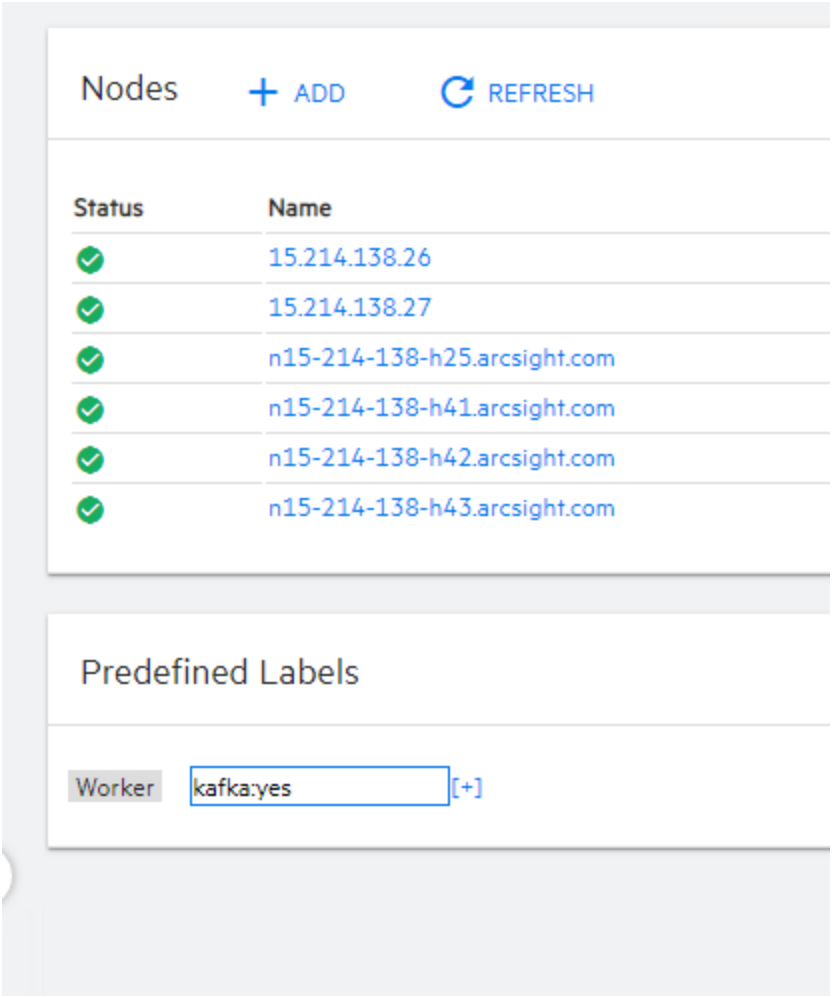
> The `master:yes` and `Worker` labels are already predefined, and already applied to your nodes based on your installation. You will not need to take any action regarding these labels.

1. Log in to CDF Management Portal by clicking the link on the **Deployment status** (Configuration complete) page or browsing to (`https://<ha-address>:5443`), where:

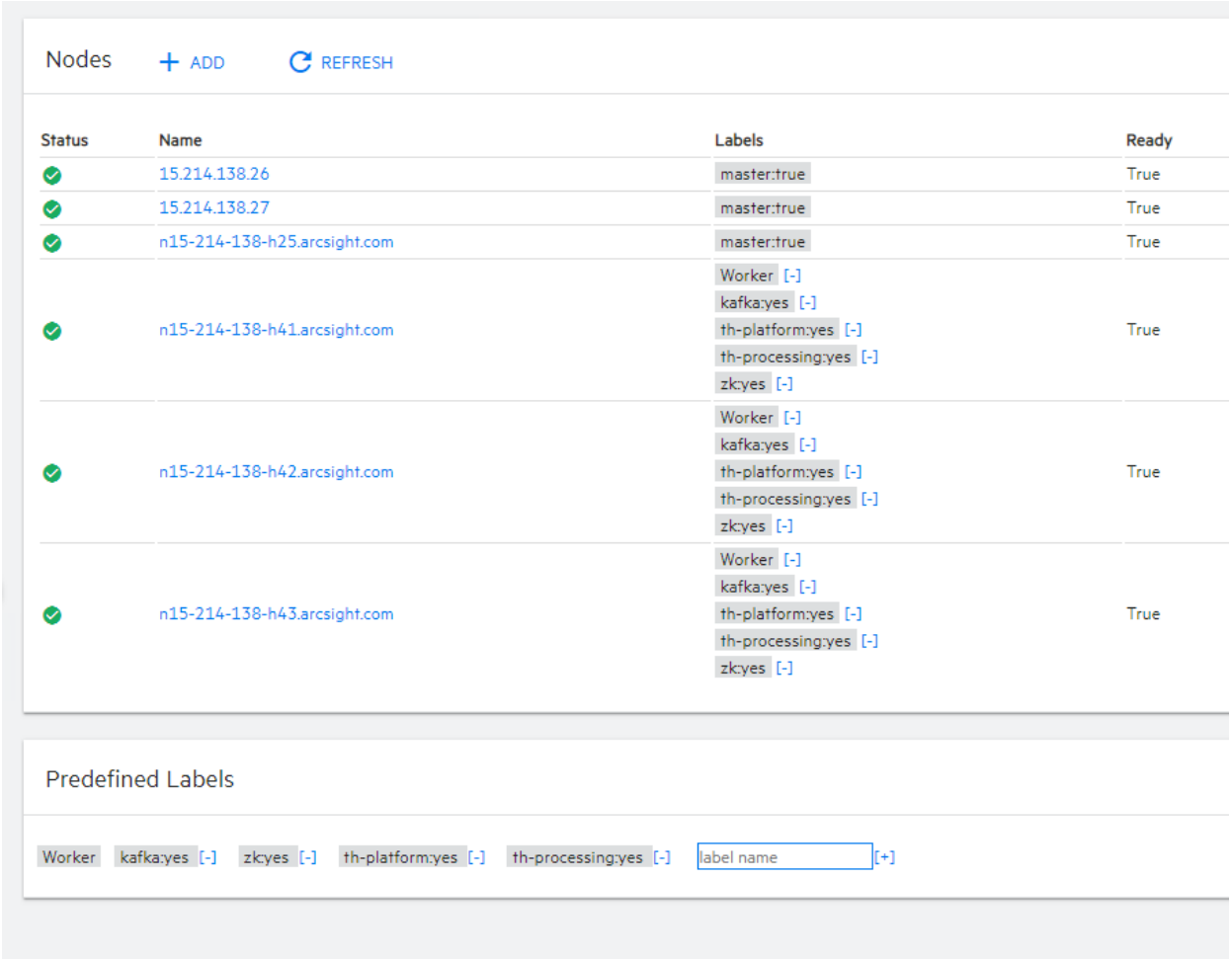
- *Ha-address:* represents the FQDN corresponding to the Virtual IP address provided during installation (`--ha-virtual-ip`) (or, for a single-master installation, the IP address of the master node).
- *User Name:* admin
- *Password:* Password that you created the first time that you logged in to the Management Portal.

2. Go to **CLUSTER** > **Nodes**.
3. In the text box for **Predefined Labels**, specify the label to add, then click the **+** icon.

   For example, for Transformation Hub, create the `zk:yes` label. As you create the labels, the CDF Management Portal adds them to the **Predefined Labels** list to the left of the text box.
4. Repeat **Step 3** for each of the labels that you want to add to the list of predefined labels. Specify the text of the entire label, as shown here, including the `:yes` text.



5. Drag and drop each of the labels that you created to their corresponding worker nodes, based on your workload-sharing configuration. The corresponding components get deployed on the labeled worker nodes.



6. Click **Refresh** to see the labels that you have applied to the nodes.

   After the nodes have been properly labeled, the status of the CDF pods in the **Configuration Complete** page displays as *Running* state.

7. To monitor the pod start up process, continue to "Checking Deployment Status" on page 369.
8. (Conditional) To scale out the cluster, add more worker nodes to it.

Checking Deployment Status

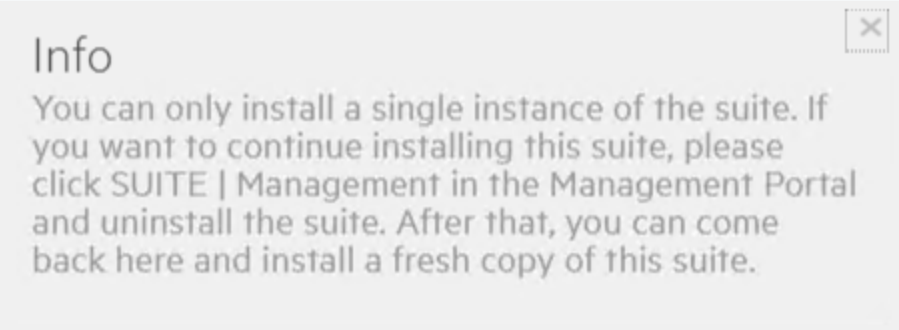
When the **Configuration Complete** page displays, the pod deployment is finished.

- Pods that have not been labeled remain in the *Pending* state until labeled.
- For a pod that is not in the *Running* state, you can find out more details on the pod by running the following command:

```
kubectl describe pod <pod name> -n <namespace>
```

The `Events` section in the output provides detailed information on the pod status.

If you see the following error when you attempt to log in to the CDF Management Portal on port 3000, this typically means that the CDF installation process has completed, port 3000 is no longer required, and has been closed. Instead of port 3000, log in to the Management Portal on port 5443.



Checking Cluster Status

To verify the success of the deployment, check the cluster status and make sure all pods are running.

You might need to wait 10 minutes or more for all pods to be in a *Running* or *Completed* state.

To check cluster status:

1. Connect to the cluster by doing one of the following:
   - For an on-premises installation, log in to the initial master node.
   - For Azure, connect to the jump host.
   - For AWS, connect to the bastion.
2. Run the command:

   ```
   kubectl get pods --all-namespaces
   ```
3. Review the output to determine the status of all pods.

If the Elasticsearch and Logstash pods enter into a CrashLoopBackOff state, refer to Known Issues in the ArcSight Intelligence 6.4.0 Release Notes for the workarounds.

Completing the Database and Kafka Scheduler Setups

This section details the process for completing the database and Kafka Scheduler setups for both on-premises and cloud deployments.

Gathering Certificates for the Kafka Scheduler Setup

The database and deployed capabilities need to establish a trusted connection. To do so, generate the key pair for the Kafka Scheduler.

This step is required even if you use non-SSL communication between the Kafka Scheduler and Transformation Hub, because the schema registry is always SSL-enabled.

1. Run these commands on your database node1 to generate the Kafka Scheduler private key file `kafkascheduler.key.pem` and the certificate signing request file `kafkascheduler.csr.pem`:

   ```
   cd <yourOwnCertPath>/
   ```

   If you installed using the ArcSight Platform Installer, the default location is `/opt/arcsight-db-tools/cert/`

   ```
   openssl req -nodes -newkey rsa:2048 -keyout kafkascheduler.key.pem -out
   kafkascheduler.csr.pem -subj "/C=US/ST=State/L=City/O=Company
   Inc./OU=IT/CN=kafkascheduler"
   ```

2. Copy the certificate signing request `kafkascheduler.csr.pem` to your cluster, bastion host, or jump host.
3. Run the following commands on your cluster or your bastion host to sign the certificate signing request using your cluster RE certificate:

   ```
   export CDF_APISERVER=$(kubectl get pods -n core -o custom-
   columns=":metadata.name"| grep cdf-apiserver)
   ```

   ```
   export PASSPHRASE=$(kubectl get secret vault-passphrase -n core -o json
   2>/dev/null | jq -r '.data.passphrase')
   ```

```
export ENCRYPTED_ROOT_TOKEN=$(kubectl get secret vault-credential -n core
-o json 2>/dev/null | jq -r '.data."root.token"')
```

```
export VAULT_TOKEN=$(echo ${ENCRYPTED_ROOT_TOKEN} | openssl aes-256-cbc -
md sha256 -a -d -pass pass:"${PASSPHRASE}")
```

```
export COMMON_NAME=kafkascheduler
```

```
export CSR=$(cat ${COMMON_NAME}.csr.pem)
```

```
WRITE_RESPONSE=$(kubectl exec -it -n core ${CDF_APISERVER} -c cdf-
apiserver -- bash -c "VAULT_TOKEN=$VAULT_TOKEN vault write -tls-skip-
verify -format=json RE/sign/coretech csr=\"${CSR}\"") && \
```

```
echo "${WRITE_RESPONSE}" | jq -r ".data | .certificate" > ${COMMON_
NAME}.crt.pem && \
```

```
echo "${WRITE_RESPONSE}" | jq -r ".data | if .ca_chain then .ca_chain[]
else .issuing_ca end" > issue_ca.crt
```

4. Copy the RE signed certificate file `kafkascheduler.crt.pem` to database node1 <*yourOwnCertPath*>.
5. Copy the `issue_ca.crt` to database node1 <*yourOwnCertPath*>.

Enabling the Database to Receive SSL Connections

The following procedures are recommended for data privacy, but they are optional. Perform the first two procedures below on database node1.

- Creating the Database Server Key and Certificate
- Setting up the Database SSL Configuration
- "Configuring Deployed Capabilities to Use SSL for Database Connection" on page 129

Creating the Database Server Key and Certificate

Follow these steps to generate database CAs and certificates:

1. Log in to database node1 as root.
2. Change to your own certificates directory path:

```
cd <yourOwnCertPath>
```

For deployment with `arcsight-platform-instsaller`, the default location is `/opt/arcsight-db-tools/cert/`

3. Run this command to create a certificate authority (CA) for the database:

```
openssl req -newkey rsa:4096 -sha256 -keyform PEM -keyout generated-db-ca.key -x509 -days 3650 -outform PEM -out generated-db-ca.crt -subj "/C=US/ST=State/L=City/O=Company Inc./OU=IT/CN=Database/emailAddress=admin@microfocus.com" -nodes
```

4. Run this command to create the database server key:

```
openssl genrsa -out generated-db-server.key 4096
```

5. Create the database server certificate signing request by running the following command:

```
openssl req -new -key generated-db-server.key -out generated-db-server.csr -subj "/C=US/ST=State/L=City/O=Company Inc./OU=IT/CN=DatabaseServer/emailAddress=admin@microfocus.com" -nodes -sha256
```

6. Sign the Certificate Signing Request with self-signed CA by running the following command:

```
openssl x509 -req -in generated-db-server.csr -CA generated-db-ca.crt -CAkey generated-db-ca.key -CAcreateserial -extensions server -days 3650 -outform PEM -out generated-db-server.crt -sha256
```

Setting up the Database SSL Configuration

These steps will update the SSL configuration in the database.

1. Move the following files to database node1 <*yourOwnCertPath*> as root by running these commands:

This step is only required for a new database installation. You can skip this step if this is an upgrade and the files are already there.

```
cd <yourOwnCertPath>/
ls <yourOwnCertPath>/
```

- The output should have the following files:
  - `generated-db-ca.crt`
  - `generated-db-server.crt`
  - `generated-db-server.key`

- generated-db-ca.key
- generated-db-ca.srl
- generated-db-server.csr
- issue_ca.crt
- kafkascheduler.crt.pem
- kafkascheduler.key.pem

2. For chained CAs, run the commands to split the CAs into individual files:

```
cat issue_ca.crt | awk 'BEGIN {c=0;} /BEGIN CERT/{c++} { print > "issue_
ca_part." c ".crt"}'
```

```
chown -R dbadmin:dbadmin <yourOwnCertPath>
```

3. Run the following commands on database node1 to update the database SSL configuration:

```
cd /opt/arcsight-db-tools
```

```
./db_ssl_setup --disable-ssl
```

If the attempt fails, drop the certificate manually by running the three commands below:

```
sudo su - dbadmin
```

```
vsql -U <dbadminuser> -w <dbadminpassword> -c "ALTER TLS CONFIGURATION
server CERTIFICATE NULL;"
```

```
vsql -U <dbadminuser> -w <dbadminpassword> -c "DROP CERTIFICATE IF EXISTS
server CASCADE;"
```

4. Enable database SSL for a single issue CA or chained issue CAs:
   - For a single issue CA, run this command:

```
./db_ssl_setup --enable-ssl --vertica-cert-path
<yourOwnCertPath>/generated-db-server.crt --vertica-key-path
<yourOwnCertPath>/generated-db-server.key --vertica-ca-path
<yourOwnCertPath>/generated-db-ca.crt --client-ca-path
<yourOwnCertPath>/issue_ca.crt
```

-or-

- For chained issue CAs, run this command, specifying each CA certificate in the chain one by one, separated by a comma in the `client-ca-path` parameter:

```
./db_ssl_setup --enable-ssl --vertica-cert-path
<yourOwnCertPath>/generated-db-server.crt --vertica-key-path
<yourOwnCertPath>/generated-db-server.key --vertica-ca-path
<yourOwnCertPath>/generated-db-ca.crt --client-ca-path
<yourOwnCertPath>/issue_ca_part.1.crt, <yourOwnCertPath>/issue_ca_
part.2.crt[,...]
```

Configuring Deployed Capabilities to Use SSL for Database Connection

1. Log in to the CDF Management Portal.
2. Navigate to **Fusion** > **Database Configuration** > **Database Certificate(s)**.
3. Enable the **Use SSL for Database Connection** option.
4. Copy the complete contents of the file `generated-db-ca.crt`, created from the steps earlier, into the Database Certificate(s) text area.
5. Click **Save** to activate the configuration changes.

Enabling the Database to Ingest Events from Transformation Hub

The database uses an event consumer, the **Kafka scheduler**, to ingest events from Transformation Hub's Kafka component. Follow these steps when configuring the Kafka Scheduler for a new installation of the ArcSight Database:

Before you perform these steps, ensure that you have enabled SSL for the database. For information, see Enabling the Database to Receive SSL Connections.

1. Log in to the database node1 as root.
2. Change to the database tools directory:

```
cd /opt/arcsight-db-tools/
```

3. Run the following command on database node1 to configure the schema registry server setting:

```
./schema_registry_setup <FQDN of ArcSight Platform Virtual IP for HA or
single master node / Cloud: <DNS name for your cluster> >
<yourOwnCertPath>/issue_ca.crt <yourOwnCertPath>/kafkascheduler.crt.pem
<yourOwnCertPath>/kafkascheduler.key.pem
```

You must provide the absolute path to the certificate.

4. Configure the SSL setup:

   On database node1, configure the SSL setting for the Kafka Scheduler by using one of the following methods, plain text or SSL:

   Plain Text (non-SSL)

   This method requires that you first enable **Allow plain text (non-TLS)** connections to Kafka. For more information, see Configuring the Deployed Capabilities.

   Run this command to disable SSL for the Kafka scheduler:

   ```
   ./sched_ssl_setup --disable-ssl
   ```

   SSL

   This method uses the crt and key files gathered or generated in earlier steps. The `issue_ca.crt` file should contain all chained CAs. For the Kafka scheduler to use SSL, run the following command:

   ```
   ./sched_ssl_setup --enable-ssl --sched-cert-path
   <yourOwnCertPath>/kafkascheduler.crt.pem --sched-key-path
   <yourOwnCertPath>/kafkascheduler.key.pem --vertica-ca-key
   <yourOwnCertPath>/generated-db-ca.key --vertica-ca-path
   <yourOwnCertPath>/generated-db-ca.crt --kafka-ca-path
   <yourOwnCertPath>/issue_ca.crt
   ```

5. Run this command on database node1 to create the Kafka Scheduler:

   - If the Kafka Scheduler was configured to use plain-text in the previous step, use port 9092:

     ```
     ./kafka_scheduler create <th_kafka_nodename1>:9092
     ```

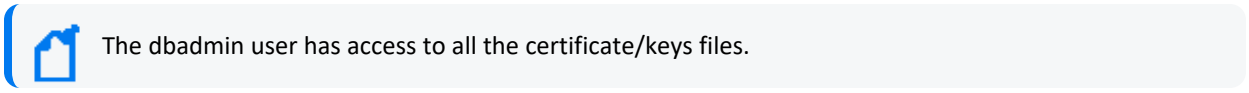
   - If SSL was enabled for the Kafka Scheduler in the previous step, use port 9093:

     ```
     ./kafka_scheduler create <th_kafka_nodename1>:9093
     ```

6. Start the Kafka Scheduler and checker on database node1:

   ```
   ./kafka_scheduler start
   ./kafka_scheduler messages
   ./kafka_scheduler events
   ```

7. Continue to the post-deployment section.



Applying the CDF 2021.05 log4j Hotfix

Some deployments of, and upgrades to, CDF 2021.05/arcsight-platform-installer-22.1.x.x.zip require application of a hotfix to remediate the log4j vulnerability, which was discovered in 2021. The hotfix will upgrade IDM for CDF to use log4j 2.17.1, to prevent exploitation of the log4j vulnerabilities (CVE-2021-44228, CVE-2021-45046, CVE-2021-45105, CVE-2021-44832). The hotfix should be applied after an upgrade.

The hotfix applies to the following types of installations and upgrades:

- Any on-premises manual installation of 22.1.x or any on-premises manual upgrade to 22.1.1. (A manual installation or upgrade is one that does not use the ArcSight Installer.)

The CDF 2021.05 hotfix will be automatically applied during any on-premises installation or upgrade using the ArcSight Installer and this procedure can be skipped.

- Any CDF fresh installation or upgrade on AWS.
- Any CDF fresh installation or upgrade on Azure.

The log4j remediation hotfix does **NOT** apply to on-premises installations or upgrades performed automatically using the ArcSight Installer, as the hotfix is applied automatically by the installer. For such installations or upgrades these procedures can be skipped.

Hotfix File

The hotfix file is named arcsight-idm-hf-22.1.0-2.zip.

1. Get the file:
   - For a manual on-premises deployment/upgrade, the hotfix is bundled in `/<download_folder>/arcsight-platform-installer-22.1.x.x/installers/hotfix`.
   - For an AWS or Azure deployment upgrade, obtain the file from the *on-premises* installation file directory at `/<download_folder>/arcsight-platform-installer-22.1.x.x/installers/hotfix`.
2. Copy the file:
   - For manual on-premises, copy the file to your master node.
   - For AWS, copy the file to your bastion.
   - For Azure, copy the file to your jump host.
3. Unzip the hotfix file. In the unzipped folder, run the following command with the '-e' argument (values: onprem, azure, aws) to apply the latest image.

```
# ./hotfix.sh -e <YOUR_ENV>
```

Verifying the Hotfix

1. Check the pod status by running the following command. It should be 'Running' as 2/2.

```
# kubectl get pods -A | grep idm
```

2. Check the image version by running the following command.

```
# kubectl get deployment/itom-idm -n core -o yaml | grep itom-idm:1.32.1-343
```

It should display as below:

```
image: <image-registry-url>/<org-name>/itom-idm:1.32.1-343
```

Rolling Back to the Previous Version

To roll back itom-idm to the previous version, run the following roll back commands :

```
# kubectl delete -f /tmp/cdf-itom-idm.yaml
```

```
# kubectl create -f /tmp/cdf-itom-idm.yaml
```

Chapter 4: Creating a Cloud Deployment

This section discusses the process of preparing for and creating a cloud deployment.

Setting Up Your Amazon Web Services Deployment Architecture

This section explains how to set up your deployment architecture for ArcSight capabilities that runs on the Amazon Web Services (AWS) cloud platform.

Checklist: Planning to Deploy ArcSight Capabilities on AWS

The complete process of deploying on AWS includes the following broad steps. Each of these steps is explained in the following sections. Most steps can be performed using either the AWS web UI or through the AWS CLI tool, and each method is explained (where possible).

Use the following checklist to deploy and using AWS. Perform the tasks in the listed order.

	Task	See
❑	1. Review the technical prerequisites and ensure that they are met before beginning the installation	"Reviewing Deployment Prerequisites" on the next page
❑	2. Create and configure the AWS Virtual Private Cloud, including security groups and IAM roles	"Create the Virtual Private Cloud" on page 140
❑	3. Create two security groups, one for the bastion host and one for intra-VPC connectivity	"Creating Security Groups" on page 150
❑	4. Prepare the EKS control plane	"Creating the IAM Role for EKS" on page 155
❑	5. Prepare the bastion host, which you will use for access to the AWS deployment environment	"Creating and Configuring the Bastion" on page 158
❑	6. Download the required installation files and associated tools	"Downloading Installation Tools and Packages" on page 172
❑	7. Install the Database in AWS	Formatting Instance Store Type Devices
❑	8. Prepare the EFS instance used for the AWS deployment environment	"Creating the Elastic File System" on page 186

❑	9. Set up your EKS cluster	"Configuring the Elastic Kubernetes Service" on page 194
❑	10. Create and label the worker nodes, where application processing takes place	"Creating and Configuring Worker Nodes" on page 202
❑	11. Transfer the product images to the ECR	"Uploading Product Images to the ECR" on page 219
❑	12. Prepare the Route 53 DNS routing	"Directing the Route 53 Record Set to the ALB" on page 252
❑	13. Create and import the user-supplied certificate into Amazon Certificate Manager	Creating the User-Supplied Certificate
❑	14. Install CDF rudiments so that you can perform a complete installation after load balancer configuration	"Bootstrapping CDF" on page 227
❑	15. Prepare the application load balancer	"Creating the Application Load Balancer" on page 244
❑	16. Install the remaining CDF components and deploy the ArcSight Suite products	"Installing CDF" on page 255
❑	17. Configure access to the CDF management portal and access to re-configuration	"Performing Post Installation Network Configuration" on page 257
❑	18. Deploy ArcSight Suite capabilities using the CDF Management Portal	Deploying ArcSight Products
❑	19. Apply the hotfix to remediate the log4j vulnerability	"Applying the CDF 2021.05 log4j Hotfix" on page 570
❑	20. Get the latest security fixes and enhancements	Upgrading to 22.1.2

Reviewing Deployment Prerequisites

In order to deploy ArcSight capabilities on AWS, the user requires an active AWS subscription, as well a properly configured IAM user account.

Installation of ArcSight Suite is performed under the local IAM user. If you do not have a local IAM user, ask your AWS administrator to create a user for you and assign the required IAM policies as described below.

- "Reviewing the Minimal Permissions for IAM User" on the next page
- "Configuring the Local Host" on page 138
- Reviewing Storage Considerations
- "Using the AWS Deployment Worksheet" on page 139

Reviewing the Minimal Permissions for IAM User

Access to various AWS resources is controlled by permissions assigned to the IAM user. For easier management, you can create a policy holding the minimal set of permissions required to complete tasks in this guide. The policy must contain the following permissions.

JSON file for required permissions

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "route53:*",
            "iam:AddRoleToInstanceProfile",
            "iam:AttachRolePolicy",
            "iam:CreateAccessKey",
            "iam:CreateInstanceProfile",
            "iam:CreatePolicy",
            "iam:CreateRole",
            "iam:DeleteAccessKey",
            "iam:DeleteInstanceProfile",
            "iam:DeletePolicy",
            "iam:DeleteRole",
            "iam:DeleteRolePolicy",
            "iam:DetachRolePolicy",
            "iam:GenerateServiceLastAccessedDetails",
            "iam:GetAccessKeyLastUsed",
            "iam:GetAccountSummary",
            "iam:GetLoginProfile",
            "iam:GetPolicyVersion",
            "iam:GetRole",
            "iam:GetRolePolicy",
            "iam:GetServiceLastAccessedDetails",
            "iam:GetServiceLastAccessedDetailsWithEntities",
            "iam:GetUser",
            "iam:ListAccessKeys",
            "iam:ListAccountAliases",
            "iam:ListAttachedGroupPolicies",
            "iam:ListAttachedRolePolicies",
            "iam:ListAttachedUserPolicies",
            "iam:ListEntitiesForPolicy",
            "iam:ListGroupPolicies",
            "iam:ListGroups",
```

```
            "iam:ListGroupsForUser",
            "iam:ListInstanceProfiles",
            "iam:ListInstanceProfilesForRole",
            "iam:ListMFADevices",
            "iam:ListOpenIDConnectProviders",
            "iam:ListPolicies",
            "iam:ListPoliciesGrantingServiceAccess",
            "iam:ListPolicyVersions",
            "iam:ListRolePolicies",
            "iam:ListRoleTags",
            "iam:ListRoles",
            "iam:ListSAMLProviders",
            "iam:ListSSHPublicKeys",
            "iam:ListServerCertificates",
            "iam:ListServiceSpecificCredentials",
            "iam:ListSigningCertificates",
            "iam:ListUserPolicies",
            "iam:ListUserTags",
            "iam:ListUsers",
            "iam:ListVirtualMFADevices",
            "iam:PassRole",
            "iam:PutRolePolicy",
            "iam:RemoveRoleFromInstanceProfile",
            "iam:TagRole",
            "iam:TagUser",
            "iam:UntagRole",
            "iam:UntagUser",
            "iam:UpdateAccessKey",
            "iam:UpdateLoginProfile"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "acm:*",
            "autoscaling:*",
            "cloudformation:*",
            "ec2:*",
            "ecr:*",
            "eks:*",
            "elasticfilesystem:*",
            "elasticloadbalancing:*",
            "s3:CreateBucket",
            "s3:DeleteObject",
            "s3:GetObject",
```

```
            "s3:PutObject",
            "sns:ListSubscriptions",
            "sns:ListTopics",
            "ssm:DescribeActivations",
            "ssm:DescribeAssociation",
            "ssm:DescribeAssociationExecutionTargets",
            "ssm:DescribeAssociationExecutions",
            "ssm:DescribeAutomationExecutions",
            "ssm:DescribeAutomationStepExecutions",
            "ssm:DescribeAvailablePatches",
            "ssm:DescribeDocument",
            "ssm:DescribeDocumentParameters",
            "ssm:DescribeDocumentPermission",
            "ssm:DescribeEffectiveInstanceAssociations",
            "ssm:DescribeEffectivePatchesForPatchBaseline",
            "ssm:DescribeInstanceAssociationsStatus",
            "ssm:DescribeInstanceInformation",
            "ssm:DescribeInstancePatchStates",
            "ssm:DescribeInstancePatchStatesForPatchGroup",
            "ssm:DescribeInstancePatches",
            "ssm:DescribeInstanceProperties",
            "ssm:DescribeInventoryDeletions",
            "ssm:DescribeMaintenanceWindowExecutionTaskInvocations",
            "ssm:DescribeMaintenanceWindowExecutionTasks",
            "ssm:DescribeMaintenanceWindowExecutions",
            "ssm:DescribeMaintenanceWindowSchedule",
            "ssm:DescribeMaintenanceWindowTargets",
            "ssm:DescribeMaintenanceWindowTasks",
            "ssm:DescribeMaintenanceWindows",
            "ssm:DescribeMaintenanceWindowsForTarget",
            "ssm:DescribeOpsItems",
            "ssm:DescribeParameters",
            "ssm:DescribePatchBaselines",
            "ssm:DescribePatchGroupState",
            "ssm:DescribePatchGroups",
            "ssm:DescribePatchProperties",
            "ssm:DescribeSessions",
            "ssm:GetAutomationExecution",
            "ssm:GetCommandInvocation",
            "ssm:GetConnectionStatus",
            "ssm:GetDefaultPatchBaseline",
            "ssm:GetDeployablePatchSnapshotForInstance",
            "ssm:GetDocument",
            "ssm:GetInventory",
            "ssm:GetInventorySchema",
            "ssm:GetMaintenanceWindow",
```

```
            "ssm:GetMaintenanceWindowExecution",
            "ssm:GetMaintenanceWindowExecutionTask",
            "ssm:GetMaintenanceWindowExecutionTaskInvocation",
            "ssm:GetMaintenanceWindowTask",
            "ssm:GetManifest",
            "ssm:GetOpsItem",
            "ssm:GetOpsSummary",
            "ssm:GetParameter",
            "ssm:GetParameterHistory",
            "ssm:GetParameters",
            "ssm:GetParametersByPath",
            "ssm:GetPatchBaseline",
            "ssm:GetPatchBaselineForPatchGroup",
            "ssm:GetServiceSetting",
            "ssm:ListAssociationVersions",
            "ssm:ListAssociations",
            "ssm:ListCommandInvocations",
            "ssm:ListCommands",
            "ssm:ListComplianceItems",
            "ssm:ListComplianceSummaries",
            "ssm:ListDocumentVersions",
            "ssm:ListDocuments",
            "ssm:ListInstanceAssociations",
            "ssm:ListInventoryEntries",
            "ssm:ListResourceComplianceSummaries",
            "ssm:ListResourceDataSync",
            "ssm:ListTagsForResource",
            "ssm:PutConfigurePackageResult"
         ],
         "Resource":"*"
      }
   ]
}
```

Configuring the Local Host

To configure a local host:

You can configure and use any local host which has Internet access for the initial steps in setting up your deployment environment. Later, you will create a bastion instance, and use the bastion to perform the installation, as well as to access the cluster after installation.

Requirements: The `AWS CLI` (v2) and `jq` tools must be installed on the local host. `AWS CLI` is a unified tool to manage AWS services. If it is not already installed, then install and configure the `AWS CLI` (version 2) tool for your platform. All references to CLI in this guide refer to the `AWS CLI` version 2 interface.

- Amazon provides the instructions for installing AWS CLI.
- After installation, configure the `AWS CLI` to properly authenticate and connect to AWS as described in Configuring AWS CLI.

Most procedures for configuring AWS are supplied in both the AWS CLI and web UI versions.

`jq` is a lightweight and flexible open-source command-line JSON processor.

- You can download the `jq` binaries from the `jq` homepage.

Reviewing Storage Considerations

Your cloud administrator needs to setup AWS S3 storage before you install the database. When setting up the AWS instance, do not create a folder for the database in the S3 bucket. You can create a folder on the database node during database installation, but you cannot configure a folder pre-created in AWS during installation.

Your cloud administrator will also need to set up default encryption for the S3 bucket before installing the database. For information about enabling S3 bucket encryption, see AWS documentation, *Enabling Amazon S3 default bucket encryption*.

In the ArcSight Platform, you can organizedata into **storage groups**, which allows you to partition the incoming events data and provide different retention periods, based on the query filter. To preserve space in the database and improve data retrieval from storage groups, you can configure the database to remove events older than a certain number of months. Your product license affects the maximum value that you set for the data retention policy.

Using the AWS Deployment Worksheet

To use the worksheet:

The process of setting up an AWS deployment environment will require configuration of many AWS resources. As a result, you will need convenient access to important details of these resources, such as resource names, IP addresses, settings for AWS entities, and so on, which you will determine during the setup process.

For ease of reference, it's strongly recommended that you print out and use the AWS worksheet to record the details of your configuration. The procedures given here assume you are using the worksheet for reference and will note when particular details should be recorded.

Next Step: Creating the Virtual Private Cloud (VPC)

Create the Virtual Private Cloud

Creating the VPC

To create the VPC, in the AWS CLI, run the following command:

```
# aws ec2 create-vpc \
--cidr-block <CIDR allocated for new VPC> \
| jq -r '.Vpc.VpcId'
```

The command will return the new VPC's VPC ID. Record the VPC ID and VPC CIDR to the AWS worksheet.

For example below is an input and output:

```
# aws ec2 create-vpc \
--cidr-block 10.0.0.0/16 \
| jq -r '.Vpc.VpcId'
```

```
vpc-0143197ca9bd9c117
```

To (optionally) verify assigned tags:

Run the command:

```
# aws ec2 describe-tags \
--filters "Name=resource-id,Values=<VPC ID>"
```

For example:

```
# aws ec2 describe-tags \
--filters "Name=resource-id,Values=vpc-0143197ca9bd9c117"
```

```
{
                        "Tags":[
                        {
                        "Key":"Name",
                        "ResourceId":"vpc-0143197ca9bd9c117",
                        "ResourceType":"vpc",
                        "Value":"srgdemo-vpc"
                        },
```

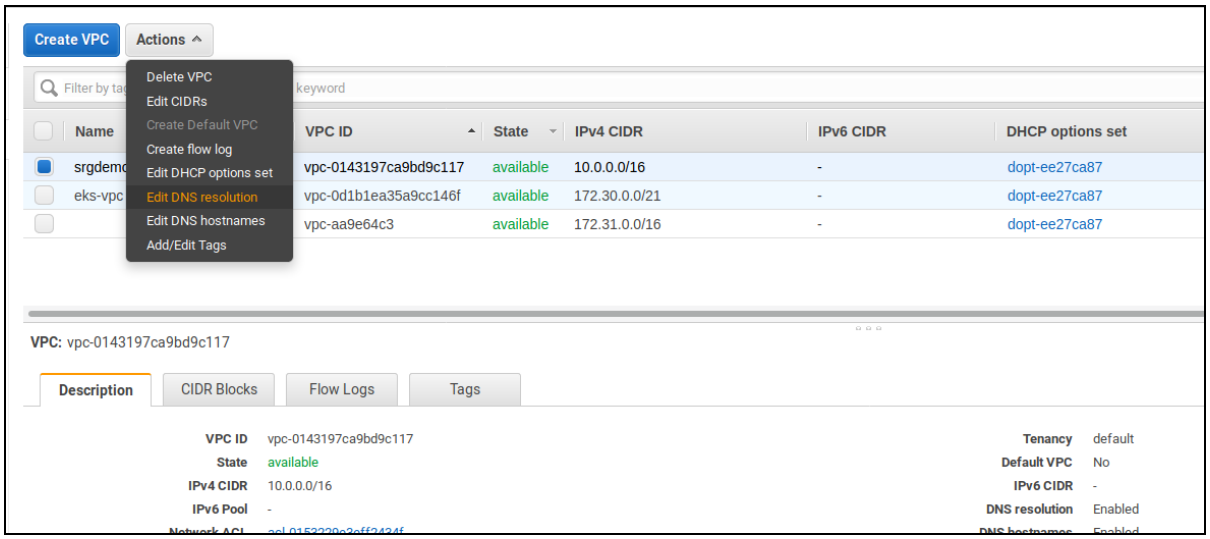
```
                    {
                    "Key":"kubernetes.io/cluster/srgdemo-cluster",
                    "ResourceId":"vpc-0143197ca9bd9c117",
                    "ResourceType":"vpc",
                    "Value":"shared"
                    }
                    ]
            }
```

Enabling DNS and Hostname Resolution

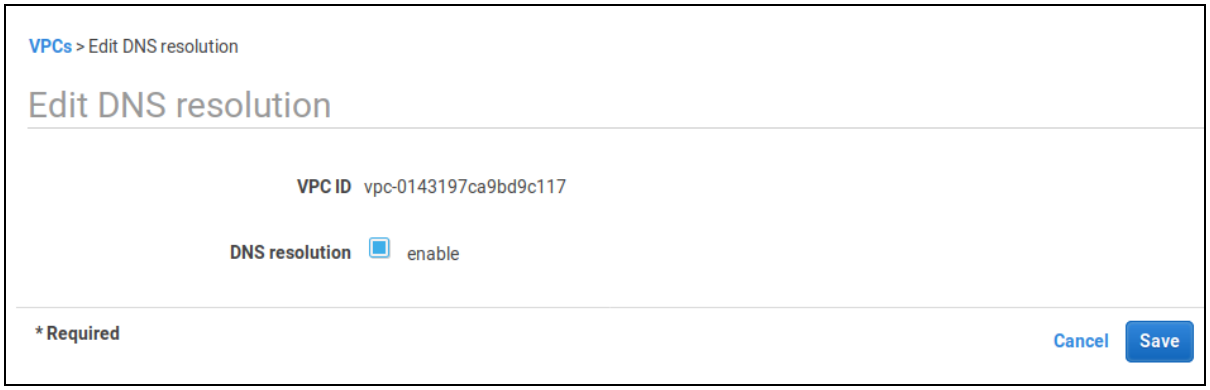
DNS support and hostname resolution should be enabled to make IP addresses more easily human-readable.

To enable DNS using the web UI:

1. Using the Find Services search tool, locate and browse to the VPC dashboard.
2. On the left navigation panel, under **Virtual Private Cloud,** click **Your VPCs.**
3. Select the check box corresponding to your VPC. Then, under **Actions**, select **Edit DNS resolution**.



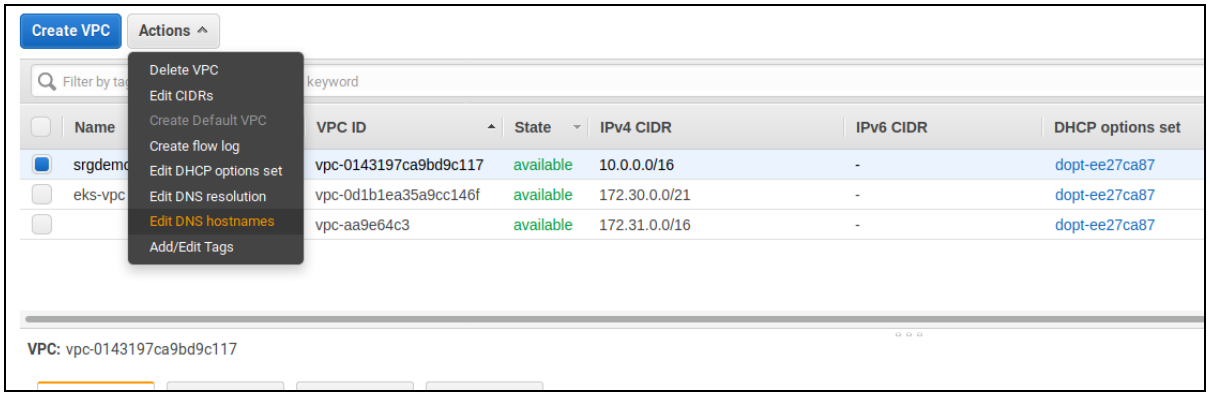
4. On the **Edit DNS Resolution** page, for **DNS resolution**, select the **enable** check box.



5. Click **Save**, then click **Close**.

To enable hostname resolution using the web UI:

1. Using the **Find Services** search tool, locate and browse to the VPC dashboard.
2. On the left navigation panel, under **Virtual Private Cloud**, click **Your VPCs.**
3. Select the check box corresponding to your VPC. Then, under **Actions**, select **Edit DNS hostnames**.



4. On the **Edit DNS Hostnames** page, for **DNS hostnames**, select the **enable** check box.
5. Click Save, then click **Close**.

To enable DNS using the AWS CLI:

Execute the following commands *in order*, using the VPC ID of your created VPC:

```
# aws ec2 modify-vpc-attribute \
--vpc-id <VPC Id> \
--enable-dns-support

# aws ec2 modify-vpc-attribute\
 --vpc-id <VPC Id> \
--enable-dns-hostnames
```

These commands have no output.

For example:

```
# aws ec2 modify-vpc-attribute \
--vpc-id vpc-0143197ca9bd9c117 \
--enable-dns-support
# aws ec2 modify-vpc-attribute \
--vpc-id vpc-0143197ca9bd9c117 \
--enable-dns-hostnames
```

Next Step: Create the External IP Address

Create the External IP Address

The external IP address (EIP) is required for the NAT Gateway, used by the worker nodes, to access the Elastic Container Registry (ECR). In this step, you will create the EIP and then tag it.

- "Creating the EIP" below
- "Tagging the EIP" below

Creating the EIP

1. Run the following command:

   ```
   aws ec2 allocate-address --domain vpc
   ```

2. Record the `AllocationId` value in the AWS worksheet.

   For example:

   ```
   {
   "PublicIp": "18.194.179.100",
   "AllocationId": "eipalloc-004be822658206abe",
   "PublicIpv4Pool": "amazon",
   "NetworkBorderGroup": "eu-central-1",
   "Domain": "vpc"
   }
   ```

Tagging the EIP

Run the following command:

```
aws ec2 create-tags \
--resources <Allocation Id> \
--tags Key=Name,Value=<eip-name>
```

Where:

- `<Allocation Id>`: Use the Allocation ID of the EIP.
- `<eip-name>`: Assign an EIP name for easier identification.

For example:

```
aws ec2 create-tags \
--resources eipalloc-004be822658206abe \
--tags Key=Name,Value=srgdemo-eip
```

Creating and Tagging the Subnets

In this section, you will create one private and one public subnet for each of the three availability zones, for a total of six subnets. Each availability zone requires one private and one public subnet to support high availability.

- "Creating a Subnet" below
- "Tagging the Subnets" on the next page

Each subnet must meet the following criteria:

- Each subnet come from the VPC IP range.
- Subnets must not overlap one another.

All six subnets will be created in the same way. They will be distinguished based on the:

- Route table
- Internet gateway
- NAT gateway attachments

Before proceeding, make sure you have completed your AWS worksheet with your subnet names, CIDRs, and availability zones.

Creating a Subnet

1. Retrieve the availability zone names by running the following command:
   ```
   # aws ec2 describe-availability-zones \
   | jq -r '.AvailabilityZones[ ].ZoneName'
   ```

Example output:

```
eu-central-1a
```

```
eu-central-1b
```

```
eu-central-1c
```

2. Create the first subnet by running the following command, which will output the subnet ID:

```
# aws ec2 create-subnet \
--availability-zone <availability zone> \
--cidr-block <CIDR> --vpc-id <VpcId> | jq -r '.Subnet.SubnetId'
```

For example:

```
# aws ec2 create-subnet \
--availability-zone eu-central-1a \
--cidr-block 10.0.1.0/24 \
--vpc-id vpc-0143197ca9bd9c117 | jq -r '.Subnet.SubnetId'
```

```
subnet-06a8caab19022c544
```

3. Repeat Step 2 for all rows from the subnet planning table in the AWS worksheet.

You should now tag the new subnets to differentiate between public and private subnets, as well as tag the private subnets for load balancing.

Tagging the Subnets

1. Tag each **public** subnet by running the following command for each public subnet:

```
# aws ec2 create-tags \
--resources <public subnet id> \
--tags Key=Name,Value=<subnet name>
```

2. Tag each **private** subnet by running this command for each private subnet:

```
# aws ec2 create-tags \
--resources <private subnet id> \
--tags Key=Name,Value=<subnet name> Key=kubernetes.io/role/internal-elb,Value=1
```

Where:

- `<public/private subnet id>`: The value from column **Subnet ID** in your planning table on the AWS worksheet.
- `<public/private subnet name>`: The value from column **Subnet name** in your planning table on the AWS worksheet.

For example:

```
# aws ec2 create-tags \
--resources subnet-06a8caab19022c544 \
--tags Key=Name,Value=srgdemo-public-subnet-1
```

```
# aws ec2 create-tags \
--resources subnet-0fb2ebb5882c061f0 \
--tags Key=Name,Value=srgdemo-private-subnet-1
Key=kubernetes.io/role/internal-elb,Value=1
```

Creating the Internet Gateway

The Internet gateway is the prerequisite for the NAT gateway, which will be created later.

To create the internet gateway and attach it to the VPC:

1. Run the following command:
   `aws ec2 create-internet-gateway`

Example output:

```
{
    "InternetGateway":{
        "Tags":[

        ],
        "InternetGatewayId":"igw-0ddcfa7511fe10b43",
        "Attachments":[

        ]
    }
}
```

2. Record the value of `InternetGatewayId` in your AWS worksheet.
3. Optionally, you might tag the internet gateway by running the following command:
   `aws ec2 create-tags --resources <InternetGatewayId> --tags Key=Name,Value=<internet gateway name>`
4. Attach the internet gateway to your previously-created VPC by running the following command (command has no output):
   `aws ec2 attach-internet-gateway --internet-gateway-id <InternetGatewayId> --vpc-id <VPC Id>`

For example:

```
aws ec2 attach-internet-gateway \
--internet-gateway-id igw-0ddcfa7511fe10b43 \
--vpc-id vpc-0143197ca9bd9c117
```

Next Step: Creating the NAT Gateway

Creating the NAT Gateway

The NAT gateway is required for worker nodes to connect to the Elastic Container Registry (ECR), which is used for downloading CDF and product images.

To create the NAT gateway:

1. Run the following command:
   ```
   aws ec2 create-nat-gateway \
   --allocation-id <EIP allocation Id> \
   --subnet-id <public subnet id>
   ```

Example input and output:

```
aws ec2 create-nat-gateway \
--allocation-id eipalloc-004be822658206abe \
--subnet-id subnet-0c0ca63f2f793907d
```

```
{
   "NatGateway":{
      "CreateTime":"2021.05-20T20:53:01.000Z",
      "NatGatewayAddresses":[
         {
            "AllocationId":"eipalloc-004be822658206abe"
         }
      ],
      "NatGatewayId":"nat-013416dad7b7656ea",
      "State":"pending",
      "SubnetId":"subnet-0c0ca63f2f793907d",
      "VpcId":"vpc-0143197ca9bd9c117"
   }
}
```

2. Record the `NatGatewayId` value in your AWS worksheet.

Next Step: Creating the Route Tables

Creating the Route Tables

Route tables define the routing paths between resources in private and public subnets and the Internet.

To create the private and public route tables:

1. Run the following command to create a route table and retrieve its ID:
   ```
   aws ec2 create-route-table \
   ```

```
--vpc-id <VpcId> \
| jq -r '.RouteTable.RouteTableId'
```

2. Run the command in Step 1 a second time, to create another route table and retrieve its ID.

Example input and output:

```
aws ec2 create-route-table \
--vpc-id vpc-0143197ca9bd9c117 \
 jq -r '.RouteTable.RouteTableId'
```

```
rtb-0deda70daa09ca3bfw
```

3. Tag the first route table as private. Run the command:
```
aws ec2 create-tags --resources <route table ID> \
--tags Key=Name,Value=<route table name indicating private>
```

Example:
```
aws ec2 create-tags \
--resources rtb-0deda70daa09ca3bf \
--tags Key=Name,Value=srgdemo-private-route-table
```

4. Repeat Step 3 for the second route table, with the `--tags` value indicating `public` instead of `private`.

Associating the Route Tables to Subnets

The route tables will now need to be associated to the subnets you have created.

To associate the route tables to your public subnets:

1. Select one of your **public** subnets.
2. Associate the **public** route table to the selected **public** subnet by running the command:
```
aws ec2 associate-route-table \
--route-table-id <public route table ID> \
--subnet-id <public subnet ID>
```
3. Repeat the command in Step 2 for each of the other two **public** subnets.

To associate the route tables to your private subnets:

1. Select one of your **private** subnets.
2. Associate the **private** route table to the selected **private** subnet by running the command:
```
aws ec2 associate-route-table \
--route-table-id <private route table ID> \
--subnet-id <private subnet ID>
```
3. Repeat the command in Step 2 for each of the other two **private** subnets.

Example input and output:

```
aws ec2 associate-route-table \
--route-table-id rtb-0deda70daa09ca3bf \
--subnet-id subnet-0fb2ebb5882c061f0
```

```
{
   "AssociationId":"rtbassoc-781d0d1a",
   "AssociationState":{
      "State":"associated"
   }
}
```

Adding the NAT Gateway Route Path to the Private Route Table

To add the NAT gateway route path to the private route table:

1. Run the following command:
   ```
   aws ec2 create-route \
   --route-table-id <private route table Id> \
   --destination-cidr-block "0.0.0.0/0" \
   --nat-gateway-id <NAT GW Id>
   ```
2. The command will return the creation status. A status of `true` indicates that the request succeeded.

Example input and output:

```
aws ec2 create-route \
--route-table-id rtb-0deda70daa09ca3bf \
--destination-cidr-block "0.0.0.0/0" \
--nat-gateway-id nat-013416dad7b7656ea
```

```
{
   "Return":true
}
```

Adding the Internet Gateway Route Path to the Public Routing Table

To add the Internet Gateway route path to the public routing table:

1. Run the following command:
   ```
   aws ec2 create-route \
   --route-table-id <public route table Id> \
   --destination-cidr-block "0.0.0.0/0" \
   --gateway-id <Internet Gateway Id>
   ```

Example:

```
aws ec2 create-route \
--route-table-id rtb-0fa9f294a3743c9aa \
--destination-cidr-block "0.0.0.0/0" \
--gateway-id igw-0ddcfa7511fe10b43
```

Creating Security Groups

A *security group* is an AWS resource that acts as a firewall for the subnets. Every AWS resource must be assigned a security group so they will be network accessible. If a resource is assigned to multiple security groups, then all rules from all groups will be applied to the resource.

You will need to create two security groups, one for the bastion host and one for intra-VPC connectivity. The procedures are explained in the following sections.

Next Step: Creating the Security Group for the Bastion Host

Creating the Security Group for the Bastion Host

In order to connect to the bastion from the Internet and perform the configuration and installation tasks, you must open the connection on the default SSH port (port 22) from any address.

Optionally, you can limit the access to the bastion by specifying your own public, static IP address while adding your own inbound rule as described below. Replace `0.0.0.0/0` with your own public IP address. If you choose to specify your own IP address, talk to your AWS infrastructure administrator before proceeding.

To create the security group for the bastion host:

1. Run the following command:
   ```
   # aws ec2 create-security-group \
   --description "Enables SSH Access to Bastion Hosts" \
   --group-name <group name> --vpc-id <VpcId>
   ```

Where:

`<group name>`: A descriptive security group name of your choice; in our examples we will use `srgdemo-bastion-sg`.

`<VpcID>`:The VPC ID of the VPC you created earlier.

Example:

```
aws ec2 create-security-group \
--description "Enables SSH Access to Bastion Hosts" \
```

```
--group-name srgdemo-bastion-sg \
--vpc-id vpc-0143197ca9bd9c117
```

```
{
    "GroupId":"sg-00b5fcc4294d234f6"
}
```

2. Record the bastion security group ID in your AWS worksheet.

Adding the Inbound Rule

You will connect to the bastion using SSH on default port 22, so the newly-created security group needs to be opened to inbound connection on port 22 and the TCP protocol.

To add the inbound rule:

1. Open the security group to inbound connections on the default SSH port 22 (TCP) by running the following command:
   ```
   aws ec2 authorize-security-group-ingress \
   --group-id <bastion security group ID> \
   --ip-permissions IpProtocol=tcp,FromPort=22,ToPort=22,IpRanges='------ip-
   permissions IpProtocol=tcp,FromPort=80,ToPort=80,IpRanges='
   [{CidrIp=10.0.0.0/0,Description="HTTP"}]'
   ```

Example:

```
# aws ec2 authorize-security-group-ingress \
--group-id sg-00b5fcc4294d234f6 \
--ip-permissions
```

2. Remove the default wide-open outbound rule by running the following command:
   ```
   aws ec2 revoke-security-group-egress \
   --group-id <security group ID> \
   --protocol all \
   --port -1 \
   --cidr 0.0.0.0/0
   ```

While working from the bastion you will need to connect to various resources on the internet. Protocols and description for ports are shown in the following table:

Port	Protocol	Allowed CIDR	Description
80	TCP	0.0.0.0/0	HTTP
443	TCP	0.0.0.0/0	HTTPS

3. Add HTTP and HTTPS outbound rules by running the following command:
   ```
   aws ec2 authorize-security-group-egress \
   --group-id sg-00b5fcc4294d234f6 \
   ```

```
--ip-permissions IpProtocol=tcp,FromPort=80,ToPort=80,IpRanges='
[{CidrIp=10.0.0.0/0,Description="HTTP"}]'
```

For `<port>`, `<protocol>`, and `<description>`: Use values from the table above.

Example:

```
aws ec2 authorize-security-group-egress \
--group-id sg-00b5fcc4294d234f6 \
--ip-permissions IpProtocol=tcp,FromPort=22,ToPort=22,IpRanges='
[{CidrIp=0.0.0.0/0,Description="SSH access; unlimited."}]'
```

Next Step: Creating the Security Group for Intra-VPC Communication

Creating the Security Group for Intra-VPC Communication

The intra-VPC security group (SG) is dedicated to resources inside the VPC, and will allow unlimited communication between them. It will also allow outbound connection to the HTTP and HTTPS worldwide.

To create the security group for intra-VPC communication, use the same steps for creating the bastion's security group. However, for the new group, change the description of the security group to an appropriate value, such as `<cluster name> intra VPC SG.`

- Note the group's name and ID to the AWS worksheet.
- Then, remove the default wide-open outbound rule. Repeat the process you performed for the bastion security group, of course using the newly-created intra-VPC security group ID.

Add inbound rule from itself

For communication between intra-VPC resources, we will add a rule enabling all communication coming from this security group. For example:

```
aws ec2 authorize-security-group-ingress \
--group-id <security group ID> \
--protocol all \
--port -1 \
--source-group <security group ID>
```

Where `<security group ID>` is the ID of the newly-created intra-VPC security group.

Add HTTP and HTTPS outbound rules

For retrieving external resources (such as for CDF) and product images from the ECR, OS updates, and similar files, resources inside the VPC need to be able to connect using HTTP/HTTPS on the internet. Repeat the process you performed for the bastion security group.

To add outbound rule to itself, run the following command:

```
aws ec2 authorize-security-group-egress \
--group-id <security group Id> \
--protocol all \
--port -1 \
--cidr <VPC CIDR>
```

- `<security group ID>`: Use the ID of the newly-created intra-VPC security group.
- `<VPC CIDR>`: Use the same CIDR you used for creating the VPC.

Next Step: IAM Roles

IAM Roles

An *IAM role* is an IAM (AWS Identity and Access Management) entity that defines a set of permissions for making AWS service requests and manipulating various resources. They are needed for those capabilities that require the ArcSight Database when deploying via AWS in the cloud. An IAM role is required for all database nodes participating in the cluster to allow connectivity to S3 bucket communal storage. For more information, see Understanding Methods for Connecting to AWS S3 Buckets.

Roles are shareable. Instead of creating new roles, you might use existing roles your organization has previously created. IAM is not region dependent, roles can be reused in all regions your organization uses.

You will create two roles: one for EKS (Elastic Kubernetes Service) and one for worker nodes, and assign them specific policies to define permissions.

Roles, policy names, and corresponding policy ARNs are shown in the following table:

Role	Policy Name	Policy ARN
EKS	AmazonEKSClusterPolicy	arn:aws:iam::aws:policy/AmazonEKSClusterPolicy
EKS	AmazonEKSServicePolicy	arn:aws:iam::aws:policy/AmazonEKSServicePolicy
Worker Nodes	AmazonEKSWorkerNodePolicy	arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy
Worker Nodes	AmazonEC2ContainerRegistryReadOnly	arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly
Worker Nodes	AmazonEKS_CNI_Policy	arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy

EKS Policies

EKS requires the following policies to be granted:

- **AmazonEKSClusterPolicy**

  This policy provides Kubernetes the permissions it requires to manage resources on your behalf. Kubernetes requires EC2: CreateTags permissions to place identifying information on EC2 resources including but not limited to Instances, Security Groups, and Elastic Network Interfaces.

  `ARN: arn:aws:iam::aws:policy/AmazonEKSClusterPolicy`

- **AmazonEKSServicePolicy**

  This policy allows Amazon Elastic Container Service for Kubernetes to create and manage the necessary resources to operate EKS Clusters.

  `ARN: arn:aws:iam::aws:policy/AmazonEKSServicePolicy`

For more information, see AWS documentation by signing into your AWS account: https://aws.amazon.com/

Worker Node Policies

For worker nodes in EKS, the following policies must be granted:

- **AmazonEKSWorkerNodePolicy**

  This policy allows Amazon EKS worker nodes to connect to Amazon EKS Clusters.

  `ARN: arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy`

- **AmazonEC2ContainerRegistryReadOnly**

  This policy provides read-only access to Amazon EC2 Container Registry repositories.

  `ARN: arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly`

- **AmazonEKS_CNI_Policy**

  This policy provides the Amazon VPC CNI Plugin (amazon-vpc-cni-k8s) the permissions it requires to modify the IP address configuration on your EKS worker nodes. This permission set enables the CNI to list, describe, and modify Elastic Network Interfaces on your behalf. For more information about the AWS VPC CNI Plugin, see the link here: https://github.com/aws/amazon-vpc-cni-k8s

  `arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy`

For more information, see AWS documentation by signing into your AWS account: https://aws.amazon.com/

Next Step: Creating the EKS Role

Creating the IAM Role for EKS

To create the EKS role and assign policies to it:

1. Run the following command:
   ```
   aws iam create-role \
   --role-name <role name> \
   --assume-role-policy-document <role policy document>
   ```

Where:

`<role name>`: A name chosen for ease of reference; in our examples, we will use `srgdemo-eks-svc-role`.

`<role policy document>`: The location of a JSON document granting temporary security credentials to perform actions on resources and defining which resources are accessible. There is a ready-to-use document named `EksRolePolicyDocument.json` of the download package `arcsight-platform-cloud-installer-XX.X.X.XXX.zip`, after unzipping, in the in the `objectdefs` folder. This document defines that the cluster can request temporary security credentials to `eks.amazonaws.com` only.

Example output:

```
{
  "Role": {
    "AssumeRolePolicyDocument": "<URL-encoded-JSON>",
    "RoleId": "AKIAIOSFODNN7EXAMPLE",
    "CreateDate": "2013-06-07T20:43:32.821Z",
    "RoleName": "Test-Role",
    "Path": "/",
    "Arn": "arn:aws:iam::123456789012:role/Test-Role"
    }
}
```

2. Record the ARN (Amazon Resource Name) value in your AWS worksheet.

Example input and output:

```
aws iam create-role \
--role-name srgdemo-eks-svc-role \
--assume-role-policy-document file://./jsons/EksRolePolicyDocument.json
```

```
{
    "Role": {
        "Path": "/",
        "RoleName": "srgdemo-eks-svc-role",
        "RoleId": "AROARVXFDN4TOT5P3E3AQ",
```

```
        "Arn": "arn:aws:iam::115370811111:role/srgdemo-eks-svc-role",
        "CreateDate": "2020-05-18T12:10:48Z",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "eks.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        }
    }
}
```

Note the `Arn` value `*arn:aws:iam::115370811111:role/srgdemo-eks-svc-role*`.

3. Attach a policy to the EKS role by running the command:
   ```
   aws iam attach-role-policy \
   --role-name <role name> \
   --policy-arn <policy arn>
   ```

Where:

`<role name>` is the role name you have chosen when creating a new role

`<policy arn>` is the policy ARN from the description above.

4. Repeat Step 3 for the next policy, changing the policy ARN to match.

Example command with example policy name:

```
aws iam attach-role-policy \
--role-name srgdemo-eks-svc-role \
--policy-arn arn:aws:iam::aws:policy/AmazonEKSClusterPolicy
```

Next Step: Creating the Worker Node Role

Creating the Worker Node Role

To create the worker node role and assign policies to it:

1. Run the following command:
   ```
   # aws iam create-role \
   --role-name <role name> \
   --assume-role-policy-document <role policy document>
   ```

Where:

`<role name>`: A name chosen for ease of reference; in our examples, we will use srgdemo-eks-svc-role.

`<role policy document>`:The location of a JSON document granting temporary security credentials to perform actions on resources and defining which resources are accessible. The CDF installation package includes a ready-to-use document named `WorkerNodesRolePolicyDocument.json` in the downloadable package `arcsight-platform-cloud-installer-XX.X.X.XXX.zip`, after unzipping, in the in the `objectdefs` folder This document defines that the cluster can request temporary security credentials to `eks.amazonaws.com` only.

Example output:

```
{
  "Role": {
    "AssumeRolePolicyDocument": "<URL-encoded-JSON>",
    "RoleId": "AKIAIOSFODNN7EXAMPLE",
    "CreateDate": "2013-06-07T20:43:32.821Z",
    "RoleName": "Test-Role",
    "Path": "/",
    "Arn": "arn:aws:iam::123456789012:role/Test-Role"
    }
}
```

2. Record the ARN (Amazon Resource Name) value in your AWS worksheet.

Example input and output:

```
            # aws iam create-role \
--role-name srgdemo-workernodes-svc-role \
--assume-role-policy-document
file://./jsons/WorkerNodesRolePolicyDocument.json
```

```
{
    "Role": {
        "Path": "/",
        "RoleName": "srgdemo-workernodes-svc-role",
        "RoleId": "AROARVXFDN4TICMZYPKJ2",
        "Arn": "arn:aws:iam::115370811111:role/srgdemo-workernodes-svc-role",
        "CreateDate": "2020-05-19T16:20:11Z",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "ec2.amazonaws.com"
```

```
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        }
    }
}
```

3. Attach a policy to the **worker node** role by running the following command:
   ```
   # aws iam attach-role-policy \
   --role-name <role name> \
   --policy-arn <policy arn>
   ```

Where:

`<role name>`:The role name you have chosen when creating a new role.

`<policy arn>`:The policy ARN from the description above.

4. Repeat Step 3 each policy, changing the policy ARN to match.

Example commands:

```
# aws iam attach-role-policy \
--role-name srgdemo-workernodes-svc-role \
--policy-arn arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy
```

```
# aws iam attach-role-policy \
--role-name srgdemo-workernodes-svc-role \
--policy-arn arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly
```

```
# aws iam attach-role-policy \
--role-name srgdemo-workernodes-svc-role \
--policy-arn arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy
```

Next Step: Creating and Configuring the Bastion

Creating and Configuring the Bastion

The *bastion* is the dedicated host which provides secure access to Linux instances located in the private and public subnets of your virtual private cloud (VPC). The bastion host must be located in one of the public subnets you created earlier. In this section, you will configure the bastion for your deployment environment.

Next Step: Creating the SSH Keypair

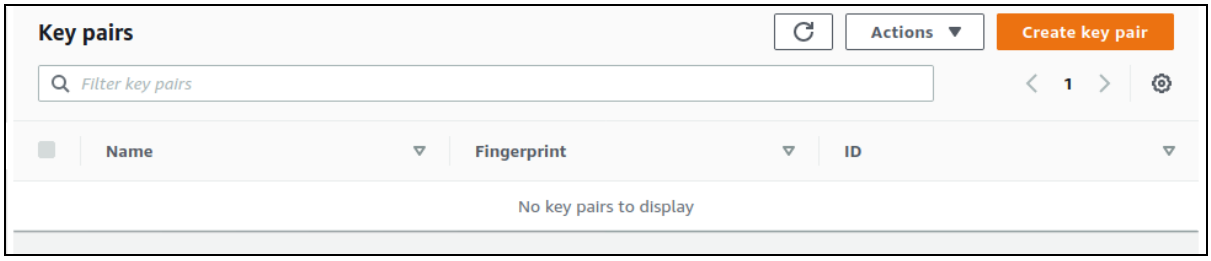
Creating the SSH Keypair

In order to connect to and perform tasks on the bastion, you will use SSH with keypair authentication. In this section, you will create a key pair and store its private value and fingerprint to local files.

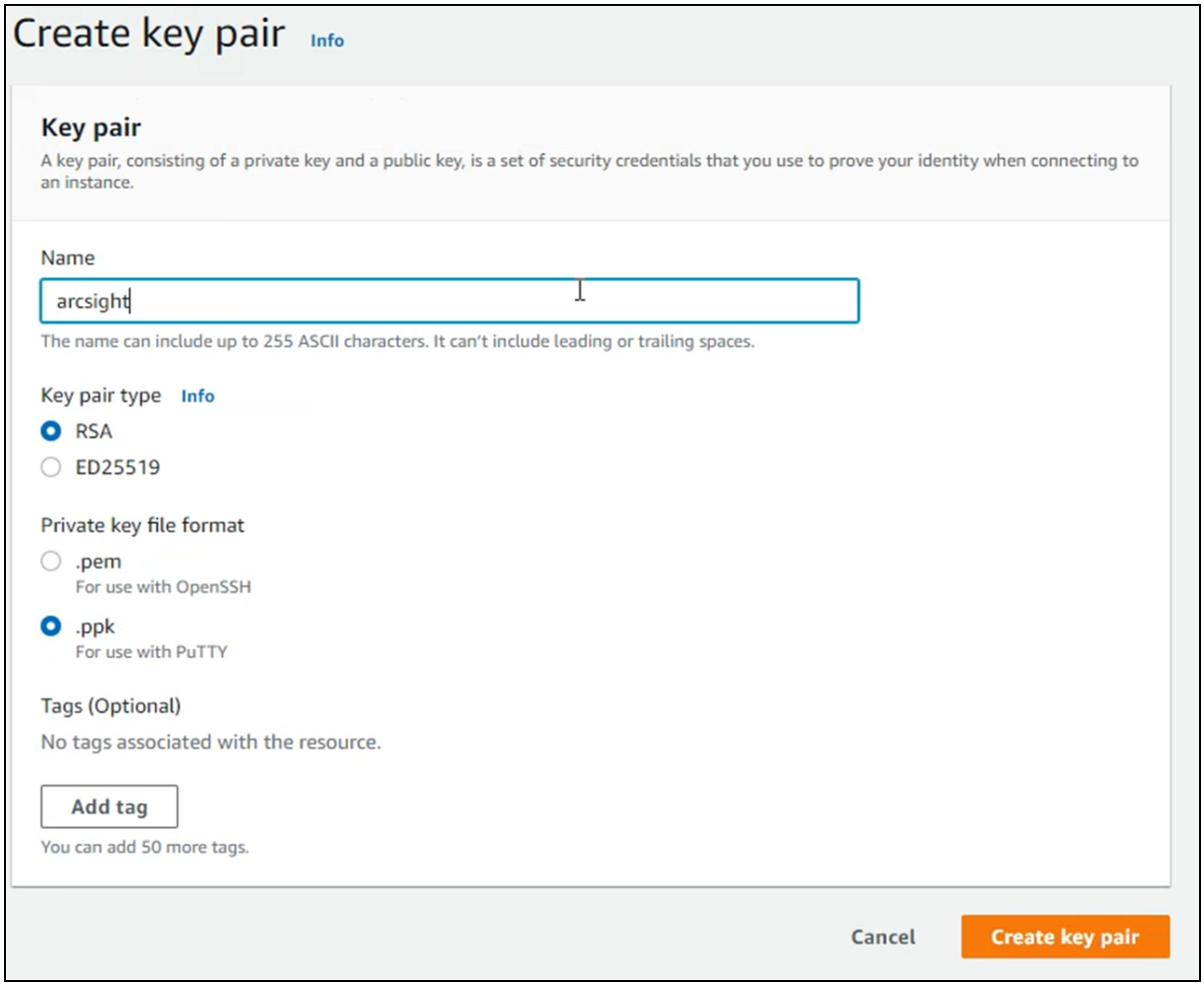
The SSH keypair will be used later for instantiating worker nodes. Optionally, you can create a separate keypair for the bastion and for worker nodes. In that case, follow the steps described here, and give each keypair a distinct name.

To create the keypair using the web UI:

1. Using the Find Services search tool, locate and browse to the EC2 dashboard.
2. In the left navigation pane, under **Network and Security**, select **Key Pairs**
3. On the **Key Pairs** management dialog, click **Create key pair.**



4. On the **Create Key Pair** page, specify values for the following:
   - **Name:** The key pair name will be later used for instantiating bastion as well as worker nodes. You will also use it as a CLI parameter when using an SSH client.
   - **File format:** Choose the format suitable for your client; check the description as shown.



5. Click **Create Key Pair.**
6. On the **Key pair** management dialog, save the private part to a secure location when prompted.

You *must* save the value now, and will not be able to save it later.

5. Optionally, save the key pair fingerprint to the same secure location. The optimal way to store this value is in the file named the same as the private part, exchanging the suffix. For example:
   `srgdemo.fingerprint.`
   You can later compare your locally-stored fingerprint value with the one presented on the table on the web UI.
6. To store the fingerprint value, copy the value in the **Fingerprint** column to a text document on your local machine.
7. Record the keypair name and keypair fingerprint to the AWS worksheet.

To create the SSH key pair using the CLI:

1. Specify the following commands:

```
export KEYPAIR_NAME=<Key pair name>
```

```
export KEYPAIR_CREATION=$(aws ec2 create-key-pair \
--key-name ${KEYPAIR_NAME})
```

```
echo $KEYPAIR_CREATION | jq -r '.KeyMaterial' \
| sed "s/\\\\n/\n/g" > ~/.ssh/${KEYPAIR_NAME}.pem
```

```
echo $KEYPAIR_CREATION \
| jq -r '.KeyFingerprint' > ~/.ssh/${KEYPAIR_NAME}.fingerprint
```

```
chmod 400 ~/.ssh/${KEYPAIR_NAME}.pem
```

Replace the `<KEYPAIR_NAME>` value with your real key pair name. In our examples, we use `srgdemo`.

Example commands:

```
export KEYPAIR_NAME=srgdemo
```

```
export KEYPAIR_CREATION=$(aws ec2 create-key-pair \
--key-name ${KEYPAIR_NAME})
```

```
echo $KEYPAIR_CREATION | jq -r '.KeyMaterial' \
| sed "s/\\\\n/\n/g" > ~/.ssh/${KEYPAIR_NAME}.pem
```

```
echo $KEYPAIR_CREATION \
| jq -r '.KeyFingerprint' > ~/.ssh/${KEYPAIR_NAME}.fingerprint
```

```
chmod 400 ~/.ssh/${KEYPAIR_NAME}.pem
```

Next Steps: Determining the Image ID

Determining the AMI ID

Determine the AMI (Amazon Machine Image) ID used for your bastion instance. You can select an OS image and its corresponding AMI from the AWS Marketplace.

Alternatively, you can find AMIs using the Amazon EC2 console. You can select from the list of AMIs when you use the launch wizard to launch an instance, or you can search through all available AMIs using the Images page. AMI IDs are unique to each AWS Region. Open the

Amazon EC2 console at https://console.aws.amazon.com/ec2/. (**Note**: This link opens an external web site.)

You can also get new image IDs by running OS-based commands:

For CentOS Linux 7, run the following command:

```
aws ec2 describe-images --filters "Name=name,Values=CentOS Linux 7 x86_64 HVM
EBS ENA*" "Name=architecture,Values=x86_64" "Name=virtualization-
type,Values=hvm" "Name=root-device-type,Values=ebs" "Name=owner-
alias,Values=aws-marketplace" | jq '.Images | sort_by(.CreationDate) |
[last]'
```

For Amazon Linux, run the following command:

```
aws ec2 describe-images --owners amazon --filters "Name=name,Values=amzn*gp2"
"Name=virtualization-type,Values=hvm" "Name=root-device-type,Values=ebs" --
query "sort_by(Images, &CreationDate)[-1].ImageId"
```

Record the `ImageId` value in the AWS worksheet.

Next Step: Selecting a Bastion Hardware Instance Type

Selecting a Bastion Hardware Instance Type

The type of host to use for your bastion depends on your deployment plans. Amazon offers a detailed list of EC2 Instance Types with hardware specifications for each. Select and prepare a host that balances and optimizes CPU, memory, storage, and pricing.

For purposes of examples here, we will assume `t2.medium` as your bastion instance type, which will be used only to perform a few configuration tasks and CDF bootstrap install. Your own environment needs might differ.

Once you've selected a bastion host, record its type in your AWS worksheet.

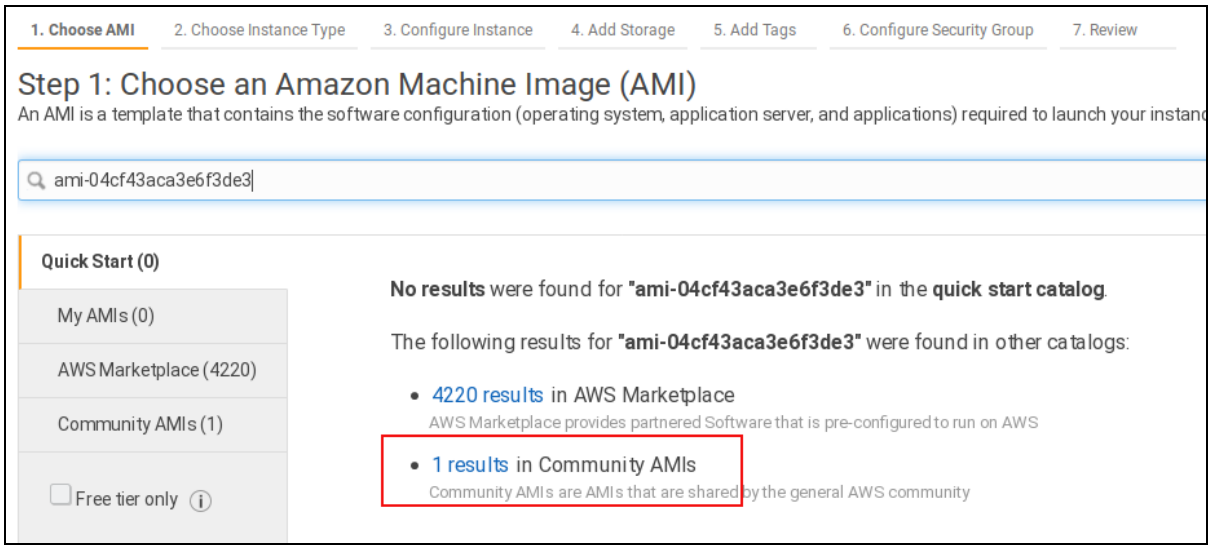
Next Step: Starting the Bastion Instance

Starting the Bastion Instance

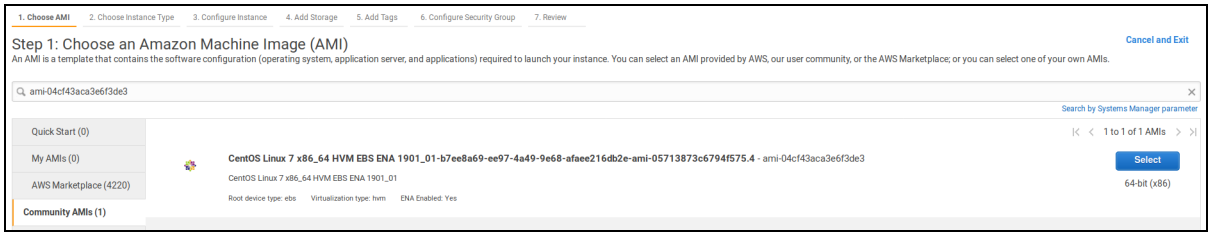
To start the bastion instance through the Web UI:

1. Using the Find Services search tool, locate and browse to the EC2 Dashboard.
2. In the left navigation pane, under **INSTANCES**, click **Instances**.
3. On the Instances management dialog, click **Launch Instance** to start the wizard.

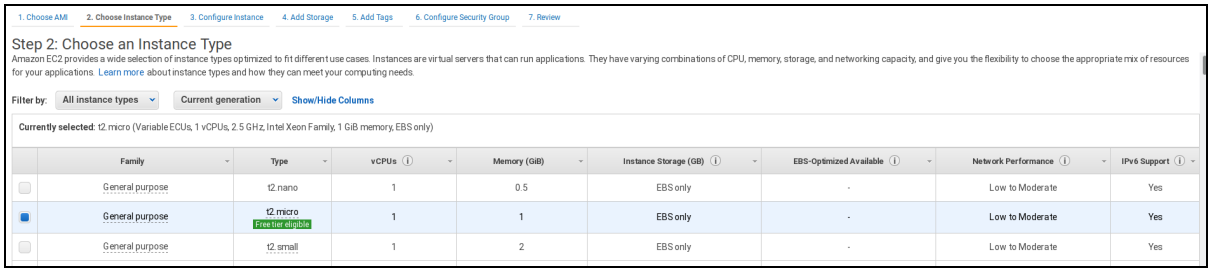
4. On the **Step 1: Choose an AMI** page, in the search box, specify your AMI ID (from your AWS worksheet) and press Enter. The page displays a single result in **Community AMIs.**



5. Click the link for **1 results in Community AMIs.** The selected OS is displayed.
6. Click **Select**.

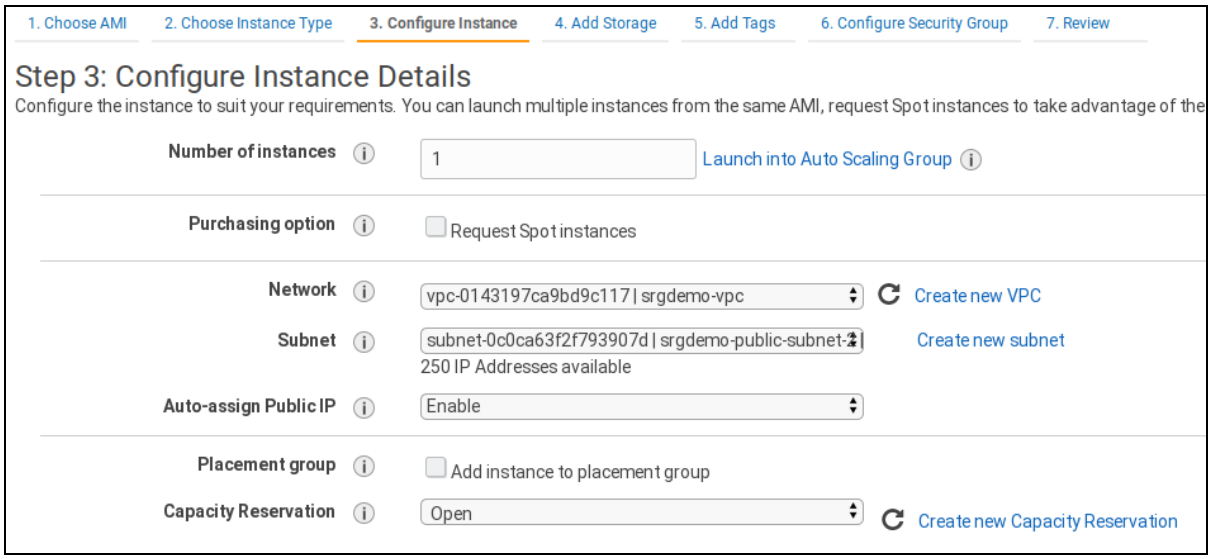


7. On the **Step 2: Choose an Instance Type** page, on the Instance Type list, search for and select your own instance type (use the browser search function if needed). Then, click **Next: Configure Instance Details.**

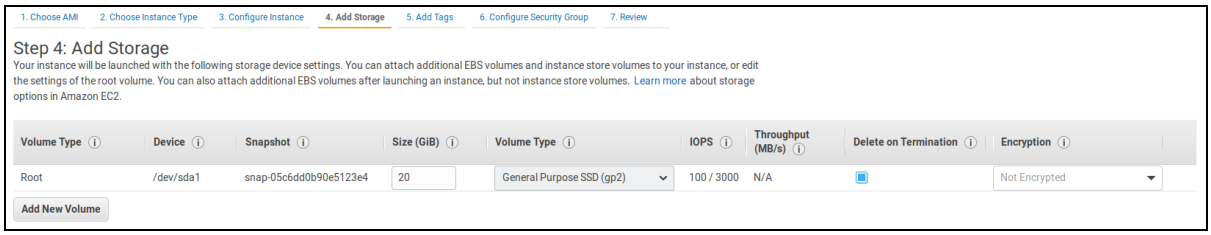


8. On the **Step 3: Configure Instance Details** page, specify these values for the following settings:
   - **Network:** Choose your VPC.
   - **Subnet:** Choose one of your three public subnets.
   - **Auto-assign Public IP:** Enable this value.

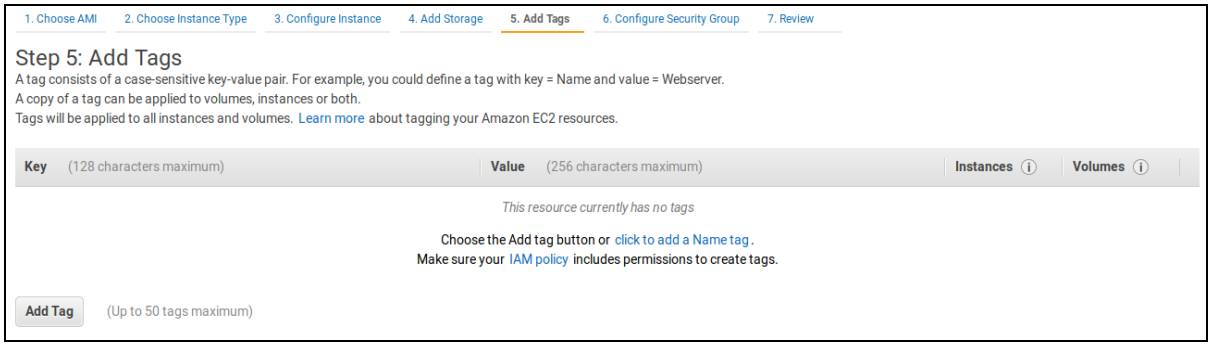
9. Click **Next: Add Storage** .



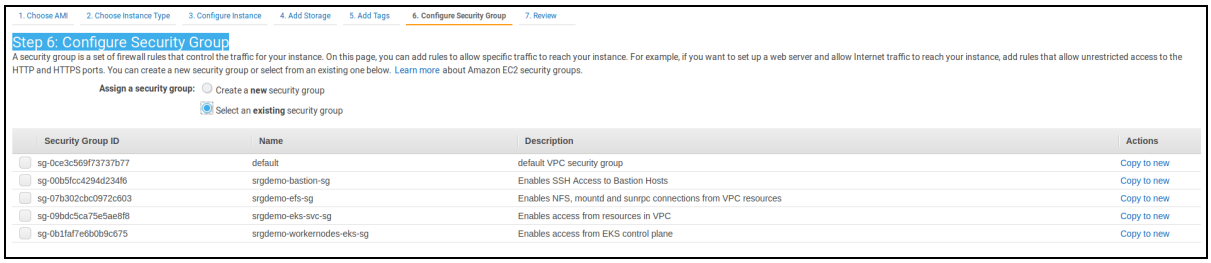
10. On the **Step 4: Add Storage** page, set the root volume size according to your previously-decided needs. In this example, we assume that we will be using it only to upload product images, so we will set it to 20 GB.



11. Enable **Delete on Termination.**
12. Click **Next: Add Tags**.
13. On the **Step 5: Add Tags** page, click **Add Tag.**
14. Specify and save a tag called *Name* with the value of your bastion name (for example, *srgdemo-bastion.*)
15. Optionally, you can add other tags as needed.
16. Click **Next: Configure Security Group.**

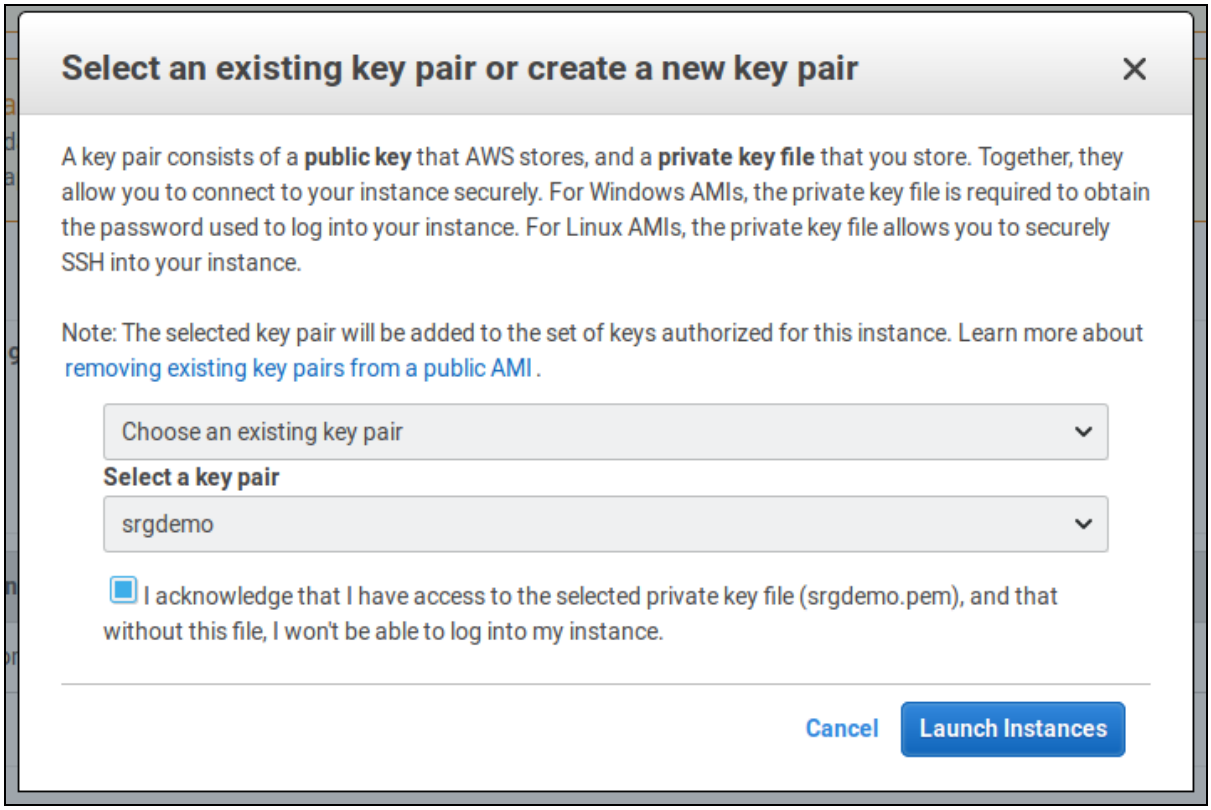


17. On the **Step 6: Configure Security Group** page, under **Assign a security group**, choose **Select an existing security group**.
18. The list shows all security groups associated with your VPC. Select **both** the Bastion security group and Intra VPC security group. (Choose by name or ID from your AWS worksheet.)
19. Click **Review and Launch.**

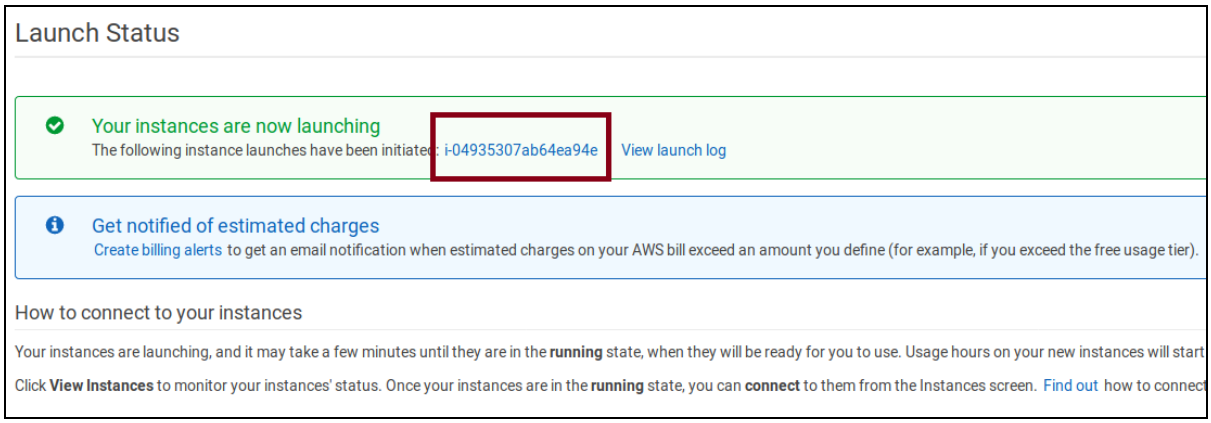


20. On the **Step 7: Review Instance and Launch** page, review all parameters for correctness and fix if necessary. Then click **Launch**.
21. On the **Select an existing key pair...** dialog, pick **Choose an existing key pair** from the drop-down, then select your previously-created key pair.

Important! Confirm that you have the private part of your key pair accessible on your local host. Without the private part, your bastion will not be accessible through SSH.



22. Click **Launch Instances**. The instance is launched and displayed.



23. From on the **Launch Status** page, from the green box, copy your instance ID to your AWS worksheet under Notes.
24. Click **View Instances** to return to the **Instances management** page.

To start the bastion instance using the AWS CLI:

1. Run the following command:

```
aws ec2 run-instances \
--image-id <Image Id> --count 1 \
--instance-type <Instance type> \
```

```
    --key-name <Key pair name> \
    --security-group-ids <security group Ids> \
    --subnet-id <public subnet Id> \
    --block-device-mappings <device mapping parameters> \
    --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=<bastion
    instance name>}]' 'ResourceType=volume,Tags=[{Key=Name,Value=<bastion
    instance volume name>}]' \
    --associate-public-ip-address | jq '.Instances[].InstanceId'
```

Where:

`<Image Id>`: Your AMI ID.

`<Instance type>`: Your bastion type.

`<Key pair name>`: Name of the key pair previously created in SSH Keypair.

`<Security group IDs>`: IDs of the two security groups created by your AWS infrastructure administrators. Add **both** the Bastion Security group Id and Intra VPC Security group ID; separate entries with a single space character.

`<public subnet Id>`: ID of one of the three public subnets created by your AWS infrastructure administrators.

`<Device mapping parameters>`: See the example; used for changing root volume size. For more information about parameters and values, please run:
`aws ec2 run-instances help`.

`<bastion instance name>`: Name assigned to the bastion instance for easier identification.

`<bastion instance volume name>`: Name for the storage volume attached to the current bastion instance.

Example:

```
aws ec2 run-instances --image-id ami-04cf43aca3e6f3de3 \
--count 1 --instance-type t2.medium \
--key-name srgdemo --security-group-ids sg-00b5fcc4294d234f6 sg-
0ce3c569f73737b77 \
--subnet-id subnet-0c0ca63f2f793907d \
--block-device-mappings DeviceName=/dev/sda1,Ebs={VolumeSize=10} \
--tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=srgdemo-
bastion}]' 'ResourceType=volume,Tags=[{Key=Name,Value=srgdemo-bastion-
volume}]' \
--associate-public-ip-address | jq '.Instances[].InstanceId'
```

2. The command returns an instance ID (for example, `i-06773a3ef6acd24f0`). Record your instance ID to the AWS worksheet.
3. Check the instance status by running the following command:
   ```
   aws ec2 describe-instances \
   --instance-ids <Instance Id> | jq '.Reservations[].Instances[].State'
   ```

Example output (JSON):

```
{
   "Code":16,
   "Name":"running"
}
```

4. Repeat the check until the result shows *Name: running*. For example:
   ```
   aws ec2 describe-instances \
   --instance-ids i-06773a3ef6acd24f0 | jq '.Reservations[].Instances
   [].State'
   ```
5. For easier identification of your bastion instance, tag it with a name by running the following command:
   ```
   aws ec2 create-tags --resources <Instance Id> \
   --tags Key=Name,Value=<tag value>
   ```

For example:

```
aws ec2 create-tags \
--resources i-06773a3ef6acd24f0 \
--tag Key=Name,Value=srgdemo-bastion
```

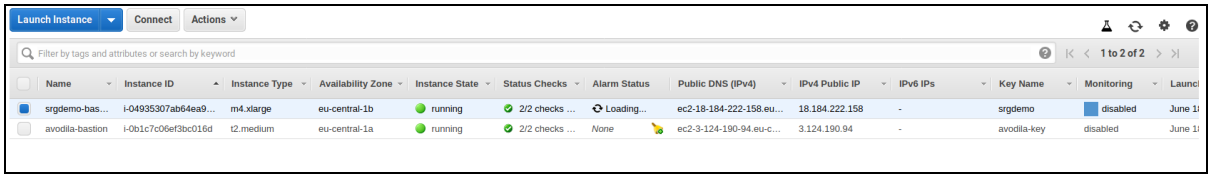
Next Step: Retrieving the Bastion Instance IP Address

Retrieving the Bastion Public IP

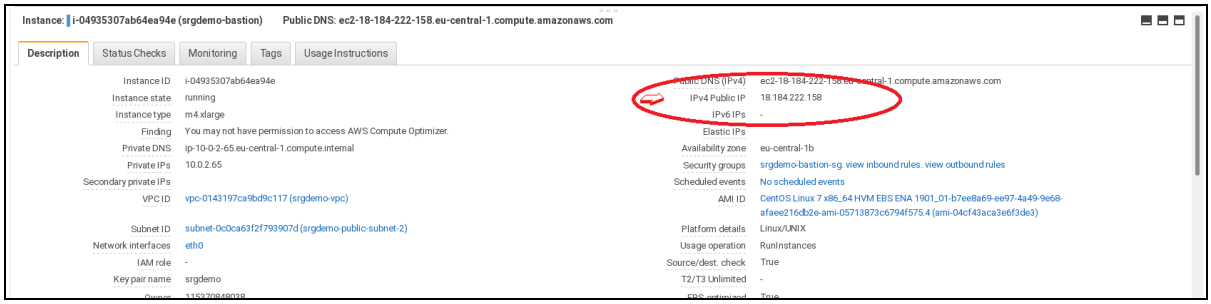
The bastion host provides the only access to the VPC and its resources. Therefore, you will need to know how to connect to it through its public IP address.

To retrieve the bastion public IP address using the Web UI:

1. On the **View Instances** page, select the bastion instance.



2. On the **Description** tab, locate the public IP address.



3. Record the bastion public IP address in the AWS worksheet.

To retrieve the bastion public IP address using the CLI:

1. Run the following command:
   ```
   aws ec2 describe-instances \
   --instance-ids <Instance Id> | jq -r '.Reservations[].Instances
   [].PublicIpAddress'
   ```
2. Record the bastion public IP in the AWS worksheet.

Example input and output:

```
aws ec2 describe-instances \
--instance-ids i-06773a3ef6acd24f0 | jq -r '.Reservations[].Instances
[].PublicIpAddress'
```

```
18.184.151.208
```

Next Step: Connect to the Bastion and Download Software

Connect to Bastion and Install Software Packages

Using the bastion's public IP address and the private part of your key pair, you will connect to the Bastion, install required tools, and perform several configuration tasks.

In examples, we assume the keypair is stored in `~/.ssh`

To connect to Bastion and install required software packages:

Run the following command:

```
ssh -i ~/.ssh/<key pair name>.pem centos@<Bastion Public IP address>
```

For example:

```
ssh -i ~/.ssh/srgdemo.pem centos@18.184.151.208
```

Installing kubectl

Next, you will need to install the kubectl tool for Kubernetes. AWS continually updates the kubectl version. You must use the version corresponding to the Kubernetes version used for the cluster (with a tolerance of one minor version).

Check the ITOM Platform: What's New page for the supported Kubernetes version for your version of CDF. Use only release and major version, for example, 1.15.10 would correspond to 1.15.

You will be managing your EKS cluster with this version of kubectl, so its version must match the version required by CDF.

To install kubectl:

1. Record the required Kubernetes version in the AWS worksheet.

In the following list of commands, find the URL for the kubectl version on the page Installing kubectl. Then replace the curl -o kubectl... command below with the correct command from that page.

2. (CentOS only) If you have not installed the epel package, run the following command:

```
sudo yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

3. Run the following commands to configure the bastion:

```
sudo yum install -y vim docker mc nfs-utils unzip jq htop ncdu nload nano xauth firefox
```

- **CentOS deployment.** Install the required tools packages:

```
sudo groupadd docker
sudo usermod -a -G docker root
sudo usermod -a -G docker centos
```

```
sudo systemctl start docker
sudo systemctl enable docker
curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
```

```
chmod +x ./kubectl
sudo mv kubectl /usr/bin
```

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o
"awscliv2.zip"
```

```
unzip awscliv2.zip
sudo ./aws/install -i /usr/local/aws -b /usr/local/bin
```

- **Amazon Linux 2 bastion.** Install the required tools packages:

```
sudo yum -y update
```

```
sudo yum install -y make gcc perl-core pcre-devel wget zlib-devel
```

```
sudo groupadd docker
sudo usermod -a -G docker root
sudo usermod -a -G docker ec2-user
```

```
sudo systemctl start docker
sudo systemctl enable docker
```

```
curl -LO "https://dl.k8s.io/release/$(curl -L -s
https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
```

```
chmod +x ./kubectl
sudo mv kubectl /usr/bin
```

4. Update the AWS CLI version by running these commands:

```
aws --version
which aws
sudo rm /usr/bin/aws
sudo rm /usr/bin/aws_completer
aws --version
```

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o
"awscliv2.zip"
```

```
unzip awscliv2.zip
sudo ./aws/install -i /usr/local/aws -b /usr/local/bin
hash aws
aws --version
```

Configuring AWS CLI

To configure AWS CLI:

Connect to the bastion machine, and run the following command:

```
aws configure
```

You will be prompted for the same security data, region, and output format when you previously configured `AWS CLI` for the local host. Repeat the process outlined for AWS CLI configuration.

Next Steps: Download Installation Packages

Downloading Installation Tools and Packages

Download the installation packages for the CDF Installer and the product of your choice from the Micro Focus Entitlement Portal to a secure network location. After download, validate the digital signature of each file. You can store all the packages on you local computer, as most of the tasks could be performed on it.

For installation, you must have the following files (each package requires its corresponding `md5` file for authentication):

```
arcsight-platform-cloud-installer-XX.X.X.XXX.zip/.md5
```

```
arcsight-installer-metadata-<version>.tar/.md5
```

```
<product package file> tar/md5 [1 set for each product package you intend to install]
```

Installation tools

The `arcsight-platform-cloud-installer-XX.X.X.XXX.zip` archive contains utility scripts and some templates used during the deployment process. The `aws_scripts/scripts` directory includes these scripts:

- `generate_aws_secrets`: Used to generate new Kubernetes Secrets for connecting from the cluster to the Elastic Container Registry (ECR). Generated credentials/secrets are valid only 12 hours after generation. For accessing the ECR after this timeframe, use this script according to Refresh the ECR credentials in the K8s.
- `init_efs`: Used to create the required folder structure on the Elastic File Storage (EFS) and assign correct ownership and permissions. You will use it when configuring EFS for the

ArcSight Suite. Parameters for this script are discussed in the following sections. Execute this script without parameters to display the help.

- `upload_images_to_ECR`: Used for uploading the CDF and product images to the ECR to make them accessible to K8s. The script performs tasks in the background required specifically by the AWS ECR. Parameters for this script are detailed below. Execute this script without parameters to display the help.
- `workernodes-userdata` : Used indirectly for enabling worker nodes to join the Kubernetes cluster.

Next Step: Creating and Configuring EFS

Installing the Database in AWS

This section provides information about installing the ArcSight Database in Amazon Web Services Deployment (AWS).

- Understanding Methods for Connecting to AWS S3 Buckets
- Launching Database Instances in AWS
- Updating CentOS (conditional)
- Creating the Swap File
- Formatting Devices for the Installation
- Persisting Operating System Settings
- Enabling Root Login for AWS Passwordless Communication
- Setting Up and Installing the Database for AWS

Understanding Methods for Connecting to AWS S3 Buckets

The database uses a single communal storage location for all data and for the catalog (metadata). Communal storage is the database's centralized storage location, shared among the database nodes. This mode supports communal storage in Amazon S3, which must be set up by your cloud administrator before you can install the database.

Other prerequisites or considerations for the S3 bucket:

- Set up default encryption for the S3 bucket in advance. For information about enabling S3 bucket encryption, see AWS documentation, *Enabling Amazon S3 default bucket encryption*.
- If you require a folder under your S3 bucket, it must be created in the communal storage procedure. Folders pre-created under the bucket via the AWS console are not supported for a new installation of the database.

  For more information, see Configuring and Installing the Database Server

- The database supports connecting to AWS S3 buckets using IAM roles. IAM roles are the default access control method for AWS resources. The database uses this method if you do not configure the legacy access control session parameters.

  To use an IAM role, the bucket must be in the same region as the database cluster and the role needs to be set with the proper permissions for reading and writing to the S3 bucket.

  For more information about IAM roles, see AWS documentation, *IAM Roles for Amazon EC2* and *Creating a role to delegate permissions to an AWS service*.

  The example policy below shows the permissions needed for the IAM policy role:

```
 {
"Sid": "s3CommunalLocationAccess",
"Action": [
"s3:ListBucket",
"s3:GetObject",
"s3:DeleteObject",
"s3:PutObject",
"s3:*MultipartUpload*"
],
"Resource": [
"arn:aws:s3:::<db-communal-storage-bucket-name>/*",
"arn:aws:s3:::<db-communal-storage-bucket-name>"
]
}
```

  In the example above you would replace the <db-communal-storage-bucket-name> with the name of the S3 bucket you created for the database communal storage. For example, `222222222-ap-southeast-1-arcsight-db`

Launching Database Instances in AWS

The ArcSight database compute operations are performed on a cluster of AWS EC2 (Elastic Compute Cloud) nodes. These nodes must be deployed in AWS before the database can be installed on them.

1. Sign in to the Amazon EC2 console and select a region. For more information, see Understanding Methods for Connecting to AWS S3 Buckets
2. Select a launch instance from the **Choose AMI** tab. The database supports Red Hat 8.4 and CentOS 8.4 AMIs. For information about finding an AMI, see AWS documentation, *Find a Linux AMI*.

   For example. If you are in the US East (N. Virginia) region, you can use the following AMIs:

- Red Hat 8.4 (AMI Id: ami-0453b4b64c860454a)
- CentOS 8.4 (AMI Id: ami-05b1369539e0e69bd)

3. After selecting the AMI, and reviewing the details, select an instance type. Choose the AWS EC2 instance type that best matches your requirements.

   We recommend using an m5d instance type, because it has the best balance of cost, performance, and ease of setup for ArcSight workloads. Take into account that m5d is an instance store. Instance stores provide temporary storage. For example, data files stored on the instance are lost when the instance is stopped. The Durable EBS instance type is also supported with Elastic Block Store (EBS). This instance type provides durable storage. For example, data files stored on the instance persist after the instance is stopped.

   For more information on which size of m5d (Instance Store Type) and /m4 (EBS Type) should be used for your workload, see the ArcSight Platform Technical Requirements.

4. Configure the number of AWS EC2 instances to deploy. A minimum of three is needed for a highly available database cluster, but more may be needed to handle your workload. For more information on how many instances you need, see the ArcSight Platform Technical Requirements.
   a. Configure your instance with the **VPC** and **private subnets**.
   b. Select the IAM role you configured to have access to the S3 bucket for the database communal storage.
   c. Click **Next: Add Storage**.
5. On the Add Storage page, change the size for the Root volume type to 30 GiB or higher. 30 GiB is the minimum size for root.

   If you are using EBS instance type m4, add an EBS volume according to ArcSight Platform Technical Requirements.

6. Click **Next: Add Tags**, and add tags as needed.
7. Click **Next: Configure the Security Group**. Configure your instance with your private security groups.
8. Click **Review and Launch** to review the details of your instance.
9. Click **Launch**, and select an existing key pair or create a new key pair.
10. Click **Launch Instances**.
11. Navigate to EC2, and verify your instances are available.

Updating CentOS (conditional)

If you are deploying the database with CentOS 8.4 2105, you need to update the distros by running the commands below on all database nodes:

```
sudo dnf --disablerepo '*' --enablerepo=extras swap centos-linux-repos
centos-stream-repos
sudo dnf distro-sync
```

If you discover that the above distro repository is broken, update the `--enablerepo` repositories with those provided by CentOS: https://www.centos.org/centos-stream/.

Creating the Swap File

A partition with swap space configured is required to enable virtual memory access. The database requires a minimum of 2 GB of swap space on all database nodes. Perform the following procedure on all database nodes to setup the swap space.

1. From the bastion machine, login to the database node using SSH and the key pair selected during instance launching.
2. Create the file that will be used for swap:

   ```
   sudo dd if=/dev/zero of=/swapfile bs=1024 count=2621440
   ```

   The count 261440 is the minimum value required for installation, but it is also an example. You can set the value higher.
3. Enable read and write to the swap file for the root user (only):

   ```
   sudo chmod 600 /swapfile
   ```
4. Set up the file as a Linux swap area:

   ```
   sudo mkswap /swapfile
   ```
5. Enable the swap with the following command:

   ```
   sudo swapon /swapfile
   ```
6. Enable the swap permanently as root user by editing the `/etc/fstab` file and appending the following line:

   ```
   echo "/swapfile swap swap defaults 0 0" | sudo tee -a /etc/fstab
   ```

7. Verify that swap is enabled:

```
sudo swapon --show
```

```
Output example:

NAME      TYPE SIZE USED PRIO
/swapfile file 2.5G   0B   -2
```

Formatting Devices for the Installation

The database requires a storage device formatted in ext4 for all nodes. Perform the following procedure on all database nodes to setup ext4 in devices.

The steps below are required to be performed on all AWS EC2 instances that are created. For more information about creating instances, see Launching Database Instances in AWS

Format the device type applicable to the AWS EC2 type you are using in the instance, either an instance Store type or EBS type:

- Formatting Instance Store Type Devices
- Formatting EBS Type Devices

Formatting Instance Store Type Devices

The steps below are required to be performed on all AWS EC2 instances that are created.

ONLY perform this procedure if you are using **Instance Store** type devices.

1. From the Bastion machine, login to the database node using SSH and the key pair selected during launching.
2. Install the `nvme` tool:

```
sudo dnf install -y nvme-cli
```

3. Create an `/opt mount` point. If `/opt` alread exists, skip this step.

```
sudo mkdir /opt/vertica
```

4. Use the `nvme` tool to locate the `nvme` device and format it:

```
EPHEMERAL_DISK=$(sudo nvme list | grep 'Amazon EC2 NVMe Instance Storage'
| awk 'NR==1{ print $1 }')
```

```
echo $EPHEMERAL_DISK
```

```
sudo mkfs.ext4 $EPHEMERAL_DISK
```

```
sudo mount -t ext4 $EPHEMERAL_DISK /opt
```

5. Add the ephemeral disk mount to `/etc/fstab`:

```
echo "$EPHEMERAL_DISK /opt ext4 defaults,nofail 0 0" | sudo tee -a
/etc/fstab
```

6. To list the mount points for verification, run the `lsblk` command:

```
lsblk
```

Formatting EBS Type Devices

The steps below are required to be performed on all AWS EC2 instances that are created. These steps for EBS also assume you are formatting a single device. For more information about creating instances, see Launching Database Instances in AWS.

ONLY perform this procedure if you are using **EBS** type devices.

1. Run the following command to locate the EBS device attached to the instance, and to select the device that is not mounted:

```
lsblk -f
```

```
#Output from lsblk -f
NAME FSTYPE LABEL UUID MOUNTPOINT
xvda
└─xvda1 xfs 35761952-413c-42e8-b047-a5deb7510f29 /
xvdb
```

2. Run the following command to format the device to ext4:

```
sudo mkfs.ext4 /dev/xvdb
```

NOTE: `xvdb` is an example and will be <yourDeviceName> from the output in the previous step.

3. Create the mount point. If `/opt` already exists, skip this step.

```
sudo mkdir /opt/vertica
```

4. Format the device:

```
sudo mount -t ext4 /dev/xvdb /opt
```

5. Modify the `/etc/fstab` file, and add an entry to mount the device on OS boot:

```
echo "/dev/xvdb /opt ext4 defaults 0 0" | sudo tee -a /etc/fstab
```

Persisting Operating System Settings

The database requires that you manually configure several general operating system settings. Perform the following procedures on all database nodes to setup ext4 in devices.

1. Run this command to set the limit for open files so that it meets database requirements. This will add the parameters to the `/etc/sysctl.conf` file.

```
cat << EOF | sudo tee -a  /etc/sysctl.conf
net.core.somaxconn = 1024
net.core.wmem_max = 16777216
net.core.rmem_max = 16777216
net.core.wmem_default = 262144
net.core.rmem_default = 262144
net.core.netdev_max_backlog = 100000
net.ipv4.tcp_mem = 16777216 16777216 16777216
net.ipv4.tcp_wmem = 8192 262144 8388608
net.ipv4.tcp_rmem = 8192 262144 8388608
net.ipv4.udp_mem = 16777216 16777216 16777216
net.ipv4.udp_rmem_min = 16384
net.ipv4.udp_wmem_min = 16384
vm.swappiness = 1
EOF
```

2. Update the`/etc/rc.`local file by running the commands below. This file contains scripts and commands that run each time the system is booted, and the database requires that I/O Scheduling be set to *deadline* or *noop*. The command will add the applicable lines to the file, based on the devices type in your AWS instance, either **instance Store** type or **EBS** type:

   Instance Store type rc.local settings (only)

```
cat << EOF | sudo tee -a /etc/rc.local
echo deadline > /sys/block/nvme0n1/queue/scheduler
echo deadline > /sys/block/nvme1n1/queue/scheduler
echo deadline > /sys/block/nvme2n1/queue/scheduler
echo deadline > /sys/block/md0p1/queue/scheduler
```

```
/sbin/blockdev --setra 2048 /dev/nvme0n1
/sbin/blockdev --setra 2048 /dev/nvme1n1
/sbin/blockdev --setra 2048 /dev/nvme2n1
/sbin/blockdev --setra 2048 /dev/md0p1

if test -f /sys/kernel/mm/transparent_hugepage/enabled; then
echo never > /sys/kernel/mm/transparent_hugepage/enabled
fi

if test -f /sys/kernel/mm/transparent_hugepage/defrag; then
echo never > /sys/kernel/mm/transparent_hugepage/defrag
fi
EOF
```

Note: Replace the device in the example command above with your own instance device on which the `/opt` is mounted. For example, the following device must be replaced with your own device: `sys/block/nvmen0n1`

EBS type rc.local settings (only)

```
cat << EOF | sudo tee -a /etc/rc.local
echo deadline > /sys/block/xvda/queue/scheduler
echo deadline > /sys/block/xvdb/queue/scheduler
echo deadline > /sys/block/xvdc/queue/scheduler

/sbin/blockdev --setra 2048 /dev/xvda
/sbin/blockdev --setra 2048 /dev/xvdb
/sbin/blockdev --setra 2048 /dev/xvdc

if test -f /sys/kernel/mm/transparent_hugepage/enabled; then
echo never > /sys/kernel/mm/transparent_hugepage/enabled
fi

if test -f /sys/kernel/mm/transparent_hugepage/defrag; then
echo never > /sys/kernel/mm/transparent_hugepage/defrag
fi
EOF
```

Note: Replace the device in the example command above with your own instance device on which the `/opt` is mounted. For example, the following device must be replaced with your own instance device:

`sys/block/dev/xvdb`

3. Modify the file permissions:

```
sudo chmod +x /etc/rc.d/rc.local
```

4. Run the following commands to disable the system firewall:

```
sudo systemctl mask firewalld
sudo systemctl disable firewalld
sudo systemctl stop firewalld
```

During installation, the database requires that host-based firewalls are disabled on database nodes. After installation, the host-based firewalls can be enabled and the database requires several ports to be open on the local network. We recommend for optimal performance using host-based firewalls between database nodes and a network-based firewall to protect the segment that database cluster is within. However, there is no restriction against using a network-based firewall between database nodes. When using any kind of firewall, ensure that all the database ports are available. For more information, see Firewall Considerations.

5. Set SELinux to permissive mode in `/etc/selinux/config`.

```
SELINUX=permissive
```

For more information, see SELinux Configuration.

6. Run this command to ensure that `rng-tools` packages are installed in all cluster nodes:

```
sudo dnf install rng-tools -y
```

7. Set the UTC time for all cluster nodes:

```
sudo timedatectl set-timezone UTC
```

For CentOS 8.4, any changes to the timezone will require a cluster nodes reboot.

8. Reboot the system for your changes to take effect.

This step is required to update the device mounts and other settings.

Enabling Root Login for AWS Passwordless Communication

All commands require root privileges which can be obtained through the sudo command.

1. Connect to one of your nodes and edit the `/etc/ssh/sshd_config` configuration file.

```
sudo vi /etc/ssh/sshd_config
```

2. Change the following parameter to `yes` if the value is not already set to that.

```
PermitRootLogin yes
```

3. Proceed as follows:
   - If you had to change the `PermitRootLogin` parameter to `yes`, run this command:

     ```
     sudo service sshd reload
     ```

   - If the `PermitRootLogin` parameter was already set to `yes`, proceed to the next step.
4. Patch the authorized keys file for the root user by copying the centos file to the root user. This enables logging in with the same key which is available for the centos-user/ec2-user (rhel-8.4).

   ```
   sudo cp ~centos/.ssh/authorized_keys ~root/.ssh/authorized_keys
   ```

5. Update the AWS cloud configuration file by editing `/etc/cloud/cloud.cfg` and changing the `disable_root` value to 0.

   ```
   sudo vi /etc/cloud/cloud.cfg
   ```

6. Repeat all the above steps for all nodes in the cluster.
7. Generate SSH keys for the database nodes:
   a. Log in to one of the database nodes. This becomes the initiator node (node1) for your cluster.
   b. Run the following command as root:

      ```
      ssh-keygen -q -t rsa -f ~/.ssh/id_rsa
      ```

   c. Run the following command as root, for each of the database nodes (initiator node inclusive):

      ```
      ssh-copy-id -f "-o IdentityFile <FilePath to the ec2 keypair>"
      root@<node1..n>
      ```

      ```
      Enter : Yes
      ```

   d. To verify the passwordless communication, run the following command:

      ```
      ssh root@<node1,node2..>
      ```

      ```
      Output example:

      Activate the web console with: systemctl enable --now cockpit.socket
      Last login: Thu Jan 20 19:26:50 2022 from <node1,node2..>
      ```

Setting Up and Installing the Database for AWS

Complete the procedures below in succession:

1. Modifying the System Clock
2. Enabling FIPS Mode on the Database Server (conditional)

   This step is needed only if your environment requires FIPS.
3. Configuring and Installing the Database Server
4. Creating the Elastic File System

Configuring and Installing the Database Server

Before installing the database, ensure that you estimate the storage needed for the incoming EPS (event per second) and event size, and also evaluate the retention policy accordingly. You also must create an IAM role prior to installing the database. For information about creating an IAM Role, see Understanding Methods for Connecting to AWS S3 Buckets.

Perform the following steps as root user:

1. On the Database cluster node1 server, create a folder for the database installer and enable permissions. For example:

   ```
   mkdir /opt/arcsight-db-tools
   chmod 755 /opt/arcsight-db-tools
   ```

   `/opt/arcsight-db-tools` should not be under `/root` or `/opt/vertica`.
2. From the master node where you performed the Downloading Installation Packages steps, copy the following file on the Database cluster node1 server:

   `{unzipped-installer-dir}/installers/database/db-installer_x.x.x-x.tar.gz`

   to the `/opt/arcsight-db-tools` directory
3. To extract the installer file and place it in the correct directory, run the following commands:

   ```
   cd /opt/arcsight-db-tools
   tar xvfz db-installer_x.x.x.x.tar.gz
   ```
4. Edit the `config/db_user.properties` file and add all database node IPs to the hosts property.

Property	Description
hosts	A comma separated list of the database servers in IPv4 format (for example, 1.1.1.1,1.1.1.2,1.1.1.3). If it is necessary to construct the cluster, avoid using local loopback (localhost, 127.0.0.1, etc.).

5. Install the database.

```
./db_installer install
```

6. When prompted, create the **database administrator** user.

   The database administrator user account is used during database deployment, configuration, upgrade, and debugging. For security reasons, the platform deployed capabilities will not ask you for the credentials for this user.

   For a list of options that you can specify when installing the database, see Understanding the Database Installer Options.

7. Specify the shard count. We recommend a shard count of 3 for single-node, or a count of 18 for multi-node to allow for scalability. The prompt options are based on your environment, single-node or multi-node:

   Once the database is installed, this value cannot be changed.

   - Single-node:

```
# ========================================
# STEP 1: Specify Database Shard Count for Eon Mode
Do you plan to keep the database cluster to a single node in the
future?
If yes, the database will be optimized for performance on a single node
by setting the default shard count to 3.
Shard Count [3]:
Shard count cannot be changed after installation.
Confirm shard count [3]?(y/n):y
Check memory size, 48GB required for single node installation with
shard count > 3.
PASS: Single node installation for shard count: 3
```

   - Multi-node:

```
# ========================================
# STEP 1: Specify Database Shard Count for Eon Mode
Recommended shard count for multi node database deployment is 18.
Shard Count [18]:
```

```
Shard count cannot be changed after installation.
Confirm shard count [18]?(y/n):y
```

8. Set up the communal storage type for S3 when prompted. For example:

```
# STEP 2: Specify communal storage details
Supported communal storage types -
1) S3
2) Azure Blob Storage
Choose a communal storage type from the above (1/2):1
Are you using IAM role authentication for AWS S3 Storage?(y/n):y
Specify S3 bucket for communal storage:<yourS3BucketName>
Specify the folder under bucket for communal storage if
applicable:<newFolderNametoCreate>
Communal storage url is: <s3://<yourS3BucketName> /newFolderNametoCreate>
```

If you require a folder under your S3 bucket, it must be created in the communal storage procedure shown above. Folders pre-created under the bucket via the AWS console are not supported.

ArcSight database AWS communal storage supports the S3 Intelligent-Tiering storage class. To learn more details about how to automatically configure the storage lifecycle rules, see AWS documentation, *Automate S3 Lifecycle rules at scale to transition data to S3 Intelligent-Tiering*.

9. Create the schema.

```
./db_installer create-schema
```

10. When prompted, create the following users:
    - **App admin user:** A regular database user granted elevated permissions for performing operations on the database to manage the database, schema, and resource pools. The credentials for this user will need to be provided later in the CDF Management Portal when you are deploying capabilities.
    - **Search user:** A regular database user with permissions restricted to event search operations. The credentials for this user will need to be provided later in the CDF Management Portal when you are deploying capabilities.
11. Monitor your database cluster status constantly. For more information, see Monitoring the Database.
    - **Database nodes status:** Ensures all nodes are up
    - **Database nodes storage status:** Ensures storage is sufficient

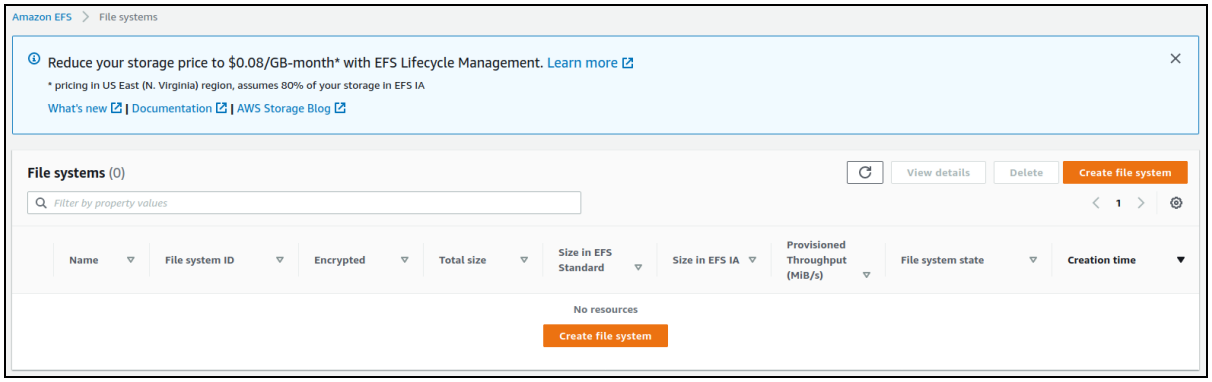
Note: If you have a Recon license, the default retention period for Default Storage Group events is 12 months. You can modify this value based on your data storage policy. If you do not have a Recon license, the retention period for the Default Storage Group is one month.

Creating the Elastic File System

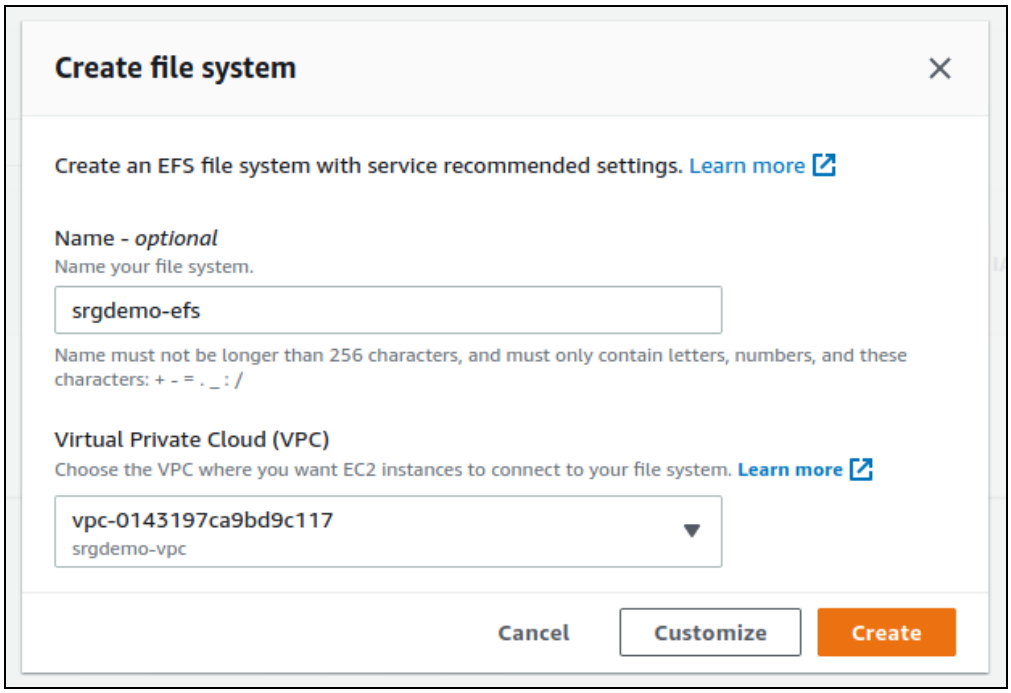
Amazon Elastic File System (Amazon EFS) provides a simple, scalable, fully managed elastic NFS file system. You can create an EFS through the web UI or the CLI.

To create an EFS using the web UI:

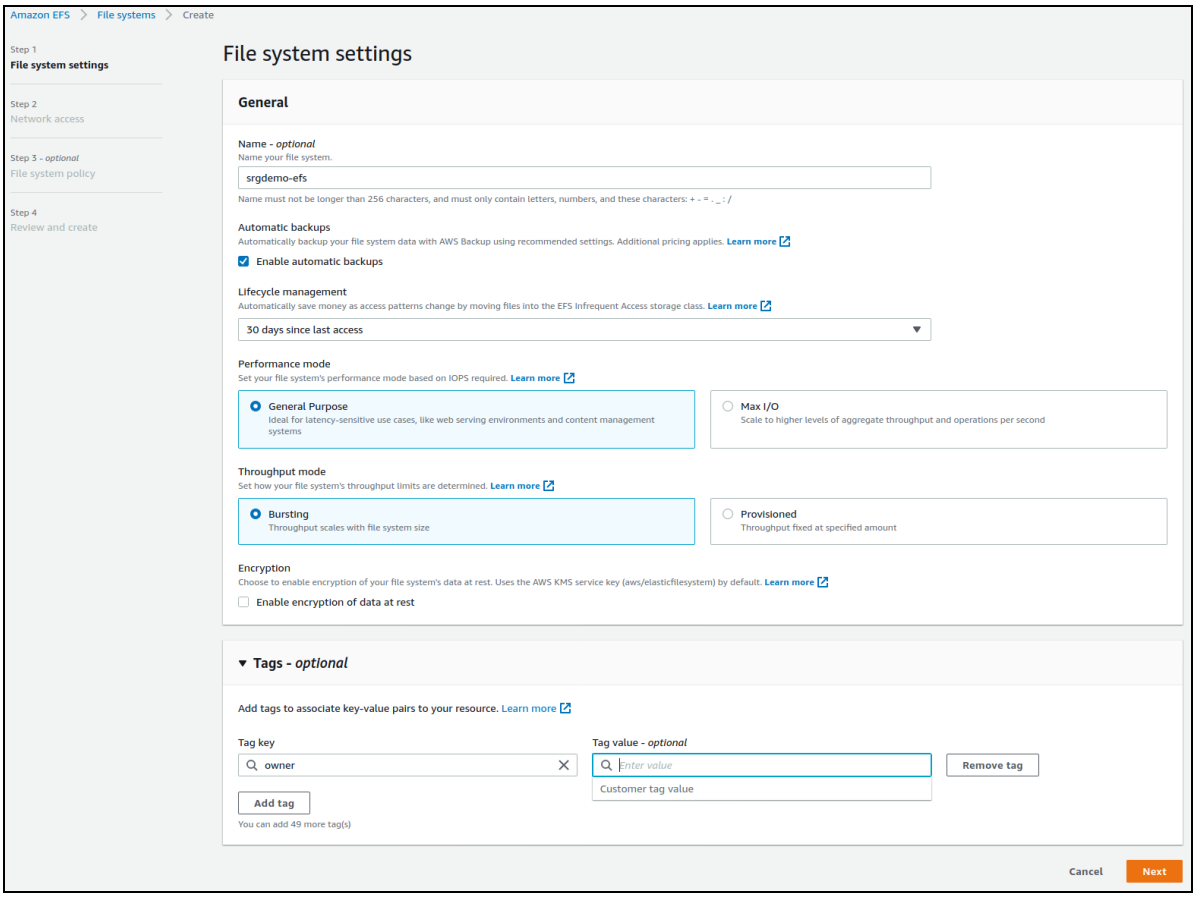
1. Using the Find Services search tool, locate and browse to the EFS dashboard.



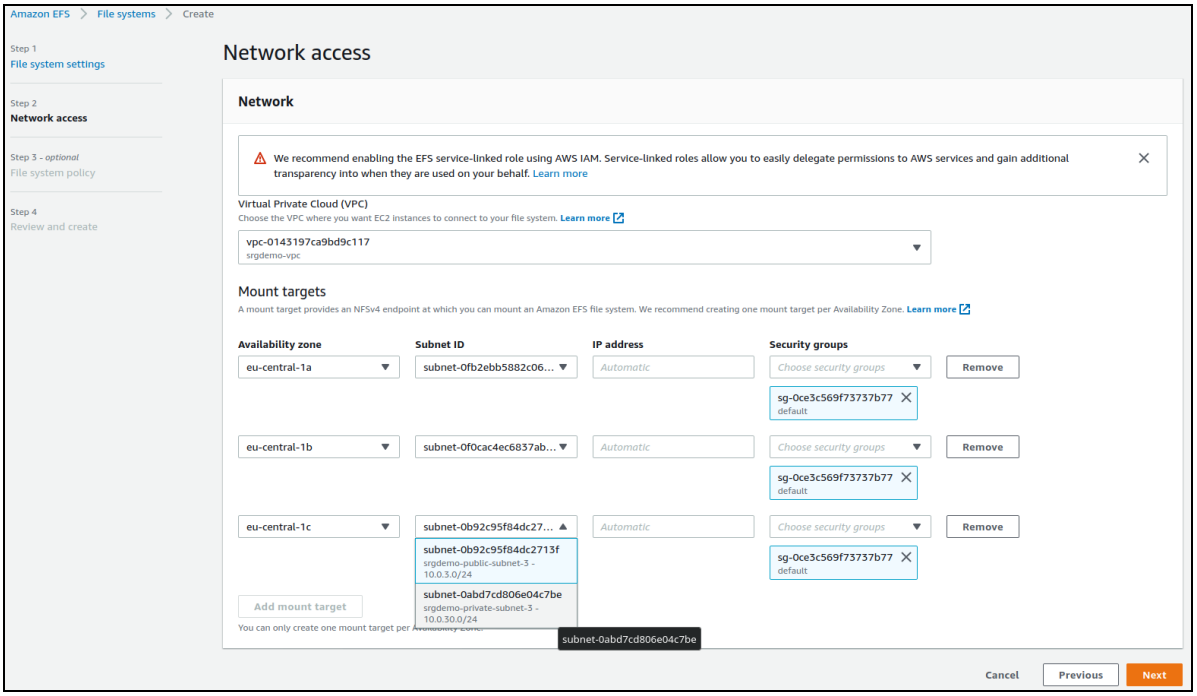
2. Click **Create file system.**



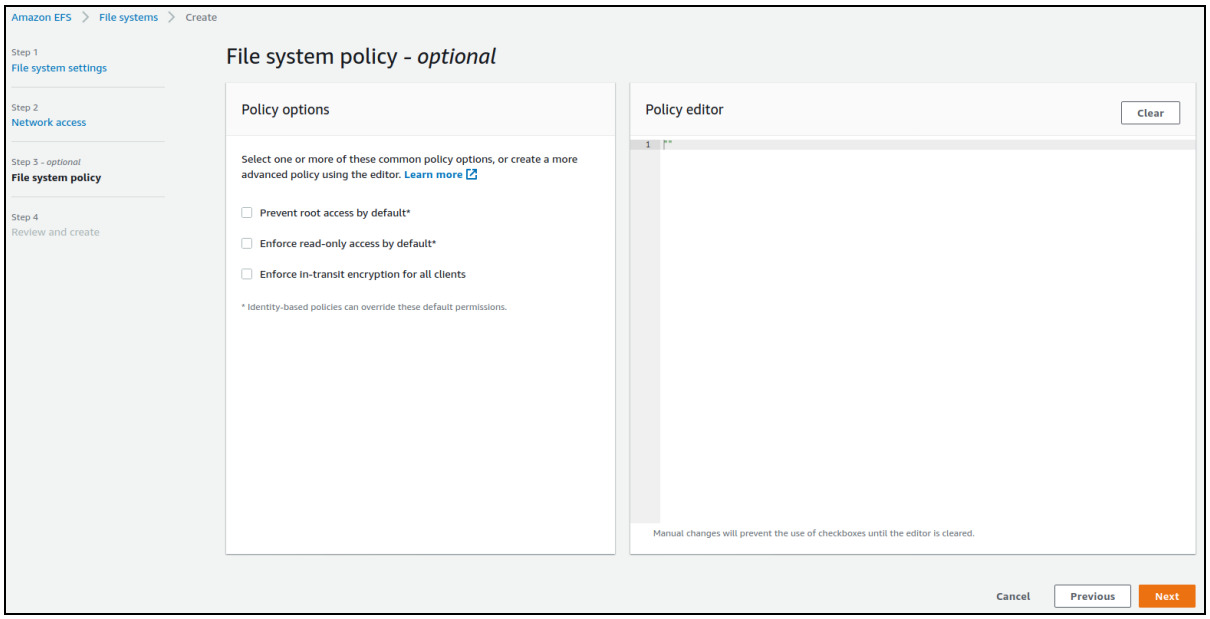
3. On the **Create file system** dialog:
   a. In **Name**, specify a name for the EFS.
   b. Under **Virtual Private Cloud (VPC)**, select the VPC you recorded in your AWS worksheet.
   c. Click **Customize** to start the custom EFS wizard.
4. On the **File system settings** page:
   a. In **General**, deselect **Enable encryption of data at rest.**
   b. In **Tags**, add one or more identification tags as desired.
   c. Click **Next**.



5. On the **Network Access** page:
   a. Under **Mount Targets**, for each target:
   b. Under **Subnet ID,** select the subnet ID of the private subnet.
   c. Change the **Security groups** by adding the Intra VPC security group to the dropdown, and remove any other security groups from the list.
   d. Click **Next**.



6. On the **File system policy - optional** page, leave all settings as is and click **Next**.



7. On the **Review and create** page, review all settings for accuracy, then click **Create**. You are redirected to the EFS dashboard.

The creation process may take a few moments; please give the process time to complete.

8. Record the File system ID in the AWS worksheet.

To create the EFS using the CLI:

1. Run the following command to create an encrypted EFS file system:

```
aws efs create-file-system \
```

```
--encrypted \
--tags Key=Name,Value=<EFS name chosen for easy identification>
```

Example input and output:

```
aws efs create-file-system \
```

```
--encrypted \
--tags Key=Name,Value=srgdemo-efs
```

```
{
   "OwnerId":"115370811111",
   "CreationToken":"a53deaf5-ecd6-4dfa-9206-0e1a3db3e1d9",
   "FileSystemId":"fs-ebe456b3",
   "CreationTime":1589557528.0,
   "LifeCycleState":"creating",
   "Name":"srgdemo-efs",
   "NumberOfMountTargets":0,
   "SizeInBytes":{
      "Value":0,
      "ValueInIA":0,
      "ValueInStandard":0
   },
   "PerformanceMode":"generalPurpose",
   "Encrypted":false,
   "ThroughputMode":"bursting",
   "Tags":[
      {
         "Key":"Name",
         "Value":"srgdemo-efs"
      }
   ]
```

2. From the description, record the `FileSystemId` in the AWS worksheet.
3. Examine the value of `LifeCycleState`. Provisioning is complete when the value changes to `available`. (In the example, it has the value `creating`.)
4. To check the provisioning status while the process is running, run the following command:

```
aws efs describe-file-systems \
--file-system-id <FileSystemId>
```

Provisioning usually takes approximately 5 minutes.

Example input and output:

```
aws efs describe-file-systems --file-system-id fs-ebe456b3
```


```
{
   "FileSystems":[
      {
         "OwnerId":"115370811111",
         "CreationToken":"a53deaf5-ecd6-4dfa-9206-0e1a3db3e1d9",
         "FileSystemId":"fs-ebe456b3",
         "CreationTime":1589557528.0,
         "LifeCycleState":"available",
         "Name":"srgdemo-efs",
         "NumberOfMountTargets":0,
         "SizeInBytes":{
            "Value":6144,
            "ValueInIA":0,
            "ValueInStandard":6144
         },
         "PerformanceMode":"generalPurpose",
         "Encrypted":false,
         "ThroughputMode":"bursting",
         "Tags":[
            {
               "Key":"Name",
               "Value":"srgdemo-efs"
            }
         ]
      }
   ]
}
```


Next Step: Create Mount Targets

Creating Mount Targets

A *mount target* connects the EFS to a specific subnet in the VPC. The instances contained in the VPC can mount the target using the NFS protocol and utilize NFS.

In this section, you will create mount targets between the newly-created EFS and all three private subnets.

To create a mount target in a private subnet:

1. Select one of your **private** subnets and run the following command:
   ```
   aws efs create-mount-target \
   --file-system-id <FileSystemId> \
   --security-groups <Intra VPC Security group Id> \
   --subnet-id <private subnet Id>
   ```

Where:

`<FileSystemId>`: The file system ID of the EFS you just created

`<Intra VPC Security group Id>`: The ID of the Intra VPC security group you previously created.

The command only accepts one subnet ID at a time. You must run this command separately for each private subnet which you are using for the cluster.

2. The command will respond with a mount target description. From the output, record the `MountTargetId` in your AWS worksheet.
3. Repeat Steps 1 and 2 for each of the other 2 private subnets (use the subnet IDs on your worksheet) and then record the values of `MountTargetId` for each in your AWS worksheet.

Example input and output:

```
aws efs create-mount-target \
--file-system-id fs-ebe456b3 \
--security-groups sg-07b302cbc0972c603 \
--subnet-id subnet-0fb2ebb5882c061f0
```

```
{
   "OwnerId":"115370811111",
   "MountTargetId":"fsmt-63eaae3a",
   "FileSystemId":"fs-ebe456b3",
   "SubnetId":"subnet-0fb2ebb5882c061f0",
   "LifeCycleState":"creating",
   "IpAddress":"10.0.10.131",
   "NetworkInterfaceId":"eni-03ecba7e5eb46dc9f",
   "AvailabilityZoneId":"euc1-az2",
   "AvailabilityZoneName":"eu-central-1a"
}
```

To check the creation status of a mount target:

1. Run the following command:
   ```
   aws efs describe-mount-targets --mount-target-id <Mount target X Id>
   ```

2. Record the value of `MountTargetId` in the AWS worksheet.

Immediately after creation, a mount target has a value for `LifeCycleState` value of `creating`. The transition to available usually takes approximately 3 minutes. To check the status, run the following command:

```
aws efs describe-mount-targets --mount-target-id <Mount target X Id>
```

Example input and output:

```
aws efs describe-mount-targets --mount-target-id fsmt-63eaae3a
```

```
{
   "OwnerId":"115370811111",
   "MountTargetId":"fsmt-63eaae3a",
   "FileSystemId":"fs-ebe456b3",
   "SubnetId":"subnet-0fb2ebb5882c061f0",
   "LifeCycleState":"creating",
   "IpAddress":"10.0.10.131",
   "NetworkInterfaceId":"eni-03ecba7e5eb46dc9f",
   "AvailabilityZoneId":"euc1-az2",
   "AvailabilityZoneName":"eu-central-1a"
}
```

Once all three mount targets are in the `available` state, you can proceed to the next step.

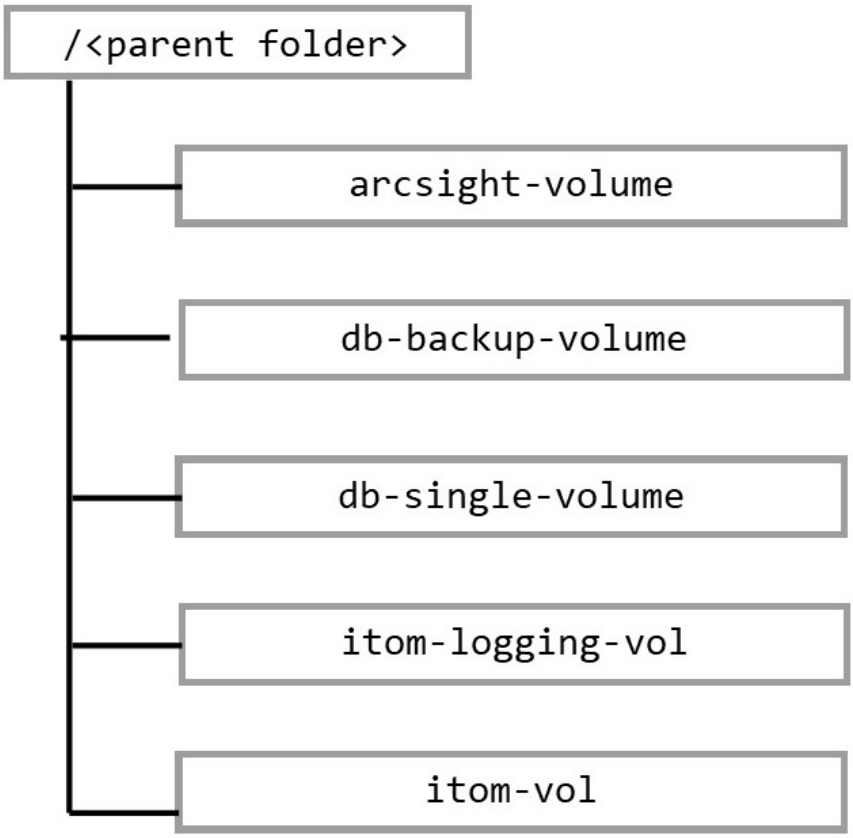
Next Step: Configuring EFS

Configuring EFS for the ArcSight Suite

CDF and the ArcSight suite require several separated folders for storing various types of information, such database files, log files, and runtime data. In this step, you will create the following folders:

- `arcsight-volume`
- `db-backup-vol`
- `db-single-vol`
- `itom-logging-vol`
- `itom-vol`

All of these folders are created in a parent folder from the filesystem, as follows:



Using different parent folders, you can use a single EFS for several different file systems (assuming they are in the same region and same VPC, and have the correct mount targets).

To configure EFS for ArcSight Suite:

1. Using an `scp` client, copy the `arcsight-platform-cloud-installer-XX.X.X.XXX.zip` package to the bastion and unpack it.
2. For creating the folders and setting respective permissions, unzip the `aws-scripts` script archive and then run the script `init_efs` from the `aws-scripts/scripts` directory.
3. Construct the filesystem FQDN. The filesystem FQDN should have the following format: `<FileSystemId>.efs.<Region>.amazonaws.com`

   Where:

   `<FileSystemId>`: Previously created and recorded in the AWS worksheet.

   `<Region>`:The ID of the region in which you have originally asked to create restricted resources.
4. Record the filesystem FQDN in the AWS worksheet.

The FQDN will be used for initializing the folder structures; and will later be used during the Bootstrap CDF step and during CDF Web UI installation processes.

5. Execute the script:

```
./arcsight-platform-cloud-installer/aws-scripts/scripts/init_efs \
-p <Parent folder name> \
-s <Filesystem FQDN>
```

Where:

`<Parent folder name>`: An optional parameter. If not specified, this value will be replaced with ArcSight. Record the chosen parent folder name to the AWS worksheet.

`<Filesystem FQDN>`: The filesystem FQDN you have just created.

Example:

```
./arcsight-platform-cloud-installer/aws-scripts/scripts/init_efs \
-p srgdemo \
-s fs-ebe456b3.efs.eu-central-1.amazonaws.com
```

6. The mount point is created as commented in the `etc/fstab` file. Open the `etc/fstab` file and uncomment the mount point. Then run the command:

```
sudo vim /etc/fstab
```

7. Run the following command:

```
sudo mount -a
```

8. (Conditional) If Intelligence is part of the deployment, run the following command only for `arcsight-volume` so that the Logstash and Elasticsearch pods do not fail because of permission issues:

```
cd /mnt/efs/<parent_folder_name>
sudo chown -R 1999:1999 arcsight-volume
```

9. Verify whether the created folders correspond to the structure described above, with respect to your chosen parent folder.

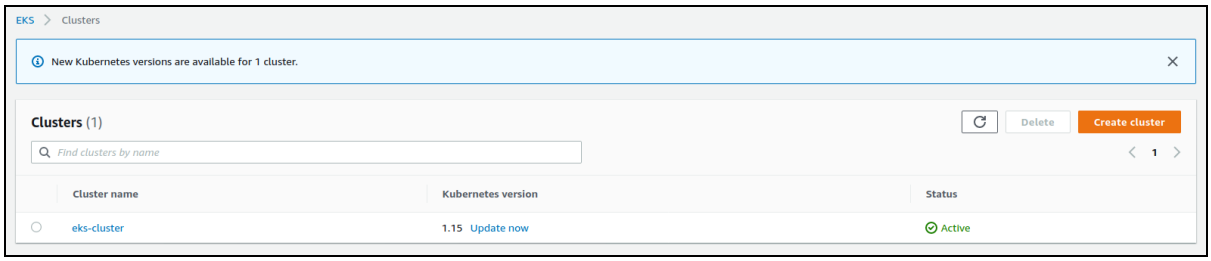
Next Step: Configure EKS

Configuring the Elastic Kubernetes Service

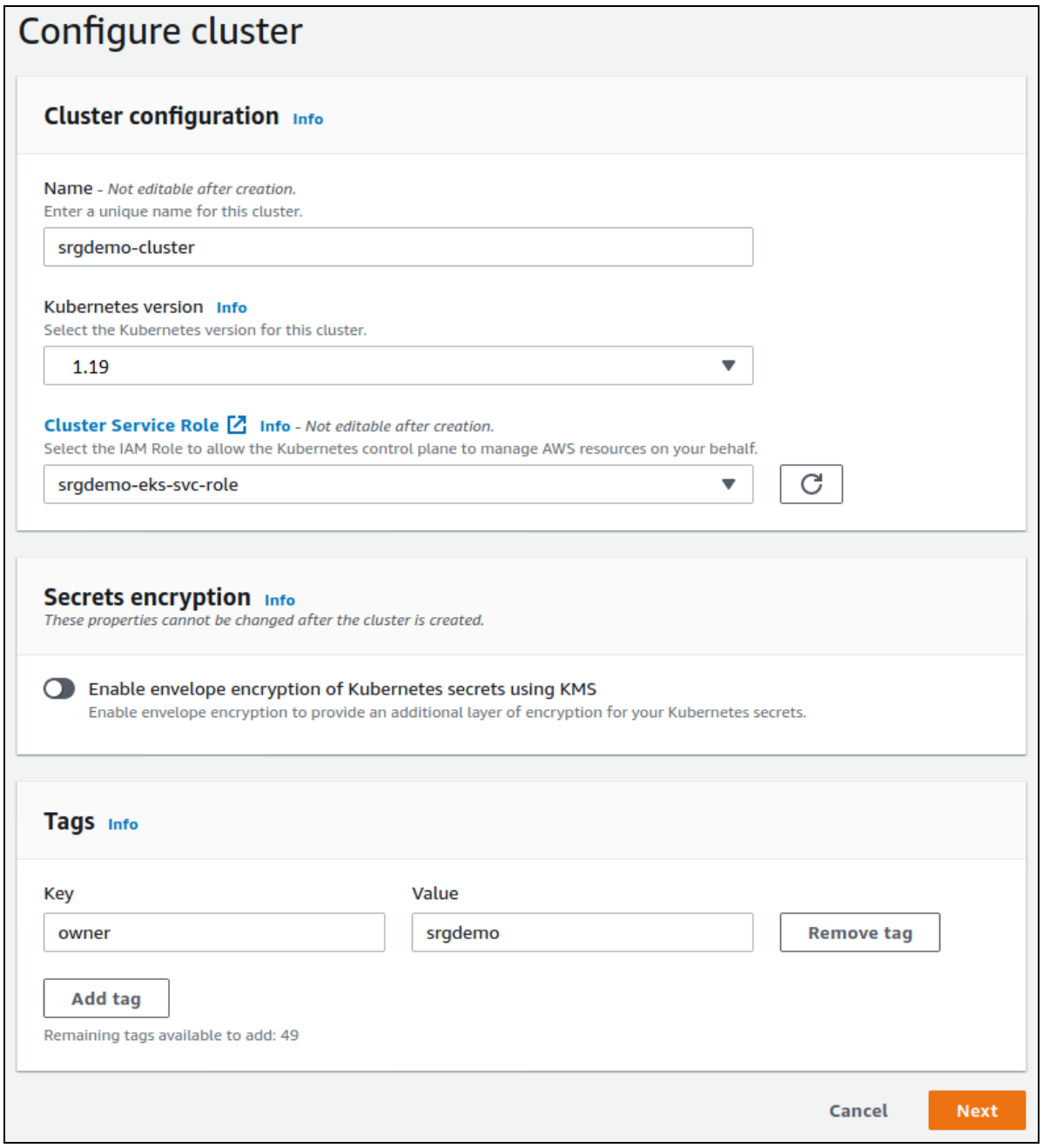
Amazon Elastic Kubernetes Service (Amazon EKS) is a fully-managed Kubernetes service control plane. In this section you will set up your EKS cluster.

To configure EKS using the web UI:

1. Using the Find Services search tool, locate and browse to the EKS Dashboard.
2. Click **Create Cluster.**

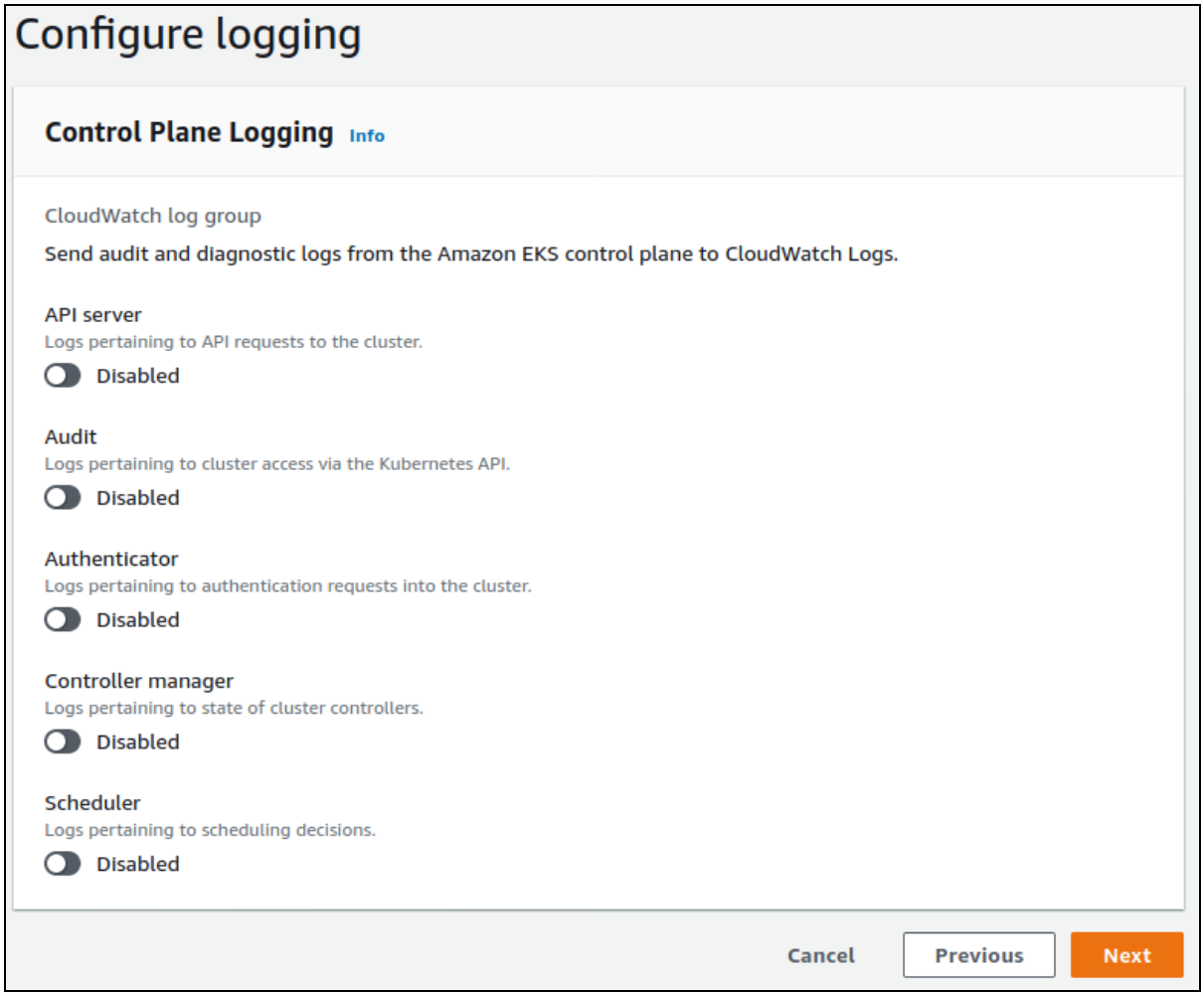


3. On the **Configure Cluster** page, specify values for the following:
   a. **Name**: Cluster name. Use the same value you passed in your resources creation request to your AWS infrastructure administrators, and recorded in the AWS worksheet. For example, `srgdemo-cluster`.
   b. **Kubernetes version**: Select 1.20.
   a. **Cluster Service Role:** Select the role specified for cluster management.
   b. **Tags:** Tags are optional, but you might add tags to identify the cluster.

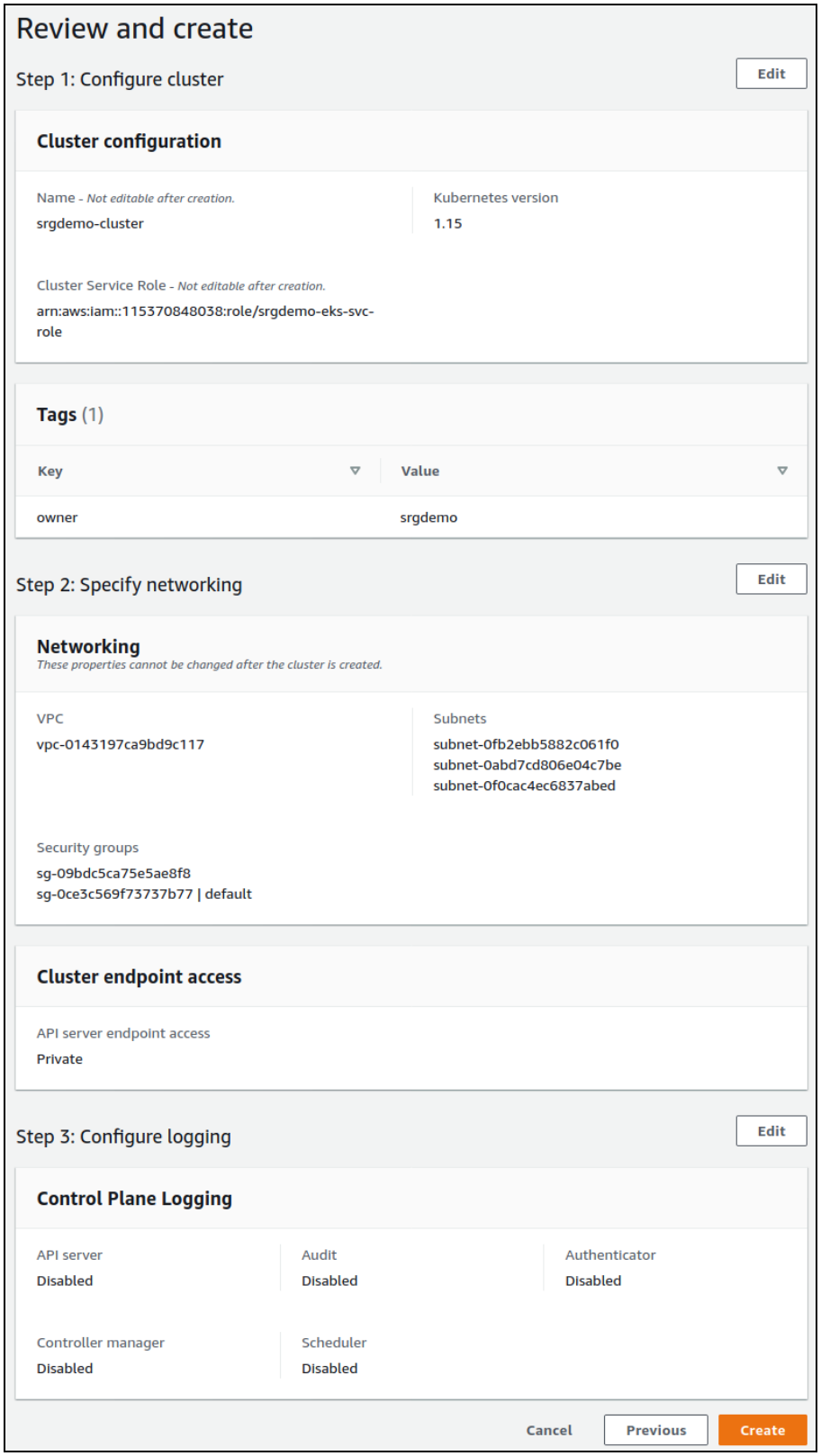


4. Click **Next.**
5. On the **Specify Networking** page, specify values for the following:
   a. **VPC:** Select your VPC from the dropdown.
   b. **Subnets:** Ensure **only** your private subnets are selected from the dropdown (subnet names are recorded in the AWS worksheet).
   c. **Security groups:** Add the Intra VPC security group named in the AWS worksheet.
   d. **Cluster endpoint access:** Select **Private** to keep the cluster isolated.

6. On the **Configure Logging** page, leave all values at default settings, then click **Next**.



7. On the **Review and Create** page, check all settings for accuracy and then click **Create**. The cluster details are displayed.


Review and create
Step 1: Configure cluster
Edit
Cluster configuration
Name - Not editable after creation.
srgdemo-cluster
Kubernetes version
1.15
Cluster Service Role - Not editable after creation.
arn:aws:iam::115370848038:role/srgdemo-eks-svc-role
Tags (1)
Key
Value
owner
srgdemo
Step 2: Specify networking
Edit
Networking
These properties cannot be changed after the cluster is created.
VPC
vpc-0143197ca9bd9c117
Subnets
subnet-0fb2ebb5882c061f0
subnet-0abd7cd806e04c7be
subnet-0f0cac4ec6837abed
Security groups
sg-09bdc5ca75e5ae8f8
sg-0ce3c569f73737b77 | default
Cluster endpoint access
API server endpoint access
Private
Step 3: Configure logging
Edit
Control Plane Logging
API server
Disabled
Audit
Disabled
Authenticator
Disabled
Controller manager
Disabled
Scheduler
Disabled
Cancel
Previous
Create

The creation process usually takes approximately 20 minutes and the status will change to *Active* when complete. Click **Refresh** to refresh the creation status display.

To configure EKS using the CLI:

1. Run the following command:
```
aws eks create-cluster \
--name <Cluster Name> \
--role-arn <EKS role ARN> \
--resources-vpc-config subnetIds=<private subnet
Ids>,endpointPublicAccess=false,endpointPrivateAccess=true,securityGroupId
s=<Intra VPC Security group Id> \
--kubernetes-version <Kubernetes version>
```

Where:

`<Cluster Name>`: The cluster name you have chosen during VPC creation; check the AWS worksheet for the value.

`<EKS role>`: The IAM role ARN created by your AWS infrastructure administrators; check the AWS worksheet for the value.

`<private subnet Ids>`: Comma-separated IDs of private subnets created together with the VPC; these values are recored in the AWS worksheet.

`<Intra VPC Security group Id>`: The ID of the previously created security group.

The value for `resources-vpc-config` cannot contain spaces; it must be one string.

`<Kubernetes version>`: Use the value from the AWS worksheet.

2. Record the ARN value in your AWS worksheet.

Example input and output:

```
aws eks create-cluster \
--name srgdemo-cluster \
--role-arn arn:aws:iam::115370811111:role/srgdemo-eks-svc-role \
--resources-vpc-config subnetIds=subnet-0fb2ebb5882c061f0,subnet-
0f0cac4ec6837abed,subnet-0abd7cd806e04c7be,\
endpointPublicAccess=false,endpointPrivateAccess=true,securityGroupIds=sg-
09bdc5ca75e5ae8f8 \
--kubernetes-version 1.20
```

```
{
   "cluster":{
      "name":"srgdemo-cluster",
      "arn":"arn:aws:eks:eu-central-1:115370811111:cluster/srgdemo-cluster",
```

```
      "createdAt":1589877429.005,
      "version":"1.15",
      "roleArn":"arn:aws:iam::115370811111:role/srgdemo-eks-svc-role",
      "resourcesVpcConfig":{
         "subnetIds":[
            "subnet-0fb2ebb5882c061f0",
            "subnet-0f0cac4ec6837abed",
            "subnet-0abd7cd806e04c7be"
         ],
         "securityGroupIds":[
            "sg-09bdc5ca75e5ae8f8"
         ],
         "vpcId":"vpc-0143197ca9bd9c117",
         "endpointPublicAccess":false,
         "endpointPrivateAccess":true,
         "publicAccessCidrs":[

         ]
      },
      "logging":{
         "clusterLogging":[
            {
               "types":[
                  "api",
                  "audit",
                  "authenticator",
                  "controllerManager",
                  "scheduler"
               ],
               "enabled":false
            }
         ]
      },
      "status":"CREATING",
      "certificateAuthority":{

      },
      "platformVersion":"eks.2",
      "tags":{

      }
   }
}
```


Cluster creation usually takes approximately 20 minutes. Check the cluster status by running the command:

```
aws eks describe-cluster \
```

```
--name <Cluster Name> \
| jq '.cluster.status'
```

The output immediately after creation should state `CREATING`. Repeat the command until the output changes to `ACTIVE`.

A newly created EKS might take up to 20 minutes to become `ACTIVE`.

Example:

```
aws eks describe-cluster \
--name srgdemo-cluster \
| jq '.cluster.status'
```

Next Steps: Configure `kubectl`

Configuring the Kubernetes Client (`kubectl`)

Several Kubernetes configuration and diagnostic tasks using `kubectl` will be performed on the bastion. In order to do that, the `kubectl` utility needs be configured with bastion credentials.

To configure `kubectl`:

1. Connect to the bastion instance and run the following command:

   ```
   aws eks update-kubeconfig --name <Cluster Name>
   ```

   The command will return an updated context <eks cluster arn> in `/home/centos/.kube/config`

   ```
   Output example:
   aws eks update-kubeconfig --name srgdemo-cluster
   ```

   Updated context: `arn:aws:eks:eu-central-1:115370811111:cluster/srgdemo-cluster` in `/home/centos/.kube/config`

2. On the bastion, check the Kubernetes service status by running:

   ```
   kubectl get svc
   ```

   ```
   Output example:

   NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
   kubernetes   ClusterIP   172.20.0.1   <none>        443/TCP   54m
   ```

   The EKS control plane is now ready and accessible from the bastion.

   Next Step: Applying the AWS Config Map

Applying the AWS ConfigMap to Enable Worker Nodes to Join the Cluster

You must apply the `AWS ConfigMap` so that the worker nodes can join your EKS cluster.

1. Connect to the bastion host.
2. Unpack the file `arcsight-platform-cloud-installer-XX.X.X.XXX.zip` located in the directory `/aws-scripts/objectdefs`.
3. From the unpacked file, open the file `cm-aws-auth.yaml` in any text editor.
4. Replace the placeholder `${WORKERS_ROLE_ARN}` with the `Role ARN` value from your AWS worksheet, and then save your changes. The `ConfigMap` will then resemble the following example:

```
apiVersion: v1
kind: ConfigMap
metadata:
name: aws-auth
namespace: kube-system
data:
mapRoles:
- rolearn: arn:aws:iam::115370811111:role/srgdemo-workernodes-svc-role
username: system:node:{{EC2PrivateDNSName}}
groups:
- system:bootstrappers
- system:nodes
```

5. On the bastion, run the following command:

```
kubectl apply -f cm-aws-auth.yaml
```

This command will output:

```
configmap/aws-auth created.
```

6. **Next Step:** Create and Configure Worker Nodes

Creating and Configuring Worker Nodes

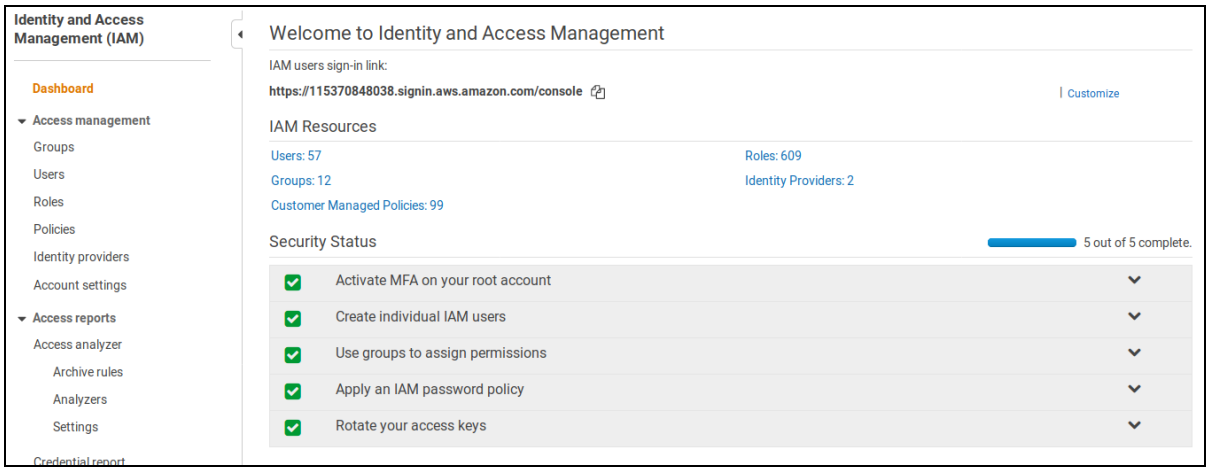
The *worker nodes* (EC2 nodes) are the Kubernetes nodes that will perform application processing. A cluster contains one or more Amazon EC2 nodes on which pods are scheduled. Amazon EKS nodes run in your AWS account and connect to your cluster's control plane through the cluster API server endpoint.

Next Step: Check for Worker Node Instance Profile

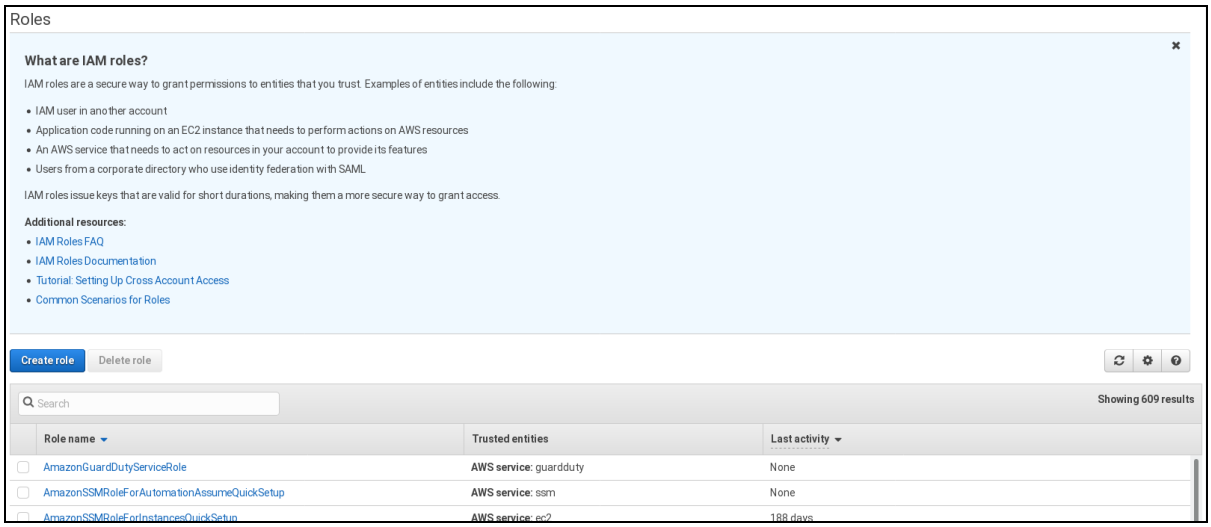
Checking for a Worker Node Instance Profile

To check for a worker node instance profile using the web UI:

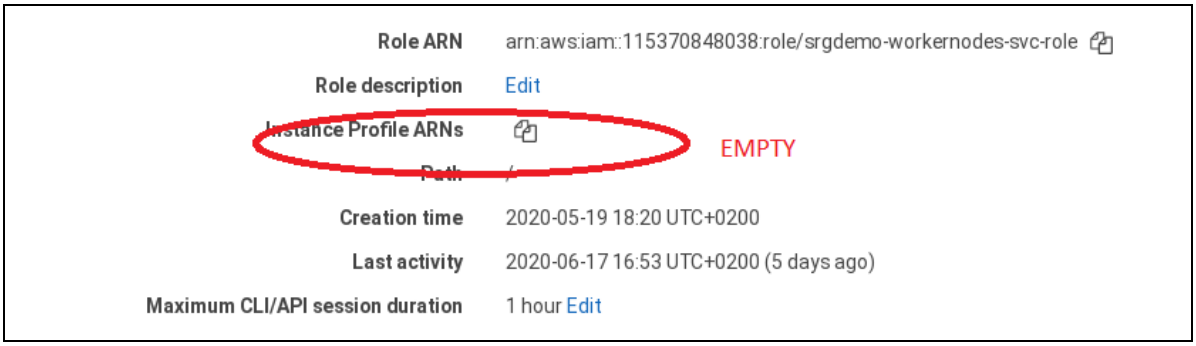
1. Using the **Find Services** search tool, browse to the IAM dashboard.
2. In the left navigation panel, under **Access management,** click **Roles** to get a list of existing roles.



3. In the search box, specify the Worker Nodes role name (from the AWS worksheet) to filter it from the other roles.



4. Click the role name to get its details, then check the row **Instance Profile ARNs.**



   a. (Conditional) If no instance profile has been assigned to the role (that is, the row Instance Profile is empty, as illustrated here), then continue with creating an instance profile.

   b. (Conditional) If the row **Instance Profile ARNs** is filled, then record the value in the AWS worksheet.

The Instance Profile creation guide works only on the command-line interface. It is not possible to create a separate instance profile without an assigned role.

To check for a worker node instance profile using the CLI:

1. Run the following command:

```
# aws iam list-instance-profiles-for-role \
--role-name <Workernodes role name from AWS worksheet>
```

```
Example:
# aws iam list-instance-profiles-for-role \
--role-name ARST-EKS-Workers-Custom-Role
```

The command will return only of the following outputs:

No instance profile for the role exists. Example output for this case:

```
{
   "InstanceProfiles":[

   ]
}
```

An instance profile exists for the role. Example output for this case:

```
{
   "InstanceProfiles":[
      {
         "Path":"/",
```

```
         "InstanceProfileName":"ARST-EKS-Workers-Custom-Role",
         "InstanceProfileId":"AIPARVXFDN4TBQBCRKX45",
         "Arn":"arn:aws:iam::115370811111:instance-profile/ARST-EKS-Workers-
Custom-Role",
         "CreateDate":"2020-06-16T05:57:59+00:00",
         "Roles":[
            {
               "Path":"/",
               "RoleName":"ARST-EKS-Workers-Custom-Role",
               "RoleId":"AROARVXFDN4TNRSAMVCVX",
               "Arn":"arn:aws:iam::115370811111:role/ARST-EKS-Workers-Custom-
Role",
               "CreateDate":"2020-06-16T05:57:58+00:00",
               "AssumeRolePolicyDocument":{
                  "Version":"2012-10-17",
                  "Statement":[
                     {
                        "Effect":"Allow",
                        "Principal":{
                           "Service":"ec2.amazonaws.com"
                        },
                        "Action":"sts:AssumeRole"
                     }
                  ]
               }
            }
         ]
      }
   ]
}
```

2. Do one of the following:
   - (Conditional) If no instance profile exists for the role, proceed with creating an instance profile, OR,
   - (Conditional) If the instance profile already exists for the role, record its name (`InstanceProfiles -> InstanceProfileName`) and ARN (`InstanceProfile -> Arn`) in the AWS worksheet, then continue with the procedure to create a launch configuration.

Next Step: Create an Instance Profile

Creating an Instance Profile

To create an instance profile:

1. Run the following command:
   ```
   aws iam create-instance-profile \
   --instance-profile-name <Workernodes Instance profile name>
   ```
2. Record your assigned worker node's instance profile name in the AWS worksheet. In our example we will use `srgdemo-workernodes-instance-profile`. The command will return a description of the newly-created instance profile. For example:

```
{
   "InstanceProfile":{
      "InstanceProfileId":"AIPAJMBYC7DLSPEXAMPLE",
      "Roles":[

      ],
      "CreateDate":"2015-03-09T20:33:19.626Z",
      "InstanceProfileName":"Webserver",
      "Path":"/",
      "Arn":"arn:aws:iam::123456789012:instance-profile/Webserver"
   }
}
```

3. Record the Arn value in the AWS Worksheet as `Workernodes Instance profile ARN.`
4. Run the following command to add the role to the instance profile:
   ```
   aws iam add-role-to-instance-profile \
   --instance-profile-name <Workernodes Instance profile name> \
   --role-name <Workernodes role name>
   ```

Where:

`<Workernodes Instance profile name>`: Use the instance profile name created above or by your AWS infrastructure administrators and recorded in the AWS worksheet. For example, `srgdemo-workernodes-instance-profile.`

`<Workernodes role name>`: Use the role name created by your AWS infrastructure administrators and recorded in the AWS worksheet. For example, `srgdemo-workernodes-svc-role.`

Example:
`aws iam add-role-to-instance-profile --instance-profile-name srgdemo-workernodes-instance-profile --role-name srgdemo-workernodes-svc-role`

The command has no output.

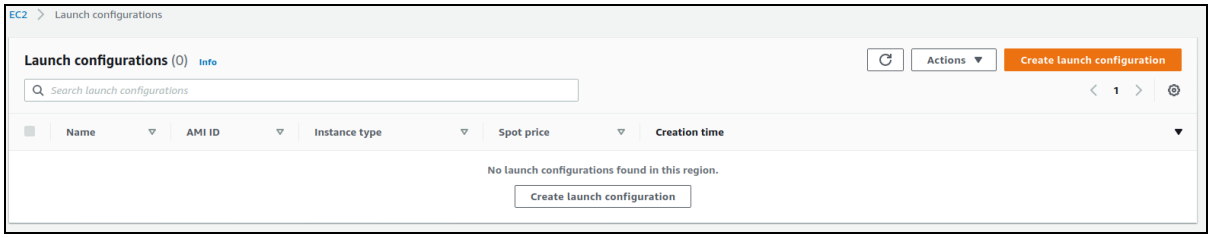
Next Step: Create and Configure a Launch Configuration

Creating and Configuring a Launch Configuration

A *launch configuration* is an instance configuration template that an Auto Scaling group uses to launch EC2 instances. Creating a launch configuration requires collecting some infrastructure data, specifically the Amazon Machine Image ID (AMI ID) and instance type. You can have more than one launch configuration created with different parameters, such as instance type or root volume size, and instantiate the auto-scaling groups from them.

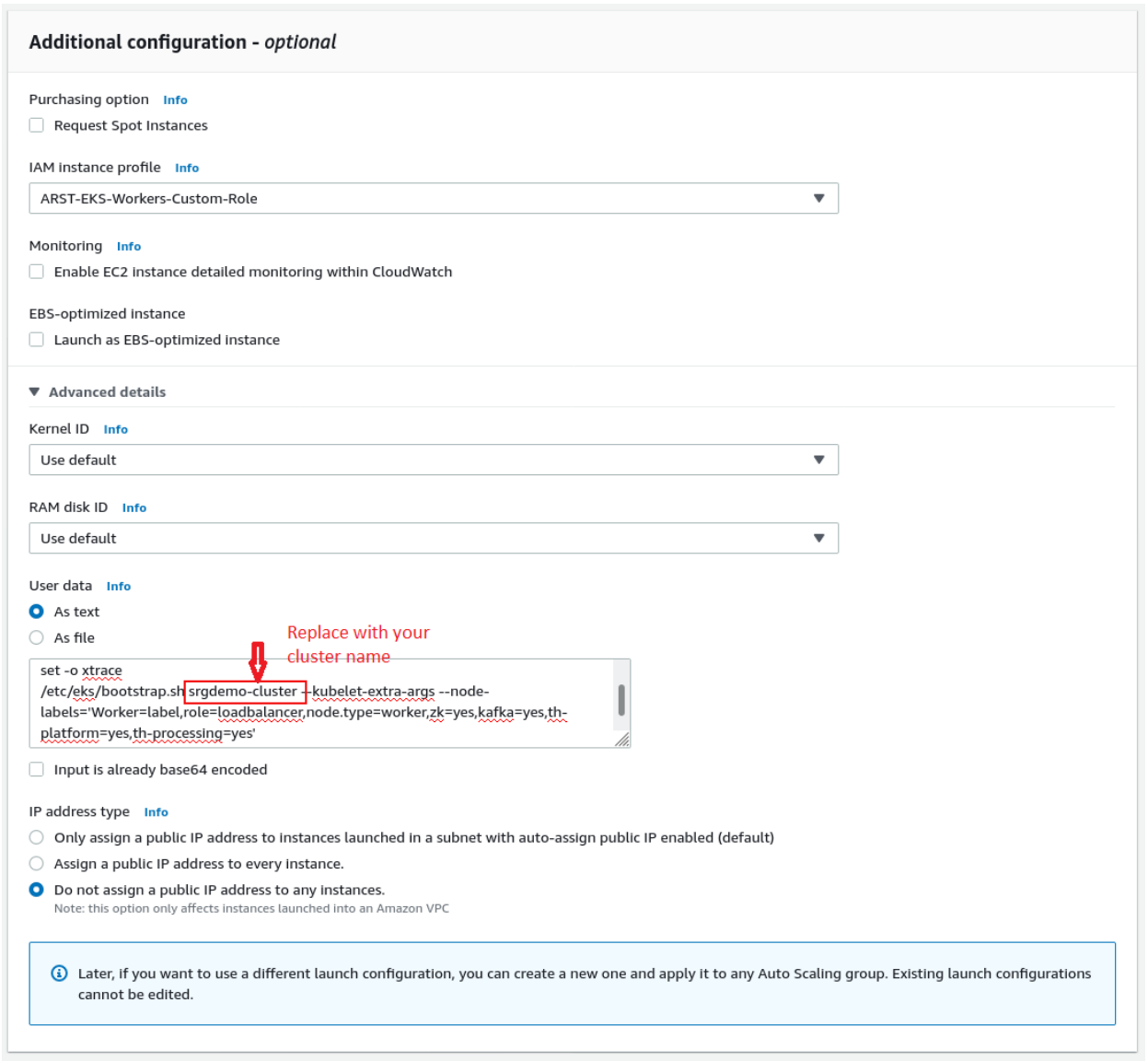
To create a launch configuration using the web UI:

1. Using the Find Services search tool, locate and browse to the EC2 Dashboard.
2. In the left navigation panel, under **Auto Scaling**, select **Launch Configurations.**
3. On the **Launch Configurations** page, click **Create Launch Configuration.**

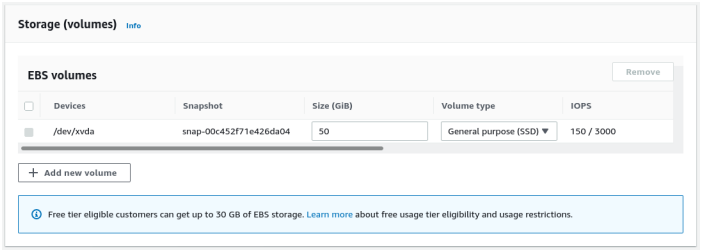


4. On the **Create Launch Configuration** page, specify values for the following:
   - **Launch configuration name:** Specify your launch configuration name.
   - **Virtual machines HW and OS:** Specify the virtual machine hardware and installed operating system.
5. Under **Additional configuration details - *optional***, specify values for the following (mandatory) settings:
   a. **IAM instance profile:** Choose the instance profile name for the worker nodes from the AWS worksheet.
   b. **User data:** Leave **As text** selected, then copy the contents of the script `workernodes-userdata` located in the `aws-scripts` folder. In the text area, replace the `<cluster name>` with your own cluster name from the AWS worksheet.
   c. **IP address type:** select **Do not assign a public IP address to any instances.**

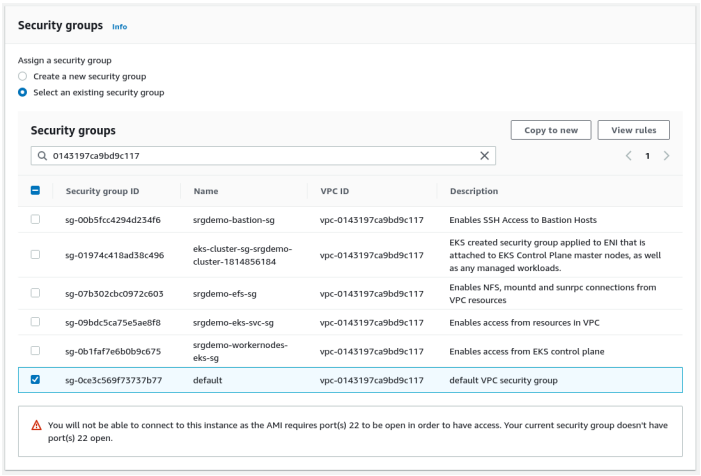
Also, notice the labels: `zk:yes, kafka:yes, th-platform:yes`, and `th-processing:yes`. Each created automatically receives the respective labels required for running Transformation Hub. For more information about labels, see Labeling Nodes.



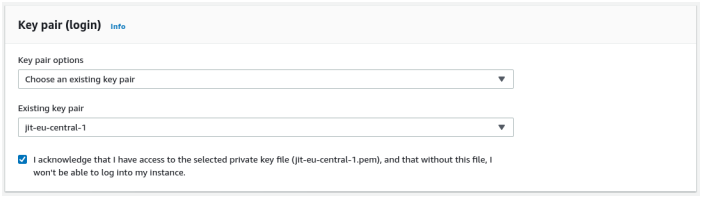
6. Under **Storage**, configure the storage for each node. Specify the following settings:
   a. **Size (GiB):** Change the size to minimum 50 GiB or more depending on your plans for product installation and load.
   b. Scroll to the right and select **Delete on termination.**



7. Each resource must be accessible through the network and all worker nodes must have the correct security groups assigned to them.
   a. Under **Security groups**, choose **Select an existing security group.**
   b. Choose the security group created for intra-VPC communications, recorded in the AWS worksheet.



8. Create a key pair to allow cluster access as well as SSH access to the worker nodes.
9. Under **Key pair (login),** choose values for the following:
   a. **Key pair options:** Select **Choose an existing key pair.**
   b. **Existing key pair:** Choose your key pair for which you own the private part.
   c. **I acknowledge... :** Select in order to proceed with launch configuration creation.



10. Click **Create launch configuration** to create the new launch configuration.

To create a launch configuration using the CLI:

1. Run the following command:

```
aws autoscaling create-launch-configuration \
 --launch-configuration-name <Launch Configuration name> \
 --image-id <Launch config AMI Id> \
 --key-name <Key pair name> \
 --security-groups <Intra VPC Security group Id> \
 --instance-type <Instance type> \
 --block-device-mappings "<block device mapping>" \
 --iam-instance-profile <Workernodes Instance profile ARN> \
 --no-associate-public-ip-address \
```

```
--user-data file://<user data filename>
```

Where:

`<Launch Configuration name>`: Choose a name that helps with easier identification. In our examples we will use `srgdemo-workers-launch-config`. Record the chosen value in your AWS worksheet.

`<Launch config AMI ID>`: Run the following command to get the actual AMI ID.

```
aws ec2 describe-images \--filters "Name=architecture,Values=x86_64"
"Name=name,Values=amazon-eks-node-<Kubernetes version>*" | jq '.Images | map
(select((.Description!=null) and (.Description | contains("GPU") | not) and
(.ImageLocation | contains("gpu") | not))) | sort_by(.CreationDate) | [last]'
```

Important! Replace the value `<Kubernetes version>` with the Kubernetes version number you used to create the EKS, and recorded as the `Kubernetes version` in your AWS worksheet. **Retain the asterisk** at the end of the value for the name filter.

`<Key pair name>`: Use the key pair name from the AWS worksheet.

`<Intra VPC Security group Id>`: ID of Intra VPC security group recorded in the AWS worksheet.

`<Instance type>`: From Amazon EC2 Instance Types, choose your machine hardware configuration. Consider CPU, RAM, storage type, network performance, and price.
DO NOT USE the same type you used for the bastion.

Instance type resources must beselected based on the Arcsight technical requirements.

`<block device mapping>`: Pass the value `DeviceName=<root device name>,Ebs={VolumeSize=<root volume size>,VolumeType=gp2,DeleteOnTermination=true}`
where:

- `<root device name>`: In the description obtained for AMI ID above, locate the value for key `RootDeviceName.`
- `<root volume size>`: Size depends on planned installation size: 50GB for SMALL, 100GB for MEDIUM, and 256GB for LARGE. Sizes refer to the sizes from ArcSight Suite metadata definition, in gigabytes.

`<instance profile ARN>`: Use the instance profile ARN you created above or created by your AWS infrastructure administrators and recorded in your AWS worksheet.

`--user-data file://<user data filename>`: This file needs to be modified before you execute the command. Copy the file `workernodes-userdata` located in `aws-byok-installer-<version>/scripts/` to the current working directory on the local host or on the bastion, then edit. Replace the parameter cluster name with the value from the AWS worksheet.

Example of workernodes-userdata:

```
#!/bin/bash
```

```
set -o xtrace
```

```
/etc/eks/bootstrap.sh srgdemo-cluster --kubelet-extra-args \
--node-
labels='Worker=label,role=loadbalancer,node.type=workerlancer,node.type=worke
r'
```

Extending labels, as shown here, during the launch configuration creation will be important for scaling up the cluster later. However, if you do not assign labels automatically, you must manually add new labels every time you add a new node.

You can also extend the labels assigned to the worker nodes by adding the Transformation Hub required labels. Use with care. Keep in mind that all worker nodes created from this launch configuration will automatically receive that set of labels. In some cases, this might be unwanted behavior.

Example:

```
aws autoscaling create-launch-configuration \
--launch-configuration-name srgdemo-workers-launch-config \
--image-id ami-025291add34df213c \
--key-name srgdemo \
--security-groups sg-0ce3c569f73737b77 \
--instance-type m4.2xlarge \
--block-device-mappings "DeviceName=/dev/xvda,Ebs=
{VolumeSize=50,VolumeType=gp2,DeleteOnTermination=true}" \
--iam-instance-profile arn:aws:iam::115370811111:instance-profile/srgdemo-
workernodes-instance-profile \
--no-associate-public-ip-address \
--user-data file://workernodes-userdata
```

Next Step: Create the Auto Scaling Group

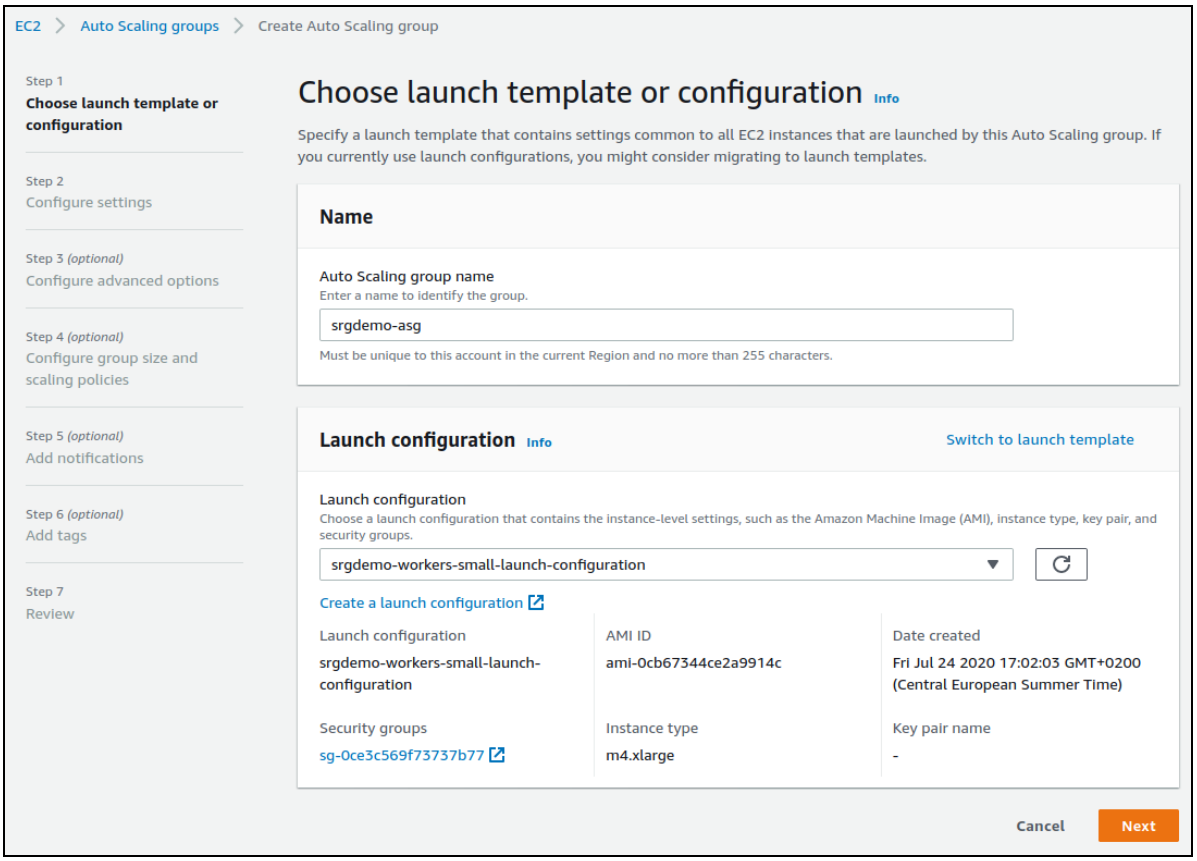
Creating the AWS Auto Scaling Group

AWS Auto Scaling enables you to build scaling plans that automate how groups of different resources respond to changes in demand. You can optimize availability, costs, or a balance of both.

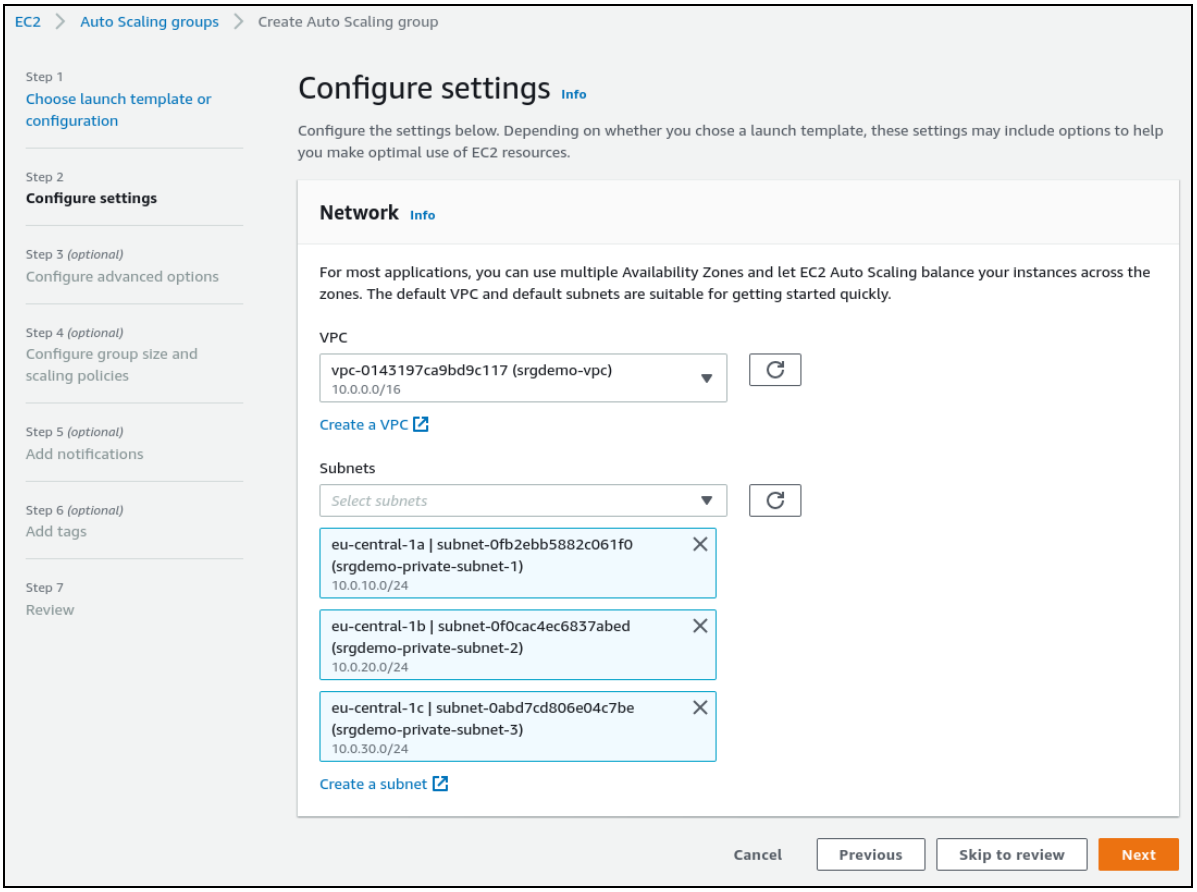
Before proceeding, verify that the correct tag has been assigned to the VPC.

To create the Auto Scaling group using the web UI:

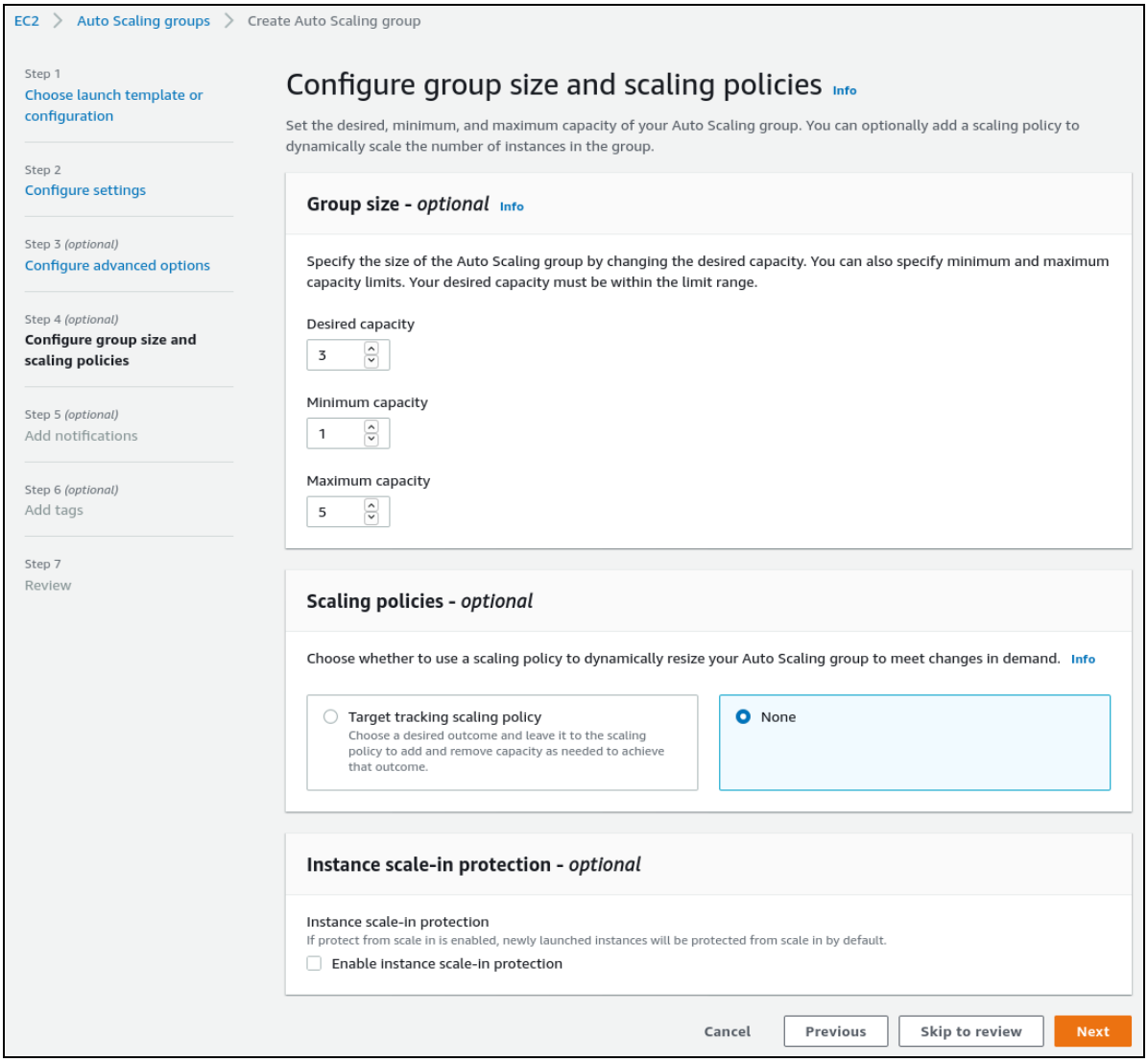
1. Using the Find Services search tool, locate and browse to the EC2 dashboard.
2. In the left navigation panel, under **Auto Scaling**, select **Auto Scaling Groups.**
3. On the landing page, click **Create Auto Scaling Group** to launch the creation wizard. If no Auto Scaling groups have been created yet, the introduction to Auto Scaling implementation in AWS is displayed.
4. On the **Choose launch template or configuration** page, specify values for the following:
   - **Name:** Specify a descriptive name for the Auto Scaling group.
   - **Launch configuration:** Click **Switch to Launch template**, then select the launch configuration you created previously.



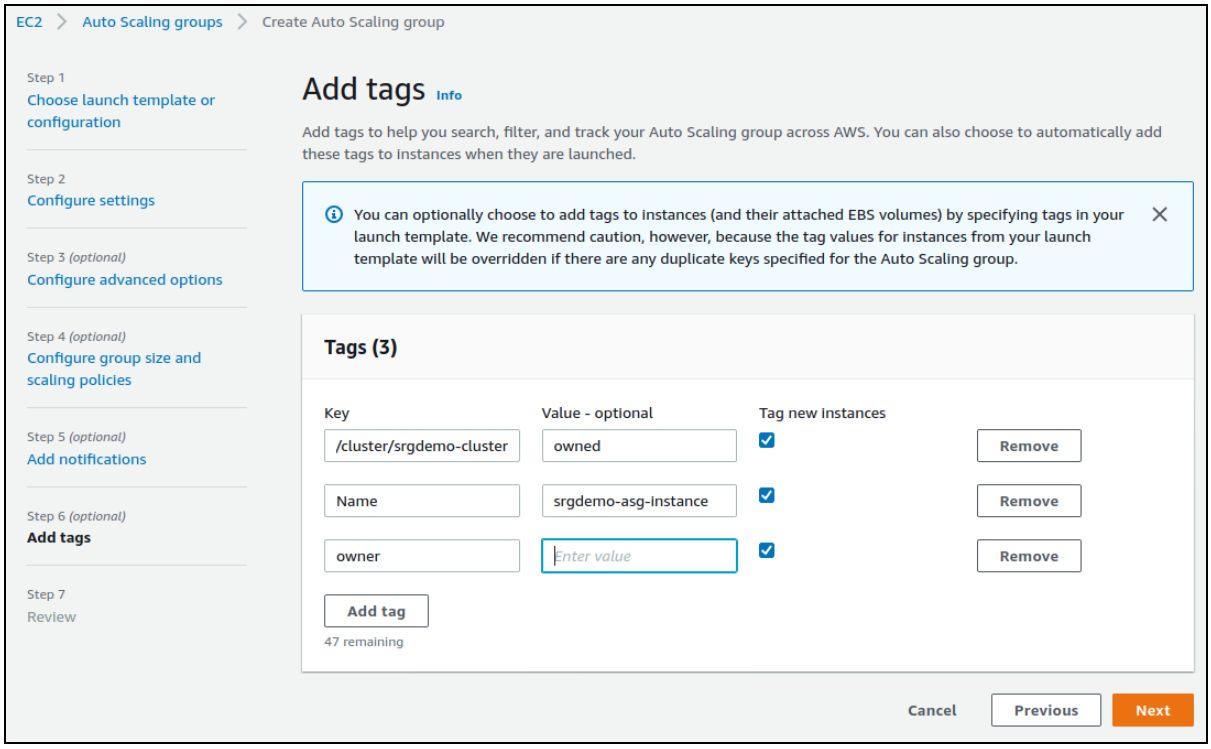
5. Click **Next**.
6. On the **Configure settings** page, specify values for the following:
   a. **VPC:** Select the previously-created VPC that you recorded in the AWS worksheet.
   b. **Subnets:** Select all 3 **private** subnets (from the AWS worksheet).



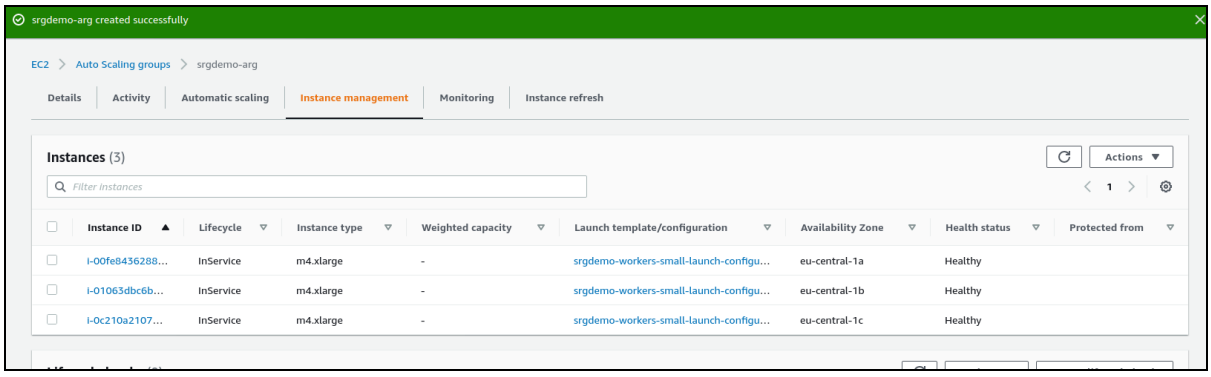
7. Click **Next**.
8. On the **Configure advanced options** page, leave all values set to their defaults and click **Next**.
9. On the **Configure group size and scaling policies** page, set values for **Desired capacity** and **Maximum capacity**. In the example shown here, there will initially be three nodes and enough space to instantiate two more by simply increasing the desired capacity. (For a production cluster, the minimum capacity setting should not be less than two.)



10. Click **Next**.
11. On the **Add Notifications** page, ignore all settings and click **Next**.
12. On the **Add Tags** page, add tags as follows:
    a. Add the mandatory tag key: `kubernetes.io/cluster/<your cluster name>` with value: `owned` (replace `<your cluster name>` with your actual cluster name).
    b. (Optionally) Add these tags as desired:
        i. Key: `Name` with value: derived from the auto-scaling group name.
        ii. Key: `owner` with value: specify your own name as a tag.
    c. For all new tags you add, select **Tag new instances** so new instances are automatically tagged.
13. Click **Next**.



14. On the **Review** page, verify your settings, then click **Create an Auto Scaling Group.**
15. For networking configurations, we will need to know the machine instance IDs in the new Auto Scaling group. On the **Auto Scaling Groups** management page, select your new group.



16. From the **Instances** tab, record all instance IDs in the AWS worksheet.

To create the Auto Scaling group using the CLI:

1. Run the following command:

```
aws autoscaling create-auto-scaling-group \
--auto-scaling-group-name <Autoscaling group name> \
--launch-configuration-name <Launch Configuration name> \
--min-size <min size> \
--desired-capacity <desired size> \
```

```
    --max-size <max size> \
    --tags "Key=kubernetes.io/cluster/<cluster name>,Value=owned"
    "Key=Name,Value=<auto scaling group name>" \
    --vpc-zone-identifier "<subnet Ids>"
```

Where:

`<Autoscaling group name>`: Specify a name which helps with easier group identification. In this guide we will use srgdemo-autoscaling-group. Record the value in the AWS worksheet.

`<Launch Configuration name>`: Name of the launch configuration created above.

`<min size>`: The minimum size of the group (for a production cluster, this should not be less than two).

`<desired size>`: The number of Amazon EC2 instances that the Auto Scaling group attempts to maintain. This number must be greater than or equal to the minimum size of the group and less than or equal to the maximum size of the group. If you do not specify a desired capacity, the default is the minimum size of the group.

`<max size>`: The maximum size of the group.

`<Cluster Name>`: Use the cluster name from the AWS worksheet.

`<subnet Ids>`: A comma-separated list of private subnet IDs for your virtual private cloud (VPC). Use values from the AWS worksheet.

Example:

```
aws autoscaling create-auto-scaling-group \
--auto-scaling-group-name srgdemo-autoscaling-group \
--launch-configuration-name srgdemo-workers-launch-config \
--min-size 1 \
--desired-capacity 3 \
--max-size 3 \
--tags "Key=kubernetes.io/cluster/srgdemo-cluster,Value=owned"
"Key=Name,Value=srgdemo-autoscaling-group" \
--vpc-zone-identifier "subnet-0fb2ebb5882c061f0,subnet-
0f0cac4ec6837abed,subnet-0abd7cd806e04c7be"
```

It can take approximately five minutes for nodes to be created and join the cluster.

Retrieve the instance IDs from auto-scaling group

For networking configurations, we will need to know the machine instances in the new Auto Scaling group.

To retrieve the instance IDs:

1. Run the command:
   ```
   aws autoscaling describe-auto-scaling-instances\
   | jq -r '.AutoScalingInstances[] | select(.AutoScalingGroupName=="<Your new Auto Scaling group name>").InstanceId'
   ```
2. Record the returned IDs in the AWS worksheet.

Example command and output:

```
aws autoscaling describe-auto-scaling-instances \
| jq -r '.AutoScalingInstances[] | select(.AutoScalingGroupName=="srgdemo-autoscaling-group").InstanceId'
```

```
i-05662f9ef84c182ca
```

```
i-07cfcd6716e9890b5
```

```
i-08d819b5ccabe83cb
```

After approximately five minutes, the nodes will be created and be joined to the cluster. You can then list all the worker nodes.

To list the worker nodes:

1. On the bastion host, run the following command:
   ```
   kubectl get nodes
   ```

At this point, you should see all nodes listed in the Ready state with the expected Kubernetes version.

Next Step: Labeling Worker Nodes

Labeling Cloud (AWS) Worker Nodes

Labeling is a means for identifying application processing and qualifying the application as a candidate to run on a specific node. For example, labeling a node with the label `kafka=yes` specifies that a Kafka instance runs on that node.

- "Product Labels" below
- "Labeling Worker Nodes" on the next page
- For more information about labeling, see "Understanding Labels and Pods" on page 578

Product Labels

The following table shows the labels and the associated Arcsight products.

Label	The node runs...	Products
`kafka=yes`	Kafka	Transformation Hub
`zk=yes`	ZooKeeper	Transformation Hub
`fusion=yes`	Fusion	ArcSight ESM Command Center ArcSight Layered Analytics ArcSight Recon Fusion Intelligence
`th-processing=yes`	Transformation Hub data	Transformation Hub
`th-platform=yes`	Transformation Hub	Transformation Hub
`intelligence=yes`	Pods that manage functions and services for the ArcSight Intelligence capability	Intelligence
`intelligence-spark=yes`	Analytics services for the ArcSight Intelligence capability	Intelligence
`intelligence-datanode=yes`	Pods that manage HDFS services for the ArcSight Intelligence capability	Intelligence
`intelligence-namenode=yes`	HDFS NameNode services for the ArcSight Intelligence capability. Place this label on one node only. The node and the hostname or IP address in the **HDFS NameNode** field in the **Intelligence** tab of the **CDF Management Portal** must match.	Intelligence

Labeling Worker Nodes

To label AWS worker nodes:

You can skip this step if you have added all the required labels to your launch configuration from which you have deployed your nodes.

1. Connect to the bastion.
2. Retrieve the list of nodes by running the command:

```
kubectl get nodes -o name | cut -d '/' -f 2
```

3. Run the following command once for each node:

```
kubectl label\
--overwrite=true node <node name> zk=yes kafka=yes th-platform=yes th-
processing=yes fusion=yes
```

For example:

```
kubectl label \
--overwrite=true node ip-10-0-10-83.eu-central-1.compute.internal \
zk=yes kafka=yes th-platform=yes th-processing=yes
```

Example output:

```
node/ip-10-0-10-83.eu-central-1.compute.internal labeled
```

Add additional labels as needed for capabilities that you plan to deploy.

1. Verify your labels by running the command:

```
kubectl get nodes --show-labels
```

Next Step: Upload Product Images to the ECR

Uploading Product Images to the ECR

The Amazon Elastic Container Registry (ECR) is an AWS managed Docker container registry. CDF and Kubernetes will search for product images to download from the ECR and instantiate them.

The ECR is accessible from the internet and protected by username and password credentials. You can perform tasks in this section from a local host or from the bastion, as long as the AWS CLI has been configured.

Uploading images requires the script `upload_images_to_ECR` to be installed and located in the `aws-scripts/scripts/` directory. This script parses the `manifest.json` description file and creates the ECR and its repositories. Then the CDF script `uploadimages.sh` is called, which passes the correct parameters and uploads the product images.

Uploading Image Requirements

In order to be able to upload images to the ECR, the following requirements must be met:

- You must be able to execute a bash script.
- The system used must have the following basic Linux/Unix utilities installed:
  - `cat`
  - `find`
  - `awk`
  - `jq`
  - `pwd`

  - unzip
  - tar
- You must have aws cli configured on your system.
- You must have fulfilled all requirements for the CDF uploadimages.sh script.

To create the ECR and then upload the product images to it:

1. Verify that you have downloaded the product image files for the capabilities you wish to install.
2. Run the following command:
```
# <path to upload script>/upload_images_to_ECR \
-d <images' folder> \
-F <product package> \
-o <organization> \
-y \
[-c <parallel uploads count>]\
[-uip <uploadimages.sh path>]
```

Where:

<path to upload script>: It is possible to execute the upload script from any folder. The recommendation is to have the current folder set to the one with downloaded images; then the path would resemble aws-scripts/.

<images' folder>: Folder where all images in their subfolders are located; usually it is the folder where you have unpacked downloaded packages. Can be specified multiple times for situations where images are located in various folders.

<product package>: Path to the package file. For example, ./transformationhub-<version>.tar. Can be specified multiple times.

<organization>: Specifies the organization name (namespace) where the suite images are placed in the ECR. Record the chosen organization name in the AWS worksheet. There might be multiple repositories in the ECR which might be shared or overlap. Pay special attention to specify the correct organization name. The organization name must be valid ASCII, and can be from 2 to 255 characters. It can only contain lowercase letters, numbers, dashes (-), and underscores (_).

<parallel upload counts>: Maximum allowed parallel uploads; this is limited based on the CPU cores. The parameter is optional. If not specified, defaults to 8.

<uploadimages.sh path>: Path to the original CDF uploadimages.sh script. Parameter is optional. When not specified, the upload_images_to_ECR script will try to locate it in the images' folder or in the unpacked cdf-deployer package, which is part of of the arcsight-platform-cloud-installer package. Note that normally you should not unpack this package.

You must specify at least one image location, either in form of a folder (`-d` option) or as a file path (`-F` option, recommended). If you use the `-d` option, you must unpack the image package before running the script.

Example:

```
./arcsight-platform-cloud-installer-22.1.00153-22.1.599/aws-scripts/upload_
images_to_ECR \
-F ./transformationhub-3.6.0.1284-master.tar -o srgdemo \
-y \
-c 8
```

3. Run the following command:

```
<path to upload script>/upload_images_to_ECR \
 -F cdf-byok-images.tar \
-o <organization>\
-y \
-c <parallel uploads count>
```

Ensure that you give the upload process sufficient time to complete. You can check the returned messages or check the log file in the directory where you are executing the upload script to determine successful upload. While the upload progresses, the repositories are created in the ECR, followed by image uploads to the repositories.

Multiple suite images can be uploaded as a single command as long as each image package is prefaced with `-F`.

Next Step: Configuring Route 53 Routing

Configuring Route 53

Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service. This section describes how to create a publicly available CDF installation, and the CDF management portal, as well as how to reconfigure a suite. Your own business requirements might dictate a different secure configuration.

Next Step: Select a Public Hosted Zone and Create a Record Set

Selecting a Public Hosted Zone and Creating a Record Set

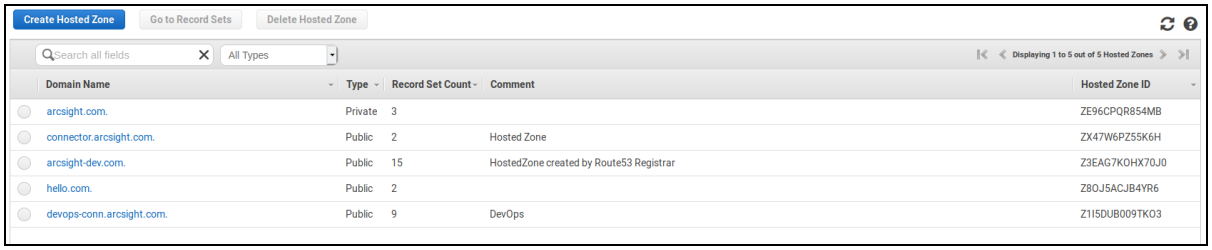
In the Route 53 service, DNS records are organized in *hosted zones.* A hosted zone is analogous to a traditional DNS zone file; it represents a collection of records that can be managed together, belonging to a single parent domain name. All resource record sets within a hosted zone must have the hosted zone's domain name as a suffix.

In this section you will select a **public** hosted zone, which must be previously created by an AWS administrator, and create the record set.

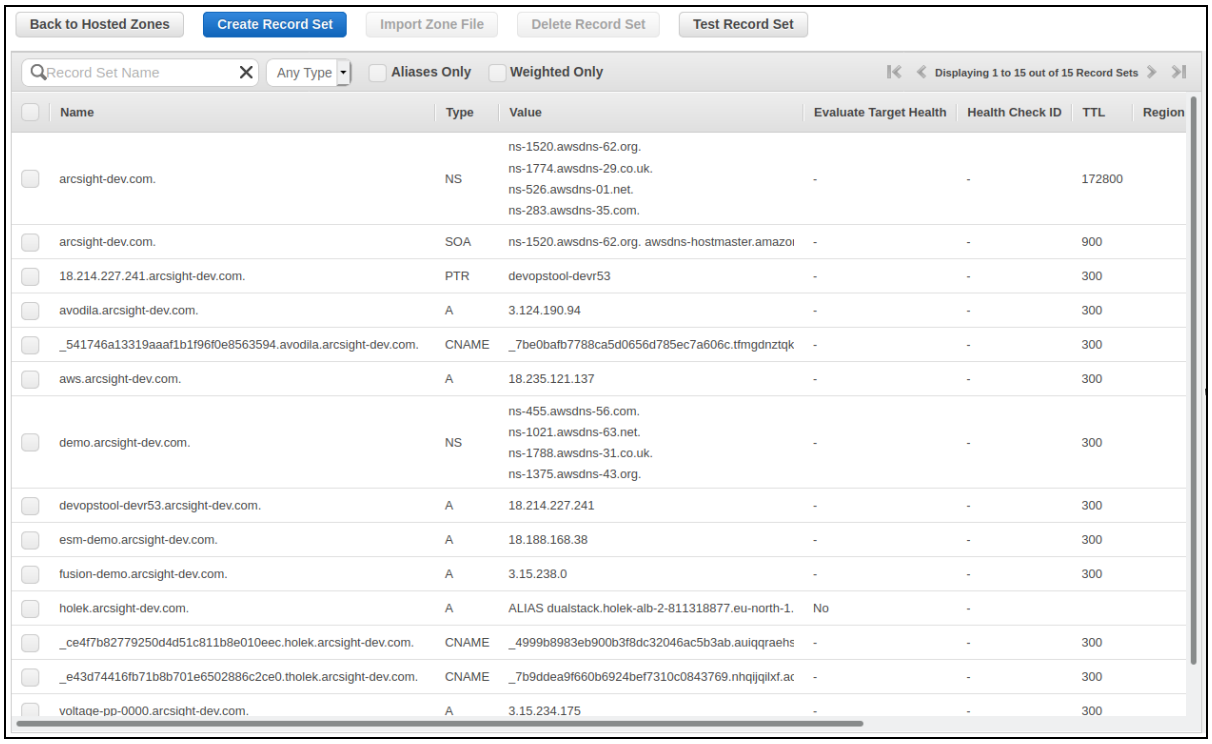
Do not use a private hosted zone.

To select an existing public (not private) hosted zone and create the record set using the web UI:

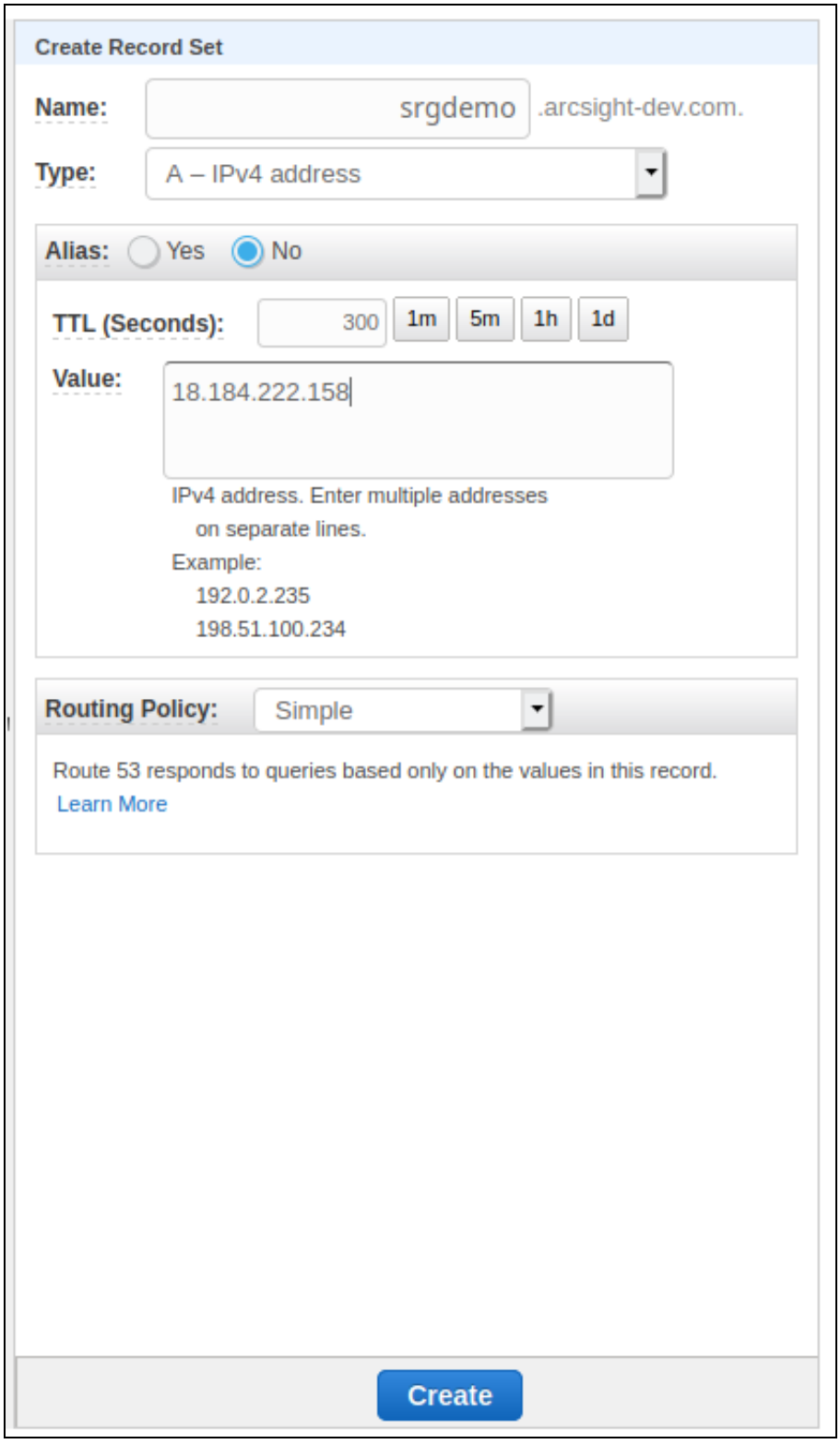
1. Using the Find Services search tool, locate and browse to the Route 53 dashboard.
2. In the left navigation panel, select **Hosted Zones**. Ignore any error message about insufficient permission.
3. From the list of hosted zones, select a **public** zone. For our example, we use `arcsight-dev.com`.



4. Click the public hosted zone domain name to list the record sets in the public zone.



5. Click **Create Record Set** and specify or verify values for the following parameters:
   - **Name:** Choose a name for the A-record. The record set FQDN will then be composed from this name and the public hosted zone domain name. In our example we will use `srgdemo`. Our example installation will then be available at the URL: `srgdemo.arcsight-dev.com`.
     Record the record set FQDN in the AWS worksheet.
   - **Type:** *A - IPv4 address.*
   - **Alias:** *No*
   - **TTL:** Leave default.
   - **Value:** Specify your bastion's public IP address.
   - **Routing Policy:** *Simple*



6. Click **Create.** The new record set is displayed in the list.

To select an existing public (not private) hosted zone and create a record set using the CLI:

1. Run the following command to select **public** hosted zones:
   ```
   aws route53 list-hosted-zones \
   | jq -r '.HostedZones[] | select(.Config.PrivateZone==false) | "Id: " +
   .Id,"Name: " + .Name," " '
   ```

A list of hosted **public** zones is returned. For example:

```
Id:   /hostedzone/ZX47W6PZ55K6H
Name: connector.arcsight.com.
```

```
Id:   /hostedzone/Z3EAG7KOHX70J0
Name: arcsight-dev.com.
```

```
Id:   /hostedzone/Z8OJ5ACJB4YR6
Name: hello.com.
```

```
Id:   /hostedzone/Z1I5DUB009TKO3
Name: devops-conn.arcsight.com.
```

The dots and period characters (.) at the end of each value are present intentionally. **Please do not remove them.**

2. Choose one of the **public** hosted zones. For example, we will use the **public** hosted zone (name shown includes a period):
   `arcsight-dev.com.`
   Record the chosen **public** hosted zone name and ID in the AWS worksheet under `Hosted zone name` and `Hosted zone Id` respectively.
3. Choose a subdomain in the selected public hosted zone. For example, we will use `srgdemo`. Combining the subdomain and hosted zone name with a final period will give us the complete DNS name where our new cluster will be accessible.
   Example:
   `srgdemo.arcsight-dev.com.`
4. From the directory `arcsight-platform-cloud-installer-<version>/aws-scripts/objectdefs/`, copy the supplied template `CreateRecordSetInHostedZone.json` to the working folder.
5. Open the template in a text editor and set values for the following placeholders:
   a. `<Record name>:` Combine the name of the hosted zone (for example, `srgdemo`) and Hosted zone name (for example, `arcsight-dev.com`) to create the DNS name and then append the dot character (.) `Example: srgdemo.arcsight-dev.com.`

b. `<Record type>`: Replace with a value of `A`.

c. `<Record value>`: Use your bastion IP address.

> The placeholders in the template use syntax `<placeholder name>`, for example, `<Record name>`.

The following example shows a modified JSON template. Notice that the trailing period in the record name is mandatory.

```
{
   "Changes":[
      {
         "Action":"UPSERT",
         "ResourceRecordSet":{
            "Name":"srgdemo.arcsight-dev.com.",
            "Type":"A",
            "TTL":300,
            "ResourceRecords":[
               {
                  "Value":"3.120.237.11"
               }
            ]
         }
      }
   ]
}
```

6. Run the following command:

```
aws route53 change-resource-record-sets \
--hosted-zone-id <Hosted zone Id> \
--change-batch file://CreateRecordSetInHostedZone.json
```

Where:

`<Hosted zone Id>`: Use the hosted zone ID retrieved above. For example: `/hostedzone/Z3EAG7KOHX70J0`

`--change-batch`: Replace the parameter here with your own modified instance of the JSON file `CreateRecordSetInHostedZone.json`.

The command returns a change request. For example:

```
{
   "ChangeInfo":{
      "Id":"/change/C04669622EJ7JNXG69KJO",
      "Status":"PENDING",
      "SubmittedAt":"2020-06-09T09:35:06.376000+00:00"
```

```
   }
}
```

Later, the status will change to `INSYNC. To validate the status, run this command:`

```
aws route53 get-change --id <change id>
```

Output example:

```
{
   "ChangeInfo":{
      "Id":"/change/C04669622EJ7JNXG69KJO",
      "Status":"INSYNC",
      "SubmittedAt":"2020-06-09T09:35:06.376000+00:00"
   }
}
```

Next Step: "Bootstrapping CDF" below

Bootstrapping CDF

Bootstrapping CDF is a method of installing a few basic pods onto the Kubernetes cluster created previously (when you configured EKS and worker nodes).

During this process, the CDF bootstrap script does the following:

- Downloads Docker images from the ECR (Elastic Container Registry).
- Instantiates pods for various checks like the EFS space and the structure created on it.
- Creates `nginx` pods for use as a load balancer, and for allowing connections to the web installation process.

After the CDF bootstrap process completes, you will need to import the intermediate certificate to the CDF, configure some required networking settings, then continue installation using the CDF web installation interface.

Note: If you used a non-root user to install Kubernetes, you must use the same non-root user to install CDF as well. In addition, the non-root user installation process will prompt for additional steps.

- Preparing the CDF Deployer
- Retrieving the ECR Credentials
- Bootstrapping CDF
- Bootstrapping CDF

Preparing the CDF Deployer

The EKS and worker nodes you have configured are completely isolated from access from the internet, each of the nodes can access it if needed. As a result, the process of bootstrapping CDF must be performed from the bastion.

You have already copied the package `arcsight-platform-cloud-installer-<version>.zip` to the bastion and unpacked it during configuration of EFS. As a part of this package, the `cdf-deployer.zip` is included.

To prepare the CDF deployer:

Unpack the `cdf-deployer.zip` archive by running the following command:

```
unzip ./arcsight-platform-cloud-installer-<version>/cdf-deployer.zip
```

This will create the directory `arcsight-platform-cloud-installer-<version>/cdf-deployer`.

Retrieving the ECR Credentials

CDF needs the credentials to access the ECR in order to be able to download images.

To retrieve the ECR credentials:

1. On the bastion, run the command:

   ```
    ./arcsight-platform-cloud-installer-<version>/aws-scripts/scripts/upload_images_to_ECR --get-ecr-credentials
   ```

2. The file `ecr_credentials` is created in the directory where the script was run, containing username, password, and ECR URL.
3. Run the following command:

   ```
   source ecr_credentials
   ```

The password retrieved here is valid for only 12 hours after creation.

To bootstrap CDF:

1. Change the working folder to `cdf-deployer` and run the following command:

```
./install \
--registry-url $ECR_URL \
--registry-username $ECR_USER_NAME \
--registry-password $ECR_USER_PASSWORD \
-P <suite admin password> \
--registry-orgname <orgname> \
--nfs-server <Filesystem FQDN> \
--nfs-folder <CDF ITOM volume> \
--cloud-provider aws --external-access-host <RecordSet name>
```

Where:

Variables **`$ECR_URL`**, **`$ECR_USER_NAME`**, and **`$ECR_USER_PASSWORD`** come from the `ecr_credentials` file which you sourced previously.

`<suite admin password>`: Choose a password between 8 to 20 characters in length. A password must include numbers, lowercase chars, uppercase chars and special characters. Exclude whitespace characters, such as space, newline, and so on.

`<orgname>` : Use the same value as for upload images; check the AWS worksheet for this value.

`<Filesystem FQDN>`: Use the value from the AWS worksheet.

`<CDF ITOM volume>`: The directory on NFS/EFS into which CDF starts installation. The path is a combination of the parent directory as specified in Configure EFS for ArcSight Suite and the predefined subfolder name. For example, `/srgdemo/itom-vol`.

`<RecordSet name>` : The A-record (FQDN) used for connecting to the CDF installation and management portal. Use the value from the AWS worksheet.

Note: Ensure that you remove the trailing period from the FQDN.

Example:

```
./install --registry-url $ECR_URL \
--registry-username $ECR_USER_NAME \
--registry-password $ECR_USER_PASSWORD \
-P "Password@123" \
--registry-orgname srgdemo \
--nfs-server fs-ebe456b3.efs.eu-central-1.amazonaws.com \
--nfs-folder /srgdemo/itom-vol \
--cloud-provider aws \
--external-access-host srgdemo.arcsight-dev.com
```

After the CDF bootstrap completes, you are prompted to log in at the following URL:
`https://<external access host>:3000`

However, you will not be able to log in successfully yet, as there are some network infrastructure resources still to prepare, as explained in the succeeding steps.

Next Step: Securing External Communication with the RE Certificate

Securing External Communication with the RE Certificate

At the center of the Platform is a Kubernetes cluster where communication occurs between pods within the cluster and with non-containerized ArcSight components outside of the cluster. In order to ensure secure trusted communication between pods within the cluster and components outside of the cluster, encrypted communication with client certificate authentication is configured by default.

- Understanding the ArcSight Platform Certificate Authorities
- Using an RE External Communication Certificate Signed by Your Trusted Certificate Authority

Understanding the ArcSight Platform Certificate Authorities

During installation, three self-signed Certificate Authorities (CA) are created automatically, two for signing certificates used exclusively for pod to pod communication within the cluster (RIC and RID CA), and the other for signing certificates for each pod that performs communication external to the cluster (RE CA). Only pods that perform external communication have a certificate that is signed by the external CA.

External cluster communication occurs not only with ArcSight components, but also with user web browsers and, in some cases, user clients of ArcSight APIs (such as the REST API). By default, when the user connects to the cluster, they will be presented with a certificate that has been signed by the self-signed external CA. Since the external CA is self-signed, the user's connection will not automatically trust the certificate because it will not be verifiable using a certificate chain that is already in the user's trust store.

To give users confidence they are connecting to the trusted cluster, we recommend signing the certificates that are presented to the user with a CA that is trusted by the user's trust store. There are two approaches to doing this that are described in the documentation below. These approaches are:

Method 1 - Signing the RE External Communication Certificate with Your Trusted Certificate Authority

This is the recommended approach, because it is theoretically more secure than the other approach, in that, it only involves transferring a CSR and public certificate between systems,

which does not put any private secrets at risk.

Method 2 - Importing an Externally Created Intermediate CA

This approach involves creating an Intermediate CA (key and certificate pair) in a system outside of the ArcSight Platform, and then importing it into the ArcSight Platform. While this approach does work, it is theoretically less secure than the other approach, because it involves transferring a CA private key between systems, which potentially exposes it to unintended parties.

Use only one of the two approaches above.

Using an RE External Communication Certificate Signed by Your Trusted Certificate Authority

Use only one of the two approaches below. The first one, "Signing the RE External Communication Certificate with Your Trusted Certificate Authority" approach is recommended for the reasons described in Understanding the ArcSight Platform Certificate Authorities.

Method 1 - Signing the RE External Communication Certificate with Your Trusted Certificate Authority

Signing the RE External Communication Certificate with Your Trusted Certificate Authority approach is recommended for the reasons described in Understanding the ArcSight Platform Certificate Authorities.

In order to sign the RE external communication certificate with your trusted CA, you need to (1) create a certificate signing request (CSR) from vault, (2) take it to your organization, (3) sign it, and (4) return the signed CSR and all the public chain-of-certificates used to sign it.

1. Export the following access token dependencies (you can remove these later if not needed):

```
export CDF_APISERVER=$(kubectl get pods -n core -o custom-
columns=":metadata.name"| grep cdf-apiserver)
```

```
export PASSPHRASE=$(kubectl get secret vault-passphrase -n core -o json
2>/dev/null | jq -r '.data.passphrase')
```

```
export ENCRYPTED_ROOT_TOKEN=$(kubectl get secret vault-credential -n core
-o json 2>/dev/null | jq -r '.data."root.token"')
```

```
export VAULT_TOKEN=$(echo ${ENCRYPTED_ROOT_TOKEN} | openssl aes-256-cbc -
md sha256 -a -d -pass pass:"${PASSPHRASE}")
```

2. Ask vault to generate the CSR by running the following command:

   Important: When you execute this command, proceed expeditiously through steps 3 and 4, as your cluster will not be able to issue external certificates while it waits for the CSR to be signed.

   ```
   kubectl exec -it -n core ${CDF_APISERVER} -c cdf-apiserver -- bash -c
   "VAULT_TOKEN=$VAULT_TOKEN vault write -tls-skip-verify -format=json
   RE/intermediate/generate/internal common_name=\"none-MF CDF RE CA on
   <FQDN of ArcSight Platform Virtual IP for HA or single master node>\"
   country=<Country> locality=<Locality> province=<Province>
   organization=<Organization> ou=<Organizational Unit>" | jq -r '.data.csr'
   > /tmp/pki_intermediate.csr
   ```

   Note: The `common_name` in the command above is an example common name. Substitue your own values for the common name to fit your environment. Additionally, your trusted certificate authority might require additional parameters in the CSR besides common_name. Ask your PKI team for what the required CSR parameters are and add the appropriate parameters to the command (similar to how the parameter common_name is specified). The parameter names for the vault command used above are documented at https://www.vaultproject.io/api-docs/secret/pki#generate-intermediate

3. Sign the CSR file with your trusted certificate authority, and save the result into the `intermediate.cert.pem` file.

   Examply only. A basic example is provided below. Your environment will likely be different.

   ```
   openssl ca -keyfile your-rootca-sha256.key -cert your-rootca-sha256.crt -
   config your-openssl-configuration-file -extensions v3_ca -notext -md
   sha256 -in /tmp/pki_intermediate.csr -out intermediate.cert.pem
   ```

   Make sure the `v3_ca` extension is enabled and a new certificate is useable as a certificate authority on its own. Otherwise, you will receive a warning in the next step that given certificates are not marked for CA use.

4. Create an `intermediate.chain.pem` file that includes the combination of the `intermediate.cert.pem`, the public certificate of your trusted certificate authority, and all intermediate public certificates in the chain between them so that `intermediate.chain.pem` includes the full trust chain.

   ```
   cp intermediate.cert.pem intermediate.chain.pem
   cat [parent-intermediate1.crt] [parent-intermediate2.crt] [...] your-
   rootca-sha256.crt >> intermediate.chain.pem
   ```

> If you have intermediate certificates between your `intermediate.cert.pem` and your trusted certificate authority, you must add the certificates in the specific order of the sequence of the chain, with the last certificate being the certificate of the root trusted CA.

5. Import the `intermediate.chain.pem` file into the cluster vault:

```
chaincerts=$(cat intermediate.chain.pem) && kubectl exec -it -n core
${CDF_APISERVER} -c cdf-apiserver -- bash -c "VAULT_TOKEN=$VAULT_TOKEN
vault write -tls-skip-verify -format=json RE/intermediate/set-signed
certificate=\"${chaincerts}\""
```

6. Update `ConfigMap RE_ca.crt` by running these commands:

```
reCrtForJson=$(sed -E ':a;N;$!ba;s/\r{0,1}\n/\\n/g'
intermediate.chain.pem) && kubectl patch configmap -n core public-ca-
certificates -p "{\"op\": \"replace\", \"data\":{\"RE_
ca.crt\":\"${reCrtForJson}\"}}"
```

```
ARCSIGHT_NS=$(kubectl get namespaces --no-headers -o custom-
columns=":metadata.name" | grep arcsight-installer)
```

```
if [ -n "$ARCSIGHT_NS" ];then reCrtForJson=$(sed -E ':a;N;$!ba;s/\r
{0,1}\n/\\n/g' intermediate.chain.pem); kubectl patch configmap -n
$ARCSIGHT_NS public-ca-certificates -p "{\"op\": \"replace\", \"data\":
{\"RE_ca.crt\":\"${reCrtForJson}\"}}";fi
```

7. (Conditional) If you already deployed ArcSight Capabilities onto the CDF, update the ArcSight Capabilities to use the updated RE external communication certificate, by following the instructions in Configuring ArcSight Kubernetes Pods to Use the Updated RE External Communication Certificate.

   If you deployed CDF but have not yet deployed any ArcSight Capabilities, you can skip those instructions.

Method 2 - Importing an Externally Created Intermediate CA

This is an alternate approach for signing certificates to connect to the trusted cluster. Before choosing this approach, ensure that your understand the other approach recommended in Understanding the ArcSight Platform Certificate Authorities.

To import an externally created intermediate CA:

1. Obtain an intermediate CA (key and certificate pair) from your trusted certificate authority.
   a. Name the certificate files as follows:
      - key file: `intermediate.key.pem`
      - certificate file: `intermediate.cert.pem`
   b. Obtain the root CA certificate (including chain), and put it in a file named ca.cert.pem.
2. Replace the existing RE CA in the ArcSight Platform with the intermediate CA you obtained in the step above, based on your type of deployment, on-premises **or** cloud.
   a. Change the directory:
      - For an on-premises deployment, run these commands:

        ```
        cd /opt/arcsight/kubernetes/scripts/
        ```

      - For a cloud deployment, run these commands:

        ```
        cd {path to cdf installer}/cdf-deployer/scripts/
        ```

   b. Run the following command to replace the existing RE CA:

      ```
      ./cdf-updateRE.sh write --re-crt=/pathto/intermediate.cert.pem --re-key=/pathto/intermediate.key.pem [--re-ca=/pathto/ca.cert.pem]
      ```

      Note: `--re-ca=/pathto/ca.cert.pem` is the path to the file containing the certificate of CA used to sign re-crt. It is not required when re-crt is self-signed or CA is included in re-crt.
3. (Conditional) If you already deployed ArcSight Capabilities onto CDF, proceed to the next section to update the ArcSight Capabilities to use the updated RE external communication certificate, Configuring ArcSight Kubernetes Pods to Use the Updated RE External Communication Certificate.

   However, if you have only deployed CDF, but have not deployed ArcSight Capabilities yet, you can skip that section.

Creating and Validating the Route 53 Certificate

A user-provided self-signed or CA-signed certificate is required for creating the Application Load Balancer (ALB). In this section, you will create a configuration file for the certificate signing request (CSR), create an intermediate certificate pair, sign the CSR, create a chained file for import, then import the self-signed certificate into Amazon Certificate Manager (ACM).

1. Create or update Route 53 certificates:
   a. Run the applicable command on a secure machine to generate the Route 53 certificate:
      - For a current version of SSL, run this command:

```
openssl req -nodes -newkey rsa:2048 -keyout
<your.route53dnsRecordsetName>.key.pem -out
<your.route53dnsRecordsetName>.csr.pem -subj
"/C=US/ST=State/L=City/O=Company
Inc./OU=IT/CN=<your.route53dnsRecordsetName>" -addext
"subjectAltName = DNS:<your.route53dnsRecordsetName>"
```

      - If your operating system does not support `-addext` for SSL, run this command:

```
openssl req -newkey rsa:2048 -sha256 -nodes -keyout
your.route53dnsRecodsetName.key.pem -out
your.route53dnsRecodsetName.csr.pem -subj
"/C=US/ST=CA/L=SU/O=MF/OU=IT/CN=<your.route53dnsRecordsetName>" -
extensions san -config <(echo '[req]'; echo 'distinguished_
name=req';echo 'req_extensions=san';echo '[san]'; echo
'subjectAltName=DNS:your.route53dnsRecodsetName')
```

      > `your.route53dnsRecordsetName` is your route53 record set name tracked in your AWS configuration worksheet. This command will create the private key file `<your.route53dnsRecordsetName>.key.pem` and the certificate signing request file `<your.route53dnsRecordsetName>.csr.pem`.

   b. Copy the certificate signing request `<your.route53dnsRecordsetName>.csr.pem` to your bastion or jump host machine.
   c. Run the following commands to sign the certificate signing request using your cluster RE certificate:

```
export CDF_APISERVER=$(kubectl get pods -n core -o custom-
columns=":metadata.name"| grep cdf-apiserver)
```

```
export PASSPHRASE=$(kubectl get secret vault-passphrase -n core -o
json 2>/dev/null | jq -r '.data.passphrase')
```

```
export ENCRYPTED_ROOT_TOKEN=$(kubectl get secret vault-credential -n
core -o json 2>/dev/null | jq -r '.data."root.token"')
```

```
export VAULT_TOKEN=$(echo ${ENCRYPTED_ROOT_TOKEN} | openssl aes-256-
cbc -md sha256 -a -d -pass pass:"${PASSPHRASE}")
```

```
export COMMON_NAME=<your.route53dnsRecordsetName>
```

```
export CSR=$(cat ${COMMON_NAME}.csr.pem)
```

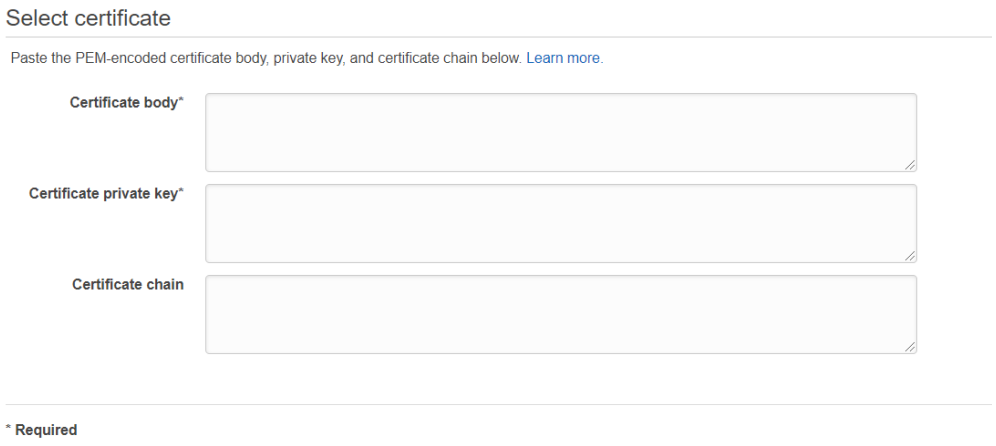
```
export WRITE_RESPONSE=$(kubectl exec -it -n core ${CDF_APISERVER} -c
cdf-apiserver -- bash -c "VAULT_TOKEN=$VAULT_TOKEN vault write -tls-
skip-verify -format=json RE/sign/coretech csr=\"${CSR}\"")
```

```
echo ${WRITE_RESPONSE} | jq -r ".data | .certificate" > ${COMMON_
NAME}.signed.crt.pem
```

```
echo ${WRITE_RESPONSE} | jq -r ".data | if .ca_chain then .ca_chain[]
else .issuing_ca end" > ${COMMON_NAME}.ca_chain.pem
```

The RE signed certificate is in file `${COMMON_NAME}.signed.crt.pem`. The certificate chain is in file `${COMMON_NAME}.ca_chain.pem`.

2. Import or update the certificate in Amazon Certificate Manager (ACM).
   a. Import the self-signed certificate into ACM (for a fresh installation):
      i. Log in to the AWS Console.
      ii. Browse to the Amazon Certificate Manager (ACM).
      iii. Click **Import a certificate**, and then complete the fields as follows:
      - **Certificate body:** Specify the contents of the signed certificate you created earlier. For example, `<your.route53dnsRecordsetName>.crt.pem`
      - **Certificate private key:** Specify the contents of the private key created by the CSR request creation. For example, `<your.route53dnsRecordsetName>.key.pem`
      - **Certificate chain:** Specify the contents of the chain file. For example, `<your.route53dnsRecordsetName>.ca.pem`



b. To update the certificate in ACM (for a current running installation):
   i. Log in to the AWS Console.
   ii. Browse to the Amazon Certificate Manager (ACM).
   iii. Search for your certificate Domain name or ID.
   iv. Select your Certificate ID, and click **Reimport**.
   v. Complete the fields as follows:
      - **Certificate body:** Specify the contents of the new signed certificate you created earlier. For example, `<your.route53dnsRecordsetName>.crt.pem`
      - **Certificate private key:** Specify the new contents of the private key created by the CSR request creation. For example, `<your.route53dnsRecordsetName>.key.pem`
      - **Certificate chain:** Specify the new contents of the chain file. For example, `<your.route53dnsRecordsetName>.ca.pem`

   For more details, see https://docs.aws.amazon.com/acm/latest/userguide/import-reimport.html
3. Click **Next**. Optionally, add any tags you wish to the import.
4. Click **Next**, and then, click **Import**.

   After the import, click the arrow next to the certificate ARN value. Note the value to your AWS worksheet for later use. For example:

Next Step: Configuring the Application Load Balancer (ALB).

Configuring the Application Load Balancer (ALB)

A load balancer serves as the single point of contact for clients. The load balancer distributes incoming application traffic across multiple targets, such as EC2 instances, in multiple

availability zones. Balancing the load increases the availability of your application. AWS supports several types of load balancers: application, network, and (obsoleted) classic. In this section, you will configure an application load balancer (ALB).

The ALB needs to be configured with locations to balance requests; this is realized by target groups. During the installation process you will create target groups for various ports: 3000 (CDF installation), 5443 (CDF management portal), and 443 (ArcSight Suite configuration).

Immediately after the core CDF bootstrap, only the installation portal is available on port 3000. The remaining two confgured will be created after the CDF UI installation process is completed.

Retrieving the CDF Ingress Service Node Port

To retrieve the CDF ingress service node port for 3000:

1. Run the following command on the bastion:
   `kubectl get svc -n core | grep frontend-ingress-controller-svc`

Example output:

```
frontend-ingress-controller-svc LoadBalancer 172.20.150.202 <none>
3000:30058/TCP 18h
```

2. Record the highlighted port number in your AWS worksheet as `Node port for 3000.` In the example shown, the port number is 30058.

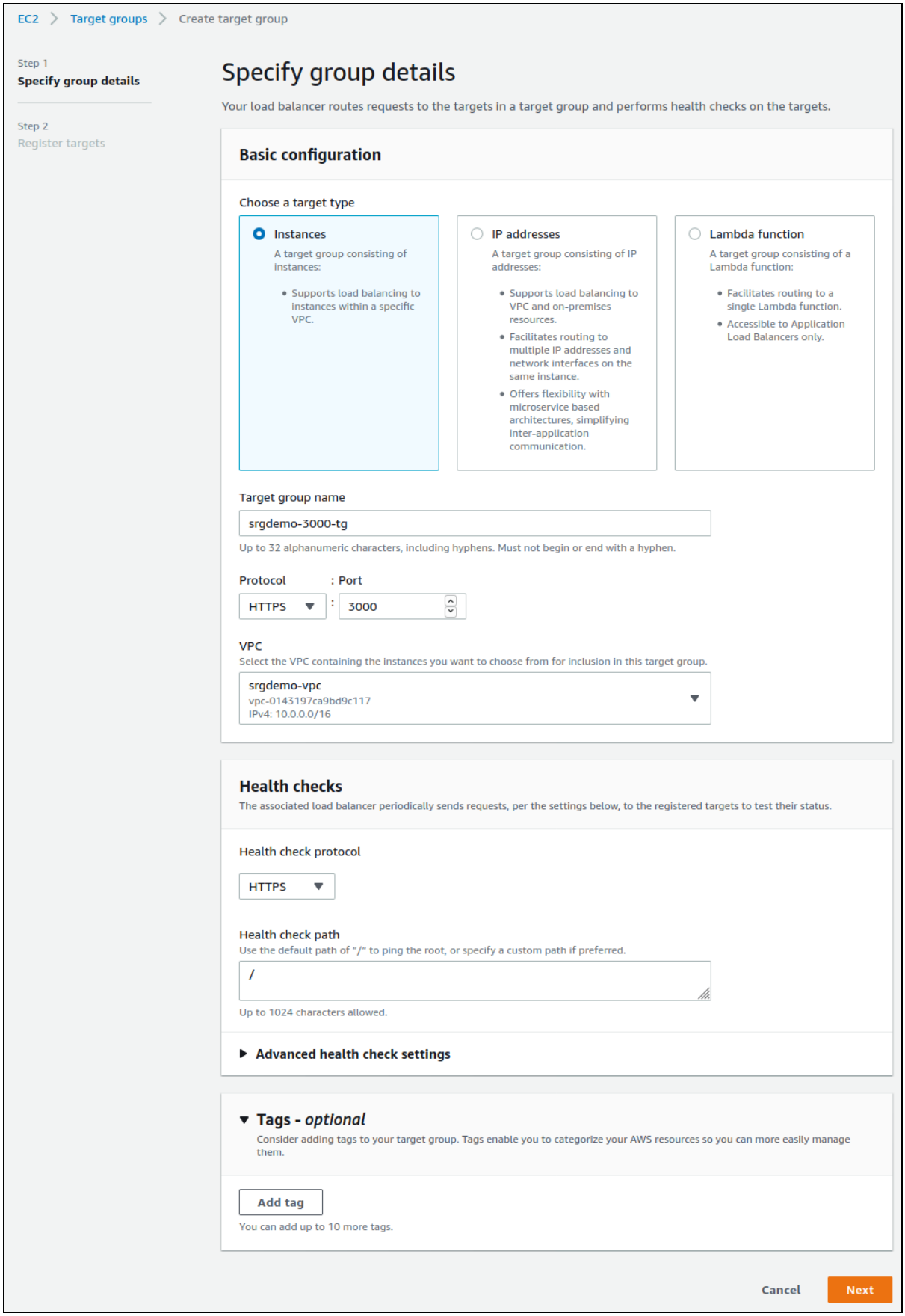
Next Step: Creating the Target Group for Port 3000

Creating the Target Group for Port 3000

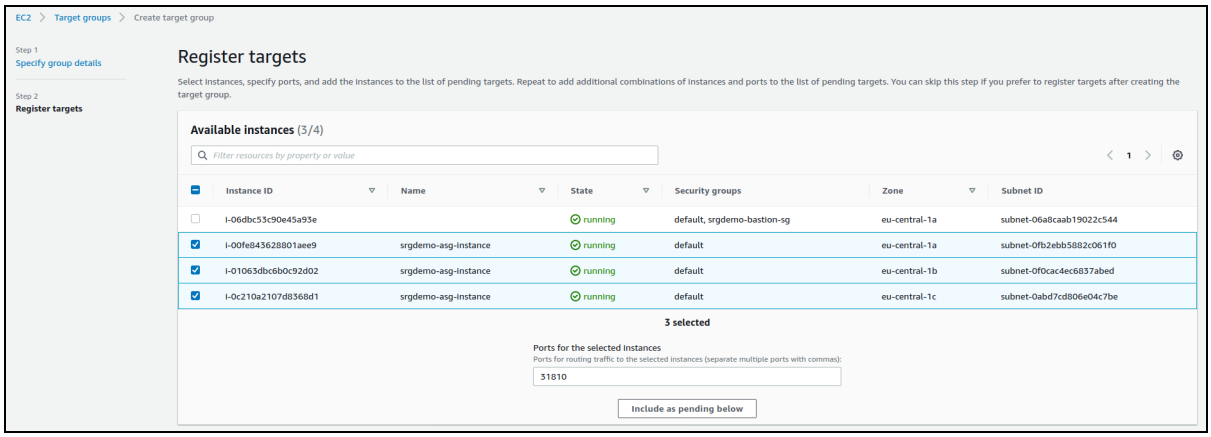
To create, tag, and add targets to the target group for port 3000 using the web UI:

1. Using the Find Services search tool, locate and browse to the EC2 Dashboard.
2. In the left navigation panel, under **Load Balancing**, click **Target Groups**.
3. On the **Target Groups** management page, click **Create target group.**
4. On the **Specify group details** page, specify values for the following:
   - Under **Choose a target type,** select **Instances**.
   - **Target group name:** Choose a descriptive name for easier identification. For example *srgdemo-3000-tg.*
   - **Protocol:** Change to *HTTPS*.
   - **Port:** Specify *3000*.

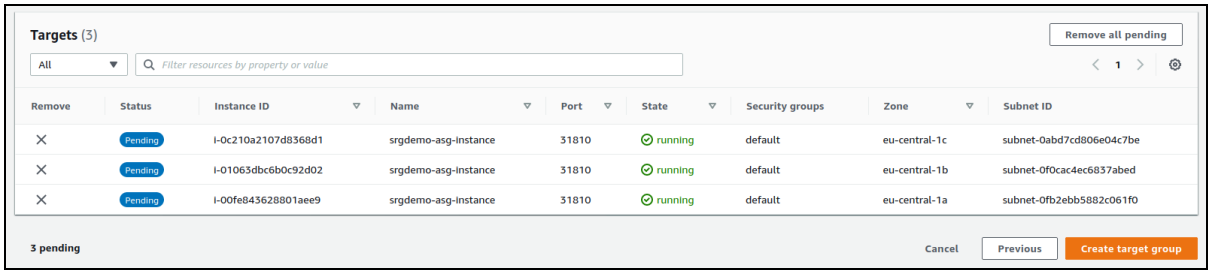
- **VPC:** Select your VPC.
- **Tags:** (Optional) Add descriptive tags as desired.


EC2 > Target groups > Create target group
Step 1
Specify group details
Step 2
Register targets
Specify group details
Your load balancer routes requests to the targets in a target group and performs health checks on the targets.
Basic configuration
Choose a target type
Instances
A target group consisting of instances:
Supports load balancing to instances within a specific VPC.
IP addresses
A target group consisting of IP addresses:
Supports load balancing to VPC and on-premises resources.
Facilitates routing to multiple IP addresses and network interfaces on the same instance.
Offers flexibility with microservice based architectures, simplifying inter-application communication.
Lambda function
A target group consisting of a Lambda function:
Facilitates routing to a single Lambda function.
Accessible to Application Load Balancers only.
Target group name
srgdemo-3000-tg
Up to 32 alphanumeric characters, including hyphens. Must not begin or end with a hyphen.
Protocol : Port
HTTPS : 3000
VPC
Select the VPC containing the instances you want to choose from for inclusion in this target group.
srgdemo-vpc
vpc-0143197ca9bd9c117
IPv4: 10.0.0.0/16
Health checks
The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.
Health check protocol
HTTPS
Health check path
Use the default path of "/" to ping the root, or specify a custom path if preferred.
/
Up to 1024 characters allowed.
Advanced health check settings
Tags - optional
Consider adding tags to your target group. Tags enable you to categorize your AWS resources so you can more easily manage them.
Add tag
You can add up to 10 more tags.
Cancel
Next

5. Under **Health Checks**, leave the **Health Check Protocol** set to *HTTPS*.
6. Click **Next**.
7. On the **Register targets** page, set values for the following:
   a. **Available instances:** Select your worker node instances, but **do not** select the bastion.
   b. **Ports:** For the selected instances, use the value you retrieved previously for the bastion as the corresponding node port for port 3000 and recorded in the AWS worksheet.



8. Click **Include as pending below.** All selected instances will be added to the list of pending instances.



9. Click **Create target group.** You will be redirected back to the target group management page.
10. From the list, select the newly created target group.
11. From the bottom of the page, record its ARN in the AWS worksheet.

To create, tag, and assign targets to the target group for port 3000 using the CLI:

1. Run the following command:

```
# aws elbv2 create-target-group \
--name <Target group 3000 Name> \
--protocol HTTPS \
--port 3000 \
--vpc-id <VPC ID> \
```

```
    --health-check-protocol HTTPS \
    --target-type instance
```

Where:

<Target group 3000 Name>: Choose some descriptive name such as srgdemo-3000-tg. Record the value in the AWS worksheet.

<VPC ID>: The ID of your VPC, as recorded in your AWS worksheet.

Example input and output:

```
# aws elbv2 create-target-group --name srgdemo-3000-tg --protocol HTTPS --
port 3000 --vpc-id vpc-0143197ca9bd9c117 --health-check-protocol HTTPS --
target-type instance
```

```
Target group for port 3000 description
```

```
{
```

```
"TargetGroupArn": "arn:aws:elasticloadbalancing:eu-central-
1:115370811111:targetgroup/srgdemo-3000-tg/c0684be94405b6b7",
```

```
"TargetGroupName": "srgdemo-3000-tg",
```

```
"Protocol": "HTTPS",
```

```
"Port": 3000,
```

```
"VpcId": "vpc-0143197ca9bd9c117",
```

```
"HealthCheckProtocol": "HTTPS",
```

```
"HealthCheckPort": "traffic-port",
```

```
"HealthCheckEnabled": true,
```

```
"HealthCheckIntervalSeconds": 30,
```

```
"HealthCheckTimeoutSeconds": 5,
```

```
"HealthyThresholdCount": 5,
```

```
"UnhealthyThresholdCount": 2,
```

```
"HealthCheckPath": "/",
```

```
"Matcher": {
"HttpCode": "200"
},
"TargetType": "instance"
}
]
}
```

From the output, record the value of `TargetGroupArn` in your AWS worksheet.

Tagging the Target Group (CLI)

Optionally, you can tag the target group for easier identification.

To tag the target group using the CLI:

1. Run the following command:
   ```
   # aws elbv2 add-tags \
   --resource-arns <Target group 3000 ARN> \
   --tags Key=owner,Value=<owner>
   ```

Example:

```
# aws elbv2 add-tags \
--resource-arns arn:aws:elasticloadbalancing:eu-central-
1:115370811111:targetgroup/srgdemo-3000-tg/c0684be94405b6b7 \
--tags Key=owner,Value=srgdemo
```

Adding Targets to the Target Group Using the CLI

To add targets to the target group:

1. Run the following command:
   ```
   # aws elbv2 register-targets \
   --target-group-arn <Target group 3000 ARN> \
   --targets Id="Instance 1 ID,Port=<Node port for 3000>" Id="Instance 2
   ID,Port=<Node port for 3000>" Id="Instance 3 ID,Port=<Node port for 3000>"
   ```

Where:

`<Instance x ID>`: Use the instance IDs you gathered for instances of the Auto Scaling group. Refer to the AWS worksheet for these values.

`<Node port for 3000>`: Use the port number for 3000 from your AWS worksheet.

Example:

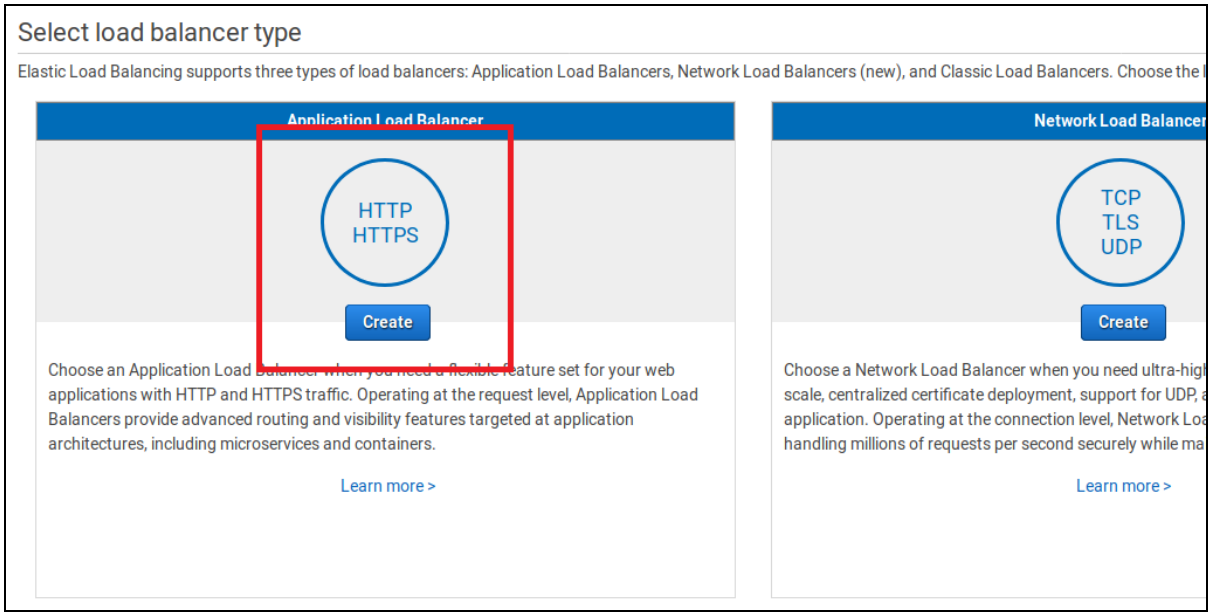
```
# aws elbv2 register-targets \
--target-group-arn arn:aws:elasticloadbalancing:eu-central-
1:115370811111:targetgroup/srgdemo-3000-tg/c0684be94405b6b7 \
--targets Id="i-05662f9ef84c182ca,Port=30058" Id="i-
07cfcd6716e9890b5,Port=30058" Id="i-08d819b5ccabe83cb,Port=30058"
```

Next Step: Creating the Application Load Balancer

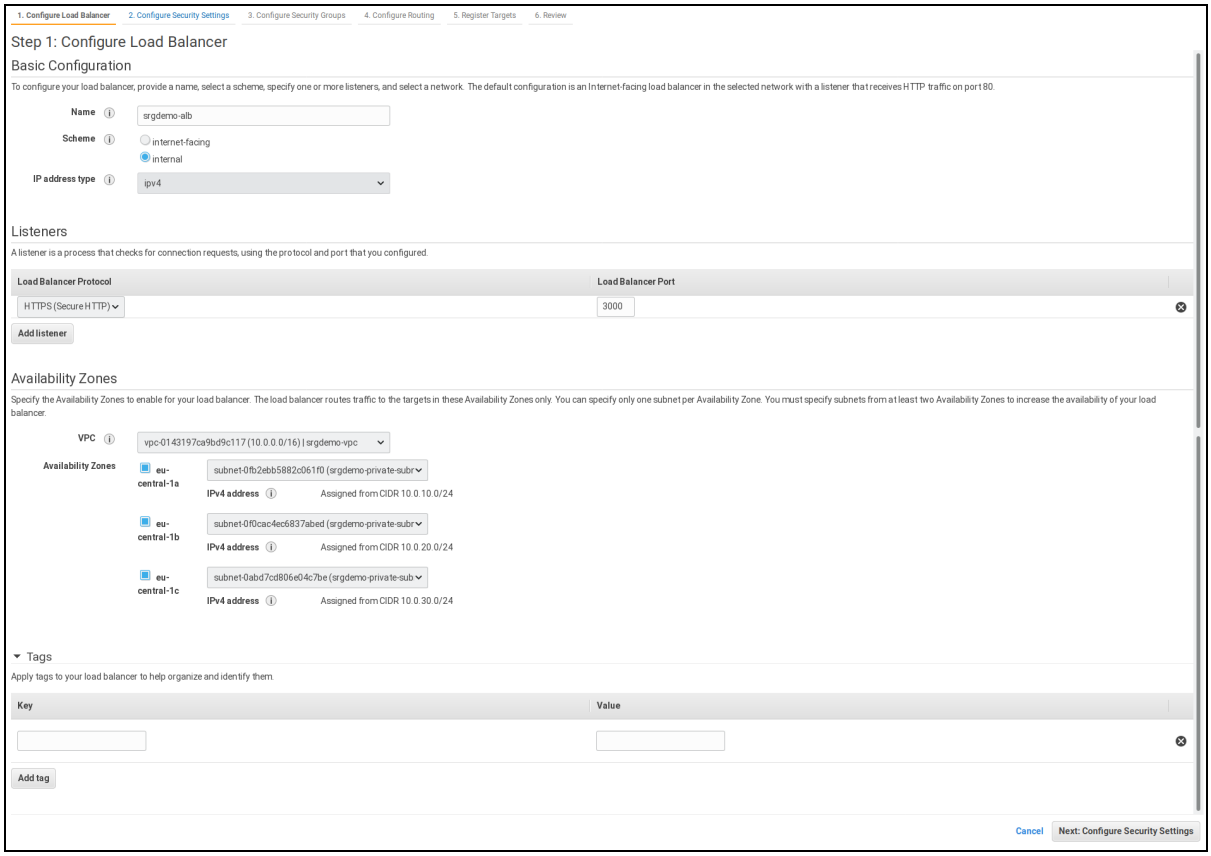
Creating the Application Load Balancer

To create the ALB using the Web UI:

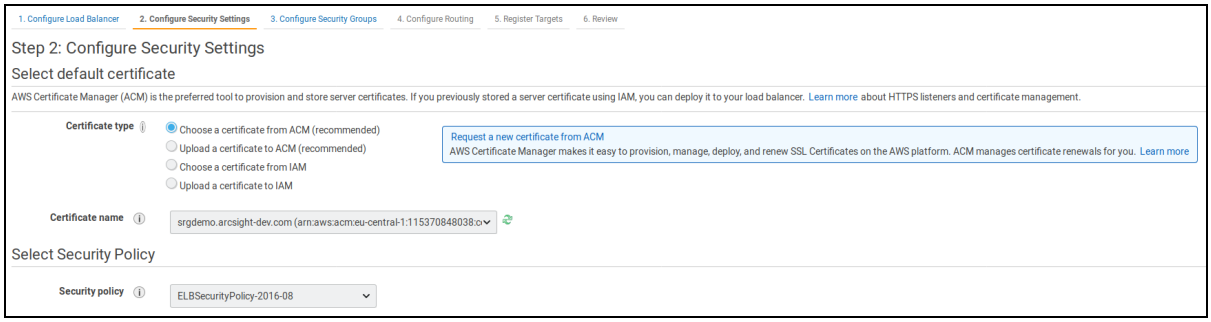
1. Using the Find Services search tool, locate and browse to the EC2 dashboard.
2. In the left navigation panel, under **Load Balancing**, select **Load Balancers.**
3. On the **Load Balancing** management page, click **Create Load Balancer.**



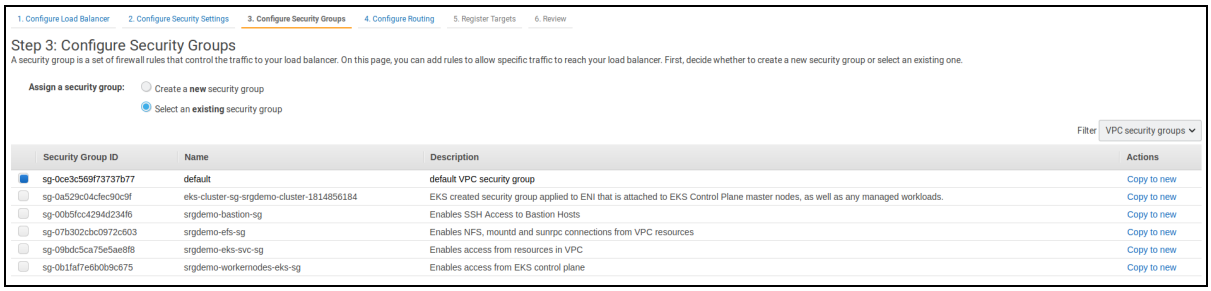
4. In the **Application Load Balancer** panel to the left, click **Create**.



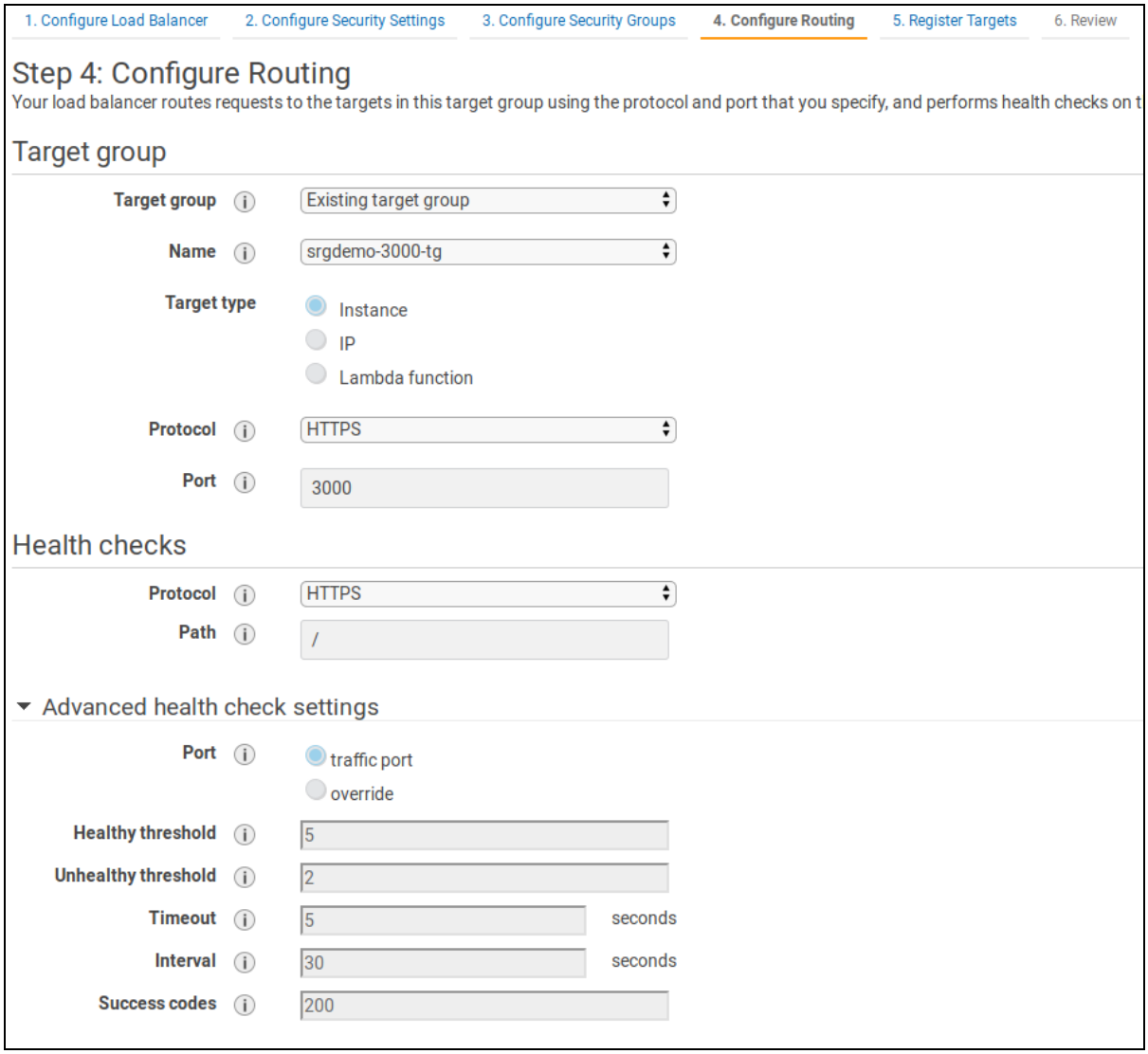
5. On the **Configure Load Balancer** page, specify values for the following:
   a. **Name:** Select an ALB name
   b. **Scheme:** Set to *Internal*.
   c. **Protocol:** Select *HTTPS*.
   d. **Port:** Specify 3000.
   e. **VPC:** Select your VPC.
   f. **Availability Zones:** Select all three zones, then in each, specify the corresponding private subnet.
   g. **Tags:** (Optional) Specify any desired tags.
6. Click **Next: Configure Security Settings**.



7. On the **Configure Security Settings** page, specify values for the following:
   a. **Certificate type:** Select a certificate from ACM (recommended).
   b. **Certificate name:** Select the certificate that you previously created.
   c. **Security policy:** Leave unchanged.
8. Click **Next: Configure Security Groups.**



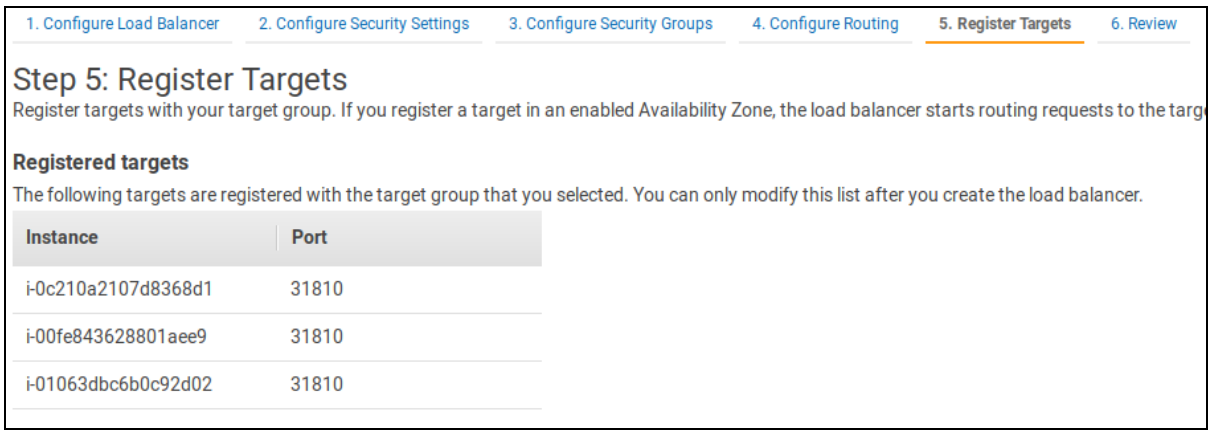
9. On the **Configure Security Groups** page, specify values for the following:
   a. **Assign a security group:** Leave set to *Select an existing security group.*
   b. In the list below, make sure only the Intra VPC security group created by your AWS infrastructure administrators (and recorded in the AWS worksheet) is selected.
10. Click **Next: Configure Routing**.



11. On the **Configure routing** page, specify values for the following:
    a. **Target group:** Change to **Existing target group.**
    b. **Name:** Select the target group that you have previously created.

    The remaining settings may be left at default values.

12. Click **Next: Register** targets.



13. On the **Register targets** page, verify your instance IDs and node ports.
14. Click **Next: Review.**
15. On the **Review** page, verify all settings and then click **Create.** An ALB creation status will be displayed but will not be updated because the page is not dynamic.) However, you might close the wizard.
16. On the **Description** tab, select **Edit attributes**.
17. For **Idle timeout**, specify a value of **300** in seconds, and click **Save**.

To create the ALB using the CLI:

1. Run the following command:
```
aws elbv2 create-load-balancer \
--name <ALB Name> \
--subnets <subnetIds> \
--security-groups <Intra VPC Security group Id> \
--scheme internal \
--type application \
--ip-address-type ipv4
```

Where:

`<ALB Name>`: Specify a name for easy application load balancer identification, and record it to the AWS worksheet.

`<subnet Ids>`: Use the space-separated IDs of all three private subnets in the VPC.

`<Intra VPC Security group Id>`: ID of the Intra VPC security group created previously and recorded in the AWS worksheet.

Example input and output:

```
#aws elbv2 create-load-balancer \
--name srgdemo-alb \
--subnets subnet-0fb2ebb5882c061f0 subnet-0f0cac4ec6837abed subnet-
```

```
0abd7cd806e04c7be \
--security-groups sg-0ce3c569f73737b77 \
--scheme internal \
--type application \
--ip-address-type ipv4
```


```
{
   "LoadBalancers":[
      {
         "LoadBalancerArn":"arn:aws:elasticloadbalancing:eu-central-
1:115370811111:loadbalancer/app/srgdemo-alb/8718b24107ef591b",
         "DNSName":"internal-srgdemo-alb-505957021.eu-central-
1.elb.amazonaws.com",
         "CanonicalHostedZoneId":"Z215JYRZR1TBD5",
         "CreatedTime":"2020-06-15T09:16:12.480000+00:00",
         "LoadBalancerName":"srgdemo-alb",
         "Scheme":"internal",
         "VpcId":"vpc-0143197ca9bd9c117",
         "State":{
            "Code":"provisioning"
         },
         "Type":"application",
         "AvailabilityZones":[
            {
               "ZoneName":"eu-central-1c",
               "SubnetId":"subnet-0abd7cd806e04c7be",
               "LoadBalancerAddresses":[

               ]
            },
            {
               "ZoneName":"eu-central-1b",
               "SubnetId":"subnet-0f0cac4ec6837abed",
               "LoadBalancerAddresses":[

               ]
            },
            {
               "ZoneName":"eu-central-1a",
               "SubnetId":"subnet-0fb2ebb5882c061f0",
               "LoadBalancerAddresses":[

               ]
            }
         ],
         "SecurityGroups":[
            "sg-0ce3c569f73737b77"
```

```
        ],
        "IpAddressType":"ipv4"
     }
  ]
}
```

2. Record the `LoadBalancerArn`, `DNSName` and `CanonicalHostedZoneId` values in the AWS worksheet as **`ALB ARN, ALB DNS name,`** and **`ALB Canonical hosted zone ID.`**
3. Creation of the ALB takes approximately five minutes. To check the ALB creation status, run the following command using the `ALB ARN` value:
   ```
   #aws elbv2 describe-load-balancers \
   --load-balancer-arns <LoadBalancerArn> \
   | jq -r '.LoadBalancers[0].State.Code'
   ```
4. Repeat Step 3 until the returned status changes to *Active*.

   Do not proceed until the status has changed to Active.

   For Example

   ```
   aws elbv2 describe-load-balancers \
   --load-balancer-arns arn:aws:elasticloadbalancing:eu-central-
   1:115370811111:loadbalancer/app/srgdemo-alb/8718b24107ef591b \
   | jq -r '.LoadBalancers[0].State.Code'
   ```
5. After the status is Active, run the following command to increase connection idle timeout for Load Balancer:

   ```
   aws elbv2 modify-load-balancer-attributes \
   --load-balancer-arn <LoadBalancerArn> \
   --attributes Key=idle_timeout.timeout_seconds,Value=300
   ```

Adding the Listener for Port 3000

This action will connect the ALB to NLB 3000 through the target group. Incoming requests to the ALB on port 3000 will be directed to the Kubernetes `frontend-ingress-controller-svc` service.

To add the listener:

Run the following command:

```
aws elbv2 create-listener \
--load-balancer-arn <ALB ARN> \
--protocol HTTPS \
--port 3000 \
--certificates CertificateArn=<Certificate ARN> \
--default-actions Type=forward,TargetGroupArn=<Target group 3000 ARN>
```

Where:

`<ALB ARN>`: Use the ALB ARN recorded in the AWS worksheet.

`<Certificate ARN>`: Use the certificate ARN recorded in the AWS worksheet.

`<Target group 3000 ARN>`: Use the target group for port 3000 ARN recorded in the AWS worksheet.

Example input and output:

```
aws elbv2 create-listener \
--load-balancer-arn arn:aws:elasticloadbalancing:eu-central-
1:115370811111:loadbalancer/app/srgdemo-alb/8718b24107ef591b \
--protocol HTTPS --port 3000 --certificates CertificateArn=arn:aws:acm:eu-
central-1:115370811111:certificate/691ec232-98ff-45ed-8e69-1d15c0447538 \
```

```
--default-actions \
Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:eu-central-
1:115370811111:targetgroup/srgdemo-3000-tg/c0684be94405b6b7
```

```
"Listeners":[
   {
      "ListenerArn":"arn:aws:elasticloadbalancing:eu-central-
1:115370811111:listener/app/srgdemo-alb/8718b24107ef591b/32a42e4edb52466b",
      "LoadBalancerArn":"arn:aws:elasticloadbalancing:eu-central-
1:115370811111:loadbalancer/app/srgdemo-alb/8718b24107ef591b",
      "Port":3000,
      "Protocol":"HTTPS",
      "Certificates":[
         {
            "CertificateArn":"arn:aws:acm:eu-central-
1:115370811111:certificate/691ec232-98ff-45ed-8e69-1d15c0447538"
         }
      ],
      "SslPolicy":"ELBSecurityPolicy-2016-08",
      "DefaultActions":[
```

```
         {
            "Type":"forward",
            "TargetGroupArn":"arn:aws:elasticloadbalancing:eu-central-
1:115370811111:targetgroup/srgdemo-3000-tg/c0684be94405b6b7",
            "ForwardConfig":{
               "TargetGroups":[
                  {
                     "TargetGroupArn":"arn:aws:elasticloadbalancing:eu-
central-1:115370811111:targetgroup/srgdemo-3000-tg/c0684be94405b6b7",
                     "Weight":1
                  }
               ],
               "TargetGroupStickinessConfig":{
                  "Enabled":false
               }
            }
         }
      ]
   }
]
}
```

Next Step: Directing the Route 53 RecordSet to the ALB

Directing the Route 53 Record Set to the ALB

Although it is technically possible to connect to the ALB using its DNS name (such as `internal-srgdemo-alb-505957021.eu-central-1.elb.amazonaws.com`), this is not recommended for the following reasons:

- The URL is hard to remember and a user is forced to bookmark it in order to use it.
- You are unable to create the certificate for this domain, so browsers will always warn users about the insecure connection.

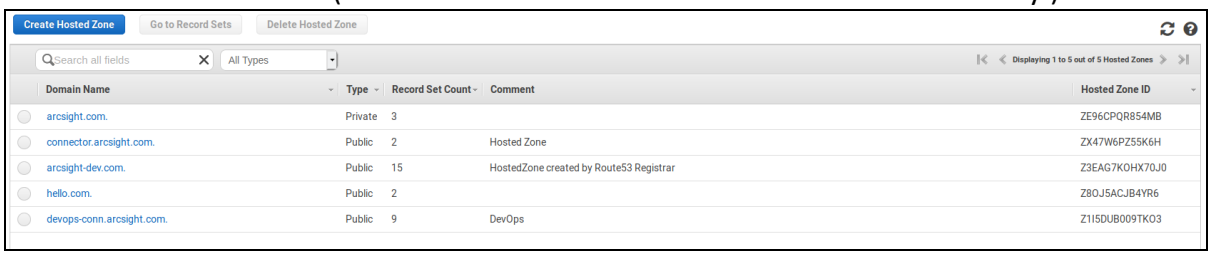
Previously, we created a record set in the Route 53 hosted zone and requested a certificate for the chosen domain name. You can now direct the record set to the application load balancer.

Using the Web UI

To direct the Route 53 record set to the ALB using the web UI:

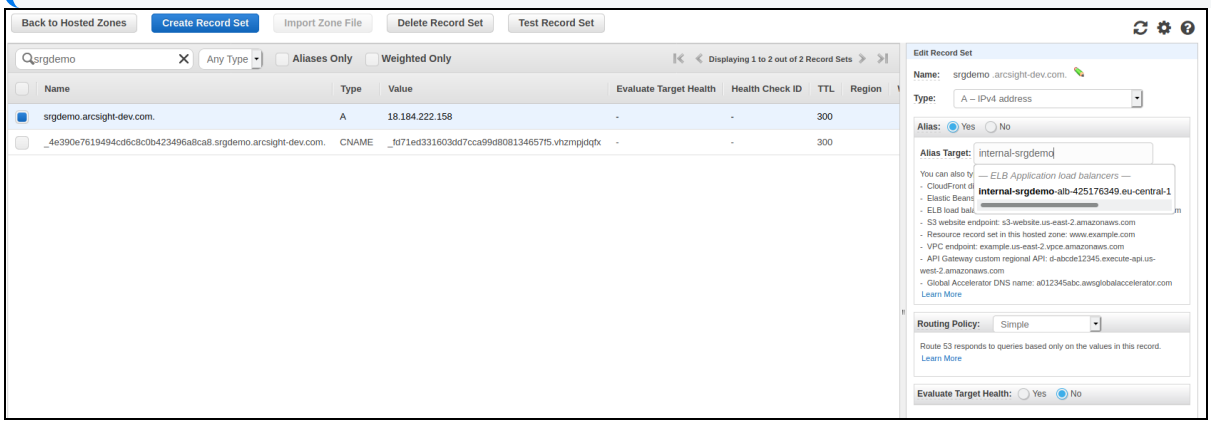
1. Using the Find Services search tool, locate and browse to the Route 53 Dashboard.
2. In the left navigation panel, select **Hosted zones**. Ignore any errors generated during this process.

3. From the hosted zones list, select the same hosted zone as you chose for creating the new Route 53 record set. (Use the search box to search for the zone if necessary.)



4. Create the record set as outlined here, using the following values in the details pane.
   a. **Alias:** Change to *Yes.*
   b. **Alias Target:** Start typing `internal-<domain name>`; for example, `internal-srgdemo`. The long list will be filtered and only your ALB will be displayed. Select it.

Upon selection the word *dialstack* will be prefixed to the name you entered. This prefix can be ignored.



5. Click **Save Record Set.**

Using the CLI

To direct the Route 53 record set to the ALB Using the CLI:

1. Copy the template `UpdateRecordSetToALB.json` and open the copy in a text editor. This template is available in the `arcsight-platform-cloud-installer-<version>/aws-scripts/objectdefs/` subfolder.
2. Edit the following:
   a. **Record name:** Combine the subdomain (such as `srgdemo`) and hosted zone name. For example, `arcsight-dev.com`). You can also refer to the AWS worksheet for the RecordSet name.

b. **ALB Canonical hosted zone ID:** Get the value from the AWS worksheet.

c. **ALB DNS name:** Get the value from the AWS worksheet.

3. Run the following command:

```
aws route53 change-resource-record-sets \
--hosted-zone-id <Hosted zone Id> \
--change-batch file://UpdateRecordSetToALB.json
```

Where:

`<Hosted zone Id>`:Use the Hosted Zone ID from the AWS worksheet. For example, `/hostedzone/Z3EAG7KOHX70J0`.

`<change-batch file>`: Replace the parameter with the name of your own modified instance of the linked JSON template.

Configure Browser

Finally, configure the browser on your bastion to trust the certificate.

For example, in Mozilla Firefox,

1. Select **Preferences > Privacy & Security > Certificates.**
2. Click **View Certificates.**
3. Cick **Authorities**.
4. Click **Import** and then browse to the ca.cert.pem to import the file.

Consult your browser documentation for the exact procedure on your browser.

Next Step: Downloading the Installation Packages

Downloading the Installation Packages for an AWS Deployment

Use this procedure to download the packages for installation or upgrade:

- **For Installation:** Follow the Checklist: Planning to Deploy ArcSight Capabilities on AWS to ensure a successful installation.
- **For Upgrade**: Follow the Checklist: Upgrading the AWS Deployment of ArcSight Platform to ensure a successful upgrade.

1. To identity the files to download to your secure network location, see Downloading and Installing the ArcSight Platform Installation Files in the ArcSight Platform Release Notes.
2. From a secure network location, download the installation packages for the CDF Installer and the product of your choice from the Software Licenses and Downloads portal.

This secure network location must be able to access your instance of AWS.

3. Extract the `arcsight-platform-cloud-installer-XX.X.X.XXX.zip` file.
4. Review the following table to understand the utility scripts and some templates used during the AWS deployment of ArcSight Platform. The utility scripts and templates are located in the `aws_scripts/scripts` directory.

Script	Description
`generate_aws_secrets`	Generates new Kubernetes Secrets for connecting from the cluster to the Elastic Container Registry (ECR). IMPORTANT: The generated credentials and secrets are valid only 12 hours after generation. If the credentials and secrets have expired, use this script following the instructions in Refresh the ECR credentials in the K8s.
`init_efs`	Creates the required folder structure on the Elastic File Storage (EFS) and assigns correct ownership and permissions. You will use this script when configuring EFS for the ArcSight Suite. Parameters for this script are discussed in the following sections. NOTE: Execute this script without parameters to display the help.
`upload_images_to_ECR`	Uploads the CDF and product images to the ECR to make them accessible to K8s. The script performs tasks in the background required specifically by the AWS ECR. Parameters for this script are discussed in the following sections. **Note**: Execute this script without parameters to display the help.
`workernodes-userdata`	The installation uses this script indirectly for enabling worker nodes to join the Kubernetes cluster.

Next Step (for installation): Installing CDF and Products

Installing CDF

With the CDF bootstrap procedure completed, the next step in installing CDF and the ArcSight Suite is to connect to the CDF web installation UI, then proceed through the installation wizard.

Accessing the CDF Installation UI

At the end of the CDF bootstrap process, you were prompted to connect to the URL `https://<external access host>:3000`, which is part of the standard CDF installation procedure.

The CDF installation port 3000 is now accessible through the chosen Route 53 record set, but only within the VPC. The VPC and any resources inside it are isolated from access from the internet (except for the bastion host, which is accessible on port 22, the SSH port).

You cannot access the created DNS record outside the VPC, since that DNS record will resolve to one of the three private subnet IP addresses which are hidden (and, in our case, in a private A-class IP range).

There are two methods for connecting a browser to the CDF port 3000: forwarding DISPLAY and forwarding local ports.

Forwarding DISPLAY

Prerequisite: An operating system capable of running X-server, such as *nix, linux, or MacOS.

For connection to the bastion, the easiest and fastest option is to connect to the bastion using SSH with the -X or -Y switch. This will set the remote DISPLAY accordingly, so the process running remotely will render its UI on the local X-server. The bastion host you configured earlier has the Mozilla Firefox browser installed.

The drawback of this method is that only one user can be connected and use the web browser, and the browser response might be quite slow. Any subsequent user will receive a message that the browser is already running, and results in significant lag while in the browser. However, the browser is used only for installation and configuration tasks, which are typically done once and by a single user, so the impact will likely be small.

To connect with this method:

1. Using SSH, connect to the bastion host with the additional parameters for dbus. Example command:
   `ssh -i /{path to ssh key} /aws.pem -X centos@54.188.142.125 'firefox https://srgdemo.arcsight-dev.com:3000'`
2. Browse to the URL that CDF returned at the end of its CLI installation. For example:
   `https://srgdemo.arcsight-dev.com:3000`

Forwarding local ports

Prerequisite: Ability to execute SSH with command line switches, as well as the Web UI ability to edit the system file /etc/hosts or the corresponding file.

To connect with this method, connect to the bastion host, adding the -L parameter. Example:
`ssh -i .ssh/srgdemo.pem -L 3000:srgdemo.arcsight-dev.com:3000 centos@3.120.237.11`

The -L parameter opens local port 3000 and connects each request to the srgdemo.arcsight-dev.com port 3000 on the remote side. So, the bastion resolves srgdemo.arcsight-dev.com and opens a connection to it on port 3000.

The second part of this approach is to edit /etc/hosts, and add your domain to the line containing localhost. Example: 127.0.0.1 localhost srgdemo.arcsight-dev.com.

When editing your `etc/hosts` file, ensure that the IP address specified each host is unique and not duplicated across hosts. A single IP address can be associated with multiple hostnames, but the same IP address may not be used for multiple hosts.

Open your preferred browser and direct it to the address that CDF output at the end of its CLI installation. Here we will use the example: `https://srgdemo.arcsight-dev.com:3000`.

CDF Web UI Installation

Once you have chosen your connection method and successfully connected to the CDF installation portal, perform the steps outlined to complete the CDF installation. During the usual installation process there are two steps where optional additional tasks or special handling might occur: during downloading images and setting file storage. These are explained in more detail in the following sections.

Downloading Images

Downloading images requires the CDF/K8s access to the ECR and checking for the presence of respective Docker images there. If more than 12 hours has passed between the bootstrapping CDF and checking image availability, then the ECR credentials will expire, and you will need to update credentials for both CDF and Kubernetes (k8s).

Follow this procedure to refresh the ECR credentials: Refresh the ECR credentials in the K8s

Setting File Storage

When setting the File Storage it is not possible to use the auto-discovery feature of remote mount points. You should specify a value for File Server and specify a storage volume manually. For **File Server**, supply the value of the `Filesystem FQDN` from the AWS worksheet. Then click the double-arrows and fill in the path to the volume. In our example it will be `/srgdemo/arcsight-volume`. This value was first displayed as the output of the `init_efs` script.

Installation finished

At the end of CDF installation, you are prompted to connect to the CDF management portal on the same host, this time using port 5443. Connection to port 5443 is not possible yet, however, as more network resources need to be configured.

Next, perform "Performing Post Installation Network Configuration" below

Performing Post Installation Network Configuration

During the CDF web UI installation, new services providing CDF management and re-configuration were created. These services listen on different ports than the CDF installation

UI. CDF management listens on port 5443 and re-configuration on port 443. In this section we will perform network configuration for these ports.

Get CDF Ingress Service Node Port for Port 5443

To get the ingress service node port for port 5443:

1. Run the following command:
   `kubectl get svc -n core | grep portal-ingress-controller-svc`

Example output:

```
portal-ingress-controller-svc NodePort 172.20.26.194 <none>
5443:31704/TCP,5444:32558/TCP 170m
```

2. Record the ingress service node port for port 5443 (in the example, 31704) in the AWS worksheet.

Next Step: Create a Target Group for Port 5443

Creating a Target Group for Port 5443

To create the target group for port 5443 using the web UI:

1. Using the Find Services search tool, locate and browse to the EC2 dashboard.
2. In the left navigation panel, under **Load Balancing**, click **Target Groups**.
3. On the **Target Groups** management page, click **Create target group.**
4. On the **Specify group details** page, specify values for the following:
   a. Under **Choose a target type,** select **Instances**.
   b. **Target group name:** Select a descriptive name for easier identification; for example, *srgdemo-5443-tg.*
   c. **Protocol:** change to *HTTPS.*
   d. **Port:**Specify 5443.
   e. **VPC:** Select your VPC.
   f. **Tags:** (Optional) Add descriptive tags as desired.
   g. **Health Checks:** Under **Health check protocol,** select HTTPS. For path, specify */th/cmak.*
5. Leave all other settings on the page at their default values, and click **Next**.
6. On the **Register Targets** page, set values for the following:
   a. **Available instances:** Select your instances; however, do not select the bastion.

b. **Ports for the selected instances:** For the selected instances, use the value you retrieved previously and recorded in the AWS worksheet as `Node Port for Port 5443.`

7. Click **Include as pending below.** All marked instances are added to the list of pending instances.
8. Click **Create target group.** You are redirected back to the target group management page.
9. From the list, select the newly created target group. From the bottom of the page, record its ARN in the AWS worksheet.

To create the target group for port 5443 using the CLI:

1. Run the following command:

```
# aws elbv2 create-target-group \
--name <Target group 5443 Name> \
--protocol HTTPS \
--port 5443 \
--vpc-id <VPC ID> \
--health-check-protocol HTTPS \
--target-type instance
```

Where:

`<Target group 5443 Name>`: Specify a descriptive name, such as `srgdemo-5443-tg`, and record the value in the AWS worksheet.

`<VPC ID>`: The ID of your VPC as recorded in your AWS worksheet.

Example input and output:

```
# aws elbv2 create-target-group \
   --name srgdemo-5443-tg \
   --protocol HTTPS \
   --port 5443 \
   --vpc-id vpc-0143197ca9bd9c117 \
   --health-check-protocol HTTPS \
   --target-type instance
```

```
{
      "TargetGroups":[
         {
            "TargetGroupArn":"arn:aws:elasticloadbalancing:eu-central-
   \1:115370811111:targetgroup/srgdemo-3000-tg/c0684be94405b6b7",
            "TargetGroupName":"srgdemo-5443-tg",
            "Protocol":"HTTPS",
            "Port":5443,
            "VpcId":"vpc-0143197ca9bd9c117",
            "HealthCheckProtocol":"HTTPS",
```

```
         "HealthCheckPort":"traffic-port",
         "HealthCheckEnabled":true,
         "HealthCheckIntervalSeconds":30,
         "HealthCheckTimeoutSeconds":5,
         "HealthyThresholdCount":5,
         "UnhealthyThresholdCount":2,
         "HealthCheckPath":"/",
         "Matcher":{
            "HttpCode":"200"
         },
         "TargetType":"instance"
      }
   ]
}
```

2. From the output, record the value of `TargetGroupArn` in your AWS worksheet.

Tagging the Target Group (CLI)

Optionally, you can tag the target group for easier identification.

To tag the target group using the CLI:

1. Run the following command:
   ```
   # aws elbv2 add-tags \
   --resource-arns <Target group 5443 ARN> \
   --tags Key=owner,Value=<owner>
   ```

Example:

```
                     # aws elbv2 add-tags \
--resource-arns arn:aws:elasticloadbalancing:eu-central-
1:115370811111:targetgroup/srgdemo-5443-tg/c0684be94405b6b7 \
--tags Key=owner,Value=srgdemo
```

Next Step: Add Targets to the Target Group for Port 5443

Adding Targets to the Target Group for Port 5443

To add targets to the target group:

1. Run the following command:
   ```
   # aws elbv2 register-targets \
   --target-group-arn <Target group 5443 ARN> \
   --targets Id="Instance 1 ID,Port=<Node port for 5443>" Id="Instance 2
   ID,Port=<Node port for 5443>" Id="Instance 3 ID,Port=<Node port for 5443>"
   ```

Where:

`<Instance x ID>`: Use the instance IDs you gathered for instances of the Auto Scaling group. Refer to the AWS worksheet.

`<Target group 5443 ARN>`: The ARN of the target group you just created.

`<Node port for 5443>`: Use the `node port number for 5443` from the AWS worksheet.

Example:

```
             # aws elbv2 register-targets \
--target-group-arn arn:aws:elasticloadbalancing:eu-central-
1:115370811111:targetgroup/srgdemo-5443-tg/a096cb67c2f9144d \
--targets Id="i-05662f9ef84c182ca,Port=31704" Id="i-07cfcd6716e9890b5,Port=31704"
Id="i-08d819b5ccabe83cb,Port=31704"
```

Next Step: Adding a Listener for Port 5443 to the ALB

Adding a Listener for Port 5443 to the ALB

Similarly to the method used to add a listener for port 3000, here we will create a path for requests on port 5443 to be routed to the Kubernetes portal-ingress-controller-svc service through the target group created above.

To add a listener for port 5443 to the ALB using the web UI:

1. Using the Find Services search tool, locate and browse to the EC2 Dashboard.
2. In the left navigation panel, under **Load Balancing**, click **Load Balancers.**
3. From the list of load balancers, select your previously created Application Load Balancer (ALB).
4. On the **Listeners** tab, click **Add Listener** and set values for the following:
   a. **Protocol: port:** Change to *HTTPS* and *5443*.
   b. **Default action(s):** Choose the action **Forward to...**, then choose your target group for port 5443.
   c. **Default SSL certificate:** Choose the SSL certificate that you created previously.
5. Click **Save.**

To add a listener for port 5443 to the ALB using the CLI:

1. Run the following command:
   ```
   # aws elbv2 create-listener \
   --load-balancer-arn <ALB ARN> \
   --protocol HTTPS \
   --port 5443 \
   ```

```
    --certificates CertificateArn=<Certificate ARN> \
    --default-actions Type=forward,TargetGroupArn=<Target group 5443 ARN>
```

Parameters:

<ALB ARN>: Use the value of ALB ARN recorded in the AWS worksheet.

<Certificate ARN>: Use the value of certificate ARN recorded in the AWS worksheet.

<Target group 5443 ARN>: Use the value for target group for port 5443 ARN recorded in the AWS worksheet.

Example input and output:

```
# aws elbv2 create-listener \
--load-balancer-arn arn:aws:elasticloadbalancing:eu-central-
1:115370811111:loadbalancer/app/srgdemo-alb/8718b24107ef591b \
--protocol HTTPS --port 5443 \
--certificates CertificateArn=arn:aws:acm:eu-central-
1:115370811111:certificate/691ec232-98ff-45ed-8e69-1d15c0447538 \
--default-actions
Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:eu-central-
1:115370811111:targetgroup/srgdemo-5443-tg/a096cb67c2f9144d
```

```
Listener for port 5443 description
```

```
{
   "Listeners":[
      {
         "ListenerArn":"arn:aws:elasticloadbalancing:eu-central-
1:115370811111:listener/app/srgdemo-alb/8718b24107ef591b/98e4aa47242b3d49",
         "LoadBalancerArn":"arn:aws:elasticloadbalancing:eu-central-
1:115370811111:loadbalancer/app/srgdemo-alb/8718b24107ef591b",
         "Port":5443,
         "Protocol":"HTTPS",
         "Certificates":[
            {
               "CertificateArn":"arn:aws:acm:eu-central-
1:115370811111:certificate/691ec232-98ff-45ed-8e69-1d15c0447538"
            }
         ],
         "SslPolicy":"ELBSecurityPolicy-2016-08",
         "DefaultActions":[
            {
               "Type":"forward",
               "TargetGroupArn":"arn:aws:elasticloadbalancing:eu-central-
1:115370811111:targetgroup/srgdemo-5443-tg/a096cb67c2f9144d",
               "ForwardConfig":{
```

```
                  "TargetGroups":[
                     {
                        "TargetGroupArn":"arn:aws:elasticloadbalancing:eu-
central-1:115370811111:targetgroup/srgdemo-5443-tg/a096cb67c2f9144d",
                        "Weight":1
                     }
                  ],
                  "TargetGroupStickinessConfig":{
                     "Enabled":false
                  }
               }
            }
         ]
      }
   ]
}
```

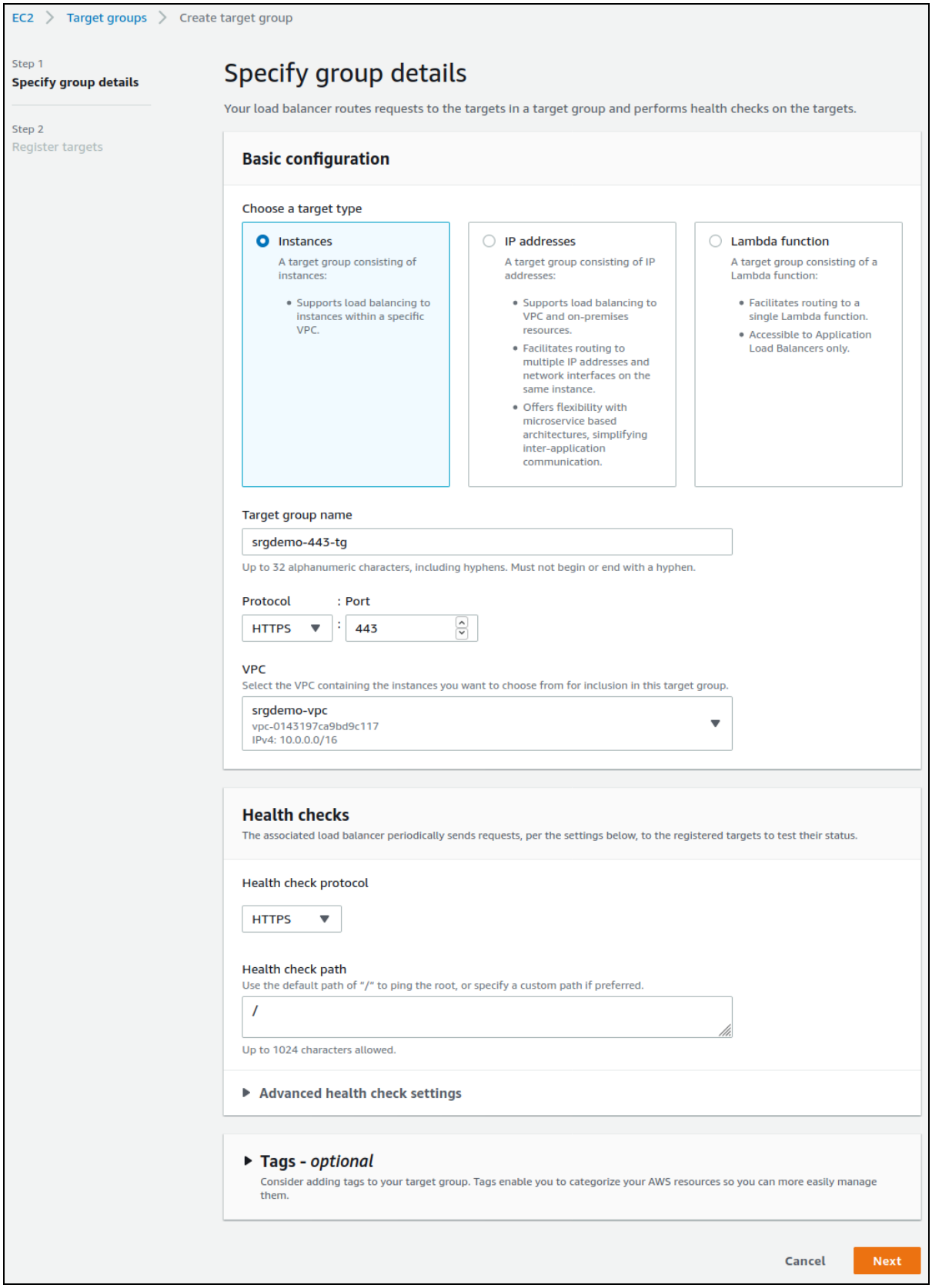
Next Step: Creating a Target Group for Ports 443, 32081, and 32080

Creating a Target Group for Ports 443, 32081, and 32080

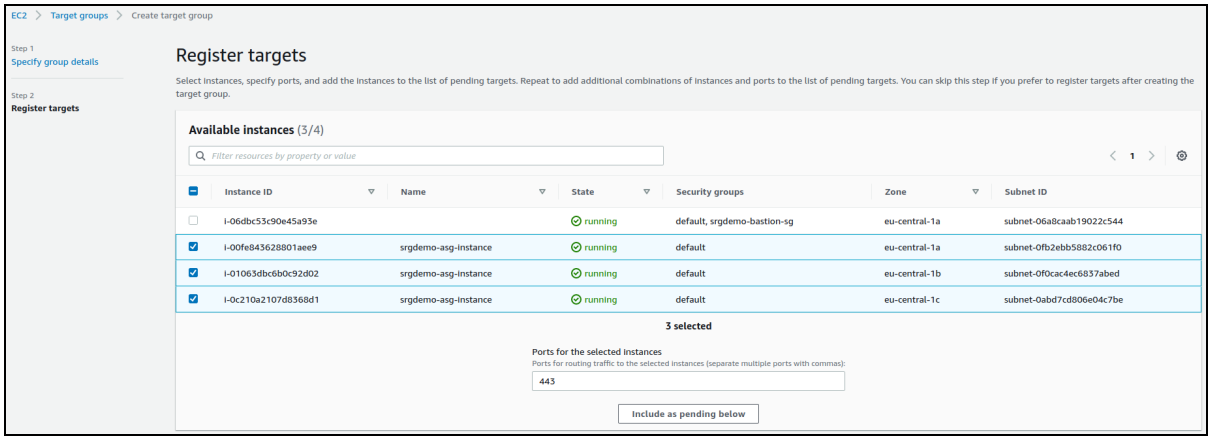
In order for deployed products to correctly operate, you must set up target groups, and then a listener, for ports 443, 32081 and 32080.

To create a target group for port 443 using the web UI:

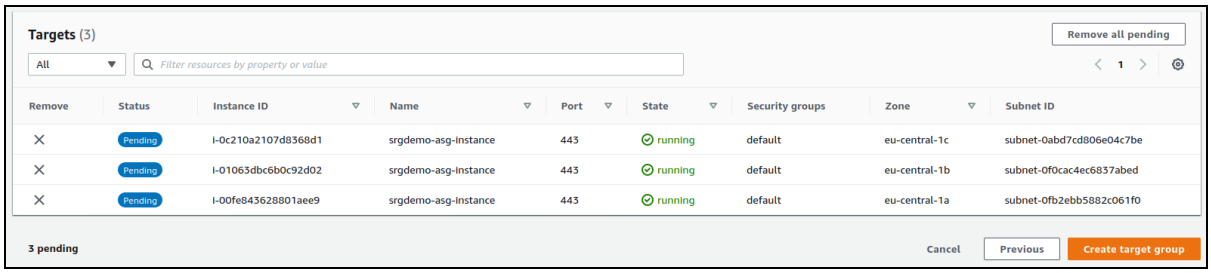
1. Using the Find Services search tool, locate and browse to the EC2 dashboard.
2. In the left navigation panel, under **Load Balancing**, click **Target Groups**.
3. On the **Target Groups** management page, click **Create target group.**
4. On the **Specify group details** page, specify values for the following:
   a. Under **Choose a target type,** select **Instances**.
   b. **Target group name:** Choose a descriptive name for easier identification; for example *srgdemo-443-tg*.
   c. **Protocol:** Change to *HTTPS*.
   d. **Port:** Select *443*.
   e. **VPC:** Select your VPC.
   f. **Tags:** (Optional) Add descriptive tags as desired.
   g. **Health Checks:** Under **Health check protocol,** select HTTPS. For path, specify */th/cmak*.


EC2 > Target groups > Create target group
Step 1
Specify group details
Step 2
Register targets
Specify group details
Your load balancer routes requests to the targets in a target group and performs health checks on the targets.
Basic configuration
Choose a target type
Instances
A target group consisting of instances:
Supports load balancing to instances within a specific VPC.
IP addresses
A target group consisting of IP addresses:
Supports load balancing to VPC and on-premises resources.
Facilitates routing to multiple IP addresses and network interfaces on the same instance.
Offers flexibility with microservice based architectures, simplifying inter-application communication.
Lambda function
A target group consisting of a Lambda function:
Facilitates routing to a single Lambda function.
Accessible to Application Load Balancers only.
Target group name
srgdemo-443-tg
Up to 32 alphanumeric characters, including hyphens. Must not begin or end with a hyphen.
Protocol : Port
HTTPS : 443
VPC
Select the VPC containing the instances you want to choose from for inclusion in this target group.
srgdemo-vpc
vpc-0143197ca9bd9c117
IPv4: 10.0.0.0/16
Health checks
The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.
Health check protocol
HTTPS
Health check path
Use the default path of "/" to ping the root, or specify a custom path if preferred.
/
Up to 1024 characters allowed.
Advanced health check settings
Tags - optional
Consider adding tags to your target group. Tags enable you to categorize your AWS resources so you can more easily manage them.
Cancel
Next

5. Click **Next**.
6. On the **Register Targets** page, set values for the following:
   a. **Available instances:** Select your instances; do not select the bastion.
   b. **Ports:** For the selected instances, use the value you retrieved previously and recorded in the AWS worksheet as the `Node Port for Port 443`.



7. Click **Include as pending below.** All marked instances will be added to the list of pending instances.



8. Click **Create target group.**
9. You will be redirected back to the target group management page. From the list, select the newly created target group. From the bottom of the page, note its ARN in the AWS worksheet.
10. Repeat Steps 1 through 9 for ports 32081.
11. Repeat Steps 1 through 9 for ports 32080.

To create a target group for port 443 using the CLI:

1. Run the following command:

```
# aws elbv2 create-target-group \
--name <Target group 443 Name> \
--protocol HTTPS \
--port 443 --vpc-id <VPC ID> \
--health-check-protocol HTTPS \
```

```
    --target-type instance
```

Where:

<Target group 443 Name>: Choose a descriptive name, such as srgdemo-443-tg, and record the value in the AWS worksheet.

<VPC ID>: The ID of your VPC as recorded in your AWS worksheet.

Example input and output:

```
# aws elbv2 create-target-group \
   --name srgdemo-443-tg --protocol HTTPS \
   --port 443 --vpc-id vpc-0143197ca9bd9c117 \
   --health-check-protocol HTTPS \
   --target-type instance
```

```
{
      "TargetGroups":[
         {
            "TargetGroupArn":"arn:aws:elasticloadbalancing:eu-central-
   1:115370811111:targetgroup/srgdemo-443-tg/6d30f1c7be588bb6",
            "TargetGroupName":"srgdemo-443-tg",
            "Protocol":"HTTPS",
            "Port":443,
            "VpcId":"vpc-0143197ca9bd9c117",
            "HealthCheckProtocol":"HTTPS",
            "HealthCheckPort":"traffic-port",
            "HealthCheckEnabled":true,
            "HealthCheckIntervalSeconds":30,
            "HealthCheckTimeoutSeconds":5,
            "HealthyThresholdCount":5,
            "UnhealthyThresholdCount":2,
            "HealthCheckPath":"/",
            "Matcher":{
               "HttpCode":"200"
            },
            "TargetType":"instance"
         }
      ]
   }
```

2. From the output, record the value of TargetGroupArn in your AWS worksheet.
3. Repeat Steps 1-2 for port 32081.
4. Repeat Steps 1-2 for port 32080.

Tagging the Target Group (CLI)

Optionally, you can tag any newly-created target group for easier identification.

To tag the target group for port 443 using the CLI:

1. Run the following command:
   ```
   # aws elbv2 add-tags \
   --resource-arns <Target group 443 ARN> \
   --tags Key=owner,Value=<owner>
   ```
2. Repeat Step 1 for port 32081.
3. Repeat Step 1 for port 32080.

Example:

```
# aws elbv2 add-tags \
--resource-arns arn:aws:elasticloadbalancing:eu-central-
1:115370811111:targetgroup/srgdemo-443-tg/c0684be94405b6b7 \
--tags Key=owner,Value=srgdemo
```

Next Step: Adding Targets to the Target Group for Ports 443, 32081, and 32080

Adding Targets to the Target Group for Ports 443, 32081, or 32080

To add targets to a target group:

1. Run the following command:
   ```
   aws elbv2 register-targets \
   --target-group-arn <Target group 443 ARN> \
   --targets Id="Instance 1 ID" \
   Id="Instance 2 ID" \
   Id="Instance 3 ID"
   ```
   Where:
   `<Instance x ID>`: Use the instance IDs you gathered for instances of the Auto Scaling group. Refer to AWS worksheet for values.
   `<Target group 443 ARN>`: Use the ARN of the target group you just created.

   `<Node port for 443>`: Use the node port number for 443 from the AWS worksheet.

   Example:

   ```
   aws elbv2 register-targets \
   --target-group-arn arn:aws:elasticloadbalancing:us-west-
   2:062373060138:targetgroup/rcchaincta-443-tg/4fcb60e16be46ada \
   ```

```
--targets Id="i-01eb0557e54d10e7b,Port=443" Id="i-
062e428614279bdee,Port=443" Id="i-0d0223498bfff57d0,Port=443"
```

2. Repeat Step 1 for port 32081.
3. Repeat Step 1 for port 32080.

Next Step: Add a Listener for Ports 443, 32081, and 32080 to the ALB

Adding a Listener for Ports 443, 32081, and 32080 to the ALB

This action will connect the ALB to the specified port through the target group. Then, incoming requests to the ALB on the port will be directed to the node instances.

To add a listener for port 443 using the web UI:

1. Using the Find Services search tool, locate and browse to the EC2 Dashboard.
2. In the left navigation panel, under **Load Balancing**, click **Load Balancers.**
3. From the list of load balancers, select your previously-created Application Load Balancer (ALB).
4. On the **Listeners** tab, click **Add Listener** and set values for the following:
   a. **Protocol: port:** Change to *HTTPS* and *443*.
   b. **Default action(s):** Select the action **Forward to...** then choose your target group for port 443.
   c. **Default SSL certificate:** Select the SSL certificate you have created previously.
5. Click **Save.**
6. Repeat Steps 1-5 for port 32081.
7. Repeat Steps 1-5 for port 32080.

To add a listener to port 443 or 32081 using the CLI:

1. Run the following command:
```
# aws elbv2 create-listener \
--load-balancer-arn <ALB ARN> \
--protocol HTTPS --port 443 \
--certificates CertificateArn=<Certificate ARN> \
--default-actionsType=forward,TargetGroupArn=<Target group 443 ARN>
```

Where:

`<ALB ARN>`: Use the ALB ARN recorded in the AWS worksheet.

`<Certificate ARN>`: Use the certifcate ARN recorded in the AWS worksheet.

`<Target group 443 ARN>`: is the target group for port 443 ARN recorded in the AWS worksheet.

Example input and output:

```
# aws elbv2 create-listener \
--load-balancer-arn arn:aws:elasticloadbalancing:eu-central-
1:115370811111:loadbalancer/app/srgdemo-alb/8718b24107ef591b \
--protocol HTTPS --port 443 --certificates CertificateArn=arn:aws:acm:eu-
central-1:115370811111:certificate/691ec232-98ff-45ed-8e69-1d15c0447538 \
--default-actions
Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:eu-central-
1:115370811111:targetgroup/srgdemo-443-tg/a096cb67c2f9144dv
```

```
{
   "Listeners":[
      {
         "ListenerArn":"arn:aws:elasticloadbalancing:eu-central-
1:115370811111:listener/app/srgdemo-alb/8718b24107ef591b/66915d0da2adb8a9",
         "LoadBalancerArn":"arn:aws:elasticloadbalancing:eu-central-
1:115370811111:loadbalancer/app/srgdemo-alb/8718b24107ef591b",
         "Port":443,
         "Protocol":"HTTPS",
         "Certificates":[
            {
               "CertificateArn":"arn:aws:acm:eu-central-
1:115370811111:certificate/691ec232-98ff-45ed-8e69-1d15c0447538"
            }
         ],
         "SslPolicy":"ELBSecurityPolicy-2016-08",
         "DefaultActions":[
            {
               "Type":"forward",
               "TargetGroupArn":"arn:aws:elasticloadbalancing:eu-central-
1:115370811111:targetgroup/srgdemo-443-tg/6d30f1c7be588bb6",
               "ForwardConfig":{
                  "TargetGroups":[
                     {
                        "TargetGroupArn":"arn:aws:elasticloadbalancing:eu-
central-1:115370811111:targetgroup/srgdemo-443-tg/6d30f1c7be588bb6",
                        "Weight":1
                     }
                  ],
                  "TargetGroupStickinessConfig":{
                     "Enabled":false
                  }
               }
```

```
            }
         ]
      }
   ]
}
```

2. Repeat the command for port 32081 for the schema registry.
3. Repeat the command for port 32028 for Fusion.

Next Step: You are now ready to deploy ArcSight Suite products using the CDF Management Portal. Proceed to Configuring the Deployed Capabilities.

However, if you are deploying ArcSight database, then proceed to Completing the Database Setup.

Completing the Database and Kafka Scheduler Setups

This section details the process for completing the database and Kafka Scheduler setups for both on-premises and cloud deployments.

- Gathering Certificates for the Kafka Scheduler Setup
- Enabling the Database to Receive SSL Connections
- Enabling the Database to Ingest Events from Transformation Hub

Gathering Certificates for the Kafka Scheduler Setup

The database and deployed capabilities need to establish a trusted connection. To do so, generate the key pair for the Kafka Scheduler.

This step is required even if you use non-SSL communication between the Kafka Scheduler and Transformation Hub, because the schema registry is always SSL-enabled.

1. Run these commands on your database node1 to generate the Kafka Scheduler private key file `kafkascheduler.key.pem` and the certificate signing request file `kafkascheduler.csr.pem`:

   ```
   cd <yourOwnCertPath>/
   ```

   If you installed using the ArcSight Platform Installer, the default location is `/opt/arcsight-db-tools/cert/`

```
openssl req -nodes -newkey rsa:2048 -keyout kafkascheduler.key.pem -out
kafkascheduler.csr.pem -subj "/C=US/ST=State/L=City/O=Company
Inc./OU=IT/CN=kafkascheduler"
```

2. Copy the certificate signing request `kafkascheduler.csr.pem` to your cluster, bastion host, or jump host.
3. Run the following commands on your cluster or your bastion host to sign the certificate signing request using your cluster RE certificate:

```
export CDF_APISERVER=$(kubectl get pods -n core -o custom-
columns=":metadata.name"| grep cdf-apiserver)
```

```
export PASSPHRASE=$(kubectl get secret vault-passphrase -n core -o json
2>/dev/null | jq -r '.data.passphrase')
```

```
export ENCRYPTED_ROOT_TOKEN=$(kubectl get secret vault-credential -n core
-o json 2>/dev/null | jq -r '.data."root.token"')
```

```
export VAULT_TOKEN=$(echo ${ENCRYPTED_ROOT_TOKEN} | openssl aes-256-cbc -
md sha256 -a -d -pass pass:"${PASSPHRASE}")
```

```
export COMMON_NAME=kafkascheduler
```

```
export CSR=$(cat ${COMMON_NAME}.csr.pem)
```

```
WRITE_RESPONSE=$(kubectl exec -it -n core ${CDF_APISERVER} -c cdf-
apiserver -- bash -c "VAULT_TOKEN=$VAULT_TOKEN vault write -tls-skip-
verify -format=json RE/sign/coretech csr=\"${CSR}\"") && \
```

```
echo "${WRITE_RESPONSE}" | jq -r ".data | .certificate" > ${COMMON_
NAME}.crt.pem && \
```

```
echo "${WRITE_RESPONSE}" | jq -r ".data | if .ca_chain then .ca_chain[]
else .issuing_ca end" > issue_ca.crt
```

4. Copy the RE signed certificate file `kafkascheduler.crt.pem` to database node1 <*yourOwnCertPath*>.
5. Copy the `issue_ca.crt` to database node1 <*yourOwnCertPath*>.

Enabling the Database to Receive SSL Connections

The following procedures are recommended for data privacy, but they are optional. Perform the first two procedures below on database node1.

Creating the Database Server Key and Certificate

Follow these steps to generate database CAs and certificates:

1. Log in to database node1 as root.
2. Change to your own certificates directory path:

```
cd <yourOwnCertPath>
```

For deployment with arcsight-platform-instsaller, the default location is /opt/arcsight-db-tools/cert/

3. Run this command to create a certificate authority (CA) for the database:

```
openssl req -newkey rsa:4096 -sha256 -keyform PEM -keyout generated-db-ca.key -x509 -days 3650 -outform PEM -out generated-db-ca.crt -subj "/C=US/ST=State/L=City/O=Company Inc./OU=IT/CN=Database/emailAddress=admin@microfocus.com" -nodes
```

4. Run this command to create the database server key:

```
openssl genrsa -out generated-db-server.key 4096
```

5. Create the database server certificate signing request by running the following command:

```
openssl req -new -key generated-db-server.key -out generated-db-server.csr -subj "/C=US/ST=State/L=City/O=Company Inc./OU=IT/CN=DatabaseServer/emailAddress=admin@microfocus.com" -nodes -sha256
```

6. Sign the Certificate Signing Request with self-signed CA by running the following command:

```
openssl x509 -req -in generated-db-server.csr -CA generated-db-ca.crt -CAkey generated-db-ca.key -CAcreateserial -extensions server -days 3650 -outform PEM -out generated-db-server.crt -sha256
```

Setting up the Database SSL Configuration

These steps will update the SSL configuration in the database.

1. Move the following files to database node1 <*yourOwnCertPath*> as root by running these commands:

   > This step is only required for a new database installation. You can skip this step if this is an upgrade and the files are already there.

   ```
   cd <yourOwnCertPath>/
   ls <yourOwnCertPath>/
   ```

   - The output should have the following files:
     - `generated-db-ca.crt`
     - `generated-db-server.crt`
     - `generated-db-server.key`
     - `generated-db-ca.key`
     - `generated-db-ca.srl`
     - `generated-db-server.csr`
     - `issue_ca.crt`
     - `kafkascheduler.crt.pem`
     - `kafkascheduler.key.pem`

2. For chained CAs, run the commands to split the CAs into individual files:

   ```
   cat issue_ca.crt | awk 'BEGIN {c=0;} /BEGIN CERT/{c++} { print > "issue_
   ca_part." c ".crt"}'
   ```

   ```
   chown -R dbadmin:dbadmin <yourOwnCertPath>
   ```

3. Run the following commands on database node1 to update the database SSL configuration:

   ```
   cd /opt/arcsight-db-tools
   ```

   ```
   ./db_ssl_setup --disable-ssl
   ```

If the attempt fails, drop the certificate manually by running the three commands below:

```
sudo su - dbadmin
```

```
vsql -U <dbadminuser> -w <dbadminpassword> -c "ALTER TLS CONFIGURATION
server CERTIFICATE NULL;"
```

```
vsql -U <dbadminuser> -w <dbadminpassword> -c "DROP CERTIFICATE IF EXISTS
server CASCADE;"
```

4. Enable database SSL for a single issue CA or chained issue CAs:
   - For a single issue CA, run this command:

     ```
     ./db_ssl_setup --enable-ssl --vertica-cert-path
     <yourOwnCertPath>/generated-db-server.crt --vertica-key-path
     <yourOwnCertPath>/generated-db-server.key --vertica-ca-path
     <yourOwnCertPath>/generated-db-ca.crt --client-ca-path
     <yourOwnCertPath>/issue_ca.crt
     ```

     -or-

   - For chained issue CAs, run this command, specifying each CA certificate in the chain one by one, separated by a comma in the `client-ca-path` parameter:

     ```
     ./db_ssl_setup --enable-ssl --vertica-cert-path
     <yourOwnCertPath>/generated-db-server.crt --vertica-key-path
     <yourOwnCertPath>/generated-db-server.key --vertica-ca-path
     <yourOwnCertPath>/generated-db-ca.crt --client-ca-path
     <yourOwnCertPath>/issue_ca_part.1.crt, <yourOwnCertPath>/issue_ca_
     part.2.crt[,...]
     ```

Configuring Deployed Capabilities to Use SSL for Database Connection

1. Log in to the CDF Management Portal.
2. Navigate to **Fusion** > **Database Configuration** > **Database Certificate(s)**.
3. Enable the **Use SSL for Database Connection** option.
4. Copy the complete contents of the file `generated-db-ca.crt`, created from the steps earlier, into the Database Certificate(s) text area.
5. Click **Save** to activate the configuration changes.

Enabling the Database to Ingest Events from Transformation Hub

The database uses an event consumer, the **Kafka scheduler**, to ingest events from Transformation Hub's Kafka component. Follow these steps when configuring the Kafka Scheduler for a new installation of the ArcSight Database:

Before you perform these steps, ensure that you have enabled SSL for the database. For information, see Enabling the Database to Receive SSL Connections.

1. Log in to the database node1 as root.
2. Change to the database tools directory:

   ```
   cd /opt/arcsight-db-tools/
   ```

3. Run the following command on database node1 to configure the schema registry server setting:

   ```
   ./schema_registry_setup <FQDN of ArcSight Platform Virtual IP for HA or
   single master node / Cloud: <DNS name for your cluster> >
   <yourOwnCertPath>/issue_ca.crt <yourOwnCertPath>/kafkascheduler.crt.pem
   <yourOwnCertPath>/kafkascheduler.key.pem
   ```

   You must provide the absolute path to the certificate.

4. Configure the SSL setup:

   On database node1, configure the SSL setting for the Kafka Scheduler by using one of the following methods, plain text or SSL:

   Plain Text (non-SSL)

   This method requires that you first enable **Allow plain text (non-TLS)** connections to Kafka. For more information, see Configuring the Deployed Capabilities.

   Run this command to disable SSL for the Kafka scheduler:

   ```
   ./sched_ssl_setup --disable-ssl
   ```

   SSL

   This method uses the crt and key files gathered or generated in earlier steps. The `issue_ca.crt` file should contain all chained CAs. For the Kafka scheduler to use SSL, run the following command:

   ```
   ./sched_ssl_setup --enable-ssl --sched-cert-path
   <yourOwnCertPath>/kafkascheduler.crt.pem --sched-key-path
   ```

```
<yourOwnCertPath>/kafkascheduler.key.pem --vertica-ca-key
<yourOwnCertPath>/generated-db-ca.key --vertica-ca-path
<yourOwnCertPath>/generated-db-ca.crt --kafka-ca-path
<yourOwnCertPath>/issue_ca.crt
```

5. Run this command on database node1 to create the Kafka Scheduler:
   - If the Kafka Scheduler was configured to use plain-text in the previous step, use port 9092:

     ```
     ./kafka_scheduler create <th_kafka_nodename1>:9092
     ```

   - If SSL was enabled for the Kafka Scheduler in the previous step, use port 9093:

     ```
     ./kafka_scheduler create <th_kafka_nodename1>:9093
     ```

6. Start the Kafka Scheduler and checker on database node1:

   ```
   ./kafka_scheduler start
   ./kafka_scheduler messages
   ./kafka_scheduler events
   ```

7. Continue to the post-deployment section.

The dbadmin user has access to all the certificate/keys files.

Applying the CDF 2021.05 log4j Hotfix

Some deployments of, and upgrades to, CDF 2021.05/arcsight-platform-installer-22.1.x.x.zip require application of a hotfix to remediate the log4j vulnerability, which was discovered in 2021. The hotfix will upgrade IDM for CDF to use log4j 2.17.1, to prevent exploitation of the log4j vulnerabilities (CVE-2021-44228, CVE-2021-45046, CVE-2021-45105, CVE-2021-44832). The hotfix should be applied after an upgrade.

The hotfix applies to the following types of installations and upgrades:

- Any on-premises manual installation of 22.1.x or any on-premises manual upgrade to 22.1.1. (A manual installation or upgrade is one that does not use the ArcSight Installer.)

The CDF 2021.05 hotfix will be automatically applied during any on-premises installation or upgrade using the ArcSight Installer and this procedure can be skipped.

- Any CDF fresh installation or upgrade on AWS.
- Any CDF fresh installation or upgrade on Azure.

The log4j remediation hotfix does **NOT** apply to on-premises installations or upgrades performed automatically using the ArcSight Installer, as the hotfix is applied automatically by the installer. For such installations or upgrades these procedures can be skipped.

Hotfix File

The hotfix file is named arcsight-idm-hf-22.1.0-2.zip.

1. Get the file:
   - For a manual on-premises deployment/upgrade, the hotfix is bundled in `/<download_folder>/arcsight-platform-installer-22.1.x.x/installers/hotfix`.
   - For an AWS or Azure deployment upgrade, obtain the file from the *on-premises* installation file directory at `/<download_folder>/arcsight-platform-installer-22.1.x.x/installers/hotfix`.
2. Copy the file:
   - For manual on-premises, copy the file to your master node.
   - For AWS, copy the file to your bastion.
   - For Azure, copy the file to your jump host.
3. Unzip the hotfix file. In the unzipped folder, run the following command with the '-e' argument (values: onprem, azure, aws) to apply the latest image.

```
# ./hotfix.sh -e <YOUR_ENV>
```

Verifying the Hotfix

1. Check the pod status by running the following command. It should be 'Running' as 2/2.

```
# kubectl get pods -A | grep idm
```

2. Check the image version by running the following command.

```
# kubectl get deployment/itom-idm -n core -o yaml | grep itom-idm:1.32.1-343
```

It should display as below:

```
image: <image-registry-url>/<org-name>/itom-idm:1.32.1-343
```

Rolling Back to the Previous Version

To roll back itom-idm to the previous version, run the following roll back commands :

```
# kubectl delete -f /tmp/cdf-itom-idm.yaml
```

```
# kubectl create -f /tmp/cdf-itom-idm.yaml
```

Enabling Pod Logs in AWS

You can enable the ArcSight products application (pod) logs in AWS, which includes a cluster logging functionality called Fluentd.

To enable Fluentd in AWS, execute this command:

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/container-insights-monitoring/quickstart/cwagent-fluentd-quickstart.yaml | sed "s/{{cluster_name}}/cluster-name/;s/{{region_name}}/cluster-region/" | kubectl apply -f -
```

In this example, example values are used for `cluster name` and `cluster region name`.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/container-insights-monitoring/quickstart/cwagent-fluentd-quickstart.yaml | sed "s/{{cluster_name}}/test_cluster/;s/{{region_name}}/us-west-2/" | kubectl apply -f -
```

For more information, see the AWS documentation here.

Using AWS Configuration Worksheets

During the setup and configuration of your AWS deployment environment, use the following worksheets.

- "AWS Infrastructure Settings" below
- "Subnets" on the next page
- "Security Groups" on the next page
- System and Environment Settings

AWS Infrastructure Settings	
Region	
VPC ID	
VPC CIDR	

VPC Name	
Cluster Name	
Public IP ID	
Public IP	
Internet GW ID	
NAT GW ID	
DNS Enabled (Y/N)	
Hostname Resolution Enabled (Y/N)	

Subnets

Availability Zone	*CIDR*	*Name*	*ID*	*Tagged Load Balancing *Y/N/NA*

Security Groups

Bastion Security Group Name	
Bastion Security Group ID	
Intra VPC Security Group Name	
Intra VPC Security Group ID	
IAM Roles	
EKS Role Name	
EKS Role ARN	

EKS Instance Profile Name	
EKS Instance Profile ARN	
Workernodes Role Name	
Workernodes Role ARN	
Workernodes Instance Profile Name	
Workernodes Instance Profile ARN	

System and Environment Settings

Bastion	Kubernetes Version	
	Key Pair Name	
	Key Pair Fingerprint	
	Image ID	
	Instance Type	
	Instance ID	
	Public IP Address	
EFS	EFS Name	
	FileSystemID	
	Filesystem FQDN	
	Mount Target 1 ID	
	Mount Target 2 ID	
	Mount Target 3 ID	
	Parent Folder Name	
EKS	Cluster ARN	
Worker Nodes	Launch Configuration Name	
	Launch Config AMI ID	
	Instance Type	
	Autoscaling Group Name	
	Instance IDs	

ECR Registry Upload	Organization Name	
Route 53 Records	Name In Hosted Zone	
	Hosted Zone Name	
	Hosted Zone ID	
	RecordSet Name	
	Certificate ARN	
Networking	ALB Name	
	ALB ARN	
	ALB DNS Name	
	ALB Canonical Hosted Zone ID	
	Node Port for 3000	
	Target Group 3000 Name	
	Target Group 3000 ARN	
	Target Group 5443 Name	
	Target Group 5443 ARN	
	Target Group 443 Name	
	Target Group 443 ARN	

Setting Up Your Azure Deployment Architecture

This section explains how to set up your deployment architecture for ArcSight capabilities that run on the Microsoft Azure cloud platform. You deploy each ArcSight containerized application in the Azure environment created by Micro Focus Container Deployment Foundation (CDF).

As shown, the Kubernetes nodes run as virtual machines in the Kubernetes node pool network security group (NSG) and under the Azure Kubernetes service. Secure administrator access to the nodes is from the jump host, which is included in the management NSG.

Checklist: Planning to Deploy ArcSight Capabilities on Azure

In order to perform the deployment of ArcSight capabilities on Azure, you need the following:

- An active Azure subscription.
- Permissions to create:
  - Resource groups.
  - An Azure Container Registry (ACR).
  - A service principal.
  - Azure virtual machines (VMs).
  - Storage disks.
- OWNER rights on the created resource group.
- *If using a NetApp NFS (network file system):* The Azure subscription needs to be granted access to the Azure NetApp Files service (details are described in the procedure).

After you have installed and configured an Azure jump host, you can run all Azure Cloud Shell (`az cli`) commands from the jump host instead of the Azure Cloud Shell.

The complete process of deploying Azure includes the following broad steps. You can perform most steps using either the Azure Portal or through the Azure Cloud Shell, and each method is explained (where possible).

	Task	See
❑	1. Create the Azure Container Registry (ACR) and the Azure resource group which will contain the deployment resources.	"Preparing the Azure Container Registry and Resource Group" on page 284
❑	2. Prepare the Azure Kubernetes Service (AKS).	"Preparing the Azure Kubernetes Service" on page 286
❑	3. Prepare the NFS (Network File System) subnet.	"Preparing the Subnet for the NFS Server and Jump Host" on page 290
❑	4. Create the jump host virtual machine and configure the jump host for connectivity to the cluster.	"Preparing the Jump Host Virtual Machine" on page 292
❑	5. Prepare the Network File System (NFS) server.	"Configuring the NFS Server" on page 300
❑	6. Create and configure the volumes.	Creating and Configuring the Volumes

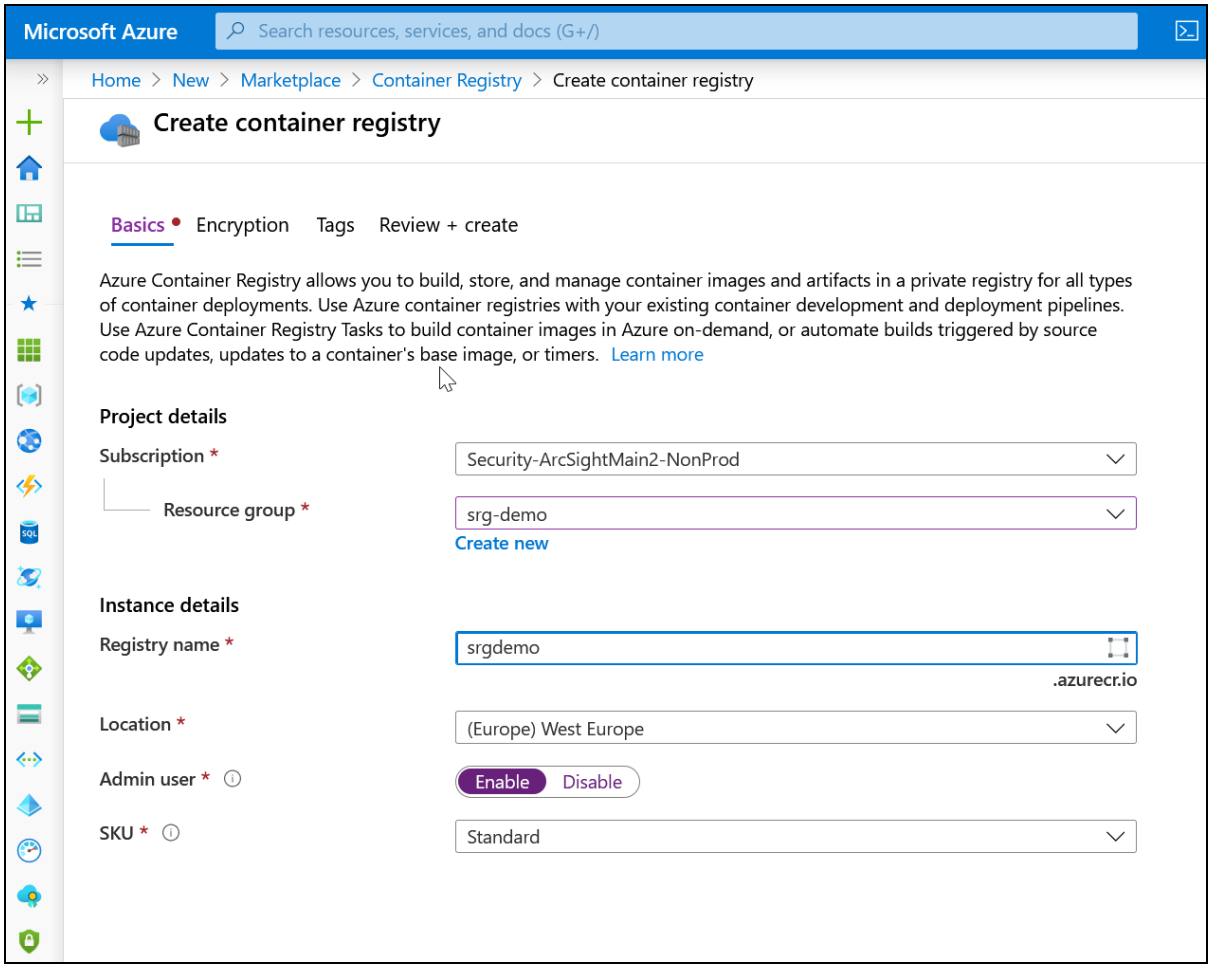
❑	7. Create and attach the data disk to nodes.	Creating and Attaching the Data Disk to Nodes
❑	8. Prepare the private DNS zone.	Preparing the Private DNS Zone
❑	9. Label nodes in your cluster to indicate their functionality.	"Labeling Azure Kubernetes Service Nodes" on page 320
❑	10. Upload product images to the Azure Container Registry for installation.	"Uploading Product Images" on page 322
❑	11. Install the CDF installer script and install CDF to Azure.	"Installing CDF" on page 338
❑	12. Configure and deploy the Kubernetes cluster of AKS nodes.	"Configuring the Kubernetes Cluster" on page 351
❑	13. Patch and configure your load balancing capability with the latest updates.	"Patching the Load Balancer" on page 340
❑	14. Open the CDF management portal.	Opening the Management Portal
❑	15. Configure and deploy, using the CDF Management Portal, Transformation Hub to run in the CDF-managed Kubernetes cluster.	"Deploying ArcSight Products" on page 363
❑	16. Configure the Management Center (ArcMC) to recognize and manage the platform.	"Managing Your ArcSight Infrastructure with ArcMC" on page 864
❑	17. Configure your SmartConnectors and Collectors as producers of events into Transformation Hub, as well as configure event Consumers such as Logger and ESM.	"Integrating the Platform Into Your Environment" on page 392
❑	18. Apply the hotfix to remediate the log4j vulnerability	"Applying the CDF 2021.05 log4j Hotfix" on page 570
❑	19. Get the latest security fixes and enhancements	Upgrading to 22.1.2

Preparing the Azure Container Registry and Resource Group

Portal

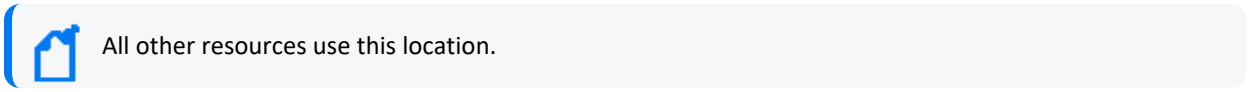
To prepare the ACR (Azure Container Registry) and resource group on the Azure Portal:

1. Log in to the Azure portal at (`https://portal.azure.com`)
2. Select an active Azure subscription and click **Create a Resource.**
3. In the search box (case-insensitive), specify *Container Registry* and click **create**.

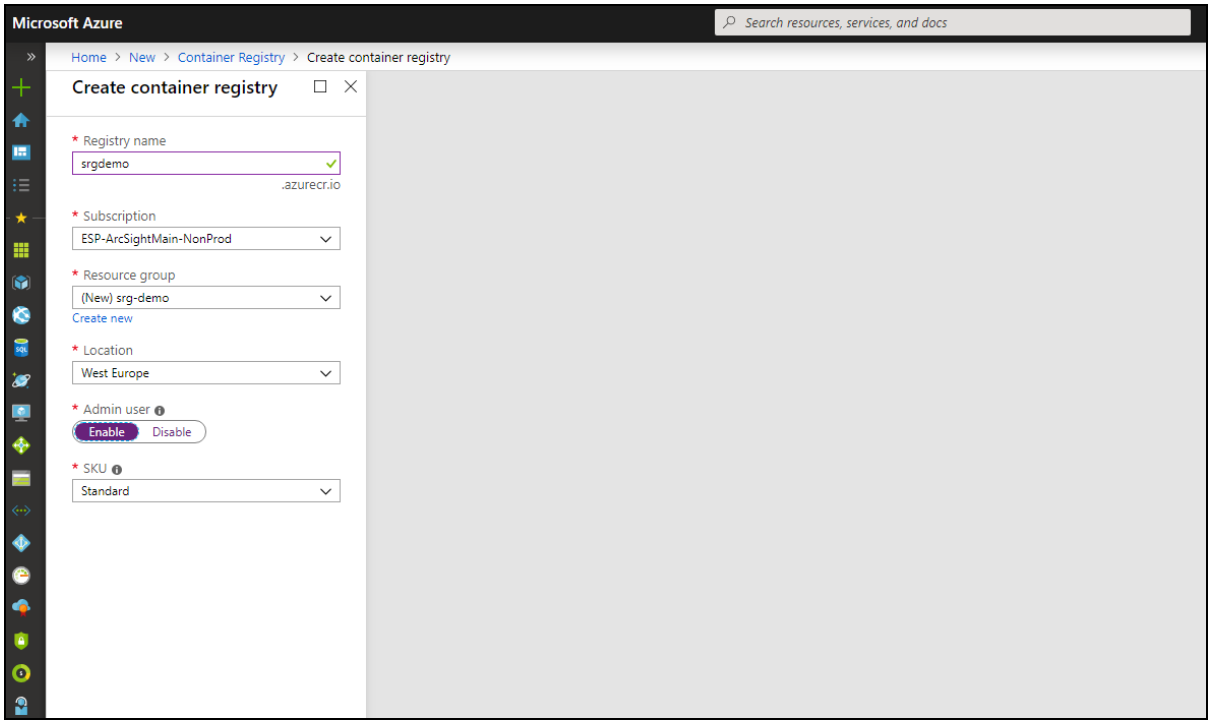


Later in this guide, the steps above will be referred to as "Create a resource of type <some resource>." For these references, take the steps shown above to create the resource of the specified type.

4. Specify a value for **Registry Name**. Note this name for later reference.
5. For **Resource group**, click **Create New**, and in **Name**, specify a resource group name.
6. For **Location**, select a location with enough resources for your deployment.



7. For **Admin user**, select *Enable*.
8. For **SKU**, choose the value appropriate for your ACR.



9. Click **Create** to create the Azure Container Registry.

CLI

To prepare the ACR and Resource Group using the Azure Cloud Shell:

You must have *create an ACR* and *create a resource group* permissions to use this feature.

1. Open the Azure Cloud Shell (on the top right of the Azure Portal page). If necessary, confirm the creation of user storage.
2. Create the resource group by running the command:

```
az group create --name <RESOURCE GROUP> \
--location <LOCATION>
```

Where:

<RESOURCE GROUP> is your group name, which will be used later for all other resources

<LOCATION> is the location where resource group will be created. To get a list of all locations, run the command:

```
az account list-locations | jq ".[] | .name"
```

For example:

```
az group create \
--name srg-demo \
--location westeurope
```

3. Check the az command response. It should contain the text:

```
"provisioningState": "Succeeded"
```

4. Create the Azure Container Registry (ACR) by running the command:

```
az acr create -n <your ACR name> -g <your resource group name>\
--admin-enabled "true" --sku "Standard"
```

For example:

```
az acr create \
-n srgdemoACR \
-g srg-demo \
--admin-enabled "true" \
--sku "Standard"
```

5. Check the az command response. It should contain the text:

```
"provisioningState": "Succeeded"
```

In succeeding procedures, the az command response should contain the same text: "provisioningState": "Succeeded"

Next Step: "Preparing the Azure Kubernetes Service" below

Preparing the Azure Kubernetes Service

Preparation of the Azure Kubernetes Service (AKS) includes these sub-steps.

- "Creating the Service Principal ID for Kubernetes" on the next page
- "Preparing the Virtual Network and AKS Subnet" on the next page
- "Creating the Azure Kubernetes Service (AKS)" on page 288

Creating the Service Principal ID for Kubernetes

Required permissions: create service principal

To create the service principal ID:

Run this command in the Azure Cloud Shell:

```
az ad sp create-for-rbac -n "PRINCIPAL ID NAME" --skip-assignment
```

For example:
`az ad sp create-for-rbac -n srgdemo-service-principal --skip-assignment`

Example results:

```
{
   "appId":"52f25b66-2700-474d-a2a0-016f0b149e22",
   "displayName":"srgdemo-service-principal",
   "name":"http://srgdemo-service-principal",
   "password":"bf47aa85-9578-4d61-a8e9-ffafe5a1e22b",
   "tenant":"6002e264-31f7-43d3-a51e-9ed1ba9ca689"
}
```

Note the values for `password` and `appID`. These values will be used in the next step.

Preparing the Virtual Network and AKS Subnet

Now you can prepare a virtual network with custom ranges and subnet for AKS. If you already have an existing virtual network with a subnet for AKS, you can skip this procedure.

Place all the created resources in the same virtual network to prevent performance issues caused by network latency. These resources include resource group, AKS cluster, jump host, and Azure NetApp Files (NFS).

To create the virtual network:

Run the following command:

```
az network vnet create \
-g <RESOURCE_GROUP> \
-n <VNET_NAME> \
--address-prefix <VNET_CIDR> \
--subnet-name <SUBNET_NAME> \
--subnet-prefix <SUBNET_CIDR>
```

Where:

`<RESOURCE_GROUP>`: the name of the resource group created in step 1.

`<VNET_NAME>`: The assigned name of this virtual network.

`<VNET_CIDR>`: The CIDR notation for this virtual network. For example, 10.1.0.0/16.

`<SUBNET_NAME>`: Name for this subnet for AKS.

`<SUBNET_CIDR>`: The CIDR notation for this subnet. For example, 10.1.1.0/24.

For example, this would create a virtual network demo-vnet, in resource group srg-demo, with range 10.1.0.0/16 and subnet aks-subnet with subnet range 10.1.1.0/24 :

```
az network vnet create \
-g srg-demo \
-n demo-vnet \
--address-prefix 10.1.0.0/16 \
--subnet-name aks-subnet \
--subnet-prefix 10.1.1.0/24
```

Creating the Azure Kubernetes Service (AKS)

Required permissions: create Azure Kubernetes service; the user must be the OWNER of the resource group

To create the AKS:

1. Get the subnet ID which you want to use for AKS and store it to an environment variable:

   ```
   SUBNET_ID=$(az network vnet subnet show \
   --resource-group <RESOURCE_GROUP> \
   --vnet-name <VNET_NAME> \
   --name <SUBNET_NAME> \
   --query id -o tsv)
   ```

   For example, to use the virtual network `demo-vnet` from the resource group `srg-demo` and subnet `aks-subnet`, you would run the following command:

   ```
   SUBNET_ID=$(az network vnet subnet show --resource-group srg-demo --vnet-
   name demo-vnet --name aks-subnet --query id -o tsv)
   ```

2. Create the AKS in this subnet by running this command:

   ```
   az aks create \
   -g <RESOURCE GROUP> \
   -n <AKS NAME> \
   -c <NUMBER OF NODES> \
   ```

```
--kubernetes-version <Kubernetes version> \
--generate-ssh-keys \
--node-vm-size <VM SIZE> \
--vm-set-type VirtualMachineScaleSets \
--service-principal "<SP APP ID>" \
--client-secret "<SP PASSWORD>" \
--load-balancer-sku basic \
--vnet-subnet-id $SUBNET_ID \
```

where:

`<RESOURCE GROUP>` is your main resource group.

`<AKS NAME>` is your AKS resource name.

`<NUMBER OF NODES>` is the number of worker nodes.

`<KUBERNETES VERSION>` is the version of the Kubernetes cluster you want to create, which your CDF version must support. You must be OWNER (or be OWNER of resource group) to be able to assign the virtual network to the AKS. Use the command `az aks get-versions --location <LOCATION>` to get the supported version number.

`<VM SIZE>` for example, `Standard_D4s_v3`.

For a production cluster, do not use a size less than `Standard_D8s_v3` with less than 32 GB of RAM.

For a list of VM sizes, run the command:

```
az vm list-sizes -l <LOCATION> | jq ".[] | .name"
```

For a list of supported Kubernetes versions on Azure, run the command:

```
az aks get-versions --location <LOCATION> | jq -r ".orchestrators | .
[].orchestratorVersion"
```

`<SP APP ID>` and `<SP_PASSWORD>` is the appID and password from the creation of the service principal ID.

Example az aks creatr command:

```
az aks create \
-g "srg-demo" \
-n "srg-demo-aks" \
-c "3" \
--kubernetes-version 1.20 \
--generate-ssh-keys \
--node-vm-size "Standard_D4s_v3" \
```

```
--vm-set-type VirtualMachineScaleSets \
--service-principal "52f25b66-2700-474d-a2a0-016f0b149e22" \
--client-secret "bf47aa85-9578-4d61-a8e9-ffafe5a1e22b" \
--load-balancer-sku basic \
--vnet-subnet-id $SUBNET_ID \
```

The `az aks create` command generates private and public keys, which are stored in the `~/.ssh` directory. Download `id_rsa` to a secure network location. Later, you will upload the `id_rsa` to the jump host. Azure uses it to connect to AKS nodes from the jump host.

Next Step: Prepare the Subnet for the NFS Server and Jump Host

Preparing the Subnet for the NFS Server and Jump Host

You can skip this step if you already have a subnet prepared for the installation of AKS.

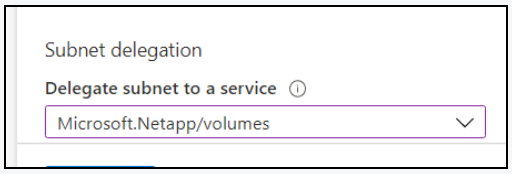
In this section, you will prepare the subnets for the NFS server and for the jump host. Place all of the resources in the same virtual network to prevent performance issues caused by network latency. Such resources include resource group, AKS cluster, jumphost, Azure NetApp files, and so on.

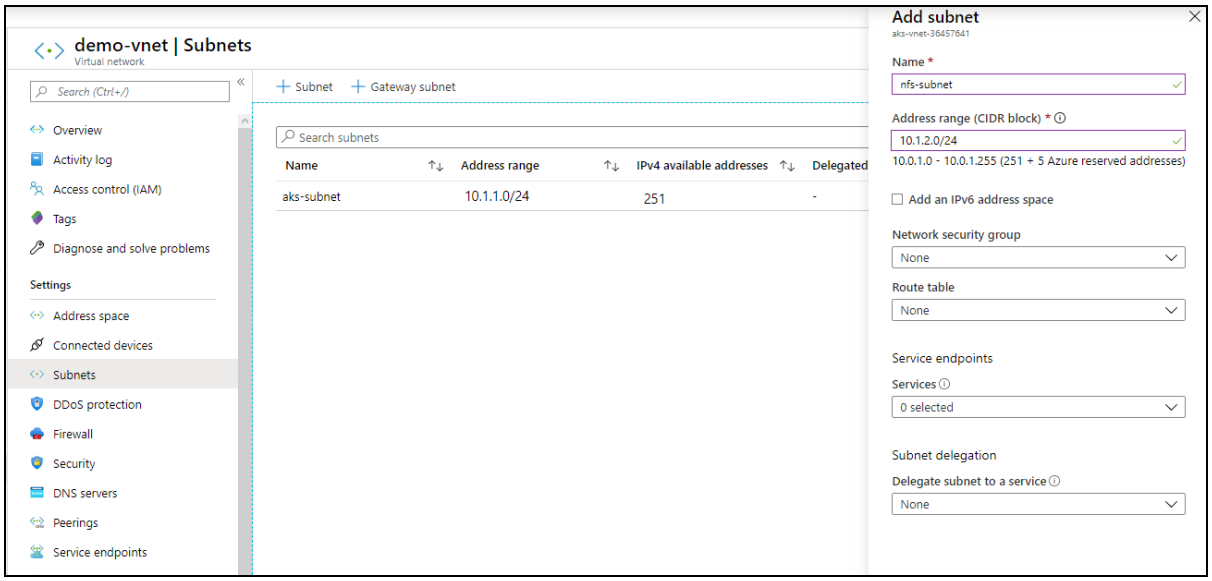
Portal

To prepare the subnet for the NFS server and jump host using the Azure Portal:

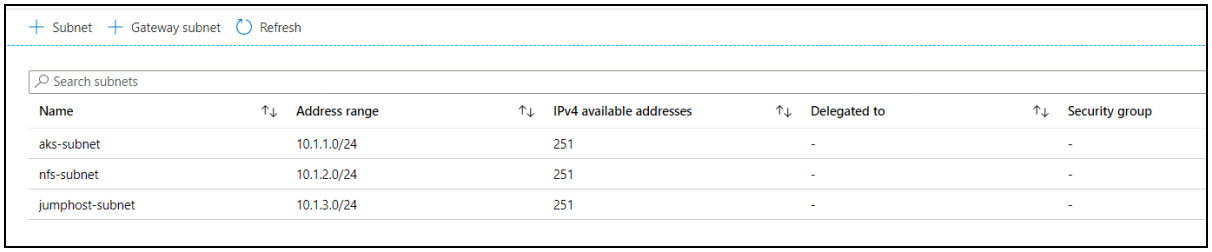
1. Open the virtual network used for the installation AKS. In our example, this is `demo-vnet` in resource group `srg-demo`.
2. On the **Virtual network** page, under **Settings**, select **Subnets**.
3. Click **+ Subnet.**
4. In **Name**, specify *nfs-subnet.*
5. In **Address range,** specify an address range based on the IP assigned by Azure. In our example, `aks-subnet` uses the address range `10.1.1.0/24`, so we use `10.1.2.0/24`.

If you plan to use NetApp as an NFS service, under **Subnet delegation**, the subnet must be delegated to `Microsoft.NetApp/volumes`.





6. Click **OK** to create the subnet.



7. Repeat steps 3 and 4 to create a subnet for the jump host.
   - For name, use *jumphost-subnet.*
   - For address range, specify an appropriate range. For example, 10.1.3.0/24.

CLI

To prepare the subnet for the NFS server and jump host using the Azure Cloud Shell:

Required permissions: create subnets inside the AKS virtual network

1. Create the NFS subnet by running the command:

```
az network vnet subnet create \
--address-prefixes <ADDRESS PREFIX> \
--name nfs-subnet \
-g <RESOURCE GROUP> \
--vnet-name <VIRTUAL NETWORK>
```

where:

- `<VIRTUAL NETWORK>` is the virtual network name where you want to create the subnet. We will use the virtual network created earlier.
- `<RESOURCE GROUP>` is the resource group where the virtual network is located.

If you are using NetApp as an NFS service, add the argument `--delegations Microsoft.NetApp/volumes` to the above command.

For example, the following command would create nfs-subnet in virtual network demo-vnet, from resource group srg- demo with range 10.1.2.0/24 :
`az network vnet subnet create --address-prefixes 10.1.2.0/24 --name nfs-subnet -g srg-demo --vnet-name demo-vnet`

2. Create the jump host subnet by running a similar command to the one in Step 1, but with a different name and address prefix.

For example, the following command would create the jumphost-subnet, in virtual network demo- vnet from resource group srg- demo with range 10.1.3.0/24
`az network vnet subnet create --address-prefixes 10.1.3.0/24 --name jumphost-subnet -g srg-demo --vnet-name demo-vnet`

Next Step: "Preparing the Jump Host Virtual Machine" below

Preparing the Jump Host Virtual Machine

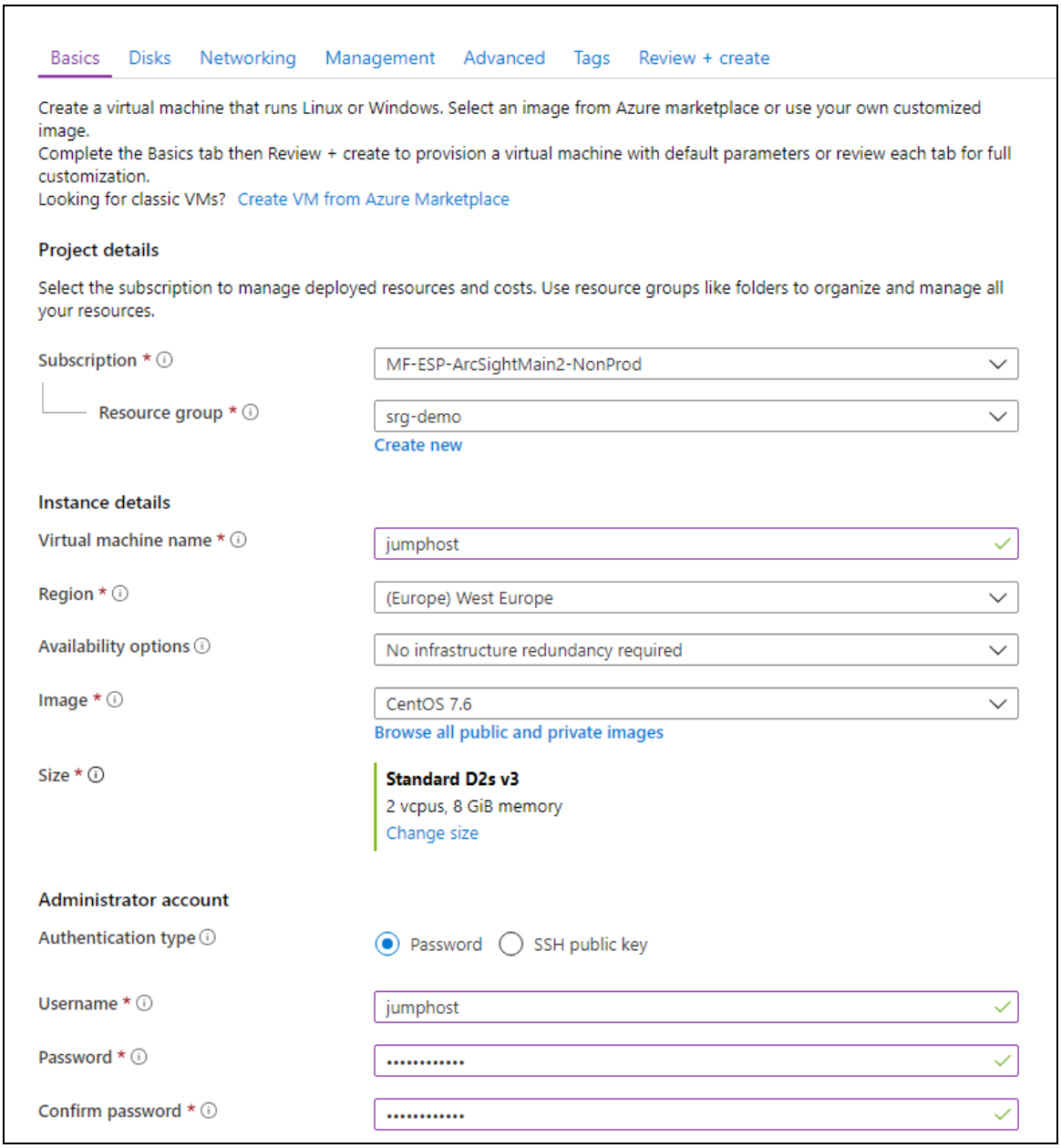
Follow the procedures in this section to prepare the jump host VM.

Portal

To prepare the jump host VM using the Azure Portal:

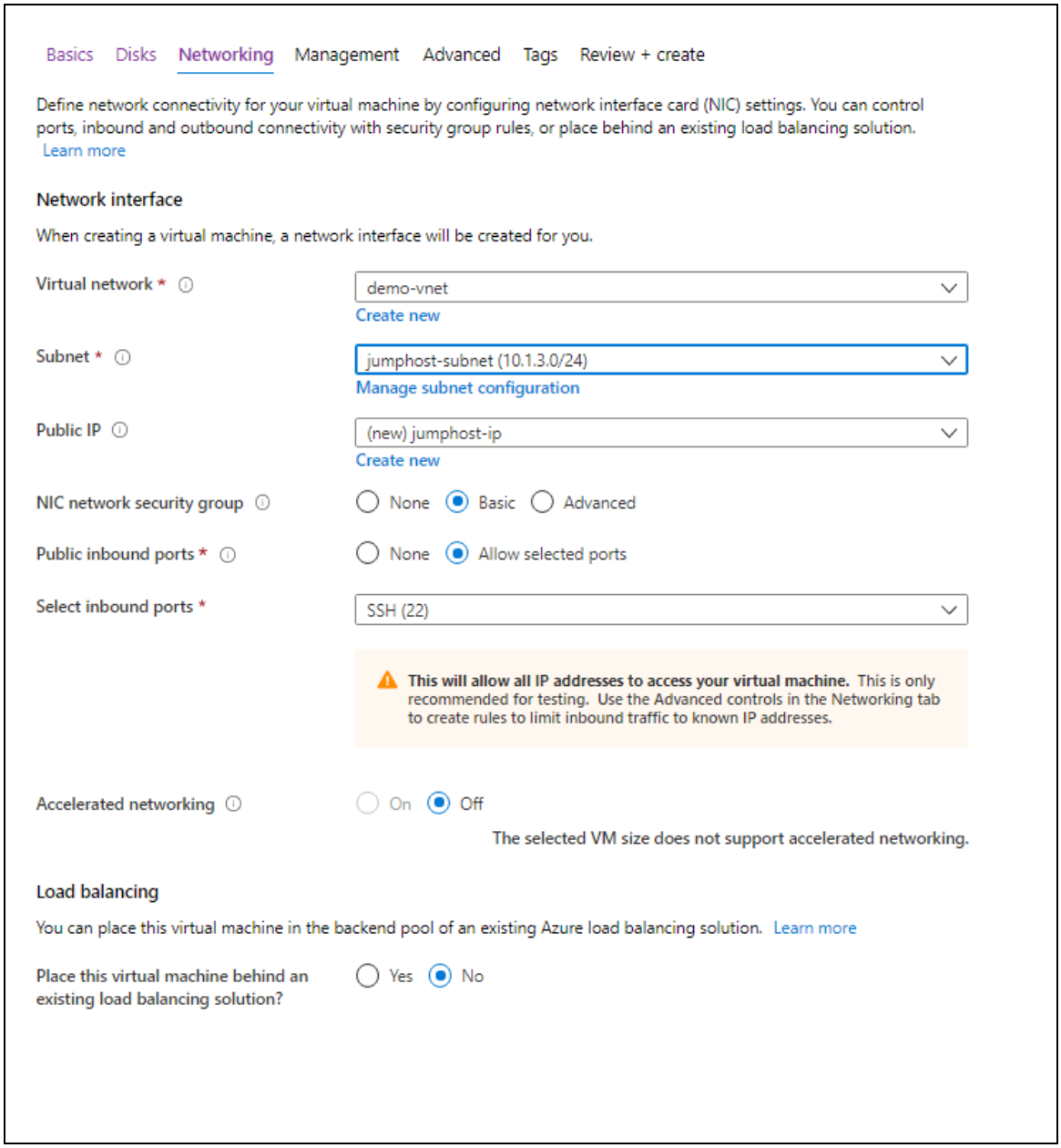
1. Create resource of type *CentOS-based* and specify these values:
   - For **Resource group**, use the resource group you created for the ACR.
   - In **Virtual machine name**, specify a VM name.
   - For **Size**, leave at the default value.
   - Set the **Authentication type** to your preferences and supply the Administration account

details accordingly. In our examples, we use the username/password authentication.



2. Click **Next: Disks.** No actions need to be taken on this page. Click **Next: Networking,** and then specify the following values:
   - For **Virtual network**, select the virtual network you created previously (its name has the format *demo-vnet*).
   - For **Subnet**, select *jumphost-subnet.*
   - For **NIC network security group,** select *Basic* and *Allow SSH port to connect.*

For optimal security, remove this rule when the jump host is not needed, or add more strict rules such IP filtering.



3. Click **Review + create.**
4. After validation, click **Create** to start the deployment.
5. When the deployment completes, browse to the VM overview and note the jump host's public IP address.

CLI

To prepare the Jump Host VM using the Azure Cloud Shell:

Required Permissions: create security groups, network interfaces, public IPs, and CentOS-based virtual machines

1. Set your main resource group name to an environment variable:
   `$RESOURCE_GROUP=<your resource group name>`

   For example:
   `RESOURCE_GROUP=srg-demo`
2. Create a network security group for the jump host by running the following command:
   `az network nsg create -g $RESOURCE_GROUP -n jumphost-nsg`

   For example:
   `az network nsg create -g srg-demo -n jumphost-nsg`
3. Open the SSH port (if needed) by running these commands:
   `az network nsg rule create -g $RESOURCE_GROUP -n ssh --nsg-nam jumphost-nsg --priority 1000 --destination-port-ranges 22`

   Keep in mind the security risks of opening ports to the Internet and consider using a VPN, restricting access to the allowed source IP address, or limiting the amount of time that this port remains available.
4. Prepare the jump host public IP:
   `az network public-ip create -n jumphost-PublicIP -g $RESOURCE_GROUP --allocation-method "Static" --sku "Standard"`
5. Get the subnet ID and store it in an environment variable for later usage.
   `SUBNET_ID=$(az network vnet list -g $RESOURCE_GROUP | jq -r '.[] | select (.name == "<your vnet name>") | .subnets[] | select(.name == "<your jumphost subnet>") | .id')`

   For example:
   `SUBNET_ID=$(az network vnet list -g $RESOURCE_GROUP | jq -r '.[] | select (.name == "<your vnet name>") | .subnets[] | select(.name == "<your jumphost subnet>") | .id')`
6. Create the network interface `jumphost-VMNic` in your resource group with public IP `jumphost-publicIP` with network security group `jumphost-nsg` by running the following command:
   `az network nic create --name "jumphost-VMNic" --resource-group $RESOURCE_GROUP --public-ip-address "jumphost-PublicIP" --ip-forwarding "true" --network-security-group "jumphost-nsg" --subnet $SUBNET_ID`

7. Create the jump host VM by running the following command:
   ```
   az vm create --name "jumphost" --resource-group $RESOURCE_GROUP --image "OpenLogic:CentOS:7.7:latest" --size "Standard_D4s_v3" --public-ip-address-allocation "static" --nics "jumphost-VMNic" --admin-username jumphost --admin-password myStrongPassword@!123
   ```
   Where:
   - Size might be a smaller value. To get a list of supported sizes, run the command:
     ```
     az vm list-sizes -l <LOCATION> | jq ".[] | .name"
     ```
   - Image can be any supported CentOS. To get a list of CentOS images, run the command:
     ```
     az vm image list -l <LOCATION> -f CentOS --all
     ```

   Example result:

   ```
   {- Finished ..
   ```

   ```
   "fqdns": "",
   ```

   ```
   "id": "/subscriptions/af379ae8-90b3-4368-8fe7-b6a55ab17720/resourceGroups/srg-demo/providers/Microsoft.Compute/virtualMachines/jumphost",
   ```

   ```
   "location": "westeurope",
   ```

   ```
   "macAddress": "00-0D-3A-BD-08-42",
   ```

   ```
   "powerState": "VM running",
   ```

   ```
   "privateIpAddress": "10.0.2.4",
   ```

   ```
   "publicIpAddress": "51.124.17.183",
   ```

   ```
   "resourceGroup": "srg-demo",
   ```

   ```
   "zones": ""
   ```

   ```
   }
   ```

Use the VM Public IP address to permit SSH access to the jump host from outside. (The SSH port needs to be open if access is permitted from outside.)

Configuring the Jump Host

To configure the jump host:

1. Using the VM's public IP, SSH to the jump host VM and become root.
2. Run the following commands:
```
curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl
chmod 755 kubectl
mv kubectl /bin
yum install epel-release -y
yum install jq -y
```
3. Install the Azure client for CentOS by running these commands:
```
rpm --import https://packages.microsoft.com/keys/microsoft.asc
sh -c 'echo -e "[azure-cli]\nname=Azure CLI\nbaseurl=https://packages.microsoft.com/yumrepos/azure-cli\nenabled=1\ngpgcheck=1\ngpgkey=https://packages.microsoft.com/keys/microsoft.asc" > /etc/yum.repos.d/azure-cli.repo'
yum install azure-cli
```
4. Log in to your Azure account and follow the console instructions by running:
```
az login
```
5. Get your Kubernetes cluster credentials by running the following command:
```
az aks get-credentials --resource-group <your resource group name> --name <your kubernetes resource name>
```
For example:
```
az aks get-credentials --resource-group srg-demo --name srg-demo-aks
```
6. Check if kubectl can access the cluster by running:
```
kubectl get nodes
```
Example output:
```
NAME                                STATUS   ROLES   AGE    VERSION
aks-agentpool-36457641-vmss000000   Ready    agent   137m   v1.13.11
aks-agentpool-36457641-vmss000001   Ready    agent   137m   v1.13.11
aks-agentpool-36457641-vmss000002   Ready    agent   137m   v1.13.11
```

Configuring Remote Desktop Protocol (RDP) on Your Jump Host

Since RDP is required for your jump host, you must complete the following configuration steps:

- The installation of xRDP
- Installation of a preferred desktop environment (choice of XFCE, MATE, or GNOME)
- Opening of an RDP port on the jump host network security group (NSG)

Consider the security risks of opening ports to the Internet. Use of a VPN, restricting access to the allowed source IP address, or limiting the amount of time that this port remains available can reduce these risks.

To configure RDP on your jump host:

1. Connect to the jump host and become `root`.
2. Install and enable xrdp. Run these commands:
   ```
   yum install -y epel-release
   yum install -y xrdp
   systemctl enable xrdp
   systemctl start xrdp
   ```
3. Ensure that the firewall is running. Once running, open RDP port 3389/tcp by running these commands:
   ```
   firewall-cmd --add-port=3389/tcp --permanent
   firewall-cmd --reload
   ```
4. Install your preferred desktop environment (XFCE, MATE or GNOME). This example uses MATE and your syntax may differ. Run these commands:
   ```
   yum install -y epel-release
   yum groupinstall "Server with GUI" -y
   yum groupinstall -y "MATE Desktop"
   ```
5. Wait for the install to complete, then reboot the jump host.
6. Connect to the jump host and stay as a jump host user.
7. Create the Xclients file for the user, which will be used for user logins. Run these commands:
   ```
   echo "mate-session" > ~/.Xclients
   chmod a+x ~/.Xclients
   ```
8. Do one of the following:
   a. On the Azure Portal, open the RDP port on the jump host network security group (NSG), then proceed to Step 9, OR,
   b. Run the following command (after which, the procedure is complete)
      ```
      az network nsg rule create -g <RESOURCE GROUP> -n rdp --nsg-nam
      jumphost-nsg --priority 1001 --destination-port-ranges 3389
      ```
      For example:
      ```
      az network nsg rule create -g srg-demo -n rdp --nsg-nam jumphost-nsg --
      priority 1001 --destination-port-ranges 3389
      ```

Find the jump host network security group (NSG) in your resource group. In our example, the NSG is called `jumphost-NSG`

9. In **Settings**, click **Inbound security rules.**
10. Click **Add** and then specify values for these settings:
    - In **Name**, specify a name for the NSG.
    - In **Destination port ranges**, specify 3389.

After you have created and configured the jump host, you can run all further `az cli` commands from your jump host instead of using the Azure Cloud Shell.

Next Step: Configuring the NFS Server

Configuring the NFS Server

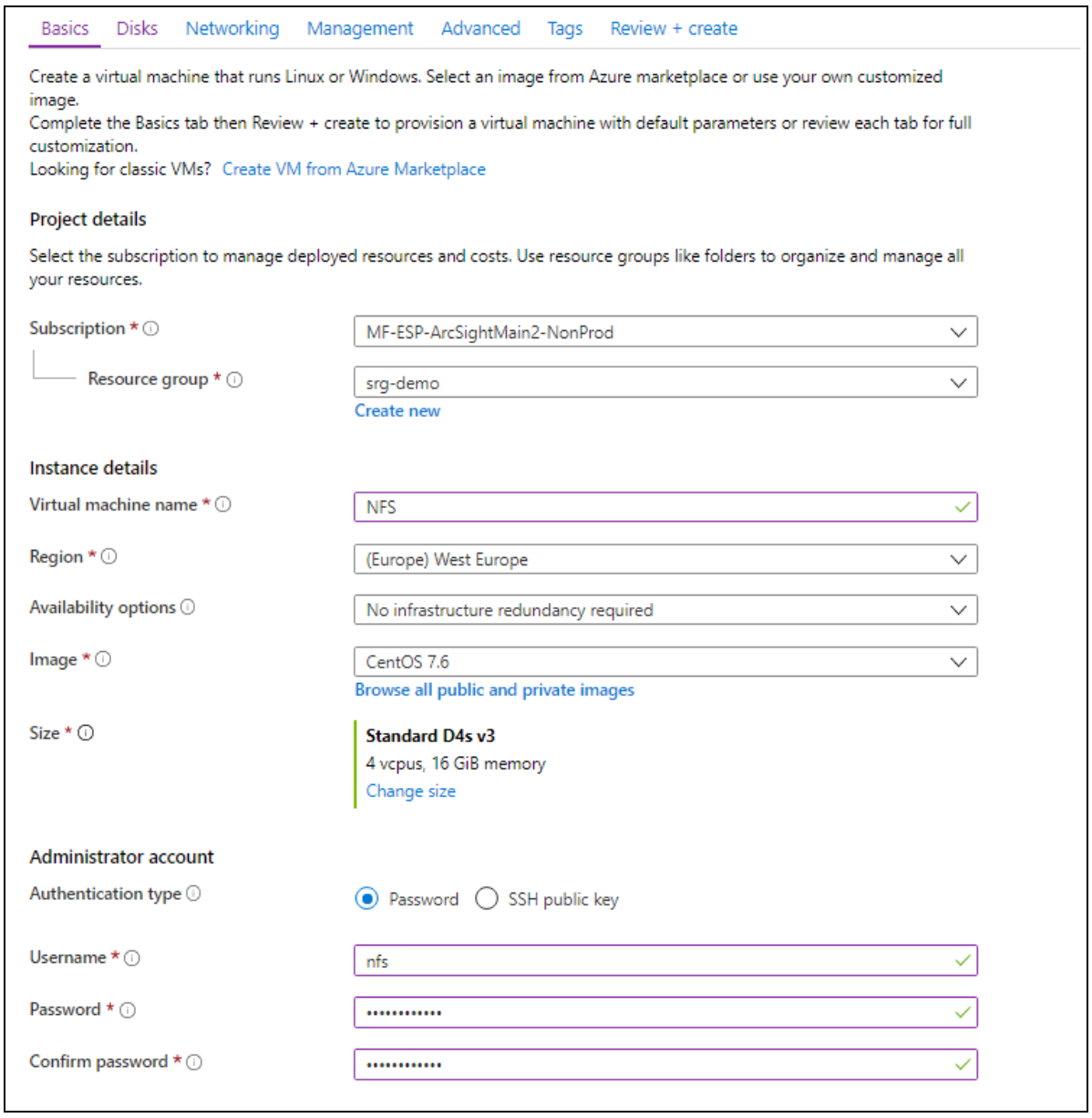
You can configure an NFS server using one of following methods:

- Configure a virtual machine which will be the host NFS server
- Configure the native NetApp service provided by Azure to provision NFS shares

Configuring a Virtual Machine as the Host NFS Server

To create a VM NFS server using the Azure Portal:

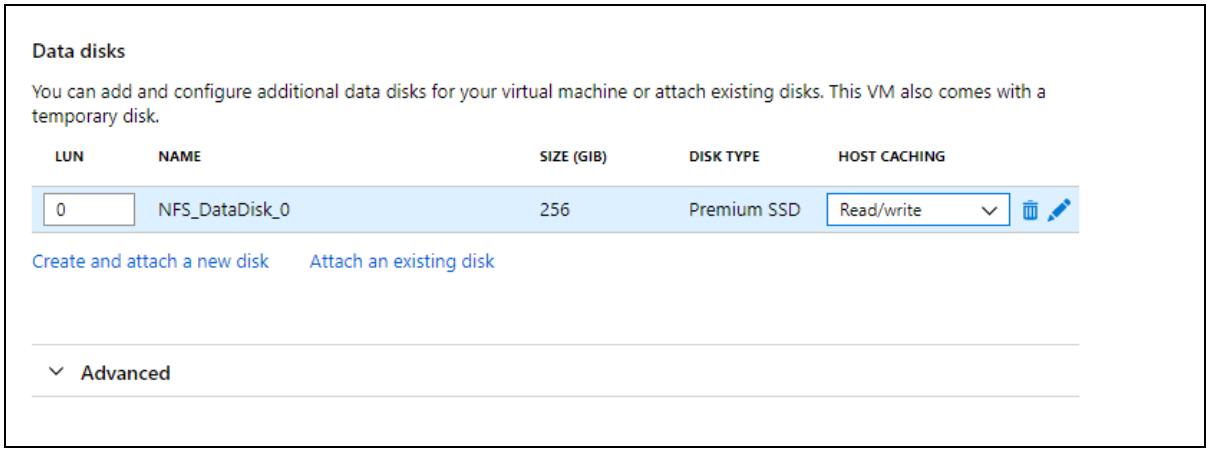
1. Create a resource of type *CentOS-based.*
2. For **Resource group,** select your resource group.
3. In **Virtual machine name**, specify a VM name.
4. For **Size**, select **Change size.** In the popup, choose *D4s_V3,* and click **OK** to confirm. (You can select a different size according to your expected workload.)
5. Set the **Authentication type** to your preferences. Specify the **Administration account details** accordingly. In the examples given here, we use username/password authentication.



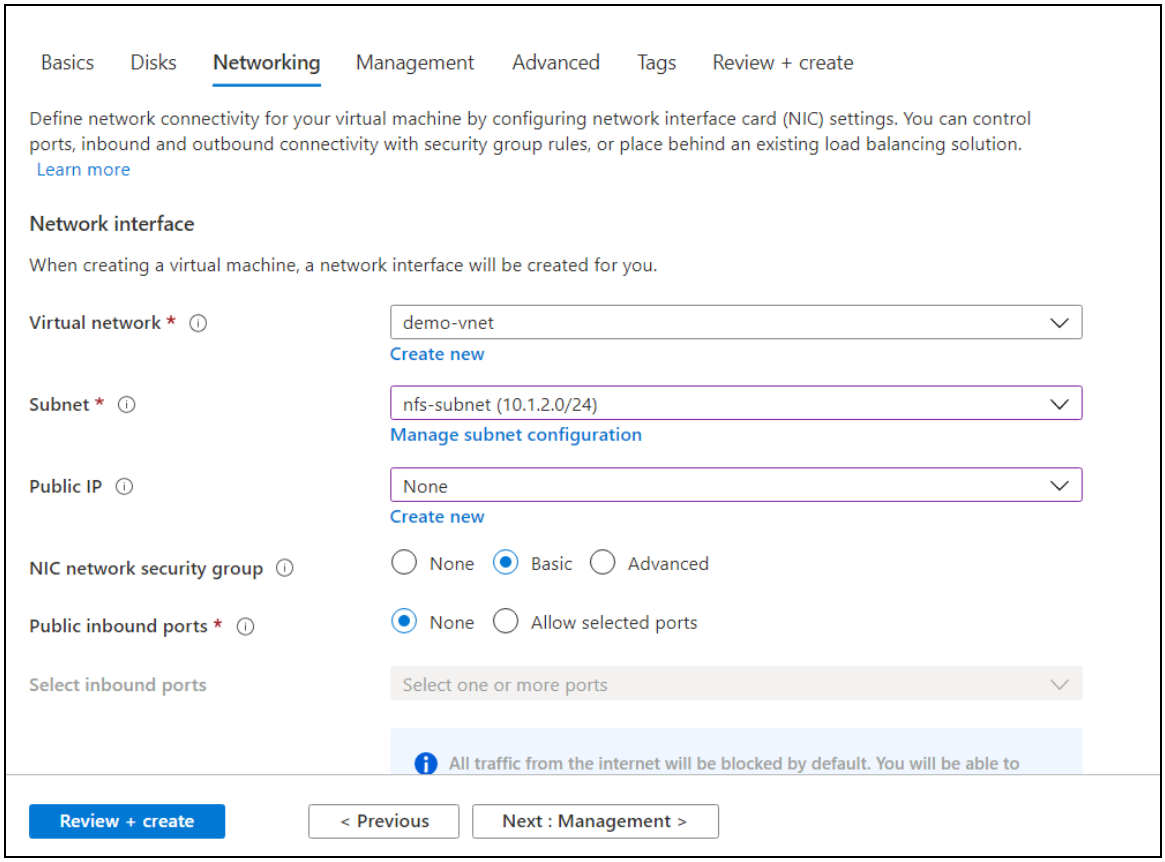
6. Click **Next: Disks**.

   By default, the VM has a small (30GB) disk for the operating system and approximately the same size for the temporary disk. For NFS, you need to attach a new disk with IOPS 1100 or later for better performance.

7. Click **Create and attach a new disk.**
8. Attach a new disk and select a size that meets your requirements. (IOPS should be 1100 or later.)
9. After you add the disk, change the value in *Host Caching* to *Read/write.*



10. Click **Next: Networking**
11. On the **Networking** tab, select values as follows:
    - **Virtual network**: select the virtual network you created earlier (for example, demo-vnet)
    - **Subnet:** select *nfs-subnet*
    - **NIC network security group:** select *Basic*
    - **Public Inbound Ports:** select None.



12. Click **Review + create**.
13. When validation completes, click **Create.**

To create a VM NFS server using the Azure Cloud Shell:

Required permissions: Create security groups, network interface and CentOS-based virtual machines

1. Set your main resource group name to an environment variable:

   ```
   RESOURCE GROUP=<your resource group name>
   ```

   For example:

   ```
   RESOURCE GROUP=srg-demo
   ```

2. Create a network security group for NFS by running this command:

   ```
   az network nsg create -g $RESOURCE_GROUP -n nfs-nsg
   ```

3. Get the nfs-subnet ID and store it to an environment variable for later usage. (You find subnet `nfs-subnet` in virtual network `demo-vnet` in resource group `srg-demo`.)

   ```
   SUBNET_ID=$(az network vnet list -g $RESOURCE_GROUP | jq -r '.[] | select
   (.name == "<your_virtual_network_name>") | .subnets[] | select(.name ==
   "<your_NFS_subnet>") | .id')
   ```

4. Create the network interface `nfs-VMNic` in the subnet from previous command in your resource using network security group nfs-nsg by running the following command:

   ```
   az network nic create --name "nfs-VMNic" --resource-group $RESOURCE_GROUP
   --ip-forwarding  "true" --network-security-group "nfs-nsg" --subnet
   $SUBNET_ID
   ```

5. Create the NFS VM by running this command:

   ```
   az vm create --name "nfs" --resource-group $RESOURCE_GROUP --image
   "OpenLogic:CentOS:7.7:latest" --size "Standard_D4s_v3" --nics "nfs-VMNic"
   --data-disk-sizes-gb "256" --admin-username nfs  --admin-password
   myStrongPassword@!123
   ```

   Where:

   - `--size` is adjusted according to expected workload. To get a list of supported sizes, run the following command:

     ```
     az vm list-sizes -l <LOCATION> | jq ".[] | .name"
     ```

- `--image` can be any supported CentOS. To get a list if CentOS images run the following command:

  ```
  az vm image list -l <LOCATION> -f CentOS --all
  ```

- `--data-disk-sizes-gb` is specified according to workload. Use 256, 512, 1024 and so on.

  For example:

  ```
  {
    "fqdns":"",
    "id":"/subscriptions/af379ae8-90b3-4368-8fe7-
  b6a55ab17720/resourceGroups/srg-
  demo/providers/Microsoft.Compute/virtualMachines/nfs",
    "location":"westeurope",
    "macAddress":"00-0D-3A-AA-E4-F7",
    "powerState":"VM running",
    "privateIpAddress":"10.1.2.4",
    "publicIpAddress":"",
    "resourceGroup":"srg-demo",
    "zones":""
  }
  ```

  The private IP will be used to access the NFS VM from the jump host.

Format the Disk on the NFS VM Using the Azure Cloud Shell

To format the disk:

1. When your NFS VM deployment completes, determine its private IP address using the `az` command. Note the value for later usage.
2. From your jump host, SSH to the VM using its private IP address.

   For example:

   ```
   ssh nfs@10.1.2.4
   ```

3. Log in using the user and password you specified earlier for the NFS VM.
4. Become `root`.
5. Find the device for the data disk by executing the command:

   ```
   fdisk -l
   ```

   This will give you a list of existing disks. Usually the one added is named `/dev/sdc`.

6. Create a new primary partition on the whole device using `fdisk /dev/sdc`. Set it as type *83 - Linux.*

   For example:

```
fdisk /dev/sdc
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.
Device does not contain a recognized partition table
Building a new DOS disklabel with disk identifier 0xc6a2cea5.
The device presents a logical sector size that is smaller than
the physical sector size. Aligning to a physical sector (or optimal
I/O) size boundary is recommended, or performance may be impacted.

Command (m for help): n
Partition type:
p   primary (0 primary, 0 extended, 4 free)
e   extended
elect (default p): p
S

Partition number (1-4, default 1):
First sector (2048-536870911, default 2048):
Using default value 2048

Last sector, +sectors or +size{K,M,G} (2048-536870911, default
536870911):
Using default value 536870911
Partition 1 of type Linux and of size 256 GiB is set

Command (m for help): w
The partition table has been altered!
```

7. After saving the new partition table, run the command:

```
mkfs.xfs /dev/sdc1
```

8. Create a mount point. Run the command:

```
mkdir /nfs
```

9. Get the partition UUID. Run the command:

```
blkid /dev/sdc1
```

   For example:

```
/dev/sdc1: UUID="3696c212-1778-43d5-9d27-d9164686c327" TYPE="xfs"
```

10. In a text editor, open the file `/etc/fstab`, and add an entry to have this new partition mounted after restart. For example:

```
UUID=3696c212-1778-43d5-9d27-d9164686c327 /nfs xfs defaults 0 0
```

11. Mount a new disk partition. Run the command:

```
mount -a
```

12. Verify that it is properly mounted. Run the command:

```
df -h
```

For example:

```
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda2        30G  1.3G   29G   5% /
devtmpfs        7.9G     0  7.9G   0% /dev
tmpfs           7.9G     0  7.9G   0% /dev/shm
tmpfs           7.9G  9.0M  7.9G   1% /run
tmpfs           7.9G     0  7.9G   0% /sys/fs/cgroup
/dev/sda1       497M   65M  433M  13% /boot
/dev/sdb1        99G   61M   94G   1% /mnt/resource <- Azure temporary drive
tmpfs           1.6G     0  1.6G   0% /run/user/0
/dev/sdc1       264G   33M  264G   1% /nfs          <- your new partition for
suite installation
```

Prepare the NFS Server and Export Mount Points

To prepare the server:

1. Check if `nfs-utils` is installed.

```
rpm -qa | grep nfs-utils
```

Sample ouput: `nfs-utils-1.3.0-0.61.el7.x86_64`. The version shown might vary depending on your operating system.

2. If `nfs-utils` is not installed, install it by running the following command:

```
yum install -y nfs-utils
```

3. Configure NFS. Below is the suggested structure of the NFS volumes.

```
/nfs/itom-vol
/nfs/db-single-vol
/nfs/db-backup-vol
/nfs/itom-logging-vol
/nfs/arcsight-volume
```

4. For every NFS volume, run the following set of commands on the VM for NFS.

```
mkdir -p /nfs/volume_name
chown -R <uid>:<gid> /nfs/volume_name
echo "/nfs/volume_name *(rw,sync,anonuid=<uid>,anongid=<gid>,all_
squash)">>/etc/exports
```

   For example:

```
mkdir -p /nfs/itom-vol
chown -R 1999:1999 /nfs/itom-vol
echo "/nfs/itom-vol *(rw,sync,anonuid=1999,anongid=1999,all_
squash)">>/etc/exports
```

   If you use a UID/GID different than 1999/1999, then provide it during the CDF installation in the install script arguments `--system-group-id` and `--system-user-id`. In addition, if you are using NetApp with NFSv4 configuration, consider applying stickybits to all <NFS_root_directory> shares with: `chmod g+w #chmod g+s`

5. (Conditional) If Intelligence is part of the deployment, run the following commands only for arcsight-volume so that the Logstash and Elasticsearch pods do not fail because of permission issues:

```
cd /nfs/arcsight-volume
chown -R 1999:1999 /nfs/arcsight-volume
echo "/nfs/arcsight-volume *(rw,sync,anonuid=1999,anongid=1999,all_
squash)">>/etc/exports
```

6. After configuring all five required volumes, run the following commands.

```
exportfs -ra
systemctl restart rpcbind
systemctl enable rpcbind
systemctl restart nfs-server
systemctl enable nfs-server
```

The NFS configuration is now complete.

Next Step: Creating and Attaching the Data Disk to Nodes

Using NetApp as an NFS Server

You can use Azure's NetApp service as an NFS server. To request access to the service, see the Azure NetApp Files waitlist submission page. You must wait for an official confirmation email from the Azure NetApp Files team before continuing.

Create the NetApp Account

To create your NetApp account using the Azure Portal:

1. Create a resource of type *Azure NetApp Files.*
2. Choose a name and your subscription for the resource.
3. For **Resource group,** select your Kubernetes resource group where you have your virtual network and subnet for NFS.
4. For **Location**, select your resource group as in Step 3.
5. Click **Create** and wait for account creation.

To create your NetApp account using the Azure Cloud Shell:

1. Set your main resource group name to an environment variable:

   ```
   RESOURCE_GROUP=<your resource group name>
   ```

   For eample:

   ```
   RESOURCE_GROUP=srg-demo
   ```

2. Create the NetApp account by running the following command:

   ```
   az netappfiles account create -g $RESOURCE_GROUP --name <ACCOUNT_NAME> -l
   <LOCATION>
   ```

   For example:

   ```
   az netappfiles account create -g $RESOURCE_GROUP --name
   SrgDemoNetAppAdmin -l westeurope
   ```

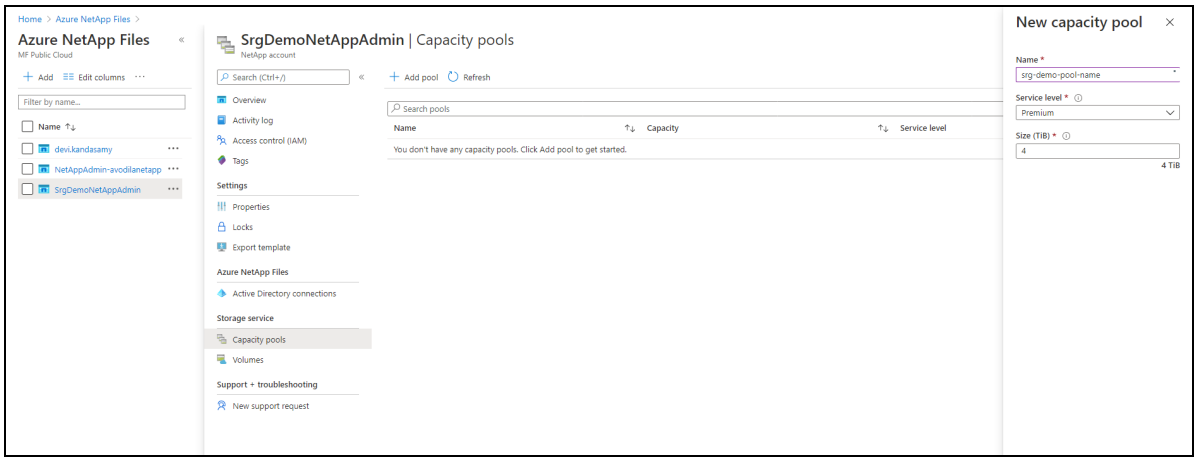
   Where:

- <ACCOUNT_NAME> is your NetApp account name.
- <LOCATION> is the same as for AKS.

Set Up the NetApp Capacity Pool

To set up the capacity pool using the Azure Portal:

1. In the **Azure NetApp Files** tab, browse to your NetApp account.
2. In **Storage Services**, select **Capacity Pools.**



3. Click **+** and specify values for the following:
   - **Name:** specify a name for the pool.
   - **Service Level:** select a service level.
   - **Pool Size:** specify 4 (TB) for the pool size. (This is a service minimum.)

To set up the capacity pool using the Azure Cloud Shell:

1. Set your main resource group name to an environment variable:

   ```
   RESOURCE_GROUP=<your resource group name>
   ```

   For example:

   ```
   RESOURCE_GROUP=srg-demo
   ```

2. Run the command:

   ```
   az netappfiles pool create -g $RESOURCE_GROUP --account-name <ACCOUNT_
   NAME> --name <POOL_NAME> -l <LOCATION> --size 4 --service-level premium
   ```

   Where:

- `<POOL_NAME>` is your new pool name
- `<ACCOUNT_NAME>` is the NetApp account name specified in previous step

For example:

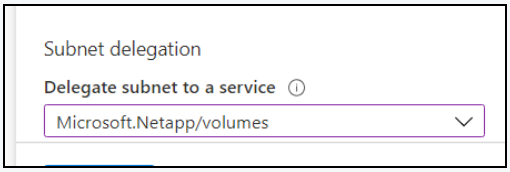
```
az netappfiles pool create -g $RESOURCE_GROUP --account-name
SrgDemoNetAppAdmin --name srg-demo-pool-name -l westeurope --size 4 --
service-level premium
```

Create and Prepare the Volume

To create and prepare the volume using the Azure Portal:

1. In your NetApp account resource, browse to **Storage service.**
2. Select **Volumes.**
3. Press **+ Add volume.**
4. Specify a name for the volume
5. Ensure that your volume is in the same virtual network as `aks-virtual-network` and `nfs-subnet.`
6. For **Subnet**, select *nfs-subnet.*

   Under **Subnet delegation,** the subnet must be delegated to `Microsoft.NetApp/volumes.`

7. Click **Next:Protocol**
8. Ensure that the **Protocol** type is *NFS* and **Version** is *NFSv4.1*.
9. In the **Export policy section**, select the check box for *0.0.0.0/0 Read & Write.*
10. Specify the file path that will be used to create the export path for the volume.
11. Click **Review + Create** at the bottom of the page. If you are satisfied with your settings, click **Create**.

To create and prepare the volume using the Azure Cloud Shell:

1. Set your main resource group name to an environment variable:

   ```
   RESOURCE GROUP=<your resource group name>
   ```

   For example:

```
RESOURCE GROUP=srg-demo
```

2. Run the command:

```
az netappfiles volume create \
-g $RESOURCE_GROUP \
--account-name <ACCOUNT_NAME> \
--pool-name <POOL_NAME> \
--name <VOLUME_ROOT> -l <LOCATION> \
--service-level premium \
--usage-threshold <VOLUME_SIZE> \
--file-path <FILE_PATH> \
--vnet <VIRTUAL_NETWORK> \
--subnet <NFS_SUBNET_NAME> \
--protocol-types NFSv4.1 \
--rule-index <INDEX> \
--allowed-clients <CLIENT_LIST>
```

Where:

- <INDEX> integer number of nfs creation rule.
- <ACCOUNT_NAME> is your netApp account name.
- <POOL_NAME> is the capacity pool created previously.
- <VOLUME_ROOT> is your volume root name.
- <LOCATION> is location of your NetApp.
- <VOLUME_SIZE> size for NFS volume in GB.
- <FILE_PATH> is the path to your volumes.
- <VIRTUAL_NETWORK> the virtual network to which your subnets belong.
- <NFS_SUBNET_NAME> is your subnet for NFS.
- <CLIENT_LIST> is specified as comma-separated value (CSV) string with IPv4 CIDRs, IPv4 host addresses, and host names to access the NFS share. Make sure to include either the range of subnets or separate IPs/hostnames, including the jumphost.
  For example:

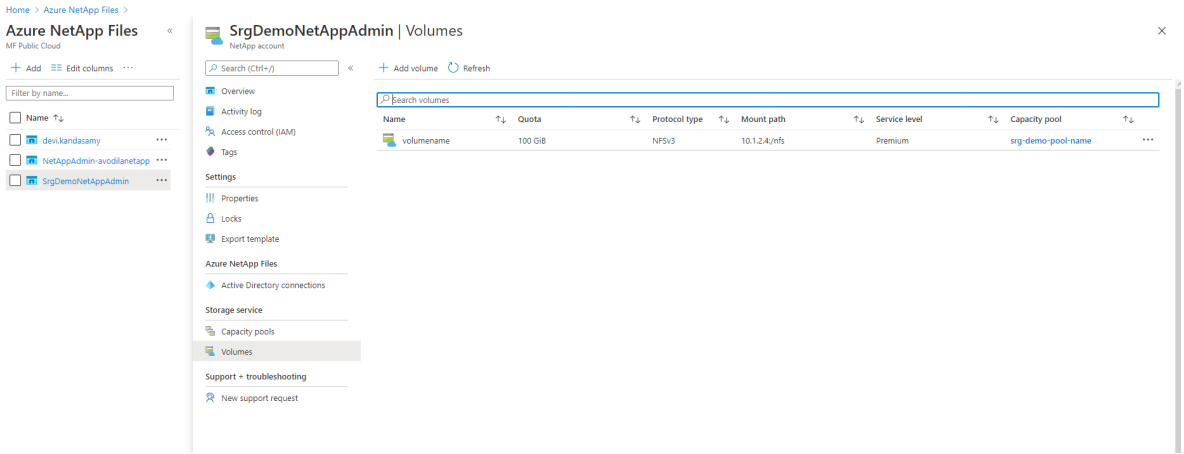
```
az netappfiles volume create \
-g $RESOURCE_GROUP \
--account-name SrgDemoNetAppAdmin \
--pool-name srg-demo-pool-name \
--name volumename -l westeurope \
--service-level premium \
```

```
--usage-threshold 100 \
--file-path "nfs" \
--vnet demo-vnet \
--subnet nfs-subnet \
--protocol-types NFSv4.1 \
--allowed-clients 20.120.3.0/24 \
--rule-index 1
```

Configure and Create the Volumes

To configure and create the volumes:

1. Find your `<mount_path_ip>` and `<mount_path_file_name>` by navigating to your volume page. These parameters were confirmed in the previous step after successful creation of the prepared volume. Note both of these parameters for later use.



2. Log in to the jump host.
3. If not already present, install the NFS client by sudo:

```
# yum install nfs-utils if not present
```

4. Get and unzip the file `cdf-deployer.zip`.
5. In the `scripts` folder, run the following command:

```
sudo ./createFileStore.sh <mount_path_ip> <mount_path_file_name> <NFS_
version>
```

For example:

```
sudo ./createFileStore.sh "10.1.2.4" "/nfs" "4.1"
```

Next Step: Creating and Attaching the Data Disk to Nodes

Creating and Attaching the Data Disk to Nodes

By default, AKS nodes are created with a temporary data disk. Disk size depends on the `--node-vm-size` parameter and might not fit your needs.

In this section, you prepare the Azure managed disk and attach it to the nodes which will host Transformation Hub.

This process has three parts.

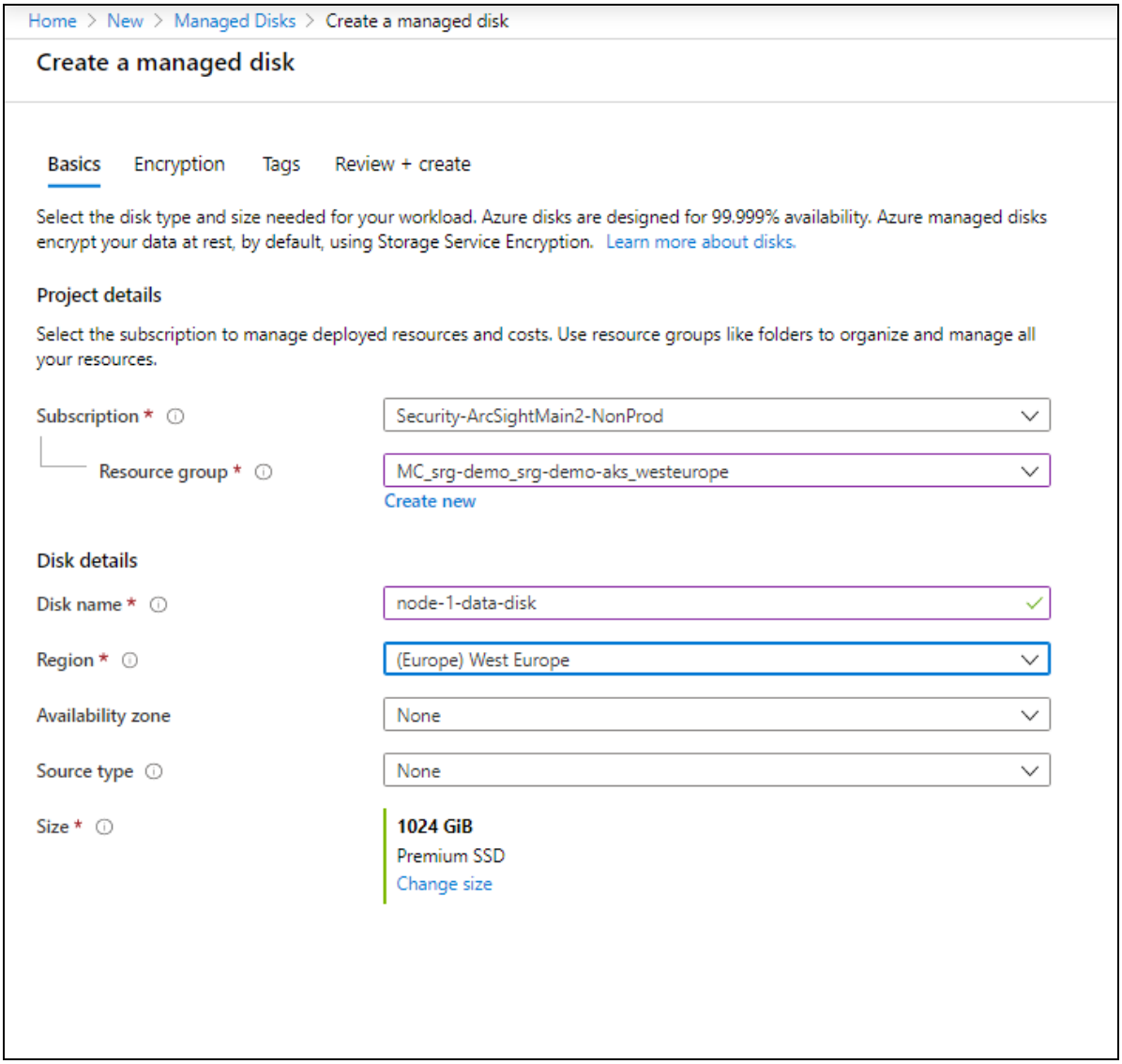
1. "Creating the Managed Data Disk" below
2. "Attaching the Disk to the AKS Node" on page 315
3. "Formatting and Mounting the Attached Disk" on page 316

Creating the Managed Data Disk

To create the managed data disk for a node using the Azure Portal:

1. Create a resource of type *Managed Disks.*
2. In the `Subscription` drop-down, select your subscription.
3. In the `Resource group` drop-down, select your AKS resource group.

   The AKS resource group is in the format `MC_<your_resource_group>_<aks_name>_<location>`.
4. In `Disk name` field, specify a name for the managed disk.
5. Based on the location you specified earlier, in the `Region` drop-down, select the region.
6. Based on the expected workload, in the `Size` drop-down, select the size.



7. Click **Review + create**.
8. After validation, click **Create**.
9. When the deployment finishes, click **Download** to get the `json` file with deployment results. Inside the archive is the `deployment.json` file.
10. For use in attaching this disk to the AKS node, make note of the `primaryResourceId` value.
11. Repeat Step 1 through Step 9 for each AKS node.

To create the managed data disk using the Azure Cloud Shell:

Required permissions: create disk

1. Get the AKS resource group and store it in an environment variable for later usage:

   ```
   CLUSTER_RESOURCE_GROUP=$(az aks show --resource-group <RESOURCE GROUP> --name <AKS NAME> --query nodeResourceGroup -o tsv)
   ```

   For example:

   ```
   CLUSTER_RESOURCE_GROUP=$(az aks show --resource-group srg-demo --name srg-demo-aks --query nodeResourceGroup -o tsv)
   ```

2. Create the managed disk by running the following command:

   ```
   az disk create --name <DISK NAME> --resource-group $CLUSTER_RESOURCE_GROUP --size-gb <DISK SIZE>
   ```

   For example:

   ```
   az disk create --name node-1-data-disk --resource-group $CLUSTER_RESOURCE_GROUP --size-gb 1024
   ```

3. From the results, get the `id` value. It will be used later to attach the disk to the AKS node.

   The value will resemble the following:

   ```
   /subscriptions/af379ae8-90b3-4368-8fe7-b6a55ab17720/resourceGroups/MC_srg-demo_srg-demo-aks_westeurope/providers/Microsoft.Compute/disks/node-1-data-disk
   ```

4. Repeat Step 1 through Step 3 for each expected AKS node.

Attaching the Disk to the AKS Node

1. Get the virtual machine scale set and store it to an environment variable:

   ```
   VMSS=$(az vmss list -g $CLUSTER_RESOURCE_GROUP | jq -r .[0].name)
   ```

   If you open a new session, run commands to set the `CLUSTER_RESOURCE_GROUP` environment variable first.

2. Attach the disk to instance by running the command:

   ```
   az vmss disk attach --resource-group $CLUSTER_RESOURCE_GROUP --vmss-name $VMSS --instance-id <INDEX OF INSTANCE> --disk <DISK ID>
   ```

   Where:

- `<INDEX OF INSTANCE>` is number of node in cluster, starting from 0
- `<DISK ID>` disk resource ID, obtained during creation.

> For example, the following command would attach the disk to the instance with an instance - id of 0.
> `az vmss disk attach --resource-group $CLUSTER_RESOURCE_GROUP --vmss-name $VMSS --instance-id 0 --disk /subscriptions/af379ae8-90b3-4368-8fe7-b6a55ab17720/resourcegroups/MC_srg-demo_srg-demo-aks_westeurope/providers/Microsoft.Compute/disks/node-1-data-disk`

3. Repeat Step 1 and Step 2 for the rest of the disks and node instances

Formatting and Mounting the Attached Disk

1. Upload the `id_rsa` to your jump host.

   > Use the `id_rsa` file you generated when creating the Azure Kubernetes Service.

2. Make `id_rsa` read only by running the following command:

   ```
   chmod 400 id_rsa
   ```

3. Get the node's private IP by running the following command:

   ```
   kubectl get nodes -o wide
   ```

   For example:

   ```
   NAME                                STATUS   ROLES   AGE   VERSION
   INTERNAL-IP   EXTERNAL-IP
   aks-nodepool1-84569686-vmss000000   Ready    agent   79m   v1.15.10
   10.240.0.4    <none>
   aks-nodepool1-84569686-vmss000001   Ready    agent   79m   v1.15.10
   10.240.0.5    <none>
   aks-nodepool1-84569686-vmss000002   Ready    agent   79m   v1.15.10
   10.240.0.6    <none>
   ```

4. Make an SSH connection from the jump host to an AKS node (use the internal IP address).

   Successive commands will be executed on the AKS node to which you are connected.

   For example:

   ```
   ssh -i id_rsa azureuser@10.240.0.4
   ```

5. On the same AKS node, become `root`.

6. Find the device for the data disk by running the following command:

```
fdisk -l
```

This command gives you a list of existing disks. Usually the one added is `/dev/sdc`.

7. Using `fdisk /dev/sdc`, create a new primary partition on the whole device and set it as type *83 - Linux*.

   For example (input and output):

```
fdisk /dev/sdc
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table
Building a new DOS disklabel with disk identifier 0xc6a2cea5.
The device presents a logical sector size that is smaller than
the physical sector size. Aligning to a physical sector (or optimal
I/O) size boundary is recommended, or performance may be impacted.

Command (m for help): n
Partition type:
p   primary (0 primary, 0 extended, 4 free)
e   extended
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-536870911, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-536870911, default
536870911):
Using default value 536870911
Partition 1 of type Linux and of size 1024 GiB is set

Command (m for help): w
The partition table has been altered!
```

8. After saving the new partition table, create the file system by running the following command:

```
mkfs.xfs /dev/sdc1
```

9. Create the mount point by running the following command:

```
mkdir /opt/arcsight
```

10. Get the partition UUID by running the following command:

```
blkid /dev/sdc1
```

For example:

```
/dev/sdc1: UUID="3696c212-1778-43d5-9d27-d9164686c327" TYPE="xfs"
```

11. Add an entry to the `/etc/fstab` file to have this new partition mounted after restart.

For example:

```
UUID=3696c212-1778-43d5-9d27-d9164686c327 /opt/arcsight xfs defaults 0 0
```

12. Mount a new disk partition by running the following command:

```
mount -a
```

13. Verify it is properly mounted by running the following command:

```
df -h
```

For example (output):

```
Filesystem      Size  Used Avail Use% Mounted on
udev            7.9G     0  7.9G   0% /dev
tmpfs           1.6G  812K  1.6G   1% /run
/dev/sda1        97G  9.4G   88G  10% /              <- Azure temporary
drive
tmpfs           7.9G     0  7.9G   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           7.9G     0  7.9G   0% /sys/fs/cgroup
/dev/sda15      105M  3.6M  101M   4% /boot/efi
/dev/sdb1        32G   48M   30G   1% /mnt
tmpfs           7.9G   12K  7.9G   1% /var/lib/kubelet/pods/7194d3a7-
cc84-42bd-accb-30b09fcd1d27/volumes/kubernetes.io~secret/kube-proxy-
token-cnxn8
overlay          97G  9.4G   88G  10%
/var/lib/docker/overlay2/3e04813889c25709c31206a48ee82fa67d677b76a6b1aab5
e7d7246b911a3bee/merged
shm              64M     0   64M   0%
/var/lib/docker/containers/bc0dd2ea23a9c0640e10ad4664addeb437f4ad4ac08302
60eef942f70bcb0c0a/mounts/shm
overlay          97G  9.4G   88G  10%
/var/lib/docker/overlay2/b8290059f18b2f9d311395abcf12ccb377ed7107db5fa5fc
cc46b6fc594e7da8/merged
tmpfs           1.6G     0  1.6G   0% /run/user/1000
```

```
/dev/sdc1       1.0T  1.1G 1023G   1% /opt/arcsight    <- your new partition for
Arcsight products
```

14. Repeat Steps 4 through Step 12 for all remaining nodes and their disks.

Next Step: Preparing a Private DNS Zone

Preparing a Private DNS Zone

Required permissions: create private DNS zone. You will also need a link to the virtual network.

To prepare the private DNS zone:

1. Set your main resource group name to an environment variable, for example:
   `RESOURCE_GROUP=srg-demo`
2. Create the private-dns zone (for example, `arcsight.private.com`) in your resource group by running the command:
   `az network private-dns zone create -g $RESOURCE_GROUP -n arcsight.private.com`

You can use another zone name in place of `arcsight.private.com`, but you must use the same DNS suffix for the `--external-access-host` argument during CDF installation.

3. Link the private-dns zone with your virtual network by running the command:

```
az network private-dns link vnet create \
-g $RESOURCE_GROUP \
-n DNSLink \
-z arcsight.private.com \
-v <your virtual network, such as demo-vnet> \
-e false
```

Next Step: Assigning an IP Address to Private DNS

Assigning an IP Address to Private DNS

In this step you assign an IP address from the aks-subnet range to the domain name (external access host).

To assign an IP address to private DNS:

1. Set your main resource group name to an environment variable; for example:
   `RESOURCE_GROUP=srg-demo`

Alternatively, use the resource group where your `vnet` and `private-dns` zone are located.

2. Get the address prefix by running the command:
   `az network vnet subnet show -g $RESOURCE_GROUP --vnet-name <your virtual network> --name <subnet for AKS>| jq -r .addressPrefix`

For example:
`az network vnet subnet show -g $RESOURCE_GROUP --vnet-name demo-vnet --name aks-subnet | jq -r .addressPrefix`

Example result:
`10.1.1.0/24`

You can select any IP from this range (excepting the first N IP addresses, which are occupied by AKS nodes).

Example selection in this range: `10.1.1.101`

3. Assign the IP by running the following command:
   `az network private-dns record-set a add-record -g $RESOURCE_GROUP -z <PRIVATE DNS ZONE> -n <RECORD SET NAME> -a <EXTERNAL-IP>`

Parameters

`<PRIVATE DNS ZONE>` is the `private-dns` zone created earlier (in our example it was `arcsight.private.com`).

`<RECORD SET NAME>` the name of the record set relative to the zone (in our example, `installer`).

`<EXTERNAL-IP>` IP must be from aks-subnet range.

Example command:
`az network private-dns record-set a add-record -g $RESOURCE_GROUP -z arcsight.private.com -n installer -a 10.1.1.101`

This command will create `installer.arcsight.private.com` with the IP address 10.1.1.101.

Next Step: Labeling Azure Kubernetes Service Nodes

Labeling Azure Kubernetes Service Nodes

Labeling is a means for identifying application processing and qualifying the application as a candidate to run on a specific node. For example, labeling an AKS node with the label `kafka=yes` specifies that a Kafka instance runs on that node.

For more information about labeling, see "Understanding Labels and Pods" on page 578

Labels required for AKS nodes include the following:

Label	The node runs...
`kafka=yes`	Kafka
`zk=yes`	ZooKeeper
`fusion=yes`	Fusion
`th-processing=yes`	Transformation Hub data
`th-platform=yes`	Transformation Hub
`intelligence=yes`	Pods that manage functions and services for the ArcSight Intelligence capability
`intelligence-spark=yes`	Analytics services for the ArcSight Intelligence capability
`intelligence-datanode=yes`	Pods that manage HDFS services for the ArcSight Intelligence capability
`intelligence-namenode=yes`	HDFS NameNode services for the ArcSight Intelligence capability. Place this label on one node only. The node and the hostname or IP address in the **HDFS NameNode** field in the **Intelligence** tab of the **CDF Management Portal** must match.

To label your AKS nodes:

1. On your jump host, get a list of AKS nodes by running the following command:
   `kubectl get nodes`

```
NAME                                STATUS   ROLES   AGE    VERSION
aks-agentpool-36457641-vmss000000   Ready    agent   137m   v1.13.11
aks-agentpool-36457641-vmss000001   Ready    agent   137m   v1.13.11
aks-agentpool-36457641-vmss000002   Ready    agent   137m   v1.13.11
```

2. Label the first AKS node by running the following command:
   `kubectl label node <node-name> zk=yes kafka=yes th-processing=yes th-platform=yes fusion=yes`

```
kubectl label node aks-agentpool-36457641-vmss000000 zk=yes kafka=yes th-processing=yes th-platform=yes fusion=yes
```

3. Repeat the command in step 2 for each additional node.

Next Step: Uploading Product Images

Uploading Product Images

To upload the product images to the Azure Container Registry, you must first determine the ACR credentials, then perform the upload.

Portal

To upload product images to the ACR, using Azure commands to get the credentials:

1. Navigate to the Azure management portal and open the Azure Container Registry.
2. Navigate to **Settings** and click **Access keys**.
3. Collect the login server, username, and password details, they will be used while uploading the images to the container registry.
4. Unzip the CDF deployer to a local directory (such as `/tmp`).

   For example:

   ```
   # cd /tmp
   # unzip cdf-deployer.zip...
   ```

5. Change directory to the deployer scripts folder.

   For example:

   ```
   # cd cdf-deployer/scripts/
   ```

6. Run the uploadimages.sh script with credentials from the ACR by running the following commands:

   ```
   # ./uploadimages.sh -o your-org-name -r <REGISTRY LOGIN SERVER> -u <USERNAME> -p
   <PASSWORD> -F /tmp/cdf-byok-images-<VERSION>.tar -c 2
   ```

   ```
   # ./uploadimages.sh -o your-org-name -r <REGISTRY LOGIN SERVER> -u <USERNAME> -p
   <PASSWORD> -F /tmp/<deployed_capability><VERSION>.tar
   ```

   For example:

   ```
   # ./uploadimages.sh -o your-org-name -r srgdemo.azurecr.io -u srgdemo -p
   GEev87wtAW+FtBGTyADxgr9Fivg6a2gC -F /tmp/cdf-byok-images-<VERSION>.tar -c 2
   ...
   Upload completed in 1690 seconds.
   Upload-process successfully completed.
   ```

```
# ./uploadimages.sh -o extremelyfocused -r extremelyfocused.azurecr.com -u jsmith
-p testpassword -F /opt/arcsight/fusion-1.3.0.1810-master.tar
...
Upload completed in 1690 seconds.
Upload-process successfully completed.
```

About Running `uploadimages.sh`

- Choose the value of the `-o` argument carefully, since it needs to be used during installation and for all `uploadimages.sh` calls.
- The `-c` argument indicates concurrent upload (maximum can be half of the CPU cores capacity), and can speed up the upload process.
- Uploading images is long process, and can take up to 60 minutes to complete. The exact time for completion depends largely on connectivity.
- See `uploadimages.sh --help` for more information.

CLI

To upload product images to the ACR using Azure commands for credentials:

1. Get the registry name and password by running these commands:

   ```
   # credentials=$(az acr credential show --name <your ACR name> -g <RESOURCE
   GROUP>)
   # echo $credentials | jq -r '.username'
   # echo $credentials | jq -r '.passwords[0].value'
   ```

   For example:

   ```
   # credentials=$(az acr credential show --name srgdemoACR -g srg-demo)
   # echo $credentials | jq -r '.username'
   # echo $credentials | jq -r '.passwords[0].value'
   ```

2. Determine the name of the registry log in server by running the following command:

   ```
   # az acr show --name <REGISTRY NAME> -g <RESOURCE GROUP NAME> | jq -r
   '.loginServer'
   ```

   For example:

   ```
   # az acr show --name srgdemoACR -g srg-demo | jq -r '.loginServer'
   ```

3. Unzip the CDF deployer to a local directory (such as `/tmp`).

For example:

```
# cd /tmp
# unzip cdf-deployer.zip...
```

4. Change directory to the deployer scripts folder.

   For example:

   ```
   # cd cdf-deployer/scripts/
   ```

5. Run the `uploadimages.sh` script with credentials from the ACR by running the following command:

   ```
   # ./uploadimages.sh -o your-org-name -r <REGISTRY LOGIN SERVER> -u <USERNAME> -p
   <PASSWORD> -F /tmp/cdf-byok-images-<VERSION>.tar -c 2
   ```

   For example:

   ```
   # ./uploadimages.sh -o your-org-name -r srgdemo.azurecr.io -u srgdemo -p
   GEev87wtAW+FtBGTyADxgr9Fivg6a2gC -F /tmp/cdf-byok-images-<VERSION>.tar -c 2
   ...
   Upload completed in 1690 seconds.
   Upload-process successfully completed.
   ```

   The -o your-org-name parameter must be in lowercase, as shown in the following.

   ```
   ./uploadimages.sh -o extremelyfocused -r extremelyfocused.azurecr.com -u jsmith -
   p testpassword -F /opt/arcsight/fusion-1.3.0.1810-master.tar
   ...
   Upload completed in 1690 seconds.
   Upload-process successfully completed.
   ```

About Running `uploadimages.sh`

- Choose the value of the `-o` argument carefully, since it needs to be used during installation and for all `uploadimages.sh` calls.
- The `-c` argument indicates concurrent upload (maximum can be half of the CPU cores capacity), and can speed up the upload process.
- Uploading images is long process, and can take up to 60 minutes to complete. The exact time for completion depends largely on connectivity.
- See `uploadimages.sh --help` for more information

Next Step: Installing the Database in Azure

Installing the Database in Azure

This section provides information about installing the ArcSight Database in Azure.

- "Preparing the Azure Virtual Machine" below
- "Installing Prerequisites" on page 329
- "Configuring and Installing the Database Server" on page 333

Preparing the Azure Virtual Machine

This section describes how to prepare the Azure virtual machine for database installation.

• Creating the Database Virtual Machines

1. Log in to the Azure portal.
2. Select an active Azure subscription.
3. Click **Create a Resource** or select an existing resource.
4. In the search box (case-insensitive), specify *CentOS 8.4* and select an image.
5. Click **create**.
6. In the **Virtual machine name** field, specify a name for the virtual machine.
7. In the **Image** drop-down list, select the supported image.
8. In the **Size** drop-down list, select the VM that will be accomplishing the database requirements. For example, **D8s_v3**.
9. Select the appropriate **Authentication type**, and specify the details.
10. In the **Select inbound ports** drop-down list, be sure **SSH (22)** is selected.
11. In the **Public inbound ports** area, select the **Allow selected ports** option.
12. Click **Next: Disks**.
13. On the **Disks** tab, create a new disk of minimum size 256 GB.
14. For **Data disks** and **Host caching** select **Read/write**.
15. Select the **Networking** and your **Virtual network**.
16. For subnet, create or select a subnet for the database and select it in the subnet drop-down..
17. Click **Review + create** to review and create a VM. (Give the deployment time to complete.)
18. Click **Create**.
19. Click **Go to your resource** and launch the new VM.

20. In the left navigation panel, click **Overview > DNS Name.**
21. Set **Assignment** to static and then specify the **DNS name** label.
22. Click **Save**.
23. In the left navigation pane, click **Networking**.
24. Select your Network Interface.
25. From the left menu, select **IP configuration**.
26. Navigate to the IP Forwarding Field, and select **Enabled**.
27. Click **Save**.

You must create one virtual machine for each database node.

• Configuring the Database Virtual Machines

1. SSH to your VM with it's public IP address.
2. Log in with your VM user, then become root.
3. Change your root password.
4. Create a folder for the ArcSight database by running the command:

```
mkdir /opt/vertica
```

5. Find the device for the data disk by running the following command:

```
fdisk -l
```

Depending on your VM, the result can be sdc, sdb, etc.

6. Create partitions on the data disk using the datadisk you obtained in the previous step.

The ArcSight database requires a minimum 2 GB swap partition irrespective of the amount of RAM installed. The remaining part of the disk, you can format using ext4 and mount it to `/opt/vertica`.

```
[root@vertica1 ~]# fdisk /dev/<datadisk>
```

   a. Specify **n** and press Enter.
   b. Press Enter.
   c. Press Enter.
   d. Press Enter
   e. Specify the value for **${sizePartitionForSwap}** and press Enter.

You must provide the value in KB, minimum value is 2 GB.

f. Specify **n** and press Enter.

g. Press Enter.

h. Press Enter.

i. Press Enter.

j. Specify **p** and press Enter.

k. Specify **w** and press Enter.

For instance:

```
Welcome to fdisk (util-linux 2.23.2)

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table
Building a new DOS disklabel with disk identifier 0x2959fe99.

The device presents a logical sector size that is smaller than
the physical sector size. Aligning to a physical sector (or optimal
I/O) size boundary is recommended, or performance may be impacted.

Command (m for help): n
Partition type:
p   primary (0 primary, 0 extended, 4 free)
e   extended
Select (default p):
Using default response p
Partition number (1-4, default 1):
First sector (2048-536870911, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-536870911, default
536870911):
${sizePartitionForSwap}
Partition 1 of type Linux and of size X GiB is set

Command (m for help): n
Partition type:
p   primary (1 primary, 0 extended, 3 free)
e   extended
Select (default p):
Using default response p
Partition number (2-4, default 2):
```

```
First sector (4196352-536870911, default 4196352):
Using default value 4196352
Last sector, +sectors or +size{K,M,G} (4196352-536870911, default
536870911):
Using default value 536870911
Partition 2 of type Linux and of size XXX GiB is set

Command (m for help): p

Disk /dev/<datadisk>: XXX GB, 274877906944 bytes, 536870912 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disk label type: dos
Disk identifier: 0xcca9a285

Device Boot             Start         End      Blocks   Id  System
/dev/<datadisk>1         2048     2097152     1047552+  83  Linux
/dev/<datadisk>2      2099200   536870911   267385856   83  Linux

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

7. Create a swap partition by running the command:

```
mkswap /dev/<datadisk>1
```

8. Activate the swap partition by running the command:

```
swapon /dev/<datadisk>1
```

9. To format the rest of disk to ext4, run the command:

```
mkfs.ext4 /dev/<datadisk>2
```

10. Get the last created UUID of the disks from the output by running the command:

```
blkid
```

11. For the swap partition (<datadisk>1) and for the rest of the disk (<datadisk>2), note of the given UUID values, modify the /etc/fstab file, and add the following lines by replacing the UUIDs:

```
UUID=<UUID <datadisk>1>    none   swap  sw        0 0
UUID=<UUID <datadisk>2>    /opt/vertica  ext4   defaults  0 0
```

12. Mount all by running the command:

```
mount -a
```

13. Check if `/opt/vertica` is assigned by running the command:

```
df -h
```

14. Enable the swap partition by running the command:

```
swapon -a
```

15. Check the swap partition size by running the command:

```
free -h
```

16. Repeat these steps for each expected database node.

Installing Prerequisites

This section describes how to install the prerequisites necessary to install the Azure database.

1. Configure passwordless communication from the node1 server to all of the node servers in the cluster.
   a. On the node1 server, run the ssh-keygen command:

```
ssh-keygen -t rsa
```

   b. Copy the key from node1 to all of the nodes, including node1, using the node IP address:

```
ssh-copy-id -i ~/.ssh/id_rsa.pub root@$node1
ssh-copy-id -i ~/.ssh/id_rsa.pub root@$node2
ssh-copy-id -i ~/.ssh/id_rsa.pub root@$node3
```

2. Set up and activate `/etc/rc.local` by running the following command:

```
#!/bin/sh
function drive {
block_device=`realpath $(df $1 | grep '^/' | cut -d' ' -f1)`
partition=$(echo $block_device | sed -e "s#/dev/##")
if [[ $partition == dm-* ]]; then
echo $partition
else
```

```
echo $partition | cut -c1-3
fi
}
cat > /etc/rc.local << EOF
#!/bin/sh
touch /var/lock/subsys/local
/sbin/blockdev --setra 2048 /dev/$(drive /)
/sbin/blockdev --setra 2048 /dev/$(drive /opt/vertica)
echo deadline > /sys/block/$(drive /)/queue/scheduler
echo deadline > /sys/block/$(drive /opt/vertica)/queue/scheduler
echo madvise > /sys/kernel/mm/transparent_hugepage/enabled
tuned-adm profile throughput-performance
EOF
chmod 755 /etc/rc.local
/etc/rc.local
```

3. Add the following parameters to `/etc/sysctl.conf`.

```
net.core.somaxconn = 1024
net.core.wmem_max = 16777216
net.core.rmem_max = 16777216
net.core.wmem_default = 262144
net.core.rmem_default = 262144
net.core.netdev_max_backlog = 100000
net.ipv4.tcp_mem = 16777216 16777216 16777216
net.ipv4.tcp_wmem = 8192 262144 8388608
net.ipv4.tcp_rmem = 8192 262144 8388608
net.ipv4.udp_mem = 16777216 16777216 16777216
net.ipv4.udp_rmem_min = 16384
net.ipv4.udp_wmem_min = 16384
vm.swappiness = 1
```

where,

Parameter	Description
`net.core.somaxconn = 1024`	Increases the number of incoming connections
`net.core.wmem_max = 16777216`	Sets the send socket buffer maximum size in bytes
`net.core.rmem_max = 16777216`	Sets the receive socket buffer maximum size in bytes
`net.core.wmem_default = 262144`	Sets the receive socket buffer default size in bytes
`net.core.rmem_default = 262144`	Controls the default size of receive buffers used by sockets

`net.core.netdev_max_backlog = 100000`	Increase the length of the network interface input queue
`net.ipv4.tcp_mem = 16777216 16777216 16777216`	
`net.ipv4.tcp_wmem = 8192 262144 8388608`	
`net.ipv4.tcp_rmem = 8192 262144 8388608`	
`net.ipv4.udp_mem = 16777216 16777216 16777216`	
`net.ipv4.udp_rmem_min = 16384`	
`net.ipv4.udp_wmem_min = 16384`	
`vm.swappiness = 1`	Defines the amount and frequency at which the kernel copies RAM contents to a swap space For more information, see Check for Swappiness.

4. Next, run the following command:

```
sysctl -p
```

5. To disable the firewall **WARN (N0010)**, use `iptables`:

```
iptables -F
iptables -t nat -F
iptables -t mangle -F
iptables -X
systemctl mask firewalld
systemctl disable firewalld
systemctl stop firewalld
```

The database requires several ports to be open on the local network. Micro Focus does not recommend that you place a firewall between nodes (all nodes should be behind a firewall), but if you must use a firewall between nodes, ensure that all the database ports are available. For more information, see Firewall Considerations.

6. Set SELinux to permissive mode in `/etc/selinux/config`.

```
SELINUX=permissive
```

For more information, see SELinux Configuration.

7. Run the following command:

```
setenforce permissive
```

8. In `/etc/default/grub`, append the following lines:

```
GRUB_CMDLINE_LINUX="crashkernel=auto rhgb quiet intel_idle.max_cstate=0
processor.max_cstate=1 intel_pstate=disable"
```

```
grub2-mkconfig -o /boot/grub2/grub.cfg />
```

9. Depending on your OS (RHEL/CentOS 8.4), run the following command:

```
echo madvise > /sys/kernel/mm/transparent_hugepage/enabled
##? echo madvise > /sys/kernel/mm/redhat_transparent_hugepage/defrag
##> echo no > /sys/kernel/mm/redhat_transparent_
hugepage/khugepaged/defrag
### Changed: cpupower frequency-set --governor performance #### CentOS
only, resolve WARN (S0140/S0141)
```

10. Depending on your OS (RHEL/CentOS 8.4), run the following command:

```
myroot=`df -h | grep '/$' | awk '{print $1}'`
myopt=`df -h | grep '/opt' | awk '{print $1}'`

echo deadline > /sys/block/sdb/queue/scheduler    #### Resolve FAIL
(S0150)
/sbin/blockdev --setra 8192 $myopt                #### Resolve FAIL
(S0020)
/sbin/blockdev --setra 2048 $myroot
echo madvise > /sys/kernel/mm/transparent_hugepage/enabled
echo deadline > /sys/block/sda/queue/scheduler
tuned-adm profile throughput-performance #### CentOS only, resolve WARN
(S0140/S0141)
```

11. If you have a high-concurrency workload and if the database is CPU bound, reboot the virtual machine by running the following command; otherwise skip this step.

```
sudo sysctl -w net.core.netdev_max_backlog=2000
```

12. Run this command to ensure that `rng-tools` packages are installed in all cluster nodes:

```
sudo dnf install rng-tools -y
```

13. Set the UTC time for all cluster nodes:

```
sudo timedatectl set-timezone UTC
```

For CentOS 8.4, any changes to the timezone will require a cluster nodes reboot.

14. Reboot for your changes to take effect.
15. For RHEL/CentOS 8.x, you must run RHEL/CentOS 8.x using the following command:

```
dnf install libnsl
```

16. Modify the /etc/bashrc by running the following command:

```
export VERTICA_FAILURE_THRESHOLD=FAIL
```

17. Repeat these steps for each expected database node.

Configuring and Installing the Database Server

This section describes how to configure and install the Azure database.

• Creating the Azure Blob storage

1. Prepare the Azure Storage account.
   a. Log in to the Azure portal.
   b. From the left portal menu, select **Storage Accounts** to display a list of your storage accounts.
   c. On the **Storage Accounts** page, select **Create**.
   d. On the **Basics** tab, select **Resource Group**.
   e. Enter **Storage account name**, **Region**, and **Performance** as standard for storage account.
   f. On the **Advanced** tab, select Blob storage **Access tier** as **Hot**.
   g. On the **Networking** tab, select **Connectivity method** as **Public endpoint (all networks)** and **Routing preference** as **Microsoft network routing.**
   h. On the **Data Protection** tab, select **Enable soft delete for blobs** and set **Days to retain deleted blobs** to 7.
   i. On the **Tags** tab, enter values for **Owner** and **Product**.
   j. On the **Review + create** tab, verify the storage account you prepared and click **Create**.
2. Create a private endpoint to connect to Azure Storage Account.
   a. Log in to the Azure portal.
   b. Select the storage account you created in Step 1.
   c. Navigate to **Security + Networking** in the left menu and click **Networking**.
   d. Select **Private endpoint connections** and click **+Private endpoint**.
   e. On the **Basics** tab, select **Resource group** from the drop down list.
   f. Enter **Name** and **Region**.

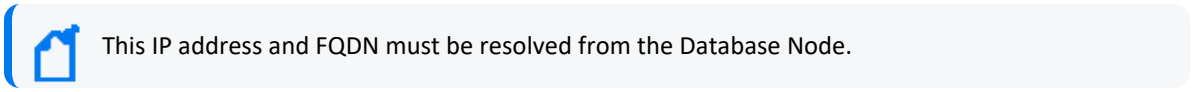
g. On the **Resource** tab, select **Target sub-resource** as Blob from the drop down list.

h. On the **Configuration** tab, select **Virtual network**, **subnet** from the drop down list.

i. Set **Integrate with private DNS zone** to **Yes**.

j. On the **Tags** tab, enter values for **Owner** and **Product**.

k. On the **Review + create** tab, verify the DNS configuration for private endpoint and click **Create**.



3. Create a Blob container in Azure Storage.

a. Log in to the Azure portal.

b. Select the storage account you created in Step 1.

c. Navigate to **Data Storage** in the left menu and click blob containers.

d. Click **+ Container**.

e. Enter a name for the new container.

f. Set the **Public access level**. Default is Private.

g. Click **Create** to create the container.

4. Collect Blob storage details.

a. Log in to the Azure portal.

b. Select the storage account you created in Step 1.

c. Navigate to **Security + Networking** in the left menu and click **Access Keys**.

d. On the **Show keys** tab, click **Rotate key** icon to generate a key.

e. Collect the **Storage account name** and the generated **key**.

f. Navigate to **Security + Networking** in the left menu and click **Networking**.

g. Click **Private endpoint connections**, the **Private endpoint** tab and then in the left menu click **DNS configuration**.

h. Make note of the FQDN and IP address.

i. Navigate to **Data Storage** in the left menu and select the container you created in Step 3.

j. Navigate to **Properties** in the left menu and collect the container URL.

- **Preparing the Virtual Machine for Installation**

1. (Conditional) Update CentOS.

   If you are deploying the database with CentOS 8.4 2105, you need to update the distros by running the commands below on all database nodes:

   ```
   sudo dnf --disablerepo '*' --enablerepo=extras swap centos-linux-repos
   centos-stream-repos
   sudo dnf distro-sync
   ```

   If the *distro* repository is broken, update the *--enablerepo* repositories from: https://www.centos.org/centos-stream/.

2. On the Database cluster node1 server, create a folder for the database installer:

   ```
   mkdir /opt/arcsight-db-tools
   ```

3. From the master node where you performed the Downloading Installation Packages steps, copy the following directory on the Database cluster node1 server:

   ```
   {unzipped-installer-dir}/installers/database/db-installer_x.x.x-x.tar.gz
   file to the /opt/arcsight-db-tools
   ```

4. To extract the installer file and place it in the correct directory, run the following commands:

   ```
   cd /opt/arcsight-db-tools
   tar xvfz db-installer_x.x.x.x.tar.gz
   ```

5. Navigate to the config directory. Edit the `db_user.properties` file and add the private IPs of the Azure VM nodes under the `host` parameter.

   For example:

   ```
   hosts=<IP node1>,<IP node2>,<IP node3>
   ```

- **Installing the Database not using Managed Authentication**

1. Navigate to the `/opt/arcsight-db-tools` folder.
2. Install the database:

   ```
   ./db_installer install
   ```

3. (Conditional) If the license file is not found, then enter y to continue using the community license.
4. When prompted, create the **database administrator** user and specify a password for it.

5. Specify the shard count. We recommend a shard count of 3 for single-node, or a count of 18 for multi-node to allow for scalability. The prompt options are based on your environment, single-node or multi-node.
6. Choose a communal storage type : 2 (Azure Blob storage)
7. Specify the account name: *As collected in Step 4*
8. Enter `N` to not use manage authentication.
9. Specify the account key: *As collected in Step 4*
10. Enter `y` to enable TLS.
11. Specify the Azure container for communal storage: *As collected in Step 4*
12. Specify a folder inside the container for communal storage.

- **Installing the Database using Managed Authentication**

1. Log in to the Azure portal.
2. Navigate to the database node 1 VM which you created.
3. Navigate to **Identity** in the left menu.
4. Switch on **System Assigned** and save the set up.
5. Repeat the two previous steps for all database node VMs.
6. Navigate to the resource group where the VM is located.
7. Navigate to **Access control (IAM)** in the left menu.
8. Click **Add Role Assignment**.
9. Select the role as **Storage Blob Data Owner**.
10. Click **Next**.
11. On the **Members** tab, set **Assigned access to** as **User, group or server principal**.
12. On **Select Members**, select your member.
13. Switch **Assigned access to** to **Managed Identity**.
14. Add your database VM as a member.
15. Click **Review + Assign**.
16. Navigate to the storage account.
17. Navigate to **Access control (IAM)** in the left menu.
18. Click **Add Role Assignment**.
19. Select the role as **Storage Blob Data Owner**.
20. Click **Next**.
21. On the **Members** tab, set **Assigned access to** as **User, group or server principal**.

22. On **Select Members**, select your member.
23. Switch **Assigned access to** to **Managed Identity**.
24. Add your database VM as a member.
25. Click **Review + Assign**.
26. On the Database cluster node1 server , navigate to the `/opt/arcsight-db-tools` folder.
27. Install the database:

    ```
    ./db_installer install
    ```

28. (Conditional) If the license file is not found, then enter y to continue using the community license.
29. When prompted, create the **database administrator** user and specify a password for it.
30. Specify the shard count. We recommend a shard count of 3 for single-node, or a count of 18 for multi-node to allow for scalability.
31. Choose a communal storage type: 2 (Azure Blob Storage)
32. Specify the account name: *As collected in Step 4*
33. Enter Y to use manage authentication.
34. Specify the Azure container for communal storage: *As collected in Step 4*
35. (Conditional) Specify a folder inside the container for communal storage.

- **Creating the Schema**

  To create the schema, run the following command:

  ```
  ./db_installer create-schema
  ```

  If you rotate the access keys to your Azure storage account after database installation, log in to the master node and run the following query to update the key credentials in the database:

  ```
  ALTER DATABASE DEFAULT SET AzureStorageCredentials = '
  [{"accountName":accountname", "accountKey":"new account key", "blobEndpoint":
  "blob endpoint"}]';
  ```

Configuring Elasticsearch Settings for Intelligence

This procedure applies only when you are deploying the Intelligence capability.

To ensure the Elasticsearch pods run after deployment and the Elasticsearch cluster is accessible:

1. Make an SSH connection from the jump host to an AKS node. For more information, see Formatting and Mounting the Attached Disk
2. Change to the following directory:

```
cd /etc/
```

3. In the `sysctl.conf` file, add the following:

```
vm.max_map_count=262144
```

4. Execute the following command:

```
sysctl -p
```

5. Repeat steps 1-4 on all the nodes.

Downloading the Installation Packages for an Azure Deployment

Use this procedure to download the packages for:

- **Installation:** Follow the Checklist: Planning to Deploy ArcSight Capabilities on Azureto ensure a successful installation.
- **Upgrade**: Follow the Checklist: Upgrading the Azure Deployment of ArcSight Platformto ensure a successful upgrade.

1. To identity the files to download to your secure network location, see Downloading and Installing the ArcSight Platform Capabilities in the Release Notes.
2. On a secure network location, download the installation packages for the CDF Installer and the product of your choice from the Software Licenses and Downloads portal .

This secure network location must be able to access Azure through the Azure Portal or the Azure Cloud Shell.

Installing CDF

1. Download the CDF deployer and the ArcSight metadata files to a secure network location.
2. SSH to your jump host and become `root`.
3. Upload the CDF deployer and ArcSight metadata files to a directory on the jump host.
4. Unzip the deployer and run the installation. For example:

```
unzip cdf-deployer.zip
```

```
...
```

```
cd cdf-deployer/
```

```
./install \
--nfs-server 10.1.2.4 \
--nfs-folder /nfs/itom-vol \
--registry-orgname your-org-name \
--registry-url srgdemo.azurecr.io \
--registry-username srgdemo \
--registry-password GEev87wtAW+FtBGTyADxgr9Fivg6a2gC \
--external-access-host installer.arcsight.private.com \
--noinfra \
--loadbalancer-info "azure-load-balancer-
internal=true;LOADBALANCERIP=10.1.1.101" \
--cloud-provider AZURE
```

The following arguments apply to the `install` command:

`--nfs-server`	Your NFS server private IP or NetApp end IP.
`--registry-url`	Login server (from the **Access keys** tab in your ACR resource)
`--registry-username`	Username (from the **Access keys** tab in your ACR resource)
`--registry-password`	Password (from the **Access keys** tab in your ACR resource)
`--registry-orgname`	Organization name. Use the same value as for the `-o` argument you specified during the uploading of your images to the ACR,
`--external-access-host`	DNS domain name configured earlier. For example, `installer.arcsight.private.com`. • If you use a different name for `private-dns` zone in the previous step, then change the value of `--external-access-host` to fit your DNS; that is, `installer.<YOUR NAME>` • You can verify this value on the Azure portal, in the main resource group, under **Private DNS Zone** resource.
`--cloud-provider`	Specifies the cloud provider when installing CDF on a cloud server. The allowed value of this parameter is AZURE.
`--loadbalancer-info`	Specifies load balancer information. The argument `azure-load-balancer-internal=true` is always required. The value of LOADBALANCERIP must be the value specified in Assigning an IP Address to Private DNS.

For a complete list of optional parameters, see CDF Installation CLI Commands.

Patching the Load Balancer

Before proceeding with annotating and patching the load balancer, execute the following command:
`kubectl get svc -A`

In the resulting output, ensure that frontend-ingress-controller-svc has an IP address assigned.

If the command is processing for a long time, it indicates that Kubernetes is unable to create an internal load balancer and assign an IP to it. The usual cause is missing necessary rights. Refer to the Prerequisites section and make sure all prerequisites are met before proceeding any further.

To annotate and patch the load balancer:

On the jump host, run the following commands.

```
kubectl annotate service -n core frontend-ingress-controller-svc
service.beta.kubernetes.io/azure-load-balancer-internal=true
```

```
kubectl patch services frontend-ingress-controller-svc -p  '{"spec":
{"type":"LoadBalancer","loadBalancerIP": "PUBLIC_IP"}}' -n core
```

Where `PUBLIC_IP` is the value of the public IP you assigned previously.

Configuring the Load Balancer

As part of load balancer configuration, to permit access to the 5443 port for product deployment, you need to add the following to the AKS load balancer:

- A health probe and load balancing rule for port 5443
- A health probe and load balancing rule for port 433

These steps are explained below.

To add a health probe for port 5443 using the Azure Portal:

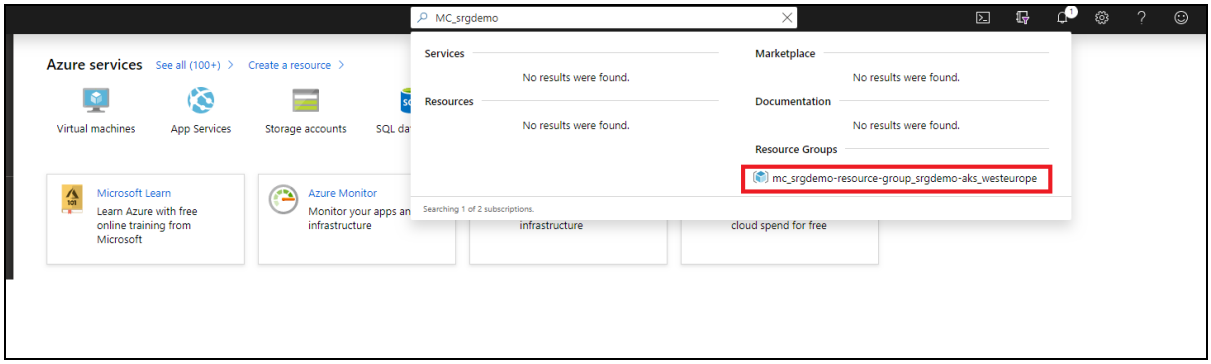
This probe may already exist if capabilities have been deployed previously. If the health probe does not exist, add it. If it already exists, then verify that the backend port is correct.

1. On your jump host, run the following command to get the value of `portal-ingress-controller-svc` for port 5443:
   `kubectl get svc -n core | grep portal-ingress-controller-svc`

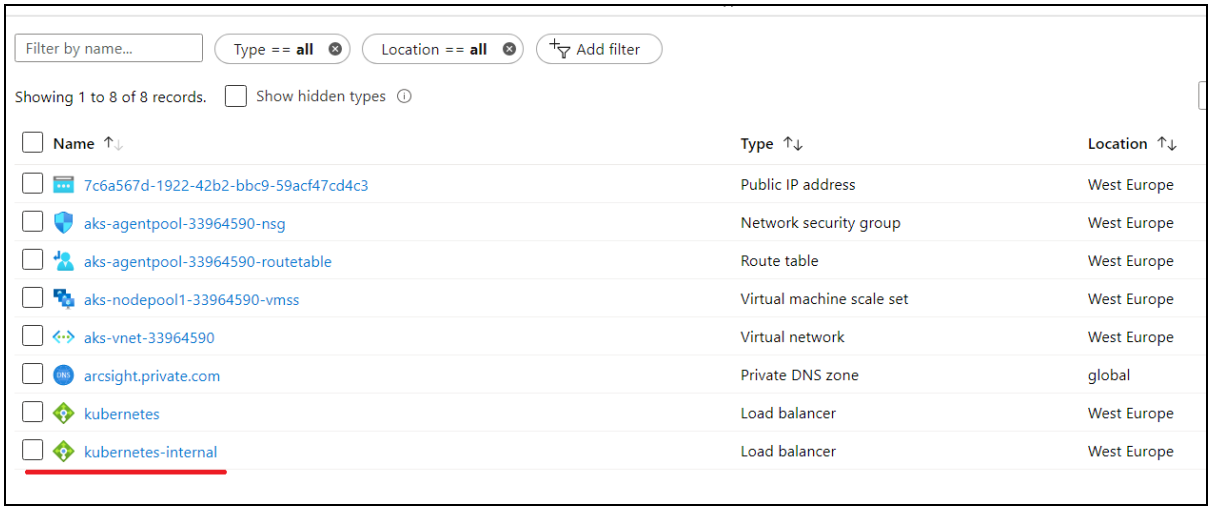
Example output, showing NodePort as 31249:

```
portal-ingress-controller-svc NodePort 10.0.146.63
5443:31249/TCP,5444:31036/TCP 21m
```

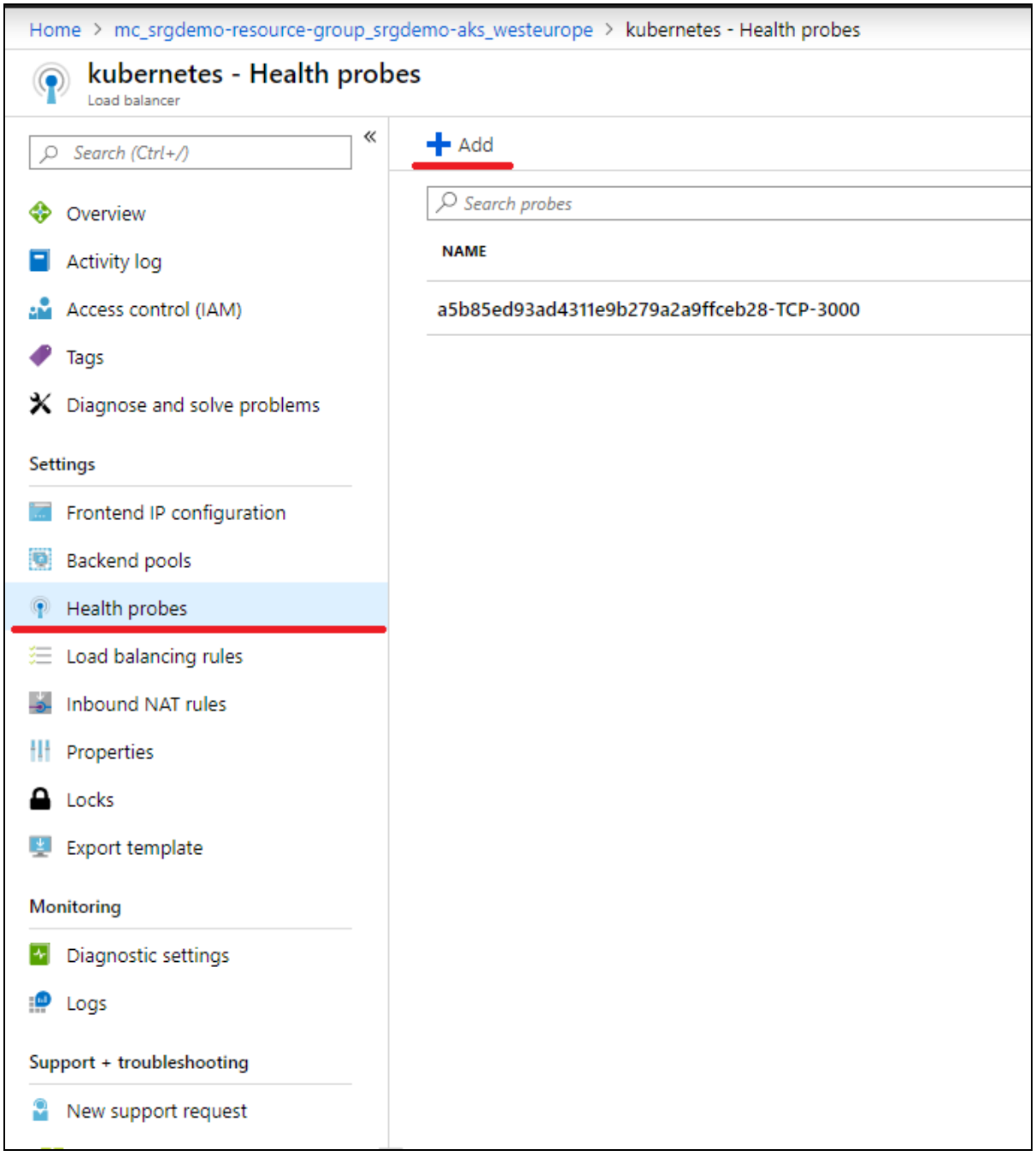
2. Open the Azure Portal and locate the Azure Kubernetes resource group. (The AKS resource group name is in format MC_<your_resource_group>_<aks_name>_<location>.)



3. Open the Kubernetes resource group.
4. Locate the Kubernetes load balancer, and then open it.



5. On the Kubernetes load balancer resource, click **Health probes.**
6. Add a health probe for 5443. Port Value should be the value obtained for the service NodePort in step 1.



To add a health probe for port 5443 using the Azure Cloud Shell:

1. Get the AKS resource group and store it in an environment variable for later usage:
   `CLUSTER_RESOURCE_GROUP=$(az aks show --resource-group <RESOURCE GROUP> --name <AKS NAME> --query nodeResourceGroup -o tsv)`

For example, for AKS `srg-demo-aks` from resource group `srg-demo`:
`CLUSTER_RESOURCE_GROUP=$(az aks show --resource-group srg-demo --name srg-demo-aks --query nodeResourceGroup -o tsv)`

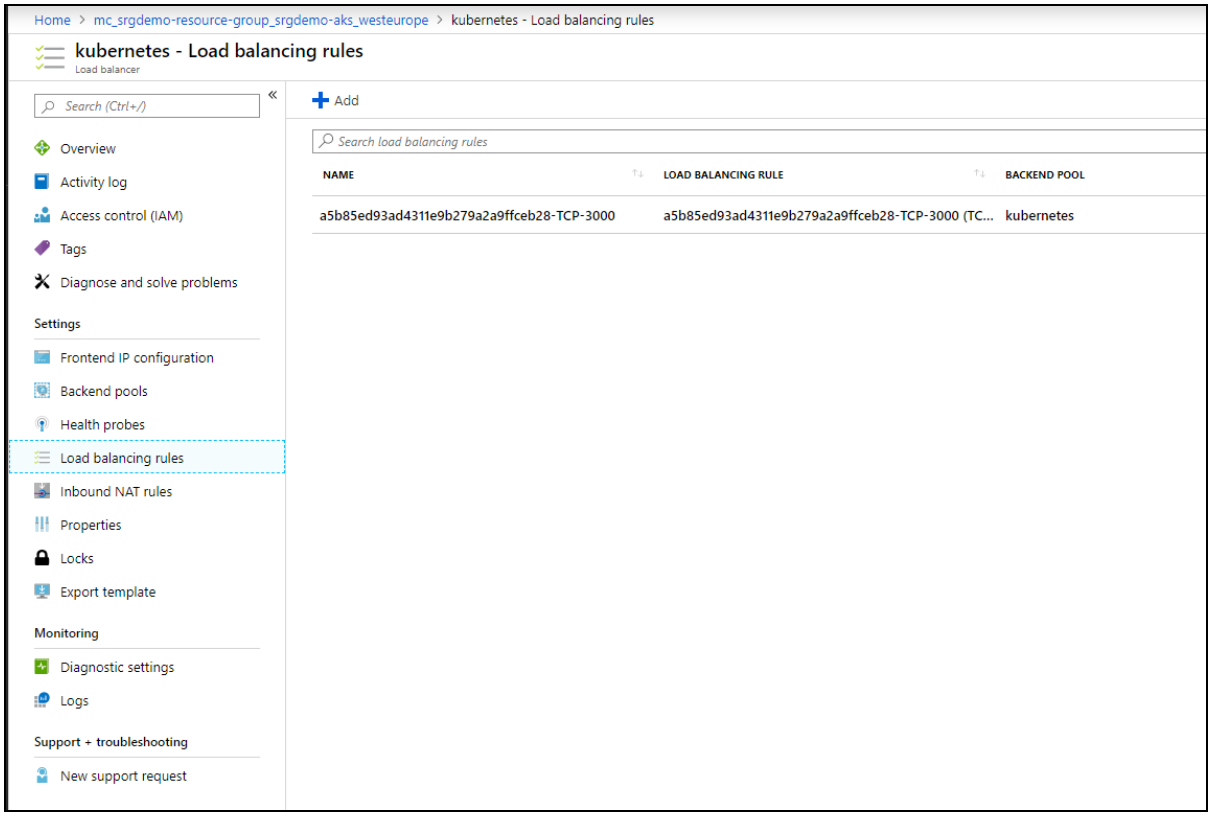
2. Create the health probe by running the command:
   az network lb probe create -g $CLUSTER_RESOURCE_GROUP --lb-name kubernetes-internal -n 5443-hp --protocol tcp --port <NODE PORT>

<NODE PORT> is the value obtained for the service NodePort in step 1

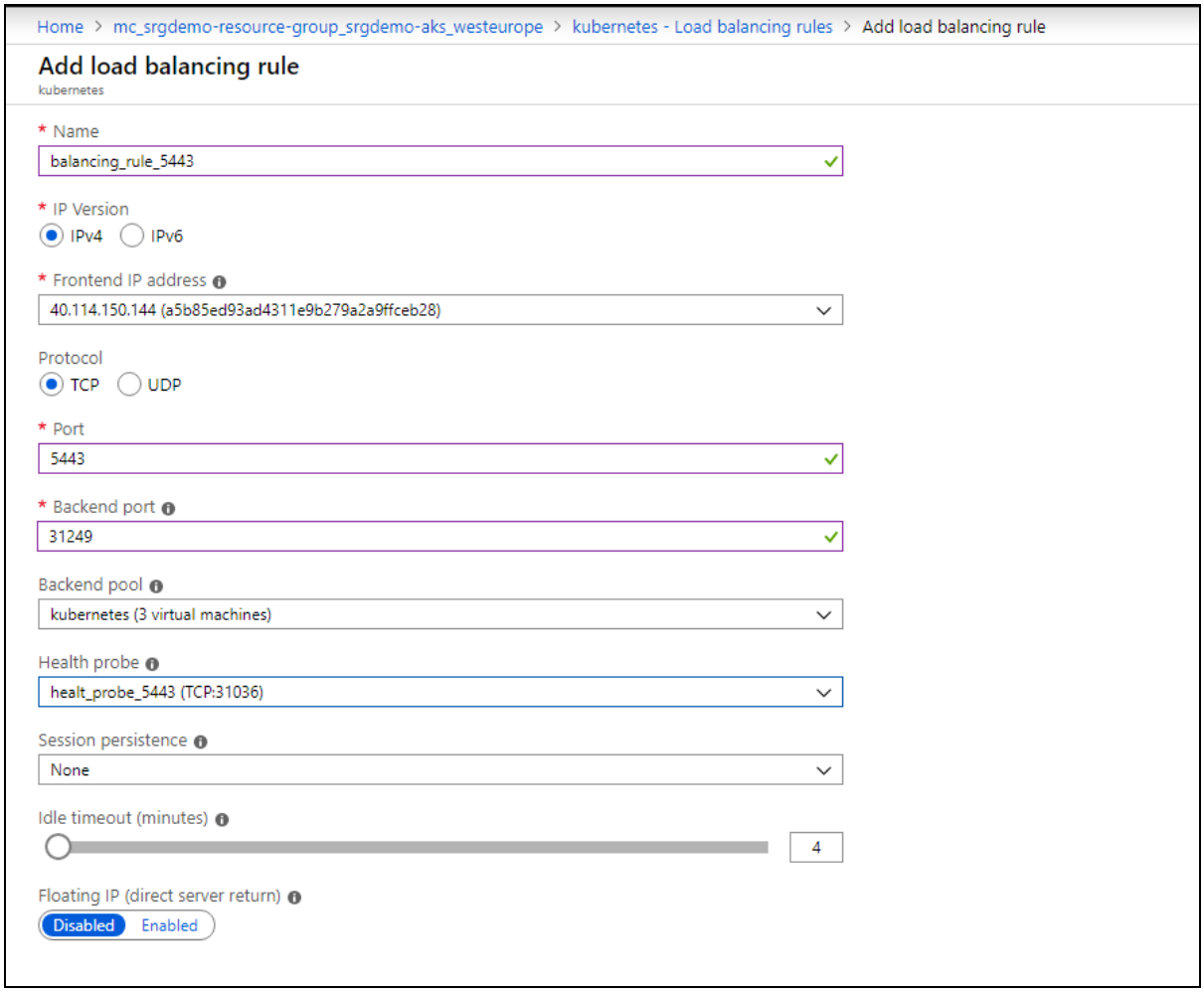
```
Example:

az network lb probe create -g $CLUSTER_RESOURCE_GROUP --lb-name kubernetes-internal -
n 5443-hp  --protocol tcp --port 31249
```

To add a load balancing rule for port 5443 using the Azure Portal:

1. Open the Kubernetes load balancer, and then click **Load balancing rules.**



2. Add a rule for port 5443. The backend port is the value for `portal-ingress-controller-svc` obtained previously and the health probe you just created.



To add a load balancing rule for port 5443 using the Azure Cloud Shell:

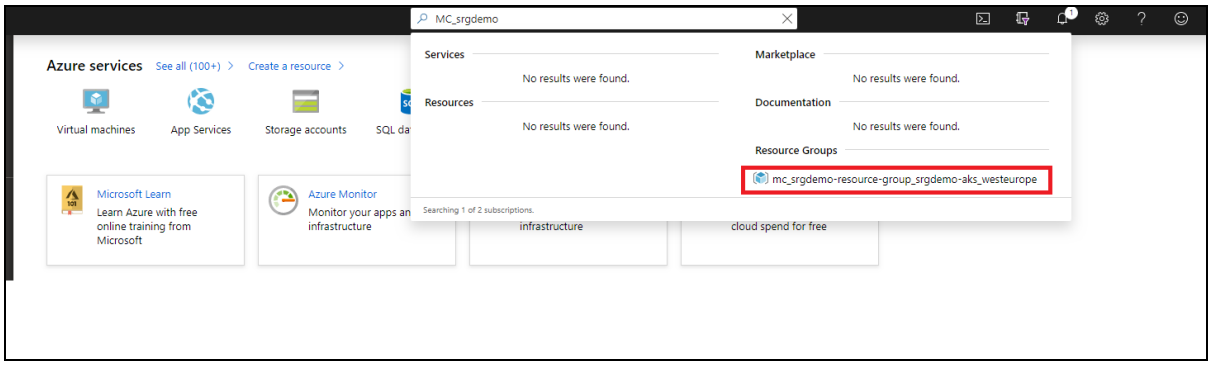
1. Run the following command:
   ```
   az network lb rule create -g <AKS RESOURCE GROUP> --lb-name kubernetes-internal -n 5443-lb-rule --protocol Tcp --frontend-port 5443 --backend-port <SERVICE PORT> --probe-name 5443-hp --backend-pool-name kubernetes
   ```

For example:

```
az network lb rule create -g mc_srg-demo_srg-demo-aks_westeurope --lb-name kubernetes-internal -n 5443-lb-rule --protocol Tcp --frontend-port 5443 --backend-port 31249 --probe-name 5443-hp --backend-pool-name kubernetes
```

To add a health probe for port 443 using the Azure Portal:

1. In the Azure portal, locate the Azure Kubernetes resource group. (The AKS resource group name is in the format `MC_<your_resource_group>_<aks_name>_<location>`.)



2. Open the Kubernetes resource group.
3. On the Kubernetes load balancer resource, click **Health probes.**
4. Click **+ Add** for Kubernetes load balancer health probes and specify values for the following:
   - **Name:** Assign a name to the probe.
   - **Protocol:** Select TCP.
   - **Port:** Specify 443.

To add a health probe for port 443 using the Azure Cloud Shell:

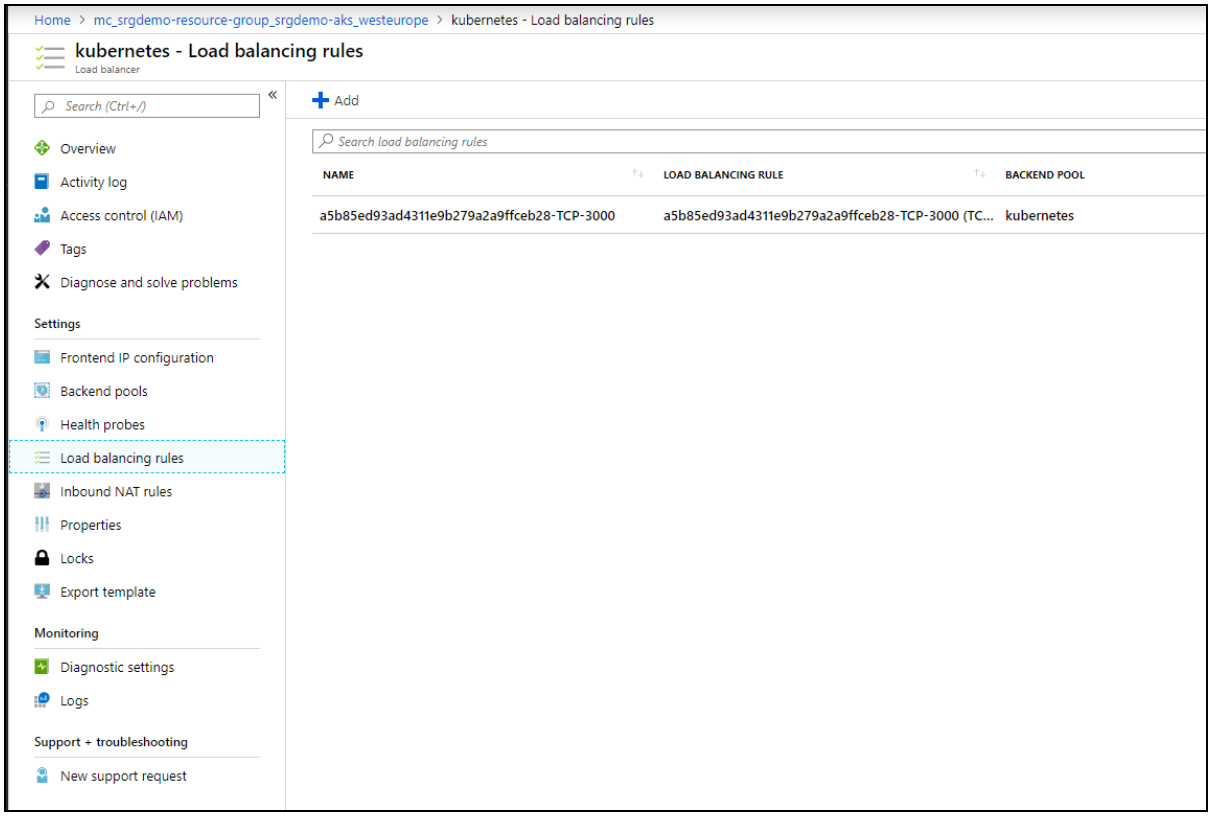
1. Run the following command:
   `az network lb probe create -g <AKS RESOURCE GROUP> --lb-name kubernetes-internal -n 443-hp --protocol tcp --port 443`

For example:

```
az network lb probe create -g mc_srg-demo_srg-demo-aks_westeurope --lb-name kubernetes-internal -n 443-hp --protocol tcp --port 443
```

To add a load balancing rule for port 443 using the Azure Portal:

1. Open the Kubernetes load balancing rule and click Load balancing rules.
2. Click **+ Add** for the Kubernetes load balancer load balancing rules and specify values for the following:
   - **Name:** assign a name to the probe.
   - **Port:** Specify 443.
   - **Backend port:** Specify 443.
   - **Health probe:** Select the probe you previously created for port 443.
   - **Session Persistence:** select Client IP and Protocol.
3. Open the Kubernetes resource group.



To add a load balancing rule for port 443 using the Azure Cloud Shell:

1. Run the following command:
   ```
   az network lb rule create -g <AKS RESOURCE GROUP> --lb-name kubernetes-internal -n 443-lb-rule --protocol Tcp --frontend-port 443 --backend-port 443 --probe-name 443-hp --backend-pool-name kubernetes --load-distribution SourceIPProtocol
   ```

For example:

```
az network lb rule create -g mc_srg-demo_srg-demo-aks_westeurope --lb-name kubernetes-internal -n 443-lb-rule --protocol Tcp --frontend-port 443 --backend-port 443 --probe-name 443-hp --backend-pool-name kubernetes --load-distribution <SourceIPProtocol>
```

Securing External Communication with the RE Certificate

At the center of the Platform is a Kubernetes cluster where communication occurs between pods within the cluster and with non-containerized ArcSight components outside of the cluster. In order to ensure secure trusted communication between pods within the cluster and components outside of the cluster, encrypted communication with client certificate authentication is configured by default.

Understanding the ArcSight Platform Certificate Authorities

During installation, three self-signed Certificate Authorities (CA) are created automatically, two for signing certificates used exclusively for pod to pod communication within the cluster (RIC and RID CA), and the other for signing certificates for each pod that performs communication external to the cluster (RE CA). Only pods that perform external communication have a certificate that is signed by the external CA.

External cluster communication occurs not only with ArcSight components, but also with user web browsers and, in some cases, user clients of ArcSight APIs (such as the REST API). By default, when the user connects to the cluster, they will be presented with a certificate that has been signed by the self-signed external CA. Since the external CA is self-signed, the user's connection will not automatically trust the certificate because it will not be verifiable using a certificate chain that is already in the user's trust store.

To give users confidence they are connecting to the trusted cluster, we recommend signing the certificates that are presented to the user with a CA that is trusted by the user's trust store. There are two approaches to doing this that are described in the documentation below. These approaches are:

Method 1 - Signing the RE External Communication Certificate with Your Trusted Certificate Authority

This is the recommended approach, because it is theoretically more secure than the other approach, in that, it only involves transferring a CSR and public certificate between systems, which does not put any private secrets at risk.

Method 2 - Importing an Externally Created Intermediate CA

This approach involves creating an Intermediate CA (key and certificate pair) in a system outside of the ArcSight Platform, and then importing it into the ArcSight Platform. While this approach does work, it is theoretically less secure than the other approach, because it involves transferring a CA private key between systems, which potentially exposes it to unintended parties.

Use only one of the two approaches above.

Using an RE External Communication Certificate Signed by Your Trusted Certificate Authority

Use only one of the two approaches below. The first one, "Signing the RE External Communication Certificate with Your Trusted Certificate Authority" approach is recommended for the reasons described in Understanding the ArcSight Platform Certificate Authorities.

Method 1 - Signing the RE External Communication Certificate with Your Trusted Certificate Authority

Signing the RE External Communication Certificate with Your Trusted Certificate Authority approach is recommended for the reasons described in Understanding the ArcSight Platform Certificate Authorities.

In order to sign the RE external communication certificate with your trusted CA, you need to (1) create a certificate signing request (CSR) from vault, (2) take it to your organization, (3) sign it, and (4) return the signed CSR and all the public chain-of-certificates used to sign it.

1. Export the following access token dependencies (you can remove these later if not needed):

```
export CDF_APISERVER=$(kubectl get pods -n core -o custom-
columns=":metadata.name"| grep cdf-apiserver)
```

```
export PASSPHRASE=$(kubectl get secret vault-passphrase -n core -o json
2>/dev/null | jq -r '.data.passphrase')
```

```
export ENCRYPTED_ROOT_TOKEN=$(kubectl get secret vault-credential -n core
-o json 2>/dev/null | jq -r '.data."root.token"')
```

```
export VAULT_TOKEN=$(echo ${ENCRYPTED_ROOT_TOKEN} | openssl aes-256-cbc -
md sha256 -a -d -pass pass:"${PASSPHRASE}")
```

2. Ask vault to generate the CSR by running the following command:

Important: When you execute this command, proceed expeditiously through steps 3 and 4, as your cluster will not be able to issue external certificates while it waits for the CSR to be signed.

```
kubectl exec -it -n core ${CDF_APISERVER} -c cdf-apiserver -- bash -c
"VAULT_TOKEN=$VAULT_TOKEN vault write -tls-skip-verify -format=json
RE/intermediate/generate/internal common_name=\"none-MF CDF RE CA on
<FQDN of ArcSight Platform Virtual IP for HA or single master node>\"
```

```
country=<Country> locality=<Locality> province=<Province>
organization=<Organization> ou=<Organizational Unit>" | jq -r '.data.csr'
> /tmp/pki_intermediate.csr
```

Note: The `common_name` in the command above is an example common name. Substitue your own values for the common name to fit your environment. Additionally, your trusted certificate authority might require additional parameters in the CSR besides common_name. Ask your PKI team for what the required CSR parameters are and add the appropriate parameters to the command (similar to how the parameter common_name is specified). The parameter names for the vault command used above are documented at https://www.vaultproject.io/api-docs/secret/pki#generate-intermediate

3. Sign the CSR file with your trusted certificate authority, and save the result into the `intermediate.cert.pem` file.

   Examply only. A basic example is provided below. Your environment will likely be different.

```
openssl ca -keyfile your-rootca-sha256.key -cert your-rootca-sha256.crt -
config your-openssl-configuration-file -extensions v3_ca -notext -md
sha256 -in /tmp/pki_intermediate.csr -out intermediate.cert.pem
```

   Make sure the `v3_ca` extension is enabled and a new certificate is useable as a certificate authority on its own. Otherwise, you will receive a warning in the next step that given certificates are not marked for CA use.

4. Create an `intermediate.chain.pem` file that includes the combination of the `intermediate.cert.pem`, the public certificate of your trusted certificate authority, and all intermediate public certificates in the chain between them so that `intermediate.chain.pem` includes the full trust chain.

```
cp intermediate.cert.pem intermediate.chain.pem
cat [parent-intermediate1.crt] [parent-intermediate2.crt] [...] your-
rootca-sha256.crt >> intermediate.chain.pem
```

   If you have intermediate certificates between your `intermediate.cert.pem` and your trusted certificate authority, you must add the certificates in the specific order of the sequence of the chain, with the last certificate being the certificate of the root trusted CA.

5. Import the `intermediate.chain.pem` file into the cluster vault:

```
chaincerts=$(cat intermediate.chain.pem) && kubectl exec -it -n core
${CDF_APISERVER} -c cdf-apiserver -- bash -c "VAULT_TOKEN=$VAULT_TOKEN
vault write -tls-skip-verify -format=json RE/intermediate/set-signed
certificate=\"${chaincerts}\""
```

6. Update `ConfigMap RE_ca.crt` by running these commands:

```
reCrtForJson=$(sed -E ':a;N;$!ba;s/\r{0,1}\n/\\n/g'
intermediate.chain.pem) && kubectl patch configmap -n core public-ca-
certificates -p "{\"op\": \"replace\", \"data\":{\"RE_
ca.crt\":\"${reCrtForJson}\"}}"
```

```
ARCSIGHT_NS=$(kubectl get namespaces --no-headers -o custom-
columns=":metadata.name" | grep arcsight-installer)
```

```
if [ -n "$ARCSIGHT_NS" ];then reCrtForJson=$(sed -E ':a;N;$!ba;s/\r
{0,1}\n/\\n/g' intermediate.chain.pem); kubectl patch configmap -n
$ARCSIGHT_NS public-ca-certificates -p "{\"op\": \"replace\", \"data\":
{\"RE_ca.crt\":\"${reCrtForJson}\"}}";fi
```

7. (Conditional) If you already deployed ArcSight Capabilities onto the CDF, update the ArcSight Capabilities to use the updated RE external communication certificate, by following the instructions in Configuring ArcSight Kubernetes Pods to Use the Updated RE External Communication Certificate.

   If you deployed CDF but have not yet deployed any ArcSight Capabilities, you can skip those instructions.

Method 2 - Importing an Externally Created Intermediate CA

This is an alternate approach for signing certificates to connect to the trusted cluster. Before choosing this approach, ensure that your understand the other approach recommended in Understanding the ArcSight Platform Certificate Authorities.

To import an externally created intermediate CA:

1. Obtain an intermediate CA (key and certificate pair) from your trusted certificate authority.
   a. Name the certificate files as follows:
      - key file: `intermediate.key.pem`
      - certificate file: `intermediate.cert.pem`
   b. Obtain the root CA certificate (including chain), and put it in a file named ca.cert.pem.
2. Replace the existing RE CA in the ArcSight Platform with the intermediate CA you obtained in the step above, based on your type of deployment, on-premises **or** cloud.
   a. Change the directory:
      - For an on-premises deployment, run these commands:

```
cd /opt/arcsight/kubernetes/scripts/
```

- For a cloud deployment, run these commands:

```
cd {path to cdf installer}/cdf-deployer/scripts/
```

b. Run the following command to replace the existing RE CA:

```
./cdf-updateRE.sh write --re-crt=/pathto/intermediate.cert.pem --re-
key=/pathto/intermediate.key.pem [--re-ca=/pathto/ca.cert.pem]
```

> **Note:** `--re-ca=/pathto/ca.cert.pem` is the path to the file containing the certificate of CA used to sign re-crt. It is not required when re-crt is self-signed or CA is included in re-crt.

3. (Conditional) If you already deployed ArcSight Capabilities onto CDF, proceed to the next section to update the ArcSight Capabilities to use the updated RE external communication certificate, Configuring ArcSight Kubernetes Pods to Use the Updated RE External Communication Certificate.

   However, if you have only deployed CDF, but have not deployed ArcSight Capabilities yet, you can skip that section.

Configuring the Kubernetes Cluster

After you install the CDF Installer, complete the following steps to configure the cluster.

1. Use your remote desktop to access the jump host.
2. Browse to the cluster using your private DNS address at port 3000.

   For example:

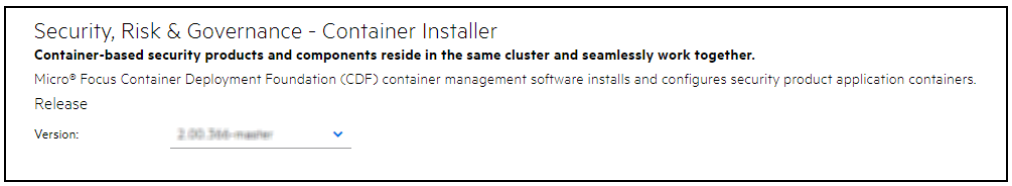
```
https://installer.private.arcsight.com:3000
```

3. Log in using **admin** (user ID) and the password you specified during the CDF installation.

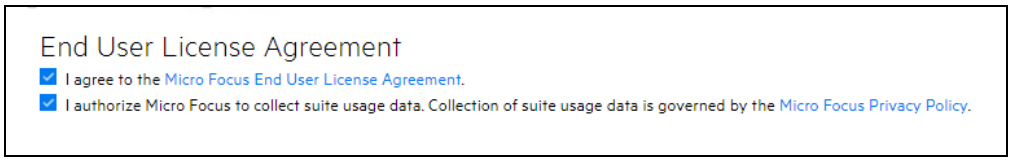
   The system prompts you to upload the following ArcSight installer metadata.tar file:

```
arcsight-installer-metadata-<version>.xx.tar
```

4. On the **Security Risk & Governance - Container Installer** page:
   a. Select the CDF base product metadata **version**.
   b. Click **Next**.



5. On the **End User License Agreement** page:
   a. Review the End User License Agreement.
   b. To accept the agreement, select the **I agree…** check box.
   c. (Optional) To have information passed to Micro Focus, select the **I authorize...** check box.
   d. Click **Next**.

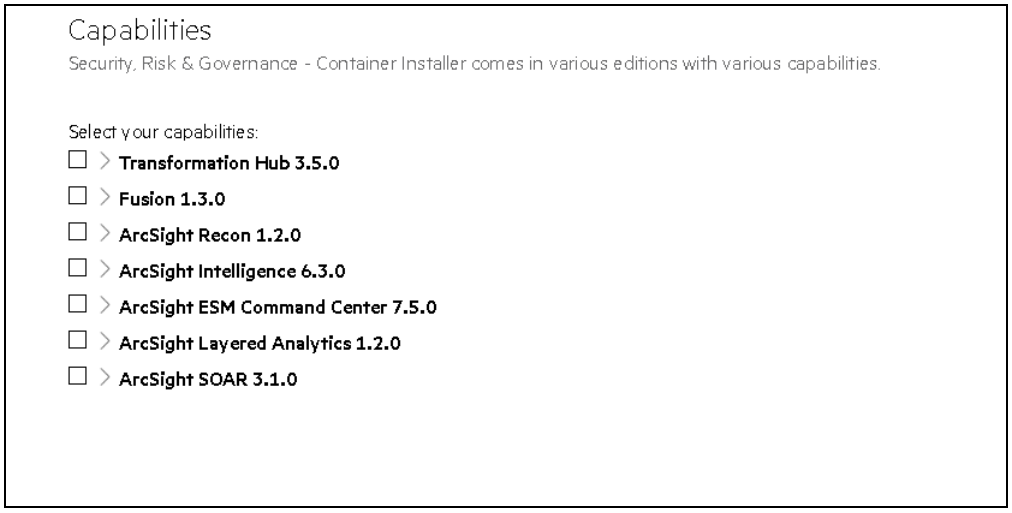


6. On the **Capabilities** page:
   a. Select the capabilities and products you want to install.

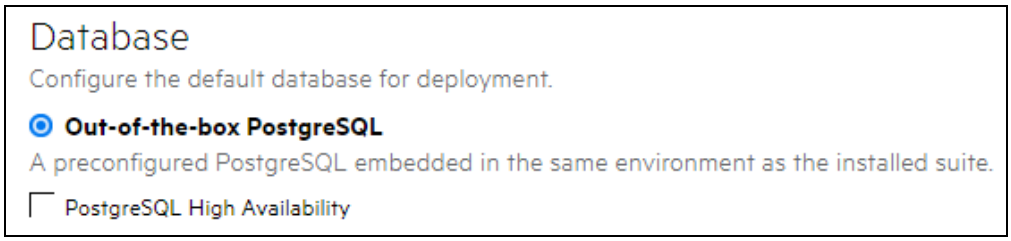
   For example, to install Transformation Hub as a standalone install, select the **Transformation Hub** check box.

   Other products might require Transformation Hub or other capabilities as prerequisites. You can view any such requirements in the pull-down text associated with the capability.

   b. To show additional information associated with the product, click the **>** (greater than) arrow.
   c. Click **Next**.



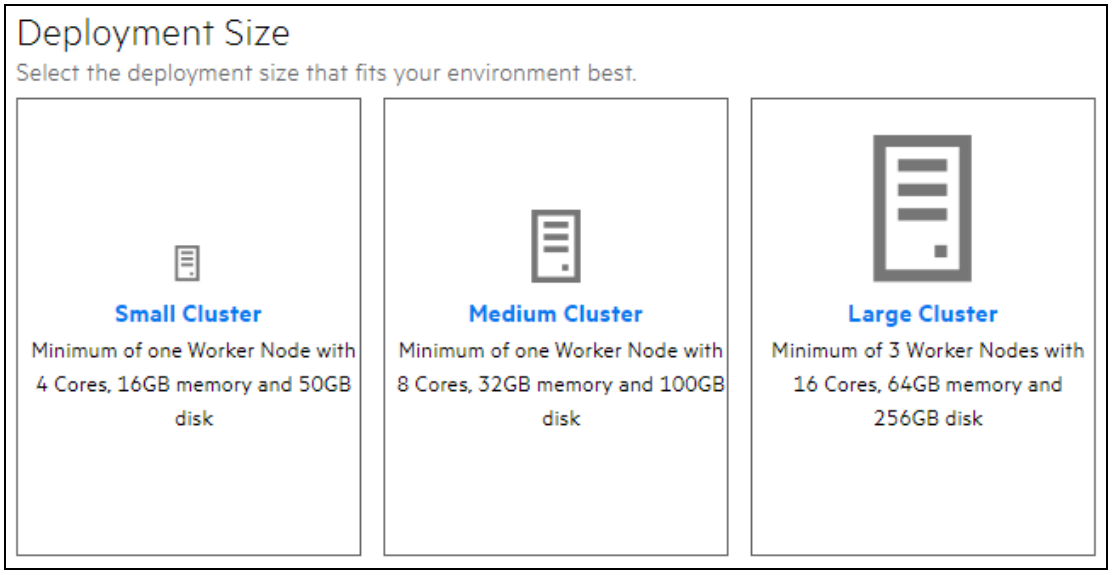
7. On the **Database** page:
   a. Ensure the **PostgreSQL High Availability** box is *unselected*. This database is not used by capabilities in SODP.
   b. To continue, click **Next**.



8. On the **Deployment Size** page:
   a. Based on your planned implementation, select a size for your deployment. (You can configure additional nodes, each running on their own host systems, in subsequent steps.)

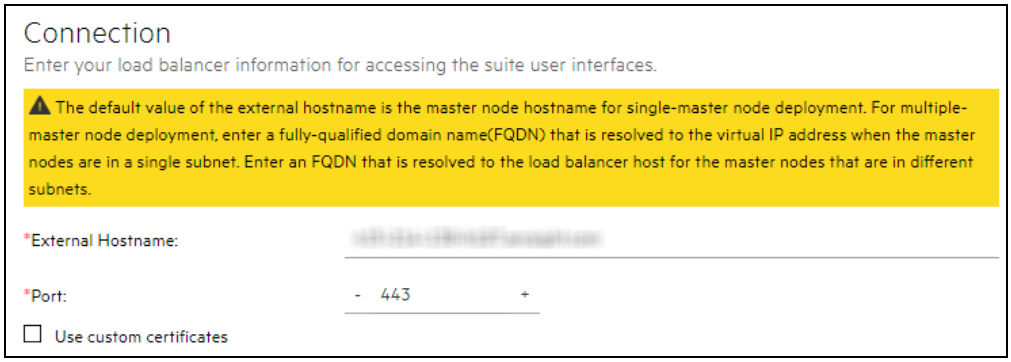
> The installation will be halted if your environment does not meet the minimal hardware requirements for the deployment.

Size	Minimum Worker Nodes	Cores	Memory	Disk
Small Cluster	1 Worker Node	4 Cores	16 GB	50 GB
Medium Cluster	1 Worker Node	8 Cores	32 GB	100 GB
Large Cluster	3 Worker Nodes	16 Cores	65 GB	256 GB



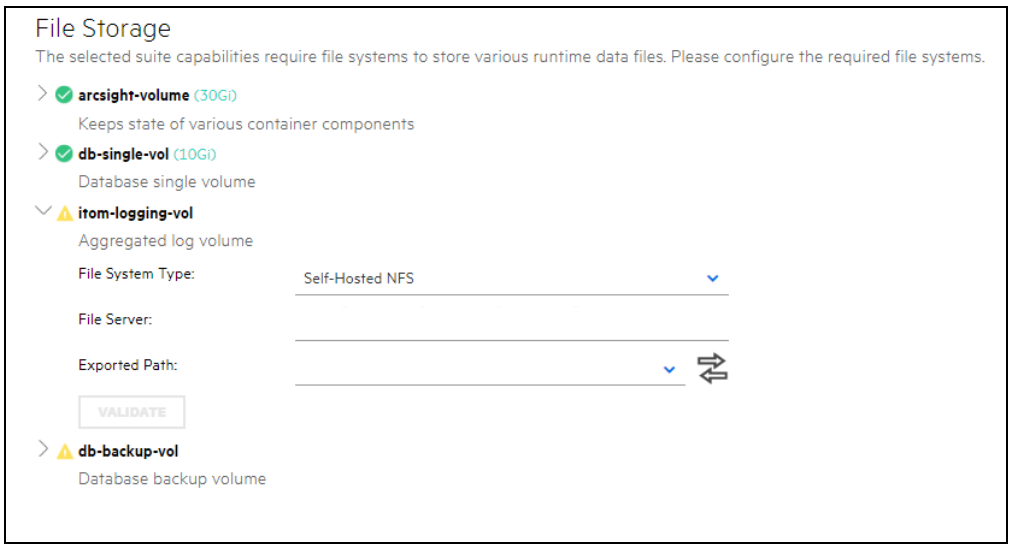
   b. Click **Next**.

9. On the **Connection** page:
   a. In **External Hostname**, the deployment populates an external hostname automatically from either:
      - The Virtual IP (VIP) specified earlier during the install of CDF (`--ha-virtual-ip parameter`)
      - The master node hostname if the `--ha-virtual-ip` parameter was not specified during CDF installation
   b. Confirm the port is correct.
   c. To continue, click **Next**.



10. On the **File Storage** page, for each NFS volume to configure:
    a. In the **File System Type** drop-down, ensure **Managed NFS** is selected.
    b. In **File Server,** specify the IP address or FQDN for the NFS server.
    c. From the **Exported Path** drop-down, select the appropriate volume. (If using NetApp, specify the path manually instead; for example, `/nfs/arcsight-volume, /nfs/db-backup-vol, /nfs/db-single-vol, /nfs/itom-logging-vol itom-vol`.)
    d. Click **Validate**. All volumes must validate successfully to continue with the installation. The following volumes must be available on your NFS server.

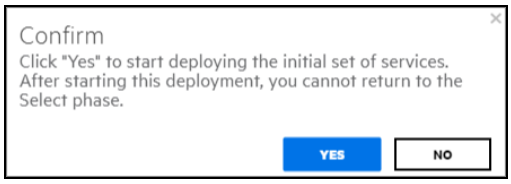
CDF NFS Volume Claim	Your NFS volume
`itom-vol`	`<NFS_ROOT_FOLDER>/itom_vol`
`db-single-vol`	`<NFS_ROOT_FOLDER>/db-single-vol`
`db-backup-vol`	`<NFS_ROOT_FOLDER>/db-backup-vol`
`itom-logging-vol`	`<NFS_ROOT_FOLDER>/itom-logging-vol`
`arcsight-volume`	`<NFS_ROOT_FOLDER>/arcsight-volume`



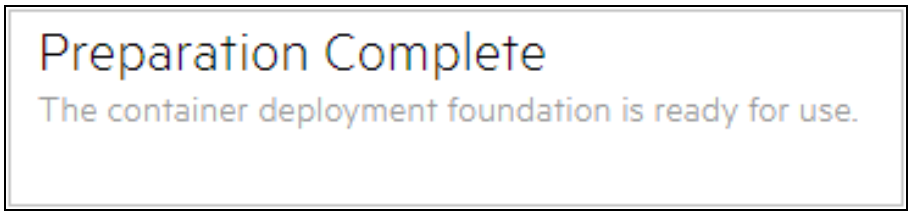
e. Click **Next**.

After you click **Next**, the infrastructure implementation is deployed. *Please ensure that your infrastructure choices are adequate to your needs.* An incorrect or insufficient configuration might require a reinstall of all capabilities.

11. On the **Confirm** dialog, to start deploying the nodes, click **Yes**.



After all nodes have been configured, and all services have been started on all nodes, the **Preparation Complete** page displays. You are now ready to configure product-specific installation attributes.



12. To configure the products and components of the deployment, click **Next**.

Completing the Database and Kafka Scheduler Setups

This section details the process for completing the database and Kafka Scheduler setups for both on-premises and cloud deployments.

Gathering Certificates for the Kafka Scheduler Setup

The database and deployed capabilities need to establish a trusted connection. To do so, generate the key pair for the Kafka Scheduler.

This step is required even if you use non-SSL communication between the Kafka Scheduler and Transformation Hub, because the schema registry is always SSL-enabled.

1. Run these commands on your database node1 to generate the Kafka Scheduler private key file `kafkascheduler.key.pem` and the certificate signing request file `kafkascheduler.csr.pem`:

   ```
   cd <yourOwnCertPath>/
   ```

   If you installed using the ArcSight Platform Installer, the default location is `/opt/arcsight-db-tools/cert/`

   ```
   openssl req -nodes -newkey rsa:2048 -keyout kafkascheduler.key.pem -out
   kafkascheduler.csr.pem -subj "/C=US/ST=State/L=City/O=Company
   Inc./OU=IT/CN=kafkascheduler"
   ```

2. Copy the certificate signing request `kafkascheduler.csr.pem` to your cluster, bastion host, or jump host.
3. Run the following commands on your cluster or your bastion host to sign the certificate signing request using your cluster RE certificate:

   ```
   export CDF_APISERVER=$(kubectl get pods -n core -o custom-
   columns=":metadata.name"| grep cdf-apiserver)
   ```

   ```
   export PASSPHRASE=$(kubectl get secret vault-passphrase -n core -o json
   2>/dev/null | jq -r '.data.passphrase')
   ```

   ```
   export ENCRYPTED_ROOT_TOKEN=$(kubectl get secret vault-credential -n core
   -o json 2>/dev/null | jq -r '.data."root.token"')
   ```

   ```
   export VAULT_TOKEN=$(echo ${ENCRYPTED_ROOT_TOKEN} | openssl aes-256-cbc -
   md sha256 -a -d -pass pass:"${PASSPHRASE}")
   ```

```
export COMMON_NAME=kafkascheduler
```

```
export CSR=$(cat ${COMMON_NAME}.csr.pem)
```

```
WRITE_RESPONSE=$(kubectl exec -it -n core ${CDF_APISERVER} -c cdf-
apiserver -- bash -c "VAULT_TOKEN=$VAULT_TOKEN vault write -tls-skip-
verify -format=json RE/sign/coretech csr=\"${CSR}\"") && \
```

```
echo "${WRITE_RESPONSE}" | jq -r ".data | .certificate" > ${COMMON_
NAME}.crt.pem && \
```

```
echo "${WRITE_RESPONSE}" | jq -r ".data | if .ca_chain then .ca_chain[]
else .issuing_ca end" > issue_ca.crt
```

4. Copy the RE signed certificate file `kafkascheduler.crt.pem` to database node1 <*yourOwnCertPath*>.
5. Copy the `issue_ca.crt` to database node1 <*yourOwnCertPath*>.

Enabling the Database to Receive SSL Connections

The following procedures are recommended for data privacy, but they are optional. Perform the first two procedures below on database node1.

- Creating the Database Server Key and Certificate
- Setting up the Database SSL Configuration
- "Configuring Deployed Capabilities to Use SSL for Database Connection" on page 360

Creating the Database Server Key and Certificate

Follow these steps to generate database CAs and certificates:

1. Log in to database node1 as root.
2. Change to your own certificates directory path:

```
cd <yourOwnCertPath>
```

> For deployment with `arcsight-platform-instsaller`, the default location is `/opt/arcsight-db-tools/cert/`

3. Run this command to create a certificate authority (CA) for the database:

```
openssl req -newkey rsa:4096 -sha256 -keyform PEM -keyout generated-db-
ca.key -x509 -days 3650 -outform PEM -out generated-db-ca.crt -subj
```

```
"/C=US/ST=State/L=City/O=Company
Inc./OU=IT/CN=Database/emailAddress=admin@microfocus.com" -nodes
```

4. Run this command to create the database server key:

```
openssl genrsa -out generated-db-server.key 4096
```

5. Create the database server certificate signing request by running the following command:

```
openssl req -new -key generated-db-server.key -out generated-db-
server.csr -subj "/C=US/ST=State/L=City/O=Company
Inc./OU=IT/CN=DatabaseServer/emailAddress=admin@microfocus.com" -nodes -
sha256
```

6. Sign the Certificate Signing Request with self-signed CA by running the following command:

```
openssl x509 -req -in generated-db-server.csr -CA generated-db-ca.crt -
CAkey generated-db-ca.key -CAcreateserial -extensions server -days 3650 -
outform PEM -out generated-db-server.crt -sha256
```

Setting up the Database SSL Configuration

These steps will update the SSL configuration in the database.

1. Move the following files to database node1 <*yourOwnCertPath*> as root by running these commands:

This step is only required for a new database installation. You can skip this step if this is an upgrade and the files are already there.

```
cd <yourOwnCertPath>/
ls <yourOwnCertPath>/
```

- The output should have the following files:
  - `generated-db-ca.crt`
  - `generated-db-server.crt`
  - `generated-db-server.key`
  - `generated-db-ca.key`
  - `generated-db-ca.srl`
  - `generated-db-server.csr`
  - `issue_ca.crt`

- `kafkascheduler.crt.pem`
- `kafkascheduler.key.pem`

2. For chained CAs, run the commands to split the CAs into individual files:

```
cat issue_ca.crt | awk 'BEGIN {c=0;} /BEGIN CERT/{c++} { print > "issue_
ca_part." c ".crt"}'
```

```
chown -R dbadmin:dbadmin <yourOwnCertPath>
```

3. Run the following commands on database node1 to update the database SSL configuration:

```
cd /opt/arcsight-db-tools
```

```
./db_ssl_setup --disable-ssl
```

If the attempt fails, drop the certificate manually by running the three commands below:

```
sudo su - dbadmin
```

```
vsql -U <dbadminuser> -w <dbadminpassword> -c "ALTER TLS CONFIGURATION
server CERTIFICATE NULL;"
```

```
vsql -U <dbadminuser> -w <dbadminpassword> -c "DROP CERTIFICATE IF EXISTS
server CASCADE;"
```

4. Enable database SSL for a single issue CA or chained issue CAs:
   - For a single issue CA, run this command:

```
./db_ssl_setup --enable-ssl --vertica-cert-path
<yourOwnCertPath>/generated-db-server.crt --vertica-key-path
<yourOwnCertPath>/generated-db-server.key --vertica-ca-path
<yourOwnCertPath>/generated-db-ca.crt --client-ca-path
<yourOwnCertPath>/issue_ca.crt
```

   -or-

   - For chained issue CAs, run this command, specifying each CA certificate in the chain one by one, separated by a comma in the `client-ca-path` parameter:

```
./db_ssl_setup --enable-ssl --vertica-cert-path
<yourOwnCertPath>/generated-db-server.crt --vertica-key-path
<yourOwnCertPath>/generated-db-server.key --vertica-ca-path
<yourOwnCertPath>/generated-db-ca.crt --client-ca-path
```

```
<yourOwnCertPath>/issue_ca_part.1.crt, <yourOwnCertPath>/issue_ca_
part.2.crt[,...]
```

Configuring Deployed Capabilities to Use SSL for Database Connection

1. Log in to the CDF Management Portal.
2. Navigate to **Fusion** > **Database Configuration** > **Database Certificate(s)**.
3. Enable the **Use SSL for Database Connection** option.
4. Copy the complete contents of the file `generated-db-ca.crt`, created from the steps earlier, into the Database Certificate(s) text area.
5. Click **Save** to activate the configuration changes.

Enabling the Database to Ingest Events from Transformation Hub

The database uses an event consumer, the **Kafka scheduler**, to ingest events from Transformation Hub's Kafka component. Follow these steps when configuring the Kafka Scheduler for a new installation of the ArcSight Database:

Before you perform these steps, ensure that you have enabled SSL for the database. For information, see Enabling the Database to Receive SSL Connections.

1. Log in to the database node1 as root.
2. Change to the database tools directory:

   ```
   cd /opt/arcsight-db-tools/
   ```

3. Run the following command on database node1 to configure the schema registry server setting:

   ```
   ./schema_registry_setup <FQDN of ArcSight Platform Virtual IP for HA or
   single master node / Cloud: <DNS name for your cluster> >
   <yourOwnCertPath>/issue_ca.crt <yourOwnCertPath>/kafkascheduler.crt.pem
   <yourOwnCertPath>/kafkascheduler.key.pem
   ```

   You must provide the absolute path to the certificate.

4. Configure the SSL setup:

   On database node1, configure the SSL setting for the Kafka Scheduler by using one of the following methods, plain text or SSL:

Plain Text (non-SSL)

This method requires that you first enable **Allow plain text (non-TLS)** connections to Kafka. For more information, see Configuring the Deployed Capabilities.

Run this command to disable SSL for the Kafka scheduler:

```
./sched_ssl_setup --disable-ssl
```

SSL

This method uses the crt and key files gathered or generated in earlier steps. The `issue_ca.crt` file should contain all chained CAs. For the Kafka scheduler to use SSL, run the following command:

```
./sched_ssl_setup --enable-ssl --sched-cert-path
<yourOwnCertPath>/kafkascheduler.crt.pem --sched-key-path
<yourOwnCertPath>/kafkascheduler.key.pem --vertica-ca-key
<yourOwnCertPath>/generated-db-ca.key --vertica-ca-path
<yourOwnCertPath>/generated-db-ca.crt --kafka-ca-path
<yourOwnCertPath>/issue_ca.crt
```

5. Run this command on database node1 to create the Kafka Scheduler:
   - If the Kafka Scheduler was configured to use plain-text in the previous step, use port 9092:

     ```
     ./kafka_scheduler create <th_kafka_nodename1>:9092
     ```

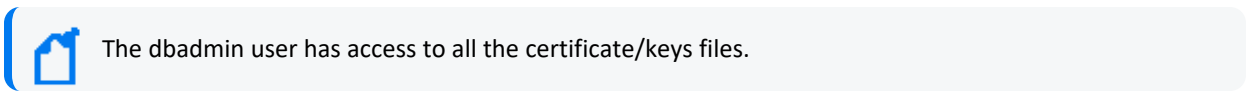
   - If SSL was enabled for the Kafka Scheduler in the previous step, use port 9093:

     ```
     ./kafka_scheduler create <th_kafka_nodename1>:9093
     ```

6. Start the Kafka Scheduler and checker on database node1:

   ```
   ./kafka_scheduler start
   ./kafka_scheduler messages
   ./kafka_scheduler events
   ```

7. Continue to the post-deployment section.



Enabling Pod Logs in Azure

You can enable the ArcSight products application (pod) logs in Azure, which includes a monitoring tool called Container Insights. For more information and instructions on enabling it, see the Azure d

Applying the CDF 2021.05 log4j Hotfix

Some deployments of, and upgrades to, CDF 2021.05/arcsight-platform-installer-22.1.x.x.zip require application of a hotfix to remediate the log4j vulnerability, which was discovered in 2021. The hotfix will upgrade IDM for CDF to use log4j 2.17.1, to prevent exploitation of the log4j vulnerabilities (CVE-2021-44228, CVE-2021-45046, CVE-2021-45105, CVE-2021-44832). The hotfix should be applied after an upgrade.

The hotfix applies to the following types of installations and upgrades:

- Any on-premises manual installation of 22.1.x or any on-premises manual upgrade to 22.1.1. (A manual installation or upgrade is one that does not use the ArcSight Installer.)

The CDF 2021.05 hotfix will be automatically applied during any on-premises installation or upgrade using the ArcSight Installer and this procedure can be skipped.

- Any CDF fresh installation or upgrade on AWS.
- Any CDF fresh installation or upgrade on Azure.

The log4j remediation hotfix does **NOT** apply to on-premises installations or upgrades performed automatically using the ArcSight Installer, as the hotfix is applied automatically by the installer. For such installations or upgrades these procedures can be skipped.

Hotfix File

The hotfix file is named arcsight-idm-hf-22.1.0-2.zip.

1. Get the file:
   - For a manual on-premises deployment/upgrade, the hotfix is bundled in `/<download_folder>/arcsight-platform-installer-22.1.x.x/installers/hotfix`.
   - For an AWS or Azure deployment upgrade, obtain the file from the *on-premises* installation file directory at `/<download_folder>/arcsight-platform-installer-22.1.x.x/installers/hotfix`.
2. Copy the file:
   - For manual on-premises, copy the file to your master node.
   - For AWS, copy the file to your bastion.
   - For Azure, copy the file to your jump host.
3. Unzip the hotfix file. In the unzipped folder, run the following command with the '-e' argument (values: onprem, azure, aws) to apply the latest image.

```
# ./hotfix.sh -e <YOUR_ENV>
```

Verifying the Hotfix

1. Check the pod status by running the following command. It should be 'Running' as 2/2.

```
# kubectl get pods -A | grep idm
```

2. Check the image version by running the following command.

```
# kubectl get deployment/itom-idm -n core -o yaml | grep itom-idm:1.32.1-343
```

It should display as below:

```
image: <image-registry-url>/<org-name>/itom-idm:1.32.1-343
```

Rolling Back to the Previous Version

To roll back itom-idm to the previous version, run the following roll back commands :

```
# kubectl delete -f /tmp/cdf-itom-idm.yaml
```

```
# kubectl create -f /tmp/cdf-itom-idm.yaml
```

Deploying ArcSight Products

Configuring the Deployed Capabilities

Refer to System Hardware Sizing and Tuning Guidelines in the *ArcSight Platform 22.1 Technical Requirements* for your workload. It might specify additional settings beyond what is described below.

You are now ready to deploy and then configure your deployed capabilities. The *Pre-Deployment Configuration* page displays to configure the products and capabilities chosen at the start of the installation process. This section explains the process of configuring deployed capabilities on a supported platform for both on-premises and cloud deployments.

Describing Parameters

The following parameters are mentioned in one or more of the example install config files.

Transformation Hub

For the TH yaml, see the following:

Name	Description
routing-processor1-replicas	Specifies the number of Routing Stream Processor Instances to start for the Group 1 Stream Processor. Routing Stream Processors convert incoming CEF events based on predefined rules associated with a unique source Topics. Group numbers are dynamically assigned by Transformation Hub. Tune the number of instances based on throughput requirements.
th-init-noOfTopicPartitions	For newly created Kafka Topics, specifies the number of partitions assigned to each Topic. Default is 6. A Partition is the unit of parallelism in Kafka, enabling write operations on both the Producer and Broker to be performed concurrently. This is a key tuning property.
transform-processor-replicas	Specifies the number of CEF-to-Avro Stream Processor Instances to start. CEF-to-Avro Stream Processors convert incoming CEF events from th-cef topic to Avro format and route these events to th-arcsight-avro topic.
th-init-kafkaRetentionBytesForVertica	Specifies the size, in gigabytes, of the retention log for th-arcsight-avro and mf-event-avro-enriched Topics (Avro primary Topics). Default is 60 GB. This is a key tuning property. This log is associated with Avro processing. It is uncompressed and might require up to 7 times more space than compressed data. When this log size is exceeded, event data will be dropped.
th-init-kafkaRetentionBytes	Specifies the size, in gigabytes, of the retention log for each Kafka Topic. Default is 60 GB. This is a key tuning property. When the retention log exceeds the size limit, event data will be dropped.
enrichment-processor1-replicas	Specifies the number of Enrichment Stream Processor Group Instances to start. Enrichment Stream Processors transform incoming events based on the set of enabled event enrichment features, and route these events to one or more destination Topics. Enrichment examples include adding Global Event IDs and event integrity checking. Tune the number of instances based on throughput requirements.

th-enrichment-processor-group1-source-topic	Specifies the source Topic to be used by the Enrichment Stream Processor Group.
th-enrichment-processor-integrity-enabled	Indicates whether to generate a verification event that accompanies a batch of events for checking the integrity of parsed fields in each event. Recon uses this verification event to check event integrity. Also, specify a value for 'Verification event batch size'.
th-enrichment-processor-integrity-batch-size	Specifies the number of events to be associated with a verification event. A lower value indicates fewer associated events need to be included in the batch for integrity checks; however, it will also result in higher resource consumption by generating more verification events.

Recon

For the Recon yaml, see the following:

Name	Description
interset-elasticsearch-data-instances	Specifies the number of Elasticsearch data processing instances.
interset-elasticsearch-index-replicas-count	Specifies the number of replicas for each Elasticsearch index. 0 means no copy, only use that value when having no HA/Production requirement.
interset-logstash-event-buffering	Specifies the internal queuing model to use for event buffering. Specify memory for legacy in-memory based queuing; persisted for disk-based queuing.
interset-logstash-instances	Specifies the number of Logstash instances.
recon-enable	Indicates whether to explore events in Recon in addition to Intelligence.

Reviewing Settings That Must Be Set During Deployment

This section describes configuration settings that must be set during deployment. Additional settings can be modified after deployment by browsing to the CDF Management Portal.

For more information on a setting, hover over the setting to display the setting tooltip, then set the values accordingly.

The following products require configuration settings to be set during deployment.

- "ArcSight Database" on the next page
- "Transformation Hub" on the next page
- "Fusion" on page 367
- "Intelligence" on page 368

ArcSight Database

If you deployed the ArcSight database and you configure SmartConnectors to use the CEF format when you send events to the Transformation Hub, in the **Transformation Hub** tab, ensure the # of CEF-to-Avro Stream Processor instances to start is set to at least 1 or what is specified in *ArcSight Platform Technical Requirements* for your workload.

On the **Fusion** tab, ensure that you set these configuration settings for your environment:

- Enable Database
- Use SSL for Database Connections

> The SSL configuration requires components to be in a running state before proceeding with the database secured configuration. To apply secure communication to the database, proceed with Completing the Database Setup for AWS S3.

- Database Host

> The host list of the database node's IP, that is node1-IP, node2-IP,..., upto nodeN-IP.

- Database Application Admin User Name
- Database Application Admin User Password
- Search User Name
- Search User Password
- Database Certificate(s)
- Database Host Name(s)

Transformation Hub

If you deployed Transformation Hub, in the **Transformation Hub** tab, ensure the following are set to the number of Kafka worker nodes in your deployment or what is specified in *ArcSight Platform Technical Requirements* for your workload.

- # of Kafka broker nodes in the Kafka cluster (th-kafka-count)
- # of ZooKeeper nodes in the ZooKeeper cluster (th-zookeeper-count)
- # of replicas assigned to each Kafka Topic (th-init-topicReplicationFactor) (This setting must be set to 1 for a single worker deployment, and 2 for a 3-node environment.)

On the **Transformation Hub** tab, configure the following security settings based on how you planned to secure communications as described in the Securing Communication Among Micro Focus Components section.

FIPS and Client-Authentication are available during installation only.

- Allow plain text (non-TLS) connections to Kafka (th-kafka-allow-plaintext)
- Enable FIPS 140-2 Mode (th-init-fips)
- Connection to Kafka uses TLS Client Authentication (th-init-client-auth)
- # of message replicas for the __consumer_offsets Topic (th-init-kafkaOffsetsTopicReplicationFactor)
- Schema Registry nodes in the cluster (th-schema-registry-count)
- # of replicas assigned to each Kafka Topic (th-init-topicReplicationFactor)

If you are deploying ESM, configure your Enrichment Stream Processor Group source Topic according to the scope for which you want to leverage ESM's event enrichment capability. For more information, refer to Enrichment Stream Processors.

Fusion

If you deployed Fusion, on the **Fusion** tab:

- Modify the **Client ID** (sso-client-id) and **Client Secret** (sso-client-secret) to a unique value for your environment.
- If you are deploying Transformation Hub and configured **# of Enrichment Stream Processor Group instances to start** (enrichment-processor1-replicas) with a value greater than zero (default is 2), which means Enrichment Stream Processor will be enabled, the Fusion ArcMC Generator ID Manager must be enabled with a sufficient range of IDs because the Enrichment Stream Processor automatically requests generator IDs from the Fusion ArcMC in the same cluster as needed for its processing. To enable the Fusion ArcMC Generator ID Manager, configure **Enable Generator ID Manager** (arcmc-generator-id-enable) to **True** (default is True) and set the the values of **Generator ID Range Start** (arcmc-generator-id-start) and **Generator ID Range End** (arcmc-generator-id-end) to provide a range of at least 100 between them. A range of 100 should be sufficient for common scenarios with a comfortable buffer, but you could also make the range larger if you have configured a large number of Enrichment Stream Processor instances or other components that use Generator IDs from this Fusion ArcMC instance.
- **Maximum Search Results:** This value specifies number of results that a search can return. Maximum limit is 10 million events.

  It is important to choose a range that does not overlap with the Generator ID Manager range configured in any other ArcMC instances in your organization, otherwise different events with duplicate Globally Unique Event IDs could be created.

Intelligence

If you deployed Intelligence, on the Intelligence tab, ensure you set these configuration settings for your environment:

- Number of Database Nodes (interset-vertica-number-of-nodes)

  Ensure that you change any password to a unique value for your environment.

- HDFS NameNode (interset-hdfs-namenode)
- Elasticsearch Index Replicas Count (interset-elasticsearch-index-replicas-count)
- H2 Password (interset-h2-password)
- Elasticsearch Password (interset-elasticsearch-password)
- Analytics KeyStore Password (interset-analytics-keystore-password)
- Investigator KeyStore Password (interset-api-keystore-password)
- SearchManager KeyStore Password (searchmanager-api-keystore-password)
- Logstash KeyStore Password (interset-logstash-keystore-password)
- H2 KeyStore Password (interset-h2-keystore-password)

Consider the following:

- If the topic name specified for the Avro Event Topic field is not the default topic, then use Transformation Hub's Avro routing rules using ArcMC 2.96 or later to filter Avro events from the default topic. Create a routing rule with the source topic as mf-event-avro-enriched and destination topic as the topic name you have provided in the Avro Event Topic field. For more information, see Creating a Route.

- For **Analytics Configuration-Spark**, set the values based on the data load. For information about the values for Spark, see System Hardware Sizing and Tuning Guidelines in the *ArcSight Platform 22.1 Technical Requirements* for your workload.
- For the **Data Identifiers to Identify Machine Users** field, if you need to consider only human users for licensing, ensure that you provide appropriate values to identify and filter out the machine users from licensing. For more information, contact Micro Focus Customer Support.

If you are specifying details under the **Hadoop File System (HDFS) Security** section, consider the following:

- If you are enabling Kerberos Authentication, then, before selecting **kerberos** in **Enable Authentication with HDFS Cluster**, ensure you configure the Kerberos Authentication. For more information, see Enabling and Configuring Kerberos Authentication.
- The Kerberos details that you provide in **Kerberos Domain Controller Server**, **Kerberos Domain Controller Admin Server**, **Kerberos Domain Controller Domain**, and **Default Kerberos Domain Controller Realm** will be considered only if you select **kerberos** in **Enable Authentication with HDFS Cluster**. They are not valid if you select **simple**.
- If you are enabling Kerberos Authentication, then you must enable **Enable Secure Data Transfer with HDFS Cluster**.
  If you disable **Enable Secure Data Transfer with HDFS Cluster**, the database and HDFS will use the same communication standard as Intelligence 6.2.

Checking Deployment Status

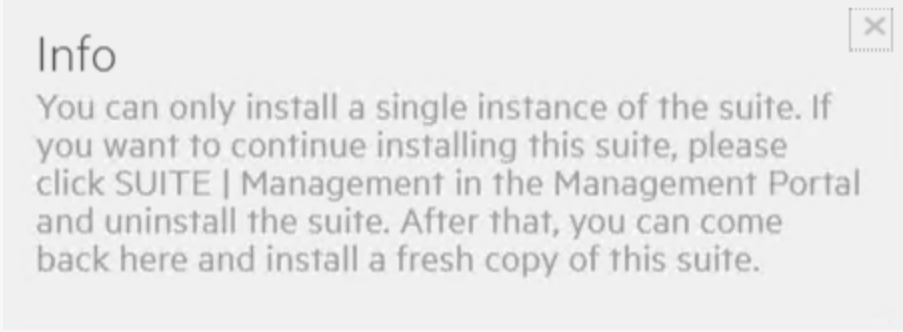
When the **Configuration Complete** page displays, the pod deployment is finished.

- Pods that have not been labeled remain in the *Pending* state until labeled.
- For a pod that is not in the *Running* state, you can find out more details on the pod by running the following command:

```
kubectl describe pod <pod name> -n <namespace>
```

The `Events` section in the output provides detailed information on the pod status.

If you see the following error when you attempt to log in to the CDF Management Portal on port 3000, this typically means that the CDF installation process has completed, port 3000 is no longer required, and has been closed. Instead of port 3000, log in to the Management Portal on port 5443.



Checking Cluster Status

To verify the success of the deployment, check the cluster status and make sure all pods are running.

You might need to wait 10 minutes or more for all pods to be in a *Running* or *Completed* state.

To check cluster status:

1. Connect to the cluster by doing one of the following:
   - For an on-premises installation, log in to the initial master node.
   - For Azure, connect to the jump host.
   - For AWS, connect to the bastion.
2. Run the command:

```
kubectl get pods --all-namespaces
```

3. Review the output to determine the status of all pods.

If the Elasticsearch and Logstash pods enter into a CrashLoopBackOff state, refer to Known Issues in the ArcSight Intelligence 6.4.0 Release Notes for the workarounds.

Tuning Your Deployment for Recon or Intelligence

This section describes tuning your deployment for Recon or Intelligence (only).

Updating Event Topic Partition Number

Refer to the ArcSight Platform 22.1 Technical Requirements, section entitled System Hardware Sizing and Tuning Guidelines to determine an appropriate event topic partition number for your workload.

To update the topic partition number from the master node1, run the following commands:

1. Find NAMESPACE ($NS), for th-kafka-0:

```
NS=`kubectl get pods --all-namespaces|grep kafka-0|awk '{print $1}'`
```

2. Update the Enrichment Stream Processor source topic (`th-arcsight-avro` or `mf-event-avro-esmfiltered`) and `mf-event-avro-enriched` topic partition numbers to the same for both topics:

```
kubectl exec -n $NS th-kafka-0 -- /usr/bin/kafka-topics --bootstrap-
server th-kafka-svc:9092 --alter --topic ENRICHMENT_SP_SOURCE_TOPIC --
partitions $number
```

3. Use the Kafka Manager to verify that the partition number of the th-cef topic, enrichment stream processor source topic (`th-arcsight-avro` or `mf-event-avro-esmfiltered`) and `mf-event-avro-enriched` topics have been updated to `$number`, where `$number` is the number used to calculate partition size.
4. Update the `th-arcsight-avro` and `mf-event-avro-enriched` topic partition numbers to the same for both topics:

   ```
   kubectl exec -n $NS th-kafka-0 -- /usr/bin/kafka-topics --bootstrap-
   server th-kafka-svc:9092 --alter --topic th-arcsight-avro --partitions
   $number
   ```

   ```
   kubectl exec -n $NS th-kafka-0 -- /usr/bin/kafka-topics --bootstrap-
   server th-kafka-svc:9092 --alter --topic mf-event-avro-enriched --
   partitions $number
   ```

   ```
   kubectl exec -n $NS th-kafka-0 -- /usr/bin/kafka-topics --bootstrap-
   server th-kafka-svc:9092 --alter --topic th-cef --partitions $number
   ```

   where `$number` is the number used to calculate partition size.

   > Standard Kafka topics settings only permit increasing the number of partitions, not decreasing them, so please consider that when performing step 2.

5. Use the Kafka manager to verify that the partition number of th-cef topic, th-arcsight-avro and mf-event-avro-enriched topics have been updated to `$number`.

> In case of Recon, partition number == DB nodes # * 12; therefore for a 3 node db cluster, the partition number is 36.

Updating the CDF Hard Eviction Policy

You need to update the Kubernetes hard eviction policy from 15% (default) to 100 GB to maximize disk usage.

To update the CDF Hard Eviction Policy, perform the following steps on each worker node, after deployment has been successfully completed. Please verify the operation is successfully executed on one work node first, then proceed on the next worker node.

The eviction-hard can be defined as either a percentage or a specific amount. The percentage or the specific amount will be determined by the volume storage.

To update the policy:

1. Run the following commands:

```
cp
/usr/lib/systemd/system/kubelet.service
/usr/lib/systemd/system/kubelet.service.orig
```

```
vim /usr/lib/systemd/system/kubelet.service
```

2. In the file, after `ExecStart=/usr/bin/kubelet \`, add the following line:

```
--eviction-
hard=memory.available<100Mi,nodefs.available<100Gi,imagefs.available<2Gi \
```

3. Save your change to the file.
4. To activate the change, run the following command:

```
systemctl daemon-reload and systemctl restart kubelet
```

5. To verify the change, run:

```
systemctl status kubelet
```

No error should be reported.

Chapter 5: Adding Additional Capabilities to an Existing Cluster

You can deploy additional capabilities, such as Recon or Intelligence, to an existing ArcSight Platform cluster in your on premises, AWS, or Azure environment. Reusing an existing cluster reduces costs and system management effort compared to deploying these capabilities in a new cluster.

Prerequisites and Considerations for Adding Capabilities

Be sure to review the capability-specific prerequisites listed here Configuring Elasticsearch Settings.

Before deploying additional capabilities to an existing cluster, review and, if necessary, perform the following tasks.

- Ensure that your existing cluster has the supported version of the Platform required to deploy the additional capabilities. If your deployment does not have the supported version, you must upgrade the Platform using the instructions in Upgrading Your Environment. For information about the supported version of the ArcSight Platform, see ArcSight Platform Technical Requirements.
- Recon and Intelligence both require the ArcSight Database. If you are adding these capabilities and your deployment does not already have the database, you will need to install the database using the instructions in this section.
- Check the system size of your existing ArcSight Platform Kubernetes cluster and, if applicable, ArcSight Database and ensure that it can handle the additional workload of the capabilities you want to add. If your existing cluster cannot handle the additional workload, scale the Kubernetes cluster or database as needed before deploying the additional capabilities. For information about host sizing for the Platform, see ArcSight Platform Technical Requirements.
- For Intelligence, configure SmartConnectors for data collection. For more information about data collection, see the SmartConnector Installation and User Guide and SmartConnector Configuration Guides.

Deploying Additional Capabilities to an Existing Cluster

You can add capabilities, such as Recon or Intelligence, to an existing cluster in your on premises, AWS, and Azure deployment.

Ensure that you review the capability-specific prerequisites listed in Configuring Elasticsearch Settings.

1. (Conditional) If you are adding Recon or Intelligence to your deployment, and you do not have a database deployed, deploy the database for your environment:
   - AWS
   - Azure
   - On-premises
2. (Conditional) Log in to the appropriate node or host, based on your environment:
   - **For an on-premises deployment:** Launch a terminal session and then log in to the master node as the `root` or as a `sudo` user.
   - **For an Azure deployment:** Log in to the jump host.
   - **For an AWS deployment:** Log in to the bastion host.
3. Create a directory for the image files that you will download in the next step:

   ```
   mkdir /tmp/download
   ```

   This directory must contain only the image files and nothing else.
4. Download the images for the capabilities that you want to add.

   For more information about images, see "Downloading ArcSight Platform Installation Files" in the ArcSight Platform Release Notes.
5. Validate the digital signature of each file.

   For a complete list of files and file versions to be downloaded, consult the ArcSight Platform Release Notes.

   Do not untar the files.
6. Change to the following directory.

   ```
   cd ${K8S_HOME}/scripts/
   ```

   For example:

   ```
   cd /opt/arcsight/kubernetes/scripts/
   ```

7. To upload the images to the local Docker Registry, run the following commands:

```
./uploadimages.sh -c 2 -F /tmp/download/fusion-x.x.x.x.tar -F
/tmp/download/recon-x.x.x.x.tar
```

Be aware of the following considerations:

- For each image to upload, use the `-F <image file>` option on the command line.

  To increase the speed of the upload, adjust the `-c 2` option up to half of your CPU cores.

- You will be prompted for a password for the Docker container registry-admin user. The registry-admin password is initially set to the same password as the admin user for the CDF Management Portal during installation when "Configuring and Running the CDF Installer" on page 96. However, changing the CDF Management Portal admin password later does not change the registry-admin password, because it is managed separately.

8. Log in to the CDF Management Portal with the following credentials:

   User name: admin

   Password: *<the password you provided during CDF installation>*

9. Click  , then click**Change**.
10. On the **Capabilities** page, select the additional capabilities to deploy.

> If you are deploying Fusion on AWS, ensure that you have already configured a listener and target group for port 32080 prior to Fusion deployment. This port is required to add Transformation Hub to Fusion ArcMC.

11. Click the arrow next to each capability checkbox to view the description of the each capability to deploy, and determine if it requires additional capabilities. For example, deployment of ArcSight Recon requires the deployment of Transformation Hub and Fusion.
12. Click **Next** until you reach the **Configure/Deploy** page.
13. See Configuring the Deployed Capabilities, then return to this page to continue.
14. Click **Next**. On the **Configuration Complete** page, wait until the deployment is complete. The deployment process might take several minutes to complete.

> Some of the pods in the **Configuration Complete** page might remain in a Pending state until the product labels are applied on worker nodes.

15. Continue with labeling the nodes according to your deployment:
    - AWS
    - Azure
    - On premises
16. (Conditional) If you deployed the database in the first step, follow the instructions in Completing the Database Setup.
17. Continue to Performing Post-deployment Configuration.

Chapter 6: Performing Post-deployment Configuration

This section provides information about the post-installation configuration you must perform.

Installing Your License Key

ESM, ArcMC, Transformation Hub, Intelligence, Recon, and SOAR all require license keys. Each product ships with a 90-day instant-on evaluation license, which will enable functionality for a 90 day evaluation period after installation. In order for a product to continue functioning past the initial evaluation period, you will need to apply a valid license key. For more information about license keys, see the Understanding License Keys section.

To ensure continuity of functionality and event flow, make sure you apply your product license **before** the evaluation license has expired.

To install your license:

1. Log in to the Management Portal (https://<ha-address>:5443).
2. Click **APPLICATION**.
3. Click **License**.

   For more information about license management capabilities, see AutoPass License Management documentation.
4. Click **License > Install.**
5. Click **ADD FILE(S).**
6. Browse to the location of your license file.
7. Click **Next**.
8. Optionally, select the **I authorize Micro Focus to collect suite and product data...** check box to send usage data to Micro Focus to help improve the product.
9. Follow the prompts to apply your license.
10. Apply all of the licenses required for your deployed capabilities.
11. (Conditional) If you just installed a Transformation Hub license, restart each Kafka pod in the cluster, one at a time, as follows:

a. For each of the Kafka pods from 0 to x, restart the selected Kafka pod with the command:
```
# kubectl delete pod th-kafka-(x) -n arcsight-installer-XXX
```
b. Watch the logs and ensure that each Kafka pod is up and running by running this command:
```
# kubectl logs th-kafka-(x) -n arcsight-installer-XXX
```
c. After the selected broker node is up and running, only then proceed to restart the next node.
d. 
You can also check the status of the restarted broker node using the Transformation Hub Kafka Manager.

12. (Conditional) If you just installed an ArcSight Management Center (ArcMC) the ArcMC pod needs to be restarted after uploading the license, as follows:
```
# kubectl delete pod fusion-arcmc-web-app-XXX -n arcsight-installer-XXX
```

Report and SOAR capabilities with an ESM License

The report and SOAR capabilities with an ESM License must be enabled by performing the following steps:

1. Visit the CDF management portal `https://<CDF_HOST>:5443` and log in.
2. Go to the **Reconfigure** section.
3. In Fusion, go to the **ESM License Configuration for SOAR & Reporting Portal** section
4. Enable the **Use ESM License** option.
5. Enter the URL of the ESM.
6. Input the **ESM Username** and **ESM Password** and save them.

Once these steps are performed, the SOAR-web-app and reporting-web-app processes will be stopped, and these pods will be restarted again.

Configuring the Database with HDFS for Intelligence


This procedure applies only when you deploy the Intelligence capability.

After deploying Intelligence, you must configure the database with HDFS for the database to receive the Intelligence Analytics results data from Spark through HDFS.

The following topics are discussed here:

Prerequisites

For a manual deployment of Intelligence, ensure that you install firewall and open the firewall ports on the nodes before you proceed with configuring the database with HDFS:

1. Log in to a Kubernetes node labeled as intelligence-namenode:yes as a root user.
2. Execute the following commands to install and enable the firewall:

```
yum -y install firewalld
systemctl enable firewalld
```

3. Execute the following command to ensure NAT is configured:

```
firewall-cmd --add-masquerade --permanent
```

4. (Conditional) Execute the following commands to open the "Intelligence" on page 38 on the node labeled as intelligence-namenode:yes:

```
firewall-cmd --permanent --add-port=30820/tcp
firewall-cmd --permanent --add-port=30070/tcp
```

5. (Conditional) Execute the following commands to open the "Intelligence" on page 38 on the node labeled as intelligence-datanode:yes.

```
firewall-cmd --permanent --add-port=30210/tcp
firewall-cmd --permanent --add-port=30010/tcp
```

6. Execute the following commands to avoid a firewall restart and to ensure that the Kubernetes services do not stop running on the node:

```
iptables -I IN_public_allow -j ACCEPT -p tcp --dport 30820 -m conntrack -
-ctstate NEW,UNTRACKED
iptables -I IN_public_allow -j ACCEPT -p tcp --dport 30210 -m conntrack -
-ctstate NEW,UNTRACKED
iptables -I IN_public_allow -j ACCEPT -p tcp --dport 30070 -m conntrack -
-ctstate NEW,UNTRACKED
iptables -I IN_public_allow -j ACCEPT -p tcp --dport 30010 -m conntrack -
-ctstate NEW,UNTRACKED
```

7. Repeat steps 1 to 5 on all nodes labeled as intelligence-datanode:yes.

Configuring the database with unsecured HDFS

1. (Conditional) For an on-premise deployment, launch a terminal session and log in to a Kubernetes node as a `root` user.
2. (Conditional) For a cloud deployment, do the following:
   - For AWS, connect to the bastion.
   - For Azure, connect to the jumphost.
3. Execute the following command to retrieve the namespace:

   ```
   export NS=$(kubectl get namespaces |grep arcsight|cut -d ' ' -f1)
   ```

4. Execute the following command to retrieve the RPC port and the web port:

   ```
   kubectl -n $NS get svc |grep hdfs-namenode
   ```

   An example of the output is:

   ```
   hdfs-namenode-svc ClusterIP None <none> 30820/TCP,30070/TCP 4h32m
   ```

   The first TCP port number (30820) is the RPC port and the second TCP port number (30070) is the web port.
5. Log in to a database node as a root user.
6. (Conditional) Create the `/etc/hadoop/conf/` directory, if it does not already exist.
7. (Conditional) Create the **core-site.xml** file if it does not already exist, then update the **fs.defaultFS** and **dfs.-namenode.http-address** properties along with the ports you retrieved in Step 3. Ensure that the `NAMENODE_HOST` value matches the hostname or IP address you provided in the **HDFS NameNode** field in the CDF Management Portal, **Configure/Deploy > Intelligence.**

   ```
   cat /etc/hadoop/conf/core-site.xml
   <configuration>
   <property>
   <name>fs.defaultFS</name>
   <value>hdfs://<NAMENODE_HOST>:<NAMENODE_RPC_PORT>/</value>
   </property>
   <property>
   <name>dfs.namenode.http-address</name>
   <value><NAMENODE_HOST>:<NAMENODE_WEB_PORT></value>
   </property>
   </configuration>
   ```

   For example:

```
cat /etc/hadoop/conf/core-site.xml
<configuration>
<property>
<name>fs.defaultFS</name>
<value>hdfs://vlab012345.interset:30820/</value>
</property>
<property>
<name>dfs.namenode.http-address</name>
<value>vlab12345.interset:30070</value>
</property>
</configuration>
```

8. Create the **hdfs-site.xml** file as follows if it does not already exist:

```
<configuration>
</configuration>
```

9. Repeat steps 5 to 8 on all database nodes.
10. Verify whether the database and HDFS configuration is successful:
    a. Change to the following directory:

```
cd /opt/vertica/bin/
```

    b. Log in as a dbadmin:

```
su dbadmin
```

    c. Log in to vsql and specify the password when prompted:

```
vsql
[password prompt]
```

    d. (Optional) Clear the cache after configuring the database with HDFS:

```
SELECT CLEAR_HDFS_CACHES();
```

    e. Execute the following commands:

```
SELECT VERIFY_HADOOP_CONF_DIR();
```

```
SELECT node_name, node_address, export_address FROM nodes;
```

    The expected output is:

```
Welcome to vsql, the Vertica Analytic Database interactive terminal.
```

```
Type: \h or \? for help with vsql commands
\g or terminate with semicolon to execute query
\q to quit

dbadmin=> SELECT VERIFY_HADOOP_CONF_DIR();
VERIFY_HADOOP_CONF_DIR
-------------------------------------------------------------------------
-----
Validation Success
v_fusiondb_node0001: HadoopConfDir [/etc/hadoop/conf] is valid

(1 row)

dbadmin=> SELECT node_name, node_address, export_address FROM nodes;
node_name | node_address | export_address
-----------------------------------------------------------
v_fusiondb_node0001 | <IP1> | <IP1>
v_fusiondb_node0002 | <IP2> | <IP2>
v_fusiondb_node0003 | <IP3> | <IP3>
(3 rows)
```

11. If you have enabled **Enable Secure Data Transfer with HDFS Cluster** in the **Intelligence** tab in the **CDF Management Portal** and if you have a non-collocated database cluster, log in to a database node, and copy the RE CA certificate from the CDF master node to /etc/pki/ca-trust/extracted/pem/tls-ca-bundle.pem. Repeat this step on all the database nodes.

Configuring the database with secured HDFS

1. (Conditional) For an on-premise deployment, launch a terminal session and log in to a Kubernetes node as a `root` user.
2. (Conditional) For a cloud deployment, do the following:
   - For AWS, connect to the bastion.
   - For Azure, connect to the jumphost.
3. Execute the following command to retrieve the namespace:

   ```
   export NS=$(kubectl get namespaces |grep arcsight|cut -d ' ' -f1)
   ```

4. Execute the following command to retrieve the RPC port and the web port:

   ```
   kubectl -n $NS get svc |grep hdfs-namenode
   ```

   An example of the output is:

   ```
   hdfs-namenode-svc ClusterIP None <none> 30820/TCP,30070/TCP 4h32m
   ```

The first TCP port number (30820) is the RPC port and the second TCP port number (30070) is the web port.

5. Log in to a database node as a root user.
6. (Conditional) Create the `/etc/hadoop/conf/` directory, if it does not already exist.
7. Create the **hdfs-site.xml** file as follows if it does not already exist:

```
<configuration>
 <property>
  <name>dfs.http.policy</name>
  <value>HTTPS_ONLY</value>
 </property>
 <property>
  <name>dfs.encrypt.data.transfer</name>
  <value>false</value>
 </property>
</configuration>
```

8. Create the **core-site.xml** file if it does not already exist, then update the **fs.defaultFS** and **dfs.-namenode.https-address** properties along with the ports you retrieved in Step 4. Ensure that the `NAMENODE_HOST` value matches the hostname or IP address you provided in the **HDFS NameNode** field in the CDF Management Portal, **Configure/Deploy > Intelligence.**

```
<configuration>
 <property>
  <name>fs.defaultFS</name>
  <value>hdfs://<NAMENODE_HOST>:30820/</value>
 </property>
 <property>
  <name>dfs.namenode.https-address</name>
  <value><NAMENODE_HOST>:30070</value>
 </property>
</configuration>
```

9. Repeat steps 5 to 8 on all database nodes.
10. Verify whether the database and HDFS configuration is successful:
    a. Change to the following directory:

```
cd /opt/vertica/bin/
```

    b. Log in as a dbadmin:

```
su dbadmin
```

    c. Log in to vsql and specify the password when prompted:

```
vsql
[password prompt]
```

d. Clear the cache after configuring the database with HDFS:

```
SELECT CLEAR_HDFS_CACHES();
```

e. Execute the following commands to verify that the database configuration is valid:

i.
```
SELECT VERIFY_HADOOP_CONF_DIR();
```

ii.
```
SELECT HDFS_CLUSTER_CONFIG_CHECK();
```

Creating the First System Admin User

This procedure applies only when you deploy a capability that requires Fusion.

To create the first user in the System Admin role:

1. Open a supported web browser.
2. Specify the following URL to log in to the application:

```
https://<cdf_masternode_hostname or virtual_ip hostname>/mgmt
```

3. Specify the required information to create a System Admin user. (**Important**: It is strongly recommended that you use a valid email address for the user, so that it can be used to recover access to the account if the password is forgotten. There is no practical way to recover the account when the password is forgotten if the email address is not valid.)
4. After the account is created, log in with the credentials you just created.
5. (Optional) Log in to the application with the Email ID and password you just created.

Setting User Role for a New User for SOAR

This procedure applies only when you deploy SOAR capability. You must have at least one user with a **Super Admin** permissions.

SOAR user roles are not mapped with the platform roles. So, **SOAR** assigns a default role of **Super Admin** to each new user.

To revoke the **Super Admin** privileges of a new user, you must change the default user role assigned to the users, to a role that has lesser privileges, for example an empty role on the SOAR application.

To change new user role from Super Admin role to Empty role:

1. Navigate to the **ArcSight SOAR** capability.
2. Select **Configuration** > **Parameters** page.
3. Set the **ArcSightUserRole** parameter to **Empty Role** for the new user.

Enabling Integration with Azure Transformation Hub

This procedure applies only when you have deployed to Azure.

For proper integration with Azure Transformation Hub, after you set up your Azure deployment architecture, you must perform the following additional procedures for the ArcSight product (ArcMC, SmartConnector, CTH, Logger, or ESM) you are integrating. You must complete the procedures before you can configure the product to consume events from or send events to Transformation Hub:

- Edit the `/etc/hosts` file.
- Configure peering.
- Configure health probes and load-balancing rules for ports 32080 and 9093.

  Note: For ESM, this applies only to port 9093.

Editing the `/etc/hosts` File

You must add each Transformation Hub node in the cluster to the product's `/etc/hosts` file:

1. On the jump host that you previously created, open the product's `/etc/hosts` file in a text editor.
2. Add the internal IP address and FQDN for each instance in the Azure Kubernetes service. You can obtain the instance IP address and FQDN by opening the AKS resource group that you previously created and then opening the aks-nodepool virtual machine scale set.
3. Save the changes to the file. The saved changes should be similar to the following:

```
127.0.0.1 localhost localhost.localdomain localhost4
localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

10.1.1.4 aks-nodepool1-12400006-vmss000000
```

```
10.1.1.5 aks-nodepool1-12400006-vmss000001
10.1.1.6 aks-nodepool1-12400006-vmss000002
```

When editing your `/etc/hosts` file, ensure that the IP address specified each host is unique and not duplicated across hosts. A single IP address can be associated with multiple hostnames, but the same IP address may not be used for multiple hosts.

Configuring Peering

If the Azure product and Azure Transformation Hub are on different VLANs, you must configure peering between the two VLANs. An example is provided in the section Peering Virtual Networks.

Configuring Health Probes

You must configure health probes and load balancing rules for ports 32080 and 9093.

Note: Some of the commands shown here will require `root` user privileges.

You can now configure the product to consume events from or, if the functionality is available, send events to Transformation Hub:

- Configuring Logger and SmartConnectors as a Transformation Hub Consumer
- Configuring ESM as a Transformation Hub Consumer
- Configuring ESM as a Transformation Hub Producer in Distributed Correlation Mode
- Configuring ArcMC to Manage a Transformation Hub

Enabling Integration with AWS Transformation Hub

This procedure applies only when you have deployed to AWS.

For proper integration with AWS Transformation Hub, after you set up your AWS deployment architecture, you must complete the applicable procedure for the ArcSight product (SmartConnectors, Logger, or ESM) you are integrating. You must complete the procedure before you can configure the product to consume events from or send events to Transformation Hub.

Completing Additional Procedures for SmartConnectors, Logger, or ESM

Obtain the cluster worker node (Kafka broker node) host names using one of the following procedures.

From the bastion host:

1. From the bastion host, run the following command:

```
# kubectl get nodes
```

2. Copy the node host names. You will need these names when configuring the product to produce events from or send events to Transformation Hub.

From AWS:

1. In the AWS user interface, go to your Auto Scaling Group.
2. To see the instance IDs, select the **Instance Management** tab.
3. To view the details of the corresponding instance, click the first instance ID.
4. Note the private DNS name for the instance.
5. Repeat Step 3 and Step 4 for each instance ID.
6. After determining these values, ensure that your SmartConnector, Logger, or ESM instance is added to the AWS Transformation Hub Cluster Security Group with rules allowing access to ports 32080 and 9093.

You can now configure the product to consume events from or, if the functionality is available, send events to Transformation Hub:

- Configuring Logger and SmartConnectors as a Transformation Hub Consumer
- Configuring ESM as a Transformation Hub Consumer
- Configuring ESM as a Transformation Hub Producer in Distributed Correlation Mode

Configuring ArcMC Parser Upgrades

Perform the steps below to configure ArcMC parser upgrades.

Access ArcSight Marketplace Through a Proxy Server

To access the ArcSight Marketplace through a proxy server when performing parser upgrades:

1. Log in to the CDF Management Portal. See "Accessing the CDF Management Portal" on page 680 for more information.
2. From the left menu select **Deployment > Deployments**.
3. Click **... (Browse)** on the far right and choose **Reconfigure**. A new screen will open in a separate tab.
4. Select the **Fusion** tab.
5. Scroll down to the **ArcMC Parser Upgrades** section, then specify the desired value for the parameters:
   a. Proxy Server for Parser Upgrades
   b. Proxy Port for Parser Upgrades
   c. Proxy Username for Parser Upgrades (if the proxy server needs authentication)
   d. Proxy Password for Parser Upgrades (if the proxy server needs authentication)
6. Click **Save**. The ArcMC pod will be restarted.

Change the Number of Parser Upgrade Versions Displayed

To select the number of parser upgrade versions retrieved from the ArcSight Marketplace:

1. Follow steps 1 through 4 in "Access ArcSight Marketplace Through a Proxy Server" on the previous page
2. Scroll down to the **ArcMC Parser Upgrades** section and select the desired value for the **Display Marketplace Parser Upgrade Versions**parameter.
3. Click **Save**. The ArcMC pod will be restarted.

Disable the Marketplace Connection

1. Follow steps 1 through 4 in "Access ArcSight Marketplace Through a Proxy Server" on the previous page
2. Scroll down to the **ArcMC Parser Upgrades** section and disable the **Enable Marketplace** option.
3. Click **Save**. The ArcMC pod will be restarted.

If you disable the connection to the ArcSight Marketplace you will not be able to see the available parser upgrade versions. In addition, the containers under **Node Management > Containers tab**, will not display the Parser Out of Date status in the Parser Version column.

Checklist: Performing Regular Maintenance

Use the following checklist to perform regular maintenance of the Platform infrastructure.

Frequency	Task	See...
Every 1-3 Days	Check the Health and Performance Dashboard for status or errors.	Using the Health and Performance Monitoring Dashboard
Every 1-3 Days	Check the Kubernetes Dashboard for status and errors.	"Checking Kubernetes Dashboard for Status and Errors" on page 684
Every week	Check automatic backup jobs for statuand errors. **NOTE:** See the links to the right for the procedures to enable the automatic backup jobs. After the jobs are enabled, they will run automatically on a schedule unless an error is encountered.	Backing Up and Restoring the Database Backing Up PostgreSQL on CDF for ArcSight Capabilities Backing Up Recon Management and Search Data Stores
Every 1-3 Days	Run CDF Doctor for status and errors.	Using the CDF Doctor Utility
Every 90 Days	Reset the expiring CDF Management Portal admin account password. he registry-admin password used when uploading capability images during system upgrade is initially set to the same password as the admin user for the CDF Management Portal during installation when Configuring and Installing CDF; however, later changing the CDF Management Portal admin password does not change the registry-admin password as it is managed separately. The registry-admin password does not automatically expire.	Resetting the CDF Administrator Password
Every 11 Months	Renew any expiring CDF certificates (default expiration is 1 year).	"Renewing External Certificate of Management Portal and Fusion Single-Sign-On Portal " on page 700

Configuring Intelligence Analytics Targeted Events

This topic and the procedure that follows are applicable only if you are deploying both the ArcSight Intelligence and ArcSight Recon capabilities with the ArcSight Platform.

ArcSight data scientists identified a targeted set of events that, through extensive research, have proven to be the best inputs into threat detection analytic models. Other events, while useful for other purposes such as investigating all stages of a threat life-cycle, are not used as input into the Intelligence analytics capabilities. Configuring Elasticsearch filtering enables Intelligence analytics resources to be focused on the events that are most valuable for Intelligence analytics capabilities. In some deployments, depending on the ingested data and especially when ArcSight Recon and Intelligence are both deployed, applying a proper filter can free up significant storage and compute costs.

This procedure installs the Logstash based event filter to limit what is stored in the Intelligence Elasticsearch component. Once configured, it will filter events coming from Transformation Hub to only ingest into the Intelligence Elasticsearch component the events that are useful with Intelligence analytics.

To install the filter for Intelligence Elasticsearch:

1. Log in to the master node of the CDF cluster.
2. Edit the `logstash-config-pipeline` file using the following command:

```
kubectl -n $(kubectl get namespaces | grep arcsight | cut -d ' ' -f1)
edit configmaps logstash-config-pipeline
```

3. Locate the code block below and add a line break at the end:

```
if [destinationUserName] =~ "\$" {
   mutate {

replace => {
         "did" => "1"
      }
   }
}
```

4. Add the following code block after the line break, while respecting code indentation:

```
if [destinationUserName] and [externalId] in
['4624','4625','4648','4768','4769','4771','4776','4777'] {
}
else if [categoryDeviceGroup] =~ "/Network Equipment" and
```

```
[categoryDeviceType] =~ "Network Monitoring" {
}
else if [sourceusername] and [externalId] in ['Squid','Microsoft','Blue
Coat'] and [deviceProduct] in ['Proxy SG','Squid Web Proxy Server','ISA
Server'] {
}
else if [categoryDeviceType] =~"Repository" and [destinationUserName] and
[deviceAction] and [deviceCustomString1] {
}
else if [destinationUserName] and [categoryOutcome] and [fileName] and
[deviceProduct] in ['','WORKGROUP','NT SERVICE','NT AUTHORITY'] {
}
else if [type] == 2 {
}
else if [categoryObject] =~ "/Host/Application/Service" {
}
else {
      drop {}
}
```

5. Run the following command:

```
kubectl -n $(kubectl get namespaces | grep arcsight | cut -d ' ' -f1)
scale statefulset interset-logstash --replicas=0
```

6. After allowing time for the process to finish, validate that pods are successfully scaled down by running:

```
kubectl -n $(kubectl get namespaces | grep arcsight | cut -d ' ' -f1) get
pods | grep logstash
```

   You will know the process is finished when no pods are shown.

7. Run the following command to complete the filter installation:

```
kubectl -n $(kubectl get namespaces | grep arcsight | cut -d ' ' -f1)
scale statefulset interset-logstash --replicas=X
```

   Were X stands for the number of Logstash instances you want to run across the cluster and generally equals the number of worker nodes.

Chapter 7: Integrating the Platform Into Your Environment

Transformation Hub integrates with many ArcSight products, and is managed and monitored by ArcSight Management Center. After you install and configure Transformation Hub you can use SmartConnectors and Collectors to produce and publish data to the Transformation Hub, and to subscribe to and consume that data with Logger, ESM, ArcSight Recon, Apache Hadoop, or your own custom consumer.

Currently, cloud (Azure and AWS) clusters only support other ArcSight products which are in the Azure or AWS cloud. Integration with on-premises products is not supported for cloud-based Transformation Hub.

Transformation Hub supports both Common Event Format (CEF) versions, 0.1 and 1.0, as well as Avro and binary data formats. Transformation Hub third-party integration and other product features are explained in detail in the following sections.

Connecting to Your SMTP Server

To ensure that ESM for Fusion users receive email notifications, configure the connection to your SMTP server. For example, if you do not use external authentication (such as LDAP or SAML), users will need notifications to help reset their forgotten passwords.

This section includes information for both on-premises and cloud deployments.

AWS SES SMTP prerequisites: If you are deploying Fusion with AWS, complete these prerequisites before configuring SMTP for Fusion. AWS SES SMTP requires TLS to be enabled.

- For the AWS SES SMTP service, ensure that your private security group allows SMTP on the ports specified in AWS documentation.
  For more information, see *Connecting to an Amazon SES SMTP endpoint*.
- Review the list of AWS SMTP protocol endpoints. For more information, see Amazon SES.
- Download the Starfield Services Root CA from https://good.sca0a.amazontrust.com/.
- Make sure your Cloud administrator enables outbound traffic on your SMTP service ports.

To configure SMTP for Fusion:

1. Log in to the CDF Management Portal.
2. Click **DEPLOYMENT**, and select **Deployments**.

3. Select **Reconfigure** in the **Three Dots** menu ⋮ and navigate to Fusion > User Management Configuration.
4. Configure SMTP:
   a. SMTP TLS Enable (Enable it for TLS, or disable it for non-TLS.)
   b. Add the certificate for the SMTP service.
   c. Add the SMTP host URL.
   d. Add the SMTP port number.
   e. Enter the SMTP Admin name.
   f. Enter the SMTP Admin password.
   g. Add the SMTP Admin email address.
   h. Click **Save** to activate the configuration changes.

      This will automatically restart application pods that offer email service.

If an installed certificate expires, its path changes or a fresh one is generated. When this happens, you must re-import it using the same process above.

If you connect to an SMTP server as an anonymous user or do not have a user name and password configured, you must add a dummy password and save.

Important: Message size constraints are applied according to the message size policy for your SMTP Service. Emailing report assets is one example that increases message size. If you encounter message limit or size warnings, or other errors, contact your SMTP administrator.

Example warning (AWS default):

AWS SES SMTP email attachments cannot exceed 10 MB. Contact your cloud administrator. See https://docs.aws.amazon.com/ses/latest/dg/quotas.html.

Fusion ArcMC SMTP

To configure SMTP for Fusion ArcMC follow the steps in "SMTP" on page 1074.

Configuring an External Identity Provider

Password-based authentication requires users to specify their User ID and Password when logging in. You can select the built-in authentication or external authentication, such SAML or LDAP.

- "Configuring LDAP Authentication" on the next page
- "Configuring SAML Authentication" on page 397

Configuring LDAP Authentication

The identity provider (IDP) user and password has governance over the platform; therefore, the user must exist in both systems, but the password is validated only in LDAP. This section details LDAP authentication steps when TLS is enabled and disabled.

To use LDAP authentication when TLS is disabled:

1. Create at least one LDAP user to log in into the platform using LDAP authentication.
2. Log in to the CDF server and navigate to the SSO default configuration folder at:

   ```
   <arcsight_nfs_vol_path>/sso/default
   ```

   where `<arcsight_nfs_vol_path>` is the NFS volume used for CDF installation; for example: `/opt/NFS_volume/arcsight-volume.`
3. Open the SSO configuration file (sso-configuration.properties), and review the LDAP parameters.

   ```
   ##### The following LDAP confgs are not utilized at this time
   # com.microfocus.sso.default.ldap.enabled = true
   # com.microfocus.sso.default.login.method = np-ldap
   # com.microfocus.sso.default.ldap.admin-dn = CN=bind_
   user,cn=Users,dc=ospad,dc=test
   # com.microfocus.sso.default.ldap.admin-pwd = password
   # com.microfocus.sso.default.ldap.host = xxx.xx.xx.xx
   # com.microfocus.sso.default.ldap.use-tls = false
   # com.microfocus.sso.default.ldap.port = 389
   #---- uncomment these if the LDAP server is Active Directory rather than
   eDirectory
   # com.microfocus.sso.default.ldap.dir-type = AD
   # com.microfocus.sso.default.as.naming-attr = sAMAccountName
   # com.microfocus.sso.default.as.users-container-dn = cn=Users,dc=ospad,dc=test
   ## uncomment these to configure URL when LDAP user forgets password
   # com.microfocus.sso.default.ldap.forgotten-pwd-url =
   # com.microfocus.sso.default.ldap.login.forgotten-password-target = _blank
   # com.microfocus.sso.default.ldap.login.forgotten-password-text-res-id =
   # com.microfocus.sso.default.ldap.login.forgotten-password-title-res-id =
   ```
4. Update the SSO configuration file (sso-configuration.properties) for your LDAP log in method by uncommenting (removing the #) of these lines in the sso-configuration.properties file, and completing the information for your LDAP environment.

   ```
   com.microfocus.sso.default.ldap.enabled = true
   com.microfocus.sso.default.login.method = np-ldap
   com.microfocus.sso.default.ldap.admin-dn = provide your LDAP User DN here
   com.microfocus.sso.default.ldap.admin-pwd = provide your LDAP Admin password here
   ```

```
com.microfocus.sso.default.ldap.host = provide your LDAP host here
com.microfocus.sso.default.ldap.use-tls = true (this corresponds to your LDAP TLS setting, true or false. However, a "false" value will fail to enable a TLS LDAP connection)
com.microfocus.sso.default.ldap.port = 636 (your LDAPS Environment port may differ - change accordingly)
```

5. For Active Directory rather than eDirectory:
   a. Update the SSO configuration file (sso-configuration.properties) to enable AD by uncommenting these lines in the sso-configuration.properties file, and completing the information for your LDAP environment.

      ```
      com.microfocus.sso.default.ldap.dir-type = AD
      com.microfocus.sso.default.as.naming-attr = provide your AD attribute here
      com.microfocus.sso.default.as.users-container-dn = provide your LDAP Base DN here
      ```

   b. Save the SSO configuration file (sso-configuration.properties).

To configure the system to require users to login with an email address (recommended), set `com.microfocus.sso.default.as.naming-attr` to 'mail'. Otherwise, to require users to login with their Active Directory username, set `com.microfocus.sso.default.as.naming-attr` to 'sAMAccountName'.

6. For URL configuration when an LDAP user forgets the password:
   a. Update the SSO configuration file (sso-configuration.properties) to enable *forgot password* for the LDAP user by uncommenting these lines in the sso-configuration.properties file, and completing the information for your LDAP environment.

      ```
      com.microfocus.sso.default.ldap.forgotten-pwd-url = provide your LDAP url for forgotten password here
      com.microfocus.sso.default.ldap.login.forgotten-password-target = provide the target here
      com.microfocus.sso.default.ldap.login.forgotten-password-text-res-id = provide the text to be shown here
      com.microfocus.sso.default.ldap.login.forgotten-password-title-res-id = provide the title to be shown here
      ```

   b. Save the SSO configuration file (sso-configuration.properties).
7. Restart the fusion-single-sign-on pod.
   a. Get the fusion-single-sign-on pod information:

      ```
      kubectl get pods --all-namespaces | grep single-sign
      ```

   b. Restart the fusion-single-sign-on by deleting the currently running pod:

```
kubectl delete pod fusion-single-sign-on-xxxxxxxxxx-xxxxx -n arcsight-
installer-xxxxx
```

8. Log in using your LDAP credentials.

To use LDAP authentication when TLS is Enabled:

1. Create at least one LDAP user to log in into the platform using LDAP authentication.
2. Log in to the CDF server and navigate to the SSO default configuration folder at:

   ```
   <arcsight_nfs_vol_path>/sso/default
   ```

   where `<arcsight_nfs_vol_path>` is the NFS volume used for CDF installation; for example: `/opt/NFS_volume/arcsight-volume`.
3. Open the SSO configuration file (sso-configuration.properties), and review the LDAP parameters.

   ```
   ##### The following LDAP confgs are not utilized at this time
   # com.microfocus.sso.default.ldap.enabled = true
   # com.microfocus.sso.default.login.method = np-ldap
   # com.microfocus.sso.default.ldap.admin-dn = CN=bind_
   user,cn=Users,dc=ospad,dc=test
   # com.microfocus.sso.default.ldap.admin-pwd = password
   # com.microfocus.sso.default.ldap.host = xxx.xx.xx.xx
   # com.microfocus.sso.default.ldap.use-tls = true
   # com.microfocus.sso.default.ldap.port = 636
   #---- uncomment these if the LDAP server is Active Directory rather than
   eDirectory
   # com.microfocus.sso.default.ldap.dir-type = AD
   # com.microfocus.sso.default.as.naming-attr = mail
   # com.microfocus.sso.default.as.users-container-dn = cn=Users,dc=ospad,dc=test
   ## uncomment these to configure URL when LDAP user forgets password
   ```
4. Update the SSO configuration file (sso-configuration.properties) for your LDAP log in method by uncommenting (remove the #) these lines in the sso-configuration.properties file, and completing the information for your LDAP environment.

   ```
   com.microfocus.sso.default.ldap.enabled = true
   com.microfocus.sso.default.login.method = np-ldap
   com.microfocus.sso.default.ldap.admin-dn = provide your LDAP User DN here
   com.microfocus.sso.default.ldap.admin-pwd = provide your LDAP Admin password here
   com.microfocus.sso.default.ldap.host = provide your LDAP host here
   com.microfocus.sso.default.ldap.use-tls = true provide your LDAP TLS setting here
   (true/false)
   com.microfocus.sso.default.ldap.port = 636 provide your LDAP port here
   ```
5. Create and copy the PEM formatted CA LDAP server certificate into the Fusion single-sign-on pod:

```
kubectl cp /opt/ldapCA.cer arcsight-installer-xxxx/fusion-single-sign-on-
xxxxxxx-xxxx:/tmp -c fusion-single-sign-on
```

6. Log in to the into the Fusion-single-sign-on pod:

```
kubectl exec -it fusion-single-sign-on-xxxxxxx-xxxx -n arcsight-
installer-xxxx -c fusion-single-sign-on -- sh
```

7. Navigate to the Fusion truststore directory:

```
cd /usr/local/tomcat/conf/default/
```

8. Install the PEM formatted CA LDAP server certificate into the Fusion single-sign-on truststore:

```
keytool -importcert -storepass $KEYSTORE_PASSWORD -destkeystore sso.bcfks
-alias ldapCA -file /tmp/okta.cert -storetype BCFKS -providerclass
org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider -providerpath
/usr/local/openjdk-8/jre/lib/ext/bc-fips-1.0.2.1.jar
```

9. Verify the list of certificates in the fusion-single-sign-on trustore:

```
keytool -list -v -alias ldapCA -keystore sso.bcfks -storepass $KEYSTORE_
PASSWORD
```

10. Close the Terminal session to the pod:

```
exit
```

11. Restart the Fusion single-sign-on pod:

```
kubectl delete pod -n arcsight-installer-xxxx fusion-single-sign-on-
xxxxxxx-xxxx
```

Configuring SAML Authentication

This section provides the steps to integrate SSO with an external SAML 2.0 IDP solution, such as NetIQ Advanced Authentication.

Fusion SSO and external SAML 2.0 IDP should be time-synchronized to the same NTP server. In the configuration UI, the session timeout must be set up with the same value that the external IDP has configured for user session timeouts.

- "Describing Information Regarding the Trusted Provider Metadata" below
- Configuring an External SAML Provider
- "Integrating with an External SAML Provider" on the next page

Describing Information Regarding the Trusted Provider Metadata

The metadata document for a trusted SAML provider with which a SSO defined provider interacts must be obtained in a provider-specific manner. While not all providers do so, many supply their metadata documents via URL.

After the trusted provider's metadata document (or the URL-accessible location of the document) is obtained, you must configure the SSO provider that will interact with the trusted provider's metadata.

In the document, modify the `<Metadata>` element within the `<AccessSettings>` element under either the `<TrustedIDP>` element or the `<TrustedSP>` element.

For example:

```
com.microfocus.sso.default.login.saml2.mapping-attr = email
```

The email attribute refers to the email attribute name from the SAML 2.0 IDP.

Configuring an External SAML Provider

Use the metadata URL of Fusion SSO to derive the specific single sign-on and single log-out URLs to configure an external SAML 2.0 IDP. These URLs include the following:

- Fusion metadata URL: https://<FQDN of ArcSight Platform Virtual IP for HA or single master node>/osp/a/default/auth/saml2/spmetadata
- Fusion Entity ID or Issuer: https://<FQDN of ArcSight Platform Virtual IP for HA or single master node>/osp/a/default/auth/saml2/metadata
- Fusion single sign-on: https://<FQDN of ArcSight Platform Virtual IP for HA or single master node>/osp/a/default/auth/saml2/spassertion_consumer
- Fusion single log-out: https://<FQDN of ArcSight Platform Virtual IP for HA or single master node>/osp/a/default/auth/saml2/spslo

A user present in the external SAML 2.0 IDP solution must also exist in Fusion to proceed with integration.

Integrating with an External SAML Provider

1. On the NFS server, open the sso-configuration.properties file, located by default in the `<arcsight_nfs_vol_path>/sso/default` directory.

   `<arcsight_nfs_vol_path>` is the nfs volume used for CDF installation.

   For Example:

   ```
   /opt/NFS_volume/arcsight-volume/sso/default
   ```

2. Open the *sso-configuration.properties* file and add the following properties:

   ```
   com.microfocus.sso.default.login.method = saml2
   ```

   ```
   com.microfocus.sso.default.saml2.enabled = true
   ```

3. To specify the address where the IDP supplies its metadata document, complete one of the following actions:

   - Add the following property to the file:

     ```
     com.microfocus.sso.default.login.saml2.metadata-url = <IDP SAML metadata URL>
     ```

   - An example of a Keycloak server URL could be:

     ```
     https://<KeycloakServer>/auth/realms/<YourRealm>/protocol/saml/descriptor
     ```

     The IDP certificates need to be imported to the Fusion SSO keystore for HTTPS to work properly. See Step 5 for more details.

   - Alternatively, you can convert the metadata xml file to base64 string and set the following variable:

     ```
     com.microfocus.sso.default.login.saml2.metadata = <base64 encoded metadata xml>
     ```

4. Save the changes to the `sso-configuration.properties` file.
5. (Conditional) If you specified the metadata URL in Step 3, complete the following steps to import the IDP certificate to the SSO keystore:

a. Copy the IDP certificate to the following location.

```
arcsight_nfs_vol_path/sso/default
```

b. Get the pod information.

```
kubectl get pods --all-namespaces | grep single-sign-on
```

c. Open a terminal in the currently running pod:

```
kubectl exec -it fusion-single-sign-on-xxxxxxxxxx-xxxxx -n arcsight-installer-
xxxxx -c fusion-single-sign-on bash
```

d. Import the IDP certificate:

i.
```
cd /usr/local/tomcat/conf/default/
```

ii.
```
keytool -importcert -storepass $KEYSTORE_PASSWORD -destkeystore \
sso.bcfks -alias AliasName -file CertificateFileName -storetype \
BCFKS -providerclass \
org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider \
-providerpath /usr/local/openjdk-8/jre/lib/ext/bc-fips-1.0.2.jar
```

- `CertificateFileName` represents the name of the certificate file that you copied to <arcsight_nfs_vol_path>/sso/default/, which automatically displays in your current directory:

```
/usr/local/tomcat/conf/default/
```

- `AliasName` represents the new alias name that you want to assign to the certificate in the SSO keystore.

6. Restart the pod:

a. Get the pod information.

```
kubectl get pods --all-namespaces | grep fusion-single-sign-on
```

b. Delete the current running pod.

```
kubectl delete pod fusion-single-sign-on-xxxxxxxxxx-xxxxx -n arcsight-
installer-xxxxx
```

7. Retrieve the Fusion SSO SAML service provider metadata from the server.

```
https://<FQDN of ArcSight Platform Virtual IP for HA or single master
node>/osp/a/default/auth/saml2/spmetadata
```

8. Use the SSO SAML service provider metadata to configure your IDP. For detailed instructions, see the IDP software documentation.

9. To establish a trust relationship between Fusion SSO and your IDP software, create certificates for your IDP software. For detailed instructions on how to create and import certificates in your IDP software, see the IDP software documentation.

Integrating ESM Data and Users

The Fusion capability allows you to integrate users and data from ESM. With single sign-on (SSO) supported between Fusion and ESM, users can easily access the ArcSight Console, ArcSight Command Center, ESM Command Center, and REST APIs with the same login.

Understanding How ESM Users Access Fusion

Rather than manually adding users to Fusion, we recommend you create users in ESM first, then import them into Fusion.

For the imported ESM users to log in to Fusion and be able to access ESM data, the following conditions apply:

- You must enable SSO access for ESM and Fusion users.
- Users must have an account in both ESM and Fusion.
- You must configure the External User ID and E-mail fields in the ESM accounts to comply with the *name@domain.com* format.
- Users must log in to Fusion with the **External User ID** from their ESM account.
- If your environment does not use external authentication (such as SAML or LDAP), ensure you have configured the SMTP server settings for Fusion. Users imported from ESM might need to set a password the first time they log in, which requires those users to initiate the Forgot Password function and receive an email notification.

Importing Users

You can import users that are already authorized. You need to have at least one role available in Fusion to assign to these users.

Importing ESM users puts them ALL into the preselected Fusion roles. You cannot downselect ESM users once you proceed. Only users with a filled e-mail address in ESM get imported.

1. In the ArcSight Console, ensure that the External User ID and E-mail fields for each account comply with the following format.

   ```
   name@domain.com
   ```

2. To log in to Fusion, use the following format.

```
https://<cdf_masternode_hostname>
```

3. Click **ADMIN** > **Account Groups** > **Import Users**.
4. Select the role that you want to assign to the imported users.
5. Ensure you have a valid ESM Root CA certificate in the **Fusion User Management TrustStore**.

As you add more users to ESM, you can run the import process again. Fusion ignores duplicates of user accounts that have been imported previously.

Enabling SSO with ESM

You must configure ESM to use **OSP Client Only Authentication**. If your ESM environment currently uses external client authentication, you must delegate the Fusion SSO provider to connect to the external authentication client.

If you do not use external authentication, see "Connecting to Your SMTP Server" on page 392 for information on supporting forgotten password activity.

- "Configuring SSO with ESM" below
- "Importing the Fusion Certificate to the Console Keystore" on the next page
- "Configuring the SSO Proxy" on page 404

Configuring SSO with ESM

This procedure assumes you have ESM installed or upgraded.

1. Change the authentication settings for the ESM Manager service:
   a. On the ESM server, start the configuration wizard by changing to the following directory and entering the following.
   Directory

   ```
   /opt/arcsight/manager/bin/
   ```

   Command

   ```
   arcsight managersetup -i console
   ```

   b. Advance through the wizard until you reach the authentication settings.
   c. Select **OSP Client Only Authentication**, then click **Next**.

d. To specify the host and port for the OSP server, use the following format:

```
domain_name:port
```

For example, Fusion by default installs OSP on port 443. So, when you are using Fusion, specify the format as:

```
 <fusion host>:443
```

e. Specify a **Tenant Name for OSP**. If you are using a typical installer for Fusion, specify `default`.

f. Click **Next** until you complete your changes in the wizard.

g. Restart the ESM Manager service using the following commands:

```
/etc/init.d/arcsight_services stop manager
```

```
/etc/init.d/arcsight_services start manager
```

2. Change the authentication settings for the ArcSight Console (the Console):

a. From the Console's `/bin` directory, specify one of the following commands:
Windows

```
arcsight.bat consolesetup
```

Linux

```
 ./arcsight consolesetup
```

b. Advance through the wizard until you reach the authentication settings.

c. Select **OSP Client Only Authentication**.

d. Click **Next** until you complete your changes in the wizard.

Importing the Fusion Certificate to the Console Keystore

1. Obtain the Fusion CA certificate in Base-64 encoded X.509 format.
2. From the Console's `/bin` directory, specify one of the following commands.
Replace `<Fusion CA certificate file path>` and `<alias_name>` with the correct information for your system.

Windows

```
arcsight.bat keytool -store clientcerts -import -file  <Fusion CA certificate
file path> -alias <alias_name>
```

Linux

```
 ./arcsight keytool -store clientcerts -import -file  <Fusion CA certificate file
path> -alias <alias_name>
```

3. Ensure that when you run the keytool command that the JAVA_HOME value listed in the output of the command is pointing to the ArcSight JRE location of your Console to ensure the certificate is imported into the correct keystore.

Configuring the SSO Proxy

If your ArcSight Platform installation is running as a cluster or using a VIP for access, you must configure the SSO proxy settings before ESM and other external clients can access Fusion.

1. Log in to the CDF server and navigate to the SSO default configuration folder:

   ```
                        <arcsight_nfs_vol_path>/sso/default/WEB-
   INF/conf/current/default
   ```

   Replace <arcsight_nfs_vol_path> with the NFS volume used for your CDF installation. For example: /opt/NFS_volume/arcsight-volume

2. Open the tenantcfg.xml file and locate the HTTPInterface sections:

   ```
   <HTTPInterface
          id="default-http-domain"
          displayName="Hercules HTTP"
          path="/osp"
          anyLocalInterface="true"
          proxyPort="443"
          proxyTls="true"
          proxyDomain="${HTTP_INTERFACE_DOMAIN}"
          />
   ```

3. Update the tenantcfg.xml file and and modify the proxyDomain line as shown in the following example.
   Replace <your.domain.name> with your VIP or the FQDN of the master node if you are not using a VIP.

   ```
   <HTTPInterface
          id="default-http-domain"
          displayName="Hercules HTTP"
          path="/osp"
          anyLocalInterface="true"
   ```

```
        proxyPort="443"
        proxyTls="true"
        proxyDomain="<your.domain.name>"
        />
```

4. Save the `tenantcfg.xml` file.
5. Restart the fusion-single-sign-on pod.
   a. Get the fusion-single-sign-on pod information:

      ```
      kubectl get pods --all-namespaces | grep single-sign
      ```

   b. Restart the fusion-single-sign-on by deleting the currently running pod:

      ```
      kubectl delete pod fusion-single-sign-on-xxxxxxxxxx-xxxxx -n arcsight-
      installer-xxxxx
      ```

Integrating Data from ESM

To view ESM data in the Dashboard, update the settings in the CDF Management Portal. The Fusion capability manages the Dashboard functions.

1. Open a new tab in a supported web browser. (For a list of supported web browsers, see Technical Requirements for the ArcSight Platform.)
2. Specify the URL for the CDF Management Portal:

   ```
   https://<cdf_masternode_hostname>:5443
   ```

   Use the fully qualified domain name of the host that you specified in the Connection step during the CDF configuration. Usually, this is the master node's FQDN.
3. Log in to the CDF Management Portal with the credentials of the administrative user that you provided during installation.
4. Select **Reconfigure**.
5. On the **Configuration** page, select **FUSION**.
6. In the **ArcSight ESM Host Configuration** section, complete the following steps:
   a. For **ESM host**, specify the fully-qualified host name or IP address of the server that hosts ESM.
   b. For **ESM port**, specify the port associated with the **ESM host**. The default value is 8443.

Configuring ESM as a Transformation Hub Producer in Distributed Correlation Mode

Distributed event forwarding is available when ESM is installed in distributed correlation mode. The feature allows you to forward events from ESM to Transformation Hub at a high rate. Distributed event forwarding leverages the distributed infrastructure of ESM to allow ESM to spread the work of event forwarding across the cluster, similar to how ESM distributes event correlation. This allows event forwarding to scale horizontally.

Events that ESM forwards to Transformation Hub can subsequently be read by another ESM instance or multiple ESM instances. Those ESM instances do not have to be installed in distributed correlation mode in order to read events from Transformation Hub.

If you need to forward events from ESM to Transformation Hub at a high rate (generally higher than 10K events per second) Micro Focus recommends that you use ESM in distributed correlation mode and use distributed event forwarding.

Note: Distributed Forwarding does not support SSL client- side authentication with Transformation Hub.

Distributed event forwarding requires the following:

- CA certificate of the Transformation Hub cluster
- ESM filter that determines which events to forward

  You can create a filter or use one that is installed with ESM.
- List of broker socket addresses (in the form host:port) of the Transformation Hub cluster to which you want to forward events

  The Transformation Hub documentation refers to these as "worker nodes."
- Transformation Hub topic name to which you want to publish events

Note: Perform the configuration steps that are described below on the persistor node of the ESM cluster.

Obtaining and Importing the Transformation Hub CA Certificate

Transformation Hub maintains its own certificate authority (CA) to issue certificates for individual nodes in the Transformation Hub cluster. ESM needs that CA certificate in its truststore so that it will trust connections to Transformation Hub. For information about

obtaining the certificate, see the information about viewing and changing the certificate authority. You might need to contact the Transformation Hub administrator to obtain the CA certificate if you do not have sufficient privileges to access the Transformation Hub cluster.

If you have root access to the Transformation Hub cluster (version 3.x or later), you can obtain the CA certificate as follows:

```
master=<master node host name or IP address>
```

```
ssh root@$master env K8S_HOME=/opt/arcsight/kubernetes
/opt/arcsight/kubernetes/scripts/cdf-updateRE.sh >
/opt/arcsight/manager/th.ca.crt
```

The command copies the CA certificate to the file `/opt/arcsight/manager/th.ca.crt`. You must then import that certificate into the ESM cluster using the certadmin tool.

To import the Transformation Hub CA certificate to ESM:

1. Run the following command to ensure that the ESM cluster is running:

   ```
   /etc/init.d/arcsight_services status
   ```

   If it is not, run the following command:

   ```
   /etc/init.d/arcsight_services start
   ```

   Wait for the cluster to completely start.
2. Import the Transformation Hub CA certificate:

   ```
   bin/arcsight certadmin -importcert /opt/arcsight/manager/th.ca.crt
   ```

   Note the alias that is reported in the output.
3. Run the following command and verify that the alias that was reported in the output from the previous command is listed as an approved certificate:

   ```
   bin/arcsight certadmin -list approved
   ```

   If the alias is not listed, re-import the Transformation Hub CA certificate.

Configuring the Filter and Destination

After you import the Transformation Hub CA certificate to ESM, configure a filter and the destination properties. You specify the destination properties in a file that is used as input to the configure-event-forwarding command.

To configure the filter and destination:

1. Create a filter to select the events that you want to forward or use a filter that is installed with ESM.

   For more information about creating event filters, see the information about filtering events in the ArcSight Console User's Guide.
2. Edit `forwarding.properties` in `/opt/arcsight/manager/config/` with the following information:
   - Name of the filter that you want to use to select the events to be forwarded.
   - Comma separated list of socket addresses of the worker nodes of the Transformation Hub cluster. Each socket address should be in the form host:port.

   Note: Optionally, you can copy `forwarding.properties` to a different file and edit that file. If so, you must specify the file name when running the `configure-event-forwarding` utility. Otherwise, `configure- event- forwarding` will read the default `forwarding.properties` file.

   The properties file is in the following format:

```
# Provide the filter that determines which events to forward.
# The filter may be specified by ID or by Name.
# If name is used then it can be the simple name or the fully qualified URI.
#
# e.g. filterID=2jq5Og-sAABCAFyopCdLChw==
# or
# filterName=Non-ArcSight Internal Events
# or
# filterName=/All Filters/ArcSight System/Event Types/Non-ArcSight Internal Events
filterName=Non-ArcSight Internal Events
# List the socket addresses of the destination worker nodes
hostPortCSV=host1.example.com:9093,host2.example.com:9093,host3.example.com:9093
# Specify the destination topic name
topicName=esm-forwarded-events
```

3. Run the following command to validate the settings in the properties file:

```
bin/arcsight configure-event-forwarding -validate <file>
```

> **Note:** If you do not specify a file name, <ARCSIGHT_HOME>/config/forwarding.properties is the default file.

The command output indicates whether the filter and connections to the forwarding destination are valid.

4. If the settings in forwarding.properties are valid, run the following command to set the configuration and save it in the information repository:

```
bin/arcsight configure-event-forwarding -commit <file>
```

> **Note:** If you do not specify a file name, <ARCSIGHT_HOME>/config/forwarding.properties is the default file.

5. When you are ready to actively filter and forward events, run the following command to enable event forwarding:

```
bin/arcsight configure-event-forwarding -enable
```

To disable event forwarding, run the following command:

```
bin/arcsight configure-event-forwarding -disable
```

> **Note:** ESM does not cache events when you disable distributed event forwarding. Events that ESM ingests while forwarding is disabled will not be forwarded, even when you subsequently enable forwarding. Events that ESM ingests after you enable forwarding will be forwarded.

6. To view the current settings for distributed event forwarding, including whether it is enabled or disabled, run the following command:

```
bin/arcsight configure-event-forwarding -print
```

Modifying the Filter and Configuration

The properties file is only used to validate and commit the configuration. The information repository stores the configuration itself. If you need to modify the configuration, it is not sufficient to simply edit the file. You must edit the file and then run bin/arcsight configure-event-forwarding -validate <file> and bin/arcsight configure-event-forwarding -commit <file> again to overwrite the old configuration. It is not necessary to stop and start the ESM cluster to modify the configuration. Distributed event forwarding will automatically detect the updated configuration. You can run bin/arcsight configure-event-forwarding -print to view the configuration that is currently in use.

If you modify the filter that distributed event forwarding uses, event forwarding automatically detects the updated filter and begins using it to select events for forwarding as soon as you save the filter update. Therefore, be cautious when modifying the filter.

Instead of modifying the filter, you can create a new filter and test it before you commit to using it. When you are sure that the filter is correct, specify the new filter in the properties file, then run `bin/arcsight configure-event-forwarding -validate <file>` and `bin/arcsight configure-event-forwarding -commit <file>` to apply the change to the current configuration.

Troubleshooting Event Forwarding Throughput

The maximum rate at which events can be forwarded depends on many factors, including the following:

- Amount of other work that distributed correlation must do to support all of the rules, data monitors, and other content that you have defined
- File input/output contention
- CPU contention
- Memory contention
- Network contention, especially the network between distributed correlation nodes and the Transformation Hub worker nodes
- Maximum rate at which Transformation Hub can accept messages

It might be that the maximum throughput of event forwarding is not enough to support the number of events that need to be forwarded in a given period of time. When that happens, events yet to be forwarded will build up inside message bus. This buildup of unforwarded events is called **lag**. If the lag is too high, the ESM cluster will stop ingesting events to reduce the lag. When this happens it is called **backpressure**.

The Acceptable Lag value in the Cluster View dashboard of Command Center defines the amount of lag that can occur before backpressure is applied. For more information, see the information about using the Cluster View dashboard in the ArcSight Command Center User's Guide.

Distributed event forwarding leverages the ESM distributed infrastructure to gain horizontal scalability. The correlator does the work of event forwarding. You might be able to add event forwarding throughput by adding correlator(s), either on an existing node or on a new distributed correlation node. However, this will only increase throughput if it is the correlators that are causing the bottleneck (for example, because of CPU or memory limitations). If the network or Transformation Hub is causing the bottleneck, adding correlators might not have any effect. For information about adding correlators, see Adding Correlators and Aggregators.

Optionally, you can disable backpressure that might occur as a result of unforwarded events, but you should only use this option if you accept that some events might never be forwarded. To disable backpressure, run the following command:

```
bin/arcsight configure-event-forwarding -backpressure disable
```

Note: When you disable backpressure, ESM will not slow the ingestion of events if event forwarding cannot keep up with the rate at which events are being filtered for forwarding. The lag might build up inside message bus to the point that the oldest unforwarded events are dropped. Those events will not be forwarded.

To enable backpressure, run the following command:

```
bin/arcsight configure-event-forwarding -backpressure enable
```

Configuring ESM as a Transformation Hub Consumer

Note: For Azure or AWS Transformation Hub, you must complete configuration procedures as detailed in Completing Integration with Azure Transformation Hub or Completing Integration with AWS Transformation Hub before you perform the procedures in this section.

This section describes how to configure ESM to consume events from Transformation Hub in both FIPS and non-FIPS mode, including setting up SSL client-side authentication in both modes:

- Configuring ESM as a Transformation Hub Consumer – Non-FIPS Mode
- Setting Up SSL Client-Side Authentication Between Transformation Hub and ESM - Non-FIPS Mode
- Configuring ESM as a Transformation Hub Consumer - FIPS Mode (Server Authentication Only)
- Setting Up SSL Client-Side Authentication Between Transformation Hub and ESM - FIPS Mode

For ESM in distributed correlation mode, you can also configure ESM to send events to Transformation Hub. For more information, see Configuring ESM as a Transformation Hub Producer in Distributed Correlation Mode.

Configuring ESM as a Transformation Hub Consumer – Non-FIPS Mode

This procedure uses the CA certificate that is embedded in Transformation Hub.

To complete the configuration, complete the following tasks:

1. Obtain the Transformation Hub CA certificate.
2. On the ESM server, configure ESM to consume from Transformation Hub.

The steps for each task are outlined below.

Obtaining the Transformation Hub CA Certificate

Transformation Hub maintains its own certificate authority (CA) to issue certificates for individual nodes in the Transformation Hub cluster. ESM needs that CA certificate in its truststore so that it will trust connections to Transformation Hub. For information about obtaining the certificate, see the information about viewing and changing the certificate authority. You might need to contact the Transformation Hub administrator to obtain the CA certificate if you do not have sufficient privileges to access the Transformation Hub cluster.

If you have `root` access to the Transformation Hub cluster (version 3.x or later), you can obtain the CA certificate as follows:

```
master=<master node host name or IP address>
```

```
ssh root@$master env K8S_HOME=/opt/arcsight/kubernetes
/opt/arcsight/kubernetes/scripts/cdf-updateRE.sh >
/opt/arcsight/manager/th.ca.crt
```

The command copies the CA certificate to the file `/opt/arcsight/manager/th.ca.crt`.

Configuring ESM to Consume from Transformation Hub

1. Run the following command:

   ```
   /opt/arcsight/manager/bin/arcsight managersetup -i console
   ```

2. In the wizard, press **Enter** until the wizard asks whether you want to read events from Transformation Hub. Select **Yes**, then provide the following information:

   a. Host name and port information for the worker nodes in Transformation Hub. Use a comma-separated list (for example: <host>:<port>,<host>:<port>) and specify the FQDN of the worker nodes.

   > **Note:** You must specify the host name and *not* the IP address.
   >
   > Transformation Hub can only accept IPv4 connections from ESM.
   >
   > If the Kafka cluster is configured to use SASL/PLAIN authentication, ensure that you specify the port configured in the cluster for the SASL_SSL listener.

b. Topics in Transformation Hub from which you want to read. These topics determine the data source.

> **Note:** You can specify up to 25 topics using a comma-separated list (for example: topic1,topic2). ESM will read Avro-format events from any topic where the name contains "avro" in lower case. For example, `th-arcsight-avro`.

c. Path to the Transformation Hub root certificate (`/opt/arcsight/manager/th.ca.crt`).

d. Leave the authentication type as None.

e. Leave the user name and password as empty.

f. If you specified an Avro topic, specify the host name and port for connecting to the Schema Registry in the format <host name:port>.

> **Note:** The default port for connecting to the Schema Registry is 32081.

Transformation Hub runs a Confluent Schema Registry that producers and consumers use to manage compatibility of Avro-format events.

The wizard uses this information to connect to the Schema Registry, read the Avro schemas for the Avro topic that you specified, and verify that the topic contains Avro events that are compatible with ESM. If ESM cannot retrieve the Avro schemas for the Avro topic that you specified and compare it to the event schema that is packaged with ESM, or if incompatible schemas are detected, the wizard generates warning messages but allows you to continue. In some cases, you might already know that Transformation Hub will use a compatible schema when the Manager is running.

g. If you choose to configure the Forwarding Connector to forward CEF events to Transformation Hub and then configure Transformation Hub to filter Avro events, use filters to ensure that ESM does not receive duplicate events. You might want to use filters to accomplish the following:

- Filter out desired events from Connectors so that ESM does not process them.
- Filter out ESM's correlation events that were forwarded (CEF events that the Forwarding Connector sent to `th-cef`) so that ESM does not re-process its own events.

  If you do *not* configure filtering, ESM must consume from the `th-arcsight-avro` topic. If you configure filtering, ESM must consume from the `mf-event-avro-esmfiltered` topic. For more information, see configuring filters and local and global event enrichment.

After providing the information, specify **Yes** and complete the remaining sections of the wizard.

3. After you complete the wizard, restart the Manager services:

   In compact mode:

   ```
   /etc/init.d/arcsight_services stop manager
   ```

   ```
   /etc/init.d/arcsight_services start manager
   ```

   In distributed mode:

   ```
   /etc/init.d/arcsight_services stop all
   ```

   ```
   /etc/init.d/arcsight_services start all
   ```

4. Verify that the connection was successful:

   ```
   grep -rnw '/opt/arcsight/var/logs/manager/' -e 'Transformation Hub
   service is initialized' -e 'Started kafka readers'
   ```

   The output should be similar to the following:

   ```
   /opt/arcsight/var/logs/manager/default/server.std.log:5036:2021-07-13
   09:51:36 =====> Transformation Hub service is initialized (49 s) <=====
   /opt/arcsight/var/logs/manager/default/server.log:11664:[2021-07-13
   09:51:36,656][INFO ][default.com.arcsight.common.messaging.events.aa]
   Started kafka readers in PT0.115S
   /opt/arcsight/var/logs/manager/default/server.log:11665:[2021-07-13
   09:51:36,657][INFO ][default.com.arcsight.server.NGServer] Transformation
   Hub service is initialized (49 s)
   ```

Setting Up SSL Client-Side Authentication Between Transformation Hub and ESM - Non-FIPS Mode

ArcSight Platform maintains its own certificate authority (CA) to issue certificates for individual nodes in the Transformation Hub cluster and external communication. ESM needs the signed certificates in its truststore so that it will trust connections to the Arcsight Platform and Transformation Hub. You might need to contact the ArcSight Platform administrator to obtain the signed certificates if you do not have sufficient privileges to access them and run the necessary commands.

Note: When configuring Transformation Hub access, you must specify the FQDN of the ArcSight Platform virtual IP for HA or single master node and *not* the IP address.

To complete the configuration, complete the following tasks:

- Enable client-side authentication between Transformation Hub and ESM
- Configure ESM to consume from Transformation Hub.

Enabling Client-side Authentication Between Transformation Hub and ESM:

1. Verify that Transformation Hub is functional and that client authentication is configured.
2. As user `arcsight`, stop the ArcSight Manager:

   ```
   /etc/init.d/arcsight_services stop manager
   ```

3. If `/opt/arcsight/manager/config/client.properties` does not exist, create it using an editor of your choice.
4. Change the store password for the keystore, `keystore.client`, which has an empty password by default. This empty password interferes with the certificate import:

   ```
   /opt/arcsight/manager/bin/arcsight keytool -store clientkeys -storepasswd
   -storepass ""
   ```

5. Run the following command to update the empty password of the generated key `services-cn` in the keystore to be the same password as that of the keystore itself. When prompted, specify the same password that you entered for the store password:

   ```
   /opt/arcsight/manager/bin/arcsight keytool -store clientkeys -keypasswd -
   keypass "" -alias services-cn
   ```

6. Run the following command to update the password in `config/client.properties`:

   ```
   /opt/arcsight/manager/bin/arcsight changepassword -f
   config/client.properties -p ssl.keystore.password
   ```

7. Generate the keypair and certificate signing request (`.csr`) file. When generating the keypair, specify the fully qualified domain name of the ArcSight Manager host as the common name (CN) for the certificate.

   Run the following commands:

   ```
   export COMMON_NAME=<your ESM host's fully qualified domain name>
   ```

   ```
   /opt/arcsight/manager/bin/arcsight keytool -store clientkeys -genkeypair
   -dname "cn=${COMMON_NAME}, ou=<your organization>, o=<your company>,
   c=<your country>" -keyalg rsa -keysize 2048 -alias ebkey -startdate -1d -
   validity 366
   ```

```
/opt/arcsight/manager/bin/arcsight keytool -certreq -store clientkeys -
alias ebkey -file  ${COMMON_NAME}.csr
```

where `${COMMON_NAME}.csr` is the output file where the `.csr` is stored.

8. To sign the ESM certificate signing request, perform the following steps in the ArcSight Platform. For an on-premises deployment, perform the steps on the master node. For a cloud deployment, perform the steps on the Bastion host.

   a. Create a temporary folder to store the generated certificates:

   ```
   mkdir –m 700 /tmp/esm
   ```

   b. Move the certificate signing request (`.csr`) file from the ESM host to the temporary folder that you created.

   c. Set the environment variables:

   ```
   export CA_CERT=re_ca.cert.pem
   ```

   ```
   export COMMON_NAME=<your ESM host's fully qualified domain name>
   ```

   ```
   export TH=<FQDN of the ArcSight Platform virtual IP for HA or single
   master node>_<Kafka TLS-enabled port>
   ```

   Note: For `COMMON_NAME`, use the same host FQDN as you used for the ESM client key pair.

   d. Run the following commands to sign the ESM certificate signing request:

   ```
   cd /tmp/esm
   ```

   ```
   export CDF_APISERVER=$(kubectl get pods -n core -o custom-
   columns=":metadata.name"| grep cdf-apiserver)
   export PASSPHRASE=$(kubectl get secret vault-passphrase -n core -o
   json 2>/dev/null | jq -r '.data.passphrase')
   export ENCRYPTED_ROOT_TOKEN=$(kubectl get secret vault-credential -n
   core -o json 2>/dev/null | jq -r '.data."root.token"')
   export VAULT_TOKEN=$(echo ${ENCRYPTED_ROOT_TOKEN} | openssl aes-256-
   cbc -md sha256 -a -d -pass pass:"${PASSPHRASE}")
   export CSR=$(cat ${COMMON_NAME}.csr)
   ```

   ```
   WRITE_RESPONSE=$(kubectl exec -it -n core ${CDF_APISERVER} -c cdf-
   apiserver -- bash -c "VAULT_TOKEN=$VAULT_TOKEN vault write -tls-skip-
   verify -format=json RE/sign/coretech csr=\"${CSR}\"") && \
   echo "$WRITE_RESPONSE" | jq -r ".data | .certificate" > ${COMMON_
   ```

```
NAME}.signed.crt && \
echo "$WRITE_RESPONSE" | jq -r ".data | .issuing_ca" > ${COMMON_
NAME}.issue_ca.crt && \
echo "$WRITE_RESPONSE" | jq -r ".data | .certificate, if .ca_chain
then .ca_chain[] else .issuing_ca end" > ${COMMON_
NAME}.signed.cert.with.ca.crt
```

The signed certificate is in the file `${COMMON_NAME}.signed.crt`. The issuing CA is in the file `${COMMON_NAME}.issue_ca.crt`. The signed certificate with the CA chain is in the file `${COMMON_NAME}.signed.cert.with.ca.crt`.

9. Retrieve the RE certificates:

   For an on-premises deployment:

   ```
   cd /opt/arcsight/kubernetes/scripts/
   ```

   ```
   ./cdf-updateRE.sh > /tmp/esm/${CA_CERT}
   ```

   For a cloud deployment:

   ```
   cd {path to cdf installer}/cdf-deployer/scripts/
   ```

   ```
   ./cdf-updateRE.sh > /tmp/esm/${CA_CERT}
   ```

10. Copy the following files from the Transformation Hub `/tmp/esm` folder to an ESM host folder (for example, `/opt/arcsight/tmp`):

    `/tmp/esm/${COMMON_NAME}.signed.cert.with.ca.crt`

    `/tmp/esm/${CA_CERT}`

    Remove the files from `/tmp/esm` after you copy them.

11. On the ESM server, import the RE certificate from file ${CA_CERT} into the ESM client truststore:

    ```
    /opt/arcsight/manager/bin/arcsight keytool -store clientcerts -alias
    <alias for the certificate> -importcert -file <absolute path to
    certificate file>
    ```

    For example:

    ```
    /opt/arcsight/manager/bin/arcsight keytool -store clientcerts -alias
    thcert -importcert -file /opt/arcsight/tmp/re_ca.cert.pem
    ```

Note: You might receive the following message:

```
Certificate already exists in keystore under alias <alias1>
Do you still want to add it? [no]:
```

It is not necessary to add an existing certificate.

12. On the ESM server, run the following command to import the signed certificate:

```
/opt/arcsight/manager/bin/arcsight keytool -store clientkeys -alias
<alias for the key> -importcert -file <path to signed cert> -trustcacerts
```

For example:

```
/opt/arcsight/manager/bin/arcsight keytool -store clientkeys -alias ebkey
-importcert -file /opt/arcsight/tmp/${COMMON_NAME}.signed.cert.with.ca.crt
–trustcacerts
```

Note: You might see the following warning:

```
...
Top-level certificate in reply:
...
... is not trusted. Install reply anyway? [no]:
```

This is because the root certificate of the RE CA is not in the ESM truststore. This does not affect the functionality of ESM. Enter **yes** to allow the new certificate to be imported.

Configuring ESM to Consume from Transformation Hub

1. Run the following command:

```
/opt/arcsight/manager/bin/arcsight managersetup -i console
```

2. In the wizard, press **Enter** until the wizard asks whether you want to read events from Transformation Hub. Select **Yes**, then provide the following information:
   a. Host name and port information for the worker nodes in Transformation Hub. Use a comma-separated list (for example: <host>:<port>,<host>:<port>) and specify the FQDN of the worker nodes.

Note: You must specify the host name and *not* the IP address.

Transformation Hub can only accept IPv4 connections from ESM.

If the Kafka cluster is configured to use SASL/PLAIN authentication, ensure that you specify the port configured in the cluster for the SASL_SSL listener.

b. Topics in Transformation Hub from which you want to read. These topics determine the data source.

> **Note:** You can specify up to 25 topics using a comma-separated list (for example: topic1,topic2). ESM will read Avro-format events from any topic where the name contains "avro" in lower case. For example, `th-arcsight-avro`.

c. Leave the path to the Transformation Hub root certificate empty, as you already imported the certificates.

d. Leave the authentication type as None.

e. Leave the user name and password empty.

f. If you specified an Avro topic, specify the host name and port for connecting to the Schema Registry in the format <FQDN of the ArcSight Platform virtual IP for HA or single master node:port>.

> **Note:** The default port for connecting to the Schema Registry is 32081.

Transformation Hub runs a Confluent Schema Registry that producers and consumers use to manage compatibility of Avro-format events.

The wizard uses this information to connect to the Schema Registry, read the Avro schemas for the Avro topic that you specified, and verify that the topic contains Avro events that are compatible with ESM. If ESM cannot retrieve the Avro schemas for the Avro topic that you specified and compare it to the event schema that is packaged with ESM, or if incompatible schemas are detected, the wizard generates warning messages but allows you to continue. In some cases, you might already know that Transformation Hub will use a compatible schema when the Manager is running.

g. If you choose to configure the Forwarding Connector to forward CEF events to Transformation Hub and then configure Transformation Hub to filter Avro events, use filters to ensure that ESM does not receive duplicate events. You might want to use filters to accomplish the following:

- Filter out desired events from Connectors so that ESM does not process them.
- Filter out ESM's correlation events that were forwarded (CEF events that the Forwarding Connector sent to `th-cef`) so that ESM does not re-process its own events.

  If you do *not* configure filtering, ESM must consume from the `th-arcsight-avro` topic. If you configure filtering, ESM must consume from the `mf-event-avro-esmfiltered` topic. For more information, see configuring filters and local and global event enrichment.

After providing the information, specify **Yes** and complete the remaining sections of the wizard.

3. Start the ArcSight Manager:

   In compact mode:

   ```
   /etc/init.d/arcsight_services start manager
   ```

   In distributed mode:

   ```
   /etc/init.d/arcsight_services stop all
   ```

   ```
   /etc/init.d/arcsight_services start all
   ```

   Ensure that all services started:

   ```
   /etc/init.d/arcsight_services status
   ```

4. Verify that the connection was successful:

   ```
   grep -rnw '/opt/arcsight/var/logs/manager/' -e 'Transformation Hub
   service is initialized' -e 'Started kafka readers'
   ```

   The output should be similar to the following:

   ```
   /opt/arcsight/var/logs/manager/default/server.std.log:5036:2021-07-13
   09:51:36 =====> Transformation Hub service is initialized (49 s) <=====
   /opt/arcsight/var/logs/manager/default/server.log:11664:[2021-07-13
   09:51:36,656][INFO ][default.com.arcsight.common.messaging.events.aa]
   Started kafka readers in PT0.115S
   /opt/arcsight/var/logs/manager/default/server.log:11665:[2021-07-13
   09:51:36,657][INFO ][default.com.arcsight.server.NGServer] Transformation
   Hub service is initialized (49 s)
   ```

Configuring ESM as a Transformation Hub Consumer - FIPS Mode (Server Authentication Only)

This section describes how to configure ESM to access Transformation Hub when FIPS mode is enabled. FIPS 140-2 is the only supported FIPS mode.

To configure ESM access to Transformation Hub in FIPS Mode:

1. As user `arcsight`, stop the ArcSight Manager:

   ```
   /etc/init.d/arcsight_services stop manager
   ```

2. From the Transformation Hub server, copy the certificate from `/opt/arcsight/kubernetes/scripts/cdf-updateRE.sh > /tmp/ca.crt` to a location on the ESM server.
3. Use the `keytool` command to import the root CA certificate into the ESM client truststore:

   ```
   /opt/arcsight/manager/bin/arcsight keytool -store clientcerts -importcert
   -file <absolute path to certificate file> -alias <alias for the
   certificate>
   ```

   For example:

   ```
   /opt/arcsight/manager/bin/arcsight keytool -store clientcerts -importcert
   -file /tmp/ca.crt -alias alias1
   ```
4. As user `arcsight`, run the following command from the `/opt/arcsight/manager/bin` directory to start the `managersetup` wizard:

   ```
   ./arcsight managersetup -i console
   ```
5. Provide the following information:

   Note: You do not need to provide the path to the Transformation Hub root certificate, as it has already been imported.

   a. Specify the host name and port information for the nodes in Transformation Hub. Include the host and port information for all nodes and not just the master node. Use a comma-separated list (for example: <host>:<port>,<host>:<port>).

      Note: You must specify the host name and *not* the IP address.

      Transformation Hub can only accept IPv4 connections from ESM.

      If the Kafka cluster is configured to use SASL/PLAIN authentication, ensure that you specify the port configured in the cluster for the SASL_SSL listener.

   b. Specify the topics in Transformation Hub from which you want to read. These topics determine the data source.

      Note: You can specify up to 25 topics using a comma-separated list (for example: topic1,topic2). ESM will read Avro-format events from any topic where the name contains "avro" in lower case. For example, `th-arcsight-avro`.

   c. Leave the authentication type as None.

   d. Leave the user name and password as empty.

   e. If you specified an Avro topic, specify the host name and port for connecting to the Schema Registry in the format <host name:port>.

Note: The default port for connecting to the Schema Registry is 32081.

Transformation Hub runs a Confluent Schema Registry that producers and consumers use to manage compatibility of Avro-format events.

The wizard uses this information to connect to the Schema Registry, read the Avro schemas for the Avro topics that you specified, and verify that the topics contain Avro events that are compatible with ESM. If ESM cannot retrieve the Avro schemas for the Avro topics that you specified and compare them to the schema that is packaged with ESM, or if incompatible schemas are detected, the wizard generates warning messages but allows you to continue. In some cases, you might already know that Transformation Hub will use a compatible schema when the Manager is running.

f. If you choose to configure the Forwarding Connector to forward CEF events to Transformation Hub and then configure Transformation Hub to filter Avro events, use filters to ensure that ESM does not receive duplicate events. You might want to use filters to accomplish the following:

- Filter out desired events from Connectors so that ESM does not process them
- Filter out ESM's correlation events that were forwarded (CEF events that the Forwarding Connector sent to `th-cef`) so that ESM does not re-process its own events.

If you do *not* configure filtering, ESM must consume from the `th-arcsight-avro` topic. If you configure filtering, ESM must consume from the `mf-event-avro-esmfiltered` topic. For more information, see configuring filters and local and global event enrichment.

6. Advance through the wizard and complete the configuration.
7. As user `arcsight`, restart the ArcSight Manager:

In compact mode:

```
/etc/init.d/arcsight_services start manager
```

In distributed mode:

```
/etc/init.d/arcsight_services stop all
```

```
/etc/init.d/arcsight_services start all
```

8. Verify that the connection was successful:

```
grep -rnw '/opt/arcsight/var/logs/manager/' -e 'Transformation Hub
service is initialized' -e 'Started kafka readers'
```

The output should be similar to the following:

```
/opt/arcsight/var/logs/manager/default/server.std.log:5036:2021-07-13
09:51:36 =====> Transformation Hub service is initialized (49 s) <=====
/opt/arcsight/var/logs/manager/default/server.log:11664:[2021-07-13
09:51:36,656][INFO ][default.com.arcsight.common.messaging.events.aa]
Started kafka readers in PT0.115S
/opt/arcsight/var/logs/manager/default/server.log:11665:[2021-07-13
09:51:36,657][INFO ][default.com.arcsight.server.NGServer] Transformation
Hub service is initialized (49 s)
```

Setting Up SSL Client-Side Authentication Between Transformation Hub and ESM - FIPS Mode

ArcSight Platform maintains its own certificate authority (CA) to issue certificates for individual nodes in the Transformation Hub cluster and external communication. ESM needs the signed certificates in its truststore so that it will trust connections to the Arcsight Platform and Transformation Hub. You might need to contact the ArcSight Platform administrator to obtain the signed certificates if you do not have sufficient privileges to access them and run the necessary commands.

Note: When configuring Transformation Hub access, you must specify the FQDN of the ArcSight Platform virtual IP for HA or single master node and *not* the IP address.

To complete the configuration, complete the following tasks:

- Enable client-side authentication between Transformation Hub and ESM
- Configure ESM to consume from Transformation Hub.

Enabling Client-side Authentication Between Transformation Hub and ESM:

1. Verify that Transformation Hub is functional and that client authentication is configured.
2. As user `arcsight`, stop the ArcSight Manager:

   ```
   /etc/init.d/arcsight_services stop manager
   ```

3. Generate the keypair and certificate signing request (`.csr`) file. When generating the keypair, specify the fully qualified domain name of the ArcSight Manager host as the common name (CN) for the certificate.

   Run the following commands:

   ```
   export COMMON_NAME=<your ESM host's fully qualified domain name>
   ```

```
/opt/arcsight/manager/bin/arcsight keytool -store clientkeys -genkeypair
-dname "cn=${COMMON_NAME}, ou=<your organization>, o=<your company>,
c=<your country>" -keyalg rsa -keysize 2048 -alias ebkey -startdate -1d -
validity 366
```

```
/opt/arcsight/manager/bin/arcsight keytool -certreq -store clientkeys -
alias ebkey -file  ${COMMON_NAME}.csr
```

where `${COMMON_NAME}.csr` is the output file where the `.csr` is stored.

4. To sign the ESM certificate signing request, perform the following steps in the ArcSight Platform. For an on-premises deployment, perform the steps on the master node. For a cloud deployment, perform the steps on the Bastion host.
   a. Create a temporary folder to store the generated certificates:

      ```
      mkdir –m 700 /tmp/esm
      ```

   b. Move the certificate signing request (`.csr`) file from the ESM host to the temporary folder that you created.
   c. Set the environment variables:

      ```
      export CA_CERT=re_ca.cert.pem
      ```

      ```
      export COMMON_NAME=<your ESM host's fully-qualified domain name>
      ```

      ```
      export TH=<FQDN of the ArcSight Platform virtual IP for HA or single
      master node>_<Kafka TLS-enabled port>
      ```

      Note: For `COMMON_NAME`, use the same host FQDN as you used for the ESM client key pair.

   d. Run the following commands to sign the ESM certificate signing request:

      ```
      cd /tmp/esm
      ```

      ```
      export CDF_APISERVER=$(kubectl get pods -n core -o custom-
      columns=":metadata.name"| grep cdf-apiserver)
      export PASSPHRASE=$(kubectl get secret vault-passphrase -n core -o
      json 2>/dev/null | jq -r '.data.passphrase')
      export ENCRYPTED_ROOT_TOKEN=$(kubectl get secret vault-credential -n
      core -o json 2>/dev/null | jq -r '.data."root.token"')
      export VAULT_TOKEN=$(echo ${ENCRYPTED_ROOT_TOKEN} | openssl aes-256-
      cbc -md sha256 -a -d -pass pass:"${PASSPHRASE}")
      export CSR=$(cat ${COMMON_NAME}.csr)
      ```

```
WRITE_RESPONSE=$(kubectl exec -it -n core ${CDF_APISERVER} -c cdf-
apiserver -- bash -c "VAULT_TOKEN=$VAULT_TOKEN vault write -tls-skip-
verify -format=json RE/sign/coretech csr=\"${CSR}\"") && \
echo "$WRITE_RESPONSE" | jq -r ".data | .certificate" > ${COMMON_
NAME}.signed.crt && \
echo "$WRITE_RESPONSE" | jq -r ".data | .issuing_ca" > ${COMMON_
NAME}.issue_ca.crt && \
echo "$WRITE_RESPONSE" | jq -r ".data | .certificate, if .ca_chain
then .ca_chain[] else .issuing_ca end" > ${COMMON_
NAME}.signed.cert.with.ca.crt
```

The signed certificate is in the file `${COMMON_NAME}.signed.crt`. The issuing CA is in the file `${COMMON_NAME}.issue_ca.crt`. The signed certificate with the CA chain is in the file `${COMMON_NAME}.signed.cert.with.ca.crt`.

5. Retrieve the RE certificates:

   For an on-premises deployment:

   ```
   cd /opt/arcsight/kubernetes/scripts/
   ```

   ```
   ./cdf-updateRE.sh > /tmp/esm/${CA_CERT}
   ```

   For a cloud deployment:

   ```
   cd {path to cdf installer}/cdf-deployer/scripts/
   ```

   ```
   ./cdf-updateRE.sh > /tmp/esm/${CA_CERT}
   ```

6. Copy the following files from the Transformation Hub `/tmp/esm` folder to an ESM host folder (for example, `/opt/arcsight/tmp`):

   `/tmp/esm/${COMMON_NAME}.signed.cert.with.ca.crt`

   `/tmp/esm/${CA_CERT}`

   Remove the files from `/tmp/esm` after you copy them.

7. On the ESM server, import the RE certificate from file ${CA_CERT} into the ESM client truststore:

   ```
   /opt/arcsight/manager/bin/arcsight keytool -store clientcerts -alias
   <alias for the certificate> -importcert -file <absolute path to
   certificate file>
   ```

   For example:

   ```
   /opt/arcsight/manager/bin/arcsight keytool -store clientcerts -alias
   thcert -importcert -file /opt/arcsight/tmp/re_ca.cert.pem
   ```

Note: You might receive the following message:

```
Certificate already exists in keystore under alias <alias1>
Do you still want to add it? [no]:
```

It is not necessary to add an existing certificate.

8. On the ESM server, run the following command to import the signed certificate:

```
/opt/arcsight/manager/bin/arcsight keytool -store clientkeys -alias
<alias for the key> -importcert -file <path to signed cert> -trustcacerts
```

For example:

```
/opt/arcsight/manager/bin/arcsight keytool -store clientkeys -alias ebkey
-importcert -file /opt/arcsight/tmp/${COMMON_NAME}.signed.cert.with.ca.crt
–trustcacerts
```

Note: You might see the following warning:

```
...
Top-level certificate in reply:
...
... is not trusted. Install reply anyway? [no]:
```

This is because the root certificate of the RE CA is not in the ESM truststore. This does not affect the functionality of ESM. Enter **yes** to allow the new certificate to be imported.

To complete the configuration, configure ESM to consume from Transformation Hub.

Configuring Logger as a Transformation Hub Consumer

Note: For Azure or AWS Transformation Hub, you complete configuration procedures as detailed in Completing Integration with Azure Transformation Hub or Completing Integration with AWS Transformation Hub before you perform the procedures in this section.

The procedure for configuring a Logger as a Transformation Hub consumer will depend on which configuration you are using. Follow the configuration that matches your environment.

Configuring Logger as a Transformation Hub Consumer – Client Authentication in FIPS Mode

Follow these steps to configure Logger to consume from Transformation Hub with client authentication in FIPS Mode

Enabling FIPS and Preparing the Logger Server

Follow these steps to enable FIPS mode on Logger and to prepare the Logger server:

1. Sign-in to the Logger Console and enable FIPS mode.

   For more information, see "Enabling and Disabling FIPS Mode on Logger" in the *ArcSight Logger Administrator's Guide*.
2. Navigate to the Logger server's `/tmp/` directory and create a logger directory if it does not already exist.

   ```
   mkdir -p logger
   ```
3. Change to the Logger directory:

   ```
   cd /tmp/logger
   ```
4. Set the environment variables for static values used by `th_cert_tool.sh` script.

   ```
   export TH=<transformation.hub.fqdn>
   ```

Generating and Signing the Certificate Signing Request

Follow these steps to generate and sign the certificate signing request (CSR).

1. Run these commands to generate the CSR and copy it to the TH Logger directory:

   ```
   sudo /opt/arcsight/current/arcsight/logger/bin/scripts/th_cert_tool.sh --
   generate-csr --th-host ${TH} --key-length 2048
   ```

   ```
   mv csr.csr /tmp/logger/
   ```
2. Run the following commands to sign the Logger certificate signing request in Transformation Hub:

   ```
   cd /tmp/logger
   export CA_CERT=re_ca.cert.pem
   export COMMON_NAME=<Logger.fqdn>
   ```

   ```
   export CDF_APISERVER=$(kubectl get pods -n core -o custom-
   columns=":metadata.name"| grep cdf-apiserver)
   ```

```
export PASSPHRASE=$(kubectl get secret vault-passphrase -n core -o json
2>/dev/null | jq -r '.data.passphrase')
```

```
export ENCRYPTED_ROOT_TOKEN=$(kubectl get secret vault-credential -n core
-o json 2>/dev/null | jq -r '.data."root.token"')
```

```
export VAULT_TOKEN=$(echo ${ENCRYPTED_ROOT_TOKEN} | openssl aes-256-cbc -
md sha256 -a -d -pass pass:"${PASSPHRASE}")
```

```
mv csr.csr ${COMMON_NAME}.csr
export CDF_APISERVER=$(kubectl get pods -n core -o custom-
columns=":metadata.name"| grep cdf-apiserver)
```

```
export PASSPHRASE=$(kubectl get secret vault-passphrase -n core -o json
2>/dev/null | jq -r '.data.passphrase')
```

```
export ENCRYPTED_ROOT_TOKEN=$(kubectl get secret vault-credential -n core
-o json 2>/dev/null | jq -r '.data."root.token"')
```

```
export VAULT_TOKEN=$(echo ${ENCRYPTED_ROOT_TOKEN} | openssl aes-256-cbc -
md sha256 -a -d -pass pass:"${PASSPHRASE}")
```

```
export CSR=$(cat ${COMMON_NAME}.csr)
```

```
WRITE_RESPONSE=$(kubectl exec -it -n core ${CDF_APISERVER} -c cdf-
apiserver -- bash -c "VAULT_TOKEN=$VAULT_TOKEN vault write -tls-skip-
verify -format=json RE/sign/coretech csr=\"${CSR}\"") && \
```

```
echo "${WRITE_RESPONSE}" | jq -r ".data | .certificate" > ${COMMON_
NAME}.signed.crt && \
```

```
echo "${WRITE_RESPONSE}" | jq -r ".data | .issuing_ca" > ${COMMON_
NAME}.issue_ca.crt && \
```

```
echo "${WRITE_RESPONSE}" | jq -r ".data | .certificate, if .ca_chain then
.ca_chain[] else .issuing_ca end" > ${COMMON_
NAME}.signed.cert.with.ca.crt
```

3. Retrieve the RE certificate:

```
/<TH Home Path>/scripts/cdf-updateRE.sh > /tmp/logger/${CA_CERT}
```

Example:/opt/arcsight/kubernetes/scripts/cdf-updateRE.sh >/tmp/logger/${CA_CERT}

> **Note:** For a cloud installation (Azure or AWS), log in to the bastion or jump host and run the script `cdf-updateRE.sh`:
>
> arcsight-platform-cloud-installer/cdf-deployer/scripts/cdf-updateRE.sh

4. Move the following files from the Transformation Hub to the Logger `/tmp/logger` directory:
   - `/tmp/logger/${COMMON_NAME}.signed.cert.with.ca.crt`
   - `/tmp/logger/${CA_CERT}`
5. Run the following commands in Logger:

```
export TH=<transformartion.hub.fqdn>
export COMMON_NAME=<Logger.fqdn>
export CA_CERT=re_ca.cert.pem
export ARCSIGHT_HOME=/opt/arcsight/current/arcsight/logger
/opt/arcsight/current/arcsight/logger/bin/scripts/th_cert_tool.sh --
import-cert --th-host ${TH} --cert-path /tmp/logger/${COMMON_
NAME}.signed.cert.with.ca.crt
${ARCSIGHT_HOME}/bin/scripts/keytool_util.sh receiver delete mykey
/opt/arcsight/current/arcsight/logger/bin/scripts/keytool_util.sh
receiver importcert /tmp/logger/${CA_CERT}
```

6. When adding the receiver in the Logger Console, configure the following settings:
   - `Use SSL/TLS = TRUE`
   - `Use Client Authentication = TRUE`

Configuring Logger as a Transformation Hub Consumer – Client Authentication in non-FIPS Mode

Follow these steps to configure Logger to consume from Transformation Hub with client authentication in non-FIPS Mode

- Preparing the Logger Server
- Generating and Signing the Certificate Signing Request

Preparing the Logger Server

Follow these steps to prepare the Logger server:

1. Navigate to the Logger server's `/tmp/` directory and create a logger directory if it does not already exist.

```
mkdir -p logger
```

2. Change to the Logger directory:

```
cd /tmp/logger
```

3. Set the environment variables for static values used by `th_cert_tool.sh` script.

```
export TH=<transformation.hub.fqdn>
```

Generating and Signing the Certificate Signing Request

Follow these steps to generate and sign the certificate signing request (CSR).

1. Run these commands to generate the CSR and copy it to the TH Logger directory:

```
sudo /opt/arcsight/current/arcsight/logger/bin/scripts/th_cert_tool.sh --
generate-csr --th-host ${TH} --key-length 2048
```

```
mv csr.csr /tmp/logger/
```

2. Run the following commands to sign the Logger certificate signing request in Transformation Hub:

```
cd /tmp/logger
export CA_CERT=re_ca.cert.pem
export COMMON_NAME=<logger.fqdn>
```

```
export CDF_APISERVER=$(kubectl get pods -n core -o custom-
columns=":metadata.name"| grep cdf-apiserver)
```

```
export PASSPHRASE=$(kubectl get secret vault-passphrase -n core -o json
2>/dev/null | jq -r '.data.passphrase')
```

```
export ENCRYPTED_ROOT_TOKEN=$(kubectl get secret vault-credential -n core
-o json 2>/dev/null | jq -r '.data."root.token"')
```

```
export VAULT_TOKEN=$(echo ${ENCRYPTED_ROOT_TOKEN} | openssl aes-256-cbc -
md sha256 -a -d -pass pass:"${PASSPHRASE}")
```

```
mv csr.csr ${COMMON_NAME}.csr
export CDF_APISERVER=$(kubectl get pods -n core -o custom-
columns=":metadata.name"| grep cdf-apiserver)
```

```
export PASSPHRASE=$(kubectl get secret vault-passphrase -n core -o json
2>/dev/null | jq -r '.data.passphrase')
```

```
export ENCRYPTED_ROOT_TOKEN=$(kubectl get secret vault-credential -n core
-o json 2>/dev/null | jq -r '.data."root.token"')
```

```
export VAULT_TOKEN=$(echo ${ENCRYPTED_ROOT_TOKEN} | openssl aes-256-cbc -
md sha256 -a -d -pass pass:"${PASSPHRASE}")
```

```
export CSR=$(cat ${COMMON_NAME}.csr)
```

```
WRITE_RESPONSE=$(kubectl exec -it -n core ${CDF_APISERVER} -c cdf-
apiserver -- bash -c "VAULT_TOKEN=$VAULT_TOKEN vault write -tls-skip-
verify -format=json RE/sign/coretech csr=\"${CSR}\"") && \
```

```
echo "${WRITE_RESPONSE}" | jq -r ".data | .certificate" > ${COMMON_
NAME}.signed.crt && \
```

```
echo "${WRITE_RESPONSE}" | jq -r ".data | .issuing_ca" > ${COMMON_
NAME}.issue_ca.crt && \
```

```
echo "${WRITE_RESPONSE}" | jq -r ".data | .certificate, if .ca_chain then
.ca_chain[] else .issuing_ca end" > ${COMMON_
NAME}.signed.cert.with.ca.crt
```

3. Retrieve the RE certificate:

   ```
   /<TH Home Path>/scripts/cdf-updateRE.sh > /tmp/logger/${CA_CERT}
   ```

   Example:`/opt/arcsight/kubernetes/scripts/cdf-updateRE.sh >/tmp/logger/${CA_CERT}`

   Note: For a cloud installation (Azure or AWS), log in to the bastion or jump host and run the script `cdf-updateRE.sh`:

   arcsight-platform-cloud-installer/cdf-deployer/scripts/cdf-updateRE.sh

4. Move the following files from the Transformation Hub to the Logger `/tmp/logger` directory:
   - `/tmp/logger/${COMMON_NAME}.signed.cert.with.ca.crt`
   - `/tmp/logger/${CA_CERT}`
5. Run the following commands in Logger:

```
export TH=<transformartion.hub.fqdn>
export COMMON_NAME=<Logger.fqdn>
export CA_CERT=re_ca.cert.pem
export ARCSIGHT_HOME=/opt/arcsight/current/arcsight/logger
/opt/arcsight/current/arcsight/logger/bin/scripts/th_cert_tool.sh --
import-cert --th-host ${TH} --cert-path /tmp/logger/${COMMON_
NAME}.signed.cert.with.ca.crt
${ARCSIGHT_HOME}/bin/scripts/keytool_util.sh receiver delete mykey
/opt/arcsight/current/arcsight/logger/bin/scripts/keytool_util.sh
receiver importcert /tmp/logger/${CA_CERT}
```

6. When adding the receiver in the Logger Console, configure the following settings:
   - `Use SSL/TLS = TRUE`
   - `Use Client Authentication = TRUE`

Configuring Logger as a Transformation Hub Consumer – No Client Authentication in FIPS Mode

Follow these steps to configure Logger to consume from Transformation Hub without client authentication in FIPS Mode.

Enabling FIPS and Preparing the Logger Server

Follow these steps to enable FIPS mode on Logger and to prepare the Logger OBC:

1. Sign-in to the Logger Console and enable FIPS mode.

   For more information, see "Enabling and Disabling FIPS Mode on Logger" in the *ArcSight Logger Administrator's Guide*.

2. Run the following commands in Logger:

```
<install dir>/logger/bin/scripts/th_cert_tool.sh --generate-csr --th-host
<MASTER_IP> --key-length 2048
```

```
scp <install dir>/logger/user/logger/th_certs/<MASTER_IP>/csr.csr <MASTER_
IP>:/tmp/
```

Signing the Certificate in Transformation Hub

Run this command to sign the certificate in Transformation Hub:

```
openssl x509 -req -CA ca.crt -CAkey ca.key -in /tmp/csr.csr -out
/tmp/signedLoggerCert.pem -days 3650 -CAcreateserial -passin pass:arcsight -
sha256
```

Importing the Certificate into Logger

Run these commands to import the certificate into Logger:

```
scp masterhost:/tmp/signedLoggerCert.pem /tmp
```

```
<install dir>/logger/bin/scripts/th_cert_tool.sh --import-cert --th-host
<MASTER_IP> --cert-path /tmp/signedLoggerCert.pem
```

Retrieving the Certificate on Transformation Hub

Follow these instructions to retrieve the certificate on Transformation Hub for an on-premises or cloud deployment.

- On-premises deployment. Run the following command:

  ```
  /<TH Home Path>/scripts/cdf-updateRE.sh > /tmp/logger/RE.crt
  ```

  Example: `/opt/arcsight/kubernetes/scripts/cdf-updateRE.sh >/tmp/logger/RE.crt`
- Cloud deployment (Azure or AWS). Follow these steps:
  a. Log in to the bastion or jump host and run the script `cdf-updateRE.sh`:

     ```
     arcsight-platform-cloud-installer/cdf-deployer/scripts/cdf-updateRE.sh
     ```

  b. Run the following commands in Logger:

     ```
     scp <MASTER_IP>:/tmp/RE.crt /tmp/export ARCSIGHT_HOME=<install
     dir>/logger/$ARCSIGHT_HOME/bin/scripts/keytool_util.sh receiver delete
     mykey
     ```

     ```
     $ARCSIGHT_HOME/bin/scripts/keytool_util.sh receiver importcert
     /tmp/RE.crt
     ```

     ```
     <install dir>/logger/bin/loggerd restart receivers
     ```

     ```
     watch -n 1 '/opt/current/arcsight/logger/bin/loggerd status'
     ```

c. When adding the receiver in the Logger Console, configure the following setting:

```
Use SSL/TLS = TRUE
```

Configuring Logger as a Transformation Hub Consumer – No Client Authentication in non-FIPS Mode with TLS

Configuring Logger to consume from Transformation Hub without client authentication in non-FIPS Mode using TLS only requires retrieving the RE certificate on the Transformation Hub, and the steps to follow depend on your type of deployment:

- On-premises deployment:

```
/<TH Home Path>/scripts/cdf-updateRE.sh > /tmp/logger/RE.crt
```

Example: `/opt/arcsight/kubernetes/scripts/cdf-updateRE.sh >/tmp/logger/RE.crt`

- Cloud deployment (Azure or AWS):
  a. Log in to the bastion or jump host and run the script `cdf-updateRE.sh`:

```
arcsight-platform-cloud-installer/cdf-deployer/scripts/cdf-updateRE.sh
```

  b. Run the following commands in Logger:

```
scp <MASTER_IP>:/tmp/RE.crt /tmp/export ARCSIGHT_HOME=<install
dir>/logger/$ARCSIGHT_HOME/bin/scripts/keytool_util.sh receiver delete
mykey
```

```
$ARCSIGHT_HOME/bin/scripts/keytool_util.sh receiver importcert
/tmp/RE.crt
```

```
<install dir>/logger/bin/loggerd restart receivers
```

```
watch -n 1 '/opt/current/arcsight/logger/bin/loggerd status'
```

  c. When adding the receiver in the Logger Console, configure the following setting:

```
Use SSL/TLS = TRUE
```

Configuring Logger as a Transformation Hub Consumer – No Client Authentication in non-FIPS Mode without TLS

Follow these steps to configure Logger to consume from Transformation Hub without client authentication in non-FIPS Mode and without TLS.

1. Sign in to the Logger Console and create a Logger TH receiver.

   For more information, see "Working with Receivers" in the *ArcSight Logger Administrator's Guide*.

2. Configure the TH receiver with the following values:
   - Transformation Hub host(s) and port = IP address or host name followed by semicolon and port 9092.

     For example: *your.th.ip.address*:9092
   - Use SSL/TLS Key = FALSE
   - Use SSL/TLS Client Authentication = FALSE
3. Fill in other fields with applicable values, and save the changes.

Configuring Logger as a Transformation Hub Producer

You can configure a Logger with a Transformation Hub destination with the encrypted security mode that you require.

Configuring Logger with a Transformation Hub Destination – Client Authentication in FIPS Mode

Follow these steps to configure a Logger TH destination with client authentication in FIPS mode.

- Enabling FIPS and Preparing the Logger OBC
- Creating a Keystore on the Logger Onboard Connector
- Signing the Logger OBC Certificate Signing Request on Transformation Hub
- Updating the Keystore and Creating a Truststore on the Logger OBC
- Creating a Logger TH Destination in the Console

Enabling FIPS and Preparing the Logger OBC

Follow these steps to enable FIPS mode on Logger and to prepare the Logger OBC:

1. Sign-in to the Logger Console and enable FIPS mode.

   For more information, see "Enabling and Disabling FIPS Mode on Logger" in the *ArcSight Logger Administrator's Guide*.
2. Navigate to the Logger OBC's `current` directory:

```
cd <install dir>/connector/current
```

3. Set the environment variables for the static values used by `keytool`:

```
export CURRENT=<full path to this "current" folder>
```

```
export BC_OPTS="-storetype BCFKS -providername BCFIPS -J-
Djava.security.egd=file:/dev/urandom -providerpath
${CURRENT}/lib/agent/fips/bc-fips-1.0.2.jar -providerclass
org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider"
```

```
export TH=<Transformation Hub hostname>_<Transformation Hub port>
export STORES=${CURRENT}/user/agent/stores
export STORE_PASSWD=changeit
export CA_CERT=re_ca.cert.pem
export COMMON_NAME=<LoggerFQDN>
```

4. Create the `${CURRENT}/user/agent/stores` directory if it does not already exist.

```
mkdir -p ${STORES}
```

5. Apply the following workaround for a Java `keytool` issue:
   a. Create a new file, `agent.security`:

```
${CURRENT}/user/agent
```

   b. Add the following content to the newly created file, and then save it:

```
security.provider.1=org.bouncycastle.jcajce.provider.BouncyCastleFipsP
rovider
security.provider.2=com.sun.net.ssl.internal.ssl.Provider BCFIPS
security.provider.3=sun.security.provider.Sun
```

   c. Move the `${CURRENT}/lib/agent/fips/bcprov-jdk15on-168.jar` file to the `current` directory.

Creating a Keystore on the Logger Onboard Connector

Follow these steps to create the OBC keystore:

1. Create the keystore for the OBC:

```
cd ${STORES}
```

```
$CURRENT/jre/bin/keytool -keystore ${TH}.keystore.bcfips -genkeypair -
dname "cn=logger.fqdn, ou=ArcSight, o=Micro Focus, c=US" -keyalg rsa -
```

```
keysize 2048 -alias ${TH} -startdate -1d -validity 365 -storepass
${STORE_PASSWD} -keypass ${STORE_PASSWD} ${BC_OPTS}
```

2. Create the certificate signing request (CSR) for the Logger OBC:

```
$CURRENT/jre/bin/keytool -certreq -alias ${TH} -keystore
${TH}.keystore.bcfips -file ${COMMON_NAME}.csr -storepass ${STORE_PASSWD}
${BC_OPTS}
```

3. Copy the CSR file `${COMMON_NAME}.csr` to the Transformation Hub `/tmp` folder:

```
cp ${COMMON_NAME}.csr /tmp/
```

Signing the Logger OBC Certificate Signing Request on Transformation Hub

1. Set the environment:

```
export CA_CERT=re_ca.cert.pem
export COMMON_NAME=<LoggerFQDN>
export TH=<Transformation Hub hostname>_<Transformation Hub port>
```

Note: Use the same values that you specified for the Logger OBC.

2. Run these commands to sign the Logger certificate signing request:

```
mkdir /tmp/logger
mv ${COMMON_NAME}.csr /tmp/logger/
cd /tmp/logger
```

```
export CDF_APISERVER=$(kubectl get pods -n core -o custom-
columns=":metadata.name"| grep cdf-apiserver)
```

```
export PASSPHRASE=$(kubectl get secret vault-passphrase -n core -o json
2>/dev/null | jq -r '.data.passphrase')
```

```
export ENCRYPTED_ROOT_TOKEN=$(kubectl get secret vault-credential -n core
-o json 2>/dev/null | jq -r '.data."root.token"')
```

```
export VAULT_TOKEN=$(echo ${ENCRYPTED_ROOT_TOKEN} | openssl aes-256-cbc -
md sha256 -a -d -pass pass:"${PASSPHRASE}")
```

```
export CSR=$(cat ${COMMON_NAME}.csr)
```

```
WRITE_RESPONSE=$(kubectl exec -it -n core ${CDF_APISERVER} -c cdf-
apiserver -- bash -c "VAULT_TOKEN=$VAULT_TOKEN vault write -tls-skip-
verify -format=json RE/sign/coretech csr=\"${CSR}\"") && \
```

```
echo "${WRITE_RESPONSE}" | jq -r ".data | .certificate" > ${COMMON_
NAME}.signed.crt && \
```

```
echo "${WRITE_RESPONSE}" | jq -r ".data | .issuing_ca" > ${COMMON_
NAME}.issue_ca.crt && \
```

```
echo "${WRITE_RESPONSE}" | jq -r ".data | .certificate, if .ca_chain then
.ca_chain[] else .issuing_ca end" > ${COMMON_
NAME}.signed.cert.with.ca.crt
```

The signed certificate is in file `${COMMON_NAME}.signed.crt`.

The issuing CA is in file `${COMMON_NAME}.issue_ca.crt`.

The signed certificate with CA chain is in file `${COMMON_NAME}.signed.cert.with.ca.crt`.

3. Retrieve the RE certificate.

```
/<TH Home Path>/scripts/cdf-updateRE.sh > /tmp/logger/${CA_CERT}
```

Example: `/opt/arcsight/kubernetes/scripts/cdf-updateRE.sh >/tmp/logger/${CA_CERT}`

> **Note:** For a cloud installation (Azure or AWS), log in to the bastion or jump host and run the script `cdf-updateRE.sh`:
>
> arcsight-platform-cloud-installer/cdf-deployer/scripts/cdf-updateRE.sh

Move the following files from the Transformation Hub to the Logger OBC `STORES` directory:

- `/tmp/logger/ ${COMMON_NAME}.signed.crt`
- `/tmp/logger/${COMMON_NAME}.issue_ca.crt`
- `/tmp/logger/${COMMON_NAME}.signed.cert.with.ca.crt`
- `/tmp/logger/${CA_CERT}`

Updating the Keystore and Creating a Truststore on the Logger OBC

Follow these steps to update the keystore and to create a truststore on the Logger OBC:

1. Update the Logger OBC keystore with a signed certificate.

```
cd ${STORES}/
```

```
$CURRENT/jre/bin/keytool -importcert -alias ${TH} -keystore
${TH}.keystore.bcfips -trustcacerts -file ${COMMON_
NAME}.signed.cert.with.ca.crt -storepass ${STORE_PASSWD} ${BC_OPTS}
```

Verification: Run the following command to verify the keystore, and ensure that it has only one entry in the keystore.

```
$CURRENT/jre/bin/keytool -v -list -keystore ${TH}.keystore.bcfips -
storepass ${STORE_PASSWD} ${BC_OPTS} |grep -i alias
```

2. Create the Logger OBC truststore.

```
cd ${STORES}/
```

```
$CURRENT/jre/bin/keytool -importcert -alias reca -trustcacerts -file
${CA_CERT} -keystore ${TH}.truststore.bcfips -storepass ${STORE_PASSWD}
${BC_OPTS}
```

When prompted, specify **yes** to trust the certificate.

Creating a Logger TH Destination in the Console

Follow these steps to create a Logger TH destination in the Console:

1. Run the following commands, and note the keystore and truststore paths:

```
echo ${STORES}/${TH}.keystore.bcfips
echo ${STORES}/${TH}.truststore.bcfips
```

2. Sign-in to the Logger Console to create a TH destination.

   For more information, see "Transformation Hub Destinations" in the *ArcSight Logger Administrator's Guide*.

3. Create the TH destination with the following values:
   - Kafka Broker on SSL/TLS = TRUE
   - SSL/TLS Truststore File Path = <*truststorePath*>
   - SSL/TLS Truststore Password = STORE_PASSWD
   - Use SSL/TLS Client Authentication = TRUE
   - SSL/TLS Keystore File Path = <*keystoreFilePath*>
   - SSL/TLS Keystore Password = STORE_PASSWD
   - SSL/TLS Key Password = STORE_PASSWD
4. Fill in other fields with applicable values, and save the changes.

Configuring Logger with a Transformation Hub Destination – Client Authentication in non-FIPS Mode

Follow these steps to configure a Logger with a Transformation Hub (TH) destination with client authentication, in non-FIPS mode.

Preparing the Logger Onboard Connector

1. Navigate to the Logger OBC's current directory:

```
cd <install dir>/connector/current
```

2. Set the environment variables for the static values used by `keytool`:

```
export CURRENT=<full path to this "current" folder>
export TH=<Transformation Hub hostname>_<Transformation Hub port>
export STORES=${CURRENT}/user/agent/stores
export STORE_PASSWD=changeit
export CA_CERT=re_ca.cert.pem
export COMMON_NAME=<LoggerFQDN>
```

3. Create the `${CURRENT}/user/agent/stores` directory if it does not already exist.

```
mkdir -p ${STORES}
```

Creating the Keystore for Logger's Onboard Connector

Follow these steps to create the OBC keystore:

1. Create the keystore for the OBC:

```
cd ${STORES}
```

```
$CURRENT/jre/bin/keytool -keystore ${TH}.keystore.bcfips -genkeypair -
dname "cn=logger.fqdn, ou=yourOU, o=yourCompany, c=US" -keyalg rsa -
keysize 2048 -alias ${TH} -startdate -1d -validity 365 -storepass
${STORE_PASSWD} -keypass ${STORE_PASSWD}
```

2. Create the certificate signing request (CSR) for the Logger OBC:

```
$CURRENT/jre/bin/keytool -certreq -alias ${TH} -keystore
${TH}.keystore.jks -file ${COMMON_NAME}.csr -storepass ${STORE_PASSWD}
```

3. Copy the CSR file `${COMMON_NAME}.csr` to the Transformation Hub `/tmp` folder:

```
cp ${COMMON_NAME}.csr /tmp/
```

Signing the Logger OBC Certificate Signing Request on Transformation Hub

1. Set the environment:

```
export CA_CERT=re_ca.cert.pem
export COMMON_NAME=<LoggerFQDN>
export TH=<Transformation Hub hostname>_<Transformation Hub port>
```

Note: Use the same values that you specified for the Logger OBC.

2. Run these commands to sign the Logger certificate signing request:

```
mkdir /tmp/logger
mv ${COMMON_NAME}.csr /tmp/logger/
cd /tmp/logger
```

```
export CDF_APISERVER=$(kubectl get pods -n core -o custom-
columns=":metadata.name"| grep cdf-apiserver)
```

```
export PASSPHRASE=$(kubectl get secret vault-passphrase -n core -o json
2>/dev/null | jq -r '.data.passphrase')
```

```
export ENCRYPTED_ROOT_TOKEN=$(kubectl get secret vault-credential -n core
-o json 2>/dev/null | jq -r '.data."root.token"')
```

```
export VAULT_TOKEN=$(echo ${ENCRYPTED_ROOT_TOKEN} | openssl aes-256-cbc -
md sha256 -a -d -pass pass:"${PASSPHRASE}")
```

```
export CSR=$(cat ${COMMON_NAME}.csr)
```

```
WRITE_RESPONSE=$(kubectl exec -it -n core ${CDF_APISERVER} -c cdf-
apiserver -- bash -c "VAULT_TOKEN=$VAULT_TOKEN vault write -tls-skip-
verify -format=json RE/sign/coretech csr=\"${CSR}\"") && \
```

```
echo "${WRITE_RESPONSE}" | jq -r ".data | .certificate" > ${COMMON_
NAME}.signed.crt && \
```

```
echo "${WRITE_RESPONSE}" | jq -r ".data | .issuing_ca" > ${COMMON_
NAME}.issue_ca.crt && \
```

```
echo "${WRITE_RESPONSE}" | jq -r ".data | .certificate, if .ca_chain then
.ca_chain[] else .issuing_ca end" > ${COMMON_
NAME}.signed.cert.with.ca.crt
```

The signed certificate is in file `${COMMON_NAME}.signed.crt`.

The issuing CA is in file `${COMMON_NAME}.issue_ca.crt`.

The signed certificate with CA chain is in file `${COMMON_NAME}.signed.cert.with.ca.crt`.

3. Retrieve the RE certificate.

```
/<TH Home Path>/scripts/cdf-updateRE.sh > /tmp/logger/${CA_CERT}
```

Example: `/opt/arcsight/kubernetes/scripts/cdf-updateRE.sh >/tmp/logger/${CA_CERT}`

Note: For a cloud installation (Azure or AWS), log in to the bastion or jump host and run the script `cdf-updateRE.sh`:

arcsight-platform-cloud-installer/cdf-deployer/scripts/cdf-updateRE.sh

Move the following files from the Transformation Hub to the Logger OBC `STORES` directory:

- `/tmp/logger/ ${COMMON_NAME}.signed.crt`
- `/tmp/logger/${COMMON_NAME}.issue_ca.crt`
- `/tmp/logger/${COMMON_NAME}.signed.cert.with.ca.crt`
- `/tmp/logger/${CA_CERT}`

Updating the Keystore and Creating a Truststore on the Logger OBC

Follow these steps to update the keystore and to create a truststore on the Logger OBC:

1. Update the Logger OBC keystore with a signed certificate.

```
cd ${STORES}/
```

```
$CURRENT/jre/bin/keytool -importcert -alias ${TH} -keystore
${TH}.keystore.jks -trustcacerts -file ${COMMON_
NAME}.signed.cert.with.ca.crt -storepass ${STORE_PASSWD}
```

When prompted, specify **yes** to trust the certificate.

Verification: Run the following command to verify the keystore, and ensure that it has only one entry in the keystore.

```
$CURRENT/jre/bin/keytool -v -list -keystore ${TH}.keystore.jks -storepass
${STORE_PASSWD} |grep -i alias
```

2. Create the Logger OBC truststore.

```
cd ${STORES}/
```

```
$CURRENT/jre/bin/keytool -importcert -alias CARoot -trustcacerts -file
${CA_CERT} -keystore ${TH}.truststore.jks -storepass ${STORE_PASSWD}
```

When prompted, specify **yes** to trust the certificate.

3. Run the following commands, and take note of the truststore path:

```
echo ${STORES}/${TH}.keystore.jks
echo ${STORES}/${TH}.truststore.jks
```

Creating a Logger TH Destination in the Console

Follow these steps to create a Logger TH destination in the Console:

1. Run the following commands, and note the keystore and truststore paths:

```
echo ${STORES}/${TH}.keystore.bcfips
echo ${STORES}/${TH}.truststore.bcfips
```

2. Sign-in to the Logger Console to create a TH destination.

   For more information, see "Transformation Hub Destinations" in the *ArcSight Logger Administrator's Guide*.

3. Create the TH destination with the following values:
   - Kafka Broker on SSL/TLS = TRUE
   - SSL/TLS Truststore File Path = <*truststorePath*>
   - SSL/TLS Truststore Password = <*changeit*>
   - Use SSL/TLS Client Authentication = TRUE

- SSL/TLS Keystore File Path = <*keystoreFilePath*>
- SSL/TLS Keystore Password = <*changeit*>
- SSL/TLS Key Password = <*changeit*>

4. Fill in other fields with applicable values, and save the changes.

Configuring Logger with a Transformation Hub Destination – No Client Authentication in FIPS Mode

Follow these steps to configure a Logger Transformation Hub destination without client authentication in FIPS mode.

- Enabling FIPS and Preparing the Logger OBC
- Creating a CA CERT File on Transformation Hub
- Creating a Truststore on the Logger OBC
- Creating a Logger TH Destination in the Console
- Cleaning up the Certificate File

Enabling FIPS and Preparing the Logger OBC

Follow these steps to enable FIPS mode on Logger and to prepare the Logger OBC:

1. Sign-in to the Logger Console and enable FIPS mode.

   For more information, see "Enabling and Disabling FIPS Mode on Logger" in the *ArcSight Logger Administrator's Guide*.

2. Navigate to the Logger OBC's `current` directory:

   ```
   cd <install dir>/connector/current
   ```

3. Set the environment variables for the static values used by `keytool`:

   ```
   export CURRENT=<full path to this "current" folder>
   ```

   ```
   export BC_OPTS="-storetype BCFKS -providername BCFIPS -J-
   Djava.security.egd=file:/dev/urandom -providerpath
   ${CURRENT}/lib/agent/fips/bc-fips-1.0.2.jar -providerclass
   org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider"
   ```

   ```
   export TH=<Transformation Hub hostname>_<Transformation Hub port>
   export STORES=${CURRENT}/user/agent/stores
   export STORE_PASSWD=changeit
   export CA_CERT=re_ca.cert.pem
   ```

4. Create the `${CURRENT}/user/agent/stores` directory if it does not already exist.

```
mkdir -p ${STORES}
```

5. Apply the following workaround for a Java `keytool` issue:
   a. Create a new file, `agent.security`:

```
${CURRENT}/user/agent
```

   b. Add the following content to the newly created file, and then save it:

```
security.provider.1=org.bouncycastle.jcajce.provider.BouncyCastleFipsP
rovider
security.provider.2=com.sun.net.ssl.internal.ssl.Provider BCFIPS
security.provider.3=sun.security.provider.Sun
```

   c. Move the `${CURRENT}/lib/agent/fips/bcprov-jdk15on-168.jar` file to the `current` directory.

Creating a CA CERT File on Transformation Hub

Follow these steps to create a `${CA_CERT}` file with the content of the root CA certificate:

1. Set the environment:

```
export CA_CERT=re_ca.cert.pem
```

2. Create a certificate:

```
/<TH Home Path>/scripts/cdf-updateRE.sh > /tmp/${CA_CERT}
```

Example: `/opt/arcsight/kubernetes/scripts/cdf-updateRE.sh > /tmp/${CA_CERT}`

> **Note:** For a cloud installation (Azure or AWS), log in to the bastion or jump host and run the script `cdf-updateRE.sh`:
>
> arcsight-platform-cloud-installer/cdf-deployer/scripts/cdf-updateRE.sh

3. Move this file from Transformation Hub to the connector STORES directory.

Creating a Truststore on the Logger OBC

Run this command to create a truststore on the Logger OBC:

```
$CURRENT/jre/bin/keytool -noprompt -importcert -storepass ${STORE_PASSWD} -
destkeystore ${STORES}/${TH}.truststore.bcfips -alias reca -file ${CA_CERT} -
```

```
storetype BCFKS -providerclass
org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider -providerpath
${CURRENT}/lib/agent/fips/bc-fips-1.0.2.jar
```

When prompted, specify **yes** to trust the certificate.

Creating a Logger TH Destination in the Console

1. Run the following command, and note the truststore path:

   ```
   echo ${STORES}/${TH}.keystore.bcfips
   ```

2. Sign-in to the Logger Console to create a TH destination.

   For more information, see "Transformation Hub Destinations" in the *ArcSight Logger Administrator's Guide*.

3. Create the TH destination with the following values:
   - Kafka Broker on SSL/TLS = TRUE
   - SSL/TLS Truststore File Path = <*truststorePath*>
   - SSL/TLS Truststore Password = <*STORE_PASSWORD*>
   - Use SSL/TLS Client Authentication = FALSE
   - SSL/TLS Keystore File Path = <*keystoreFilePath*>
   - SSL/TLS Keystore Password = <*keystorePassword*>
   - SSL/TLS Key Password = <*keyPassword*>

4. Fill in other fields with applicable values, and save the changes.

Cleaning up the Certificate File

Run the following command to delete the certificate file:

Caution: The following file should be deleted to prevent the distribution of security certificates that could be used to authenticate against the Transformation Hub. These files are very sensitive and must not be distributed to other machines.

```
rm ${STORES}/${CA_CERT}
```

Configuring Logger with a Transformation Hub Destination – No Client Authentication in non-FIPS Mode

Follow these steps to configure Logger with a Transformation Hub destination without client authentication in non-FIPS mode. This is the default security mode configuration when installing Transformation Hub.

Preparing the Logger Onboard Connector

Follow these steps to prepare the Logger Onboard Connector.

1. Navigate to the Logger Onboard Connector's (OBC) current directory:

   ```
   cd <install dir>/connector/current
   ```

2. Run the following commands to set the environment variables for the static values used by keytool:

   ```
   export CURRENT=<full path to this "current" folder>
   ```

   Example: `export CURRENT=/opt/arcsight/connector/current`

   ```
   export TH=<Transformation Hub hostname>_<Transformation Hub port>
   export CA_CERT=re_ca.cert.pem
   export STORE_PASSWD=changeit
   export STORES=${CURRENT}/user/agent/stores
   ```

3. Create the directory if it does not already exist.

   ```
   mkdir -p ${STORES}
   ```

Creating a CA CERT File on Transformation Hub

Follow these steps to create a `${CA_CERT}` file with the content of the root CA certificate:

1. Set the environment:

   ```
   export CA_CERT=re_ca.cert.pem
   ```

2. Create a certificate:

```
/<TH Home Path>/scripts/cdf-updateRE.sh > /tmp/${CA_CERT}
```

Example: `/opt/arcsight/kubernetes/scripts/cdf-updateRE.sh > /tmp/${CA_CERT}`

Note: For a cloud installation (Azure or AWS), log in to the bastion or jump host and run the script `cdf-updateRE.sh`:

arcsight-platform-cloud-installer/cdf-deployer/scripts/cdf-updateRE.sh

3. Move this file from Transformation Hub to the connector STORES directory.

Importing the CA Certificate on the Logger OBC

1. Import the CA certificate to the trust store in the `${CURRENT}` folder.

```
$CURRENT/jre/bin/keytool -importcert -file ${STORES}/${CA_CERT} -alias
CARoot -keystore ${STORES}/${TH}.truststore.jks -storepass ${STORE_
PASSWD}
```

2. When prompted, specify **yes** to trust the certificate.
3. Run the following command and note the trust store path:

```
echo ${STORES}/${TH}.truststore.jks
```

4. Run the following command to delete the certificate file:

The following file should be deleted to prevent the distribution of security certificates that could be used to authenticate against the Transformation Hub. These files are very sensitive and should not be distributed to other machines.

```
rm ${STORES}/${CA_CERT}
```

Creating a Logger Transformation Hub Destination in the Console

You need to sign-in to the Logger Console to create a TH destination. For more information, see "Transfomation Hub Destinations" in the *ArcSight Logger Administrator's Guide*.

After logging into the console, create the TH destination with the following values:

- Kafka Broker on SSL/TLS = TRUE
- SSL/TLS Truststore File Path = <*truststorePath*>
- SSL/TLS Truststore Password = STORE_PASSWD

- Use SSL/TLS Client Authentication = FALSE
- SSL/TLS Keystore File Path = <*keystoreFilePath*>
- SSL/TLS Keystore Password = <*keystorePassword*>
- SSL/TLS Key Password = <*keyPassword*>

Fill in other fields with applicable values, and save the changes.

Configuring SmartConnector as a Transformation Hub Producer

Follow the steps in this section to configure a SmartConnector with a Transformation Hub destination with the encrypted security mode that you require. These procedures are provided with the following assumptions:

- You use the default password. To set a non-default password, see Password Management in the SmartConnector User Guide.
- You are on the Linux platform. For Windows platforms, use backslashes (\) when entering commands instead of the forward slashes given here.
- You are using a command window to specify Windows commands. Do not use Windows PowerShell.

In these instructions, `<th hostname>` refers to the FQDN hostname used to access the master node (for on-premises or Azure installations) or, for AWS, the FQDN of the application load balancer; and `<th_port>` refers to either port 9092 (plain text) or 9093 (TLS)

Configuring a SmartConnector with a Transformation Hub Destination without Client Authentication in non-FIPS Mode

Follow these steps to configure a SmartConnector with a Transformation Hub destination without client authentication in non-FIPS mode. This is the default security mode configuration when installing Transformation Hub.

This procedure enables SSL/TLS on the connector.

- "Preparing the SmartConnector Server" on the next page
- "Creating a CA CERT File on Transformation Hub" on page 451
- "Importing the CA Certificate on the Connector" on page 452

Preparing the SmartConnector Server

1. Prepare the SmartConnector:
   - **If the connector is not yet installed:**
     a. Run the installer.
     b. After the core software is installed, do the following in the window that opens: select **Select Global Parameters** > **Set FIPS Mode**, and set the FIPS Mode to **Disabled**.
   - **If the connector is already installed:**
     a. Run the installer.
     b. Select **Set Global Parameters** > **Set FIPS Mode**, and set the FIPS Mode to **Disabled**.
2. Navigate to the `current` directory of the Connector:

   Linux command:

   ```
   cd <install dir>/current
   ```

   Windows command:

   ```
   cd <install dir>\current
   ```

3. Run the following commands to set the environment variables for the static values used by keytool:

   Linux commands:

   ```
   export CURRENT=<full path to this "current" folder>
   ```

   Example: `export CURRENT=/opt/CONNECTORS/TA003/current`

   ```
   export TH=<Transformation Hub hostname>_<Transformation Hub port>
   ```

   Example: `export TH=15.214.***.**`

   ```
   export CA_CERT=re_ca.cert.pem
   export STORE_PASSWD=changeit
   export STORES=${CURRENT}/user/agent/stores
   ```

Windows commands:

```
set CURRENT=<full path to this "current" folder>
set TH=<Transformation Hub hostname>_<Transformation Hub port>
set STORES=%CURRENT%\user\agent\stores
set STORE_PASSWD=changeit
set CA_CERT=re_ca.cert.pem
```

4. Create the directory if it does not already exist.

Linux command:

```
mkdir -p ${STORES}
```

Windows commands:

```
mkdir -p %STORES%
```

Creating a CA CERT File on Transformation Hub

Follow these steps to create a `${CA_CERT}` file with the content of the root CA certificate:

1. Set the environment:

```
export CA_CERT=re_ca.cert.pem
```

2. Create a certificate:

```
/<TH Home Path>/scripts/cdf-updateRE.sh > /tmp/${CA_CERT}
```

Example: `/opt/arcsight/kubernetes/scripts/cdf-updateRE.sh > /tmp/${CA_CERT}`

> **Note:** For a cloud installation (Azure or AWS), log in to the bastion or jump host and run the script `cdf-updateRE.sh`:
>
> arcsight-platform-cloud-installer/cdf-deployer/scripts/cdf-updateRE.sh

3. Move this file from the Transformation Hub to the connector STORES directory.

Importing the CA Certificate on the Connector

1. Import the CA certificate to the trust store in the ${CURRENT} folder.

   Linux command:

   ```
   $CURRENT/jre/bin/keytool -importcert -file ${STORES}/${CA_CERT} -alias
   CARoot -keystore ${STORES}/${TH}.truststore.jks -storepass $STORE_PASSWD
   ```

   $STORE_PASSWD

   Windows command:

   ```
   %CURRENT%\jre\bin\keytool -importcert -file %STORES%\%CA_CERT% -alias
   CARoot -keystore %STORES%\%TH%.truststore.jks -storepass $STORE_PASSWD
   ```

2. When prompted, specify **yes** to trust the certificate.
3. Run the following command and note the trust store path:

   Linux command:

   ```
   echo ${STORES}/${TH}.truststore.jks
   ```

   Windows command:

   ```
   echo %STORES%\%TH%.truststore.jks
   ```

4. Navigate to the bin directory and run the agent setup script to install a connector with Transformation Hub as the destination.

   Linux commands:

   ```
   cd <installation dir>/current/bin
   ```

   ```
   ./runagentsetup.sh
   ```

   Windows commands:

   ```
   cd <installation dir>\current\bin
   ```

   ```
   runagentsetup.bat
   ```

5. Set **Use SSL/TLS** to **true**.

6. Set **Use SSL/TLS Authentication** to **false**.
7. When completing the Transformation Hub destination fields, use the value noted from **step 3** for the trust store path and the password used earlier (For example: `STORE_PASSWD=changeit`) for the trust store password.
8. Run the following command to delete the certificate file:

   Caution: The following file must be deleted to prevent the distribution of security certificates that might be used to authenticate against the Transformation Hub. These files are very sensitive and must not be distributed to other machines.

   Linux command:

   ```
   rm ${STORES}/${CA_CERT}
   ```

   Windows command:

   ```
   del %\STORES%\%CA_CERT%
   ```

Configuring a SmartConnector with a Transformation Hub Destination with Client Authentication in FIPS Mode

Follow these steps to configure a SmartConnector with a Transformation Hub (TH) destination with client authentication in FIPS mode.

You will need to supply an intermediate certificate and key.

- Preparing the SmartConnector Server
- Creating Keystore for SmartConnector on the SmartConnector Server
- Signing a SmartConnector Certificate Signing Request on Transformation Hub
- Updating Keystore and Creating Truststore on the SmartConnector Server
- Running the SmartConnector Setup

Preparing the SmartConnector Server

1. Prepare the SmartConnector:
   - **If the connector is not yet installed:**
     a. Run the installer.
     b. After the core software is installed, do the following in the window that opens: select **Select Global Parameters** > **Set FIPS Mode**, and set the FIPS Mode to **Disabled**.
   - **If the connector is already installed:**
     a. Run the installer.
     b. Select **Set Global Parameters** > **Set FIPS Mode**, and set the FIPS Mode to **Enabled**.
2. Navigate to the Connector's `current` directory:

   Linux command:

   ```
   cd <install dir>/current
   ```

   Windows command:

   ```
   cd <install dir>\current
   ```

3. Set the environment variables for the static values used by `keytool`:

   Linux commands:

   ```
   export CURRENT=<full path to current folder>
   ```

   Example: `export CURRENT=/opt/CONNECTORS/TA003/current`

   ```
   export BC_OPTS="-storetype BCFKS -providername BCFIPS -J-
   Djava.security.egd=file:/dev/urandom -providerpath
   ${CURRENT}/lib/agent/fips/bc-fips-1.0.2.jar -providerclass
   org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider"
   ```

   ```
   export TH=<Transformation Hub hostname>_<Transformation Hub port>
   export STORES=${CURRENT}/user/agent/stores
   export STORE_PASSWD=changeit
   export TH_HOST=<TH master host name>
   export CA_CERT=ca.cert.pem
   export FIPS_CA_TMP=/opt/fips_ca_tmp
   export COMMON_NAME=<your.connector.fqdn>
   ```

Windows commands:

```
set CURRENT=<full path to this folder>
```

```
set BC_OPTS="-storetype BCFKS -providername BCFIPS -J-
Djava.security.egd=file:/dev/urandom -providerpath
${CURRENT}/lib/agent/fips/bc-fips-1.0.2.jar -providerclass
org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider"
```

```
set TH=<Transformation Hub hostname>_<Transformation Hub port>
set STORES=%CURRENT%\user\agent\stores
set STORE_PASSWD=changeit
set TH_HOST=<TH master host name>
set CA_CERT=C:\Temp\ca.cert.pem
set FIPS_CA_TMP=C:\Temp\fips_ca_tmp
set COMMON_NAME=<your.connector.fqdn>
```

4. Create the `${CURRENT}/user/agent/stores` directory if it does not already exist.

   Linux command:

   ```
   mkdir -p ${STORES}
   ```

   Windows command:

   ```
   mkdir -p %STORES%
   ```

5. Apply the following workaround for a Java `keytool` issue:
   a. Create a new file, `agent.security`, in the applicable location for Linux or Windows:
      - **Linux:** `${CURRENT}/user/agent`
      - **Windows:** `%CURRENT%\current\user\agent`
   b. Add the following content to the newly created file, and then save it:

      ```
      security.provider.1=org.bouncycastle.jcajce.provider.BouncyCastleFipsP
      rovider
      security.provider.2=com.sun.net.ssl.internal.ssl.Provider BCFIPS
      security.provider.3=sun.security.provider.Sun
      ```

   c. Move the `${CURRENT}/lib/agent/fips/bcprov-jdk15on-168.jar` file to the `current` directory.

Creating a Keystore for SmartConnector on the SmartConnector Server

Follow the applicable steps according to your platform type, Windows or Linux.

Linux platform:

1. Create the keystore for SmartConnector:

```
cd ${STORES}
```

```
$CURRENT/jre/bin/keytool -keystore ${TH}.keystore.bcfips -genkeypair -
dname "cn=<your.connector.fqdn>, ou=<yourOU>, o=<yourCompany>, c=US" -
keyalg rsa -keysize 2048 -alias ${TH} -startdate -1d -validity 366 -
storepass ${STORE_PASSWD} -keypass ${STORE_PASSWD} ${BC_OPTS}
```

2. Create the certificate signing request (CSR) for SmartConnector:

```
$CURRENT/jre/bin/keytool -certreq -alias ${TH} -keystore
${TH}.keystore.bcfips -file ${COMMON_NAME}.csr -storepass ${STORE_PASSWD}
${BC_OPTS}
```

3. Copy the CSR file `${COMMON_NAME}.csr` to the Transformation Hub `/tmp` folder:

```
cp ${COMMON_NAME}.csr /tmp/
```

Windows platform:

1. Create the keystore for SmartConnector:

```
cd %STORES%
```

```
%CURRENT%\jre\bin\keytool %BC_OPTS% -genkeypair -alias %TH% -keystore
%STORES%\%TH%.keystore.bcfips -dname "cn=WIN-FO37IMBPNOI,OU=Arcsight,O=MF
,L=<location>,ST=<state(XX format)>,C=US" -keyalg rsa -keysize 2048 -
alias %TH% -startdate -1d -validity 366 -storepass %STORE_PASSWD% -
keypass %STORE_PASSWD%
```

2. Create the certificate signing request (CSR) for SmartConnector:

```
%CURRENT%\jre\bin\keytool -certreq -alias %TH% -keystore
%STORES%\%TH%.keystore.bcfips -file %STORES%\%TH%-cert-req -storepass
password
```

3. Copy the CSR file `${COMMON_NAME}.csr` to the Transformation Hub `/tmp` folder:

```
cp ${COMMON_NAME}.csr /tmp/
```

Signing a SmartConnector Certificate Signing Request on Transformation Hub

1. Set the environment:

```
export CA_CERT=ca.cert.pem
export COMMON_NAME=<your.connector.fqdn>
export TH=<Transformation Hub hostname>_<Transformation Hub port>
```

It is mandatory to use the same values that you specified in the SmartConnector server.

2. Run the following commands to sign the SmartConnector certificate signing request:

```
mkdir /tmp/smartconnector
mv ${COMMON_NAME}.csr /tmp/smartconnector/
cd /tmp/smartconnector
```

```
export CDF_APISERVER=$(kubectl get pods -n core -o custom-
columns=":metadata.name"| grep cdf-apiserver)
```

```
export PASSPHRASE=$(kubectl get secret vault-passphrase -n core -o json
2>/dev/null | jq -r '.data.passphrase')
```

```
export ENCRYPTED_ROOT_TOKEN=$(kubectl get secret vault-credential -n core
-o json 2>/dev/null | jq -r '.data."root.token"')
```

```
export VAULT_TOKEN=$(echo ${ENCRYPTED_ROOT_TOKEN} | openssl aes-256-cbc -
md sha256 -a -d -pass pass:"${PASSPHRASE}")
```

```
export CSR=$(cat ${COMMON_NAME}.csr)
```

```
WRITE_RESPONSE=$(kubectl exec -it -n core ${CDF_APISERVER} -c cdf-
apiserver -- bash -c "VAULT_TOKEN=$VAULT_TOKEN vault write -tls-skip-
verify -format=json RE/sign/coretech csr=\"${CSR}\"") && \
```

```
echo "${WRITE_RESPONSE}" | jq -r ".data | .certificate" > ${COMMON_
NAME}.signed.crt && \
```

```
echo "${WRITE_RESPONSE}" | jq -r ".data | .issuing_ca" > ${COMMON_
NAME}.issue_ca.crt && \
```

```
echo "${WRITE_RESPONSE}" | jq -r ".data | .certificate, if .ca_chain then
.ca_chain[] else .issuing_ca end" > ${COMMON_
NAME}.signed.cert.with.ca.crt
```

The signed certificate is available in the file: `${COMMON_NAME}.signed.crt`.
The issuing CA is available in the file: `${COMMON_NAME}.issue_ca.crt`.
The signed certificate with CA chain is available in the file: `${COMMON_NAME}.signed.cert.with.ca.crt`.

3. Retrieve the RE certificate:

```
/<TH Home Path>/scripts/cdf-updateRE.sh > /tmp/smartconnector/${CA_CERT}
```

Example: `/opt/arcsight/kubernetes/scripts/cdf-updateRE.sh >/tmp/smartconnector/${CA_CERT}`

> **Note:** For a cloud installation (Azure or AWS), log in to the bastion or jump host and run the script `cdf-updateRE.sh`:
>
> arcsight-platform-cloud-installer/cdf-deployer/scripts/cdf-updateRE.sh

4. Move the following files from Transformation Hub to the SmartConnector STORES directory:
   - `/tmp/smartconnector/ ${COMMON_NAME}.signed.crt`
   - `/tmp/smartconnector/${COMMON_NAME}.issue_ca.crt`
   - `/tmp/smartconnector/${COMMON_NAME}.signed.cert.with.ca.crt`
   - `/tmp/smartconnector/${CA_CERT}`

Updating the Keystore and Creating a Truststore on the SmartConnector Server

1. Update SmartConnector keystore with signed certificate:

   Linux commands:

```
cd ${STORES}/
```

```
$CURRENT/jre/bin/keytool -importcert -alias ${TH} -keystore
${TH}.keystore.bcfips -trustcacerts -file ${COMMON_
NAME}.signed.cert.with.ca.crt -storepass ${STORE_PASSWD} ${BC_OPTS}
```

   Verification: Run this command to verify the keystore. It must only have one entry in the key store.

```
$CURRENT/jre/bin/keytool -v -list -keystore ${TH}.keystore.bcfips -
storepass ${STORE_PASSWD} ${BC_OPTS} |grep -i alias
```

Windows commands:

```
cd %STORES%\
```

```
%CURRENT%\jre\bin\keytool -importcert -alias %TH% -keystore
%TH%.keystore.bcfips -trustcacerts -file %COMMON_
NAME%.signed.cert.with.ca.crt -storepass %STORE_PASSWD% %BC_OPTS%
```

Verification: Run this command to verify keystore. It must only have one entry in the key store.

```
%CURRENT%\jre\bin\keytool -v -list -keystore %TH%.keystore.bcfips -
storepass %STORE_PASSWD% %BC_OPTS% |grep -i alias
```

2. Create the SmartConnector truststore:

Linux commands:

```
cd ${STORES}/
```

```
$CURRENT/jre/bin/keytool -importcert -alias reca -trustcacerts -file
${CA_CERT} -keystore ${TH}.truststore.bcfips -storepass ${STORE_PASSWD}
${BC_OPTS}
```

When prompted, specify **yes** to trust the certificate.

Windows commands:

```
cd %{STORES}%/
```

```
%CURRENT\jre\bin\keytool -importcert -alias reca -trustcacerts -file %CA_
CERT% -keystore %TH%.truststore.bcfips -storepass %STORE_PASSWD% %BC_
OPTS%
```

When prompted, specify **yes** to trust the certificate.

Running the SmartConnector Setup

1. Run the following commands and note the keystore and truststore paths:

   Linux commands:

   ```
   echo ${STORES}/${TH}.keystore.bcfips
   echo ${STORES}/${TH}.truststore.bcfips
   ```

   Windows commands:

   ```
   echo %STORES%\%TH%.keystore.bcfips
   echo %STORES%\%TH%.truststore.bcfips
   ```

2. Navigate to the `bin` directory and run the agent setup script to install a connector with Transformation Hub as the destination.

   Linux commands:

   ```
   cd <installation dir>/current/bin
   ```

   ```
   ./runagentsetup.sh
   ```

   Windows commands:

   ```
   cd <installation dir>\current\bin
   ```

   ```
   runagentsetup.bat
   ```

3. Set **Use SSL/TLS** to **true**.
4. Set **Use SSL/TLS Authentication** to **true**.
5. When completing the Transformation Hub destination fields, use the value noted from step 2 for the truststore paths and the password used above for the truststore passwords.

Configuring a SmartConnector with Transformation Hub Destination with Client Authentication in Non-FIPS Mode

Follow these steps to configure a SmartConnector with a Transformation Hub (TH) destination with client authentication, in non-FIPS mode.

You will need to supply an intermediate certificate and key.

Preparing the SmartConnector Server

1. Prepare the SmartConnector:
   - **If the connector is not yet installed:**
     a. Run the installer.
     b. After the core software is installed, do the following in the window that opens: select **Select Global Parameters** > **Set FIPS Mode**, and set the FIPS Mode to **Disabled**.
   - **If the connector is already installed:**
     a. Run the installer.
     b. Select **Set Global Parameters** > **Set FIPS Mode**, and set the FIPS Mode to **Disabled**.
2. Navigate to the Connector's `current` directory:

   Linux command:

   ```
   cd <install dir>/current
   ```

   Windows command:

   ```
   cd <install dir>\current
   ```

3. Set the environment variables for the static values used by `keytool`:

   Linux commands:

   ```
   export CURRENT=<full path to this "current" folder>
   ```

   Example: `export CURRENT=/opt/CONNECTORS/TA003/current`

   ```
   export TH=<Transformation Hub hostname>_<Transformation Hub port>
   ```

   Example: `export TH=15.214.***.**`

   ```
   export CA_CERT=re_ca.cert.pem
   ```

   Example: `export CA_CERT=re_ca.crt.pem`

   ```
   export STORE_PASSWD=changeit
   export STORES=${CURRENT}/user/agent/stores
   export COMMON_NAME=<your.connector.fqdn>
   ```

Windows commands:

```
export CURRENT=<full path to this "current" folder>
set TH=<Transformation Hub hostname>_<Transformation Hub port>
set STORES=%CURRENT%\user\agent\stores
set STORE_PASSWD=changeit
```

4. Create the `${CURRENT}/user/agent/stores` directory if it does not already exist.

Linux command:

```
mkdir -p ${STORES}
```

Windows command:

```
mkdir -p %STORES%
```

Creating the Keystore for SmartConnector on the SmartConnector Server

Follow the applicable steps according to your platform type, Windows or Linux

1. Create the keystore for SmartConnector:

Linux commands:

```
cd ${STORES}
```

```
keytool$CURRENT/jre/bin/keytool -keystore ${TH}.keystore.jks -genkeypair
-dname "cn=your.connector.fqdn, ou=ArcSight, o=Micro Focus, c=US" -keyalg
rsa -keysize 2048 -alias ${TH} -startdate -1d -validity 366 -storepass
${STORE_PASSWD} -keypass ${STORE_PASSWD}
```

Window commands:

```
cd %STORES%
```

```
keytool%CURRENT%\jre\bin\keytool -keystore %TH%.keystore.jks -genkeypair
-dname "cn=your.connector.fqdn, ou=ArcSight, o=Micro Focus, c=US" -keyalg
rsa -keysize 2048 -alias %TH% -startdate -1d -validity 366 -storepass
%STORE_PASSWD% -keypass %STORE_PASSWD%
```

2. Create the certificate signing request (CSR) for SmartConnector:

Linux command:

```
$CURRENT/jre/bin/keytool -certreq -alias ${TH} -keystore
${TH}.keystore.jks -file ${COMMON_NAME}.csr -storepass ${STORE_PASSWD}
```

Windows command:

```
%CURRENT%\jre\bin\keytool -certreq -alias %TH% -keystore
%TH%.keystore.jks -file %COMMON_NAME%.csr -storepass %STORE_PASSWD%
```

3. Copy the CSR file `${COMMON_NAME}.csr` to the Transformation Hub `/tmp` folder:

```
cp ${COMMON_NAME}.csr /tmp/
```

Signing the SmartConnector Certificate Signing Request on Transformation Hub

1. Set the environment:

```
export CA_CERT=re_ca.cert.pem
export COMMON_NAME=<your.connector.fqdn>
export TH=<Transformation Hub hostname>_<Transformation Hub port>
```

Note: Use the same values that you specified in the smartconnector server.

2. Sign the smartconnector certificate signing request:

```
mkdir /tmp/smartconnector
```

```
mv ${COMMON_NAME}.csr /tmp/smartconnector/
```

```
cd /tmp/smartconnector
```

```
export CDF_APISERVER=$(kubectl get pods -n core -o custom-
columns=":metadata.name"| grep cdf-apiserver)
```

```
export PASSPHRASE=$(kubectl get secret vault-passphrase -n core -o json
2>/dev/null | jq -r '.data.passphrase')
```

```
export ENCRYPTED_ROOT_TOKEN=$(kubectl get secret vault-credential -n core
-o json 2>/dev/null | jq -r '.data."root.token"')
```

```
export VAULT_TOKEN=$(echo ${ENCRYPTED_ROOT_TOKEN} | openssl aes-256-cbc -
md sha256 -a -d -pass pass:"${PASSPHRASE}")
```

```
export CSR=$(cat ${COMMON_NAME}.csr)
```

```
WRITE_RESPONSE=$(kubectl exec -it -n core ${CDF_APISERVER} -c cdf-
apiserver -- bash -c "VAULT_TOKEN=$VAULT_TOKEN vault write -tls-skip-
verify -format=json RE/sign/coretech csr=\"${CSR}\"") && \
```

```
echo "${WRITE_RESPONSE}" | jq -r ".data | .certificate" > ${COMMON_
NAME}.signed.crt && \
```

```
echo "${WRITE_RESPONSE}" | jq -r ".data | .issuing_ca" > ${COMMON_
NAME}.issue_ca.crt && \
```

```
echo "${WRITE_RESPONSE}" | jq -r ".data | .certificate, if .ca_chain then
.ca_chain[] else .issuing_ca end" > ${COMMON_
NAME}.signed.cert.with.ca.crt
```

The signed certificate is in file `${COMMON_NAME}.signed.crt`.

The issuing CA is in file `${COMMON_NAME}.issue_ca.crt`.

The signed certificate with CA chain is in file `${COMMON_NAME}.signed.cert.with.ca.crt`.

3. Retrieve the RE certificate.

```
/<TH Home Path>/scripts/cdf-updateRE.sh > /tmp/${CA_CERT}
```

Example: `/opt/arcsight/kubernetes/scripts/cdf-updateRE.sh >${CA_CERT}`

Note: For a cloud installation (Azure or AWS), log in to the bastion or jump host and run the script `cdf-updateRE.sh`:

arcsight-platform-cloud-installer/cdf-deployer/scripts/cdf-updateRE.sh

Move the following files from the Transformation Hub to the connector `STORES` directory:

- `/tmp/smartconnector/ ${COMMON_NAME}.signed.crt`
- `/tmp/smartconnector/${COMMON_NAME}.issue_ca.crt`
- `/tmp/smartconnector/${COMMON_NAME}.signed.cert.with.ca.crt`
- `/tmp/smartconnector/${CA_CERT}`

Updating the Keystore and Create a Truststore on the SmartConnector Server

1. Update the SmartConnector keystore with a signed certificate.

   Linux commands:

   ```
   cd ${STORES}/
   ```

   ```
   $CURRENT/jre/bin/keytool -importcert -alias ${TH} -keystore
   ${TH}.keystore.jks -trustcacerts -file ${COMMON_
   NAME}.signed.cert.with.ca.crt -storepass ${STORE_PASSWD}
   ```

   When prompted, specify **yes** to trust the certificate.

   Verification: Run the following command to verify the keystore, and ensure that it has only one entry in the keystore.

   ```
   $CURRENT/jre/bin/keytool -v -list -keystore ${TH}.keystore.jks -storepass
   ${STORE_PASSWD} |grep -i alias
   ```

   Windows commands:

   ```
   cd %STORES%\
   ```

   ```
   %CURRENT%\jre\bin\keytool -importcert -alias %TH% -keystore
   %TH%.keystore.jks -trustcacerts -file %COMMON_
   NAME%.signed.cert.with.ca.crt -storepass %STORE_PASSWD%
   ```

   Verfication: Run the following command to verify the keystore, and ensure that it has only one entry in the keystore.

   ```
   %CURRENT%\jre\bin\keytool -v -list -keystore %TH%.keystore.jks -storepass
   %STORE_PASSWD% |grep -i alias
   ```

2. Create the SmartConnector truststore.

   Linux commands:

   ```
   cd ${STORES}/
   ```

   ```
   $CURRENT/jre/bin/keytool -importcert -alias CARoot -trustcacerts -file
   ${CA_CERT} -keystore ${TH}.truststore.jks -storepass ${STORE_PASSWD}
   ```

   When prompted, specify **yes** to trust the certificate.

Windows command:

```
%CURRENT%\jre\bin\keytool -importcert -alias CARoot -trustcacerts -file
%CA_CERT% -keystore %TH%.truststore.jks -storepass %STORE_PASSWD%
```

When prompted, specify **yes** to trust the certificate.

Run the SmartConnector Setup

1. Run the following command and note the trust store path:

   Linux commands:

   ```
   echo ${STORES}/${TH}.keystore.jks
   ```

   ```
   echo ${STORES}/${TH}.truststore.jks
   ```

   Windows commands:

   ```
   echo %STORES%\%TH%.keystore.jks
   ```

   ```
   echo %STORES%\%TH%.truststore.jks
   ```

2. Navigate to the bin directory, and run agent setup script to install a connector with Transformation Hub as the destination.

   Linux commands:

   ```
   cd <installation dir>/current/bin
   ```

   ```
   ./runagentsetup.sh
   ```

   Windows commands:

   ```
   cd <installation dir>\current\bin
   ```

   ```
   runagentsetup.bat
   ```

3. Set **Use SSL/TLS** to **true**.
4. Set **Use SSL/TLS Authentication** to **true**.
5. When completing the Transformation Hub destination fields, use the value noted from Step 1 for the key store and trust store paths and the password used above for the store password.

Configuring a SmartConnector with a Transformation Hub Destination without Client Authentication in FIPS Mode

Follow these steps to configure a SmartConnector with a Transformation Hub destination without client authentication in FIPS mode.

Preparing the SmartConnector Server

1. Prepare the SmartConnector:
   - **If the connector is not yet installed:**
     a. Run the installer.
     b. After the core software is installed, do the following in the window that opens: select **Select Global Parameters** > **Set FIPS Mode**, and set the FIPS Mode to **Disabled**.
   - **If the connector is already installed:**
     a. Run the installer.
     b. Select **Set Global Parameters** > **Set FIPS Mode**, and set the FIPS Mode to **Enabled**.
2. Navigate to the Connector's `current` directory:

   Linux command:

   ```
   cd <install dir>/current
   ```

   Windows command:

   ```
   cd <install dir>\current
   ```

3. Set the environment variables for the static values used by `keytool`:

   Linux commands:

   ```
   export CURRENT=<full path to this "current" folder>
   ```

   ```
   export BC_OPTS="-storetype BCFKS -providername BCFIPS -providerclass
   org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider -providerpath
   ${CURRENT}/lib/agent/fips/bc-fips-1.0.2.jar -J-
   Djava.security.egd=file:/dev/urandom"
   ```

   For Connector 8.0, use `bc-fips-1.0.0.jar` in the command above.

```
export TH=<Transformation Hub hostname>_<Transformation Hub port>
export STORES=${CURRENT}/user/agent/stores
export STORE_PASSWD=changeit
export TH_HOST=<TH master host name>
export CA_CERT=re_ca.cert.pem
export FIPS_CA_TMP=/opt/fips_ca_tmp
```

Windows commands:

```
set CURRENT=<full path to this "current" folder>
```

```
set BC_OPTS="-storetype BCFKS -providername BCFIPS -J-
Djava.security.egd=file:/dev/urandom -providerpath
${CURRENT}/lib/agent/fips/bc-fips-1.0.2.jar -providerclass
org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider"
```

```
set TH=<Transformation Hub hostname>_<Transformation Hub port>
set STORES=%CURRENT%\user\agent\stores
set STORE_PASSWD=changeit
set TH_HOST=<TH master host name>
set CA_CERT=C:\Temp\ca.cert.pem
set INTERMEDIATE_CA_CRT=C:\Temp\intermediate.cert.pem
set FIPS_CA_TMP=C:\Temp\fips_ca_tmp
```

4. Create the `${CURRENT}/user/agent/stores` directory if it does not already exist.

   Linux command:

   ```
   mkdir -p ${STORES}
   ```

   Windows command:

   ```
   mkdir -p %STORES%
   ```

5. Apply the following workaround for a Java `keytool` issue:
   a. Create a new file, `agent.security`, in the applicable location for Linux or Windows:
      - **Linux:** `${CURRENT}/user/agent`
      - **Windows:** `%CURRENT%\current\user\agent`
   b. Add the following content to the newly created file, and then save it:

      ```
      security.provider.1=org.bouncycastle.jcajce.provider.BouncyCastleFipsP
      rovider
      ```

```
security.provider.2=com.sun.net.ssl.internal.ssl.Provider BCFIPS
security.provider.3=sun.security.provider.Sun
```

c. Move the `${CURRENT}/lib/agent/fips/bcprov-jdk15on-168.jar` file to the `current` directory.

Creating a CA CERT File on Transformation Hub

Follow these steps to create a `${CA_CERT}` file with the content of the root CA certificate:

1. Set the environment:

```
export CA_CERT=re_ca.cert.pem
```

2. Create a certificate:

```
/<TH Home Path>/scripts/cdf-updateRE.sh > /tmp/${CA_CERT}
```

Example: `/opt/arcsight/kubernetes/scripts/cdf-updateRE.sh > /tmp/${CA_CERT}`

> **Note:** For a cloud installation (Azure or AWS), log in to the bastion or jump host and run the script `cdf-updateRE.sh`:
>
> arcsight-platform-cloud-installer/cdf-deployer/scripts/cdf-updateRE.sh

3. Move this file from Transformation Hub to the connector STORES directory.

Creating Truststore on the SmartConnector Server

Create truststore by running the following command:

Linux command:

```
keytool$CURRENT/jre/bin/keytool -noprompt -importcert -storepass ${STORE_
PASSWD} -destkeystore ${STORES}/${TH}.truststore.bcfips -alias reca -file
${CA_CERT} -storetype BCFKS -providerclass
org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider -providerpath
${CURRENT}/lib/agent/fips/bc-fips-1.0.2.jar
```

When prompted, specify **yes** to trust the certificate.

Windows command:

```
keytool%CURRENT%\jre\bin\keytool -noprompt -importcert -storepass %STORE_
PASSWD% -destkeystore %STORES%\%TH%.truststore.bcfips -alias reca -file %CA_
```

```
CERT% -storetype BCFKS -providerclass
org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider -providerpath
%CURRENT%\lib\agent\fips\bc-fips-1.0.2.jar
```

When prompted, specify **yes** to trust the certificate.

Running the SmartConnector Setup

1. Run the following command and note the keystore and truststore paths:

   Linux command:

   ```
   echo ${STORES}/${TH}.truststore.bcfips
   ```

   Windows command:

   ```
   echo %STORES%\%TH%.truststore.bcfips
   ```

2. Navigate to the bin directory and run the agent setup script to install a connector with Transformation Hub as the destination.

   Linux commands:

   ```
   cd <installation dir>/current/bin
   ```

   ```
   ./runagentsetup.sh
   ```

   Windows commands:

   ```
   cd <installation dir>\current\bin
   ```

   ```
   runagentsetup.bat
   ```

3. Set **Use SSL/TLS** to **true**.
4. Set **Use SSL/TLS Authentication** to **false**.
5. When completing the Transformation Hub destination fields, use the value noted from Step 2 for the truststore paths and the password used above for the truststore passwords.

Cleaning up the Certificate File

Run the following command to delete the certificate file:

Caution: The following file should be deleted to prevent the distribution of security certificates that could be used to authenticate against the Transformation Hub. These files are very sensitive and must not be distributed to other machines.

Linux command:

```
rm ${STORES}/${CA_CERT}
```

Windows command:

```
del %\STORES%\%CA_CERT%
```

Configuring ArcMC to Manage a Transformation Hub

Note: For Azure or AWS Transformation Hub, you complete configuration procedures as detailed in Completing Integration with Azure Transformation Hub or Completing Integration with AWS Transformation Hub before you perform the procedures in this section.

ArcMC serves as the management UI for Transformation Hub. In order for ArcMC to manage a Transformation Hub, the Transformation Hub must be added as a managed host to ArcMC.

This process will include these steps, which are explained below:

Fusion ArcMC Instructions

- "Retrieving the CDF certificate:" below
- "Configuring the CDF cluster" on the next page
- "Configuring Fusion ArcMC: " on the next page

Standalone ArcMC Instructions

- "Retrieving the ArcMC certificate " on page 473
- "Configuring the CDF cluster" on page 473
- "Configuring ArcMC:" on page 473

Fusion ArcMC Instructions

Retrieving the CDF certificate:

1. On the initial master node of the cluster, run the following:
   `# ${K8S_HOME}/scripts/cdf-updateRE.sh`
2. Copy the contents of this certificate, from `--BEGIN cert to END cert--` to the clipboard.

Configuring the CDF cluster

1. Log in to the CDF management portal.
2. Select **Deployment > Deployments.**
3. Click **...** (Browse) on the far right and choose **Reconfigure**. A new screen will be opened in a separate tab.
3. Scroll down to the Management Center Configuration section. Then, specify values as described for the following:
   - **Username:**`admin`
   - Password: Use your Transformation Hub password.
   - Enter the ArcMC hostname and port 443 (for example, `arcmc.example.com:443`). If ArcMC was installed as a non-`root` user, specify port 9000 instead.
   - **ArcMC certificates:** Paste the text of the generated CDF server certificates for Fusion ArcMC (or Standalone ArcMC generated certificate) you copied to the clipboard as described above.
4. Click **Save**. Web services pods in the cluster will be restarted

Configuring Fusion ArcMC:

1. Log in to the Fusion UI.
2. From the Dashboard's left side panel select **ArcMC**, a new tab will open.
3. Click **Node Management > View All Nodes.**
4. In the navigation bar, click **Default** (or the ArcMC location where you wish to add Transformation Hub). Then click **Add Host,** and specify the following values:
   - **Type:** Select Transformation Hub - Container Deployment Foundation (CDF)
   - **Hostname:** Virtual IP of the Transformation Hub for an HA environment, or master node hostname for any single-master node environment.
   - **Port**: 32080
   - **Cluster Port:** 443
   - **Cluster Username:** admin
   - **Cluster Password:** <use CDF Management Portal password>
   - **Cluster Certificate:** Paste the contents of the CDF certificate you copied earlier.
5. Click **Add**. The Transformation Hub is added as a managed host.

Standalone ArcMC Instructions

Retrieving the ArcMC certificate

1. Log into ArcMC.
2. Click **Administration > System Admin > SSL Server Certificate > Generate Certificate.**
3. On the **Enter Certificate Settings** dialog, specify the required settings. In **Hostname**, your certificate settings must match the FQDN of your ArcMC.
4. Click **Generate Certificate.**
5. Once the certificate is generated, click **View Certificate** and copy the full content from `--BEGIN cert to END cert--` to the clipboard.

Configuring the CDF cluster

1. Log in to the CDF management portal.
2. Select **Deployment > Deployments.**
3. Click **...** (Browse) on the far right and choose **Reconfigure**. A new screen will be opened in a separate tab.
3. Scroll down to the Management Center Configuration section. Then, specify values as described for the following:
   - **Username:** `admin`
   - Password: Use your Transformation Hub password.
   - Enter the ArcMC hostname and port 443 (for example, `arcmc.example.com:443`). If ArcMC was installed as a non-`root` user, specify port 9000 instead.
   - **ArcMC certificates:** Paste the text of the generated CDF server certificates for Fusion ArcMC (or Standalone ArcMC generated certificate) you copied to the clipboard as described above.
4. Click **Save**. Web services pods in the cluster will be restarted

Configuring ArcMC:

1. Log in to the ArcMC.
2. Click **Node Management > View All Nodes.**
3. In the navigation bar, click Default (or the ArcMC location where you wish to add Transformation Hub). Then click **Add Host,** and specify the following values:

- **Type:** Select Transformation Hub - Container Deployment Foundation (CDF)
- **Hostname:** Virtual IP of the Transformation Hub for an HA environment, or master node hostname for any single-master node environment.
- **Port**: 32080
- **Cluster Port:** 443
- **Cluster Username:** admin
- **Cluster Password:** <use CDF Management Portal password>
- **Cluster Certificate:** Paste the contents of the CDF certificate you copied earlier.

4. Click **Add**. The Transformation Hub is added as a managed host.

Understanding How Data is Produced and Consumed

Transformation Hub's publish-subscribe messaging system uses SmartConnectors and Collectors to produce event data, and supports Logger, Recon, and ESM, as well as Apache Hadoop and other third-party consumers.

While Transformation Hub can support a very high event flow (millions of events per second), the event rate for each producer and consumer will generally be much smaller (tens of thousands of events per second). Actual event flow will depend on your specific implementation and tuning applied, as well as server resources available, such as memory and CPU.

Producing Events with SmartConnectors

SmartConnectors can publish events to Transformation Hub topics. In order to publish events, you must configure your SmartConnectors to use the Transformation Hub destination. To send events to multiple topics, you can configure multiple concurrent destinations with the same Transformation Hub using different topics.

Once configured with a Transformation Hub destination, the SmartConnector sends events to Transformation Hub's Kafka cluster, which can then further distribute events to real-time analysis and data warehousing systems. Other applications, including Recon, ESM, Logger, and any third-party application that supports retrieving data from Kafka can receive them, for example, Apache Hadoop.

Transformation Hub balances incoming events between nodes, by distributing them evenly between the partitions in the configured topic.

Acknowledgments ("acks") ensure that Transformation Hub has received the event before the SmartConnector removes it from its local queue. You can disable acknowledgments, require acknowledgment only from the primary replica, or require every replica to acknowledge the event. (Acknowledgments do not indicate that consumers, such as Logger, have received the event data, only that Transformation Hub itself has.)

Supported SmartConnector versions encode their own IP address as meta-data in the Kafka message for consumers that require that information such as Logger Device Groups.

- For information on supported SmartConnector versions, see the *SODP Support Matrix*.
- For more information about SmartConnectors and how to configure a Transformation Hub destination, refer to the CEF Destinations chapter of the *SmartConnect Install and User Guide*.

Legacy Micro Focus documentation is available for download from the Micro Focus support community.

Consuming Events with ESM

ESM agents are the consumers for Transformation Hub's publish-subscribe messaging system. An ESM agent can connect to Transformation Hub and consume all events in binary or Avro format for the topics to which it is subscribed.

Additionally, ESM provides data monitors to monitor Transformation Hub health.

- For information on supported versions of ESM and SmartConnectors, see the *SODP Support Matrix.*
- For instructions on configuring a supported version of ESM as a consumer, see the *ESM Administrator's Guide*.

Consuming Events with Logger

To subscribe to Transformation Hub topics with Logger, you must configure a receiver on a supported Logger version to receive the Transformation Hub events. Logger's Transformation Hub receivers are consumers for Transformation Hub's publish-subscribe messaging system. They receive events in Common Event Format (CEF) from Transformation Hub topics. A Logger Transformation Hub receiver connects to Transformation Hub and consumes all events for the topics it subscribes to.

When configuring a Logger Transformation Hub receiver, specify the worker node FQDNs, topics to consume from, and consumer group name. You can configure multiple Loggers to consume from the same topic as a part of a consumer group.

For more information about Logger and how to configure a Transformation Hub receiver, refer to the *Logger Administrator's Guide,* available for download from the Micro Focus support community.

Kafka consumers can take up to 24 hours for the broker nodes to balance the partitions among the consumers. Check the Transformation Hub Kafka Manager **Consumers** page to confirm all consumers are consuming from the topic.

Sending Transformation Hub Data to Logger

For a Logger to be able to consume Transformation Hub events, the Logger must have a Transformation Hub receiver configured with the Transformation Hub worker nodes, consumer group, and event topic list. SmartConnectors that send data to Transformation Hub must have a Transformation Hub destination.

A group of Loggers, called a pool, can be configured to receive and distribute events between themselves. This works similarly to the Logger pool created by using the Logger Smart Message Pool destination on SmartConnectors. The difference is that when the SmartConnectors have a Logger Smart Message Pool destination, the event load is balanced by each SmartConnector, but when the SmartConnectors have a Transformation Hub destination, the event load is balanced by the Loggers.

Additional Loggers can be added to the pool simply by configuring the same Transformation Hub worker nodes, consumer group, and event topic list in the new Logger's Transformation Hub receivers, without having to reconfigure either the existing Loggers or any SmartConnectors.

The events retrieved by the Logger pool are distributed among the Loggers in the pool. If one Logger is down, new events are rebalanced among existing Loggers. When a Logger is added or removed from the Consumer Group, the event load is distributed across the pool of Loggers.

To send events from a group of SmartConnectors to a pool of Loggers, configure their Transformation Hub destinations to send events to the topic from which the Logger pool is consuming.

To configure Logger to subscribe to event data from specific SmartConnectors, you can do either of the following:

- Configure all the SmartConnectors to publish events to the same topic. Configure the Logger's Transformation Hub receiver to subscribe to this event topic.
- Configure each SmartConnector to publish events to different topics and then configure the Transformation Hub receiver on the Logger to subscribe to multiple event topics.

Tip: Loggers in the same Logger pool do not consume the same events, since they are in the same Consumer Group. In high availability situations, you need events to be stored on two different Loggers. To store the same events on two Loggers, configure the Loggers to have different Consumer Group names, but subscribe them to the same event topic.

The number of Loggers in a Logger pool is restricted by the number of event topic partitions configured on the Container Deployment Foundation. For example, if there are only five partitions configured, only five Loggers will receive the events. If you have more than five Loggers configured in the same Consumer Group, some Loggers will not normally receive events, but will be available as hot spares. When adding receivers, be sure to increase the number of event topic partitions. See Managing Topics for more information.

Procedure to Send Transformation Hub Data to Logger

1. Configure the SmartConnector:
   - Set up a SmartConnector to publish to a particular Transformation Hub topic. Connectors can only send to a single topic for each destination. Additional destinations need to be configured if each event needs to go to multiple topics. Note the number of partitions in the topic.
   - For more information about SmartConnectors and how to configure a Transformation Hub destination, refer to the CEF Destinations chapter of the SmartConnector User's Guide, available for download from the Micro Focus support community.
2. Configure Logger:
   - Create a Transformation Hub receiver on each Logger in the Logger pool.
   - Configure each receiver to subscribe to the topics to which the SmartConnectors are publishing data. To subscribe to multiple topics, indicate the topics by specifying them in the Event Topic List parameter (a list of comma-separated values) while configuring the Transformation Hub receiver.
   - Configure each receiver to be in the same Consumer Group.

Example Setup with Multiple Loggers in a Pool

You can set up your Logger pools to subscribe to events from a particular device type, such as "Firewall." To do this, you would:

1. In ArcMC, create a Kafka topic named *Firewall*.
2. Configure all the SmartConnectors that handle firewall events to publish these events to topic "Firewall."

3. Configure the Loggers in the Logger pool:
   - Create a Transformation Hub Receiver on each Logger in the pool.
   - Configure the receivers to subscribe to the event topic “Firewall,” and include them in the “Logger_Firewall” Consumer Group.

Once the configuration is set up properly, the Logger pool will subscribe to device type *Firewall*.

This example assumes that the Transformation Hub is being managed by an ArcSight Management Center for topic creation. Topics can also be managed through the Kafka Manager UI.

Consuming Events with Third-Party Applications

Transformation Hub is designed with support for third-party tools. You can create a standard Kafka consumer and configure it to subscribe to Transformation Hub topics. By doing this you can pull Transformation Hub events into your own data lake.

Custom consumers must use Kafka client libraries of version 0.11 or later.

- All Transformation Hub nodes, consumers, and producers must be properly configured for forward and reverse DNS lookup, and be time-synchronized, using a time server such as NTP.
- Events are sent in standard CEF (CEF text), binary (exclusively for ESM consumption), or Avro format. Any software application that can consume from Kafka and handle the event format can process events.
- You can set up multiple consumer groups, and each group will get a copy of every event. Therefore you can have Logger and Apache Hadoop configured to consume from the same topic and each will get a copy of every event. This enables fanning out multiple copies of events without reconfiguring SmartConnectors or using additional CPU or network resources for them.

Consuming Transformation Hub Events with Apache Hadoop

Apache Hadoop is a software framework that enables the distributed processing of large data sets across clusters of computers. You can send Transformation Hub events to Hadoop by using Apache Flume.

This section describes how to set up the Apache Flume agent to transfer Common Event Format (CEF) events from an Transformation Hub Kafka cluster to Hadoop Distributed File System (HDFS).

Architecture for Kafka to Hadoop Data Transfer

Apache Flume uses a source module to read a Kafka topic containing CEF events, and it then transfers the events using a memory channel, and persists them to HDFS using a sink module. The CEF files are stored on HDFS by time, in a year/month/day/hour directory structure.

Using Apache Flume to Transfer Events to Hadoop

One of the applications you could use to transfer Transformation Hub events into your data lake is Apache Flume. Flume is designed to push data from many sources to the various storage systems in the Hadoop ecosystem, such as HDFS and HBase. This section describes how to use Apache Flume as a data transfer channel to transfer events from Transformation Hub to Apache Hadoop or other storage systems.

Prerequisites

- Transformation Hub installed.
- Flume installed: For information on how to install and configure Flume, refer to the *Flume documentation*.
- Storage system installed: Refer to your storage system documentation.

Procedure

Flume is controlled by an agent configuration file. You must configure Transformation Hub as the source agent, your storage system as the sink agent, and ZooKeeper as the channel agent in this file.

To configure Transformation Hub as the source:

Edit the agent configuration file to include the required properties, as in the table below. Configure other properties as needed for your environment.

Required Kafka Source Configuration

Property	Description
`type`	Set to `org.apache.flume.source.kafka.KafkaSource`.
`topic`	The Event Topic from which this source reads messages. Flume supports only one topic per source.

To configure the sink:

The required configuration varies. Refer to the Flume documentation for details on your storage system. The section Consuming Events with Apache Flume provides an example of how to configure Apache Hadoop as the sink.

Setting Up Flume to Connect with Hadoop

In the simplest deployment model, you need to deploy the Apache Flume agent on a Hadoop node server to pull events, and send them to Hadoop Distributed File System (HDFS).

Hadoop must be installed before you can connect it with Flume. If you do not already have your own Hadoop deployment, you can deploy Hadoop on a Red Hat Enterprise Linux 7.2 host. For more information, see Setting Up Hadoop.

For a detailed discussion of connecting Apache Flume with Hadoop, consult the *Apache Documentation*.

Sample Flume Configuration File

Before starting Apache Flume, create a configuration file based on the template below.

The configuration file should reside in `bin/flume/conf/`. This file is called *kafka.conf* in our example. You can name your own configuration file whatever is appropriate.

```
####################################################
#Sample Flume/Kafka configuration file
####################################################
#defines Kafka Source, Channel, and Destination aliases
tier1.sources = source1
tier1.channels = channel1
tier1.sinks = sink1
#Kafka source configuration
tier1.sources.source1.type = org.apache.flume.source.kafka.KafkaSource
tier1.sources.source1.kafka.bootstrap.servers= kafkaIP1:9092, kafkaIP2:9092,…
tier1.sources.source1.kafka.topics = th-cef
tier1.sources.source1.kafka.consumer.group.id = flume
tier1.sources.source1.channels = channel1
tier1.sources.source1.interceptors = i1
```

```
tier1.sources.source1.interceptors.i1.type = timestamp
tier1.sources.source1.kafka.consumer.timeout.ms = 150
tier1.sources.source1.kafka.consumer.batchsize = 100
#Kafka Channel configuration
tier1.channels.channel1.type = memory
tier1.channels.channel1.capacity = 10000
tier1.channels.channel1.transactionCapacity = 1000
#Kafka Sink (destination) configuration
tier1.sinks.sink1.type = hdfs
tier1.sinks.sink1.channel = channel1
tier1.sinks.sink1.hdfs.path = hdfs://localhost:9000/opt/\
hadoop/cefEvents/year=%y/month=%m/day=%d
tier1.sinks.sink1.hdfs.rollInterval = 360
tier1.sinks.sink1.hdfs.rollSize = 0
tier1.sinks.sink1.hdfs.rollCount = 0
tier1.sinks.sink1.hdfs.fileType = DataStream
tier1.sinks.sink1.hdfs.filePrefix = cefEvents
tier1.sinks.sink1.hdfs.fileSuffix = .cef
tier1.sinks.sink1.hdfs.batchSize = 100
tier1.sinks.sink1.hdfs.timeZone = UTC
```

Setting Up Hadoop

This is an overview of the steps necessary to install Apache Hadoop 2.7.2 and set up a one-node cluster. For more information, refer to the *Hadoop Documentation* for your version.

To install Hadoop:

1. Be sure that your environment meets the operating system and Java prerequisites for Hadoop.
2. Add a user named 'hadoop'.
3. Download and unpack Hadoop.
4. Configure Hadoop for pseudo-distributed operation.
   - Set the environment variables.
   - Set up passphraseless SSH.
   - Optionally, set up Yarn. (You will not need Yarn if you want to use Hadoop only storage and not for processing.)
   - Edit the Hadoop configuration files to set up a core location, a Hadoop Distributed File System (HDFS) location, a replication value, a NameNode and a DataNode.
   - Format the Name node.
5. Start the Hadoop server using the tools provided.
6. Access Hadoop Services in a browser and login as the user "hadoop."
7. To create the Hadoop cefEvents directory, run the following command:

   ```
   hadoop fs -mkdir /opt
   hadoop fs -mkdir /opt/hadoop
   hadoop fs -mkdir /opt/hadoop/cefEvents
   ```

8. To grant permissions for Apache Flume to write to this HDFS, run the following command:

   ```
   hadoop fs -chmod 777 -R /opt/hadoop
   hadoop fs -ls
   ```

9. To check Hadoop system status, run the following command:

   ```
   hadoop dfsadmin -report
   ```

10. To view the files transferred by Flume to Hadoop, run the following command:

    ```
    hadoop fs -ls -R /
    ```

Configuring Consumers and Producers for High Availability

Configure the Transformation Hub Kafka cluster endpoint to avoid single points of failure in both the producers sending data to Transformation Hub (such as SmartConnectors), and the

consumers subscribing to data from the Transformation Hub (such as Logger and ESM).

For Producers

Configure the **Initial Host:Port(s)** parameter field in the Transformation Hub Destination to include all Kafka broker (worker) nodes as a comma-separated list.

Provide all Kafka broker (worker) nodes for a producer and a consumer configuration to avoid a single point of failure. For example, broker_hostname1:9093, broker_hostname2:9093, broker_hostname3:9093.

For more information about how Kafka handles this using bootstrap.servers, see the *Kafka Documentation*.

For Consumers

Configure the **Transformation Hub host(s) and port** parameter field in the Receiver to include all Kafka cluster nodes as a comma-separated list.

For more information about how Kafka handles this using bootstrap servers, see the *Kafka Documentation*.

Understanding Data Compression

Transformation Hub compression settings affect data in two general places, communication and storage. Specifically, this refers to data stored on disk, in Kafka topic partitions, and data that is in transit.

- All external producers such as connectors, collectors, and internal producers, like routing and CEF2Avro processors, compress data before sending it.
- For data in transit, data compression is controlled by the producer's configuration.

Data Consumers

There is no property that controls data compression on consumers. Consumers read metadata from each message, which indicates the correct decompression algorithm to use. Since this is evaluated on a message-by-message basis, the consumer's behavior does not depend on which topic it is consuming from. A single topic might contain messages which have been compressed with different compression algorithms (also referred to as compression types or codecs).

Data Storage (Data at Rest)

The algorithm used to compress stored data is determined by the topic configuration. All Transformation Hub topics, except `th-arcsight-avro` and `mf-event-avro-enriched`, currently use the default compression type, which is the same as that used by `producer`. This

configuration choice means the topic will retain the original compression algorithm set by the producer. By leaving this as producer-defined, there is flexibility for the producer to send either compressed (using any supported codec) or uncompressed data.

The `mf-event-avro-enriched` topic is an exception because the database scheduler reads from this topic, but does not yet have support for reading messages encoded with the ZStandard (zstd) compression algorithm. Therefore, there is a specific, out-of-the-box value for this topic, to insure that the database scheduler can read it, no matter what over-the-wire compression was used.

Topic	Compression Type	Transformation Hub Version Support
All topics except `th-arcsight-avro` and `mf-event-avro-enriched`	producer (default)	3.4.0 and earlier (3.5.0 and earlier for `mf-event-avro-enriched`)
`th-arcsight-avro`	gzip	3.4.0 and 3.3.0
`th-arcsight-avro`	uncompressed	3.2.0 and earlier
`mf-event-avro-enriched`	gzip	3.5.0

Configuring Compression

There are two places in the Kafka architecture where compression can be configured: the producer and the topic.

- Producer-level compression is set on the producer; for example, in SmartConnector Transformation Hub destination parameters. For producers that reside inside TH, such as routing and stream processors, the compression algorithm is configured on the Transformation Hub configuration page, during deployment.
- Topic-level compression can be set with Kafka Manager (using **Topic > Update Config Menu**); however, it is strongly recommended that settings be left at default values.

Compression Types

While Kafka supports a handful of compression types, Transformation Hub implements only two types: gzip and zstd.

- **gzip:** By default, gzip is used for Transformation Hub routing and stream processors, as well as for SmartConnectors. This is for backward compatibility and might change in a future release.
- **zstd:** Testing has shown that zstd uses less bandwidth, storage, and CPU resources than gzip. For bandwidth constrained networks, higher EPS is typically seen when using zstd; however actual results are unique to each environment. Third-party Java producers should use kafka-

clients version 2.1.0 or later, for zstd support. ArcSight consumers compatible with zstd include Logger 7.0, ESM 7.2, IDI 1.1, or later.

Pushing JKS files from ArcMC

You can push JKS (Java Keystore) files to multiple managed SmartConnectors in ArcMC. First, you will upload the files to a file repository in ArcMC, then push them out to their destination SmartConnectors. You must then configure and enable the Kafka destination on all SmartConnectors.

To upload the Java Keystore files:

1. Prepare the .jks files you want to push and store them in a secure network location.
2. In ArcMC, click **Administration > Repositories > New Repository.**
3. In **Name, Display Name**, and **Item Display Name**, specify KAFKA_JKS
4. Enter other required details as needed, then click **Save**.
5. Click **Upload to Repository.**
6. Follow the prompts in the upload wizard and browse to the first .jks file. Make sure to choose the individual file option.
7. Upload as many files as needed by repeating the upload wizard.

To push the files to multiple SmartConnectors:

1. In ArcMC, browse to the file repository for the `.jks` files.
2. Click the **Upload** arrow.
3. Follow the prompts in the wizard and select your destination SmartConnectors.
4. The files are pushed to the managed SmartConnectors and stored in the designated SmartConnector folder.

To configure the Kafka destination on all SmartConnectors:

In ArcMC, click **Node Management > Connectors** tab.

1. Select the SmartConnectors to be configured.
2. Choose **Add a destination** and pick the Kafka destination type.
3. Add the destination details along with the `.jks` path and password, and save the changes.

Integrating Intelligence with ESM

To enable ESM to receive the analysed entities and alerts information from Intelligence, you need to install and configure the ArcSight REST FlexConnectors.

The REST FlexConnector provides a configurable method to collect events from Intelligence and send them to ESM. Intelligence's Alerts and Entities APIs serve as the REST API endpoints from which the REST FlexConnectors collect data.

The REST FlexConnectors use the OAuth2 authentication to get permission to receive events from Intelligence. The events collected by the FlexConnectors are in JSON format.

With the help of one JSON parser file each for Alert data and Entities data, these events are converted into a format that can be understood and received by ESM.

Using the JSON Parser Files

The parser file that is used for alerts data is **alerts.jsonparser.properties**.

```
trigger.node.location=/data
token.count=14
token[0].name=alertId
token[0].type=String
token[0].location=alertId

token[1].name=datasource
token[1].type=String
token[1].location=datasource

token[2].name=alertTime
token[2].type=Long
token[2].location=timestamp

token[3].name=risk
token[3].type=Integer
token[3].location=risk

token[4].name=contribution
token[4].type=Integer
token[4].location=contribution

token[5].name=significance
token[5].type=String
token[5].location=significance
```

```
token[6].name=threat
token[6].type=String
token[6].location=templates/threat

token[7].name=family
token[7].type=String
#token[7].format=__uri()
token[7].location=templates/family

token[8].name=teaser
token[8].type=String
token[8].location=templates/teaser

token[9].name=alert
token[9].type=String
token[9].location=templates/alert

token[10].name=anomalyTypes
token[10].type=String
token[10].location=anomalyTypes

token[11].name=numAnomalies
token[11].type=Integer
token[11].location=numAnomalies

token[12].name=category
token[12].type=String
token[12].location=category

token[13].name=scrollId
token[13].type=String
token[13].location=/scrollId

#(End Of Token Definitions)

#tokens

event.externalId=alertId
event.deviceCustomNumber1=risk
event.deviceCustomNumber1Label=__stringConstant("RiskScore")
event.deviceCustomNumber2=contribution
event.deviceCustomNumber2Label=__stringConstant("Contribution")
event.deviceCustomNumber3=__safeToLong(__regexToken(alert,.?risk=([^\\s]+)
.*))
event.deviceCustomNumber3Label=__stringConstant("Entity Risk Score")

event.fileName=__regexToken(alert,.?entity name="([^"]+)".*)
event.fileHash=__regexToken(alert,.?hash="([^"]+)".*)
```

```
event.fileType=__regexToken(alert,.?type="([^"]+)".*)

event.message=alert
event.reason=teaser
event.aggregatedEventCount=numAnomalies
event.deviceEventCategory=category
event.deviceReceiptTime=__createLocalTimeStampFromSecondsSinceEpoch
(alertTime)

#tags
#event.destinationUserId=id
#event.deviceCustomString5=tags
#event.destinationUserName=otherName
#event.deviceCustomString2=source
#event.message=desc

#Other Mappings
event.name=family
event.deviceEventClassId=threat
event.deviceVendor=__stringConstant("Micro Focus")
event.deviceProduct=__stringConstant("Interset")
event.deviceSeverity=significance

#Agent Severity
severity.map.veryhigh.if.deviceSeverity=9,10
severity.map.high.if.deviceSeverity=7,8
severity.map.medium.if.deviceSeverity=4,5,6
severity.map.low.if.deviceSeverity=2,3
severity.map.verylow.if.deviceSeverity=0,1

#Conditional mappings
conditionalmap.count=1

conditionalmap[0].field=event.fileType
conditionalmap[0].mappings.count=3

conditionalmap[0].mappings[0].values=user
conditionalmap[0].mappings[0].event.destinationUserName=__regexToken
(alert,.?entity name="([^"]+)".*)

conditionalmap[0].mappings[1].values=ip
conditionalmap[0].mappings[1].event.destinationAddress=__
regexTokenAsAddress(alert,.?entity name="([^"]+)".*)

conditionalmap[0].mappings[2].values=machine
conditionalmap[0].mappings[2].event.destinationHostName=__regexToken
(alert,.?entity name="([^"]+)".*)
```

The parser file that is used for entities data is **entities.jsonparser.properties**.

```
trigger.node.location=/data

token.count=12

token[0].name=entityHash
token[0].type=String
token[0].location=entityHash

token[1].name=entityType
token[1].type=String
token[1].location=entityType

token[2].name=entityName
token[2].type=String
token[2].location=entityName

token[3].name=risk
token[3].type=Integer
token[3].location=risk

token[4].name=riskChange
token[4].type=Integer
token[4].location=riskChange

token[5].name=storyCount
token[5].type=Integer
token[5].location=storyCount

token[6].name=lastActivity
token[6].type=String
token[6].location=lastActivity

token[7].name=tags
token[7].type=String
token[7].format=__uri()
token[7].location=tags

token[8].name=otherName
token[8].type=String
token[8].location=../../tags/name

token[9].name=source
token[9].type=String
token[9].location=../source

token[10].name=desc
```

```
token[10].type=String
token[10].location=../tags/description

token[11].name=scrollId
token[11].type=String
token[11].location=/scrollId

#(End Of Token Definitions)

#tokens

event.fileHash=entityHash
event.fileType=entityType
event.fileName=entityName
event.deviceCustomNumber1=risk
event.deviceCustomNumber1Label=__stringConstant("RiskScore")
event.deviceCustomNumber2=riskChange
event.deviceCustomNumber2Label=__stringConstant("RiskChange")
event.deviceCustomString3=lastActivity
event.deviceCustomString3Label=__stringConstant("LastActivity")
#event.deviceCustomDate1=lastActivity
#__parseMutableTimeStampSilently(start)

#tags
#event.destinationUserId=id
event.deviceCustomString5=tags
#event.destinationUserName=otherName
#event.deviceCustomString2=source
#event.message=desc

#nextUrl?
event.deviceCustomString6=scrollId

#Other Mappings
event.name=__stringConstant("Interset Risky User Information")
event.deviceEventClassId=__stringConstant("IRU")
event.deviceVendor=__stringConstant("Micro Focus")
event.deviceProduct=__stringConstant("Interset")
event.deviceSeverity=2

#Agent Severity
severity.map.low.if.deviceSeverity=2
```

Installing and Configuring the FlexConnectors

You need to install two REST FlexConnectors: one to collect and parse the Alerts data, and another to collect and parse the Entities data.

The following section has been verified with the installation of REST FlexConnectors on the Windows 10 platform.

Prerequisites

Complete the following steps before you begin with the REST FlexConnector installation and configuration:

1. Create the `OAuth2.properties` file for using the OAuth2 authentication with Intelligence as follows and save it in the desired location (For example, `C:\Users\Administrator\Desktop\`):

   ```
   client_id= <The client_id value. Click here to identify the client_id
   value.>
   client_secret=<The client_secret value. Click here to identify the
   client_secret value.>
   redirect_uri=http://localhost:8081/oauth2callback
   auth_url=https://<FQDN of ArcSight Platform Virtual IP for HA or single
   master node>/osp/a/default/auth/oauth2/grant
   token_url=https://<FQDN of ArcSight Platform Virtual IP for HA or
   single master node>/osp/a/default/auth/oauth2/grant
   scope=
   timestamp_format_of_api_vendor=
   ```

   To identify the client_id and client_secret values, do the following:

   a. Login to the Management portal as the administrator.

      https://<virtual_FQDN>:5443

   b. Click **CLUSTER** > **Dashboard**. You will be redirected to the **Kubernetes Dashboard**.

   c. Under **Namespace**, search and select the `arcsight-installer-xxxx` namespace.

   d. Under **Config and Storage**, click **Config Maps**.

   e. Click the filter icon, and search for `investigator-default-yaml`.

   f. Open the `investigator-default-yaml` file and look for the client_id and client_secret values in the **OAuth2 Authentication with OSP** section.

2. Do the following to register the callback URL in OSP. The callback URL is the URL where the OSP directs the user after a successful authentication.

a. Launch a terminal session and log in to the node where NFS is present.

b. Change to the following directory:

```
cd <NFS_root_DIRECTORY>/arcsight-volume/sso/default/WEB-
INF/conf/current/default/services/
```

c. Execute the following command to open the authcfg.xml:

```
vi authcfg.xml
```

d. Add `<Url>http://localhost:8081/oauth2callback</Url>` within:

```
<RedirectUrlList>
<Url>${EXTERNAL_URI:http://localhost:9191}/mgmt/callback</Url>
<Url>${OSP_CLIENT_REDIRECT_URI_
1:http://localhost:9191/mgmt/callback}</Url>
<Url>${OSP_CLIENT_REDIRECT_URI_
2:http://localhost:9191/mgmt/callback}</Url>
<!-- For InetSoft Reporting Engine -->
<!-- <Url>${EXTERNAL_
URI:http://localhost:8181}/report/openid/login</Url> -->
<Url>${EXTERNAL_URI}/report/openid/login</Url>
<Url>${EXTERNAL_URI}:443/report/openid/login</Url>
<!-- Endpoint to receive authcode -->
<Url>${EXTERNAL_
URI:http://localhost:9090}/interset/api/actions/login/oauth2/callback<
/Url>
<!-- Endpoint required while logout, this will set in target -->
<Url>${EXTERNAL_URI:http://localhost:3002}/interset/</Url>
<!-- For ArcSight SOAR -->
<Url>${EXTERNAL_URI}/soar/oauth-callback</Url>
</RedirectUrlList>
```

e. Execute the following commands to restart OSP by deleting the fusion-single-sign-on container:

```
kubectl get pods --all-namespaces|grep fusion-single-sign-on
kubectl delete pod <fusion-single-sign-on-xxxxxxxxxx-xxxxx> -n
<arcsight-installer-xxxxx>
```

Install and Configure the REST FlexConnector

To install and configure a REST FlexConnector, see ArcSight FlexConnector REST Developer Guide.

Ensure the following when you install and configure the REST FlexConnector:

- Select **ArcSight FlexConnector REST** as the **Connector Type**.
- When adding the parameters information, specify the following:
  - For the Configuration File field, specify only alerts if the FlexConnector is for collecting and parsing alerts data, else specify only entities if the FlexConnector is for collecting and parsing entities data.
  - For the **Events URL** field, specify **https: //<FQDN of ArcSight Platform Virtual IP for HA or single master node>/interset/api/search/0/alerts?sort=timestamp&sortOrder=desc&riskSort=maximum** if the FlexConnector is for collecting and parsing alerts data, else specify **https: //<FQDN of ArcSight Platform Virtual IP for HA or single master node>/interset/api/search/0/topRisky?count=100** if the FlexConnector is for collecting and parsing entities data.
  - For the **Authentication Type** field, select **OAuth2**.
  - For the **OAuth2 Client Properties File** field, browse to the location where you have created and saved the **OAuth2.properties** file, then select the file.
- Import the OSP Certificate in the REST FlexConnector.
- When configuring the destination, select either ArcSight Manager (encrypted) or Transformation Hub as the destination. For more information, see SmartConnector Installation and User Guide. When adding the parameters information, specify the following if you have selected Transformation Hub as the destination:
  - For the **Content type** field, select **ESM**.
  - For the **Topic (hover for recommendations)** field, specify either **th-binary_esm** or avro topics.
  - For the **For ESM topic, the ESM version** field, select **7.2.x** or above versions.
  - To install the FlexConnector as a standalone application (recommended), select **install as a standalone application**, else to install the FlexConnector as a service, select **install as a service.**

Importing the OSP Certificate in the REST FlexConnector

To import the OSP certificate in the REST FlexConnector:

1. Launch a terminal session and log in to any of the Kubernetes nodes.
2. Execute the following command:

```
kubectl exec -it th-kafka-0 -n <namespace> bash
```

3. Navigate to the following directory where the `issue_ca.crt` certificate file is present. This certificate is the OSP Issuer Certificate (CA).

   ```
   cd /vault-crt/RE
   ```

4. Copy the contents of the `issue_ca.crt` file in a new file, name the file as `issue_ca.cer`, and save it in the desired location (for example, C:\Users\<user_name>\Desktop\).
5. Do the following to import the OSP CA certificate to the FlexConnector truststore cacerts:
   a. Open a command window and navigate to the following location:

      ```
      cd $ARCSIGHT_HOME/current/jre/bin/
      ```

   b. Execute the following command:

      ```
      ./keytool -importcert -file /opt/issue_ca.cer -keystore
      "/root/ArcSightSmartConnectors_
      Alerts/current/jre/lib/security/cacerts" -storepass changeit
      ```

   c. When you run this command, you are prompted to provide your input for the following message: “Trust this certificate [no]:” Specify Yes.

Performing FlexConnector Post-Installation Tasks

After you install and configure the FlexConnector and before you run the FlexConnector, copy the desired JSON parser files in the **ARCSIGHT_HOME\user\agent\flexagent** location.

Installing ESM and Configuring Transformation Hub with ESM

Installing ESM

To install ESM and ArcSight Console to leverage Intelligence entities and alerts information, see Installation Guide for ESM.

Configuring Transformation Hub with ESM

To configure Transformation Hub with ESM, see Configuring ESM as a Transformation Hub Consumer

Sending Data to Transformation Hub From Intelligence

To send data to Transformation Hub from Intelligence, you need to start the FlexConnector. You can run the FlexConnector in standalone mode or as a service, depending on the mode you selected during installation.

Running in Standalone Mode

If you have installed the FlexConnector in the standalone mode, you need to start it manually (periodically or as per your requirement). Also, you need to start the FlexConnector whenever the host on which it is installed is restarted, because the FlexConnector is not automatically active when the host is restarted.

Perform the following steps to start the FlexConnector agent so that it can send the entities and alerts information from Intelligence to the configured topic.

1. Navigate to:

   ```
   cd $ARCSIGHT_HOME\current\bin\
   ```

2. Execute the following command:

   ```
   ./arcsight agents
   ```

Running as a Windows Service

To start or stop the FlexConnector installed as a service on the Windows platform:

1. Right-click **My Computer**, then select **Manage** from the **Context** menu.
2. Expand the **Services and Applications** folder and select **Services**.
3. Right-click the FlexConnector service name and select **Start** to run the FlexConnector or **Stop** to stop the service.

To verify that the FlexConnector service has started, view the following file:

```
$ARCSIGHT_HOME/logs/agent.out.wrapper.log
```

To reconfigure the FlexConnector as a service, run the FlexConnectorConfiguration Wizard again. Open a command window on $ARCSIGHT_HOME/current/bin and run:

```
./runagentsetup
```

Viewing the Intelligence Entities and Alerts Information in the ArcSight (ESM) Console

Perform the following steps to view the Intelligence entities and alerts information in the ArcSight (ESM) Console:

1. Download the **Interset_Sample_Content.arb.zip** file from the Micro Focus Marketplace. and save it in a desired location (For example, C:/Desktop/Interset_Sample_

Content.arb.zip).

2. Extract the downloaded file:

```
unzip Interset_Sample_Content.arb.zip
```

3. Log in to the ArcSight Console.
4. Click the **Packages** tab in the left pane, then click **Import**.
5. Browse to the location where you extracted the **Interset_Sample_Content.arb.zip** file.
6. Click **Install**. The installation process starts.
7. After the installation is successful, click the **Resources** tab in the left pane.
8. Navigate to **Active Channels** > **Shared** > **All Active Channels** > **Interset**.
9. Double-click **Interset** or **Interset Anomalies** to view the Intelligence entities and alerts information.
10. Navigate to **Dashboards** > **Shared** > **All Dashboards** > **Interset**.
11. Double-click **Interset Overview** to view a summary of the Intelligence entities and alerts information.

Integrating SOAR with ESM

SOAR integrates with ESM to log and forward detailed reporting on every single incident to facilitate prioritization and investigation of alerts as well as the remediation of incidents.

SOAR ingests correlated events from ESM and converts them into an alert. When an alert is generated, a new incident is created on SOAR's Incident Management Service Desk. Analyst can then investigate the incident and take remedial actions.

The ESM and SOAR integrations presents following capabilities to:

- Ingest Correlated Alerts
- Retrieve Base Events
- Create Case
- Update Case
- Search Cases
- Get Case Details
- Query Active List
- Add Entries to Active List
- Delete Entries from Active List

The bidirectional integration of ESM and SOAR requires configuration at both the platforms.

Understanding the Prerequisites for ESM and SOAR Integration

Complete the following steps before you begin the ESM and SOAR integration:

- Download the ESM-SOAR integration content from Marketplace.

  Note: You need to download the ESM-SOAR integration content only if you are using ESM 7.5. For ESM 7.6 you do not need to import arb file for ESM-SOAR integration.

- Import the integration content to the ArcSight Console.

  Note: You need to follow this step only if you are using ESM 7.5. For ESM 7.6 you do not need to import the integration content to ArcSight Console.

- Allow network traffic from ESM to SOAR towards port 32200/TCP. The Arcsight SOAR listener for correlated event data (alerts) is accessible from this port.

  Run the following command on ESM to verify network traffic from ESM to SOAR:

  ```
  openssl s_client -connect <CDF HOST>:32200
  ```

- Open REST API port 8443/TCP at ESM to allow HTTPS traffic. SOAR connects with the ESM REST API on this port.

  Run the following command on SOAR to verify HTTPS traffic:

  ```
  openssl s_client -connect <Address of the ESM Manager>:8443
  ```

- Configure a SOAR user account to connect with the ESM API.
- Set the parameter **ArcSightListenerProtocol** in SOAR at **Configuration** > **Parameters** as **tls**.
- Enable the parameter **ArcSightListenerEnabled** in SOAR at **Configuration** > **Parameters** before configuring the forwarding destination on the connector.

  Note: If the parameter **ArcSightListenerEnabled** is not enabled, an error message is displayed as connection refused.

- Install a forwarding connector on ESM and configure it to forward events from ESM to SOAR.

To install a forwarding connector:

1. **Create a forwarding connector**: To create a forwarding connector, see Forwarding Correlation Events. For this example, you can create the forwarding connector for ESM and SOAR integration with the following values:

- User ID: forwardSOAR
- User Type: Forwarding Connector

2. **Install and configure the forwarding connector package**: Install the forwarding connector package on ESM. Then complete the following steps for configuration:
   To add the local connector certificate:
   a. Run the following command to download the certificate from CDF:

```
CDF_APISERVER=$(kubectl get pods -n core -o custom-
columns=":metadata.name"| grep cdf-apiserver)
CA_CHAIN=$(kubectl exec -n core ${CDF_APISERVER} -c cdf-apiserver --
bash -c "vault read -tls-skip-verify -field=certificate RE/cert/ca_
chain"); if [ -n "$CA_CHAIN" ]; then echo "$CA_CHAIN" > /tmp/cdf-
soar.cert; else kubectl exec -n core ${CDF_APISERVER} -c cdf-apiserver
-- bash -c "vault read -tls-skip-verify -field=certificate RE/cert/ca"
> /tmp/cdf-soar.cert; fi
```

   b. Copy the certificate to the following path:

```
/opt/arcsight/MicroFocus_
ArcsightSmartConnectors/SuperConnector/current/jre/lib/security
```

   c. Navigate to the following keytool path:

```
cd /opt/arcsight/MicroFocus_
ArcsightSmartConnectors/SuperConnector/current/jre/bin/
```

   d. Run the following command to import the connector certificate:

```
./keytool -importcert -file ../lib/security/cdf-soar.cert -keystore
../lib/security/cacerts -alias "CDF-cert"
```

   Note: Keystore password is **changeit**.

   e. In the **Connector Setup Wizard**, select the **Add a Connector** option, then the **ArcSight Forwarding Connector (Enhanced)** option to configure the connector as a forwarding connector.
   f. Enter the parameter details as follows:
      - **ArcSight Source Manager Host Name[localhost]**: <Specify local host IP>
      - **ArcSight Source Manager Port**: 8443
      - **ArcSight Source Manager User Name**: <Specify the user name that you have created for ESM>

- **ArcSight Source Manager Password**: <Specify the password that you have created>

g. Select **Yes**, if the values are correct.

To configure the forwarding connector for forwarding events from ESM to SOAR:

a. To set up the ArcSight Agent, run the following command:

```
cd /opt/arcsight/MicroFocus_
ArcsightSmartConnectors/SuperConnector/current/bin
```

```
./runagentsetup.sh
```

b. In the **Connector Setup Wizard**, select **Select the type of destination**, and then select **CEF Syslog**.

c. Specify the parameter details as follows:

- **IP/Host**: <Specify the ArcSight Platform FQDN corresponding to the Virtual IP address provided during installation for HA or, for a single-master installation, the IP address of the master node>
- **Port**: SOAR 32200
- **Protocol**: TLS
- **Forwarder**: False

> Note: If ArcSight Platform is installed on AWS and ESM capability is not installed in the same VPC, you must use a network load balancer (NLB) instead of an application load balancer (ALB) on AWS since ALB does not support TCP communication channel required for sending correlated events from ESM to SOAR.

d. Select **Yes**, if the values are correct.

After you have met the prerequisites, complete the integration in SOAR.

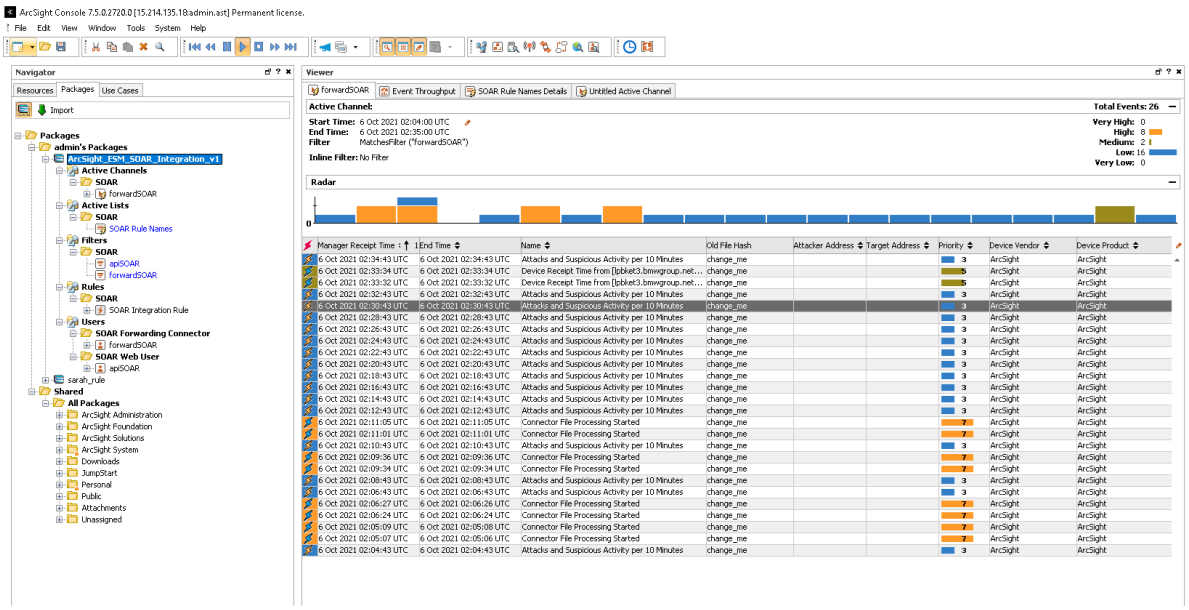
Importing the ESM-SOAR Integration Content

After you download the ESM-SOAR integration content from Marketplace, import it to the ArcSight Console and configure it.

> **Note:** You need to download the ESM-SOAR integration content and import it to ArcSight Console and configure it, only if you are using ESM 7.5. For ESM 7.6 you do not need to import arb file for ESM-SOAR integration.

1. Import the integration content to the ArcSight Console.

   The following shows the content imported to the console:



2. Reset the password for the SOAR Forwarding Connector user, **forwardSOAR**.
3. Reset the password for the SOAR Web user, **apiSOAR**.
4. Add the correlation rule names that you want to forward from ESM to SOAR to the **SOAR Rule Names** active list.
5. The integration content adds **change me** as the default value for the **Old File Hash** field.

   This value is used during the process of adding ESM as an alert source for SOAR. The default value of the **Old file Hash** field is specified on the **Key** textbox in the **Alert Source Editor**.
6. Open **ACL Editor** for **apiSOAR** user on ESM console and add read and write permissions for all active lists for this user. Now you can access all the active lists on ESM from SOAR side.

> **Note:** You can change the value of the **Old File Hash** field on the **SOAR Integration Rule** action tab in the ArcSight Console.

Completing the Integration in SOAR

The ESM and SOAR integration requires some configuration in SOAR. You must add the credentials for the Web User account that you created in ESM to SOAR. This user account is used to read, write, and access the active list in ESM. This account is also responsible for accessing all of the required events, including the base events in ESM. To listen to the events, you must configure ESM as an alert source in SOAR. After you configure ESM as an alert source, SOAR can pull the events from ESM and convert them into alerts for investigation purposes.

Adding Credentials

To support ESM and SOAR integration, you must add the credentials for the ESM SOAR Web user to SOAR. SOAR uses this account to fetch and update events as well as to invoke other supported actions.

To add the credentials to SOAR:

1. In SOAR, navigate to **Configuration** > **Credentials**.
2. Click **Create Credential** to view the **Credential Editor** window.
3. Specify the following values in the **Credential Editor** window:

   For Internal Credential:

   - **Type**: Internal credential
   - **Name**: <Display name of credential set>

     For example: ArcSight ESM Credentials
   - **Username**: <User name created for the SOAR Web user in ESM>
   - **ESM Password**: <Password of the SOAR Web user>
   - **Private Key**: <Empty>

Configuring ESM as an Alert Source

The active list in ESM has correlated events that ESM passes to SOAR. SOAR converts these events to alerts and performs investigation and response procedures. To receive alerts in SOAR, you must configure ESM as an alert source to SOAR.

To configure ESM as an alert source to SOAR :

1. In SOAR, navigate to **Configuration** > **Alert Source**.
2. Click **Create Alert Source Configuration** and specify the following values in the **Create Alert Source Configuration** window:
   - **Name**:< Display name of the ESM alert source in SOAR >
   - **Type**: Micro Focus ArcSight ESM
   - **Address**: <Address of the ESM Manager>

     For example, https://192.168.5.5:8443.
   - **Key**: <Specify the name of the key from the ESM pre-persistence rule>
   - **Alert Severity**: <Specify the alert severity values mapping, with SOAR incident severity>
   - **Configuration**: Specify the following parameters:

Parameter Name	Parameter Description	Parameter Usage
CEF field **[severity.field]**	Used as severity value when mapping severity value to SOAR incident severity. You can set this parameter for priority, severity, flexString1 and flexNumber 1	severity.field=priority
CEF-extension **[severity.field]**	Used as rule name value.	
Scope fields: **[src]**	a. The value of scope field is extracted from correlated event. src:NETWORK_ ADDRESS:OFFENDER, dst:NETWORK_ ADDRESS:IMPACT, request:URL:OFFENDER fields are always extracted by default. b. This parameter can also specify additional fields to be extracted:	a. (field1:CATEGORY:ROLE, (field2:CATEGORY:ROLE, ...) CATEGORY is any EMAIL_ADDRESS, HASH, HOST, MAC_ADDRESS, NETWORK_ ADDRESS, COMPUTER_NAME, UNKNOWN, URL, USERNAME, PROCESS ROLE is any OFFENDER, IMPACT, RELATED b. correlated.scope=s_ user:USERNAME:OFFENDER, dvc:NETWORK_ADDRESS:RELATED correlated.scope=src:NETWORK_ ADDRESS:OFFENDER, dst:NETWORK_ ADDRESS:IMPACT, request:URL:OFFENDER
Additional Scope Field: **[baseevent.scope]**	These values are extracted from base events (field1:CATEGORY:ROLE) and use JSON pointer notation. See the correlated.scope property for Category and Role values details. This parameter can specify additional fields to be extracted, and will not override the default behavior.	Example: baseevent.scope=/device/address:NETWORK_ADDRESS:RELATED # baseevent.scope=
[cache.reusing.duration]	Used to configure how far (in minutes) into the past this enrichment is checked.	cache.reusing.duration=20
enable/disable **[enable.baseevent.activity]**	Used to enable/disable base events activity in the incident timeline.	enable.baseevent.activity=false

> **Note**: To collect information for MITRE Attack, you can add `mitre.id.field` in the configuration pane with the default value as `cs6`.

3. Click **Save** to complete the ESM and SOAR integration.
4. Click **Test** to test the integration. A **Test Alert Source** pop-up is displayed to confirm that you have entered the valid credentials and address.
5. Navigate to **Configuration** > **Parameters** and set the value of following parameters:
   - **ArcSightListenerEnabled** to **true**.
   - **ArcSightListenerProtocol** to **tls**.

> **Note**: Ensure that the **Forwarding Connector** for the SOAR connection protocol is set to **TLS**.

Configuring ESM as an Integration

ESM must be configured in SOAR as an integration for executing SOAR actions and enrichment capabilities.

To configure ESM as an integration:

1. In SOAR, navigate to **Configuration** > **Integrations**.
2. Click **Create Integration** to view the **Configuration** window.
3. Specify the following values in the **Configuration** window:
   - **Name**: <Display name of the ESM integration in SOAR>
   - **Type**: Micro Focus ArcSight ESM
   - **Address**: <Address of the ESM Manager>

     For example, https://192.168.5.5:8443
   - **Configuration**: #proxy.id=5422
   - **Credential**: <Name of the credential set created>

     For example, ArcSight ESM Credentials
   - **Trust Invalid SSL Certificates**: <Select this option if the server certificate is self-signed or not recognized by the browsers>
   - **Require Approval From**: <Select users that can provide approval before executing actions on this integration>
   - **Notify**: <Select users to be notified when SOAR performs an action on this integration>

4. Click **Save** to complete the integration.
5. Click **Test** to test the integration. A **Test Alert Source** pop-up is displayed to confirm that you have entered the valid credentials and address.

Tuning ESM and SOAR Integration

The ESM and SOAR integration can be customized as per your requirements. Following parameter values can be tuned to suit your environment:

Consult with ArcSight SOAR Field Engineering Team if tuning is required.

Parameter Name	Parameter Description	Default Value
ArcSightAutoEnrichEnabled	Enable ArcSight auto-enrichment with base-event data	False
ArcSightListenerEnabled	Enable Arcsight Listener	False
ArcSightListenerKeyField	ArcSight listener key field for alert source identification	oldFileHash
ArcSightListenerProtocol	ArcSight listener protocol	tcp
ArcSightListenerThreadPoolCoreSize	ArcSight listener thread pool core pool size (0 = unlimited)	0
ArcSightListenerThreadPoolKeepAlive	ArcSight listener thread pool keep-alive seconds (ignored if core pool size = 0)	60
ArcSightListenerThreadPoolMaxSize	ArcSight listener thread pool maximum size (ignored if core pool size = 0)	20
ArcSightListenerThreadPoolQueueCapacity	ArcSight listener thread pool queue capacity (ignored if core pool size = 0)	1000

To enable receiving MITRE Attack for specific incidents, you can tune following parameters:

Parameter	Description
MITREAttackControllerFixedDelay	The frequency for SOAR to update the MITRE data, the default value is 86400 (in seconds).
MITREAttackUrl	Represents the source from where to fetch the MITRE details.
MITREAttackControllerProxyIntegrationId	Supports proxy integration, the default value is -1, implying no proxy usage.

Integrating SOAR with Intelligence

Micro Focus ArcSight Intelligence uses unsupervised machine learning to calculate probabilistic risk assessments based on behavioral analytics from millions of events, ultimately generating a short list of high value targets to allow security teams to detect, investigate, and respond to threats that might hide in the enterprise before any case occurs.

SOAR has the following integration capabilities with Intelligence:

- Ingest Anomalies as Alert
- Get Entity Details

Use Cases

Use Case #1: Prioritizing Cases

SOAR is integrated with Intelligence, to help prioritization and investigation of cases as well as remediation of cases. When an alert is received, a new case is created in the Case Management Service Desk of SOAR. SOAR then automatically checks the risk scores of entities and prioritizes the case based on these risk scores. Get Entity Details enrichment results return latest 1000 records in maximum.

Use Case #2: Mitigating Account Compromise

SOAR ingests anomaly data from Intelligence and creates case ticket in the Case Management Service Desk. With its broad integration portfolio, orchestration, and automation capabilities, SOAR investigates, ascertains the case, and takes necessary actions to prevent the compromise.

The bidirectional integration of Intelligence and SOAR requires configuration at both the capabilities.

Configuration

Prerequisites

- SOAR connects to Micro Focus ArcSight Intelligence API via HTTPS. By default, the interface works on 443/tcp port. Make sure that you have access permission to this port.
- A user account for SOAR to connect to Intelligence API.

Configuring ArcSight Intelligence

No specific configuration is needed on Intelligence.

Configuring SOAR

1. Click **Configuration** > **Credentials** > **Create Credential**.
2. Specify the following parameter values in the **Credential Editor**:

Parameter	Value
Type	Internal Credential
Name	Display name of credential set (For example, Intelligence Credentials)
Username	Name of the SOAR user created on Intelligence.
Password	Password of the SOAR user created on Intelligence.
Private Key	Leave the field blank.

3. If **use.basic.authentication** configuration parameter value is **False**, then get the **Client id** and **Client secret** from Intelligence to ensure that the Intelligence Alert Source and Intelligence Integration work as expected.

   Note: The default value of **use.basic.authentication** parameter is **False**.

   To get Client ID and Client Secret for Micro Focus Intelligence open a command prompt:

   a. Specify the name of the server on which Intelligence works.

   b. Run the following command to get **Client ID** and **Client Secret** from Intelligence:

   ```
   osp-client-id and osp-client-secret : kubectl get secret osp-secret -n
   arcsight-installer-tyoib -o yaml
   ```

   The output is displayed in the following format:

   ```
   data:
   osp-client-id: NTZjODkyYWE3NDMzZThiOTYzZGVkMjE5ZGIzODU3ZDg=
   osp-client-secret:

   ZjRiZDUzODBiZjQ2NTY5MWQ4NDAzMTFhZTJmMjY1ZGJlZGRjOWU0NDhlZmE3ZDhjN2Q5Yz
   JlY2VjMDkzMmExNw==
   ```

   c. Run the following command to decode the **Client ID** and **Client Secret**:

   ```
   echo 'NTZjODkyYWE3NDMzZThiOTYzZGVkMjE5ZGIzODU3ZDg=' | base64 --decode
   echo
   'ZjRiZDUzODBiZjQ2NTY5MWQ4NDAzMTFhZTJmMjY1ZGJlZGRjOWU0NDhlZmE3ZDhjN2Q5Y
   zJlY2VjMDkzMmExNw==' | base64 --decode
   ```

   d. Run the following command to add the **Client ID** and **Client Secret** to the Alert Source / Integration configuration on Intelligence.

```
# Client id that defined in OSP
client.id=id
# Client secret that defined in OSP
client.secret=secret
```

Configuring Intelligence as an Alert Source

1. Click **Configuration** > **Integrations** > **Create Alert Source**.
2. Specify the following parameter values in the **Configuration** form:

Parameter	Value
Name	Display name of Intelligence Alert Source on SOAR.
Type	Micro Focus ArcSight Intelligence.
Address	Address of the Intelligence server (the format must be https://172.16.11.9).

Configuration	Specify the following configuration parameters:
	`tenant.id=`
	`# ID of the proxy integration to use when connecting to current source.`
	`# If not provided, ArcSight SOAR will try to use a direct connection.`
	`#proxy.id=123`
	`# configure how far (in minutes) into the past this enrichment will look.`
	`#cache.reusing.duration=20`
	`# Base path of the Micro Focus Intelligence. SOAR adds it to end of the URL to access Micro Focus Intelligence.`
	`interset.context.path=/interset`
	`# Client id that defined in OSP`
	`client.id=id`
	Note: By default,Intelligence uses 0 for tenant id. However, Intelligence - SOAR integration supports different tenants.

Credential	Name of the credential set you have created (For example, Micro Focus ArcSight Credentials)
Trust Invalid SSL Certificates	Select this if Web UI's certificate is self-signed or is not recognized by browsers
Visible Alert Fields	You might define the alarm fields that will be displayed on the Case Management Service Desk

3. Click **Save** to complete the integration.
4. Click **Test** to test the integration.

Configuring Intelligence as Integration

1. Click **Configuration** > **Integrations** > **Create Integration**.
2. Specify the following parameter values in the **Configuration** form:

Parameter	Value
Name	Display name of Intelligence integration on SOAR.
Type	Micro Focus ArcSight Intelligence.
Address	Address of the Intelligence server (the format must be https://172.16.11.9).

Configuration	Specify the following configuration parameters:
	`tenant.id=`
	`# ID of the proxy integration to use when connecting to current source.`
	`# If not provided, ArcSight SOAR will try to use a direct connection.`
	`#proxy.id=123`
	`# configure how far (in minutes) into the past this enrichment will look.`
	`#cache.reusing.duration=20`
	`# Base path of the Micro Focus Intelligence. SOAR adds it to end of the URL to access Micro Focus Intelligence.`
	`interset.context.path=/interset`
	`# Client id that defined in OSP`
	`client.id=id`

Credential	Name of the credential set you have created (For example, Micro Focus ArcSight Credentials)
Trust Invalid SSL Certificates	Select this if Web UI's certificate is self-signed or is not recognized by browsers
Require Approval From	Select user(s) from the list to request for approval before executing actions on this integration. Because SOAR only executes enrichments on Intelligence, leave it empty.
Notify	Select user(s) from the list to notify when SOAR performs an action on this integration. Beacuse SOAR only executes enrichments on Intelligence, leave it empty.

3. Click **Save** to complete the integration.
4. Click **Test** to test the integration.

Additional Notes

- The following configuration parameters can be used for fine tuning the integration. You must consult SOAR field engineering team before editing them:

- MicroFocusIntelligenceListenerMaxRetrySeconds Micro Focus Interset listener queue max message retry in seconds 1800
- MicroFocusIntelligenceListenerQueueConcurrency Upper limit of Micro Focus Interset Listener consumer thread count 3
- MicroFocusIntelligenceSyncPeriod Period in seconds to sync Micro Focus Interset anomalies 60

Capabilities

1. **Get Details**

   Enrichment capability to get the risk score of a given entity and related alert details.

   The following table presents the **Get Details** capability details:

Input Parameter	Description	Type	Scope Rescticted (Yes/No)	Required (Yes/No)
Integration	Name of the third party integration.	Integration	N/A	Yes
Entity	Entity to be queried on ArcSight Intelligence.	Network Address Host File Name URL	Yes	Yes
Do not use cache	SOAR does not use cached results if this box is checked.	Checkbox	N/A	No

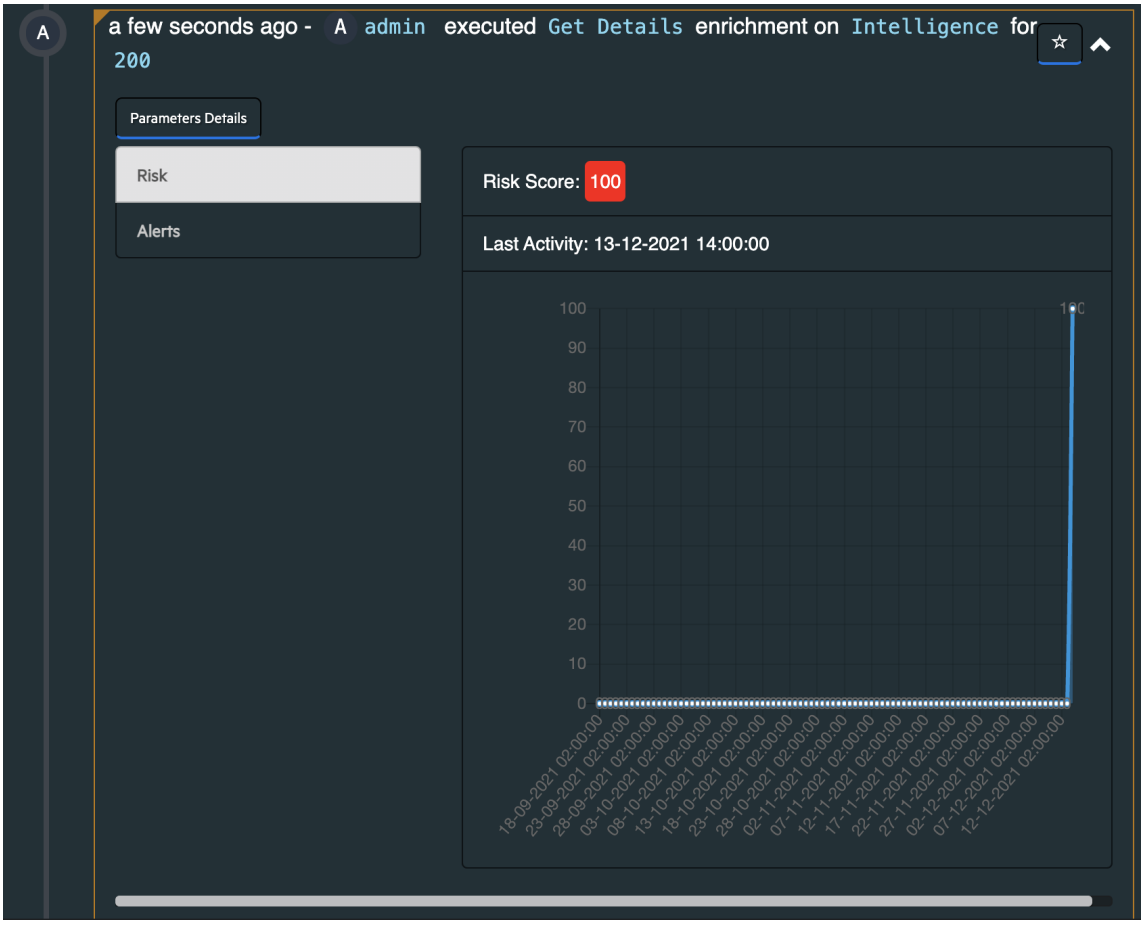
   Output:

   Case Scope:

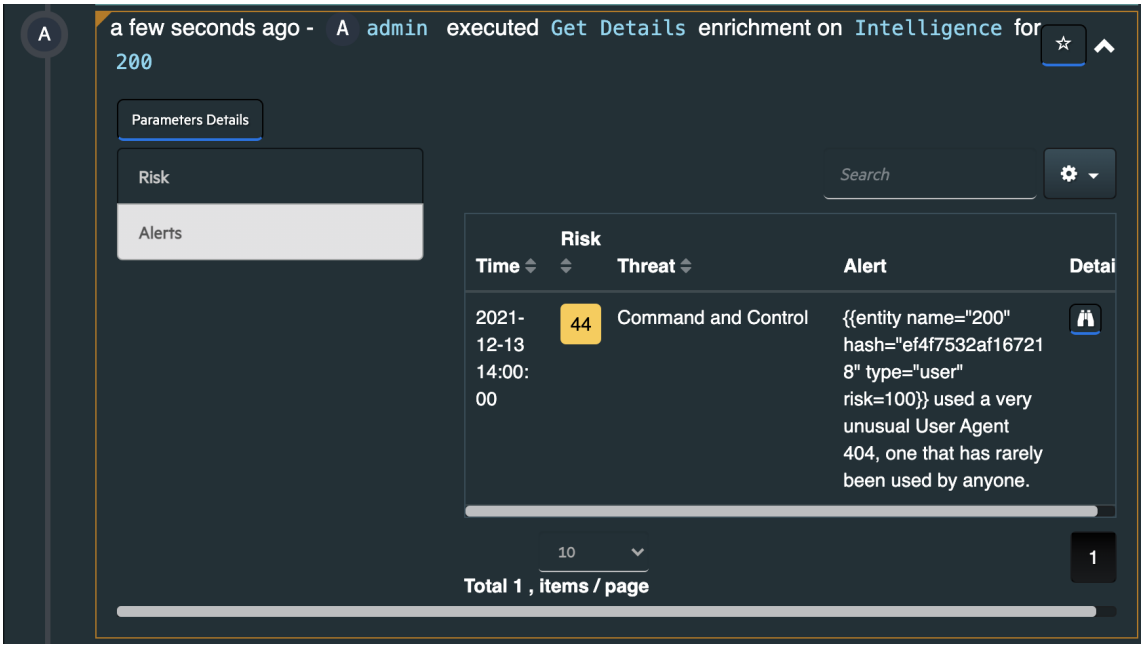
Action	Type	Category/ Value
Add Scope Item Property	Integer	Micro Focus Intelligence Entity Risk
Add Scope Item Property	TEXT	Micro Focus Intelligence Entity Hash
Add Scope Item Property	TEXT	Micro Focus Intelligence Entity Type

   Human Readable Output:

a. Risk tab:



b. Alerts tab:



Chapter 8: Upgrading Your Environment

This section provides information about upgrading your environment. Several options for upgrading your environment are available.

Micro Focus does not recommend that ArcSight Intelligence customers attempt to upgrade without consulting with Micro Focus, as upgrading at this time will result in loss of baseline data. We are aware that this is inconvenient for customers, and we are currently working on addressing this limitation in an future release.

Upgrading to 22.1.0

You can upgrade to ArcSight Platform 22.1.0 from 21.1.x. Select the upgrade method for your environment:

For more information about this release and which files to download, see the *Release Notes for ArcSight Platform 22.1*.

Upgrading an On-premises Deployment

To ensure a successful upgrade, be sure to follow the tasks in "Checklist: Upgrading Your Environment" below.

Micro Focus does not recommend that ArcSight Intelligence customers attempt to upgrade without consulting with Micro Focus as upgrading at this time will result in loss of baseline data. We are aware that this is inconvenient for customers, and we are currently working on addressing this limitation in an future release.

Checklist: Upgrading Your Environment

To ensure a successful on-premises upgrade, complete the following tasks in the listed order:

This release significantly changes the ArcSight Database such that you cannot upgrade the database. It must be installed as new. However, this release does allow you to deploy or upgrade Recon and Intelligence in your environment, as well as install either capability for the first time.

	Task	See
❑	1. Download the installation packages.	"Downloading the Installation Packages for an On-Premises Deployment" below
❑	2. (Conditional) To deploy Intelligence or Recon, install the ArcSight Database.	"Installing the Database" on page 81
❑	3. (Conditional) To prevent both the original and newly installed databases from ingesting duplicate events, stop data ingestion on the original database.	"Checklist: Stopping Event Ingestion" on page 564
❑	4. Upgrade the CDF infrastructure.	"Preparing the Upgrade Manager " on page 520
❑	5. (Conditional) If you have previously deployed SOAR and plan to upgrade, delete the old resource definitions.	"Preparing for SOAR 3.2" on page 523
❑	6. Upgrade the deployed capabilities.	"Upgrading Deployed Capabilities" on page 523
❑	7. Complete post-upgrade tasks.	Completing Post Upgrade Tasks
❑	8. Complete the post-upgrade procedure.	"Restarting the Event Consumers" on page 570
❑	9. For a manual upgrade or one using autoUpgrade, apply the hotfix to remediate the log4j vulnerability.	Applying the log4j Hotfix
❑	10. Apply security patches and latest updates to the deployed capabilities and the ArcSight Database.	"Upgrading to 22.1.2" on page 572
❑	11. Upgrade ESM.	"Upgrading ESM" on page 530

Downloading the Installation Packages for an On-Premises Deployment

Use this procedure to download the packages for:

- **Installation:** Follow the "Checklist: Creating an On-premises Deployment" on page 55 to ensure a successful installation.
- **Upgrade**: Follow the "Checklist: Upgrading Your Environment" on page 511 to ensure a successful upgrade.

1. Launch a terminal session and log in to the primary master node as `root`.

   If you elect to install as a sudo user, log in to the primary master node as the non-root user.

2. To identify and access the files to download into a directory, see Downloading and Installing the ArcSight Platform Capabilities in the *Release Notes for ArcSight Platform*.
3. Unzip `arcsight-platform-installer-x.x.x.x.zip` into a directory, which we will refer toas *`{unzipped-installer-dir}`*.

   Do not unzip under `/root` or any sub-directory of `/root`..

4. Move the ArcSight Metadata file into the *`{unzipped-installer-dir}`*`/metadata/` directory.

   Do not untar the file. The filename must have the prefix `arcsight-installer-metadata`. Also, do not move signature files as it might cause warnings and errors from the installer script. Therefore, only copy the tar files you need based on what you are deploying.

5. For each ArcSight product to install and upgrade, move the corresponding image tar file into the *`{unzipped-installer-dir}`*`/images/` directory.

   Do not untar the file. Also, do not move signature files as it might cause warnings and errors from the installer script. Therefore, only copy the tar files you need based on what you are deploying.

   For example, if you deploy Transformation Hub, Fusion, and Recon, the image tar filename for each product is as follows:

Transformation Hub	`transformationhub-x.x.x.x.tar`
Fusion	`fusion-x.x.x.x.tar`
Recon	`recon-x.x.x.x.tar`

Stopping Event Ingestion for the ArcSight Database

If your environment has the ArcSight Database deployed or you are replacing an existing instance of the database, you must perform this group of procedures before upgrading the database or installing the new instance. To prevent the upgraded or replaced database from ingesting duplicate data, this process stops some services and creates a script for assessing the status of event processing. For example, you need to stop the Kafka scheduler and pause the database's watchdog service to prevent it from restarting the Kafka scheduler in the database.

The database ingests events using the Kafka scheduler. You need to know whether the database that you want to upgrade or replace has the most current events or is still processing a backlog of events so that you can stop all the components once they have read the same messages. To identify whether the scheduler is current, you must review the **offset values** that are based on the number of partitions in the *mf-event-avro-enriched* topic. The offset indicates how far apart the deployed capabilities and the database are when reading from the topic.

- "Checklist: Stopping Event Ingestion" on page 564
- "Stopping the Database Event Consumer" below
- "Stopping the Intelligence Event Consumers" on the next page
- "Preventing the Database from Consuming Duplicate Events" on page 517

Checklist: Stopping Event Ingestion

Before upgrading your environment, complete the following procedures in the listed order:

	Task	See
❑	1. Stop the database event consumer (the Kafka Scheduler) so you can record its offsets	"Stopping the Database Event Consumer" below
❑	2. (Conditional) If you have Intelligence deployed, monitor the Intelligence event consumers to stop them when they catch up to the database event consumer	"Stopping the Intelligence Event Consumers" on the next page
❑	3. Prevent the database from consuming duplicate events	"Preventing the Database from Consuming Duplicate Events" on page 517

Stopping the Database Event Consumer

You must stop the database event consumer (Kafka Scheduler) so that you can record its offsets for use later in this process.

1. Log in to the primary node of the ArcSight Database.
2. To stop the database watchdog service, run the following command:

```
crontab -l | sed '/^[^#].*scripts.watchdog/s/^/#/' | crontab -
```

3. To stop the Kafka scheduler for the database, run the following command:

```
/opt/arcsight-db-tools/kafka_scheduler stop
```

4. To record the offset values for each partition into a `db_scheduler_offset.csv` file, run the following command:

```
/opt/vertica/bin/vsql -U {dbadmin_user_name} -w {dbadmin_user_password} -
F ',' -P footer=off -Aqc "select source_partition as partition,
LastEndOffset as offset, 'DB' as ConsumerType, LastFrame as time from
(select source_name, source_cluster, source_partition, max(end_offset) as
LastEndOffset, max(frame_start) as LastFrame from default_secops_adm_
scheduler.stream_microbatch_history group by 1,2,3 order by 1 asc, 2 asc,
3 asc) as A order by 1 asc;" > db_scheduler_offset.csv
```

You will use this `db_scheduler_offset.csv` file to record the offset values for the *mf-event-avro-enriched* topic in a future step.

5. Copy the `db_scheduler_offset.csv` file to a Transformation Hub node.

Stopping the Intelligence Event Consumers

Applies only when you have Intelligence in your environment

You will need to stop the Intelligence consumers after they have read past the offsets that the database has read. Then you must remove older events so the database does not receive duplicates after the upgrade. Although this procedure prevents the database from receiving duplicate events, Intelligence might receive duplicate events but then it would automatically de-duplicate them.

Complete the following steps to prevent the loss of events during the upgrade. This procedure needs the `db_scheduler_offset.csv` file that you created in Step 6.

1. Log in to a Transformation Hub node. Ensure that you have copied the `db_scheduler_offset.csv` script to this node.
2. Create an executable shell script file, named `compare-current-offsets.sh`, with the following content:

```
#!/bin/bash
dof="$1"
if [[ -z $dof || ! -f "$dof" ]]; then
     echo "Database offsets not found: $vof"
     exit 1
fi
NS=$(kubectl get ns | awk '/^arcsight-installer/{print $1}')
t="mf-event-avro-enriched"
ig="interset-logstash-es"
ic=Intelligence
tmp=$(mktemp -u --tmpdir offsets.XXXXX)
printf "\nGetting current %s offsets for Intelligence\n\n" $t
kubectl exec -it -n "$NS" th-kafka-0 -- bash -c "kafka-consumer-groups --
bootstrap-server localhost:9092 --offsets --describe --all-groups" | grep
"$t\s" | sed -re "s/^${ig}/$ic/" | awk '{c=$1; p=$3; o=$4; printf(
```

```
"%s,%s,%s\n", p, o, c );}' | sort -n > "$tmp"
partns=$( cut -d',' -f1 "$tmp" | sort -u)
for p in $partns ; do for c in $ic ; do
     vo=$( awk -v FS="," "/^$p,.*,DB/"'{print $2}' $dof)
     if [[ -n $vo ]]; then
         desc="no $c offset data"
         co=$(awk -v FS="," "/^$p,.*,${c}$/"'{print $2}' "$tmp")
         if [[ -n $co && $co =~ ^[0-9]+$ ]]; then
               lead=$(( co - vo ))
               desc="$c is $(( - lead )) messages behind the db"
               if [[ $lead -ge 0 ]]; then
                       desc="$c is caught up to the db and is $lead messages ahead"
               fi
       fi
       printf "partition %d: %s\n" "$p" "$desc"
     fi
done ; done
rm -f $tmp
```

3. Using the CSV file and script that you created previously, run the following command:

   ```
   ./compare-current-offsets.sh db_scheduler_offset.csv
   ```

   This command generates output that shows the offsets between the database and the data consumers for all partitions:

   ```
   Getting current offsets for Intelligence
   partition 0: Intelligence is caught up to the db and is 100231 messages ahead
   partition 1: Intelligence is caught up to the db and is 100123 messages ahead
   partition 2: Intelligence is 1002 messages behind the db
   partition 3: Intelligence is caught up to the db and is 100141 messages ahead
   partition 4: Intelligence is caught up to the db and is 100343 messages ahead
   partition 5: Intelligence is caught up to the db and is 100999 messages ahead
   ```

4. Monitor the output from the command, which can indicate the following scenarios:
   - Whether a consumer has caught up to the database offset. For example, Intelligence is reading the *mf-event-avro-enriched* topic events ahead of the database:

     ```
     partition 0: Intelligence is caught up to the db and is 9995 messages ahead
     ```

- Whether the database offset is ahead of the consumers reading the *mf-event-avro-enriched* topic. For example, Intelligence has not caught up to the database:

  ```
  partition 0: Intelligence is 1002 messages behind the db
  ```

- Whether a consumer is not reading the *mf-event-avro-enriched* topic. For example, when this occurs, the output indicates **no offset data**:

  ```
  partition 0: no Intelligence offset data
  ```

If a capability is not reading messages from mf-event-avro-enriched, you do not need to compare the offsets.

5. When Intelligence has caught up to the database, complete the following steps to stop the Intelligence consumers:
   a. Log in the master node of the cluster.
   b. Run the following command:

   ```
   NS=$(kubectl get namespaces | awk '/arcsight-installer/{print $1}')
   kubectl scale statefulset interset-logstash -n $NS  --replicas=0
   ```

   c. Run the following command periodically until you see a response indicating that none of the instances are ready:

   ```
   kubectl -n $NS get sts interset-logstash
   ```

   For example, you would want to see a response that indicates 0 out of 3 instances are ready:

   ```
   NAME                 READY   AGE
   interset-logstash    0/3     21d
   ```

   d. In the **READY** column, note the second number, which represents the number of instances. For example, 3. You will need this number for the ""Restarting the Event Consumers" on page 570" procedure.

Preventing the Database from Consuming Duplicate Events

To prevent the database from consuming duplicate events, you must reset the *mf-event-avro-enriched* topic so that it does not have events that the database has already consumed.

1. Log in to the Transformation Hub node.
2. To reset the offset record for the *mf-event-avro-enriched* topic, run the following commands:

   Use the offset value stored in the `db_scheduler_offset.csv` file.

```
NS=$(kubectl get namespaces | awk '/arcsight/{print $1}')
```

```
in=db_scheduler_offset.csv; n=$(grep -c "^[0-9]" $in); fmt='{"topic":
"mf-event-avro-enriched", "partition": %s, "offset": %s}%s\n'; awk -v
FS="," -v n="$n" -v f="$fmt" 'BEGIN{print "{\"partitions\": [ "; c=","}
/^[0-9]/{if(++r==n){c=""};o=$2+1;printf(f, $1, o, c)} END{print "],
\"version\":1 }" }' $in | tee /tmp/offsets.json
```

```
kubectl cp /tmp/offsets.json $NS/th-kafka-0:/tmp/offsets.json; kubectl
exec -n $NS th-kafka-0 -- kafka-delete-records --bootstrap-server
localhost:9092 --offset-json-file /tmp/offsets.json
```

The output from the first two commands should be similar to the following content:

```
{"partitions": [
{"topic": "mf-event-avro-enriched", "partition": 0, "offset": 123000},
{"topic": "mf-event-avro-enriched", "partition": 1, "offset": 123400},
{"topic": "mf-event-avro-enriched", "partition": 2, "offset": 123500},
{"topic": "mf-event-avro-enriched", "partition": 3, "offset": 123600},
{"topic": "mf-event-avro-enriched", "partition": 4, "offset": 123700},
{"topic": "mf-event-avro-enriched", "partition": 5, "offset": 123800},
], "version":1 }
```

3. (Optional) To verify that the offsets have been correctly updated, run the following command:

```
kubectl exec -n $NS th-kafka-0 -- kafka-run-class
kafka.tools.GetOffsetShell --broker-list localhost:9092 --topic mf-event-
avro-enriched  --time -2
```

The output for this command should be similar to the output described Step 2.

4. Continue to upgrading CDF.

Upgrading CDF

Micro Focus does not recommend that ArcSight Intelligence customers attempt to upgrade without consulting with Micro Focus as upgrading at this time will result in loss of baseline data. We are aware that this is inconvenient for customers, and we are currently working on addressing this limitation in a future release.

Follow the "Checklist: Upgrading Your Environment" on page 511 to ensure a successful upgrade.

As part of the process, you must upgrade CDF. The following upgrade options are available.

- Upgrading CDF Automatically with arcsight-install
- "Upgrading CDF Automatically with autoUpgrade" below
- "Upgrading CDF Manually" on page 522

We recommend performing the automatic installation with `arcsight-install`, as it is the easiest to use. However, if the automatic installation method does not meet your needs, you can upgrade manually.

If you installed your environment with the ignore-swap flag previously, then swap space needs to be disabled before you start the upgrade. Otherwise, the upgrade will fail, with first master not starting up. For more information see, Disabling Swap Space.

Upgrading CDF Automatically with `arcsight-install`

This is the simplest method for upgrading CDF, as it also will automatically perform required pre-upgrade and post-upgrade actions, which may contain installation bug fixes or workarounds usually described in release notes or known issues for manual installation.

To perform the automatic upgrade with `arcsight-install`:

1. Download the upgrade files for CDF to a download directory (referred to as `<download_directory>`) to a secure network location.
2. Navigate to *{unzipped-installer-dir}*.
3. Run the following command:
   `./arcsight-install --cmd upgrade --tmp-folder /my/tmp/folder`

Where `tmp-folder` is the name of your temporary upgrade directory.

Example: `./arcsight-install --cmd upgrade --tmp-folder /my/tmp/folder`

Example: `./arcsight-install --cmd upgrade` (/tmp will be used by default)

4. The upgrade will run without interruption. After the upgrade completes, remove the temporary folder by running the command:
   `rm -rf <path_to_custom_temporary_folder`

Upgrading CDF Automatically with autoUpgrade

The automated upgrade of CDF is performed using a single command and requires no interaction until completion of each phase. Typically, each automated upgrade phase takes around 1 hour for a cluster with 3 master nodes and 3 worker nodes. The process must be run from one of the cluster nodes.

Preparing the Upgrade Manager

Automatic upgrade should be run from a host that for purposes of these instructions is known as the upgrade manager. The upgrade manager (UM) may be one of the following host types:

- One of the cluster nodes
- A host outside the cluster (a secure network location)

The following uses the cluster master node1 as an example.

Configuring Passwordless Communication

You must configure passwordless SSH communication between the UM and all the nodes in the cluster.

1. Run the following command on the UM to generate key pair.

   ```
   ssh-keygen -t rsa
   ```

2. Run the following command on the UM to copy the generated public key to every node of your cluster.

   ```
   ssh-copy-id -i ~/.ssh/id_rsa.pub root@<node_fqdn_or_ip>
   ```

Downloading the Upgrade File

Download the upgrade files for CDF to a download directory (referred to as `<download_directory>`) on the UM.

There are three directories involved in the auto-upgrade process:

1. An auto-upgrade directory `/tmp/autoUpgrade` will be auto generated on the UM. It will store the upgrade process steps and logs.
2. A backup directory `/tmp/CDF_202005_upgrade` will be auto generated on every node (approximate size 1.5 GB).
3. A working directory will be auto generated on the UM and every node at the location provided by the `-d` parameter The upgrade package will be copied to this directory. (approximate size 9 GB). The directory will be automatically deleted after the upgrade.

   The working directory can be created manually on UM and every node and then passed as -d parameter to the auto-upgrade script. If you are a non-root user on the nodes inside the cluster, make sure you have permission to this directory.

Performing the CDF Automatic Upgrade

To perfom the CDF automatic upgrade using autoUpgrade:

1. Log in to the master node where you downloaded the upgrade files.
2. Change to the following directory:

   ```
   {unzipped-installer-dir}/installers/cdf/
   ```

3. Run the following command:

   ```
   ./autoUpgrade -d /path/to/working_directory -n {any_cluster_node_adress_
   or_ip}
   ```

   For example:

   ```
   ./autoUpgrade -d /tmp/upgrade -n yourdomain-
   masternode1.yourenterprise.net
   ```

Removing the Auto-upgrade Temporary Directory from UM

The auto-upgrade temporary directory contains the upgrade steps and logs.

To upgrade another cluster from the same UM, remove that directory using the following.

```
rm -rf /tmp/autoUpgrade
```

Upgrading CDF Manually

Beginning with the master node1, upgrade your CDF infrastructure on every node of the cluster. Rrun the following process *on each node*.

1. Run the following command:

   ```
   mkdir /tmp/upgrade-download
   ```

2. From the "Downloading the Installation Packages for an On-Premises Deployment" on page 512 section, copy the CDF bits.

   ```
   arcsight-platform-installer-<version>.zip to /tmp/upgrade-download
   ```

3. Unzip the upgrade package by running these commands.

   ```
   cd /tmp/upgrade-download
   unzip arcsight-platform-installer-<version>.zip
   ```

4. Run the following commands on each node (follow this pattern: master1, master2, master3, to worker1, worker2, worker3, etc.).

   ```
   cd /tmp/upgrade-download/arcsight-platform-installer-
   <version>/installers/cdf
   ```

   ```
   ./upgrade.sh -i
   ```

5. On the initial master node1, run the following commands to upgrade CDF components.

   ```
   cd /tmp/upgrade-download/arcsight-platform-installer-
   <version>/installers/cdf
   ```

   ```
   ./upgrade.sh -u
   ```

6. Clean the unused docker images by running the following commands on all nodes (masters and workers). This can be executed simultaneously.

   ```
   cd /tmp/upgrade-download/arcsight-platform-installer-
   <version>/installers/cdf
   ```

   ```
   ./upgrade.sh -c
   ```

7. To verify the cluster status, complete the following steps:
   a. Check the CDF version on each node by running the command:

   ```
   cat ${K8S_HOME}/version.txt
   ```

   b. Check the status of CDF on each node by running these commands:

   ```
   cd ${K8S_HOME}/bin
   ./kube-status.sh
   ```

8. To avoid possible "incorrect API route" error message while accessing IdM administration execute following line as root user on your master node:

   ```
   kubectl patch ing itom-idm-admin -ncore --type json -p '
   [{"op":"add","path":"/spec/rules/0/host","value":"'$(kubectl get cm -
   ncore base-configmap -ojsonpath='{.data.EXTERNAL_ACCESS_HOST}')'"}]'
   ```

Preparing for SOAR 3.2

If you have not previously deployed the SOAR capability, or if you do not plan to deploy it in future, skip this step.

SOAR is now a part of Fusion capability. Before upgrading to platform version 22.1, which includes SOAR 3.2, you must undeploy any existing version of SOAR to remove the resource definitions.

To undeploy SOAR:

1. Click **DEPLOYMENT**, and select **Deployments**.
2. Click the **Three Dots** on the right and select **Change**. A new screen tab is displayed.
3. Uncheck the Arcsight SOAR checkbox and click **NEXT** until you return to the **Deployment** page again.

Upgrading Deployed Capabilities

Make sure you have completed the "Checklist: Upgrading Your Environment" on page 511 process before proceeding with the steps listed here, to ensure a successful upgrade. As part of the process, you must upgrade your deployed capabilities using the CDF Management Portal.

- "Accepting the Certificate" on the next page
- "Considerations for Upgrading Intelligence" on the next page
- "Upgrading Deployed Capabilities" on the next page

Accepting the Certificate

1. On the CDF Management Portal, click **DEPLOYMENT**, and then select **Deployments.**
2. Click the **Three Dots** (Browse) on the far right and then choose **Reconfigure**.

   If you are unable to access the CDF Management Portal Reconfigure Page during the upgrade process, see the known issue Accessing the CDF Management Portal Reconfigure in the *ArcSight Platform Release Notes* for a workaround.
3. (Optional) If you have deployed Fusion in the **Fusion** tab, you can set a Fusion global search limit to specify the number of results a search can return. (The maximum is 10 million events.)
4. Accept the certificate.

Considerations for Upgrading Intelligence

Micro Focus does not recommend that ArcSight Intelligence customers attempt to upgrade without consulting with Micro Focus as upgrading at this time will result in loss of baseline data. We are aware that this is inconvenient for customers, and we are currently working on addressing this limitation in an future release.

If you have deployed Intelligence and are upgrading it, consider the performing the following steps on the NFS before upgrading the deployed capability:

- Ensure that you move your SQL loader scripts from: `/<arcsight_nfs_vol_path>/interset/analytics/vertica_loader_sql/0/<existing_folder_name>` to `/<arcsight_nfs_vol_path>/interset/analytics/vertica_loader_sql/0/1.<existing_folder_name>`.
- If you are using custom data identifiers, ensure that you back up the `logstash-config-pipeline` config map that is accessible through the Kubernetes dashboard.

Upgrading Deployed Capabilities

1. (Conditional) Before upgrading Intelligence, review the considerations for an upgrade.
2. To delete any pre-existing upgrade pods, run the following command from the master node:

```
kubectl delete deployments suite-upgrade-pod-arcsight-installer -n
`kubectl get namespaces | grep arcsight-installer | awk ' {print $1} '`
```

If the command returns an error message, it can be ignored and you may proceed to the next step.

3. Log in to the master node where you downloaded the upgrade files.
4. Change to the following directory:

```
cd ${K8S_HOME}/scripts
```

where **{K8S_HOME}** has a format similar to: `/opt/arcsight/kubernetes/scripts`.

5. Run the following commands to upload the images to the local Docker Registry. Use the `-F <image file>` option on the command line multiple times for each image to upload. Adjust the `-c 2` option up to half of your CPU cores in order to increase the speed of the upload. An example is shown here for Fusion, Transformation Hub, and Recon capabilities.

```
./uploadimages.sh -c 2 -F {unzipped-installer-dir}/images/fusion-
x.x.x.x.tar -F {unzipped-installer-dir}/images/transformationhub-
x.x.x.x.tar -F {unzipped-installer-dir}/images/recon-x.x.x.x.tar
```

You will be prompted for a password for the docker container registry-admin user. The registry-admin password is initially set to the same password as the admin user for the CDF Management Portal during installation when "Configuring and Running the CDF Installer" on page 96; however, later changing the CDF Management Portal admin password does not change the registry-admin password as it is managed separately.

6. Add new metadata.

Make sure to copy the `arcsight-installer-metadata-x.x.x.x.tar` to the system where your web browser is running before performing the process below.

7. Browse to the management portal at https://<virtual_FQDN>:5443, or at https://<master_node1_FQDN>:5443.
   a. Click **DEPLOYMENT > Metadata** and click **+ Add**.
   b. Select `arcsight-installer-metadata-x.x.x.x.tar` from your system. The new metadata is added to the system.
8. Start the upgrade process.
   a. Go to **DEPLOYMENT > Deployments**. Notice the number **1** in the red circle in the Update column.

> Minor version changes do not display like regular updates. (For example: 22.1.0.15 -> 22.1.0.16.)

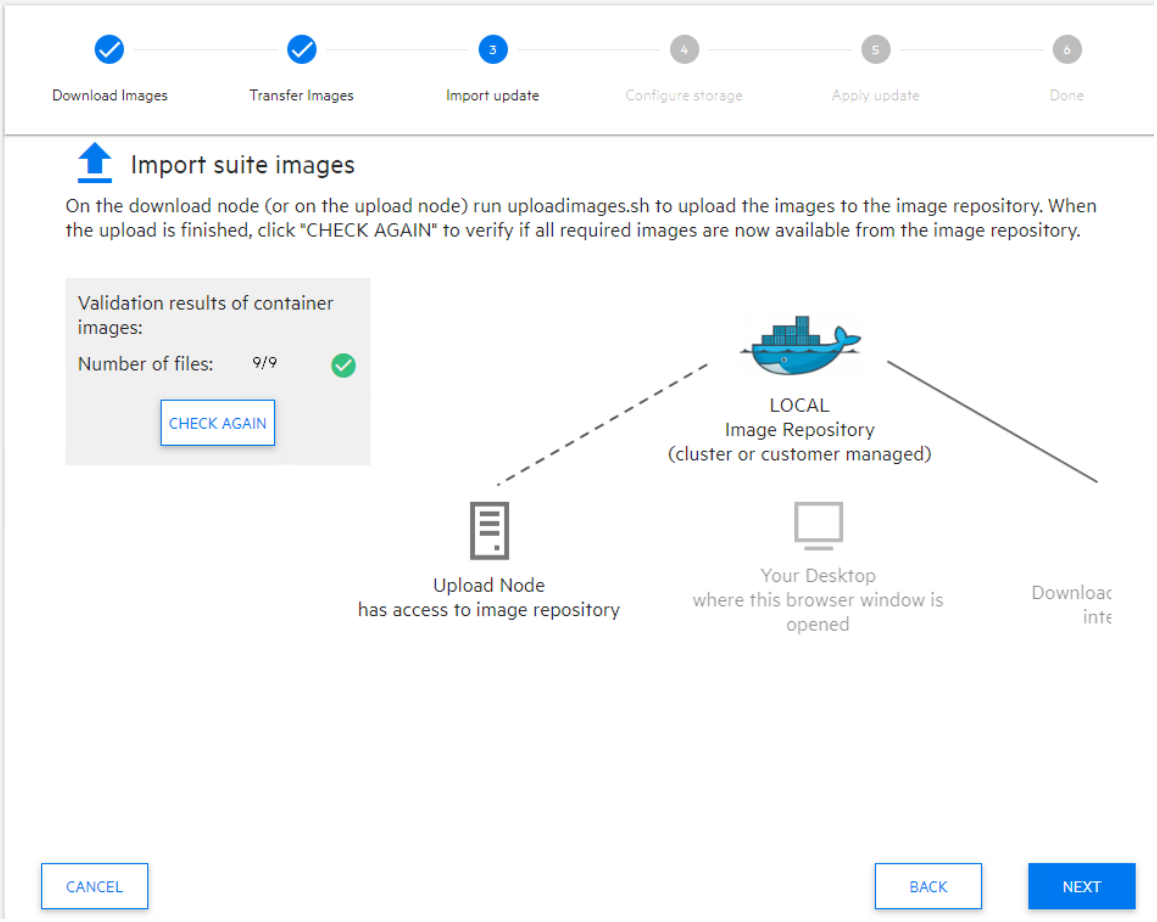
b. Click the red circle and select your recently added metadata to initiate the upgrade.

9. From the **Update to** page, click **NEXT** until you reach the **Import suite images** page.

> When prompted to download or transfer images, you can simply click Next to skip the steps. You performed these steps earlier.

10. Ensure that the validation results of container images show a complete number of files.

> When you arrive at the Import suite images page, the images should already be imported, as you performed these steps earlier.



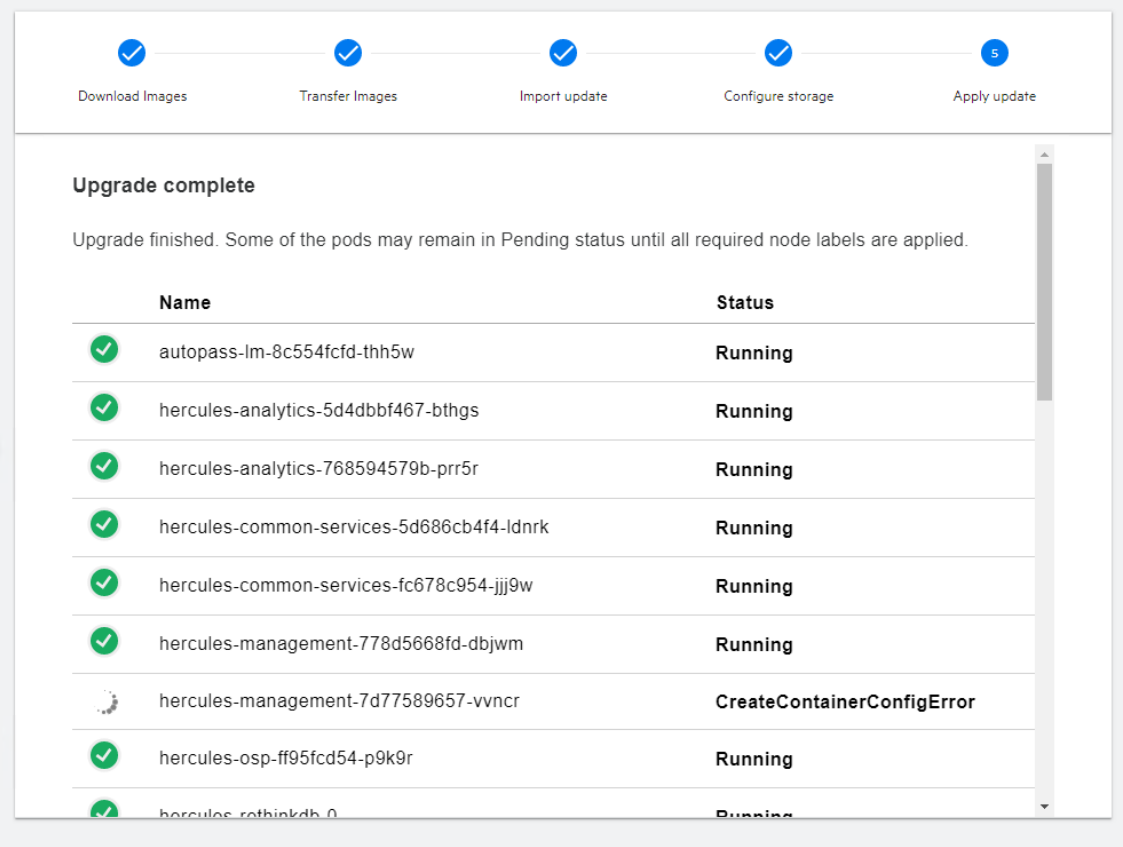
11. (Conditional) If you've deployed Transformation Hub, perform the following action:

    On the **Transformation Hub** tab, select whether you want to enable the generation of verification events for parsed field integrity checks used by Recon. If enabled, set the **verification event size** accordingly (default value is recommended).

For more information about how the Event Integrity feature works, see Stream Processor Groups.

When enabling the generation of verification events for parsed field integrity checks feature, consider the guidelines in Adjusting the Partition Count for New Kafka Topics.

12. Click **NEXT** until you reach the **Upgrade Complete** page.



Completing Post Upgrade Tasks

After installing the ArcSight Database and upgrading the ArcSight Platform, you need to update the Database Configuration in the CDF Management Portal, and then clean up the NFS directory. For more information, see:

- Reconfiguring the Database Host Address
- Cleaning Up the NFS Directory

If you have upgraded Intelligence, you must also perform the following post upgrade tasks:

- If you have been using custom SQL loader scripts in the previous versions of Intelligence, you need to apply the custom SQL loader scripts.
- If you have been using custom data identifiers, then you need to update the `logstash-`

`config-pipeline` config map. For more information, see Updating the Logstash ConfigMap for Custom Data Identifiers.

Reconfiguring the Database Host Address

As part of the upgrade, you need to update the Database Host address in order for the database to connect with your cluster.

The ArcSight Database is a separate installation and is not included in the platform upgrade. For more information, see Installing the Database

To update the Database Host address:

1. Log in to the CDF Management Portal.
2. Navigate to **Deployment** > **Deployments**. > ⋮ > **Reconfigure**.
3. Select the **Fusion** tab, and scroll down to the Database Configuration section.
4. Update the Database Host field with the IP addresses of all database nodes, beginning with Node 1 and ending with Node N. For example: *`node1-IP,node2-IP,node3-IP,...`* and so forth.
5. Click **Save** to activate the configuration changes.

Cleaning Up the NFS Directory

After installing the ArcSight Database and upgrading the Platform, you must clean up the NFS directory to remove instances of existing configuration data before you begin a fresh install.

The ArcSight Database is a separate installation and is not included in the platform upgrade. For more information, see Installing the Database.

You must backup necessary configuration data like custom dashboards, searches etc. to avoid data loss on account of the clean up. For more information, see Backing Up and Restoring Configuration Data for Deployed Capabilities.

1. Remove the "Reporting" directory under NFS:

```
rm -rf /opt/arcsight-nfs/arcsight-volume/reporting
```

2. Remove the "rethinkdb" directory under NFS:

```
rm -rf /opt/arcsight-nfs/arcsight-volume/investigate/search/rethinkdb
```

3. Restart the pods that require rethinkdb:

```
NSP=$( kubectl get namespaces | grep arcsight | cut -d ' ' -f1 )
```

```
kubectl get pods -n $( kubectl get namespaces | grep arcsight | cut -d '
' -f1 ) --no-headers=true | awk '/fusion/{print $1}'| xargs  kubectl
delete -n  $NSP pod
```

Applying Custom SQL Loader Scripts

Applies if Intelligence has been upgraded.

If you have been using custom SQL loader scripts in any of the previous versions of Intelligence, then, after the upgrade, the analytics pod enters into a CrashLoopBackOff state. To recover from this state and enable the analytics pod to run properly, do the following:

1. Launch a terminal session and as a root user, log in to the node where NFS is present.
2. If you have been using custom SQL loader scripts, review and add the necessary modifications to the new SQL loader scripts present in the following directory:

   ```
   cd /<arcsight_nfs_vol_path>/interset/analytics/vertica_loader_
   sql/0/1.9.1.x
   ```

3. Navigate to the following directory:

   ```
   /<arcsight_nfs_vol_path>/interset/analytics/vertica_loader_
   sql/0/1.<existing_folder_name>
   ```

   where 1.<existing_folder_name> is the folder where you moved your SQL loader scripts prior to the upgrade.
4. Update the md5 files with the md5 sums corresponding to the modified SQL loader scripts.
5. Execute the following commands to restart the analytics pod:

   ```
   export NS=$(kubectl get namespaces |grep arcsight|cut -d ' ' -f1)
   kubectl -n $NS scale deployment interset-analytics --replicas=0
   kubectl -n $NS scale deployment interset-analytics --replicas=1
   ```

Restarting the Event Consumers

If your ArcSight Platform deployment includes a capability that needs the ArcSight Database, such as Intelligence, complete the following procedure after upgrading the Platform.

1. (Conditional) If you have Intelligence deployed, complete the following steps:
   a. Log in to the Transformation Hub node.
   b. Run the following commands:

      ```
      NS=$(kubectl get namespaces | awk '/arcsight/{print $1}')
      ```

```
kubectl scale statefulset interset-logstash -n $NS  --replicas=
{replica count}
```

where {replica count} represents one of the following values:

- The number in the **READY** column that you captured in Step 6 of "Stopping the ESM and Intelligence Event Consumers."
- A new replica count value that you defined for Intelligence during the recent upgrade

Upgrading ESM

As part of the upgrade process, If ESM is deployed, you might want to change your configuration and upgrade it.

- To change your ESM enrichment configuration and Transformation Hub topic routing, see Local and Global ESM Event Enrichment.
- To upgrade ESM, see the Upgrade Guide for ESM.
- After completing the upgrade, you must reconfigure the ESM host in the CDF Management Portal.

Applying the CDF 2021.05 log4j Hotfix

Some deployments of, and upgrades to, CDF 2021.05/arcsight-platform-installer-22.1.x.x.zip require application of a hotfix to remediate the log4j vulnerability, which was discovered in 2021. The hotfix will upgrade IDM for CDF to use log4j 2.17.1, to prevent exploitation of the log4j vulnerabilities (CVE-2021-44228, CVE-2021-45046, CVE-2021-45105, CVE-2021-44832). The hotfix should be applied after an upgrade.

The hotfix applies to the following types of installations and upgrades:

- Any on-premises manual installation of 22.1.x or any on-premises manual upgrade to 22.1.1. (A manual installation or upgrade is one that does not use the ArcSight Installer.)

The CDF 2021.05 hotfix will be automatically applied during any on-premises installation or upgrade using the ArcSight Installer and this procedure can be skipped.

- Any CDF fresh installation or upgrade on AWS.
- Any CDF fresh installation or upgrade on Azure.

The log4j remediation hotfix does **NOT** apply to on-premises installations or upgrades performed automatically using the ArcSight Installer, as the hotfix is applied automatically by the installer. For such installations or upgrades these procedures can be skipped.

Hotfix File

The hotfix file is named arcsight-idm-hf-22.1.0-2.zip.

1. Get the file:
   - For a manual on-premises deployment/upgrade, the hotfix is bundled in `/<download_folder>/arcsight-platform-installer-22.1.x.x/installers/hotfix`.
   - For an AWS or Azure deployment upgrade, obtain the file from the *on-premises* installation file directory at `/<download_folder>/arcsight-platform-installer-22.1.x.x/installers/hotfix`.
2. Copy the file:
   - For manual on-premises, copy the file to your master node.
   - For AWS, copy the file to your bastion.
   - For Azure, copy the file to your jump host.
3. Unzip the hotfix file. In the unzipped folder, run the following command with the '-e' argument (values: onprem, azure, aws) to apply the latest image.

```
# ./hotfix.sh -e <YOUR_ENV>
```

Verifying the Hotfix

1. Check the pod status by running the following command. It should be 'Running' as 2/2.

```
# kubectl get pods -A | grep idm
```

2. Check the image version by running the following command.

```
# kubectl get deployment/itom-idm -n core -o yaml | grep itom-idm:1.32.1-343
```

It should display as below:

```
image: <image-registry-url>/<org-name>/itom-idm:1.32.1-343
```

Rolling Back to the Previous Version

To roll back itom-idm to the previous version, run the following roll back commands :

```
# kubectl delete -f /tmp/cdf-itom-idm.yaml
```

```
# kubectl create -f /tmp/cdf-itom-idm.yaml
```

Upgrading Your Amazon Web Services Deployment

Micro Focus does not recommend that ArcSight Intelligence customers attempt to upgrade without consulting with Micro Focus as upgrading at this time will result in loss of baseline data. We are aware that this is inconvenient for customers, and we are currently working on addressing this limitation in an future release.

This release significantly changes the ArcSight Database. **You cannot upgrade the database. It must be installed as new.** However, this release does enable you to deploy or upgrade Recon and Intelligence in your environment, as well as install either capability for the first time.

The complete process of upgrading an AWS deployment involves the following tasks.

	Task	See
❑	1. Verify upgrade prerequisites are met	"Reviewing Deployment Prerequisites" on page 134
❑	2. Upgrade the EKS control plane to version 1.20	"Upgrading the EKS Control Plane" on the next page
❑	3. Update kubectl on the bastion	"Updating kubectl on the Bastion" on page 534
❑	4. Prepare the upgrade package	"Preparing the Upgrade Package" on page 534
❑	5. Upload product images to the ECR	"Uploading the New Images to the Elastic Container Registry (ECR)" on page 535
❑	6. (Conditional) To deploy Intelligence or Recon, install the ArcSight Database	"Installing the Database in AWS" on page 535
❑	7. (Conditional) To prevent both the original and newly installed databases from ingesting duplicate events, stop data ingestion on the original database.	"Checklist: Stopping Event Ingestion" on page 564
❑	8. Upgrade CDF on AWS	"Running the Upgrade" on page 546
❑	9. Change value for portal-ingress-controller-svc, if necessary	"(Conditional) Changing the Values of frontend-ingress-controller-svc and portal-ingress-controller-svc" on page 550
❑	10. Perform the upgrade of deployed capabilities	"Upgrading Deployed Capabilities on AWS" on page 550

❑	11. Clean up the EFS Directory post upgrade	Cleaning Up the EFS Directory (AWS)
❑	12. Modify producers, consumers, and ArcMC	"Modify the (Connector) Producers and Consumers" on page 554
❑	13. Restarting the Event Consumers	"Restarting the Event Consumers" on page 570
❑	14. (Optional) Add Transformation Hub to Fusion ArcMC	"(Optional) Adding Transformation Hub to Fusion ArcMC" on page 556
❑	15. Apply the hotfix to remediate the log4j vulnerability	"Applying the CDF 2021.05 log4j Hotfix" on page 570
❑	16. Apply security patches and latest updates to the deployed capabilities and the ArcSight Database	"Upgrading to 22.1.2" on page 572

Upgrade Prerequisites

The upgrade of CDF on AWS requires the following:

- Root access to the bastion VM.
- The kubernetes command line tool kubectl installed on your bastion and connected to your cluster

Prior to beginning the upgrade, perform the following tasks:

- All pods must be running. Run this command to ensure that all pods are running:
  `kubectl get pods -A`
- Execute this command to ensure any existing suite-upgrade-pod-arcsight-installer deployment have been deleted:
  `kubectl delete deployment suite-upgrade-pod-arcsight-installer -n $(kubectl get namespaces | grep arcsight-installer | awk ' {print $1} ')`
- Download the upgrade packages, including the ArcSight Platform Cloud Installer, to a secure network location.

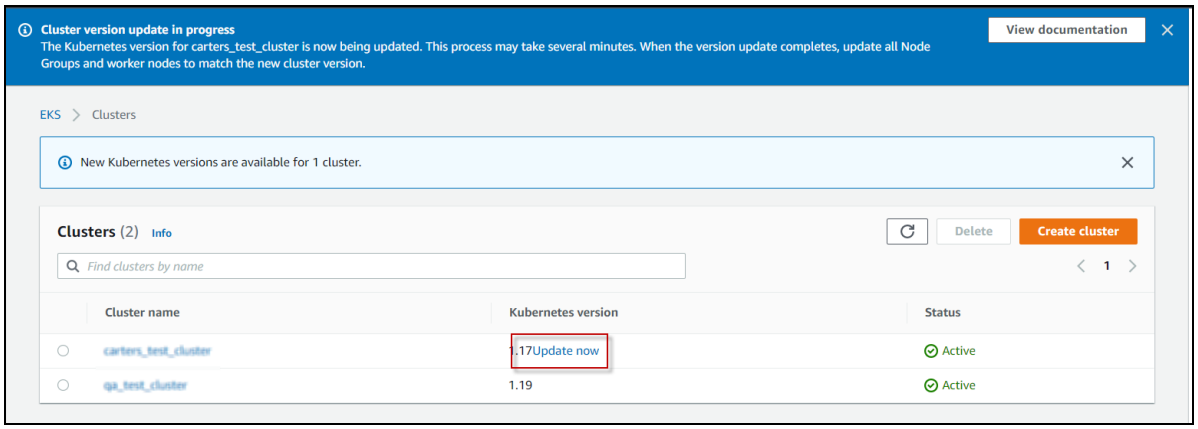
You are now ready to perform the upgrade.

Upgrading the EKS Control Plane

This version of the platform requires Kubernetes 1.20. Because Amazon EKS runs a highly-available control plane, you can upgrade only one minor version in one step. Therefore, if your current version is 1.18 and you wish to upgrade to 1.20, then you must first upgrade your cluster to 1.19 and then upgrade it from 1.19 to 1.20. If your current version is 1.19, you will be able to upgrade in one step.

Note: Each EKS control plane upgrade can take 20-30 minutes to complete.

1. In AWS, browse to the EKS console.
2. Click **Clusters.**
3. In the list of clusters, next to your cluster, click **Upgrade now.**



4. Select the next Kubernetes minor version, and then click **Update.**
5. (Conditional) If you are originally upgrading from version 1.18, repeat steps 3-4 to complete the upgrade from version 1.19 to 1.20.

Before proceeding, wait until the Status changes from *Updating* to *Active*.

Updating `kubectl` on the Bastion

With the update of EKS, the `kubectl` binary on the bastion host must also be updated. You must use a `kubectl` version that is within one minor version difference of your Amazon EKS cluster control plane. For example, a 1.20 `kubectl` client works with Kubernetes 1.19, 1.20 and 1.21 clusters. A list of supported Kubernetes binaries can be found here.

For information on installing `kubectl`, see Creating and Configuring the Bastion.

Preparing the Upgrade Package

1. Log in to the secure network location you stored the Arcsight Platform Cloud Installer earlier when checking prerequisites.
2. Unzip `arcsight-platform-cloud-installer-<VERSION>.zip`

   Example:

   ```
   cd /tmp
   unzip arcsight-platform-cloud-installer-<VERSION>.zip
   ```

3. Upload the file `cdf-deployer.zip` to the bastion host. Example command:

```
scp -i bastion_key arcsight-platform-cloud-installer-<VERSION>/cdf-
deployer.zip centos@3.127.98.59:/tmp/
```

Uploading the New Images to the Elastic Container Registry (ECR)

All of the following steps can be performed on the bastion or a host with Internet access and the AWS command line configured.

1. Switch to the aws scripts installer directory:
   `cd arcsight-platform-cloud-installer-<VERSION>`
2. Unzip aws-scripts.zip
   `unzip aws-scripts.zip`
3. Switch to the scripts directory
   `cd aws-scripts/scripts`
4. Run upload_images_to_ECR with your credentials from the ECR with the following command:

   `./upload_images_to_ECR -o <your-org-name> -F arcsight-platform-cloud-installer-<VERSION>/cdf-byok-images.tar -c 4`

   Ensure the organization argument `-o` is same as the one used during the installation. You can check your organization name by running the command:
   `kubectl get cm -n core base-configmap -o yaml | grep REGISTRY_ORGNAME:`

For more information on running the script, see `upload_images_to_ECR --help`.

Installing the Database in AWS

This section provides information about installing the ArcSight Database in Amazon Web Services Deployment (AWS).

- Understanding Methods for Connecting to AWS S3 Buckets
- Launching Database Instances in AWS
- Updating CentOS (conditional)
- Creating the Swap File
- Formatting Devices for the Installation
- Persisting Operating System Settings
- Enabling Root Login for AWS Passwordless Communication
- Setting Up and Installing the Database for AWS

Understanding Methods for Connecting to AWS S3 Buckets

The database uses a single communal storage location for all data and for the catalog (metadata). Communal storage is the database's centralized storage location, shared among the database nodes. This mode supports communal storage in Amazon S3, which must be set up by your cloud administrator before you can install the database.

Other prerequisites or considerations for the S3 bucket:

- Set up default encryption for the S3 bucket in advance. For information about enabling S3 bucket encryption, see AWS documentation, *Enabling Amazon S3 default bucket encryption*.
- If you require a folder under your S3 bucket, it must be created in the communal storage procedure. Folders pre-created under the bucket via the AWS console are not supported for a new installation of the database.

  For more information, see Configuring and Installing the Database Server
- The database supports connecting to AWS S3 buckets using IAM roles. IAM roles are the default access control method for AWS resources. The database uses this method if you do not configure the legacy access control session parameters.

  To use an IAM role, the bucket must be in the same region as the database cluster and the role needs to be set with the proper permissions for reading and writing to the S3 bucket.

  For more information about IAM roles, see AWS documentation, *IAM Roles for Amazon EC2* and *Creating a role to delegate permissions to an AWS service*.

  The example policy below shows the permissions needed for the IAM policy role:

```
 {
"Sid": "s3CommunalLocationAccess",
"Action": [
"s3:ListBucket",
"s3:GetObject",
"s3:DeleteObject",
"s3:PutObject",
"s3:*MultipartUpload*"
],
"Resource": [
"arn:aws:s3:::<db-communal-storage-bucket-name>/*",
"arn:aws:s3:::<db-communal-storage-bucket-name>"
]
}
```

In the example above you would replace the <db-communal-storage-bucket-name> with the name of the S3 bucket you created for the database communal storage. For example, `222222222-ap-southeast-1-arcsight-db`

Launching Database Instances in AWS

The ArcSight database compute operations are performed on a cluster of AWS EC2 (Elastic Compute Cloud) nodes. These nodes must be deployed in AWS before the database can be installed on them.

1. Sign in to the Amazon EC2 console and select a region. For more information, see Understanding Methods for Connecting to AWS S3 Buckets
2. Select a launch instance from the **Choose AMI** tab. The database supports Red Hat 8.4 and CentOS 8.4 AMIs. For information about finding an AMI, see AWS documentation, *Find a Linux AMI*.

   For example. If you are in the US East (N. Virginia) region, you can use the following AMIs:

   - `Red Hat 8.4 (AMI Id: ami-0453b4b64c860454a)`
   - `CentOS 8.4 (AMI Id: ami-05b1369539e0e69bd)`
3. After selecting the AMI, and reviewing the details, select an instance type. Choose the AWS EC2 instance type that best matches your requirements.

   We recommend using an m5d instance type, because it has the best balance of cost, performance, and ease of setup for ArcSight workloads. Take into account that m5d is an instance store. Instance stores provide temporary storage. For example, data files stored on the instance are lost when the instance is stopped. The Durable EBS instance type is also supported with Elastic Block Store (EBS). This instance type provides durable storage. For example, data files stored on the instance persist after the instance is stopped.

   For more information on which size of m5d (Instance Store Type) and /m4 (EBS Type) should be used for your workload, see the ArcSight Platform Technical Requirements.
4. Configure the number of AWS EC2 instances to deploy. A minimum of three is needed for a highly available database cluster, but more may be needed to handle your workload. For more information on how many instances you need, see the ArcSight Platform Technical Requirements.
   a. Configure your instance with the **VPC** and **private subnets**.
   b. Select the IAM role you configured to have access to the S3 bucket for the database communal storage.
   c. Click **Next: Add Storage**.
5. On the Add Storage page, change the size for the Root volume type to 30 GiB or higher. 30 GiB is the minimum size for root.

If you are using EBS instance type m4, add an EBS volume according to ArcSight Platform Technical Requirements.

6. Click **Next: Add Tags**, and add tags as needed.
7. Click **Next: Configure the Security Group**. Configure your instance with your private security groups.
8. Click **Review and Launch** to review the details of your instance.
9. Click **Launch**, and select an existing key pair or create a new key pair.
10. Click **Launch Instances**.
11. Navigate to EC2, and verify your instances are available.

Updating CentOS (conditional)

If you are deploying the database with CentOS 8.4 2105, you need to update the distros by running the commands below on all database nodes:

```
sudo dnf --disablerepo '*' --enablerepo=extras swap centos-linux-repos
centos-stream-repos
sudo dnf distro-sync
```

If you discover that the above distro repository is broken, update the `--enablerepo` repositories with those provided by CentOS: https://www.centos.org/centos-stream/.

Creating the Swap File

A partition with swap space configured is required to enable virtual memory access. The database requires a minimum of 2 GB of swap space on all database nodes. Perform the following procedure on all database nodes to setup the swap space.

1. From the bastion machine, login to the database node using SSH and the key pair selected during instance launching.
2. Create the file that will be used for swap:

   ```
   sudo dd if=/dev/zero of=/swapfile bs=1024 count=2621440
   ```

   The count 261440 is the minimum value required for installation, but it is also an example. You can set the value higher.

3. Enable read and write to the swap file for the root user (only):

   ```
   sudo chmod 600 /swapfile
   ```

4. Set up the file as a Linux swap area:

```
sudo mkswap /swapfile
```

5. Enable the swap with the following command:

```
sudo swapon /swapfile
```

6. Enable the swap permanently as root user by editing the `/etc/fstab` file and appending the following line:

```
echo "/swapfile swap swap defaults 0 0" | sudo tee -a /etc/fstab
```

7. Verify that swap is enabled:

```
sudo swapon --show
```

```
Output example:

NAME      TYPE SIZE USED PRIO
/swapfile file 2.5G   0B   -2
```

Formatting Devices for the Installation

The database requires a storage device formatted in ext4 for all nodes. Perform the following procedure on all database nodes to setup ext4 in devices.

The steps below are required to be performed on all AWS EC2 instances that are created. For more information about creating instances, see Launching Database Instances in AWS

Format the device type applicable to the AWS EC2 type you are using in the instance, either an instance Store type or EBS type:

- Formatting Instance Store Type Devices
- Formatting EBS Type Devices

Formatting Instance Store Type Devices

The steps below are required to be performed on all AWS EC2 instances that are created.

ONLY perform this procedure if you are using **Instance Store** type devices.

1. From the Bastion machine, login to the database node using SSH and the key pair selected during launching.

2. Install the `nvme` tool:

```
sudo dnf install -y nvme-cli
```

3. Create an `/opt` `mount` point. If `/opt` alread exists, skip this step.

```
sudo mkdir /opt/vertica
```

4. Use the `nvme` tool to locate the `nvme` device and format it:

```
EPHEMERAL_DISK=$(sudo nvme list | grep 'Amazon EC2 NVMe Instance Storage'
| awk 'NR==1{ print $1 }')
```

```
echo $EPHEMERAL_DISK
```

```
sudo mkfs.ext4 $EPHEMERAL_DISK
```

```
sudo mount -t ext4 $EPHEMERAL_DISK /opt
```

5. Add the ephemeral disk mount to `/etc/fstab`:

```
echo "$EPHEMERAL_DISK /opt ext4 defaults,nofail 0 0" | sudo tee -a
/etc/fstab
```

6. To list the mount points for verification, run the `lsblk` command:

```
lsblk
```

Formatting EBS Type Devices

The steps below are required to be performed on all AWS EC2 instances that are created. These steps for EBS also assume you are formatting a single device. For more information about creating instances, see Launching Database Instances in AWS.

ONLY perform this procedure if you are using **EBS** type devices.

1. Run the following command to locate the EBS device attached to the instance, and to select the device that is not mounted:

```
lsblk -f
```

```
#Output from lsblk -f
NAME FSTYPE LABEL UUID MOUNTPOINT
xvda
```

```
└─xvda1 xfs 35761952-413c-42e8-b047-a5deb7510f29 /
xvdb
```

2. Run the following command to format the device to ext4:

```
sudo mkfs.ext4 /dev/xvdb
```

NOTE: xvdb is an example and will be <yourDeviceName> from the output in the previous step.

3. Create the mount point. If /opt already exists, skip this step.

```
sudo mkdir /opt/vertica
```

4. Format the device:

```
sudo mount -t ext4 /dev/xvdb /opt
```

5. Modify the /etc/fstab file, and add an entry to mount the device on OS boot:

```
echo "/dev/xvdb /opt ext4 defaults 0 0" | sudo tee -a /etc/fstab
```

Persisting Operating System Settings

The database requires that you manually configure several general operating system settings. Perform the following procedures on all database nodes to setup ext4 in devices.

1. Run this command to set the limit for open files so that it meets database requirements. This will add the parameters to the /etc/sysctl.conf file.

```
cat << EOF | sudo tee -a  /etc/sysctl.conf
net.core.somaxconn = 1024
net.core.wmem_max = 16777216
net.core.rmem_max = 16777216
net.core.wmem_default = 262144
net.core.rmem_default = 262144
net.core.netdev_max_backlog = 100000
net.ipv4.tcp_mem = 16777216 16777216 16777216
net.ipv4.tcp_wmem = 8192 262144 8388608
net.ipv4.tcp_rmem = 8192 262144 8388608
net.ipv4.udp_mem = 16777216 16777216 16777216
net.ipv4.udp_rmem_min = 16384
net.ipv4.udp_wmem_min = 16384
vm.swappiness = 1
EOF
```

2. Update the `/etc/rc.local` file by running the commands below. This file contains scripts and commands that run each time the system is booted, and the database requires that I/O Scheduling be set to *deadline* or *noop*. The command will add the applicable lines to the file, based on the devices type in your AWS instance, either **instance Store** type or **EBS** type:

   Instance Store type rc.local settings (only)

   ```
   cat << EOF | sudo tee -a /etc/rc.local
   echo deadline > /sys/block/nvme0n1/queue/scheduler
   echo deadline > /sys/block/nvme1n1/queue/scheduler
   echo deadline > /sys/block/nvme2n1/queue/scheduler
   echo deadline > /sys/block/md0p1/queue/scheduler

   /sbin/blockdev --setra 2048 /dev/nvme0n1
   /sbin/blockdev --setra 2048 /dev/nvme1n1
   /sbin/blockdev --setra 2048 /dev/nvme2n1
   /sbin/blockdev --setra 2048 /dev/md0p1

   if test -f /sys/kernel/mm/transparent_hugepage/enabled; then
   echo never > /sys/kernel/mm/transparent_hugepage/enabled
   fi

   if test -f /sys/kernel/mm/transparent_hugepage/defrag; then
   echo never > /sys/kernel/mm/transparent_hugepage/defrag
   fi
   EOF
   ```

   Note: Replace the device in the example command above with your own instance device on which the `/opt` is mounted. For example, the following device must be replaced with your own device: `sys/block/nvmen0n1`

   EBS type rc.local settings (only)

   ```
   cat << EOF | sudo tee -a /etc/rc.local
   echo deadline > /sys/block/xvda/queue/scheduler
   echo deadline > /sys/block/xvdb/queue/scheduler
   echo deadline > /sys/block/xvdc/queue/scheduler

   /sbin/blockdev --setra 2048 /dev/xvda
   /sbin/blockdev --setra 2048 /dev/xvdb
   /sbin/blockdev --setra 2048 /dev/xvdc

   if test -f /sys/kernel/mm/transparent_hugepage/enabled; then
   echo never > /sys/kernel/mm/transparent_hugepage/enabled
   fi
   ```

```
if test -f /sys/kernel/mm/transparent_hugepage/defrag; then
echo never > /sys/kernel/mm/transparent_hugepage/defrag
fi
EOF
```

Note: Replace the device in the example command above with your own instance device on which the `/opt` is mounted. For example, the following device must be replaced with your own instance device:

`sys/block/dev/xvdb`

3. Modify the file permissions:

```
sudo chmod +x /etc/rc.d/rc.local
```

4. Run the following commands to disable the system firewall:

```
sudo systemctl mask firewalld
sudo systemctl disable firewalld
sudo systemctl stop firewalld
```

During installation, the database requires that host-based firewalls are disabled on database nodes. After installation, the host-based firewalls can be enabled and the database requires several ports to be open on the local network. We recommend for optimal performance using host-based firewalls between database nodes and a network-based firewall to protect the segment that database cluster is within. However, there is no restriction against using a network-based firewall between database nodes. When using any kind of firewall, ensure that all the database ports are available. For more information, see Firewall Considerations.

5. Set SELinux to permissive mode in `/etc/selinux/config`.

```
SELINUX=permissive
```

For more information, see SELinux Configuration.

6. Run this command to ensure that `rng-tools` packages are installed in all cluster nodes:

```
sudo dnf install rng-tools -y
```

7. Set the UTC time for all cluster nodes:

```
sudo timedatectl set-timezone UTC
```

For CentOS 8.4, any changes to the timezone will require a cluster nodes reboot.

8. Reboot the system for your changes to take effect.

This step is required to update the device mounts and other settings.

Enabling Root Login for AWS Passwordless Communication

All commands require root privileges which can be obtained through the sudo command.

1. Connect to one of your nodes and edit the `/etc/ssh/sshd_config` configuration file.

   ```
   sudo vi /etc/ssh/sshd_config
   ```

2. Change the following parameter to `yes` if the value is not already set to that.

   ```
   PermitRootLogin yes
   ```

3. Proceed as follows:
   - If you had to change the `PermitRootLogin` parameter to `yes`, run this command:

     ```
     sudo service sshd reload
     ```

   - If the `PermitRootLogin` parameter was already set to `yes`, proceed to the next step.

4. Patch the authorized keys file for the root user by copying the centos file to the root user. This enables logging in with the same key which is available for the centos-user/ec2-user (rhel-8.4).

   ```
   sudo cp ~centos/.ssh/authorized_keys ~root/.ssh/authorized_keys
   ```

5. Update the AWS cloud configuration file by editing `/etc/cloud/cloud.cfg` and changing the `disable_root` value to 0.

   ```
   sudo vi /etc/cloud/cloud.cfg
   ```

6. Repeat all the above steps for all nodes in the cluster.
7. Generate SSH keys for the database nodes:
   a. Log in to one of the database nodes. This becomes the initiator node (node1) for your cluster.
   b. Run the following command as root:

      ```
      ssh-keygen -q -t rsa -f ~/.ssh/id_rsa
      ```

   c. Run the following command as root, for each of the database nodes (initiator node inclusive):

```
ssh-copy-id -f "-o IdentityFile <FilePath to the ec2 keypair>"
root@<node1..n>
```

```
Enter : Yes
```

d. To verify the passwordless communication, run the following command:

```
ssh root@<node1,node2..>
```

```
Output example:

Activate the web console with: systemctl enable --now cockpit.socket
Last login: Thu Jan 20 19:26:50 2022 from <node1,node2..>
```

Setting Up and Installing the Database for AWS

Complete the procedures below in succession:

1. Modifying the System Clock
2. Enabling FIPS Mode on the Database Server (conditional)

   This step is needed only if your environment requires FIPS.

3. Configuring and Installing the Database Server
4. Creating the Elastic File System

Checklist: Stopping Event Ingestion

Before upgrading your environment, complete the following procedures in the listed order:

	Task	See
❑	1. Stop the database event consumer (the Kafka Scheduler) so you can record its offsets	"Stopping the Database Event Consumer" on page 514
❑	2. (Conditional) If you have Intelligence deployed, monitor the Intelligence event consumers to stop them when they catch up to the database event consumer	"Stopping the Intelligence Event Consumers" on page 515
❑	3. Prevent the database from consuming duplicate events	"Preventing the Database from Consuming Duplicate Events" on page 517

Running the Upgrade

1. On the bastion, verify that all pods in core namespaces are in status *Running* or *Completed* by running this command:

   ```
   kubectl get pods -n core
   ```

   ### Output example:

   ```
   cdf-apiserver-7965dcf689-4qvkx                                    2/2     Running
    0          145m

   fluentd-7q4dw                                                     2/2     Running
    0          136m

   fluentd-kkf2p                                                     2/2     Running
    0          136m

   fluentd-mwqh8                                                     2/2     Running
    0          136m

   idm-77b4f9fbfb-cfwkg                                              2/2     Running
    0          136m

   idm-77b4f9fbfb-g5pcb                                              2/2     Running
    0          136m

   itom-cdf-deployer-2020.05-2.2-2.3-3.1-tncp8                       0/1     Completed
    0          137m

   itom-cdf-deployer-xg6cw                                           0/1     Completed
    0          147m

   itom-cdf-ingress-frontend-56c9987b7-bvrsn                         2/2     Running
    0          145m

   itom-cdf-ingress-frontend-56c9987b7-n8tbc                         2/2     Running
    0          145m

   itom-logrotate-deployment-6cf9546f8b-rbcvs                        1/1     Running
    0          136m

   itom-postgresql-default-77479dfbff-t87tv                          2/2     Running
    0          137m
   ```

```
itom-vault-6f558dc6cc-bz52l                                  1/1     Running
 0          146m

kubernetes-vault-67f8698568-csd54                            1/1     Running
 0          145m

mng-portal-7cfc584db5-hcmjf                                  2/2     Running
 0          133m

nginx-ingress-controller-6f6d4c95b9-7fhbs                    2/2     Running
 0          133m

nginx-ingress-controller-6f6d4c95b9-nv2zw                    2/2     Running
 0          133m

suite-conf-pod-arcsight-installer-86c9687b69-kctjz           2/2     Running
 0          132m

suite-db-68bfc4fbd5-v6nvm                                    2/2     Running
 0          145m

suite-installer-frontend-6f49f88797-msb7j                    2/2     Running
 0          145m
```

2. Patch itom-cdf-ingress-frontend-svc and nginx-ingress-controller-svc as type LoadBalancer by running the following commands:

```
kubectl patch services itom-cdf-ingress-frontend-svc \
-p '{"metadata":{"annotations":{"service.beta.kubernetes.io/aws-load-
balancer-internal":"0.0.0.0/0","service.beta.kubernetes.io/aws-load-
balancer-type":"nlb"}},"spec":{"type":"LoadBalancer"}}' -n core
```

```
kubectl patch services nginx-ingress-controller-svc \
-p '{"metadata":{"annotations":{"service.beta.kubernetes.io/aws-load-
balancer-internal":"0.0.0.0/0","service.beta.kubernetes.io/aws-load-
balancer-type":"nlb"}},"spec":{"type":"LoadBalancer"}}' -n core
```

3. Wait until the command status in Step 2 changes from <pending>, and then run the following commands to get the load balancer DNS names:

```
kubectl get svc itom-cdf-ingress-frontend-svc -n core| grep -v EXTERNAL-
IP |awk '{print $4}'
kubectl get svc nginx-ingress-controller-svc -n core| grep -v EXTERNAL-IP
|awk '{print $4}'
```

4. Switch directory to cdf-deployer directory:
   `cd /tmp/`
5. Unzip cdf-deployer.zip:
   `unzip cdf-deployer.zip`
6. Switch to the cdf-deployer directory:
   `cd cdf-deployer/`
7. Run the upgrade script with the following command:
   `./upgrade.sh -u`

   Output example:

```
**********************************************************************
**********
```

```
WARNING: This step is used to upgrade CDF components to 2021.02 release.
```

```
The upgrade process is irreversible. You can NOT roll back.
```

```
Make sure that all nodes in your cluster are in Ready status.
```

```
Make sure that all Pods and Services are Running.
```

```
**********************************************************************
**********
```

```
Do you want to continue (Y/N): y
```

```
** Pre-checking before upgrade ...
```

```
** BYOK upgrade check...
```

```
** Common upgrade check...
```

```
[1]  Checking CDF endpoints status ...
```

```
Passed
```

```
** Prerequisite tasks for components upgrade... (Step 1/3)
```

```
Setting BYOK environment values ...
```

```
Copying itom-cdf-alias.sh to /etc/profile.d/ ...
```

```
** Updating Kubernetes RBAC ... (Step 2/3)

RBAC update successfully.

Updating images configmap ...

Update images configmap successfully.

Updating k8s-object-mapping configmap ...

Update k8s-object-mapping configmap successfully.

Updating base configmap...

Updating DEPLOYMENT_MODE:SUITE

** Configure and start the cdf-deployer ... (Step 3/3)

Creating resources from YAML: /home/centos/arcsight-platform-cloud-
installer-21.1.0.754-master/cdf-deployer/objectdefs/itom-cdf-deployer-
upgrade.yaml

Waiting for CDF components upgrade process complete ...

.................................

CDF components upgrade process completed.

Successfully completed CDF components upgrade process.
```

8. When the upgrade completes, check that all pods are *Running* or *Completed* with this command:
   `kubectl get pods -A`

The command `upgrade.sh -u` may fail on apphub. If this occurs, run the following command:
`helm rollback apphub 1 -n core`

The CDF upgrade is now complete. The new platform version can be checked by running this command:
`kubectl get cm base-configmap -o yaml -n core | grep VERSION`

(Conditional) Changing the Values of `frontend-ingress-controller-svc` and `portal-ingress-controller-svc`

After the CDF upgrade, the values for `frontend-ingress-controller-svc` and `portal-ingress-controller-svc` may have changed. Since you previously created a target group for ports 3000 and 5443 using the original `ingress-controller-svc` value, you must change the targets to match the new value.

To do this, in the target groups previously created for ports 3000 and 5443, de-register the targets. Then, add the targets back with the new values with the new value of `frontend-ingress-controller-svc` and `portal-ingress-controller-svc.`

Upgrading Deployed Capabilities on AWS

Follow the Checklist: Upgrading Your AWS Cluster to ensure a successful upgrade.

As part of the upgrade process, you must upgrade your deployed capabilities using the CDF Management Portal.

1. "Understanding the Upgrade Prerequisites " below
2. "Accepting the Certificate" on the next page
3. "Upload the Upgrade Bits" on the next page
4. "Upgrading Deployed Capabilities" on page 552

Micro Focus does not recommend that ArcSight Intelligence customers attempt to upgrade without consulting with Micro Focus as upgrading at this time will result in loss of baseline data. We are aware that this is inconvenient for customers, and we are currently working on addressing this limitation in an future release.

Understanding the Upgrade Prerequisites

Before upgrading, ensure these requirements are met.

- If you are using custom data identifiers for Intelligence, ensure that you back up the `logstash-config-pipeline` config map that is accessible through the Kubernetes dashboard.
- Ability to access the management portal on port 5443.
- The AWS client is configured.
- The Kubernetes command line tool (`kubectl`) is installed on the bastion and connected to your cluster.
- Run `kubectl get pods -A` to ensure all pods are running.

- Download the installation packages.
- **Before performing a suite upgrade,** ensure any existing `suite-upgrade-pod-arcsight-installer` deployment has been deleted using the following command:
  `kubectl delete deployment suite-upgrade-pod-arcsight-installer -n $(kubectl get namespaces | grep arcsight-installer | awk ' {print $1} ')`

The command will return an error if no upgrade has been done previously.

Accepting the Certificate

1. Browse to the management portal at https://<virtual_FQDN>:5443.
2. Click **DEPLOYMENT**, and select **Deployments.**
3. Click the **Three Dots** (Browse) on the far right and choose **Reconfigure**.
4. Accept the certificate.

Upload the Upgrade Bits

1. Upload the metadata for the upgrade version to your bastion host.

   ```
   arcsight-installer-metadata-uv.x.tar.gz
   ```

   For example: `arcsight-installer-metadata-22.1.0.37.tar`
2. Upload offline images of the upgrade version to your bastion host.

   ```
   {product}-uv.x.tgz
   ```

   For example: `transformationhub-3.6.0.37.tar`
3. Unpack the Arcsight Platform Cloud Installer. For example, this unpacks the installer in the /tmp folder.

   ```
   cd /tmp
   ```

   ```
   unzip arcsight-platform-cloud-installer-<VERSION>.zip
   ```
4. Unpack aws-scripts.zip from Arcsight Platform Cloud Installer using these commands:

   ```
   cd arcsight-platform-cloud-installer-<VERSION>
   ```

   ```
   unzip aws-scripts.zip
   ```

Upgrading Deployed Capabilities

1. Log in to the bastion where you downloaded the upgrade files.
2. Change to the following directory.

```
cd arcsight-platform-cloud-installer-<VERSION>/aws-scripts/scripts
```

3. Run the following commands to upload the images to the local Docker Registry. Use the `-F <image file>` option on the command line multiple times for each image to upload. Adjust the `-c 2` option up to half of your CPU cores in order to increase the speed of the upload.

```
./upload_images_to_ECR  -o {organization} -c 2 -F {unzipped-installer-dir}/images/fusion-x.x.x.x.tar -F {unzipped-installer-dir}/images/recon-x.x.x.x.tar
```

4. Add new metadata.

Make sure to copy the `arcsight-installer-metadata-x.x.x.x.tar` to the system where your web browser is running before performing the process below.

5. Browse to the management portal at https://<virtual_FQDN>:5443.
   a. Click **DEPLOYMENT>Metadata** and click **+ Add**.
   b. Select `arcsight-installer-metadata-x.x.x.x.tar` from your system. The new metadata is added to the system.
6. If Intelligence is deployed, you must label the worker nodes again. The **interset** label is now **intelligence**, the **interset-datanode** label is now **intelligence-datanode**, the **interset-namenode** label is now **intelligence-namenode**, and the **interset-spark** label is now **intelligence-spark**.
7. Start the upgrade process.
   a. Go to **DEPLOYMENT > Deployments**. Notice the number **1** in the red circle in the Update column.

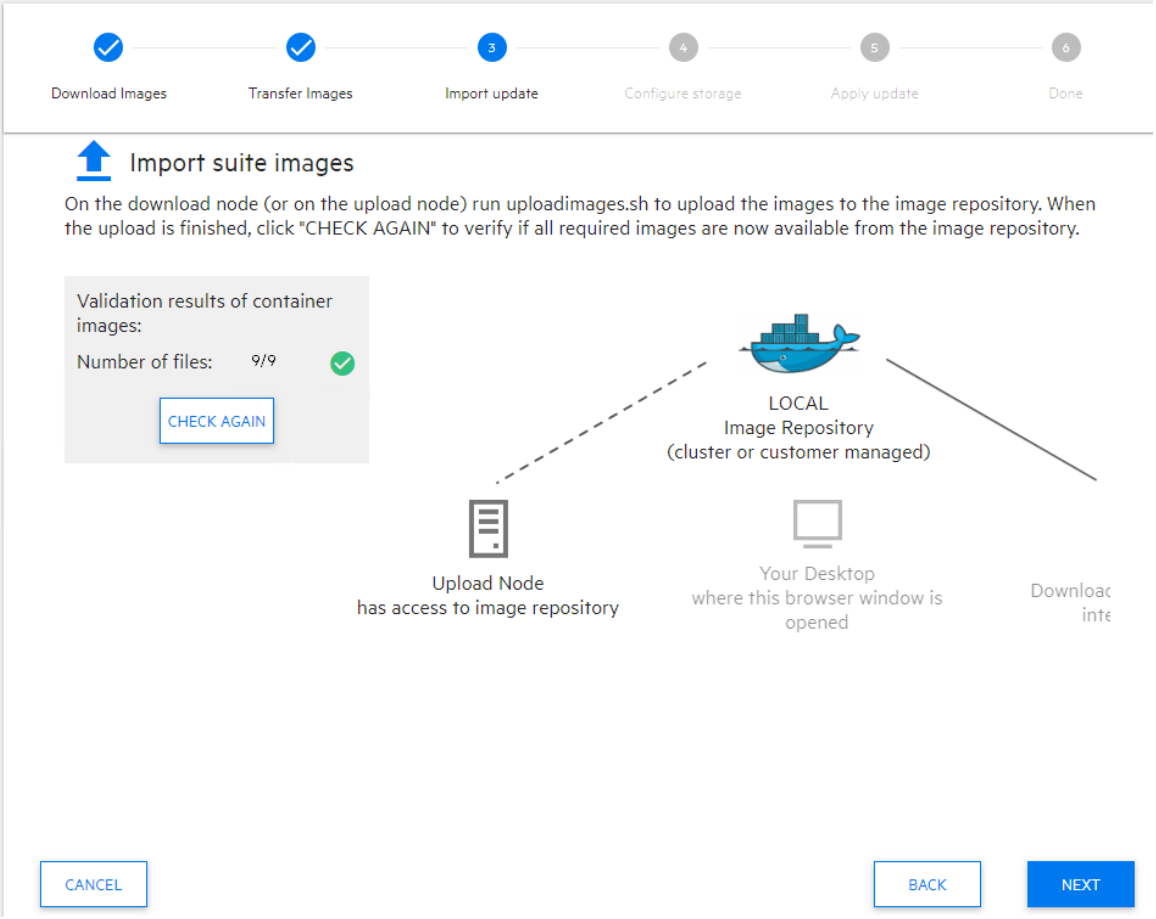
Minor version changes do not display like regular updates. (For example: 22.1.0.15 -> 22.1.0.16.)

   b. Click the red circle and select your recently added metadata to initiate the upgrade.
8. From the **Update to** page, click **NEXT** until you reach the **Import suite images** page.

When prompted to download or transfer images, you can simply click Next to skip the steps. You performed these steps earlier.

9. Ensure that the validation results of container images show a complete number of files.

   When you arrive at the Import suite images page, the images should already be imported, as you performed these steps earlier.



10. (Conditional) If you've deployed Transformation Hub, perform the following action:

    On the **Transformation Hub** tab, select whether you want to enable the generation of verification events for parsed field integrity checks used by Recon. If enabled, set the **verification event size** accordingly (default value is recommended).

    For more information about how the Event Integrity feature works, see Stream Processor Groups.

    When enabling the generation of verification events for parsed field integrity checks feature, consider the guidelines in Adjusting the Partition Count for New Kafka Topics.

11. Optionally, execute the following command to update your RE certificates:

```
./cdf-updateRE.sh write --re-ca=/tmp/latest/ca.cert.pem --re-
key=/tmp/latest/intermediate.key.pem --re-
crt=/tmp/latest/intermediate.cert.pem
```

For more information, see Updating RE Certificates.

12. Click **NEXT** until you reach the **Upgrade Complete** page.

> If it has been more that 12 hours since the cluster was installed, you will need to refresh the ECR credentials. See Refresh the ECR credentials in the K8s.

Cleaning Up the EFS Directory (AWS)

After installing the ArcSight Database and upgrading the Platform, you must clean up the EFS directory to remove instances of existing configuration data before you begin a fresh install.

> The ArcSight Database is a separate installation and is not included in the platform upgrade. For more information, see Formatting Instance Store Type Devices.

> You must backup necessary configuration data like custom dashboards, searches etc. to avoid data loss on account of the clean up. For more information, see Backing Up and Restoring Configuration Data for Deployed Capabilities.

To clean up the EFS directory, connect to one of your database nodes and run the following commands:

1. Remove the "Reporting" directory under EFS:

```
rm -rf /opt/arcsight-nfs/arcsight-volume/reporting
```

2. Remove the "rethinkdb" directory under EFS:

```
rm -rf /opt/arcsight-nfs/arcsight-volume/fusiondb/search/rethinkdb
```

3. Restart the pods that require rethinkdb:

```
kubectl get pods \
-n $( kubectl get namespaces | grep arcsight | cut -d ' ' -f1 ) \
--no-headers=true | awk '/fusion/{print $1}'| xargs
```

4. Delete the Fusion pods:

```
NSP=$( kubectl get namespaces | grep arcsight | cut -d ' ' -f1 )
kubectl delete -n  $NSP pod
```

Modify the (Connector) Producers and Consumers

After changing the CDF RE certificate, the events flow to consumers will be interrupted due to communication loss with the Transformation Hub. Existing non-containerized ArcMC

management of the Transformation Hub will also be affected.

- Connectors will cache events due to the change in the CDF RE certificate. If the certificate change is completed in a timely fashion, all events will be transmitted after the change.
- If the duration exceeds the available cache, the most recent events will be transmitted after the change.

As a result, the truststore and keystores in any existing consumers and producers must be modified to use the new updated certificate.

On all connector producers: Create new truststores and keystores using the new intermediate certificate used in updating the CDF RE certificate. Using either ArcMC or the command line, edit the each connector's Transformation Hub destination parameters and change the truststore, keystore paths, and passwords as required.

On all consumers: Modify existing truststores and keystores using the new intermediate certificate used in updating the CDF RE certificate.

On ArcMC: On ArcMC, perform the following steps to update the Transformation Hub certificate.

1. Select **Configuration Management > Bulk Operations.**
2. Click **Hosts**.
3. Select the Transformation Hub host.
4. Click **Update Cluster Details.**
5. Add the required parameters, then click **Save**.

ArcMC will display the new certificate after a few minutes of processing time.

Restarting the Event Consumers

If your ArcSight Platform deployment includes a capability that needs the ArcSight Database, such as Intelligence, complete the following procedure after upgrading the Platform.

1. (Conditional) If you have Intelligence deployed, complete the following steps:
   a. Log in to the Transformation Hub node.
   b. Run the following commands:

```
NS=$(kubectl get namespaces | awk '/arcsight/{print $1}')
```

```
kubectl scale statefulset interset-logstash -n $NS  --replicas=
{replica count}
```

   where {replica count} represents one of the following values:

- The number in the **READY** column that you captured in Step 6 of "Stopping the ESM and Intelligence Event Consumers."
- A new replica count value that you defined for Intelligence during the recent upgrade

(Optional) Adding Transformation Hub to Fusion ArcMC

Micro Focus recommends using the new Fusion ArcMC to manage your Transformation Hub.

As an optional step, you can add Transformation Hub to Fusion ArcMC, which will enable Fusion ArcMC to manage your Transformation Hub. See Configuring ArcMC to Manage a Transformation Hub for details.

Applying the CDF 2021.05 log4j Hotfix

Some deployments of, and upgrades to, CDF 2021.05/arcsight-platform-installer-22.1.x.x.zip require application of a hotfix to remediate the log4j vulnerability, which was discovered in 2021. The hotfix will upgrade IDM for CDF to use log4j 2.17.1, to prevent exploitation of the log4j vulnerabilities (CVE-2021-44228, CVE-2021-45046, CVE-2021-45105, CVE-2021-44832). The hotfix should be applied after an upgrade.

The hotfix applies to the following types of installations and upgrades:

- Any on-premises manual installation of 22.1.x or any on-premises manual upgrade to 22.1.1. (A manual installation or upgrade is one that does not use the ArcSight Installer.)

The CDF 2021.05 hotfix will be automatically applied during any on-premises installation or upgrade using the ArcSight Installer and this procedure can be skipped.

- Any CDF fresh installation or upgrade on AWS.
- Any CDF fresh installation or upgrade on Azure.

The log4j remediation hotfix does **NOT** apply to on-premises installations or upgrades performed automatically using the ArcSight Installer, as the hotfix is applied automatically by the installer. For such installations or upgrades these procedures can be skipped.

Hotfix File

The hotfix file is named arcsight-idm-hf-22.1.0-2.zip.

1. Get the file:
   - For a manual on-premises deployment/upgrade, the hotfix is bundled in `/<download_folder>/arcsight-platform-installer-22.1.x.x/installers/hotfix`.

- For an AWS or Azure deployment upgrade, obtain the file from the *on-premises* installation file directory at `/<download_folder>/arcsight-platform-installer-22.1.x.x/installers/hotfix`.

2. Copy the file:
   - For manual on-premises, copy the file to your master node.
   - For AWS, copy the file to your bastion.
   - For Azure, copy the file to your jump host.
3. Unzip the hotfix file. In the unzipped folder, run the following command with the '-e' argument (values: onprem, azure, aws) to apply the latest image.

```
# ./hotfix.sh -e <YOUR_ENV>
```

Verifying the Hotfix

1. Check the pod status by running the following command. It should be 'Running' as 2/2.

```
# kubectl get pods -A | grep idm
```

2. Check the image version by running the following command.

```
# kubectl get deployment/itom-idm -n core -o yaml | grep itom-idm:1.32.1-343
```

It should display as below:

```
image: <image-registry-url>/<org-name>/itom-idm:1.32.1-343
```

Rolling Back to the Previous Version

To roll back itom-idm to the previous version, run the following roll back commands :

```
# kubectl delete -f /tmp/cdf-itom-idm.yaml
```

```
# kubectl create -f /tmp/cdf-itom-idm.yaml
```

Upgrading to 22.1.2

Requires ArcSight Platform 22.1.0 or 22.1.1 to be installed in your environment.

To ensure that your environment has the latest fixes and enhancements, we recommend that you upgrade your ArcSight Platform 22.1.0 or 22.1.1 environment to 22.1.2. For more information about this release and to download the files, see the *Release Notes for ArcSight Platform 22.1.2*.

Perform the upgrade in the following order:

1. "Back Up Components and the Database" below
2. "Upgrade the Deployed Capabilities" below
3. "Upgrade the ArcSight Database" on page 560
4. "(Conditional) Restart Pods After Reconfiguring Single Sign-on Settings" on page 562

Back Up Components and the Database

Before upgrading to this release, ensure that you have a performed a backup of the ArcSight Platform and the ArcSight Database.

Upgrade the Deployed Capabilities

To upgrade the deployed capabilities, you will need image files for the patch except the `db-installer` file.

1. (Conditional) If you are using custom data identifiers for Intelligence, ensure that you back up the `logstash-config-pipeline` config map that is accessible through the Kubernetes dashboard.
2. (Conditional) **If you are upgrading an AWS or Azure environment from 22.1.0,** complete the following steps:
   a. Launch a terminal session and as a root user, log in to the node where NFS is present.
   b. Navigate to the following directory:

      ```
      cd  /<arcsight_nfs_vol_path>/interset/analytics/vertica_loader_sql/0/
      ```

   c. Execute the following command to create the 1.9.1.9 directory:

      ```
      mkdir 1.9.1.9
      ```

   d. Navigate to the following directory:

      ```
      cd <arcsight_nfs_vol_path>/interset/analytics/vertica_loader_sql/0
      ```

   e. Execute the following command to move the SQL loader scripts from `<arcsight_nfs_vol_path>/interset/analytics/vertica_loader_sql/0` to `<arcsight_nfs_vol_path>/interset/analytics/vertica_loader_sql/0/1.9.1.9`:

      ```
      mv *.md5 *.sql 1.9.1.9
      ```

   f. Execute the following command to grant permissions to the 1.9.1.9 directory:

      ```
      chown -R 1999:1999 1.9.1.9
      ```

3. Download the files as described in the *Release Notes for ArcSight Platform 22.1.2*.
4. Copy the downloaded files to their specific locations for your deployment:
   - *On premises:* to the master node
   - *AWS:* to bastion
   - *Azure:* to the jump host
5. (Conditional) For an **on-premises environment**, complete the procedures in "Upgrading Deployed Capabilities" on page 523 except for Step 11.
6. (Conditional) For an **AWS environment**, complete the following steps to upgrade the capabilities:
   a. Refresh ECR credentials.
   b. To upgrade the capabilities, follow the procedures in "Upgrading Deployed Capabilities on AWS" on page 550.

      You can ignore Steps 3 and 4 in Downloading the Upgrade Bits procedure, as well as Steps 6 and 10 in Upgrading Deployed Capabilities.
7. (Conditional) For an **Azure environment**, follow the procedures in "Upgrading Deployed Capabilities on Azure" on page 567.

   You can skip the Step 6 in "Upgrading Deployed Capabilities."
8. (Conditional) **If you are upgrading from 22.1.0,**, complete the following steps:
   a. Run Analytics to start the next analytics run. For more information, see Running Analytics on Demand in the Administrator's Guide for ArcSight Platform.
   b. During the analytics run, the 1.9.2.9 folder is created in the following directory with the default SQL loader scripts:

      ```
      cd <arcsight_nfs_vol_path>/interset/analytics/vertica_loader_
      sql/0/1.9.2.9
      ```
   c. (Conditional) If you have been using custom SQL loader scripts in 22.1.0, then the SQL loader scripts with inconsistent md5 sums between the current and previous versions are displayed in the Analytics logs. Perform the following steps to review and modify the SQL loader scripts:
      - Execute the following commands to check the logs of the analytics pod:

        ```
        export NS=$(kubectl get namespaces |grep arcsight|cut -d ' ' -f1)
        ```

```
PN=$(kubectl get pods -n $NS | grep -e 'interset-analytics' | awk '
{print $1}')
```

```
kubectl logs -f $PN -n $NS -c interset-analytics
```

- Review and add the necessary modifications to the new SQL loader scripts present in the following directory:

```
cd <arcsight_nfs_vol_path>/interset/analytics/vertica_loader_
sql/0/1.9.2.9
```

- Update the md5 files with the md5 sums corresponding to the modified SQL loader scripts in the following directory:

```
cd <arcsight_nfs_vol_path>/interset/analytics/vertica_loader_
sql/0/1.9.1.9
```

Analytics is triggered automatically after all the SQL loader scripts with inconsistent md5 sums are updated.

9. (Conditional) For **AWS** and **Azure** environments, if Kafka Manager is not accessible after the upgrade, then you should restart the *fusion-user-management* pod:

```
kubectl delete pod -n $(kubectl get namespaces | grep arcsight-installer
| awk ' {print $1} ') fusion-user-management-xxxxxxxxxx-xxxxx
```

where:

- `fusion-user-management-xxxxxxxxxx-xxxxx` is the unique name of the *fusion-user-management* pod

Upgrade the ArcSight Database

1. Download the database file, `db-installer_x.x.x-x.tar.gz`, to node 1 of the database cluster.

   For more information about the download files and to verify the signature file, see the *Release Notes for ArcSight Platform 22.1.2*.

2. To untar the file, run the following commands as user `root`.

   If you logged in as a non-root user and need to switch to a root user to perform this procedure, we've found that it's necessary to use the 'sudo su -' command (including the hyphen) to make the switch.

```
mkdir /opt/upgrade_files
```

```
cd /opt/upgrade_files
```

```
mv <path_to_db-installer>/db-installer_x.x.x-x.tar.gz /opt/upgrade_files/
```

```
tar xvfz db-installer_x.x.x-x.tar.gz
```

3. To stop event ingestion, run the following command:

```
/opt/arcsight-db-tools/kafka_scheduler stop
```

4. To start the upgrade, run the following commands:

```
./db_upgrade -c upgrade-utilities
```

```
./db_upgrade -c upgrade-db-rpm
```

> if an "**Incompatible version detected for current database**" message appears after the current database rpm version is listed in the output, please disregard it. It just means that your database had already been upgraded to the 11.0.2-4 version. Please skip the next step (s).

5. To complete the upgrade, run the following commands:

```
/opt/arcsight-db-tools/db_installer start-db
```

```
/opt/arcsight-db-tools/kafka_scheduler start
```

6. (Conditional) To use the database in FIPS mode, continue to "Enabling FIPS Mode on the Database Server" on page 709.
7. (Conditional) To check the number of events, and confirm that the environment kept receiving events correctly, go to the `/opt/arcsight-db-tools` path and execute the following command:

   `./Kafka_scheduler events`

Delete Old Metadata

After a successful upgrade to 22.1.2, remove any older patch metadata files from the CDF management portal, as follows:

1. On the CDF management portal, browse to **Deployment > Metadata**.
2. Delete any metadata files corresponding to versions prior to 22.1.2, such as 22.1.1.

(Conditional) Restart Pods After Reconfiguring Single Sign-on Settings

If you reconfigure the **Single Sign-on** settings in the CDF Management Portal after successfully upgrading to 22.1.2, ArcSight Platform might fail to display the Reports Portal and the SOAR feature. If this issue occurs, you should restart the following pods:

- soar-web-app
- reporting-web-app

Please allow the pods about 10 minutes to restart before you attempt to log in to the ArcSight Platform.

Upgrading Your Azure Deployment

Micro Focus does not recommend that ArcSight Intelligence customers attempt to upgrade without consulting with Micro Focus as upgrading at this time will result in loss of baseline data. We are aware that this is inconvenient for customers, and we are currently working on addressing this limitation in an future release.

This release significantly changes the ArcSight Database, you cannot upgrade the database. It must be installed as new. However, this release does allow you to deploy or upgrade Recon and Intelligence in your environment, as well as install either capability for the first time.

The upgrade process for ArcSight Platform deployed on Azure uses the CDF Management Portal. ArcSight Platform22.1.0 and later versions require ArcSight Fusion, and you must add Fusion to your deployment as part of the upgrade.

Checklist: Upgrading Your Azure Deployment

To ensure a successful on-premises upgrade, complete the following tasks in the listed order:

This release significantly changes the ArcSight Database, you cannot upgrade the database. It must be installed as new. However, this release does allow you to deploy or upgrade Recon and Intelligence in your environment, as well as install either capability for the first time.

	Task	See
❑	1. Download the files that you need to upgrade your environment.	"Downloading the Installation Packages for an Azure Deployment" on the next page

❑	2. (Conditional) To deploy Intelligence or Recon, install or upgrade the ArcSight Database.	"Installing the Database in Azure" on page 325
❑	3. (Conditional) To prevent both the original and newly installed databases from ingesting duplicate events, stop data ingestion on the original database.	"Checklist: Stopping Event Ingestion" on the next page
❑	4. Upgrade the CDF infrastructure.	"Upgrading the CDF Infrastructure on Azure" on the next page
❑	5. Authorize your browser to perform the upgrade. Then, Upgrade your deployed capabilities, such as Transformation Hub and Recon, which run on the CDF infrastructure. You must also deploy Fusion (required for ArcSight Platform 22.1.2 and later).	"Upgrading Deployed Capabilities on Azure" on page 567
❑	6. After you upgrade the ArcSight Platform and install the ArcSight Database, there are steps that you need to perform post upgrade to clean up the NFS directory.	"Cleaning Up the NFS Directory - Azure Deployment" on page 569
❑	7. Restarting the Event Consumers	"Restarting the Event Consumers" on page 570
❑	8. Apply the hotfix to remediate the log4j vulnerability.	"Applying the CDF 2021.05 log4j Hotfix" on page 570
❑	9. Apply security patches and latest updates to the deployed capabilities and the ArcSight Database	"Upgrading to 22.1.2" on page 572

Downloading the Installation Packages for an Azure Deployment

Use this procedure to download the packages for:

- **Installation:** Follow the Checklist: Planning to Deploy ArcSight Capabilities on Azureto ensure a successful installation.
- **Upgrade**: Follow the Checklist: Upgrading the Azure Deployment of ArcSight Platformto ensure a successful upgrade.

1. To identity the files to download to your secure network location, see Downloading and Installing the ArcSight Platform Capabilities in the Release Notes.
2. On a secure network location, download the installation packages for the CDF Installer and the product of your choice from the Software Licenses and Downloads portal .

This secure network location must be able to access Azure through the Azure Portal or the Azure Cloud Shell.

Checklist: Stopping Event Ingestion

Before upgrading your environment, complete the following procedures in the listed order:

	Task	See
❑	1. Stop the database event consumer (the Kafka Scheduler) so you can record its offsets	"Stopping the Database Event Consumer" on page 514
❑	2. (Conditional) If you have Intelligence deployed, monitor the Intelligence event consumers to stop them when they catch up to the database event consumer	"Stopping the Intelligence Event Consumers" on page 515
❑	3. Prevent the database from consuming duplicate events	"Preventing the Database from Consuming Duplicate Events" on page 517

Upgrading the CDF Infrastructure on Azure

In this step, you will upgrade the CDF infrastructure that you deployed on Azure. This infrastructure supports the ArcSight Platform and its capabilities, which you will upgrade after the CDF upgrade is complete.

Prerequisites

- You must be able to access the jump host VM as `root` in the Azure cluster. You can perform all upgrade steps from the jump host. For more information, see "Preparing the Jump Host Virtual Machine" on page 292.
- The Kubernetes command-line tool `kubectl` must be installed and connected to your cluster on the jump host.

It is possible to perform the upgrade from another host besides the jump host, but you must still have `kubectl` connected to your cluster and you must disable proxy settings.

Before proceeding with upgrade, ensure any existing suite-upgrade-pod-arcsight-installer deployment has been deleted using the following command, which will return an error if an upgrade has been performed previously:

```
kubectl delete deployment suite-upgrade-pod-arcsight-installer -n `kubectl
get namespaces | grep arcsight-installer | awk ' {print $1} '`
```

To upgrade CDF on Azure:

1. Execute the following command to ensure that all pods are running:

   ```
   kubectl get pods -A
   ```

2. From the "Downloading the Installation Packages for an On-Premises Deployment" on page 512 section, copy the CDF deployer `arcsight-platform-cloud-installer-x.x.x.x.zip` to the directory `/tmp/upgrade-download`.
3. Unzip the deployer package by running these commands.

   ```
   cd /tmp/upgrade-download
   unzip arcsight-platform-cloud-installer-x.x.x.x.zip
   ```

4. Upload new images to Azure Container Registry (ACR).
   a. Browse to the Azure management portal and open the ACR, then click **Access keys** > **Login Server**. You will need a user name and password to upload images. For more information, see "Uploading Product Images" on page 322
   b. Change to the deployer scripts directory:

      ```
      cd  arcsight-platform-cloud-installer-x.x.x.x/#
      unpack "cdf-deployer.zip"

      unzip cdf-deployer.zip
      cd cdf-deployer/scripts/
      ```

   c. Execute the script `uploadimages.sh` with the credentials from ACR:

      ```
      ./uploadimages.sh -o <your-org-name> -r <login-server> -u <user name>
      -p <password> -F /tmp/upgrade-download/arcsight-platform-cloud-
      installer-x.x.x.x/cdf-byok-images.tar -c 4
      ```

      The -o argument for orgnamemust be the same as the one used for the original installation.You can check your orgname with the following command: `kubectl get cm -n core base-configmap -o yaml | grep REGISTRY_ORGNAME:`
      For more information, see `uploadimages.sh --help`.

5. Delete the health probe and load balancing rules previously configured for port 5443.
6. Annotate and patch the nginx-ingress-controller-svc service by running the following 2 commands:

```
kubectl annotate service -n core nginx-ingress-controller-svc
service.beta.kubernetes.io/azure-load-balancer-internal=true
kubectl patch services nginx-ingress-controller-svc -p '{"spec":
{"type":"LoadBalancer","loadBalancerIP": "PUBLIC_IP"}}' -n core
```

where PUBLIC_IP is the IP address you configured during installation for your external access host. For example: 10.1.1.101.

7. Run the upgrade using the following steps.
   a. Ensure all PODs in the core namespaces are **Running** or **Completed**.

      ```
      kubectl get pods -n core
      ```

   b. Change to the `cdf-deployer` directory:

      ```
      cd cdf-deployer/
      ```

   c. Execute the upgrade script:

      ```
      ./upgrade.sh -u
      ```

   d. At the end of the upgrade, ensure that all pods are in a **Running** or **Completed** state by executing this command:

      ```
      kubectl get pods -A
      ```

8. The upgrade process will recreate Azure resources that will replace (and therefore break) any existing health probe and load balancing rules. Accordingly, you need to recreate all of the pre-upgrade health probe and load balancing rules, as follows:
   a. Find the IP address assigned to your external access host for the CDF by pinging it from the jump host with the following command:

      ```
      ping installer.arcsight.private.com
      ```

      Determine the hostname, if needed, by running the command:

      ```
      kubectl get cm -n core base-configmap -o yaml | grep EXTERNAL_ACCESS_
      HOST:
      ```

   b. Run the following commands to patch services:

      ```
      kubectl patch services -n core portal-ingress-controller-svc -p '
      {"spec":{"type":"LoadBalancer","loadBalancerIP": "PUBLIC_IP"}}'
      kubectl patch services -n core frontend-ingress-controller-svc -p '
      {"spec":{"type":"LoadBalancer","loadBalancerIP": "PUBLIC_IP"}}
      ```

      where PUBLIC_IP is the IP address you configured during installation for your external access host. For example: 10.1.1.101.

   c. Recreate the health probe and load balancer rules for port 443. For more information, see Configuring the Load Balancer.

Upgrading Deployed Capabilities on Azure

When the the CDF infrastructure upgrade is complete, you can upgrade your deployed capabilities, such as Transformation Hub, Recon, Intelligence, or other ArcSight Platform products that run on the CDF. This process will also deploy Fusion, which is required for ArcSight Platform 22.1.2 and later. (If you have already deployed Fusion, then this process upgrades it.)

- "Accepting the Browser Security Certificate" below
- "Upgrading Deployed Capabilities on Azure" above
- "Upgrading Deployed Capabilities" on the next page

Micro Focus does not recommend that ArcSight Intelligence customers attempt to upgrade without consulting with Micro Focus as upgrading at this time will result in loss of baseline data. We are aware that this is inconvenient for customers, and we are currently working on addressing this limitation in an future release.

Accepting the Browser Security Certificate

Before starting the upgrade process, you must accept the browser security certificate, which will authorize your browser to access the CDF management portal and perform the upgrade.

1. Log in to the CDF Management Portal.
2. Click **DEPLOYMENT** > **Deployments**.
3. Click **...** (browse), then select **Reconfigure**.
4. Accept the browser certificate.

Considerations for Upgrading Intelligence

Micro Focus does not recommend that ArcSight Intelligence customers attempt to upgrade without consulting with Micro Focus as upgrading at this time will result in loss of baseline data. We are aware that this is inconvenient for customers, and we are currently working on addressing this limitation in an future release.

If you have deployed Intelligence and are upgrading it, consider the performing the following steps on the NFS before upgrading the deployed capability:

- Ensure that you move your SQL loader scripts from: `/<arcsight_nfs_vol_path>/interset/analytics/vertica_loader_sql/0/<existing_folder_name>` to `/<arcsight_nfs_vol_path>/interset/analytics/vertica_loader_sql/0/1.<existing_folder_name>`.

- If you are using custom data identifiers, ensure that you back up the `logstash-config-pipeline` config map that is accessible through the Kubernetes dashboard.

Upgrading Deployed Capabilities

Before upgrading the deployed capabilities, confirm that any existing suite-upgrade-pod-installer deployment has been deleted by using the following command:

```
kubectl delete deployment suite-upgrade-pod-arcsight-installer -n `kubectl get namespaces | grep arcsight-installer | awk ' {print $1} '`
```

The command will return an error if no upgrade has been performed previously.

To upgrade your deployed capabilities:

1. Upload your metadata and product images to the ACR.
2. From the jump host, browse to the CDF management portal at https://<virtual_FQDN>:5443.
   a. Click **DEPLOYMENT > Metadata** and click **+ Add**.
   b. Browse to and select the metadata file `arcsight-installer-metadata-x.x.x.x.tar`. This process adds the new metadata to the system.
3. Start the upgrade process.
   a. Click **DEPLOYMENT > Deployments**. In the **Update** column, find the numeral **1** inside the red circle.

      Minor version changes do not display in the same way as regular updates. (For example: 22.1.0.15 -> 22.1.0.16.)

   b. Click the red circle, then select your recently-added metadata to initiate the upgrade.
4. From the **Update to** page, click **NEXT** until you reach the **Import suite images** page.

   When prompted to download or transfer images, you can click **NEXT** to skip the steps, since you performed these steps earlier.

5. Ensure that the validation results of container images show a complete number of files.

   On the **Import suite images** page, the process will already display your image files, as you performed these steps earlier.

6. (Conditional) If you've deployed Transformation Hub, perform the following action:

On the **Transformation Hub** tab, select whether you want to enable the generation of verification events for parsed field integrity checks used by Recon. If enabled, set the **verification event size** accordingly (default value is recommended).

For more information about how the Event Integrity feature works, see Stream Processor Groups.

When enabling the generation of verification events for parsed field integrity checks feature, consider the guidelines in Adjusting the Partition Count for New Kafka Topics.

7. Click **NEXT** until you reach the **Upgrade Complete** page.

Cleaning Up the NFS Directory - Azure Deployment

After installing the ArcSight Database and upgrading the Platform, you must clean up the NFS directory to remove instances of existing configuration data before you begin a fresh install.

The ArcSight Database is a separate installation and is not included in the platform upgrade. For more information, see Installing the Database in Azure.

You must back up necessary configuration data like custom dashboards, searches etc. to avoid data loss on account of the clean up. For more information, see Backing Up and Restoring Configuration Data for Deployed Capabilities.

To clean up the NFS directory, connect to one of your database nodes and run the following commands:

1. Remove the "Reporting" directory under NFS:

```
rm -rf /opt/arcsight-nfs/arcsight-volume/reporting
```

2. Remove the "rethinkdb" directory under NFS:

```
rm -rf /opt/arcsight-nfs/arcsight-volume/fusiondb/search/rethinkdb
```

3. Restart the pods that require rethinkdb:

```
kubectl get pods \
-n $( kubectl get namespaces | grep arcsight | cut -d ' ' -f1 ) \
--no-headers=true | awk '/fusion/{print $1}'| xargs
```

4. Delete the Fusion pods:

```
NSP=$( kubectl get namespaces | grep arcsight | cut -d ' ' -f1 )
kubectl delete -n  $NSP pod
```

Restarting the Event Consumers

If your ArcSight Platform deployment includes a capability that needs the ArcSight Database, such as Intelligence, complete the following procedure after upgrading the Platform.

1. (Conditional) If you have Intelligence deployed, complete the following steps:
   a. Log in to the Transformation Hub node.
   b. Run the following commands:

```
NS=$(kubectl get namespaces | awk '/arcsight/{print $1}')
```

```
kubectl scale statefulset interset-logstash -n $NS  --replicas=
{replica count}
```

   where {replica count} represents one of the following values:

   - The number in the **READY** column that you captured in Step 6 of "Stopping the ESM and Intelligence Event Consumers."
   - A new replica count value that you defined for Intelligence during the recent upgrade

Applying the CDF 2021.05 log4j Hotfix

Some deployments of, and upgrades to, CDF 2021.05/arcsight-platform-installer-22.1.x.x.zip require application of a hotfix to remediate the log4j vulnerability, which was discovered in 2021. The hotfix will upgrade IDM for CDF to use log4j 2.17.1, to prevent exploitation of the log4j vulnerabilities (CVE-2021-44228, CVE-2021-45046, CVE-2021-45105, CVE-2021-44832). The hotfix should be applied after an upgrade.

The hotfix applies to the following types of installations and upgrades:

- Any on-premises manual installation of 22.1.x or any on-premises manual upgrade to 22.1.1. (A manual installation or upgrade is one that does not use the ArcSight Installer.)

> The CDF 2021.05 hotfix will be automatically applied during any on-premises installation or upgrade using the ArcSight Installer and this procedure can be skipped.

- Any CDF fresh installation or upgrade on AWS.
- Any CDF fresh installation or upgrade on Azure.

The log4j remediation hotfix does **NOT** apply to on-premises installations or upgrades performed automatically using the ArcSight Installer, as the hotfix is applied automatically by the installer. For such installations or upgrades these procedures can be skipped.

Hotfix File

The hotfix file is named arcsight-idm-hf-22.1.0-2.zip.

1. Get the file:
   - For a manual on-premises deployment/upgrade, the hotfix is bundled in `/<download_folder>/arcsight-platform-installer-22.1.x.x/installers/hotfix`.
   - For an AWS or Azure deployment upgrade, obtain the file from the *on-premises* installation file directory at `/<download_folder>/arcsight-platform-installer-22.1.x.x/installers/hotfix`.
2. Copy the file:
   - For manual on-premises, copy the file to your master node.
   - For AWS, copy the file to your bastion.
   - For Azure, copy the file to your jump host.
3. Unzip the hotfix file. In the unzipped folder, run the following command with the '-e' argument (values: onprem, azure, aws) to apply the latest image.

```
# ./hotfix.sh -e <YOUR_ENV>
```

Verifying the Hotfix

1. Check the pod status by running the following command. It should be 'Running' as 2/2.

```
# kubectl get pods -A | grep idm
```

2. Check the image version by running the following command.

```
# kubectl get deployment/itom-idm -n core -o yaml | grep itom-idm:1.32.1-343
```

It should display as below:

```
image: <image-registry-url>/<org-name>/itom-idm:1.32.1-343
```

Rolling Back to the Previous Version

To roll back itom-idm to the previous version, run the following roll back commands :

```
# kubectl delete -f /tmp/cdf-itom-idm.yaml
```

```
# kubectl create -f /tmp/cdf-itom-idm.yaml
```

Upgrading to 22.1.2

Requires ArcSight Platform 22.1.0 or 22.1.1 to be installed in your environment.

To ensure that your environment has the latest fixes and enhancements, we recommend that you upgrade your ArcSight Platform 22.1.0 or 22.1.1 environment to 22.1.2. For more information about this release and to download the files, see the *Release Notes for ArcSight Platform 22.1.2*.

Perform the upgrade in the following order:

1. "Back Up Components and the Database" below
2. "Upgrade the Deployed Capabilities" below
3. "Upgrade the ArcSight Database" on page 575
4. "(Conditional) Restart Pods After Reconfiguring Single Sign-on Settings" on page 576

Back Up Components and the Database

Before upgrading to this release, ensure that you have a performed a backup of the ArcSight Platform and the ArcSight Database.

Upgrade the Deployed Capabilities

To upgrade the deployed capabilities, you will need image files for the patch except the `db-installer` file.

1. (Conditional) If you are using custom data identifiers for Intelligence, ensure that you back up the `logstash-config-pipeline` config map that is accessible through the Kubernetes dashboard.
2. (Conditional) **If you are upgrading an AWS or Azure environment from 22.1.0,** complete the following steps:
   a. Launch a terminal session and as a root user, log in to the node where NFS is present.
   b. Navigate to the following directory:

      ```
      cd  /<arcsight_nfs_vol_path>/interset/analytics/vertica_loader_sql/0/
      ```

   c. Execute the following command to create the 1.9.1.9 directory:

      ```
      mkdir 1.9.1.9
      ```

   d. Navigate to the following directory:

```
cd <arcsight_nfs_vol_path>/interset/analytics/vertica_loader_sql/0
```

e. Execute the following command to move the SQL loader scripts from `<arcsight_nfs_vol_path>/interset/analytics/vertica_loader_sql/0` to `<arcsight_nfs_vol_path>/interset/analytics/vertica_loader_sql/0/1.9.1.9`:

```
mv *.md5 *.sql 1.9.1.9
```

f. Execute the following command to grant permissions to the 1.9.1.9 directory:

```
chown -R 1999:1999 1.9.1.9
```

3. Download the files as described in the *Release Notes for ArcSight Platform 22.1.2*.
4. Copy the downloaded files to their specific locations for your deployment:
   - *On premises:* to the master node
   - *AWS:* to bastion
   - *Azure:* to the jump host
5. (Conditional) For an **on-premises environment**, complete the procedures in "Upgrading Deployed Capabilities" on page 523 except for Step 11.
6. (Conditional) For an **AWS environment**, complete the following steps to upgrade the capabilities:
   a. Refresh ECR credentials.
   b. To upgrade the capabilities, follow the procedures in "Upgrading Deployed Capabilities on AWS" on page 550.

   You can ignore Steps 3 and 4 in Downloading the Upgrade Bits procedure, as well as Steps 6 and 10 in Upgrading Deployed Capabilities.
7. (Conditional) For an **Azure environment**, follow the procedures in "Upgrading Deployed Capabilities on Azure" on page 567.

   You can skip the Step 6 in "Upgrading Deployed Capabilities."
8. (Conditional) **If you are upgrading from 22.1.0,**, complete the following steps:
   a. Run Analytics to start the next analytics run. For more information, see Running Analytics on Demand in the Administrator's Guide for ArcSight Platform.
   b. During the analytics run, the 1.9.2.9 folder is created in the following directory with the default SQL loader scripts:

```
cd <arcsight_nfs_vol_path>/interset/analytics/vertica_loader_
sql/0/1.9.2.9
```

c. (Conditional) If you have been using custom SQL loader scripts in 22.1.0, then the SQL loader scripts with inconsistent md5 sums between the current and previous versions are displayed in the Analytics logs. Perform the following steps to review and modify the SQL loader scripts:

- Execute the following commands to check the logs of the analytics pod:

```
export NS=$(kubectl get namespaces |grep arcsight|cut -d ' ' -f1)
```

```
PN=$(kubectl get pods -n $NS | grep -e 'interset-analytics' | awk '
{print $1}')
```

```
kubectl logs -f $PN -n $NS -c interset-analytics
```

- Review and add the necessary modifications to the new SQL loader scripts present in the following directory:

```
cd <arcsight_nfs_vol_path>/interset/analytics/vertica_loader_
sql/0/1.9.2.9
```

- Update the md5 files with the md5 sums corresponding to the modified SQL loader scripts in the following directory:

```
cd <arcsight_nfs_vol_path>/interset/analytics/vertica_loader_
sql/0/1.9.1.9
```

Analytics is triggered automatically after all the SQL loader scripts with inconsistent md5 sums are updated.

9. (Conditional) For **AWS** and **Azure** environments, if Kafka Manager is not accessible after the upgrade, then you should restart the *fusion-user-management* pod:

```
kubectl delete pod -n $(kubectl get namespaces | grep arcsight-installer
| awk ' {print $1} ') fusion-user-management-xxxxxxxxxx-xxxxx
```

where:

- `fusion-user-management-xxxxxxxxxx-xxxxx` is the unique name of the *fusion-user-management* pod

Upgrade the ArcSight Database

1. Download the database file, `db-installer_x.x.x-x.tar.gz`, to node 1 of the database cluster.

   For more information about the download files and to verify the signature file, see the *Release Notes for ArcSight Platform 22.1.2*.

2. To untar the file, run the following commands as user `root`.

   > If you logged in as a non-root user and need to switch to a root user to perform this procedure, we've found that it's necessary to use the 'sudo su -' command (including the hyphen) to make the switch.

   ```
   mkdir /opt/upgrade_files
   ```

   ```
   cd /opt/upgrade_files
   ```

   ```
   mv <path_to_db-installer>/db-installer_x.x.x-x.tar.gz /opt/upgrade_files/
   ```

   ```
   tar xvfz db-installer_x.x.x-x.tar.gz
   ```

3. To stop event ingestion, run the following command:

   ```
   /opt/arcsight-db-tools/kafka_scheduler stop
   ```

4. To start the upgrade, run the following commands:

   ```
   ./db_upgrade -c upgrade-utilities
   ```

   ```
   ./db_upgrade -c upgrade-db-rpm
   ```

   > if an "**Incompatible version detected for current database**" message appears after the current database rpm version is listed in the output, please disregard it. It just means that your database had already been upgraded to the 11.0.2-4 version. Please skip the next step (s).

5. To complete the upgrade, run the following commands:

   ```
   /opt/arcsight-db-tools/db_installer start-db
   ```

   ```
   /opt/arcsight-db-tools/kafka_scheduler start
   ```

6. (Conditional) To use the database in FIPS mode, continue to "Enabling FIPS Mode on the Database Server" on page 709.

7. (Conditional) To check the number of events, and confirm that the environment kept receiving events correctly, go to the `/opt/arcsight-db-tools` path and execute the following command:

   ```
   ./Kafka_scheduler events
   ```

Delete Old Metadata

After a successful upgrade to 22.1.2, remove any older patch metadata files from the CDF management portal, as follows:

1. On the CDF management portal, browse to **Deployment > Metadata**.
2. Delete any metadata files corresponding to versions prior to 22.1.2, such as 22.1.1.

(Conditional) Restart Pods After Reconfiguring Single Sign-on Settings

If you reconfigure the **Single Sign-on** settings in the CDF Management Portal after successfully upgrading to 22.1.2, ArcSight Platform might fail to display the Reports Portal and the SOAR feature. If this issue occurs, you should restart the following pods:

- soar-web-app
- reporting-web-app

Please allow the pods about 10 minutes to restart before you attempt to log in to the ArcSight Platform.

Chapter 9: Maintaining the Platform and Deployed Capabilities

This section describes the maintenance activities that you should perform for the Platform capabilities deployed in your environment. Unless otherwise specified, the procedure applies to both an on-premises and cloud deployment

Changing ArcSight Platform Configuration Properties

Reconfiguring properties causes the capabilities related to the property to stop and restart and this might cause operations underway to fail. Therefore, ensure that effected capabilities that cannot be easily retried are not running when you reconfigure any of these properties. For example, check the pod logs to see what operations are underway.

(For Intelligence) Check the Analytics pod logs to see if Analytics is already running by executing the following commands on any of the nodes:

```
export NS=$(kubectl get namespaces |grep arcsight|cut -d ' ' -f1)
kubectl -n $NS logs <interset-analytics-pod>
```

To change ArcSight Platform configuration properties:

1. Open a certified web browser.
2. Specify the following URL to log in to the CDF Management Portal: https://<cdf_masternode_hostname or virtual_ip hostname>:5443.
3. Select **Deployment > Deployments.**
4. Click **...** (Browse) on the far right and choose **Reconfigure**. A new screen will be opened in a separate tab.
5. Update configuration properties as needed.
6. Click **Save.**

All services in the cluster affected by the configuration change will be restarted (in a rolling manner) across the cluster nodes.

(For Intelligence) If you are specifying details under the **Hadoop File System (HDFS) Security** section, consider the following:

- (Conditional) If you are enabling Kerberos Authentication for the first time, then, before selecting **kerberos** in **Enable Authentication with HDFS Cluster**, ensure you configure the Kerberos Authentication. For more information, see Enabling and Configuring Kerberos Authentication.
- (Conditional) If you need to modify the Kerberos details, ensure that you first enable and configure Kerberos Authentication with the new Kerberos details.
- The Kerberos details that you provide in **Kerberos Domain Controller Server**, **Kerberos Domain Controller Admin Server**, **Kerberos Domain Controller Domain**, and **Default Kerberos Domain Controller Realm** will be considered only if you select **kerberos** in **Enable Authentication with HDFS Cluster**. They are not valid if you select **simple**.
- If you are enabling Kerberos Authentication, then you must enable **Enable Secure Data Transfer with HDFS Cluster**.
  If you disable **Enable Secure Data Transfer with HDFS Cluster**, the database and HDFS will use the same communication standard as Intelligence 6.2.
- If you have enabled **Enable Secure Data Transfer with HDFS Cluster** and if you have a non-collocated database cluster, log in to a database node, and copy the RE CA certificate from the CDF master node to /etc/pki/ca-trust/extracted/pem/tls-ca-bundle.pem. Repeat this step on all the database nodes.

(For Intelligence) The updated configuration properties related to analytics and Hadoop File System (HDFS) security in the **Intelligence** tab are considered only in the next Intelligence analytics run. If you have modified **Enable Secure Data Transfer with HDFS Cluster** and if HDFS namenode enters the safe mode when you run analytics, perform step 8 of Configuring HDFS Security in CDF

Understanding Labels and Pods

During installation, you apply labels, which are associated with the deployed capabilities, to the worker nodes in the Kubernetes cluster. The labels indicate to Kubernetes the various types of workloads that can run on a specific host system. Based on the labels, Kubernetes then assigns pods to the nodes to provide functions, tasks, and services. Each pod belongs to a specific namespace in the CDF Management portal. On occasion, you might need to restart pods or reconfigure the environment by moving labels to different nodes, thus reassigning the workload of the pods.

When using the CDF Management Portal, the label format is `<label name>:yes`. However, when using the kubectl command line the label format is `<label name>=yes`.

Adding Labels to Worker Nodes

Depending on the capabilities that you deploy, you must assign a set of labels to the Worker Nodes. Each of the following sections defines the pods and their associated capabilities that areinstalled for an assigned label.

To avoid issues caused by conflicting label assignments, review the following considerations.

- **Labeling for the Intelligence capability**
  - The HDFS NameNode, which corresponds with the `intelligence-namenode:yes` label, should run on one worker node only. The worker node must match the hostname or IP address that you provided in the **HDFS NameNode** field in the **CDF Management Portal** > **Configure/Deploy** page > **Intelligence**.
  - Assign the label for Spark2, `intelligence-spark:yes`, to the same worker nodes where you placed the `intelligence-datanode:yes` label.
- For Transformation Hub's Kafka and ZooKeeper, make sure that the number of the nodes you have labeled corresponds to the number of worker nodes in the Kafka cluster and the number of worker nodes running Zookeeper in the Kafka cluster properties from the pre-deployment configuration page. The default number is 3 for a Multiple Worker deployment.
- Although ESM Command Center, Recon, Intelligence, and SOAR all require Fusion, you do not need to assign the label for Fusion to more than one worker node.

fusion:yes

The Fusion capability includes many of the core services needed for your deployed products, including the Dashboard and user management; all deployed capabilities require Fusion. Add the `fusion:yes` label to the Worker Nodes where you want to run the associated pods. For high availability, add this label to multiple worker nodes.

Pod	Description	Namespace	Associated Capability
esm-acc-web-app	Manages the user interface for ESM Command Center. The interface connects to an ESM Manager server running outside the Kubernetes cluster.	arcsight-installer	ESM Command Center
esm-web-app	Manages how ESM Command Center links to main navigation of the Platform user interface.	arcsight-installer	ESM Command Center
esm-widgets	Manages the dashboards and widgets that are designed to incorporate data from ESM. The widgets connect to an ESM Manager server running outside of the Kubernetes cluster. For example, when you start this pod, it installs the provided *How is my SOC running?* dashboard.	arcsight-installer	ESM Command Center
fusion-arcmc-web-app	Manages the user interface for ArcSight Management Center. A `fusion-arcmc-web-app:yes` label can optionally be applied to one or more worker nodes to control where this pod runs. Otherwise, it falls back to running on a node where the `fusion:yes` label is applied.	arcsight-installer	Fusion
fusion-common-doc-web-app	Provides the context-sensitive user guides for Fusion (the Platform), Recon, and Reporting.	arcsight-installer	Fusion
fusion-metadata-web-app	Manages the REST API for the metadata of the Dashboard feature.	arcsight-installer	Fusion
fusion-dashboard-web-app	Manages the framework, including the user interface, for the Dashboard feature.	arcsight-installer	Fusion
fusion-db-monitoring-web-app	Manages the REST API for the database monitoring function.	arcsight-installer	Fusion
fusion-db-search-engine	Provides APIS to access data in the ArcSight Database. NOTE: This pod requires communication outside of the Kubernetes cluster.	arcsight-installer	Fusion
fusion-metadata-rethinkdb	Manages the RethinkDB database, which stores information about a user's preferences and configuration.	arcsight-installer	Fusion
fusion-single-sign-on	Manages the SSO service that enables users to log in to any of the deployed capabilities and the consoles for ArcSight Intelligence, SOAR, and ESM Command Center.	arcsight-installer	Fusion
fusion-ui-services	Manages the framework, including the user interface, for the primary navigation functions in the user interface.	arcsight-installer	Fusion

Pod	Description	Namespace	Associated Capability
fusion-user-management	Manages the framework, including the user interface, for the user management function.	arcsight-installer	Fusion
soar-message-broker	Manages SOAR JMS messages.	arcsight-installer	Fusion
soar-web-app	Manages SOAR services and capabilities.	arcsight-installer	Fusion
soar-db-init	Manages the SOAR DB schema and creates associated structures.	arcsight-installer	Fusion
soar-jms-migration	Manages the migration of SOAR JMS messages to the next release.	arcsight-installer	Fusion
soar-frontend	Manages the SOAR user interface.	arcsight-installer	Fusion
soar-widgets	Deploys SOAR Fusion widgets.	arcsight-installer	Fusion
interset-widgets	Manages the widgets that are designed to incorporate data from ArcSight Intelligence.	arcsight-installer	Intelligence
layered-analytics-widgets	Manages and installs the widgets that can incorporate data from multiple capabilities. For example, the provided *Entity Priority* widget connects to Intelligence and ESM Command Center server outside the Kubernetes cluster to display entity data.	arcsight-installer	Layered Analytics
recon-analytics	Manages the backend of Outlier Analytics; the user interface for Outlier Analytics is managed by the recon-search-web-app pod.	arcsight-installer	Recon
recon-search-web-app	Manages the Search, Lookup lists, and Data Quality Dashboard functions, as well as the user interface for Outlier Analytics.	arcsight-installer	Recon
reporting-web-app	Manages the REST API and user interface for the Reporting feature. NOTE: This pod requires communication outside of the Kubernetes cluster.	arcsight-installer	Recon
recon-search-and-storage-web-app	Manages the configuration of and sends events to storage groups.	arcsight-installer	Recon

intelligence:yes

Add the `intelligence:yes` label to Worker Nodes where you want to run the pods that manage functions and services for the ArcSight Intelligence capability. For high availability, add

this label to multiple worker nodes.

Pod	Description	Namespace	Associated Capability
elasticsearch-data	Manages the Elasticsearch functions that store all raw events for Intereset Analytics and provide all data that drives the user interface.	arcsight-installer	Intelligence
elasticsearc-master	Manages the Elasticsearch services.	arcsight-installer	Intelligence
h2	Stores user identities required to authenticate and authorize users.	arcsight-installer	Intelligence
interset-analytics	Determines the individual baselines , then discovers and ranks devisions from those baselines for the Intelligence Analytics feature. This pod requires communication outside of the Kubernetes cluster.	arcsight-installer	Intelligence
interset-api	Manages the REST API that the Intelligence user interface uses to gather the Intelligence Analytics results. This pod requires communication outside of the Kubernetes cluster.	arcsight-installer	Intelligence
interset-exports	Generates the PDF reports of organization risks and the users involved in risky behaviors.	arcsight-installer	Intelligence
interset-logstash	Manages Logstash, which collects raw events from Transofrmation Hub and sends them to Elasticsearch for indexing.	arcsight-installer	Intelligence
interset-spark-config-file-server	Hosts a file server to provide configuration files for Spark3 to consume.	arcsight-installer	Intelligence
interset-ui	Manages the user interface that displays the Intelligence Analytics results and the raw data in the Intelligence dashboard	arcsight-installer	Intelligence
intelligence-arcsightconnector-api	Manages APIs related to licensing support and provides Fusion menu registration for Intelligence.	arcsight-installer	Intelligence

Pod	Description	Namespace	Associated Capability
intelligence-tuning-api	Manages APIs that tune the Intelligence Analytics metadata that can change the Intelligence Analytics results. This pod requires communication outside of the Kubernetes cluster.	arcsight-installer	Intelligence
searchmanager-api	Manages APIs that provide administrative tools related to Elasticsearch and the search capability in general.	arcsight-installer	Intelligence
searchmanager-engine	Manages jobs that provide administrative tools related to Elasticsearch and the search capability in general. This pod requires communication outside of the Kubernetes cluster.	arcsight-installer	Intelligence

intelligence-datanode:yes

Add the `intelligence-datanode:yes` label to Worker Nodes where you want to run the pods that manage HDFS services for the ArcSight Intelligence capability.

Pod	Description	Namespace	Associated Capability
hdfs-datanode	Manages how HDFS stores the results of Intelligence Analytics searches before transferring them to the ArcSight database. The HDFS Datanodes contain blocks of HDFS files.	arcsight-installer	Intelligence

intelligence-namenode:yes

Add the `intelligence-namenode:yes` label to a Worker Node for the HDFS NameNode.

Place this label on one worker node only. The worker node and the hostname or IP address in the **HDFS NameNode** field in the **Intelligence** tab of the **CDF Management Portal** must match.

Pod	Description	Namespace	Associated Capability
hdfs-namenode	Manages how the HDFS NameNode stores the location of all HDFS files distributed across the cluster.	arcsight-installer	Intelligence

intelligence-spark:yes

Add the `intelligence-spark:yes` label to Worker Nodes where you want to run the Analytics services for the ArcSight Intelligence capability. For high availability, add this label to multiple

worker nodes. To reduce network traffic, add the label to the same worker nodes where you placed the `intelligence-datanode:yes` label.

Pod	Description	Namespace	Associated Capability
Spark2	Launches when users run the Intelligence Analytics feature. Spark2 generates multiple pods, changing the names of the pods according to the different phases of the analytics tasks.	arcsight-installer	Intelligence

kafka:yes

Add the `kafka:yes` label to Worker Nodes where you want to run the Kafka Broker functions and services for the Transformation Hub capability.

Ensure that you assign this label to the same quantity of nodes that you specified for the `# of Kafka broker nodes in the Kafka cluster` setting in the **CDF Management Portal** > **Configure/Deploy** > **Transformation Hub** > **Kafka and Zookeeper Configuration**. The default number is 3.

Pod	Description	Namespace	Associated Capability
th-kafka	Manages the Kafka Broker, to which publishers and consumers connect so they can exhange messages over Kafka. NOTE: This pod requires communication outside of the Kubernetes cluster.	arcsight-installer	Transformation Hub

th-platform:yes

Add the `th-platform:yes` label to Worker Nodes where you want to run the Kafka Manager, schema registry, and WebServices for the Transformation Hub capability. For high availability, add this label to multiple worker nodes.

Pod	Description	Namespace	Associated Capability
th-kafka-manager	Provides the user interface that allows the Kafka Manager to manage the Kafka Brokers.	arcsight-installer	Transformation Hub
th-schemaregistry	Provides the scheme registry that is used for managing the schema of data in Avro format. NOTE: This pod requires communication outside of the Kubernetes cluster.	arcsight-installer	Transformation Hub

Pod	Description	Namespace	Associated Capability
th-web-service	Manages the WebServices module of Transformation Hub. WebServices provides the API that ArcMC uses to retrieve data. NOTE: This pod requires communication outside of the Kubernetes cluster to receive client requests from and initiate connections to ArcMC.	arcsight-installer	Transformation Hub

th-processing:yes

Add the `th-processing:yes` label to Worker Nodes where you want to run services that manage processing for the Transformation Hub capability. For high availability, add this label to multiple worker nodes.

Pod	Description	Namespace	Associated Capability
th-c2av-processor	Manages the instances that convert CEF messages on the topic th-cef to Avro on the topic th-arcsight-avro. The quantity of instances depends on the number of partition in the th-cef topic and load. The default is 0 instances.	arcsight-installer	Transformation Hub
th-cth	Manages up to 50 instances of connectors in Transformation Hub that distribute the load of data received from collectors by creating a consumer group that is based on the source top and destination and topic names.	arcsight-installer	Transformation Hub
th-c2av-processor-esm	Manages the instances that convert CEF messges on the topic mf-event-cef-esm-filtered to Avro on the topic mf-eent-avro-emsfiltered. The quantity of instances depends on the number of partition in the th-cef topic and load. The default is 0 instances.	arcsight-installer	Transformation Hub
th-routing-processor-group	Manages the routing rules for topics. Use ArcMC to configure the rules.	arcsight-installer	Transformation Hub

zk:yes

Add the `th-zookeeper:yes` label to Worker Nodes where you want to Kafka Zookeeper for the Transformation Hub capability.

Ensure that you assign this label to the same quantity of nodes that you specified for the `# of Zookeeper nodes in the Zookeper cluster` setting in the **CDF Management Portal** > **Configure/Deploy** > **Transformation Hub** > **Kafka and Zookeeper Configuration**. The default number is 3.

Pod	Description	Namespace	Associated Capability
th-zookeeper	Manages Kafka Zookeeper, which stores metadata about partitions and brokers.	arcsight-installer	Transformation Hub

Understanding the Pods that Do Not Have Labels

The Platform includes several pods that are not associated with a deployed capability and thus do not require a label. The installation process automatically creates these pods.

Pod	Description	Namespace
autopass-lm	Manages the Autopass service, which tracks license keys.	arcsight-installer
itom-pg-backup	Performs backup of the PostgreSQL database.	arcsight-installer
suite-reconf-pod-arcsight-intaller	Manages the Reconfiguration features in the CDF Management Portal.	arcsight-installer

Understanding Pods that Run Master Nodes

The Platform includes pods that run master nodes.

Pod	Description	Namespace
itom-postgresql-default	Manages the PostgreSQL database, which stores information for SOAR, ArcMC, CDF status, and license keys.	
idm	Manages user authentication and authorization for the CDF Management Portal.	core
nginx-ingress-controller	Provides the proxy web server that end-users need to connect to the deployed capabilities. By default, server uses HTTPS and port 443. NOTE: This pod requires communication outside of the Kubernetes cluster.	arcsight-installer

Managing CDF Logs

CDF uses Fluentd to collect and gather logs for CDF system components, Docker containers, and Kubernetes. After collection, CDF exports the logs to a remote destination that you

configure before CDF installation.

This section contains the following topics:

About CDF Logs

The CDF logs consist of the following:

- **Container logs:** Logs for all Kubernetes workloads.
- **System logs:** Logs for the journal, including logs from `kubelet.service` and `docker.service`.
- **Application logs:** Logs for all applications. The applications must write their logs to a shared volume, which is mounted to the `itom-logging-vol` persistent volume.

A Kubernetes Daemon Set fluentd pod runs as an instance of the Fluentd collector and forwarder on each node. Each Fluentd pod gets its configuration from fluentd in the ConfigMap. By default, Fluentd collects local log files and saves them to the `itom-logging-vol` persistent volume.

Log Retention

By default, CDF retains logs (both on the cluster nodes and on the `itom-logging-vol` persistent volume) for two days. To change the log retention, follow these steps:

1. Set an environment variable to define whether you will change the log retention on the cluster nodes or on the itom-logging-vol persistent volume. To do this, run one of the following commands:

- To configure the retention of logs on cluster nodes:

```
cm_name=logrotate-node-level
```

- To configure the retention of logs on the itom-logging-vol persistent volume:

```
cm_name=itom-logrotate
```

2. Run the following command to set the log retention period:

```
kubectl patch cm ${cm_name} -n core -p "{\"data\":
{\"logrotate.properties\":$(kubectl get cm ${cm_name} -n core -o json |
jq  '.data."logrotate.properties"' | sed -e 's/-mtime +[0-9]\{1,\}/-mtime
+&lt;retention_days&gt;/')}}"
```

where <retention_days> is the number of days for which you want to retain logs. For example, to retain logs for 10 days, run the following command:

```
kubectl patch cm ${cm_name} -n core -p "{\"data\":
{\"logrotate.properties\":$(kubectl get cm ${cm_name} -n core -o json |
jq  '.data."logrotate.properties"' | sed -e 's/-mtime +[0-9]\{1,\}/-mtime
+10/')}}"
```

Log Rotation and Deletion

Logs are either rotated or deleted, depending on the logging mechanism of the components. Files in directories which are defined by CDF will be deleted after a certain period according to the log retention configuration. These files and directories include:

- $CDF_HOME/log
- Directories that are mounted to the itom-logging persistent volume

Configurations listed below prefixed by “SCRIPT_DELETE_*” control the log deletion strategy of the above directories.

Run the following commands to determine the NFS server and NFS path to which the itom-logging persistent volume is mounted:

```
kubectl get pv itom-logging -o json|$CDF_HOME/bin/jq -r '.spec.nfs.server'
kubectl get pv itom-logging -o json|$CDF_HOME/bin/jq -r '.spec.nfs.path'
```

Some system log files in the /var/log/ directory will be rotated after a certain period according to the log rotation configuration. By default, CDF only rotates the /var/log/messages log file.

The configurations that start with “SYSLOG_*” listed below controls the log-rotation strategy of the `/var/log` directories.

Changing the Log Rotation or Deletion

The log rotate configurations are specified in the `itom-logrotate` ConfigMap file on the `itom-logging-vol` persistent volume and in the `logrotate-node-level` ConfigMap on cluster nodes.

1. Set an environment variable to define whether you will change the log retention on the cluster nodes or on the itom-logging-vol persistent volume. To do this, run one of the following commands:
   - To configure the retention of logs on cluster nodes:

     ```
     cm_name=logrotate-node-level
     ```

     To configure the retention of logs on the `itom-logging-vol` persistent volume:

     ```
     cm_name=itom-logrotate
     ```
2. Run the following command to change the default log rotate configuration:

   ```
   kubectl edit cm ${cm_name} -n core
   ```

   The new log rotate configuration takes effect automatically in about 2 minutes.

Additional ConfigMap Parameters

Some of the parameters listed below may not exist in the `itom-logrotate` and `logrotate-node-level` ConfigMaps. However, you can add them to the ConfigMaps with to configure some log rotate functions. Below are the details for the parameters in the `itom-logrotate` and `logrotate-node-level` ConfigMap instances:

Parameter	Description
`SCRIPT_DELETE_INSTALL_UPGRADE`	Delete the logs files of CDF install and upgrade. By default, this parameter is set to `false`. That means the log files of CDF install and upgrade will be retained. Micro Focus suggests that you use the default value for this parameter because these log files require little space and are useful for your troubleshooting.
`SCRIPT_DELETE_LOG_SURVIVAL`	Files will be removed if it meets the configured rule. For example, `-mtime +2` means logs will be removed that last for more than 2 days. `-size +51200k` means logs will be removed whose size is larger than 51200 KB. For format, refer to Linux command `find`.

Parameter	Description

SCRIPT_DELETE_CRONINTERVAL	The interval of file check for delete. The supported formats: `@hourly`, `@daily`, `@weekly`, `@monthly` `@yearly`, or in the cron job format: `0 0 * * * *`. A brief explanation of the format is shown here.
	`# ┌─────────────── second (0 - 59)`
	`# │ ┌─────────────── minute (0 - 59)`
	`# │ │ ┌─────────────── hour (0 - 23)`
	`# │ │ │ ┌─────────────── day of the month (1 - 31)`
	`# │ │ │ │ ┌─────────────── month (1 - 12)`
	`# │ │ │ │ │ ┌─────────────── day of the week (0 - 6) (Sunday to Saturday;`
	`7 is also Sunday on some systems)`
	`# │ │ │ │ │ │`
	`# │ │ │ │ │ │`
	`# │ │ │ │ │ │`
	`# * * * * * * command to execute`
	Refer to https://godoc.org/github.com/robfig/cron for more details.

SCRIPT_DELETE_FILE_ENDINGS	The file with specified endings will be deleted immediately on each checking for delete. For example, `txt tmp` means every log file named `*.txt` or `*.tmp` will be deleted, no matter if they meet the configured `SCRIPT_DELETE_LOG_SURVIVAL` rule or not.
SYSLOG_ROTATE_FILES	Files that need to be rotated under `/var/log`. Each file name must be separated by a white-space character. Default value: `messages`.
SYSLOG_ROTATE_INTERVAL	The rotate interval of the system log files. Supported values: `every hourly, daily, weekly, monthly, yearly`. Default value: `daily.`
SYSLOG_MAX_SIZE_OF_FILE	Log files are rotated when the files are larger than the specified size, or before the system logs are rotated within the specified time interval. Default value: `500M`
SYSLOG_MAX_ROTATE_OF_FILE	The log rotated time before the files are being removed. Value must be 0 or 1. 0: old versions are removed rather than rotated. 1: (default) old versions are rotated.
SYSLOG_ROTATE_MODE	The rotate mode of the log file. Default value: `copytruncate`
SYSLOG_ROTATE_PARAMETERS	The `logrotate` command line parameters. Supported values: `v`: Verbose. `d`: Debug (Logrotate will be emulated but never executed). `f`: Force.
SYSLOG_MIN_SIZE_OF_FILE	The minimum size of log files to be rotated.

Parameter	Description
SYSLOG_ ROTATE_ DATEFORMAT	Specify the date text extension for rotated logs using the notation similar to `strftime` function. Only `%Y %m %d %H` and `%s` specifiers are allowed. The default value is `-%Y%m%d` except hourly, which uses `-%Y%m%d%H` as its default value.
SYSLOG_ ROTATE_MAXAGE	The maximum time for the logs to be removed. This parameter is only checked if the log file is about to be rotated.
SYSLOG_ ROTATE_ COMPRESS	Compress rotated log files using gzip. The supported values are: `compress` and `nocompress` (default value).

Configuring the Automatic Log Cleanup Settings

The log automatic log cleanup settings are specified by the following parameters in the itom-logrotate ConfigMap file on the itom-logging-vol persistent volume and in the logrotate-node-level ConfigMap on cluster nodes.
SCRIPT_DELETE_FILE_ENDINGS
SCRIPT_DELETE_INSTALL_UPGRADE
SCRIPT_DELETE_LOG_SURVIVAL
SCRIPT_DELETE_CRONINTERVAL

To configure the automatic log cleanup settings, follow these steps:

1. Set an environment variable to define whether you will change the log retention on the cluster nodes or on the itom-logging-vol persistent volume. To do this, run one of the following commands:

   To configure the retention of logs on cluster nodes:

   ```
   cm_name=logrotate-node-level
   ```

   To configure the retention of logs on the itom-logging-vol persistent volume:

   ```
   cm_name=itom-logrotate
   ```

2. Run the following command to configure the parameter settings:

   ```
   kubectl edit cm ${cm_name} -n core
   ```

   Below is an example of the parameter settings in the itom-logrotate and logrotate-node-level ConfigMaps:

   ```
   SCRIPT_DELETE_FILE_ENDINGS=""
   SCRIPT_DELETE_INSTALL_UPGRADE="false"
   SCRIPT_DELETE_LOG_SURVIVAL="-mtime +2"
   SCRIPT_DELETE_CRONINTERVAL="@daily"
   ```

Configuring the System Log Settings

You can configure the log settings of the `/var/log/` system log directory by using the following parameters in the `itom-logrotate` and `logrotate-node-level` ConfigMaps:

```
SYSLOG_ROTATE_FILES
SYSLOG_ROTATE_INTERVAL
SYSLOG_MAX_SIZE_OF_FILE
SYSLOG_MAX_ROTATE_OF_FILE
SYSLOG_ROTATE_MODE
SYSLOG_ROTATE_PARAMETERS
SYSLOG_MIN_SIZE_OF_FILE
SYSLOG_ROTATE_DATEFORMAT
SYSLOG_ROTATE_MAXAGE
SYSLOG_ROTATE_COMPRESS
```

The following is an example of the parameter settings in the `itom-logrotate` and `logrotate-node-level` ConfigMaps:

```
SYSLOG_ROTATE_FILES="messages"
SYSLOG_ROTATE_INTERVAL="daily"
SYSLOG_MAX_SIZE_OF_FILE="500M"
SYSLOG_MAX_ROTATE_OF_FILE=1
SYSLOG_ROTATE_MODE="copytruncate"
SYSLOG_ROTATE_PARAMETERS="v"
SYSLOG_MIN_SIZE_OF_FILE="1M"
SYSLOG_ROTATE_DATEFORMAT="-%Y%m%d%H"
SYSLOG_ROTATE_MAXAGE=7
SYSLOG_ROTATE_COMPRESS="nocompress"
```

Installation Log Locations

During CDF installation, all the installation logs are located under the `$TMP_FOLDER`. When the installation is complete, the installation logs are under `$CDF_HOME/log/scripts/install`. Otherwise, the logs are under `$TMP_FOLDER`. An example of an installation log filename: `install-20200316234235.log`.

For an `arcsight-install` installation, all the installation logs are located under the `$TMP_FOLDER`. When a silent installation is complete, the logs are located under `$CDF_HOME/log/scripts/install` (for CDF installation) and `$CDF_HOME/log/scripts/silent-install` (for ArcSight Suite installation and node extension). An example of the silent installation log filename would be: `silent-install-20200317104001.log`. Otherwise, the logs are located under the `$TMP_FOLDER`.

Log Rotation of Docker Services

CDF supports log rotation of Docker services. Log rotation is enabled by default. The default maximum log file size is 10 MB. The maximum number of log files is five. To change the maximum log file size and maximum log file number, follow these steps:

1. Run the following commands to open the docker file:

```
cd $CDF_HOME/cfg
vim docker
```

2. Change the values of the max-size and max-file variables in the DOCKER_LOG_OPTS parameter. For example:

```
DOCKER_LOG_OPTS="--log-driver=json-file --log-opt
labels=io.kubernetes.container.name,io.kubernetes.pod.uid --log-opt max-
size=12m --log-opt max-file=6"
```

3. Run the following command to restart Docker and enable the changes:

```
systemctl restart docker
```

Micro Focus recommends that you use the default maximum log size number and maximum log file number. Don't set a large number for the max-size and max-file variables. Overly large maximum size and maximum file numbers may affect the free disk space.

Log and Trace Model

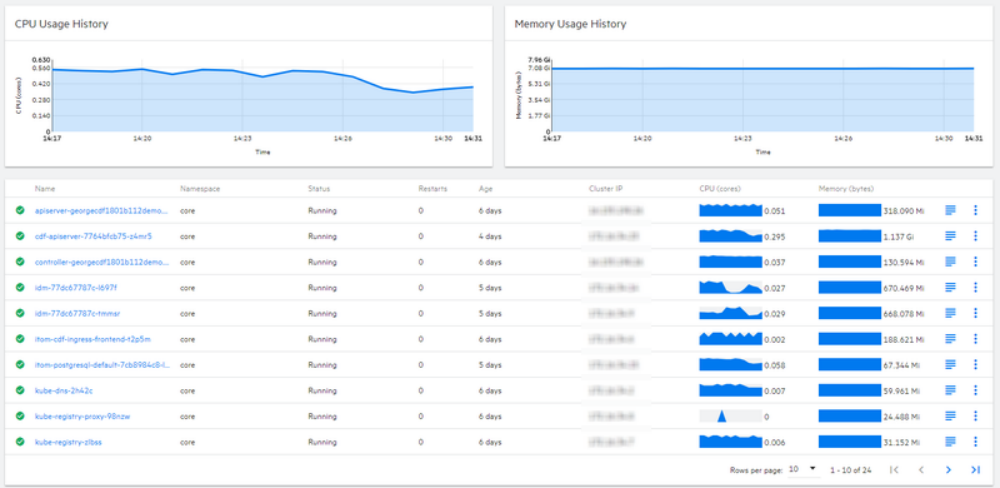
The following is recommended for the log and trace model.

- Pay attention to the log level, and don't unnecessarily enable tracing or debug parameters.
- Pay attention to log rotation and switching.

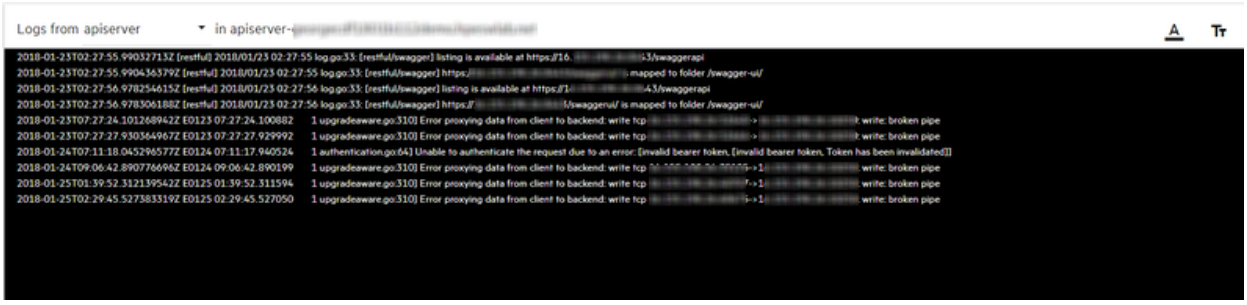
Accessing Pod Logs

To access the log for a particular pod, do the following:

1. In the CDF Management Portal, click **CLUSTER > Dashboard > Workloads > Pods**.
2. In the list of pods, click the pod for which you wish to view logs.



3. In the Pod area, click **View logs**. A page that resembles the following is displayed:



All suite logs are stored within a persistent volume, so that logs will persist even if the pods go down.

Configuring Log Levels

You can configure the log level as desired for troubleshooting purposes.

To change the log level:

1. Browse to the management portal at https://<virtual_FQDN>:5443, or at https://<master_node1_FQDN>:5443.
2. Click **DEPLOYMENT**, and select **Deployments.**
3. Click the **Three Dots** (Browse) on the far right and choose **Reconfigure**. A new screen will be opened in a separate tab.
4. Under the appropriate capability tabs there are log level configuration options for each component. Select the appropriate value to update the Log Levels. The change goes into effect automatically.

Uninstalling and Reinstalling the Platform

This section describes Platform uninstall and reinstall activities in your environment. Unless otherwise specified, the procedure applies to both an on-premises and cloud deployment.

Undeploying a Capability

It's possible that you might want to undeploy, or remove, one or more capabilities without uninstalling the ArcSight Platform. For example, you might want to remove Intelligence to resolve an issue with its environment. You can redeploy the capability later.

1. To remove the capabilities that you want to undeploy, complete the following steps:
   a. Log in to the CDF Management Portal.
   b. Navigate to **Deployment** > **Deployments**.
   c. In the ⋮ menu, select **Change**.
   d. Select the capabilities that you want to remove.

   > Be aware that most capabilities require Fusion and Transformation Hub. Thus, if you remove these two capabilities, the capabilities that depend on them cannot function appropriately.

   > You must back up Fusion secrets before you attempt to undeploy and redeploy Fusion; restore backed up Fusion secrets after too.

   e. Click **Next**, and then follow the wizard to complete the process (continue clicking **Next**).

2. To check that all pods are in Running status, run the following command:

```
kubectl get pods --all-namespaces
```

3. (Optional) To back up Fusion secrets before you undeploy Fusion, use the following command to locate and save them:

```
echo $(kubectl get secret rethink-secret -n $( kubectl get namespaces |
grep arcsight | cut -d ' ' -f1) -o json | jq '.data["rethink-password"]'
| sed 's/"//g' ) > rethink-secret-bkp
```

```
echo $(kubectl get secret reporting-secret -n $( kubectl get namespaces |
grep arcsight | cut -d ' ' -f1) -o json | jq '.data["reporting-
password"]' | sed 's/"//g' ) > reporting-secret-bkp
```

```
echo $(kubectl get secret acs-secret-db -n $( kubectl get namespaces |
grep arcsight | cut -d ' ' -f1) -o json | jq '.data["dbuser-pwd"]' | sed
's/"//g' ) > acs-db-secret-bkp
```

```
echo $(kubectl get secret acs-svc-secret -n $( kubectl get namespaces |
grep arcsight | cut -d ' ' -f1) -o json | jq '.data["acs-svc-password"]'
| sed 's/"//g' ) > acs-svc-secret-bkp
```

4. (Optional) To restore Fusion secrets after you redeploy Fusion, complete the following steps:
   a. Locate the secrets that you backed up in the previous step.
   b. To restore the secrets, run the following commands:

```
export RETHINK_SECRET=$(cat rethink-secret-bkp); echo $(kubectl get
secret rethink-secret -n $( kubectl get namespaces | grep arcsight |
cut -d ' ' -f1) -o json | jq '.data["rethink-password"]=env.RETHINK_
SECRET' | kubectl apply -f -)
```

```
export REPORTING_SECRET=$(cat reporting-secret-bkp); echo $(kubectl
get secret reporting-secret -n $( kubectl get namespaces | grep
arcsight | cut -d ' ' -f1) -o json | jq '.data["reporting-
password"]=env.REPORTING_SECRET' | kubectl apply -f -)
```

```
export ACS_DB_SECRET=$(cat acs-db-secret-bkp); echo $(kubectl get
secret acs-secret-db -n $( kubectl get namespaces | grep arcsight |
cut -d ' ' -f1) -o json | jq '.data["dbuser-pwd"]=env.ACS_DB_SECRET' |
kubectl apply -f -)
```

```
export ACS_SVC_SECRET=$(cat acs-svc-secret-bkp); echo $(kubectl get
secret acs-svc-secret -n $( kubectl get namespaces | grep arcsight |
cut -d ' ' -f1) -o json | jq '.data["acs-svc-password"]=env.ACS_SVC_
SECRET' | kubectl apply -f -)
```

Uninstalling Installed Products and CDF from an On-Premises Installation

You have several options for uninstallation of CDF and your installed products from an on-premises installation. Each of these options is explained in detail below.

- You can uninstall any or all installed products.
- In addition to uninstalling installed products, you can also uninstall CDF but leave your cluster resources in place. Perform this option if you plan to re-use the cluster and re-install CDF later.

  - If CDF was installed using the ArcSight Installer, then use the ArcSight Installer to uninstall CDF.
  - If CDF was installed manually, then manually uninstall CDF.
- You can uninstall products and CDF as above, and then destroy all resources created during platform setup. Only perform this option when the cluster is no longer needed.

To uninstall CDF from your on-premises installation using the ArcSight Installer:

The ArcSight Platform Uninstaller simplifies the uninstall procedure.

Prerequisites

Ensure the following before you begin:

- Verify that the ArcSight Platform Installer was used to deploy the original cluster.
- The uninstaller requires all installed software images and the `yaml` configuration file to reside in appropriate locations on the master node used during the original installation.

You are now ready to begin the uninstallation.

1. On the master node:
   a. Log in to the master node and execute the uninstall command from `/opt/arcsight-platform-installer-x.x.x.x/`

      ```
      #./arcsight-install -c /opt/my-install-config.yaml --cmd uninstall
      ```

   b. Execute the following commands to remove these directories from `/opt`.

      ```
      # rm -rf /opt/arcsight/
      ```

      ```
      # rm -rf /opt/arcsight-db-tools/
      ```

      ```
      # rm -rf /opt/containerd/
      ```

   c. Execute the following commands to remove these directories from `/root`.

      ```
      # rm -rf /root/.docker/
      ```

      ```
      # rm -rf /root/.kube/
      ```

      ```
      (Conditional) # rm -rf /root/.ssh/
      ```

   d. Manually delete the `/opt/arcsight` directory.
2. On the CDF worker nodes:

a. Execute the following commands to these current installation directories from `/opt`

```
# rm -rf /opt/arcsight/
```

```
# rm -rf /opt/containerd/
```

b. Execute the following commands to remove these directories from `/root`

```
# rm -rf /root/.kube/
```

```
(Conditional) # rm -rf /root/.ssh/
```

c. Manually delete the `/opt/arcsight` directory.

3. On each ArcSight Database node:

a. Execute the following commands to remove the `arcsight-database` directory as follows:

```
# rm -rf /opt/arcsight-db-tools
```

```
(Conditional) # rm -rf /root/.ssh/
```

4. On the NFS nodes:

a. Execute the following commands to remove the `arcsight-nfs` directory :

```
# rm -rf /opt/arcsight-nfs
```

b. Execute the following command to remove all content (related to `/opt/arcsight-nfs`) from the `exports` file:

```
# vi /etc/exports
```

c. Execute the following command to unexport all directories deleted in the previous step:

```
# exportfs -ra
```

d. (Conditional) Execute the following commands to remove the `.ssh/` directory from `/root`:

```
# rm -rf /root/.ssh/
```

You can now proceed to uninstall your installed products.

To uninstall CDF from your AWS installation manually:

1. On the master node and all worker nodes :
   a. Execute the uninstall command from `/opt/arcsight/kubernetes/`:

   ```
   # ./uninstall.sh
   ```

   b. Manually delete the `/opt/arcsight` directory.
2. On the master node, uninstall the ArcSight Database:
   a. Execute the uninstall command from `/opt/arcsight-db-tools`:

   ```
   ./db_installer uninstall
   ```

3. On each ArcSight Database node:
   a. Execute the following command to remove the `/opt/arcsight-db-tools` directory as follows:

   ```
   # rm -rf /opt/arcsight-db-tools
   ```

   b. (Conditional) Execute the following command to remove `.ssh/` directory from `/root`:

   ```
   # rm -rf /root/.ssh/
   ```

You can now proceed to uninstall your installed products.

To uninstall installed products:

If you are also uninstalling CDF, then prior to uninstalling your products, perform the uninstallation of CDF first (using either the ArcSight Installer or manually), and then return here to proceed with uninstalling your products.

1. Log in to the master node and become `root`.
2. Get the names of all namespaces by running the command:
   `kubectl get namespaces`

For example:

```
kubectl get namespaces
```

```
NAME                      STATUS   AGE
```

```
arcsight-installer-blk62   Active   41m
```

```
core                      Active   48m
```

```
default                   Active   84m
```

```
kube-public               Active   84m
```

```
kube-system               Active   84m
```

3. Delete the product namespaces you wish to delete, and the core namespace by running the command:
   `kubectl delete namespace <namespace name>`

For example:

```
kubectl delete namespace arcsight-installer-blk62
```

```
namespace "arcsight-installer-blk62" deleted
```

```
kubectl delete namespace core
```

```
namespace "core" deleted
```

Your own product namespace will have the name format `arcsight-installer-XXXXX`.

4. Wait for the selected namespaces to be deleted before continuing.
5. Get the names of all PVs (persistent volumes) by running the command:
   `kubectl get pv`

For example:

```
kubectl get pv
```

```
NAME                                    CAPACITY   ACCESS MODES   RECLAIM
POLICY   STATUS
```

```
arcsight-installer-blk62-arcsight-volume   30Gi       RWX            Retain
        Released
```

```
arcsight-installer-blk62-db-backup-vol     1Mi        RWX            Retain
        Released
```

```
db-single                                  10Gi       RWX            Retain
        Released
```

```
itom-logging                               1Mi        RWX            Retain
        Released
```

```
itom-vol                                   5Gi        RWX            Retain
        Released
```

6. Delete all PVs by running the following command for each PV:
   `kubectl delete pv <PV_name>`

```
kubectl delete pv arcsight-installer-blk62-arcsight-volume
```

```
persistentvolume "arcsight-installer-blk62-arcsight-volume" deleted
```

```
kubectl delete pv arcsight-installer-blk62-db-backup-vol
```

```
persistentvolume "arcsight-installer-blk62-db-backup-vol" deleted
```

```
kubectl delete pv db-single
```

```
persistentvolume "db-single" deleted
```

```
kubectl delete pv itom-logging
```

```
persistentvolume "itom-logging" deleted
```

```
kubectl delete pv itom-vol
```

```
persistentvolume "itom-vol" deleted
```

7. Clear the data from your NFS volumes by connecting with SSH and clearing (but **not** deleting) all exported directories.

If you deleted the SSH inbound rule, you will need to add it again to be able to SSH to your NFS.

Disposal of Cluster Resources

The procedures detailed above will leave your cluster resources intact. CDF and applications can be re-installed again on the cluster, either now or in the future, without having to re-create these resources.

If instead the cluster is no longer needed, you can safely destroy all resources created earlier for CDF and your installed applications. Consult the on-premises documentation for details on how to destroy resources.

Uninstalling Installed Products and CDF from Azure

You have several options for uninstallation of CDF and your installed products from Azure. Each of these options is explained in detail below.

- You can uninstall any or all installed products.
- In addition to uninstalling installed products, you can also uninstall CDF but leave your cluster resources in place. Perform this option if you plan to re-use the cluster and re-install CDF later.
  - If CDF was installed using the ArcSight Installer, then use the ArcSight Installer to uninstall CDF.
  - If CDF was installed manually, then manually uninstall CDF.
- You can uninstall products and CDF as above, and then destroy all resources created during platform setup. Only perform this option when the cluster is no longer needed.

To uninstall CDF from your Azure installation using the ArcSight Installer:

The ArcSight Platform Uninstaller simplifies the uninstall procedure.

Prerequisites

Ensure the following before you begin:

- Verify that the ArcSight Platform Installer was used to deploy the original cluster.
- The uninstaller requires all installed software images and the `yaml` configuration file to reside in appropriate locations on the jump host used during the original installation.

You are now ready to begin the uninstallation.

1. On the jump host:
   a. Log in to the jump host and execute the uninstall command from `/opt/arcsight-platform-installer-x.x.x.x/`

   ```
   #./arcsight-install -c /opt/my-install-config.yaml --cmd uninstall
   ```

   b. Execute the following commands to remove these directories from `/opt`.

   ```
   # rm -rf /opt/arcsight/
   ```

   ```
   # rm -rf /opt/arcsight-db-tools/
   ```

   ```
   # rm -rf /opt/containerd/
   ```

   c. Execute the following commands to remove these directories from `/root`.

   ```
   # rm -rf /root/.docker/
   ```

   ```
   # rm -rf /root/.kube/
   ```

```
(Conditional) # rm -rf /root/.ssh/
```

d. Manually delete the `/opt/arcsight` directory.

2. On the CDF worker nodes:
   a. Execute the following commands to these current installation directories from `/opt`

```
# rm -rf /opt/arcsight/
```

```
# rm -rf /opt/containerd/
```

   b. Execute the following commands to remove these directories from `/root`

```
# rm -rf /root/.kube/
```

```
(Conditional) # rm -rf /root/.ssh/
```

   c. Manually delete the `/opt/arcsight` directory.
3. On each ArcSight Database node:
   a. Execute the following commands to remove the `arcsight-database` directory as follows:

```
# rm -rf /opt/arcsight-db-tools
```

```
(Conditional) # rm -rf /root/.ssh/
```

4. On the NFS nodes:
   a. Execute the following commands to remove the `arcsight-nfs` directory :

```
# rm -rf /opt/arcsight-nfs
```

   b. Execute the following command to remove all content (related to `/opt/arcsight-nfs`) from the `exports` file:

```
# vi /etc/exports
```

   c. Execute the following command to unexport all directories deleted in the previous step:

```
# exportfs -ra
```

   d. (Conditional) Execute the following commands to remove the `.ssh/` directory from `/root`:

```
# rm -rf /root/.ssh/
```

You can now proceed to uninstall your installed products.

To uninstall CDF from your Azure installation manually:

1. On the jump host and all worker nodes :
   a. Execute the uninstall command from `/opt/arcsight/kubernetes/`:

   ```
   # ./uninstall.sh
   ```

   b. Manually delete the `/opt/arcsight` directory.
2. On the jump host, uninstall the ArcSight Database:
   a. Execute the uninstall command from `/opt/arcsight-db-tools`:

   ```
   ./db_installer uninstall
   ```

3. On each ArcSight Database node:
   a. Execute the following command to remove the `/opt/arcsight-db-tools` directory as follows:

   ```
   # rm -rf /opt/arcsight-db-tools
   ```

   b. (Conditional) Execute the following command to remove `.ssh/` directory from `/root`:

   ```
   # rm -rf /root/.ssh/
   ```

You can now proceed to uninstall your installed products.

To uninstall installed products:

If you are also uninstalling CDF, then prior to uninstalling your products, perform the uninstallation of CDF first (using either the ArcSight Installer or manually), and then return here to proceed with uninstalling your products.

1. Log in to the jump host and become `root`.
2. Get the names of all namespaces by running the command:
   `kubectl get namespaces`

For example:

```
kubectl get namespaces
```

```
NAME                        STATUS   AGE
```

```
arcsight-installer-blk62    Active   41m
```

```
core                        Active   48m
```

```
default                   Active   84m
```

```
kube-public               Active   84m
```

```
kube-system               Active   84m
```

3. Delete the product namespaces you wish to delete, and the core namespace by running the command:
   `kubectl delete namespace <namespace name>`

For example:

```
kubectl delete namespace arcsight-installer-blk62
```

```
namespace "arcsight-installer-blk62" deleted
```

```
kubectl delete namespace core
```

```
namespace "core" deleted
```

Your own product namespace will have the name format `arcsight-installer-XXXXX`.

4. Wait for the selected namespaces to be deleted before continuing.
5. Get the names of all PVs (persistent volumes) by running the command:
   `kubectl get pv`

For example:

```
kubectl get pv
```

```
NAME                                    CAPACITY   ACCESS MODES   RECLAIM
POLICY   STATUS
```

```
arcsight-installer-blk62-arcsight-volume   30Gi       RWX            Retain
        Released
```

```
arcsight-installer-blk62-db-backup-vol     1Mi        RWX            Retain
        Released
```

```
db-single                                  10Gi       RWX            Retain
        Released
```

```
itom-logging                               1Mi        RWX            Retain
        Released
```

```
itom-vol                                    5Gi        RWX            Retain
        Released
```

6. Delete all PVs by running the following command for each PV:
   `kubectl delete pv <PV_name>`

```
kubectl delete pv arcsight-installer-blk62-arcsight-volume
```

```
persistentvolume "arcsight-installer-blk62-arcsight-volume" deleted
```

```
kubectl delete pv arcsight-installer-blk62-db-backup-vol
```

```
persistentvolume "arcsight-installer-blk62-db-backup-vol" deleted
```

```
kubectl delete pv db-single
```

```
persistentvolume "db-single" deleted
```

```
kubectl delete pv itom-logging
```

```
persistentvolume "itom-logging" deleted
```

```
kubectl delete pv itom-vol
```

```
persistentvolume "itom-vol" deleted
```

7. Clear the data from your NFS volumes by connecting with SSH and clearing (but **not** deleting) all exported directories.

If you deleted the SSH inbound rule, you will need to add it again to be able to SSH to your NFS.

Disposal of Cluster Resources

The procedures detailed above will leave your cluster resources intact. CDF and applications can be re-installed again on the cluster, either now or in the future, without having to re-create these resources.

If instead the cluster is no longer needed, you can safely destroy all resources created earlier for CDF and your installed applications. Consult the Azure documentation for details on how to destroy resources.

Uninstalling Installed Products and CDF from AWS

You have several options for uninstallation of CDF and your installed products from AWS. Each of these options is explained in detail below.

- You can uninstall any or all installed products.
- In addition to uninstalling installed products, you can also uninstall CDF but leave your cluster resources in place. Perform this option if you plan to re-use the cluster and re-install CDF later.
  - If CDF was installed using the ArcSight Installer, then use the ArcSight Installer to uninstall CDF.
  - If CDF was installed manually, then manually uninstall CDF.
- You can uninstall products and CDF as above, and then destroy all resources created during platform setup. Only perform this option when the cluster is no longer needed.

To uninstall CDF from your AWS installation using the ArcSight Installer:

The ArcSight Platform Uninstaller simplifies the uninstall procedure.

Prerequisites

Ensure the following before you begin:

- Verify that the ArcSight Platform Installer was used to deploy the original cluster.
- The uninstaller requires all installed software images and the `yaml` configuration file to reside in appropriate locations on the bastion used during the original installation.

You are now ready to begin the uninstallation.

1. On the bastion:
   a. Log in to the bastion and execute the uninstall command from `/opt/arcsight-platform-installer-x.x.x.x/`

   ```
   #./arcsight-install -c /opt/my-install-config.yaml --cmd uninstall
   ```

   b. Execute the following commands to remove these directories from `/opt`.

   ```
   # rm -rf /opt/arcsight/
   ```

   ```
   # rm -rf /opt/arcsight-db-tools/
   ```

   ```
   # rm -rf /opt/containerd/
   ```

   c. Execute the following commands to remove these directories from `/root`.

```
# rm -rf /root/.docker/
```

```
# rm -rf /root/.kube/
```

```
(Conditional) # rm -rf /root/.ssh/
```

d. Manually delete the `/opt/arcsight` directory.

2. On the CDF worker nodes:
   a. Execute the following commands to these current installation directories from `/opt`

```
# rm -rf /opt/arcsight/
```

```
# rm -rf /opt/containerd/
```

   b. Execute the following commands to remove these directories from `/root`

```
# rm -rf /root/.kube/
```

```
(Conditional) # rm -rf /root/.ssh/
```

   c. Manually delete the `/opt/arcsight` directory.
3. On each ArcSight Database node:
   a. Execute the following commands to remove the `arcsight-database` directory as follows:

```
# rm -rf /opt/arcsight-db-tools
```

```
(Conditional) # rm -rf /root/.ssh/
```

4. On the NFS nodes:
   a. Execute the following commands to remove the `arcsight-nfs` directory :

```
# rm -rf /opt/arcsight-nfs
```

   b. Execute the following command to remove all content (related to `/opt/arcsight-nfs`) from the `exports` file:

```
# vi /etc/exports
```

   c. Execute the following command to unexport all directories deleted in the previous step:

```
# exportfs -ra
```

d. (Conditional) Execute the following commands to remove the `.ssh/` directory from `/root`:

```
# rm -rf /root/.ssh/
```

You can now proceed to uninstall your installed products.

To uninstall CDF from your AWS installation manually:

1. On the bastion and all worker nodes :
   a. Execute the uninstall command from `/opt/arcsight/kubernetes/`:

   ```
   # ./uninstall.sh
   ```

   b. Manually delete the `/opt/arcsight` directory.
2. On the bastion, uninstall the ArcSight Database:
   a. Execute the uninstall command from `/opt/arcsight-db-tools`:

   ```
   ./db_installer uninstall
   ```

3. On each ArcSight Database node:
   a. Execute the following command to remove the `/opt/arcsight-db-tools` directory as follows:

   ```
   # rm -rf /opt/arcsight-db-tools
   ```

   b. (Conditional) Execute the following command to remove `.ssh/` directory from `/root`:

   ```
   # rm -rf /root/.ssh/
   ```

You can now proceed to uninstall your installed products.

To uninstall installed products:

If you are also uninstalling CDF, then prior to uninstalling your products, perform the uninstallation of CDF first (using either the ArcSight Installer or manually), and then return here to proceed with uninstalling your products.

1. Log in to the bastion and become `root`.
2. Get the names of all namespaces by running the command:
   `kubectl get namespaces`

For example:

```
kubectl get namespaces
```

```
NAME                          STATUS   AGE
arcsight-installer-blk62      Active   41m
core                          Active   48m
default                       Active   84m
kube-public                   Active   84m
kube-system                   Active   84m
```

3. Delete the product namespaces you wish to delete, and the core namespace by running the command:
   `kubectl delete namespace <namespace name>`

For example:

```
kubectl delete namespace arcsight-installer-blk62
namespace "arcsight-installer-blk62" deleted
kubectl delete namespace core
namespace "core" deleted
```

Your own product namespace will have the name format `arcsight-installer-XXXXX`.

4. Wait for the selected namespaces to be deleted before continuing.
5. Get the names of all PVs (persistent volumes) by running the command:
   `kubectl get pv`

For example:

```
kubectl get pv
NAME                                        CAPACITY   ACCESS MODES   RECLAIM
POLICY   STATUS
arcsight-installer-blk62-arcsight-volume    30Gi       RWX            Retain
        Released
arcsight-installer-blk62-db-backup-vol      1Mi        RWX            Retain
        Released
```

```
db-single                                  10Gi       RWX            Retain
        Released
```

```
itom-logging                               1Mi        RWX            Retain
        Released
```

```
itom-vol                                   5Gi        RWX            Retain
        Released
```

6. Delete all PVs by running the following command for each PV:
   `kubectl delete pv <PV_name>`

```
kubectl delete pv arcsight-installer-blk62-arcsight-volume
```

```
persistentvolume "arcsight-installer-blk62-arcsight-volume" deleted
```

```
kubectl delete pv arcsight-installer-blk62-db-backup-vol
```

```
persistentvolume "arcsight-installer-blk62-db-backup-vol" deleted
```

```
kubectl delete pv db-single
```

```
persistentvolume "db-single" deleted
```

```
kubectl delete pv itom-logging
```

```
persistentvolume "itom-logging" deleted
```

```
kubectl delete pv itom-vol
```

```
persistentvolume "itom-vol" deleted
```

7. Clear the data from your NFS volumes by connecting with SSH and clearing (but **not** deleting) all exported directories.

If you deleted the SSH inbound rule, you will need to add it again to be able to SSH to your NFS.

Disposal of Cluster Resources

The procedures detailed above will leave your cluster resources intact. CDF and applications can be re-installed again on the cluster, either now or in the future, without having to re-create these resources.

If instead the cluster is no longer needed, you can safely destroy all resources created earlier for CDF and your installed applications. Consult the AWS documentation for details on how to destroy resources.

Reinstalling the Platform

If you uninstalled Platform and plan to reinstall Platform and the deployed capabilities in the same cluster, perform the following steps before reinstalling Platform and Intelligence:

1. Launch a terminal session and as a root user, log in to the node where NFS is present.
2. Delete the NFS directory:

```
rm -rf /opt/arcsight-nfs
```

3. Launch a terminal session and then log in to the node where the Kubernetes hostpath is present.
4. Navigate to the following directory:

```
cd /opt/arcsight/
```

5. Delete the following directory:

```
rm -rf k8s-hostpath-volume
```

6. Repeat Step 4 and Step 5 on all CDF worker nodes.
7. Launch a terminal session and then log in to a database node.
8. Navigate to the following directory:

```
cd /[database_install_directory]/
```

9. Stop the Kafka Scheduler:

```
./kafka_scheduler stop
```

10. (Conditional) If you have the ArcSight database installed), complete the following steps as a dbadmin user:
    a. Execute the following command and specify your dbadmin password:

```
/opt/vertica/bin/vsql
Password:<password>
```

    b. Execute the following command to delete the data in the default_secops_adm.events table:

```
DELETE FROM default_secops_adm.events;
```

c. Execute the following command to delete the default_secops_intelligence schema:

```
drop schema default_secops_intelligence cascade;
```

11. Continue with reinstalling the Platform and deploying the capabilities. Complete one of the following actions:
    a. Deploy the capabilities manually. For more information, see Performing a Manual Deployment.
    b. Deploy the capabilities by using the ArcSight Platform Installer.

Using REST APIs

User interfaces use REST APIs to manage and access data and configuration information. You can also access the APIs directly, if needed. For example, you might want to update a particular user's dashboard or the end point documentation.

- "Setting Up Access to REST APIs" below
- "Authenticating to and Calling the REST API" on the next page
- Links to REST API Documentation

Setting Up Access to REST APIs

You must configure one Client ID and Secret to authenticate with the REST APIs. After you have established the secret, you might want to update it according to your password rotation policies.

If you update the secret, you must update all REST API clients to reflect the update.

1. Generate your Client ID and Client Secret respectively.

   For example, you can use OpenSSL to generate random values that are more secure:

   For Client ID:

   ```
   openssl rand -hex 16
   ```

   For Client Secret:

   ```
   openssl rand -hex 32
   ```

2. Log in to the CDF Management Portal.
3. Click the browse icon on the far right, then choose **Reconfigure**.
4. In the **Single Sign-on Configuration** section, specify values for **Client ID** and **Client Secret**.
5. Click **Save.**

Authenticating to and Calling the REST API

Before calling a REST API, you must authenticate your session, which involves generating an access and refresh token. The REST API client uses these tokens when you call the REST API server.

1. To generate access tokens, in your API client, use the method POST and the following URL:

   `https://<cdf-machine-hostname>/osp/a/default/auth/oauth2/grant`

   where <cdf-machine-hostname> represents your ArcSight product.

   Select and specify *Header* and *Body* information as follows:

   Authorization

   a. Authorization type as **Basic**

   b. Client_ID:Client_Secret as **base64 encoded**

   Header

   a. Content-Type as **application/x-www-form-urlencoded**

   b. Accept as **application/json**

   Body

   c. Enter **grant_type** as password

   d. Enter **Username** as User ID

   e. Enter **password** as the password of the UserID

   The server replies with the access_token, the expires_in number of seconds for the access_ token validity, and a refresh_token to generate a new access token when the access token expires. To understand how to generate a new access token using the refresh_token, see Refreshing Access Tokens.

   For example:

```
curl --location --request POST
'https://cdf.dom.lab/osp/a/default/auth/oauth2/grant' \
--header 'Accept: application/json' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'Authorization: Basic
MDM0YzE2MzU1N2I0OTUxOWE5ZjRlYjVlNDBkOGJiZGQ6ZGEzMzA2NzBiMGQ0YWIzN2JjOGRhM
zgyNDBiMjM2NzVlZDVkMjUwNTBkNzIzZmNjNDhmNDUxNGJjYzY0NTUwZA=='
--data-urlencode 'grant_type=password' \
--data-urlencode 'username=*****' \
--data-urlencode 'password=*****'
```

2. To generate a session token using the access token, in your API client use the method GET and the following URL:

   `https://<cdf-machine-hostname>/mgmt/users/me/details`

   For example:

   https://cdf.dom.lab/mgmt/users/me/details

   Select and specify *Header* and *Body* information as follows:

   Authorization

   a. Authorization type as **Bearer**

   b. Use the Access token generated in step 1

   For example:

```
curl -v --location --request GET
'https://<cdf.dom.lab>/mgmt/users/me/details \
--header 'Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZ
SI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36P
Ok6yJV_adQssw5c'
```

3. To review the documentation for REST API endpoints, access the following URLs:

   - **Recon**:

     `https://<master_FQDN or IP>/rec/rest-api-docs`

     `https://<master_FQDN or IP>/re/rest-api-docs`

   - **System Metadata**:

     `https://<master_FQDN or IP>/metadata/rest-api-docs`

   where `<master_FQDN or IP>` represents your ArcSight product.

   For example:

   `https://<cdf.dom.lab>/metadata/rest-api-docs`

4. Access your REST API endpoint with the session token as a cookie from step 2.

For example, to search all the dashboards owned by the logged in user ID, and the dashboards that are being shared with the logged in user ID's role, you might use the following content:

```
curl --location --request GET
'https://cdf.dom.lab/metadata/api/v1/dashboards' \
--header 'Cookie: SESSIONTOKEN=1E4C45F0B8DC821FF251EC17558B1ABF'
```

Refreshing Access Tokens

To generate the access token again with the refresh token in your API client, use the method POST and the following URL:

https://<cdf-machine-hostname>/osp/a/default/auth/oauth2/token

Select and specify Header and Body information as follows, where:

Authorization

a. Authorization type as **Basic**

b. Client_ID:Client_Secret as **base64 encoded**

Header

a. Content-Type as **application/x-www-form-urlencoded**

b. Accept as **application/json**

c. Authorization type as **Basic**

Body

a. Set **grant_type** as **refresh_token**

b. Set **refresh_token** as generated in step 1

For example:

```
curl --location --request POST
'https://cdf.dom.lab/osp/a/default/auth/oauth2/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'Accept: application/json' \
--header 'Authorization: Basic Q2xpZW50SWQ6Q2xpZW50U2VjcmV0' \
--data-urlencode 'grant_type=refresh_token' \
--data-urlencode 'refresh_token=IWmk3ugO-KI-XlM16EXSS0WJKBeN08pGh3o'
```

Links to REST API Documentation

Name	REST API Endpoint Documentation
ArcMC and Fusion ArcMC	https://<*master_FQDN or IP*>/arcmc/rest-api-docs
Database Monitoring	https://<*master_FQDN or IP*>/db-mon/rest-api-docs
ESM	https://<master_FQDN or IP>/detect-api
Intelligence	Developer's Guide to ArcSight Intelligence
Recon	https://<*master_FQDN or IP*>/rec/rest-api-docs
System Metadata	https://<*master_FQDN or IP*>/metadata/rest-api-docs
Transformation Hub	https://<*master_FQDN or IP*>:32080/rest-api-docs

Retrieving the CDF Root CA

You can retrieve the CDF root CA from a supported web browser or by using the command line.

Retrieving the CDF Root CA from a Browser

This procedure assumes you are using Google Chrome.

1. Specify the following URL in the browser:

   ```
   https://<master_node_FQDN>:5443
   ```

2. Click the icon next to the left of the URL, then click **Certificate**.
3. Click **Certification Path**.
4. Double-click the CA certificate. A pop-up window displays.
   a. In the pop-up window, click **Details**, then click **Copy to File...**
   b. Click **Next**.
   c. Select **Base-64 encoded X.509 (.CER)** and click **Next**.
   d. Specify a file name (for example, ca.cer) and click **Next**.
   e. Click **Finish** and close the pop-up window.
5. (Conditional) If you have multiple CA certificates, repeat Step 4 for each CA certificate in the certificate chain.

Retrieving the CDF Root CA Using Command Line

1. Log in to the initial master node of the cluster.
2. Execute the following command to retrieve the CDF CA certificate:

```
${K8S_HOME}/scripts/cdf-updateRE.sh read > ca.cer
```

Understanding License Keys

ArcSight Platform products ship with an evaluation license, which you can change to a valid license during or after installation. Many ArcSight Platform features are common across your deployed products, but some features might be restricted to a specific license or limited in scope depending on your license.

- "Understanding the Types of Licenses" on the next page
- "How Your License Affects Available Features" on page 620
- "How Your License Affects Data Storage Policies" on page 623
- "How Data Ingestion Affects Your License" on page 623

Considerations for Product Licensing

- If you use an evaluation license, ensure that you apply your product license **before** the license expires to avoid disruption to event flow and functionality.
- Your Intelligence license is based on the number of users that you want Intelligence to run analytics on. The license policy is violated when the number of users exceeds the maximum limit. Renew your license before its validity expires or if the license policy has been violated.
- For a Transformation Hub license, you can use a legacy ArcMC ADP license key or a more recent Transformation Hub license.
- If you install multiple term or permanent licenses for a product or function, the expiration date matches the date for the license that expires first.
- When used with an ArcSight product that requires the ArcSight Database, the storage capacity for the ArcSight Database is license limited to 976 PB. If your ArcSight Database storage utilization exceeds 976 PB, contact technical support for assistance.
- Starting 30 days before a license expires, ArcSight Platform displays a warning about the coming expiration date. However, if your deployment includes an MSSP license, the system does not display this warning.

- Several product licenses include the use of common features, such as Search or Reporting. In general, when you have deployed multiple products, you only need one valid license for the feature to function.
- Your product license might affect the maximum value that you can set for the data retention policy. For example, with a Recon license you can configure stored data to never expire.
- If the calculated MMEPS value exceeds your licensed events per second (EPS) capacity, ArcSight Platform displays a warning until the data ingestion rate normalizes. For example, the warning might display when data ingestion into the ArcSight Database is higher than your licensed EPS.
- When a license expires or is missing, ArcSight Platform redirects users to an invalid license page and disables the functionality associated with the license. To resolve this issue, you can install a valid license or troubleshoot licensing issues.
- You can check the status of an installed license.

Understanding the Types of Licenses

ArcSight products allow a short evaluation period before requiring a long-term license. Your products could have the following types of licenses:

- 90-day Instant-on License for Evaluation
- Long-term Licenses
- MSSP License

90-day Instant-on License for Evaluation

ArcSight products ship with an instant-on evaluation license, which enables functionality for 90 days after you install the product. To continue using the product for a longer term, you must install a valid license key. Installing a term or permanent license will override the instant-on license.

To ensure continuity of functionality and event flow, apply your valid product license **before** the evaluation license has expired.

Long-term Licenses

Most valid product licenses cover a specific period of time (a term license) or are a permanent license. If you install multiple term or permanent licenses for a product or function, the expiration date matches the date for the license that expires last.

MSSP License

An MSSP license grants you access to all functions and features available in the Intelligence and Recon capabilities, regardless of the non-MSSP licenses that you might also have installed.

You can purchase an MSSP license, a non-MSSP license, or both to meet your requirements. If you have both licenses, the system responds in the following ways upon expiration:

- If the MSSP license expires and the non-MSSP license is still valid, there is no impact in the accessibility and usage of the product with the non-MSSP license. However, your users will not be able to access any additional features that the MSSP license grants. Renew your license or licenses before the non-MSSP license expires or if its license policy is violated.
- If the non-MSSP license expires or its license policy is violated and the MSSP license is still valid, there is no impact in the accessibility and usage of the product. Renew your license or licenses before the MSSP license expires.

If you have purchased the MSSP license, ensure that you add an MSSP contract and create an MSSP profile in Fusion. For more information, see the Help for **Admin** > **Contract & Usage**.

How Your License Affects Available Features

When a user logs in to the ArcSight Platform or attempts to access a function that requires a special license, the system checks the licenses associated with the deployed products.

The common features such as the Reports Portal and SOAR are enabled if at least one license to enable them is valid. For example, the Transformation Hub license on its own does not enable the Reports Portal, but does if the Recon license is also deployed. Both the Recon and Intelligence licenses enable the Reports Portal. Therefore if your Intelligence license were to expire but the Recon license remains valid, the Reports Portal remains enabled on account of the valid Recon license.

The following table shows the functionality available per product license:

	ESM	Intelligence	MSSP	Recon
Data Quality Dashboard			✔	✔
Event Integrity Check			✔	✔
Outlier Analytics			✔	✔
Reports Portal	✔	✔	✔	✔
ArcMC	✔	✔	✔	✔
Transformation Hub	✔	✔	✔	✔
Search		✔	✔	✔
Storage Groups		✔	✔	✔

Data Quality Dashboard

The Data Quality Dashboard provides detailed information about the gap between Device Receipt Time from the raw event itself versus the Normalized Event Time and Database Receipt Time. The dashboard identifies the sources that have a gap. Based on the information analyzed through the dashboard, you can accurately mitigate the problem. This feature also provides history of your data over time. For more information about using this feature, see the Help for **Insights** > **Data Quality**.

Event Integrity Check

The Event Integrity Check enables you to validate that the event information in your database matches the content sent from SmartConnectors, helping you check whether event data might be compromised. In addition to reviewing the raw event data received from SmartConnectors, you can enable Transformation Hub to generate verification events for more than 20 parsed fields to include in the check. By expanding the number of fields within an event that the check examines, you reduce the opportunities for malicious users to hide their activity. For more information about using this feature, see the Help for **Admin** > **Event Integrity**.

Outlier Analytics

To help you identify anomalous behavior, the Outlier Analytics feature allows you to compare incoming EventCount, BytesIn, and BytesOut values to typical values for your environment. The `EventCount`, `BytesIn` and `BytesOut` values are aggregations over certain time periods for each host/IP address. Outlier Analytics can create and persist a baseline of host behavior. To derive outliers, you compare this baseline with aggregations over new time periods. Basically, the lower the anomaly score, the more likely the event is anomalous. For more information about using this feature, see the Help for **Insights** > **Outliers**.

Reports Portal

To help you hunt for undetected threats and vulnerabilities, the Reports Portal includes a set of built-in dashboards and reports associated with industry security standards such as the Cloud Security Alliance and OWASP. Additional reports and dashboards focus on fundamental security issues, such as monitoring firewalls and malware. For more information about using this feature, see the Help for **Reports**.

Search

The Search feature enables you to look for and investigate events that meet specified criteria so you can detect anomalies that point to security threats. Each search consists of specifying query input, search result fields, and the time period for which you want to search events. Commonly available Search features include fieldsets, lookup lists, and scheduled searches. Users can save their search results, search queries, or queries plus search criteria. For more information about using this feature, see the Help for **Search**.

SOAR

ArcSight SOAR provides a secure orchestration, automation, and response solution where customers can automate a lot of their incident management activities so analysts can perform more in-depth threat hunting and case response. When a user acceses SOAR features, the system checks for an active ESM, Recon, or Intelligence license. SOAR also supports manual/legacy ESM license types. For example, customers using ESM 6.11 and later can also use the SOAR capability. For more information about using this feature, see the *ArcSight SOAR User Guide*.

Storage Groups

You can divide data into storage groups, which allows you to partition the incoming events data and provide different retention periods, based on the query filter. Because you can set data retention policies per storage group, you can retain certain high volume events for a short time period and other important events for longer time period. For more information about using this feature, see the Help for **Configuration** > **Storage**.

Depending on your license, storage retention might be limited.

How Your License Affects Data Storage Policies

You can divide data into **storage groups**, which allows you to partition the incoming events data and provide different retention periods, based on the query filter. To preserve space in the database and improve data retrieval from storage groups, you can configure the database to remove events older than a certain number of months. Your product license affects the maximum value that you set for the data retention policy. If you have deployed multiple capabilities in the ArcSight Platform, the product license that allows the longest maximum value takes precedence over the maximum allowed value of the other licenses.

Product License	Maximum retention value
Intelligence	30 days
Recon – Instant On	90 days
Recon	Never Expire

For more information about setting the data retention policy, see the Help for **Configuration** > **Storage**.

How Data Ingestion Affects Your License

Your product license specifies the maximum number of events per second that your system can ingest, based on the moving **median events per second (MMEPS)** value. To calculate MMEPS, the system performs the following actions:

1. **Calculate Events Per Day (EPD)**: Events Per Day is the total number of events ingested into database in a twenty-four hour period. For Day1, the system calculates the EPD based from the time you install the product until GTM+0 hours. The time frame is based on GTM+0 hours starting at 00:00:00 and ending at 23:59:59, regardless of any local times that might be in use.
2. **Calculate Sustained EPS (SEPS)**: Sustained EPS is the “constant” events per second that the system sustained within the 240hour period. For Day1, the system calculates the EPD based from the time you install the product until GTM+0 hours. It normalizes peaks and valleys to give a better indication of use. The formula used for this calculation is (EPD/ ((60*60)*24)).
3. **Calculate MMEPS for the last 45 days**: Using the SEPS information recorded per day, the system calculates a moving median EPS value using the data set from the last 45-days. After the first 45 days, the system adjusts the calculation window one day every24 hours. The official clock for calculation purposes is defined by GTM+0 hours starting at 00:00:00 to 23:59:59 regardless of local time.

Your product license remains in compliance as long as the MMEPS value remains at the limit or below the purchased license capacity. If three or more consecutive MMEPS value indicators exceed their capacity based on the purchased license, the license is considered to be out of compliance. ArcSight Platform displays a warning until the data ingestion rate normalizes.

Creating Widgets for the Dashboard

The license for your deployed application also grants you access to the **Widget Software Development Kit** (the Widget SDK), which you can download to your local production or test environment. The Widget SDK enables you to build new widgets or modify existing widgets for deployed applications.

- "Using the Widget SDK" below
- "Considerations for Updating the Widget Store" below

Using the Widget SDK

The Widget SDK requires nodejs 12.7.0, at a minimum, which comes with yarn version 1.16.0.

1. Extract the contents of the `widget-sdk-n.n.n.tgz` file to your developer workstation.
2. Follow the steps in the Getting Started section of the included *ReadMe*.
3. After you compile the new or modified widget, add it to the widget store for use in the Dashboard.
4. (Optional) To allow additional Fusion users to incorporate your custom widget into their environment, submit the widget to the ArcSight Marketplace.

Considerations for Updating the Widget Store

Review the following considerations before modifying or creating new widgets:

- Widgets provided with a deployed application are included in the default widget store directory.

  ```
  /opt/arcsight-nfs/arcsight-volume/fusion/widget-store
  ```

- Each new widget must have a unique name.

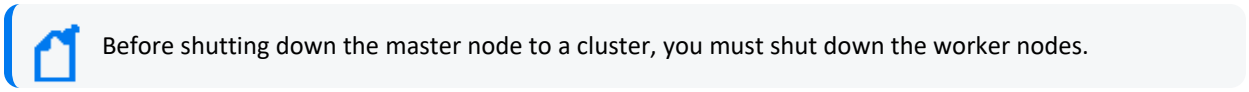
- You cannot edit an out-of-the-box widget. However, you can use the widget as a template for creating a new one. To prevent the modified widget from being erased or overwritten by a product upgrade, give the widget a non-default name.

Restarting or Shutting Nodes in a Kubernetes Cluster

If you need to restart or shut down any single or all worker or master nodes in the cluster, you must stop Kubernetes and the databases services running on the node in order to enable the Kubernetes pods to start after a node restart, and to prevent database corruption.

- Restart or Shut Down Nodes or Bring Down a Cluster Manually
- Restart Worker Nodes in the AWS Cluster

To restart or shut down nodes or bring down a cluster manually:



Applies only if you have installed the cluster manually.

1. (Conditional) If Intelligence has been deployed, execute the following command from the database Node1 to stop the data ingestion:

   ```
   /opt/arcsight-db-tools/kafka_scheduler stop
   ```

2. (Conditional) If Intelligence has been deployed, perform the following steps from the master node to stop the Elasticsearch cluster:

   a. Execute the following command to retrieve the namespace:

   ```
   export ARCSIGHT_INSTALLER_NS=`kubectl get namespaces| grep arcsight-installer | awk '{ print $1}'`
   ```

   b. Execute the following command to scale down the Logstash pods:

   ```
   kubectl scale sts interset-logstash -n $ARCSIGHT_INSTALLER_NS --replicas=0
   ```

   c. Execute the following command to scale down the h2 pod:

   ```
   kubectl scale sts h2 -n $ARCSIGHT_INSTALLER_NS --replicas=0
   ```

   d. Execute the following commands to scale down the Elasticsearch data nodes and Elasticsearch master node:

```
kubectl scale sts elasticsearch-data -n $ARCSIGHT_INSTALLER_NS --
replicas=0
```

```
kubectl scale sts elasticsearch-master -n $ARCSIGHT_INSTALLER_NS --
replicas=0
```

3. Perform the following steps on each of the worker nodes first and later the master node:
   a. Log in to the node as the root user.
   b. Change to the following directory:

```
cd <K8S_HOME>/bin/
```

```
For example: /opt/arcsight/kubernetes/bin
```

   c. Execute the following command to stop the Kubernetes services:

```
kube-stop.sh
```

   d. Execute the following command to unmount Kubernetes volumes:

```
kubelet-umount-action.sh
```

   e. Restart or shut down node:

```
reboot
```

```
shutdown -h now
```

4. (Conditional) If restart fails, perform a hard reboot of the node.
5. (Conditional) If the master node has been restarted or rebooted, the Kubernetes services automatically start. On the master node, check whether the Kubernetes services are running:

```
kube-status.sh
```

Ensure that the swap is off on the master node, otherwise the kubelet service may not start.

6. If the node contains the database, see Rebooting Database Cluster.
7. After the worker nodes restart, do the following in each of the worker nodes:

a. Log in to the node as root.

b. Change to the following directory:

```
cd <K8S_HOME>/bin/
```

```
For example: /opt/arcsight/kubernetes/bin
```

c. Start the Kubernetes services:

```
kube-start.sh
```

```
kubectl get nodes
```

d. Check whether all Kubernetes services are running:

```
kube-status.sh
```

Ensure that the swap is off on all worker nodes, otherwise the kubelet service may not start.

8. If the status of any of the master or worker nodes is *SchedulingDisabled*, then execute the following command to enable scheduling of pods:

```
kubectl uncordon <workernodename> or kubectl uncordon <masternodename>
```

9. (Conditional) If Intelligence has been deployed, run the following command from the database Node1 to start the data ingestion:

```
/opt/arcsight-db-tools/kafka_scheduler start
```

10. (Conditional) If Intelligence has been deployed, perform the following steps from the master node to start the Elasticsearch cluster:

a. Execute the following command to retrieve the namespace:

```
export ARCSIGHT_INSTALLER_NS=`kubectl get namespaces| grep arcsight-
installer | awk '{ print $1}'`
```

b. Execute the following commands to scale up the Elasticsearch data nodes and Elasticsearch master node:

```
kubectl scale sts elasticsearch-data -n $ARCSIGHT_INSTALLER_NS --
replicas=3
```

```
kubectl scale sts elasticsearch-master -n $ARCSIGHT_INSTALLER_NS --
replicas=1
```

c. Execute the following command to scale up the h2 pod:

```
kubectl scale sts h2 -n $ARCSIGHT_INSTALLER_NS --replicas=1
```

d. Execute the following command to scale up the Logstash pods:

```
kubectl scale sts interset-logstash -n $ARCSIGHT_INSTALLER_NS --
replicas=3
```

To restart worker nodes in the AWS cluster:

1. Log in to the bastion host.
2. Using the Find Services search tool, locate and browse to the EC2 Dashboard.
3. Perform one of the following actions to start and stop the worker nodes:
   - By using the **Instances** option:
     a. In the left navigation pane, under **Instances**, click **Instances**. A list of the node instances is displayed.
     b. Select the worker node instance or instances, click **Instance state**, then select **Stop instance** from the drop-down list.
     c. Click the Refresh icon until a new set of worker node instance IDs is created.
   - By using the **Auto Scaling** option:
     a. In the left navigation pane, under **Auto Scaling**, click **Auto Scaling Groups**. A list of the created Auto Scaling groups is displayed.
     b. Select the required Auto Scaling group, then click **Edit**.
     c. Update the values of **Desired capacity**, **Minimum capacity**, **Maximum capacity** to 0.
     d. Click **Update**.
     e. In the left navigation pane, under **Instances**, click **Instances**. A list of the node instances is displayed.
     f. Ensure that the worker nodes corresponding to the Auto Scaling group that you edited are in the Terminated state.
     g. Repeat steps a and b.
     h. Update the values of **Desired capacity**, **Minimum capacity**, **Maximum capacity** to the earlier values.
     i. Click **Update**.

j. In the left navigation pane, under **Instances**, click **Instances**. A list of the node instances is displayed.

k. Click the Refresh icon until a new set of worker node instance IDs is created.

4. Label the worker nodes.
5. Update the target groups for Ports 5443 and Ports 443, 32081, or 32080 with the new worker node Instance IDs.
6. Log in to the CDF Management Portal with the following credentials:

   User name: admin

   Password: <the password you provided during CDF installation>

7. Click  , then click **Reconfigure**.
8. Depending on the capabilities for which you have assigned labels to the worker nodes, click the relevant tabs and reconfigure the properties.

   If the node you have restarted is an HDFS namenode, then ensure that you update the **HDFS NameNode** field in the **Intelligence** tab with the hostname or IP address of the worker node labeled as **intelligence-namenode:yes**.

9. Click **Save**.
10. Configure the database with HDFS to update the core-site.xml with the new value of the HDFS namenode.
11. Verify that all the pods are in the Running state.

Understanding the Schema for Events

The following table describes the columns of the **default_sec_ops_adm.Events** table:

Column Name	Data Type	Description
agentAddressBin/agentAddress	Binary(16)	The IP address of the ArcSight connector that processed the event.
agentHostName	Varchar(1023)	The hostname of the ArcSight connector that processed the event.
agentNtDomain	Varchar(255)	
agentSeverity	Varchar(9)	

Column Name	Data Type	Description
agentType	Varchar(63)	The agent type of the ArcSight connector that processed the event.
agentZoneURI	Varchar(2048)	Specify hourly, daily, weekly, or monthly.
applicationProtocol	Varchar(40)	Application level protocol, example values are HTTP, HTTPS, SSHv2, Telnet, POP, IMPA, IMAPS, and so on.
baseEventCount	Integer	A count associated with this event. How many times was this same event observed? Count can be omitted if it is 1.
bytesIn	Integer	Number of bytes transferred inbound, relative to the source to destination relationship, meaning that data was flowing from source to destination.
bytesOut	Integer	Number of bytes transferred outbound relative to the source to destination relationship. For example, the byte number of data flowing from the destination to the source.
categoryBehavior	Varchar(1023)	The action or behavior associated with the event.
categoryDeviceGroup	Varchar(1023)	The type of events for the device.
categoryObject	Varchar(1023)	The type of the object.
categoryOutcome	Varchar(1023)	The outcome of the event.
categorySignificance	Varchar(1023)	The significance of the event.
categoryTechnique	Varchar(1023)	

Column Name	Data Type	Description
destinationAddressBin/destinationAddress	Binary(16)	Identifies the destination address that the event refers to in an IP network. The format is an IPv4 address. Example: “192.168.10.1”
destinationDnsDomain	Varchar(255)	The DNS domain part of the complete fully qualified domain name (FQDN).
destinationHostName	Varchar(1023)	Identifies the destination that an event refers to in an IP network. The format should be a fully qualified domain name (FQDN) associated with the destination node, when a node is available. Examples: “host.domain.com” or “host”.
destinationMacAddressBin/ destinationMacAddress	Binary(16)	Six colon-seperated hexadecimal numbers. Example: “00:0D:60:AF:1B:61”
destinationNtDomain	Varchar(255)	The Windows domain name of the destination address.
destinationPort	Integer	The valid port numbers are between 0 and 65535.
destinationProcessName	Varchar(1023)	The name of the event’s destination process. Example: “telnetd” or “sshd”.
destinationServiceName	Varchar(1023)	The service targeted by this event. Example: “sshd”
destinationTranslatedAddressBin/ destinationTranslatedAddress	Binary(16)	
destinationUserId	Varchar(1023)	Identifies the destination user by ID. For example, in UNIX, the root user is generally associated with user ID 0.

Column Name	Data Type	Description
destinationUserName	Varchar(1023)	Identifies the destination user by name. This is the user associated with the event's destination. Email addresses are often mapped into the UserName fields. The recipient is a candidate to put into this field.
destinationUserPrivileges	Varchar(1023)	The typical values are "Administrator", "User", and "Guest". This identifies the destination user's privileges. In UNIX, for example, activity executed on the root user would be identified with destinationUser Privileges of "Administrator".
destinationZoneURI	Varchar(2048)	The URI for the Zone that the destination asset has been assigned to in ArcSight.
deviceAction	Varchar(63)	Action taken by the device.
deviceAddressBin/deviceAddress	Binary(16)	Identifies the device address that an event refers to in an IP network. The format is an IPv4 address. Example: "192.168.10.1".
deviceCustomDate1	Integer	One of two timestamp fields available to map fields that do not apply to any other in this dictionary. Use sparingly and seek a more specific, dictionary supplied field when possible.
deviceCustomDate1Label	Varchar(1023)	All custom fields have a corresponding label field. Each of these fields is a string and describes the purpose of the custom field.

Column Name	Data Type	Description
deviceCustomDate2	Integer	One of two timestamp fields available to map fields that do not apply to any other in this dictionary. Use sparingly and seek a more specific, dictionary supplied field when possible.
deviceCustomDate2Label	Varchar(1023)	All custom fields have a corresponding label field. Each of these fields is a string and describes the purpose of the custom field.
deviceCustomNumber1	Integer	One of three number fields available to map fields that do not apply to any other in this dictionary. Use sparingly and seek a more specific, dictionary supplied field when possible.
deviceCustomNumber1Label	Varchar(1023)	All custom fields have a corresponding label field. Each of these fields is a string and describes the purpose of the custom field.
deviceCustomNumber2	Integer	One of three number fields available to map fields that do not apply to any other in this dictionary. Use sparingly and seek a more specific, dictionary supplied field when possible.
deviceCustomNumber2Label	Varchar(1023)	All custom fields have a corresponding label field. Each of these fields is a string and describes the purpose of the custom field.
deviceCustomNumber3	Integer	One of three number fields available to map fields that do not apply to any other in this dictionary. Use sparingly and seek a more specific, dictionary supplied field when possible.

Column Name	Data Type	Description
deviceCustomNumber3Label	Varchar(1023)	All custom fields have a corresponding label field. Each of these fields is a string and describes the purpose of the custom field.
deviceCustomString1	Varchar(4000)	One of six strings available to map fields that do not apply to any other in this dictionary. Use sparingly and seek a more specific, dictionary supplied field when possible.
deviceCustomString1Label	Varchar(1023)	All custom fields have a corresponding label field. Each of these fields is a string and describes the purpose of the custom field.
deviceCustomString2	Varchar(4000)	One of six strings available to map fields that do not apply to any other in this dictionary. Use sparingly and seek a more specific, dictionary supplied field when possible.
deviceCustomString2Label	Varchar(1023)	All custom fields have a corresponding label field. Each of these fields is a string and describes the purpose of the custom field.
deviceCustomString3	Varchar(4000)	One of six strings available to map fields that do not apply to any other in this dictionary. Use sparingly and seek a more specific, dictionary supplied field when possible.
deviceCustomString3Label	Varchar(1023)	All custom fields have a corresponding label field. Each of these fields is a string and describes the purpose of the custom field.

Column Name	Data Type	Description
deviceCustomString4	Varchar(4000)	One of six strings available to map fields that do not apply to any other in this dictionary. Use sparingly and seek a more specific, dictionary supplied field when possible.
deviceCustomString4Label	Varchar(1023)	All custom fields have a corresponding label field. Each of these fields is a string and describes the purpose of the custom field.
deviceCustomString5	Varchar(4000)	One of six strings available to map fields that do not apply to any other in this dictionary. Use sparingly and seek a more specific, dictionary supplied field when possible.
deviceCustomString5Label	Varchar(1023)	All custom fields have a corresponding label field. Each of these fields is a string and describes the purpose of the custom field.
deviceCustomString6	Varchar(4000)	One of six strings available to map fields that do not apply to any other in this dictionary. Use sparingly and seek a more specific, dictionary supplied field when possible.
deviceCustomString6Label	Varchar(1023)	All custom fields have a corresponding label field. Each of these fields is a string and describes the purpose of the custom field.
deviceEventCategory	Varchar(1023)	Represents the category assigned by the originating device. Devices often use their own categorization schema to classify event. Example: “/Monitor/Disk/Read”

Column Name	Data Type	Description
deviceEventClassId	Varchar(100)	Unique code assigned to an event.
deviceExternalId	Varchar(255)	A name that uniquely identifies the device generating this event.
deviceHostName	Varchar(100)	The format should be a fully qualified domain name (FQDN) associated with the device node, when a node is available. Example: “host.domain.com” or “host”.
deviceInboundInterface	Varchar(128)	Interface on which the packet or data entered the device.
deviceOutboundInterface	Varchar(128)	Interface on which the packet or data left the device.
deviceProduct	Varchar(100)	The device product of the client.
deviceReceiptTime	Integer	The time at which the event related to the activity was received. The format is MMM dd yyyy HH:mm:ss or milliseconds since epoch (Jan 1st 1970)
deviceSeverity	Varchar(63)	The HTTP response status.
deviceVendor	Varchar(100)	The device vendor of the client.
deviceVersion	Varchar(31)	The device version.
deviceZoneURI	Varchar(2048)	The URI for the Zone that the device asset has been assigned to in ArcSight.
endTime	Integer	The time at which the activity related to the event ended. The format is MMM dd yyyy HH:mm:ss or milliseconds since epoch (Jan 1st1970). An example would be reporting the end of a session.

Column Name	Data Type	Description
eventId	Integer	This is a unique ID that ArcSight assigns to each event.
externalId	Varchar(40)	The ID used by an originating device. They are usually increasing numbers, associated with events.
fileName	Varchar(1023)	Name of the file only (without its path).
filePath	Varchar(1023)	Full path to the old file, including the file name itself. Examples: c:\Program Files\ WindowsNT\Accessories \wordpad.exe or /usr/bin/zip
flexDate1	Integer	A timestamp field available to map a timestamp that does not apply to any other defined timestamp field in this dictionary. Use all flex fields sparingly and seek a more specific, dictionary supplied field when possible. These fields are typically reserved for customer use and should not be set by vendors unless necessary.
flexDate1Label	Varchar(128)	The label field is a string and describes the purpose of the flex field.
flexNumber1	Integer	
flexNumber1Label	Varchar(128)	
flexNumber2	Integer	
flexNumber2Label	Varchar(128)	

Column Name	Data Type	Description
flexString1	Varchar(1023)	One of four floating point fields available to map fields that do not apply to any other in this dictionary. Use sparingly and seek a more specific, dictionary supplied field when possible. These fields are typically reserved for customer use and should not be set by vendors unless necessary.
flexString1Label	Varchar(128)	The label field is a string and describes the purpose of the flex field.
flexString2	Varchar(1023)	One of four floating point fields available to map fields that do not apply to any other in this dictionary. Use sparingly and seek a more specific, dictionary supplied field when possible. These fields are typically reserved for customer use and should not be set by vendors unless necessary.
flexString2Label	Varchar(128)	The label field is a string and describes the purpose of the flex field.
globalEventId	Integer	
message	Varchar(1023)	An arbitrary message giving more details about the event. Multi-line entries can be produced by using \n as the new line separator.
name	Varchar(1023)	
requestClientApplication	Varchar(1023)	The User-Agent associated with the request.
requestContext	Varchar(2048)	Description of the content from which the request originated (for example, HTTP Referrer)

Column Name	Data Type	Description
requestMethod	Varchar(1023)	The method used to access a URL. Possible values: “POST”, “GET”, etc.
requestUrl	Varchar(2048)	In the case of an HTTP request, this field contains the URL accessed. The URL should contain the protocol as well. Example: “http://www/secure.com”
requestUrlFileName	Varchar(2048)	
requestUrlQuery	Varchar(2048)	
sourceAddressBin/sourceAddress	Binary(16)	Identifies the source that an event refers to in an IP network. The format is an IPv4 address. Example: “192.168.10.1”.
sourceHostName	Varchar(1023)	Identifies the source that an event refers to in an IP network. The format should be a fully qualified domain name (FQDN) associated with the source node, when a mode is available. Examples: “host” or “host.domain.com”.
sourceMacAddressBin/sourceMacAddress	Binary(16)	Six colon-separated hexadecimal numbers. Example: “00:0D:60:AF:1B:61”
sourceNtDomain	Varchar(255)	The Windows domain name for the source address.
sourcePort	Integer	The valid port numbers are 0 to 65535.
sourceProcessName	Varchar(1023)	The name of the event’s source process.
sourceServiceName	Varchar(1023)	The service that is responsible for generating this event.

Column Name	Data Type	Description
sourceTranslatedAddressBin/sourceTranslatedAddress	Binary(16)	Identifies the translated source that the event refers to in an IP network. The format is an IPv4 address. Example: “192.168.10.1”.
sourceUserId	Varchar(1023)	Identifies the source user by ID. This is the user associated with the source of the event. For example, in UNIX, the root user is generally associated with user ID 0.
sourceUserName	Varchar(1023)	Identifies the source user by name. Email addresses are also mapped into the UserName fields. The sender is a candidate to put into this field.
sourceUserPrivileges	Varchar(1023)	The typical values are “Administrator”, “User”, and “Guest”. It identifies the source user’s privileges. In UNIX, for example, activity executed by the root user would be identified with “Administrator”.
sourceZoneURI	Varchar(2048)	The URI for the Zone that the source asset has been assigned to in ArcSight.
startTime	Integer	The time when the activity the event referred to started. The format is MMM dd yyyy HH:mm:ss or milliseconds since epoch (Jan 1st 1970)
transportProtocol	Varchar(31)	Identifies the Layer-4 protocol used. The possible values are protocols such as TCP or UDP.

Column Name	Data Type	Description
type	Varchar(1023)	0 means base event, 1 means aggregated, 2 means correlation, and 3 means action. This field can be omitted for base events (type 0).
agentDnsDomain	Varchar(255)	The DNS domain name of the ArcSight connector that processed the event.
agentId	Varchar(40)	The agent ID of the ArcSight connector that processed the event.
agentMacAddressBin	Binary(16)	
agentReceiptTime	Integer	The time at which information about the event was received by the ArcSight connector.
agentTimeZone	Varchar(255)	The agent time zone of the ArcSight connector that processed the event.
agentTranslatedAddressBin	Binary(16)	
agentTranslatedZoneExternalID	Varchar(200)	
agentTranslatedZoneURI	Varchar(2048)	
agentVersion	Varchar(31)	The version of the ArcSight connector that processed the event.
agentZoneExternalID	Varchar(200)	
categoryDeviceType	Varchar(1023)	The events generated by a device type irrespective of the device group the events belong to.
cryptoSignature	Varchar(512)	
customerExternalID	Varchar(200)	
customerURI	Varchar(2048)	
destinationGeoCountryCode	Varchar(1023)	
destinationGeoLatitude	Float	The latitudinal value from which the destination’s IP address belongs.

Column Name	Data Type	Description
destinationGeoLocationInfo	Varchar(1023)	
destinationGeoLongitude	Float	The longitudinal value from which the destination's IP address belongs.
destinationGeoPostalCode	Varchar(1023)	
destinationGeoRegionCode	Varchar(1023)	
destinationProcessId	Integer	Provides the ID of the destination process associated with the event. For example, if an event contains process ID 105, "105" is the process ID.
destinationTranslatedPort	Integer	Port after it was translated; for example, a firewall. Valid port numbers are 0 to 65535.
destinationTranslatedZoneExternalID	Varchar(200)	
destinationTranslatedZoneURI	Varchar(2048)	
destinationZoneExternalID	Varchar(200)	
deviceAssetId	Varchar(40)	
deviceCustomDescriptorId	Varchar(1023)	
deviceCustomFloatingPoint1	Float	One of four floating point fields available to map fields that do not apply to any other in this dictionary.
deviceCustomFloatingPoint1Label	Varchar(1023)	All custom fields have a corresponding label field. Each of these fields is a string and describes the purpose of the custom field.
deviceCustomFloatingPoint2	Float	One of four floating point fields available to map fields that do not apply to any other in this dictionary.
deviceCustomFloatingPoint2Label	Varchar(1023)	All custom fields have a corresponding label field. Each of these fields is a string and describes the purpose of the custom field.

Column Name	Data Type	Description
deviceCustomFloatingPoint3	Float	One of four floating point fields available to map fields that do not apply to any other in this dictionary.
deviceCustomFloatingPoint3Label	Varchar(1023)	All custom fields have a corresponding label field. Each of these fields is a string and describes the purpose of the custom field.
deviceCustomFloatingPoint4	Float	One of four floating point fields available to map fields that do not apply to any other in this dictionary.
deviceCustomFloatingPoint4Label	Varchar(1023)	All custom fields have a corresponding label field. Each of these fields is a string and describes the purpose of the custom field.
deviceCustomIPv6Address1Bin	Varbinary(16)	
deviceCustomIPv6Address1Label	Varchar(1023)	
deviceCustomIPv6Address2Bin	Varbinary(16)	
deviceCustomIPv6Address2Label	Varchar(1023)	
deviceCustomIPv6Address3Bin	Varbinary(16)	
deviceCustomIPv6Address3Label	Varchar(1023)	
deviceCustomIPv6Address4Bin	Varbinary(16)	
deviceCustomIPv6Address4Label	Varchar(1023)	
deviceDirection	Varchar(1023)	Any information about what direction the observed communication has taken. The following values are supported: “0” for inbound or “1” for outbound.
deviceDnsDomain	Varchar(255)	The DNS domain part of the complete fully qualified domain name (FQDN).
deviceDomain	Varchar(1023)	

Column Name	Data Type	Description
deviceFacility	Varchar(1023)	The facility generating this event. For example, Syslog has an explicit facility associated with every event.
deviceMacAddressBin	Binary(16)	
deviceNtDomain	Varchar(255)	The Windows domain name of the device address.
deviceProcessId	Integer	Provides the ID of the process on the device generating the event.
deviceProcessName	Varchar(1023)	Process name associated with the event. An example might be the process generating the syslog entry in UNIX.
deviceTimeZone	Varchar(255)	The timezone for the device generating the event.
deviceTranslatedAddressBin	Binary(16)	
deviceTranslatedZoneExternalID	Varchar(200)	
deviceTranslatedZoneURI	Varchar(2048)	The URI for the Translated Zone that the device asset has been assigned to in ArcSight.
deviceZoneExternalID	Varchar(200)	
eventOutcome	Varchar(63)	Displays the outcome, usually as ‘success’ or ‘failure’.
fileCreateTime	Integer	Time when the file was created.
fileHash	Varchar(255)	Hash of a file.
fileId	Varchar(1023)	An ID associated with a file could be the inode.
fileModificationTime	Integer	Time when the file was last modified.
filePermission	Varchar(1023)	Permissions of the file.
fileSize	Integer	Size of the file.

Column Name	Data Type	Description
fileType	Varchar(1023)	Type of file (pipe, socket, etc.)
id	Integer	
locality	Varchar(1023)	
normalizedEventTime	Integer	
oldFileCreateTime	Integer	Time when old file was created.
oldFileHash	Varchar(255)	Hash of the old file.
oldFileId	Varchar(1023)	
oldFileModificationTime	Integer	Time when old file was last modified.
oldFileName	Varchar(1023)	Name of the old file.
oldFilePath	Varchar(1023)	Full path to the old file, including the file name itself. Examples: c:\Program Files\ WindowsNT\Accessories \wordpad.exe or /usr/bin/zip
oldFilePermission	Varchar(1023)	Permissions of the old file.
oldFileSize	Integer	Size of the old file.
oldFileType	Varchar(1023)	Type of the old file (pipe, socket, etc.)
originator	Varchar(1023)	
persistedTime	Integer	
rawEvent	Varchar(4000)	
reason	Varchar(1023)	The reason an audit event was generated. For example “bad password” or “unknown user”. This could also be an error or return code. Example: “0x1234”
requestCookies	Varchar(1023)	Cookies associated with the request.
severity	Integer	

Column Name	Data Type	Description
sourceDnsDomain	Varchar(255)	The DNS domain part of the complete fully qualified domain name (FQDN).
sourceGeoCountryCode	Varchar(1023)	
sourceGeoLatitude	Float	
sourceGeoLocationInfo	Varchar(1023)	
sourceGeoLongitude	Float	
sourceGeoPostalCode	Varchar(1023)	
sourceGeoRegionCode	Varchar(1023)	
sourceProcessId	Integer	The ID of the source process associated with the event.
sourceTranslatedPort	Integer	A port number after being translated by, for example, a firewall. Valid port numbers are 0 to 65535.
sourceTranslatedZoneExternalID	Varchar(200)	
sourceTranslatedZoneURI	Varchar(2048)	The URI for the Translated Zone that the destination asset has been assigned to in ArcSight.
sourceZoneExternalID	Varchar(200)	
version	Varchar(10)	
priority	Integer	
base_event_ids	Varchar(8000)	
correlated_event_id	Integer	
extraFields	Varchar(8192)	

Managing ArcMC

This section provides information about managing ArcMC.

Snapshots

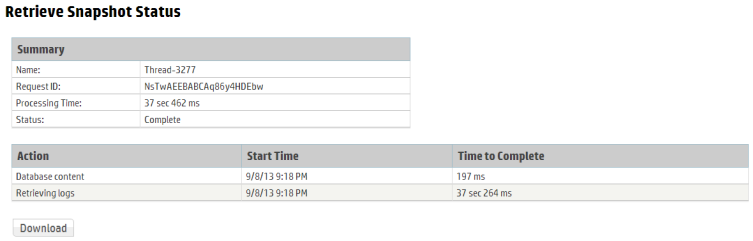
Arcsight Management Center records audit and debug information, including details of any issues that can occur during normal operations. These system logs form a *snapshot* of your Arcsight Management Center activity. System logs are helpful in troubleshooting issues.

ArcSight Customer Support may ask you to retrieve and submit system logs as part of an incident investigation.

The following topics are discussed here.

Creating a Snapshot

Creating a snapshot of Arcsight Management Center creates a set of zipped log files, which you can download locally.



To create a snapshot:

1. Click **Administration > Application > Snapshot.**
2. The **Retrieve Snapshot Status** page displays. Depending on the size of the log files, the snapshot may take a few moments to generate.
3. When ready, click **Download** to download the ZIP file locally.

Submit the snapshot file as instructed by ArcSight Customer Support.

Note: An Arcsight Management Center snapshot does not include information on the activity of the Arcsight Management Center Agent on remotely-managed hosts.

To obtain logs for Arcsight Management Center Agent activity on a managed host, access the remote host. Under **Setup > Appliance Snapshot,** click the **Download** button.

Managing Repositories

Certain management operations require a specific upgrade or content update (`.enc`) file, or a certificate. Other operations, such as viewing logs, require you to load the logs to a Log repository. Arcsight Management Center can also maintain centralized repositories for files needed for host configuration and management.

By default, a number of pre-defined repositories are provided. However, you can create more repositories to suit your needs. Any repositories you create are referred to as *user-defined* repositories.

The following controls are used for repository functions:

- **Retrieve Container Files** copies a file from one or more managed hosts to the repository.
- **Upload to Repository** sends a file from your local computer (the computer running the browser) or a network host accessible from your local computer to the repository.
- **Retrieve** downloads a file from the repository.
- **Upload** copies a file from the repository to one or more managed nodes.

You can perform these operations using repositories:

- Manage logs in the Logs repository
- Manage CA certificates in the CA Certs repository
- Upgrade a connector using an upgrade file available in the Upgrade repository
- Apply a Content ArcSight Update Pack (AUP) on one or more connectors
- Maintain centralized repositories of files for connector configuration and management

The following topics are discussed here.

Logs Repository

To view logs, you need to first **Load** the logs of the container that contains the connector to the Logs repository, then **Retrieve** the logs to view them.

Note: If a container contains more than one connector, logs for all connectors are retrieved.

For information on loading, retrieving, and deleting container logs, see "Viewing Container Logs" on page 955.

Uploading a File to the Logs Repository

Uploading a file into the Log repository is useful for sharing annotated log or other files with other users. An uploaded file needs to be in `.zip` format.

To upload a ZIP file:

1. Click **Administration > Repositories.**
2. Click **Logs** from the left panel.
3. Click **Upload** from the management panel.
4. Specify the local file path or click **Browse** to select the ZIP file.
5. Click **Submit** to add the specified file to the repository or **Cancel** to quit.

Note: Due to a browser limitation in Internet Explorer 11, the progress of the file upload will not be shown.

CA Certs Repository

Connectors require a Certificate Authority (CA) issued or self-signed SSL certificate to communicate securely with a destination. The CA Certs repository (shown below) enables you to store CA Certs files (that contain one or multiple certificates) and single CA certificates. When certificates are stored in the CA Certs repository, you can add the certificates to a container so that the connectors in the container can validate their configured destinations.

You can add a single certificate to a container that is in FIPS or non-FIPS mode. You can only add a CA Certs file to a container that is in non-FIPS mode.

To associate a CA certificate to a connector, you need to:

- Upload the CA certificate or CA Certs file to the CA Certs repository, as described below.
- Add a CA certificate from the CA Certs repository to the container that contains the connector, as described in "Managing Certificates on a Container" on page 959.

Uploading CA Certificates to the Repository

You can upload a CA Certs file or a single certificate to the CA Certs repository.

Tip: Before you upload a single CA certificate, change the name of the certificate on the local computer to a name that you can recognize easily. This helps you distinguish the certificate when it is displayed in the Certificate Management wizard.

To upload certificates to the repository:

1. Click **Administration > Repositories.**
2. Click **CA Certs** in the left panel.
3. Click **Upload** in the management panel.
4. Specify the local path for the CA Certs file or the certificate, or click **Browse** to select it.
5. Click **Submit** to add the specified CA Certs file or the certificate to the repository, or **Cancel** to quit.

The CA Certs Repositories tab shows all the CA Certs files and single certificates that have been uploaded. The Type column shows CERTIFICATE for a single certificate and CACERT for a CA Certs file.

Removing CA Certificates from the Repository

When you delete a CA Certs file or a single certificate from the repository, it is deleted from Arcsight Management Center.

Note: When you delete a CA Certs file or a single certificate from the CA Certs repository, containers are not affected; the connectors continue to use the certificates, which are located in a trust store after being added to a container. For information about adding a CA certificate to a container, see "Managing Certificates on a Container" on page 959.

To remove a certificate from the repository:

1. Click **Administration > Repositories.**
2. Click **CA Certs** in the left panel.
3. Identify the certificate or the CA Certs file you want to remove and click the **Remove** button (✕).

Upgrade Files Repository

The Upgrade files repository enables you to maintain a number of connector upgrade files. You can apply any of these upgrade files to containers when you need to upgrade to a specific version. As a result, all connectors in a container are upgraded to the version you apply to the container.

Note: Logger ENC files are required for the remote upgrade of a Logger Appliance. For more information, see "Upgrading a Logger " on page 948.

About the AUP Upgrade Process

Note: The process discussed in this section only applies to upgrading connectors and to upgrading a remotely-managed Connector Appliance.

To upgrade a connector or to upgrade a remotely-managed Connector Appliance, you need to:

- Upload the appropriate `.aup` upgrade file to the Upgrade Files repository, as described below.
- Apply the `.aup` upgrade file from the Upgrade Files repository to the container (see "Upgrading All Connectors in a Container" on page 953).

Uploading an AUP Upgrade File to the Repository

To upload AUP upgrade files to the repository:

1. Download the upgrade files for the connector or the remote Connector Appliance from the ArcSight Customer Support site at https://softwaresupport.softwaregrp.com/ to the computer that you use to connect to the browser-based interface.

2. Log in to the browser-based interface.
3. Click **SetupConfiguration** > **Administration** > **Repositories.**
4. Click **Upgrade AUP** from the left panel.
5. Click **Upload** from the management panel.
6. Click **Browse** and select the file you downloaded earlier.
7. Click **Submit** to add the specified file to the repository or click **Cancel** to quit.
8. You can now use the AUP upgrade file to upgrade a container to a specific version. For instructions, see "Upgrading All Connectors in a Container" on page 953.

Removing a Connector Upgrade from the Repository

You can remove a connector upgrade file from the repository when you no longer need it. When you remove a connector upgrade file from the repository, it is deleted from Arcsight Management Center.

To remove a Connector upgrade from the repository:

1. Click **SetupConfiguration** > **Administration** > **Repositories.**
2. Click **Upgrade AUP** from the left panel.
3. Locate the upgrade file that you want to delete and click the associated ✕ icon.

Content AUP Repository

ArcSight continuously develops new connector event categorization mappings, often called *content*. This content is packaged in ArcSight Update Packs (AUP) files. All existing content is included with major product releases, but it is possible to stay completely current by receiving up-to-date, regular content updates through ArcSight announcements and the Customer Support site. The AUP files are located under Content Subscription Downloads.

The ArcSight Content AUP feature enables you to apply an AUP file to applicable connector destinations that you are managing. Only the event categorization information can be applied to the connectors using this feature.

You can maintain a number of Content AUP files in the Content AUP repository. When an AUP file with a version number higher than the ones already in the repository is loaded, it is automatically pushed out to the connector destinations being managed. However, these connectors or connector destinations are skipped:

- Connectors that are unavailable at the time of the AUP file push
- Connectors whose current version does not fall in the range of versions that the Content AUP supports

- The ESM destination on a connector
- All destinations of a connector that have an ESM destination with the AUP Master flag set to Yes

Also, when a new connector is added, the highest number Content AUP is pushed automatically to its destinations.

Applying a New Content AUP

You can add a new content AUP file to the repository and push it automatically to all applicable managed nodes.

To apply a new Content AUP:

1. Download the new Content AUP version from the support site at https://softwaresupport.softwaregrp.com/ to the computer that you use to connect to the browser-based interface.
2. Log in to the browser-based interface.
3. Click **Administration > Repositories.**
4. Click **Content AUP** from the left panel.
5. Click **Upload** from the management panel.
6. Click **Browse** and select the file you downloaded earlier.
7. Click **Submit** to add the specified file to the repository and push it automatically to all applicable connectors, or **Cancel** to quit.

You can verify the current Content AUP version on a connector by performing either of these steps:

- Run the `GetStatus` command on the node destination and check that the value for aup `[acp].version` is the same as the AUP version you applied. For information about running a command on a connector destination, see "Sending a Command to a Connector" on page 977.
- Hover over a host name to see the AUP version applied to all destinations of that connector.

Applying an Older Content AUP

If you need to apply an older Content AUP from the Content AUP repository, delete all versions newer than the one you want to apply in the repository. The latest version (of the remaining AUP files) is pushed automatically to all applicable connectors.

To delete a Content AUP from the Content AUP repository:

1. Click **Administration > Repositories.**
2. Click **Content AUP** from the left panel.
3. Locate the AUP file that you want to delete and click the associated  icon. Repeat for multiple files.

User-Defined Repositories

A *user-defined repository* is a user-named collection of settings that control upload and download of particular files from connectors to the repository. Each repository uses a specified path, relative to `$ARCSIGHT_HOME/user/agent`, for files to be uploaded or downloaded. ArcSight connectors use a standard directory structure, so map files, for example, are always found in `$ARCSIGHT_HOME/user/agent`, (that is, the root directory, `$ARCSIGHT_HOME`, of the installation path) in a folder called `map/`.

After they are created, user-defined repositories are listed on the left-side menu, under the **New Repository** heading, and appear with the user-specified display name.

User-defined repositories should be grouped by file type and purpose, such as log files, certificate files, or map files. Each user-defined repository has a name, a display name, and an item display name, which are described under the repository **Settings** tab.

Files viewed in a user-defined repository can be bulk processed with specified hosts and can be exchanged with the user's browser host.

Creating a User-Defined Repository

You can create a new repository at any time.

The repository requires correct directory paths. Your file will be applied to the wrong directory if the entered path contains errors, such as extra spaces or incorrect spellings. You can verify your directory paths by accessing the Directory.txt file, which lists the directory structure for every entered path. View the Directory.txt file by accessing your container logs and finding the Directory.txt file.

To create a new user-defined repository:

1. Click **Administration > Repositories.**
2. Click **New Repository** under the **Repositories** section in the left panel.

3. For the new repository, specify the parameters listed in the following table.

Parameter	Description
Name	A unique name for the repository, typically based on the type of files it contains.
Display Name	The name that will be displayed on the left-side menu and for tabs: Process *names*, View *names*, Settings for *names*. Typically plural.
Item Display Name	The name used to describe a single item.
Recursive	Check to include sub-folders.
Sort Priority	-1 by default
Restart Connector Process	Check to restart the connector process after file operations.
Filename Prefix	An identifying word that is included in the names of retrieved files. For example, map files are identified by `Map` in the file name: `localhost_Container_-1.Map-2009-04-06_12-22-25-607.zip`
Relative path (Download)	The path for download, relative to `$ARCSIGHT_HOME`, for example, `user/agent/map` or `user/agent/flexagent`. Leave this field blank to specify files in `$ARCSIGHT_HOME`. **Note**: The relative path is used for download only.
Include Regular Expression	A description of filenames to include. Use `.*` to specify all files. The following example selects properties files that consist of `map.` followed by one or more digits, followed by `.properties`: `map\.[0-9]+\.properties$`
Exclude Regular Expression	A description of filenames to exclude. The following example excludes all files with a certain prefix or in the `agentdata` folder. `(agentdata/|cwsapi_fileset_).*$`
Delete Before Upload	Check to delete earlier copies before upload. **CAUTION**: If you check **Delete Before Upload** and do not specify a Relative path (Upload), all files and folders in `current/user/agent` will be deleted.
Delete Groups	Whether to delete folders recursively in `$ARCSIGHT_HOME/user/agent/map directory`.
Relative path (Upload)	The path for upload, relative to `$ARCSIGHT_HOME/current/user/agent/flexagent/`<*connectorname*>
Delete Relative Path	Whether the directory specified in **Relative Path (Upload)** and its contents should be removed when a file is uploaded from the repository.
Delete Include Regular Expression	Typically the same as the Include Regular Expression.
Delete Exclude Regular Expression	Typically the same as the Exclude Regular Expression.

4. Click **Save** at the bottom of the page.

   The new repository displays under the **New Repository** heading in the left-side window panel.

Retrieving Container Files

The **Retrieve Container Files** button copies a file from one or more containers to a repository. The specific files that are retrieved depend on the settings of the repository.

To retrieve a container file:

1. Click **Administration > Repositories.**
2. In the left panel, under **Repositories**, click the name of the repository to which you want to copy connector files.
3. Click **Retrieve Container Files** in the management panel.
4. Follow the instructions in the Retrieve Container Files wizard.

Uploading Files to a Repository

To upload files to a repository:

1. Click **Administration > Repositories.**
2. In the lower left panel (under **Repositories**), click the name of the repository to which you want to upload files.
3. Click **Upload To Repository** from the management panel.
4. Follow the instructions in the Repository File Creation wizard. Select **Individual files** to create a ZIP file with appropriate path information.

   Caution: Be sure **not** to change the default sub-folder name `lib` in the **Enter the sub folder where the files will be uploaded** page of the Repository File Creation wizard.

Deleting a User-Defined Repository

To delete a user-defined repository:

1. Click **Administration > Repositories.**
2. From the left panel, click the name of the repository you want to delete.
3. Click **Remove Repository** from the management panel.

Updating Repository Settings.

To update the settings of a user-defined repository:

1. Click **Administration > Repositories.**
2. In the left panel, click the name of the repository whose settings you want to update.
3. Click the **Settings for *Repository_Name*** tab from the management panel.
4. Update the settings.
5. Click **Save** at the bottom of the page.

Managing Files in a Repository

Retrieving a File from the Repository

To retrieve a file from the repository:

1. Click **Administration > Repositories.**
2. From the left panel, click the name of the repository in which the file exists.
3. Click ✕ from the management panel for the file that you want to retrieve.
4. Follow the file download instructions to copy the file to your local computer.

Uploading a File to the Repository

To upload a file to the repository:

1. Click **Administration > Repositories.**
2. In the left panel, click the name of the repository in which the file exists.
3. In the management panel, click **Upload to Repository** for the file that you want to upload.
4. Follow the Upload Container Files wizard instructions to upload the file to the containers of your choice.
5. Verify that the file was uploaded correctly:
   - If you have SSH access to the connectors, connect to them and check the file structure.
   - Obtain the connector logs and check the contents of the `Directory.txt` file for each connector.

Removing a File from the Repository

To remove a file from the repository:

1. Click **Administration > Repositories.**
2. In the left panel, click the name of the repository in which the file exists.
3. In the management panel, click ✕ for the file that you want to delete.

Pre-Defined Repositories

You can define repositories for any connector-related files. The following repositories are pre-defined:

- **Backup Files**: connector cloning (see " Backup Files" on page 661).
- **Map Files**: enrich event data
- **Parser Overrides**: customize the parser (see "Adding Parser Overrides" on page 661)
- **FlexConnector Files**: user-designed connector deployment
- **Connector Properties**: `agent.properties`; subset of cloning
- **JDBC Drivers**: database connectors

To view the settings for a pre-defined repository, click the name of the repository and then click the **Settings** tab in the management panel. Settings for a pre-defined repository are read-only.

Settings for Map Files

This table lists the default settings for map files.

Map File Settings

Name	Default Setting
Name	map
Display Name	Map Files
Item Display Name	Map File
Recursive	Deselected (No)
Sort Priority	5
Restart Connector Process	Deselected (No)
Filename Prefix	Map
Download Relative Path	map

Map File Settings, continued

Name	Default Setting
Download Include regular expression	map\.[0-9]+\.properties$
Download Exclude regular expression	
Delete before upload	Selected (Yes)
Delete groups	Deselected (No)
Upload Relative Path	
Delete Relative Path	map
Delete Include regular expression	map\.[0-9]+\.properties$
Delete Exclude regular expression	

Settings for Parser Overrides

This table lists the default settings for parser overrides.

Parser Override Settings

Name	Default Setting
Name	parseroverrides
Display Name	Parser Overrides
Item Display Name	Parser Override
Recursive	Selected (Yes)
Sort Priority	10
Restart Connector Process	Selected (Yes)
Filename Prefix	Parsers
Download Relative Path	fcp
Download Include regular expression	.*
Download Exclude regular expression	
Delete before upload	Selected (Yes)
Delete groups	Selected (Yes)
Upload Relative Path	
Delete Relative Path	fcp
Delete Include regular expression	.*
Delete Exclude regular expression	

Settings for FlexConnector Files

This table lists the default settings for FlexConnector files.

FlexConnector Settings

Name	Default Setting
Name	flexconnectors
Display Name	FlexConnector Files
Item Display Name	FlexConnector File
Recursive	Selected (Yes)
Sort Priority	15
Restart Connector Process	Selected (Yes)
Filename Prefix	FlexConnector
Download Relative Path	flexagent
Download Include regular expression	.*
Download Exclude regular expression	
Delete before upload	Selected (Yes)
Delete groups	Selected (Yes)
Upload Relative Path	
Delete Relative Path	flexagent
Delete Include regular expression	.*
Delete Exclude regular expression	

Settings for Connector Properties

Connector Default Property Settings

Name	Default Setting
Name	connectorproperties
Display Name	Connector Properties
Item Display Name	Connector Property File
Recursive	Deselected (No)
Sort Priority	20
Restart Connector Process	Selected (Yes)
Filename Prefix	ConnectorProperties

Connector Default Property Settings, continued

Name	Default Setting
Download Relative Path	
Download Include regular expression	agent\.*
Download Exclude regular expression	
Delete before upload	Deselected (No)
Delete groups	Deselected (No)
Upload Relative Path	
Delete Relative Path	
Delete Include regular expression	agent\.*
Delete Exclude regular expression	

Settings for JDBC Drivers

This table lists the default settings for JDBC Drivers.

JDBC Driver Settings

Name	Default Setting
Name	jdbcdrivers
Display Name	JDBC Drivers
Item Display Name	Connector JDBC Driver File
Recursive	Deselected (No)
Sort Priority	25
Restart Connector Process	Selected (Yes)
Filename Prefix	
Download Relative Path	lib
Download Include regular expression	
Download Exclude regular expression	
Delete before upload	Deselected (No)
Delete groups	Deselected (No)
Upload Relative Path	
Delete Relative Path	lib
Delete Include regular expression	
Delete Exclude regular expression	

Backup Files

Using the **Backup Files** repository, you can quickly copy a container to other containers. As a result, all connectors in the source container are copied to the destination container. This process is called *cloning* a container configuration. You can clone a container to several containers at once. The contents of the source container replace the existing contents of the destination container.

Caution: Containers on Arcsight Management Center are pre-installed with the latest connector release. Do not clone older, software-based connectors (such as build 4.0.8.4964) to containers with newer connector builds (such as 4.0.8.4976 or later).

Cloning a connector using the Backup repository only works if the connector version numbers are the same.

To clone a container using the Backup Files repository:

1. Click **Node Management > View All Nodes.**
2. Click the **Containers** tab to list the containers and determine the source and destination for cloning.
3. Click **Administration > Repositories.**
4. Click **Backup Files** under the **Repositories** section in the management panel.
5. If the backup file that you need to use for cloning exists in the repository, go to the next step. Otherwise, follow the instructions in "Retrieving a File from the Repository" on page 656 to retrieve the container's backup file to the Backup repository.

   The retrieved filename is in the format *<connector name>* ConnectorBackup *<date>*.
6. Follow the instructions in "Uploading a File to the Repository" on page 656 to upload the backup file to one or more containers.

   The destination containers are unavailable while the backup file is applied and the connectors are restarted.

   Note: The backup file does not include the container certificates. You have to re-apply the certificates to the container after you upload the backup file.

   After applying the certificates, check the status of the destination container to make sure it is available.

Adding Parser Overrides

A parser override is a file provided by ArcSight used to resolve an issue with the parser for a specific connector, or to support a newer version of a supported device where the log file

format changed slightly or new event types were added.

To use parser overrides, you need to:

- Upload a parser override file to the **Parser Overrides** repository.
- Download the parser override file to the container that contains the connector that will use the parser override.

Follow the steps below.

To upload a parser override file:

1. Click **Administration > Repositories.**
2. Click **Parser Overrides** under the **Repositories** section in the management panel.
3. On the **Parser Overrides** tab, click the **Upload To Repository** button.
4. Follow the wizard to upload the file. When prompted by the wizard, make sure you:
   - Select the **Individual Files** option from the **Select the type of file that you want to upload** field.
   - Add a slash (/) after fcp before adding the folder name in the **Enter the sub folder where the files will be uploaded** field. For example, `fcp/multisqlserverauditdb`.

     Note: The folder name may only contain letters and numbers. Do not include special characters such as (,), <, or >.

     When the upload is complete, the parser override file is listed in the table on the **Parser Overrides** tab.

To download the parser override file to a container:

1. Click **Administration > Repositories.**
2. Click **Parser Overrides** under the **Repositories** section in the management panel.
3. In the table on the **Parser Overrides** tab, locate the parser override file you want to download and click the up arrow next to the file.
4. Follow the wizard to select the container to which you want to add the parser overrides.

   When the wizard completes, the parser overrides are deployed in the selected container.

   Note: You can download a parser override file from ArcExchange. For more information, refer to "Sharing Connectors in ArcExchange" on page 981.

To verify that the parser override has been applied successfully, issue a Get Status command to the connector. See "Sending a Command to a Connector" on page 977. In the report that appears, check for the line starting with `ContentInputStreamOverrides`.

Audit Logs

The following topics are discussed here.

Audit Event Types

You can forward Arcsight Management Center application audit events, which are in Common Event Format (CEF), to a destination of your choice.

Several types of audit events are generated by Arcsight Management Center:

- **Application events:** related to Arcsight Management Center functions and configuration changes
- **Platform events:** related to the Arcsight Management Center system
- **System health events:** related to Arcsight Management Center health.

Audit Event Information

An Arcsight Management Center audit event contains information about the following prefix fields.

- Device Event Class ID
- Device Severity
- Name
- Device Event Category (`cat`)

See "Audit Logs" on page 1 for details on how to generate audit logs.

Note: If no Syslog Daemon connector is installed or configured on your local machine, then no audit events will be visible.

Application Events

Application Events

Signature	Severity	Description	deviceEventCategory
Connector			
connector:101	1	Register connector successful	/Connector/Add/Success

Application Events, continued

Signature	Severity	Description	deviceEventCategory
connector:102	1	Connector removed successfully	/Connector/Delete
connector:103	1	Update connector parameters successful	/Connector/Parameter/Update/Success
connector:104	1	AUP Package create successful	/Connector/AUP Package/Create/Success
connector:105	1	AUP Package deploy successful	/Connector/AUP Package/Deploy/Success
connector:201	1	Connector add failed	/Connector/Add/Fail
connector:202	1	Connector delete failed	/Connector/Delete/Fail
connector:203	1	Connector parameters update failed	/Connector/Parameter/Update/Fail
Arcsight Management Center			
arcmc:101	1	ConfigurationBackupScheduler add success	/BackupScheduler/Add/Success
arcmc:102	1	ConfigurationBackupScheduler update successful	/BackupScheduler/Update/Success
arcmc:103	1	ConfigurationBackupScheduler delete success	/BackupScheduler/Delete/Success
arcmc:104	1	Scheduled Backup triggered	/Backup/Scheduled/Trigger
arcmc:105	1	Scheduled Backup completed	/Backup/Scheduled/Complete/Success
arcmc:106	1	Manual Backup completed	/Backup/Manual/Complete/Success
arcmc:107	1	Local Backup completed	/Backup/Local/Complete/Success
arcmc:108	1	You have exceeded the maximum number of managed connectors allowed by your license	/RemotelyManagedConnectors/Exceeded
arcmc:110	1	You have attempts to exceed the maximum number of managed products allowed by your license	/managedproducts/exceeded

Application Events, continued

Signature	Severity	Description	deviceEventCategory
arcmc:111	1	Reboot command launched successfully	Node/reboot/launched/Success
arcmc:112	1	New configuration created successfully	/Configuration/Add/Success
arcmc:113	1	Edit configuration succesful	/Configuration/Edit/Success
arcmc:114	1	Delete configurations successful	/Configuration/Delete/Success
arcmc:115	1	Push configuration successful	/Configuration/Push/Success
arcmc:116	1	Import configuration successful	/Configuration/Import/Success
arcmc:117	1	Add subscriber to configuration successful	/Configuration/Subscribe/ Success
arcmc:118	1	Unsubscribe node for configuration successful	/Configuration/Unsubscribe/ Success
arcmc:119	1	Check compliance of configuration successful	/Configuration/Check Compliance/Success
arcmc:120	1	Configuration set successfully	/Node/Set/Configuration/ Success
arcmc:121	1	Configuration appended successfully	/Node/Append/Configuration/ Success
arcmc:122	1	Agent install success	/ArcMCAgent/Install/Success
arcmc:123	1	Upgrade agent successfully	/ArcMCAgent/Upgrade/Success
arcmc:124	1	Add/Push Logger Peers Successful	/Logger/AddPeers/Success
arcmc:125	1	Remove Logger Peers Successful	/Logger/RemovePeers/Success
arcmc:127	1	Create/Import Logger Peer Group Successful	/Logger/AddPeerGrp/Success
arcmc:128	1	Delete Logger Peer Group Successful	/Logger/DeletePeerGrp/Success
arcmc:129	1	Edit Logger Peer Group Successful	/Logger/EditPeerGrp/Success

Application Events, continued

Signature	Severity	Description	deviceEventCategory
arcmc:130	1	Import Initial Configuration Successful	/Logger/ImportInitConfig/Success
arcmc:131	1	Pushed Initial Configuration	/Logger/PushInitConfig/Success
arcmc:132	1	Deleted Initial Configuration	/Logger/DelInitConfig/Success
arcmc:133	1	Host upgrade started.	/Node/Upgrade/Start
arcmc:134	1	Host upgrade successful.	/Node/Upgrade/Success
arcmc:138	1	Update rule/s	/ArcMC/UpdateRules/Success
arcmc142	1	Rule add success	/ArcMC/AddRule" + SUCCESS
arcmc143	1	Rule delete success	/ArcMC/DeleteRule" + SUCCESS
arcmc:201	1	ConfigurationBackupScheduler add failed	/BackupScheduler/Add/Fail
arcmc:202	1	ConfigurationBackupScheduler update failed	/BackupScheduler/Update/Fail
arcmc:203	1	ConfigurationBackupScheduler delete failed	/BackupScheduler/Delete/Fail
arcmc:205	1	Scheduled Backup failed	/Backup/Scheduled/Complete/Fail
arcmc:206	1	Manual Backup failed	/Backup/Manual/Complete/Fail
arcmc:212	1	New configuration creation failed	/Configuration/Add/Fail
arcmc:213	1	Edit configuration failed	/Configuration/Update/Fail
arcmc:214	1	Configuration deletion failed	/Configuration/Delete/Fail
arcmc:215	1	Push configuration failed	/Configuration/Import/Fail
arcmc:216	1	Import configuration failed	/Backup/Local/Push/Fail
arcmc:217	1	Add subscriber to configuration failed	/Configuration/Subscribe/Fail
arcmc:218	1	Unsubscribe node for configuration failed	/Configuration/Unsubscribe/Fail
arcmc:219	1	Check compliance of configuration failed	/Configuration/Check Compliance/Success

Application Events, continued

Signature	Severity	Description	deviceEventCategory
arcmc:220	1	Configuration set failed	/Node/Set/Configuration/ Fail
arcmc:221	1	Configuration append failed	/Node/Append/Configuration/ Fail
arcmc:222	1	Agent install failed	/ArcMCAgent/Install/Failure
arcmc:223	1	Upgrade agent failed	/ArcMCAgent/Upgrade/Fail
arcmc:224	1	Add/Push Logger Peers Failed	/Logger/AddPeers/Fail
arcmc:225	1	Remove Logger Peers Failed	/Logger/RemovePeers/Fail
arc mc:226	1	Alert message payload	/ArcMCMonitor/Breach
arcmc:230	1	Import Initial Configuration Failed	/Logger/ImportInitConfig/Fail
arcmc:234	1	Host upgrade failed.	/Node/Upgrade/Fail
arcmc:250	1	Push user assignment <assignment name>	/ArcMCUM/Push
arcmc:251	1	Decommission user <UserName>	/ArcMCUM/DeleteUser
arcmc:252	1	Add user <UserName>	/ArcMCUM/AddUser
Destination			
destination:102	1	Update destination successful	/Connector/Destination/Update/Success
destination:103	1	Remove destination successful	/Connector/Destination/Delete/Success
destination:104	1	Update destination configuration successful	/Connector/Destination/ Configuration/Update/Success
destination:105	1	Register destination successful	/Connector/Destination/ Registration/Success
destination:106	1	Create destination configuration successful	/Connector/Destination/ Configuration/Add/Success
destination:107	1	Destination configuration delete successful	/Connector/Destination/ Configuration/Delete/Success
destination:202	1	Destination update to a connector failed	/Connector/Destination/Update/Fail

Application Events, continued

Signature	Severity	Description	deviceEventCategory
destination:203	1	Destination delete from a connector failed	/Connector/Destination/Delete/Fail
destination:204	1	Destination configuration update failed	/Connector/Destination/Configuration/Update/Fail
destination:205	1	Register destination failed	/Connector/Destination/Registration/Fail
destination:206	1	Destination configuration add failed	/Connector/Destination/Configuration/Add/Fail
destination:207	1	Destination configuration delete failed	/Connector/Destination/Configuration/Delete/Fail
Container			
container:101	1	Container upgrade successful	/Container/Upgrade/Success
container:102	1	Push user file successful	/Container/UserFiles/Push/Success
container:103	1	User file delete from container	/Container/UserFiles/Delete
container:104	1	CA cert push to a container successful	/Container/CACert/Push/Success
container:105	1	Container demo CA enable successful	/Container/DemoCA/Enable/Success
container:106	1	Container demo CA disable successfu	/Container/DemoCA/Disable/Success
container:109	1	Delete property from a container successful	/Container/Property/Delete/Success
container:110	1	Modify properties successful	/Container/Property/Update/Success
container:111	1	Container password update successful	/Container/Password/Update/Success
container:112	1	Container add successful	/Container/Add/Success
container:113	1	Container edit	/Container/Update
container:114	1	Remove container	/Container/Delete
container:115	1	Add certificate for a container successful	/Container/Certificate/Add/Success

Application Events, continued

Signature	Severity	Description	deviceEventCategory
container:116	1	Removing certificates successful [addtrust class 1ca]	/Container/Certificate/Delete/ Success
container:117	1	Enabling FIPS mode successful	/Container/FIPS/Enable/Success
container:118	1	Disabling FIPS mode successful	/Container/FIPS/Disable/ Success
container:119	1	Upgrade was triggered for container that resides on end of life appliance model	Container/FromEndOfLifeModel/Upgrade/Triggered
container:123	1	Emergency restore failed	/Container/EmergencyRestore/Fail
container:201	1	Container upgrade failed	/Container/Upgrade/Fail
container:202	1	User file push to a container failed	/Container/UserFiles/Push/Fail
container:204	1	CA cert push to a container failed	/Container/CACert/Push/Fail
container:205	1	Enable demo CA for a container failed	/Container/DemoCA/Enable/Fail
container:206	1	Disable demo CA for a container failed	/Container/DemoCA/Disable/ Fail
container:209	1	Delete property from a container failed	/Container/Property/Delete/Fail
container:210	1	Update property to a container failed	/Container/Property/Update/Fail
container:211	1	Container password update failed	/Container/Password/Update/ Fail
container:212	1	Container add failed	/Container/Add/Fail
container:215	1	Add certificate for a container failed	/Container/Certificate/Add/Fail
container:216	1	Delete certificate for a container failed	/Container/Certificate/Delete/ Fail
container:217	1	Enable FIPS on a container failed	/Container/FIPS/Enable/Fail
container:218	1	Disable FIPS on a container failed	/Container/FIPS/Disable/Fail

Application Events, continued

Signature	Severity	Description	deviceEventCategory
container:219	1	SSL Certificate downloaded successfully	/Container/Certificate/Download/Success
container:220	1	SSL Certificate download failed	/Container/Certificate/Download/Fail
container:221	1	SSL Certificate imported successfully	/Container/Certificate/Import/ Success
container:222	1	SSL Certificate import failed	/Container/Certificate/Import/ Fail
container:226	1	Emergency restore successful	/Container/EmergencyRestore/Success
container:301	1	Container upgrade started	/Container/Upgrade/Start
Transformation Hub			
eventbroker:146	1	Transformation Hub Add Topic succesful	/EventBroker/Topic/Add/Success
eventbroker:147	1	Transformation Hub delete route/s successful	/EventBroker/Route/Add/Success
eventbroker:148	1	Transformation Hub Add Route/s successful	/EventBroker/Route/Add/Success
eventbroker:149	1	Transformation Hub Update Route successful	/EventBroker/Route/Update/Success
eventbroker:241	1	Transformation Hub Add Topic failed	/EventBroker/Topic/Add/Fail
eventbroker:242	1	Transformation Hub delete route/s failed	/EventBroker/Route/Add/Fail
eventbroker:243	1	Transformation Hub Add Route failed	/EventBroker/Route/Add/Fail
eventbroker:244	1	Transformation Hub Update Route failed	/EventBroker/Route/Update/Fail
Location			
location:101	1	Location add successful	/Location/Add/Success
location:102	1	Location edit	/Location/Update
location:103	1	Remove location	/Location/Delete
location:201	1	Location add failed	/Location/Add/Fail
Host			

Application Events, continued

Signature	Severity	Description	deviceEventCategory
host:101	1	Host add successful	/Host/Add/Success
host:103	1	Remove host	/Host/Delete
host:105	1	Host certificate download and import successful	/Host/Certificate/Download /Import/Success
host:201	1	Host add failed	/Host/Add/Fail
host:205	1	Host certificate download and import failed	/Host/Certificate/Download Import/Fail
host:363	1	Move Host	/Host/Move/Success
host:364	1	Move Host	/Host/Move/Fail
Marketplace			
marketplace:150	1	Successfully saved Marketplace user in ArcMC	/Marketplace/User/Add/Success
marketplace:245	1	Failed to save Marketplace user in ArcMC	/Marketplace/User/Add/Fail
Deployment Templates			
deploymenttemplates:151	1	Successfully deleted template instance(s) in ArcMC	/DeploymentTemplates/TemplateInstance/Delete/Success
deploymenttemplates:246	1	Failed to delete template instance(s) in ArcMC	/DeploymentTemplates/TemplateInstance/Delete/Fail
deploymenttemplates:152	1	Successfully added template instance in ArcMC	/DeploymentTemplates/TemplateInstance/Add/Success
deploymenttemplates:247	1	Failed to add template instance in ArcMC	/DeploymentTemplates/TemplateInstance/Add/Fail
deploymenttemplates:153	1	Successfully updated template instance in ArcMC	/DeploymentTemplates/TemplateInstance/Update/Success
deploymenttemplates:248	1	Failed to update template instance in ArcMC	/DeploymentTemplates/TemplateInstance/Update/Fail
Generator ID			

Application Events, continued

Signature	Severity	Description	deviceEventCategory
generatorid:157	1	Generate ID create successful	/GeneratorID/Add/Success
generatorid:251	1	Generate ID create failed	/GeneratorID/Add/Fail
generatorid:158	1	Generate ID edit successful	/GeneratorID/Update/Success
generatorid:158	1	Generate ID edit failed	/GeneratorID/Update/Fail
generatorid:159	1	Generate ID delete successful	/GeneratorID/Delete/Success
generatorid:159	1	Generate ID delete failed	/GeneratorID/Delete/Fail

Platform Events

Platform Events

Signature	Severity	Definition	Category
platform:200	7	Failed password change	/Platform/Authentication/ PasswordChange/Failure
platform:201	7	Failed login attempt	/Platform/Authentication/Failure/ Login
platform:202	5	Password changed	/Platform/Authentication/ Password
platform:203	7	Login attempt by inactive user	/Platform/Authentication/ InactiveUser/Failure
platform:205	7	Automated password reset attempt made for admin account	/Platform/Authentication/PasswordChange/AdminFailure
platform:206	7	Failed automated password reset attempt for user	/Platform/Authentication/PasswordChange/Failure
platform:207	7	Automated password reset attempted for non-existent user	/Platform/Authentication/PasswordChange/UnknownUser
platform:213	7	Audit forwarding modified	/Platform/Configuration/Global/ AuditEvents
platform:220	5	Installed certificate	/Platform/Certificate/Install

Platform Events, continued

Signature	Severity	Definition	Category
platform:221	7	Certificate mismatch failure	/Platform/Certificate/Mismatch
platform:222	1	Created certificate signing request	/Platform/Certificate/Request
platform:224	5	Re-generate self-signed certificate	/Platform/Certificate/Regenerate
platform:226	7	Uploaded update file damaged or corrupt	/Platform/Update/Failure/CorruptPackage
platform:227	5	Update installation success	/Platform/Update/Applied
platform:228	7	Update installation failure	/Platform/Update/Failure/Installation
platform:230	3	Successful login	/Platform/Authentication/Login
platform:234	7	Failed login attempt (LOCKED)	/Platform/Authentication/Failure/LOCKED
platform:239	1	User logout	/Platform/Authentication/Logout
platform:240	3	Added user group	/Platform/Groups/Add
platform:241	3	Updated user group	/Platform/Groups/Update
platform:242	5	Removed all members from group	/Platform/Authorization/Groups/Membership/Update/Clear
platform:244	3	Deleted user group	/Platform/Groups/Remove
platform:245	3	Added user	/Platform/Users/Add
platform:246	3	Updated user	/Platform/Users/Update
platform:247	3	Deleted user	/Platform/Users/Delete
platform:248	3	Session expired	/Platform/Authentication/Logout/SessionExpiration
platform:249	7	Account locked	/Platform/Authentication/AccountLocked
platform:250	3	Added remote mount point	/Platform/Storage/RFS/Add
platform:251	5	Edited remote mount point	/Platform/Storage/RFS/Edit
platform:252	7	Failed to create remote mount point	/Platform/Storage/RFS/Failure

Platform Events, continued

Signature	Severity	Definition	Category
platform:253	5	Removed remote mount point	/Platform/Storage/RFS/Remove
platform:260	5	Static route modified	/Platform/Configuration/Network/Route/Update
platform:261	5	Static route removed	/Platform/Configuration/Network/Route/Remove
platform:262	5	Appliance time modified	/Platform/Configuration/Time
platform:263		NIC settings modified	/Platform/Configuration/NIC
platform:264		NTP server settings modified	/Platform/Configuration/NTP
platform:265	5	DNS settings modified	/Platform/Configuration/Network/DNS
platform:266	5	Hosts file modified	/Platform/Configuration/Network/Hosts
platform:267	5	SMTP settings modified	/Platform/Configuration/SMTP
platform:268	5	Static route added	/Platform/Configuration/Network/Route/Add
platform:269	5	Updated Platform Settings	/Platform/Configuration
platform:280	7	Appliance reboot initiated	/Appliance/State/Reboot/Initiate
platform:281	3	Appliance reboot canceled	/Appliance/State/Reboot/Cancel
platform:282	9	Appliance poweroff initiated	/Appliance/State/Shutdown
platform:284	5	Enabled SAN Multipathing	/Platform/Storage/Multipathing/Enable
platform:285	5	Disabled SAN Multipathing	/Platform/Storage/Multipathing/Disable
platform:300	5	Installed trusted certificate	/Platform/Certificate/Install
platform:301	5	Installed certificate revocation list	/Platform/Certificate/Revocation/ Install
platform:302	5	Deleted trusted certificate	/Platform/Certificate/Delete
platform:303	5	Deleted certificate revocation list	/Platform/Certificate/Revocation/ Delete

Platform Events, continued

Signature	Severity	Definition	Category
platform:304	7	Failed installing trusted certificate	/Platform/Certificate/Install/Failure
platform:305	7	Failed installing certificate revocation list	/Platform/Certificate/Revocation/ Install/Failure
platform:306	5	Start process	/Platform/Process/Start
platform:307	5	Stop process	/Platform/Process/Stop
platform:308	5	Restart process	/Platform/Process/Restart
platform:310	5	Enabled FIPS mode	/Platform/Configuration/FIPS/Enable
platform:311	7	Disabled FIPS mode	/Platform/Configuration/FIPS/Disable
platform:312	7	Web server cipher strength changed	/Platform/Configuration/WebServer/CipherStrength
platform:313	5	Enable SSH	/Platform/Configuration/SSH/Enable
platform:314	7	Disable SSH	/Platform/Configuration/SSH/Disable
platform: 315	7	Enable SSH only during startup/reboot	/Platform/Configuration/SSH/StartupOnly
platform:316	7	Enable SSH only for 8 hours	/Platform/Configuration/SSH/Enable8Hours
platform: 320	3	Appliance poweroff canceled	/Appliance/State/Shutdown/Cancel
platform:371	5	Restarted OS service	/Platform/Service/Restart
platform:400	1	Ran diagnostic command	/Platform/Diagnostics/Command
platform:407	7	SSL certificate expiration warning	/Platform/Certificate/SSL/Expiration
platform:408	5	Appliance startup completed	/Appliance/State/Startup
platform:409	3	Configure login warning banner	/Platform/Configuration/LoginBanner
platform:410	3	Network settings modified	
platform:411	5	Automated password reset	/Platform/Authentication/PasswordChange
platform:412	3	Set locale	/Platform/Configuration/Locale

Platform Events, continued

Signature	Severity	Definition	Category
platform:440	3	SNMP configuration modified	Platform/Configuration/SNMP
platform:450	3	FTP service enabled	
platform:451	3	FTP service disabled	
platform:454	3	FTP service configuration changed	
platform:455	3	Added sub directory	
platform:456	3	Removed sub directory	
platform:460	3	NIC alias added	/Platform/Network/Alias/Add
platform:462	3	NIC alias removed	/Platform/Network/Alias/Remove
platform:500	5	Remove member from group	/Platform/Authorization/Groups/ Membership/Remove
platform:501	5	Group member added	/Platform/Authorization/Groups/ Membership/Add
platform:502	5	User removed from group	/Platform/Authorization/Users/ Groups/Remove
platform:503	5	User added to group	/Platform/Authorization/Users/ Groups/Add
platform:530	5	Authentication Session settings successfully changed	/Platform/Configuration/ Authentication/Sessions/Success
platform:540	5	Password Lockout settings successfully updated	/Platform/Configuration/ Authentication/Password/Lockout/ Success
platform:550	5	Password Expiration settings successfully updated	/Platform/Configuration/ Authentication/Password/Expiration/Success
platform:560	5	Password Validation settings successfully updated	/Platform/Configuration/ Authentication/Password/Validation/Success
platform:570	5	Allow Automated Password Reset settings successfully changed	/Platform/Configuration/ Authentication/Password/ AutomatedReset/Success

Platform Events, continued

Signature	Severity	Definition	Category
platform:590	5	RADIUS authentication settings successfully changed	/Platform/Configuration/Authentication/RADIUS/Success
platform:600	5	LDAP authentication settings successfully changed	/Platform/Configuration/Authentication/LDAP/Success
platform:610	5	Global authentication settings successfully changed	/Platform/Configuration/Authentication/Global/Success

System Health Events

System health events provide four status indicators:

- OK
- Degraded
- Rebuilding
- Failed

An **OK** event, indicating normal system behavior, is generated once every ten minutes (six events per hour, per sensor). For a status other than **OK** (**Degraded**, **Rebuilding**, or **Failed**), the event is sent every minute until the sensor returns an **OK** status.

SNMP Related Properties

The following list provides the event fields for system health events sent via SNMP traps. For detailed instructions on setting up SNMP traps, see "SNMP" on page 1077.

• event.deviceReceiptTime	• event.endTime
• event.deviceVendor	• event.deviceProduct
• event.deviceVersion	• event.deviceEventClassId
• event.name	• event.deviceSeverity
• event.deviceEventCategory	• event.deviceCustomNumber1
• event.deviceCustomNumber1Label	• event.deviceCustomString1
• event.deviceCustomString1Label	• event.deviceCustomString2
• event.deviceCustomString2Label	• event.deviceCustomString3
• event.deviceCustomString3Label	• event.deviceCustomString4

• event.deviceCustomString4Label	• event.deviceCustomString5
• event.deviceCustomString5Label	• event.deviceCustomString6
• event.deviceCustomString6Label	• event.destinationAddress
• event.deviceAddress	

The **snmp.mib.version** is set to 5.0.

System Health Events

Signature	Severity	Definition	Category
CPU			
cpu:100	1	CPU Usage	/Monitor/CPU/Usage
cpu:101	1	Health statistics per CPU	/Monitor/CPU*n*/Usage
Disk			
disk:101	1	Root Disk Space Remaining	/Monitor/Disk/Space/Remaining/Data
disk:102	1	Disk bytes read	/Monitor/Disk/*drive*/Read
disk:103	1	Disk bytes written	/Monitor/Disk/*drive*/Write
disk:104	1	Disk Space Remaining	/Monitor/Disk/Space/Remaining/Root
Hardware			
hardware:101	1	Electrical (Current) OK	/Monitor/Sensor/Current/Ok**
hardware:102	5	Electrical (Current) Degraded	/Monitor/Sensor/Current/Degraded**
hardware:103	8	Electrical (Current) Failed	/Monitor/Sensor/Current/Failed**
hardware:111	1	Electrical (Voltage) OK	/Monitor/Sensor/Voltage/Ok**
hardware:112	1	Electrical (Voltage) Degraded	/Monitor/Sensor/Voltage/Degraded**
hardware:113	1	Electrical (Voltage) Failed	/Monitor/Sensor/Voltage/Failed**
hardware:121	1	Battery OK	/Monitor/Sensor/Battery/Ok**
hardware:122	5	Battery Degraded	/Monitor/Sensor/Battery/Degraded **
hardware:123	8	Battery Failed	/Monitor/Sensor/Battery/Failed**
hardware:131	1	Fan OK	/Monitor/Sensor/Fan/Ok
hardware:132	5	Fan Degraded	/Monitor/Sensor/Fan/Degraded
hardware:133	8	Fan Failed	/Monitor/Sensor/Fan/Failed
hardware:141	1	Power Supply OK	/Monitor/Sensor/PowerSupply/Ok
hardware:142	5	Power Supply Degraded	/Monitor/Sensor/PowerSupply/ Degraded

System Health Events, continued

Signature	Severity	Definition	Category
hardware:143	8	Power Supply Failed	/Monitor/Sensor/PowerSupply/Failed
hardware:151	1	Temperature OK	/Monitor/Sensor/Temperature/Ok
hardware:152	1	Temperature Degraded	/Monitor/Sensor/Temperature/Degraded
hardware:153	1	Temperature Failed	/Monitor/Sensor/Temperature/Failed
Memory			
memory:100	1	Platform memory usage	/Monitor/Memory/Usage/Platform
memory:101	1	Health statistics for JVM memory	/Monitor/Memory/Usage/Jvm
memory:102	1	Health statistics for platform buffers memory	/Monitor/Memory/Usage/Platform/Buffers
memory:103	1	Health statistics for platform cached memory	/Monitor/Memory/Usage/Platform/Cached
memory:104	1	Health statistics for platform free memory	/Monitor/Memory/Usage/Platform/Free
memory:105	1	Health statistics for JVM heap memory	/Monitor/Memory/Usage/Jvm/Heap
memory:106	1	Health statistics for JVM non-heap memory	/Monitor/Memory/Usage/Jvm/NonHeap
Network			
network:100	1	Network usage—Inbound	/Monitor/Network/Usage/*iface*/In
network:101	1	Network usage—Outbound	/Monitor/Network/Usage/*iface*/Out
network:200	1	Number of Apache connections	
NTP			
ntp:100	1	NTP synchronization	
RAID			
raid:101	1	RAID Controller OK	/Monitor/RAID/Controller/OK
raid:102	5	RAID Controller Degraded	/Monitor/RAID/Controller/Degraded
raid:103	8	RAID Controller Failed	/Monitor/RAID/Controller/Failed
raid:111	1	RAID BBU OK	/Monitor/RAID/BBU/Ok
raid:112	5	RAID BBU Degraded	/Monitor/RAID/BBU/Degraded
raid:113	8	RAID BBU Failed	/Monitor/RAID/BBU/Failed

System Health Events, continued

Signature	Severity	Definition	Category
raid:121	1	RAID Disk OK	/Monitor/RAID/DISK/Ok
raid:122	5	RAID Disk Rebuilding	/Monitor/RAID/DISK/Rebuilding
raid:123	8	RAID Disk Failed	/Monitor/RAID/DISK/Failed

Managing the CDF Infrastructure

For information about CDF infrastructure, see " Understanding the CDF Infrastructure" on page 19

This section provides information about managing the CDF infrastructure.

Accessing the CDF Management Portal

The CDF management portal enables management, deployment, and configuration of CDF and CDF-based products.

To open the management portal for an on-premises installation:

1. Browse to `https://<ha-address>:5443`.
2. Enter the username *admin* and the password where *Ha-address:* FQDN corresponding to the Virtual IP address provided during installation (`--ha-virtual-ip`) (or, for a single-master installation, the IP address of the master node).

To open the management portal for an Azure-based cluster:

1. On the jump host, browse to `http://<private_DNS>:5443`.
2. Enter the username *admin* and the password.

To open the management portal for an AWS-based cluster:

1. On the bastion host, use either the forwarding display or forwarding local ports methods to browse to `http://<ALB DNS name>:5443`.
2. Enter the username *admin* and the password.

Managing CDF Management Portal Access

At times, you may be unable to log in to the CDF Management Portal with the admin user. When this situation occurs, you can unlock the user's account or reset the user's password.

- "Resetting the CDF Administrator Password" below
- "Unlocking the CDF Management Portal User Account" below
- "Resetting the User's Password" on the next page

Resetting the CDF Administrator Password

You can reset the administrator password on a CDF installation.

1. Browse to CDF Management Portal.
2. Log in using admin USERID and the password you specified during the platform installation in the command line argument. (This URL is displayed at the successful completion of the CDF installation shown earlier.)
3. In the left navigation page, click **IDM Administration.**
4. In the main panel, click **SRG**.
5. In the left navigation bar, click **Users**.
6. In the list of users on the right, select *Admin* and click **Edit**.
7. In the bottom right, click **Remove Password.**
8. Click **Add Password.**
9. Enter a new admin password, then click **Save**.

Unlocking the CDF Management Portal User Account

1. Log in to a master node as root.
2. To access the shell of the `idm` container, run the following command:

```
 kubectl exec -it $(kubectl get pod -n core -ocustom-
columns=NAME:.metadata.name |grep idm|head -1) -n core sh -c idm
```

3. To unlock the user, run the following command:

```
sh /idmtools/idm-installer-tools/idm.sh databaseUser unlockUser -org
Provider -name admin
```

Resetting the User's Password

1. Log in to a master node as root.
2. Run the following command to access the `idm` pod:

```
kubectl exec -it $(kubectl get pod -n core -ocustom-
columns=NAME:.metadata.name |grep idm|head -1) -n core sh idm
```

3. Run the following command to reset the password to a temporary value. (Replace `<new_tmp_password>` with your new temporary password.)

```
sh /idmtools/idm-installer-tools/idm.sh databaseUser resetPassword -org
Provider -name "admin" -plainPwd "<new_tmp_password>"
```

If the user account is locked due to too many failed login attempts, run unlock, as described above in "Unlocking the CDF Management Portal User account"

4. Log into the CDF Management Portal with the new temporary password, then set the new non-temporary password on the password reset page.
5. Log in to the CDF Management Portal with the new password.

Adding Additional Nodes to the Cluster

To scale out the cluster for increased events processing and analytics computing power, you can add master or worker nodes. You can add master nodes only if you configured high availability while deploying the Kubernetes cluster.

To add one or more nodes to a cluster:

1. Log in to the CDF Management Portal.
2. Click **Cluster** > **Nodes**.
3. Click **+ ADD**.
4. In the **Add Node** dialog, select **Master** or **Worker**, then specify the configuration information for the new node.
5. Click **ADD**.
6. Repeat Steps 2 and 3 to add more nodes as needed.
7. In **Predefined Labels**, specify a node label in the text box and click the **+** icon. Repeat this step to add more labels as needed.

For more information about labeling nodes, see Labeling the Nodes.

8. Drag and drop each of the labels you added to the corresponding nodes based on your workload sharing configuration. The corresponding components get deployed on the corresponding worker nodes.
9. Click **Refresh** to see the labels you applied to the nodes.
10. Click  and then click **Reconfigure**.
11. Depending on the capabilities for which you have assigned labels to the worker nodes, click the relevant tabs and reconfigure the properties.
12. Click **Save**.
13. Verify that all the pods are in the Running state:
    a. Launch a terminal session and log in to the master node as the root user.
    b. Execute the following command:

```
kubectl get pods --all-namespaces -o wide
```

Changing the IP Address of a Master or Worker Node

You can assign a master or worker node a new IP address by deleting the node, and then re-adding the node with the new FQDN.

To change the IP address of a master or worker node:

1. Note the FQDN, current IP address, and new IP address of the node for which you wish to assign a new IP address.
2. Log in to the CDF Management Portal (`https://<ha-address>:5443`).
3. Click **Cluster > Nodes.**
4. Click **+ Add.**
5. Next to the node for which you plan to change the IP address, in the **Operations** column, click **Delete**.
6. Enter the username and password/keyphrase to confirm node deletion. Remain on the page and verify that the node has been deleted. Give the process time to complete.
7. Outside of CDF, perform the necessary changes within your network administration or host settings to change the old IP address of the node to the new IP address.
8. In the CDF Management Portal, on **Cluster > Nodes.**
9. Click **+ Add.**

10. Enter the FQDN of the node which you are re-adding.
11. Enter values for the pop-up dialog as prompted. For host name, use the new (existing) FQDN of the node you deleted in Step 5.
12. Click **ADD**, and then wait for the confirmation that new node has been added to the cluster.
13. Ensure that you add the appropriate labels to the node that you re-added.

Checking Kubernetes Dashboard for Status and Errors

1. Log in to the CDF Management Portal.
2. Navigate to **Cluster** > **Dashboard** to access the Kubernetes Dashboard.
3. In Kubernetes Dashboard change Namespace to arcsight-installer-*.
4. Navigate to **Workloads** > **Pods**.
5. View the status of pods. For more information about each pod, see Understanding Labels and Pods.
6. Clicking a pod reveals more status details of that pod.
   a. Logs for the pod can be viewed by clicking on View Logs button in the right side of the blue banner near the top.
   b. Each pod may contain multiple containers, so when viewing logs, be sure to use the Logs from <container> to view the logs for the specific container you need to view.
   c. Logging levels can be modified as described at here.

Maintaining the RE Certificate

Use the information provided in the sections below to update or maintain the RE External Communication Certificate for the ArcSight Platform.

Securing External Communication with the RE Certificate

At the center of the Platform is a Kubernetes cluster where communication occurs between pods within the cluster and with non-containerized ArcSight components outside of the cluster. In order to ensure secure trusted communication between pods within the cluster and components outside of the cluster, encrypted communication with client certificate authentication is configured by default.

- Understanding the ArcSight Platform Certificate Authorities
- Using an RE External Communication Certificate Signed by Your Trusted Certificate Authority

Understanding the ArcSight Platform Certificate Authorities

During installation, three self-signed Certificate Authorities (CA) are created automatically, two for signing certificates used exclusively for pod to pod communication within the cluster (RIC and RID CA), and the other for signing certificates for each pod that performs communication external to the cluster (RE CA). Only pods that perform external communication have a certificate that is signed by the external CA.

External cluster communication occurs not only with ArcSight components, but also with user web browsers and, in some cases, user clients of ArcSight APIs (such as the REST API). By default, when the user connects to the cluster, they will be presented with a certificate that has been signed by the self-signed external CA. Since the external CA is self-signed, the user's connection will not automatically trust the certificate because it will not be verifiable using a certificate chain that is already in the user's trust store.

To give users confidence they are connecting to the trusted cluster, we recommend signing the certificates that are presented to the user with a CA that is trusted by the user's trust store. There are two approaches to doing this that are described in the documentation below. These approaches are:

Method 1 - Signing the RE External Communication Certificate with Your Trusted Certificate Authority

This is the recommended approach, because it is theoretically more secure than the other approach, in that, it only involves transferring a CSR and public certificate between systems, which does not put any private secrets at risk.

Method 2 - Importing an Externally Created Intermediate CA

This approach involves creating an Intermediate CA (key and certificate pair) in a system outside of the ArcSight Platform, and then importing it into the ArcSight Platform. While this approach does work, it is theoretically less secure than the other approach, because it involves transferring a CA private key between systems, which potentially exposes it to unintended parties.

Use only one of the two approaches above.

Using an RE External Communication Certificate Signed by Your Trusted Certificate Authority

Use only one of the two approaches below. The first one, "Signing the RE External Communication Certificate with Your Trusted Certificate Authority" approach is recommended for the reasons described in Understanding the ArcSight Platform Certificate Authorities.

Method 1 - Signing the RE External Communication Certificate with Your Trusted Certificate Authority

Signing the RE External Communication Certificate with Your Trusted Certificate Authority approach is recommended for the reasons described in Understanding the ArcSight Platform Certificate Authorities.

In order to sign the RE external communication certificate with your trusted CA, you need to (1) create a certificate signing request (CSR) from vault, (2) take it to your organization, (3) sign it, and (4) return the signed CSR and all the public chain-of-certificates used to sign it.

1. Export the following access token dependencies (you can remove these later if not needed):

```
export CDF_APISERVER=$(kubectl get pods -n core -o custom-
columns=":metadata.name"| grep cdf-apiserver)
```

```
export PASSPHRASE=$(kubectl get secret vault-passphrase -n core -o json
2>/dev/null | jq -r '.data.passphrase')
```

```
export ENCRYPTED_ROOT_TOKEN=$(kubectl get secret vault-credential -n core
-o json 2>/dev/null | jq -r '.data."root.token"')
```

```
export VAULT_TOKEN=$(echo ${ENCRYPTED_ROOT_TOKEN} | openssl aes-256-cbc -
md sha256 -a -d -pass pass:"${PASSPHRASE}")
```

2. Ask vault to generate the CSR by running the following command:

Important: When you execute this command, proceed expeditiously through steps 3 and 4, as your cluster will not be able to issue external certificates while it waits for the CSR to be signed.

```
kubectl exec -it -n core ${CDF_APISERVER} -c cdf-apiserver -- bash -c
"VAULT_TOKEN=$VAULT_TOKEN vault write -tls-skip-verify -format=json
RE/intermediate/generate/internal common_name=\"none-MF CDF RE CA on
<FQDN of ArcSight Platform Virtual IP for HA or single master node>\"
country=<Country> locality=<Locality> province=<Province>
organization=<Organization> ou=<Organizational Unit>" | jq -r '.data.csr'
> /tmp/pki_intermediate.csr
```

> **Note:** The `common_name` in the command above is an example common name. Substitue your own values for the common name to fit your environment. Additionally, your trusted certificate authority might require additional parameters in the CSR besides common_name. Ask your PKI team for what the required CSR parameters are and add the appropriate parameters to the command (similar to how the parameter common_name is specified). The parameter names for the vault command used above are documented at https://www.vaultproject.io/api-docs/secret/pki#generate-intermediate

3. Sign the CSR file with your trusted certificate authority, and save the result into the `intermediate.cert.pem` file.

   Examply only. A basic example is provided below. Your environment will likely be different.

   ```
   openssl ca -keyfile your-rootca-sha256.key -cert your-rootca-sha256.crt -
   config your-openssl-configuration-file -extensions v3_ca -notext -md
   sha256 -in /tmp/pki_intermediate.csr -out intermediate.cert.pem
   ```

   > Make sure the `v3_ca` extension is enabled and a new certificate is useable as a certificate authority on its own. Otherwise, you will receive a warning in the next step that given certificates are not marked for CA use.

4. Create an `intermediate.chain.pem` file that includes the combination of the `intermediate.cert.pem`, the public certificate of your trusted certificate authority, and all intermediate public certificates in the chain between them so that `intermediate.chain.pem` includes the full trust chain.

   ```
   cp intermediate.cert.pem intermediate.chain.pem
   cat [parent-intermediate1.crt] [parent-intermediate2.crt] [...] your-
   rootca-sha256.crt >> intermediate.chain.pem
   ```

   > If you have intermediate certificates between your `intermediate.cert.pem` and your trusted certificate authority, you must add the certificates in the specific order of the sequence of the chain, with the last certificate being the certificate of the root trusted CA.

5. Import the `intermediate.chain.pem` file into the cluster vault:

   ```
   chaincerts=$(cat intermediate.chain.pem) && kubectl exec -it -n core
   ${CDF_APISERVER} -c cdf-apiserver -- bash -c "VAULT_TOKEN=$VAULT_TOKEN
   vault write -tls-skip-verify -format=json RE/intermediate/set-signed
   certificate=\"${chaincerts}\""
   ```

6. Update `ConfigMap RE_ca.crt` by running these commands:

```
reCrtForJson=$(sed -E ':a;N;$!ba;s/\r{0,1}\n/\\n/g'
intermediate.chain.pem) && kubectl patch configmap -n core public-ca-
certificates -p "{\"op\": \"replace\", \"data\":{\"RE_
ca.crt\":\"${reCrtForJson}\"}}"
```

```
ARCSIGHT_NS=$(kubectl get namespaces --no-headers -o custom-
columns=":metadata.name" | grep arcsight-installer)
```

```
if [ -n "$ARCSIGHT_NS" ];then reCrtForJson=$(sed -E ':a;N;$!ba;s/\r
{0,1}\n/\\n/g' intermediate.chain.pem); kubectl patch configmap -n
$ARCSIGHT_NS public-ca-certificates -p "{\"op\": \"replace\", \"data\":
{\"RE_ca.crt\":\"${reCrtForJson}\"}}";fi
```

7. (Conditional) If you already deployed ArcSight Capabilities onto the CDF, update the ArcSight Capabilities to use the updated RE external communication certificate, by following the instructions in Configuring ArcSight Kubernetes Pods to Use the Updated RE External Communication Certificate.

   If you deployed CDF but have not yet deployed any ArcSight Capabilities, you can skip those instructions.

Method 2 - Importing an Externally Created Intermediate CA

This is an alternate approach for signing certificates to connect to the trusted cluster. Before choosing this approach, ensure that your understand the other approach recommended in Understanding the ArcSight Platform Certificate Authorities.

To import an externally created intermediate CA:

1. Obtain an intermediate CA (key and certificate pair) from your trusted certificate authority.
   a. Name the certificate files as follows:
      - key file: `intermediate.key.pem`
      - certificate file: `intermediate.cert.pem`
   b. Obtain the root CA certificate (including chain), and put it in a file named ca.cert.pem.
2. Replace the existing RE CA in the ArcSight Platform with the intermediate CA you obtained in the step above, based on your type of deployment, on-premises **or** cloud.

a. Change the directory:

- For an on-premises deployment, run these commands:

```
cd /opt/arcsight/kubernetes/scripts/
```

- For a cloud deployment, run these commands:

```
cd {path to cdf installer}/cdf-deployer/scripts/
```

b. Run the following command to replace the existing RE CA:

```
./cdf-updateRE.sh write --re-crt=/pathto/intermediate.cert.pem --re-
key=/pathto/intermediate.key.pem [--re-ca=/pathto/ca.cert.pem]
```

Note: `--re-ca=/pathto/ca.cert.pem` is the path to the file containing the certificate of CA used to sign re-crt. It is not required when re-crt is self-signed or CA is included in re-crt.

3. (Conditional) If you already deployed ArcSight Capabilities onto CDF, proceed to the next section to update the ArcSight Capabilities to use the updated RE external communication certificate, Configuring ArcSight Kubernetes Pods to Use the Updated RE External Communication Certificate.

   However, if you have only deployed CDF, but have not deployed ArcSight Capabilities yet, you can skip that section.

Configuring ArcSight Components to Use the Updated RE External Communication Certificate

After signing the RE External Communication Certificate for a new or upgraded ArcSight installation, you need to configure the Kubernetes pods and the ArcSight Database to use the updated certificate. The sections below provide the instructions to configure those components for the updated certificate. For information about signing the RE certificate, see Method 1 - Signing the RE External Communication Certificate with Your Trusted Certificate Authority

- Configuring ArcSight Kubernetes Pods to Use the Updated RE External Communication Certificate
- Configuring the ArcSight Database to Use the Updated RE External Communication Certificate

Configuring ArcSight Kubernetes Pods to Use the Updated RE External Communication Certificate

Following this procedure will restart ArcSight Kubernetes pods so that they immediately refresh their trust stores to use an updated RE external communication certificate chain. This will result in temporary downtime of the services these pods provide while the pods are restarting.

1. Restart the following ArcSight pods, so they can use the new RE certificate:
   a. Commands for all deployments:

```
ARCSIGHT_NS=$(kubectl get namespaces --no-headers -o custom-
columns=":metadata.name" | grep arcsight-installer)
labels=autopass-lm-apps,soar-web-app,soar-
frontend,osp,management,reporting,search-engine,arcmc,web-
service,schema-registry,kafka,zookeeper,c2av-stream-
processor,enrichment-processor,kafka-manager,c2av-
esmprocessor,routing-processor,ceb,searchmanager-engine,interset-
api,interset-analytics,intelligence-tuning-api,hdfs-namenode,hdfs-
datanode,interset-logstash,arcsightconnector-api
```

```
kubectl delete pods -n $ARCSIGHT_NS -l "name in (suite-reconf-sel-
arcsight-installer)"
```

```
kubectl delete pods -n $ARCSIGHT_NS -l "app in ($labels)"
```

> Some pods may not return to running state after restart due to the certificate change while trying to connect to database. For example, `intelligence-tuning-api`, `interset-analytics`, and `interset-api`. These pods should return to running state once you update database certificates.

   b. (Conditional) Command for an on-premises deployment only, after running the commands above:

```
kubectl delete pod -n core -l "app.kubernetes.io/name in (itom-kube-
dashboard)"
```

2. Update the ArcSight Platform's embedded reverse proxy Nginx to use the updated RE external communication certificate chain, based on your type of deployment, on-premises **or** cloud by running the commands below:

a. Change the directory:

- For an on-premises deployment, run:

```
cd /opt/arcsight/kubernetes/scripts/
```

- For a cloud deployment, run:

```
cd {path to cdf installer}/cdf-deployer/scripts/
```

b. Run the `cdf-updateRE` script:

```
./cdf-updateRE.sh renewPortals
```

3. (Conditional) If you deployed the ArcSight Database, update the ArcSight Database to use the updated RE external communication certificate by following the instructions in Configuring the ArcSight Database to Use the Updated RE External Communication Certificate.

Create or Update Route 53 Certificates (AWS Only)

A user-provided self-signed or CA-signed certificate is required for creating the Application Load Balancer (ALB). In this section, you will create a configuration file for the certificate signing request (CSR), create an intermediate certificate pair, sign the CSR, create a chained file for import, then import the self-signed certificate into Amazon Certificate Manager (ACM).

1. Create or update Route 53 certificates:

a. Run the applicable command on a secure machine to generate the Route 53 certificate:

- For a current version of SSL, run this command:

```
openssl req -nodes -newkey rsa:2048 -keyout
<your.route53dnsRecordsetName>.key.pem -out
<your.route53dnsRecordsetName>.csr.pem -subj
"/C=US/ST=State/L=City/O=Company
Inc./OU=IT/CN=<your.route53dnsRecordsetName>" -addext
"subjectAltName = DNS:<your.route53dnsRecordsetName>"
```

- If your operating system does not support `-addext` for SSL, run this command:

```
openssl req -newkey rsa:2048 -sha256 -nodes -keyout
your.route53dnsRecodsetName.key.pem -out
your.route53dnsRecodsetName.csr.pem -subj
"/C=US/ST=CA/L=SU/O=MF/OU=IT/CN=<your.route53dnsRecordsetName>" -
extensions san -config <(echo '[req]'; echo 'distinguished_
name=req';echo 'req_extensions=san';echo '[san]'; echo
'subjectAltName=DNS:your.route53dnsRecodsetName')
```

`your.route53dnsRecordsetName` is your route53 record set name tracked in your AWS configuration worksheet. This command will create the private key file `<your.route53dnsRecordsetName>.key.pem` and the certificate signing request file `<your.route53dnsRecordsetName>.csr.pem`.

b. Copy the certificate signing request `<your.route53dnsRecordsetName>.csr.pem` to your bastion or jump host machine.

c. Run the following commands to sign the certificate signing request using your cluster RE certificate:

```
export CDF_APISERVER=$(kubectl get pods -n core -o custom-
columns=":metadata.name"| grep cdf-apiserver)
```

```
export PASSPHRASE=$(kubectl get secret vault-passphrase -n core -o
json 2>/dev/null | jq -r '.data.passphrase')
```

```
export ENCRYPTED_ROOT_TOKEN=$(kubectl get secret vault-credential -n
core -o json 2>/dev/null | jq -r '.data."root.token"')
```

```
export VAULT_TOKEN=$(echo ${ENCRYPTED_ROOT_TOKEN} | openssl aes-256-
cbc -md sha256 -a -d -pass pass:"${PASSPHRASE}")
```

```
export COMMON_NAME=<your.route53dnsRecordsetName>
```

```
export CSR=$(cat ${COMMON_NAME}.csr.pem)
```

```
export WRITE_RESPONSE=$(kubectl exec -it -n core ${CDF_APISERVER} -c
cdf-apiserver -- bash -c "VAULT_TOKEN=$VAULT_TOKEN vault write -tls-
skip-verify -format=json RE/sign/coretech csr=\"${CSR}\"")
```

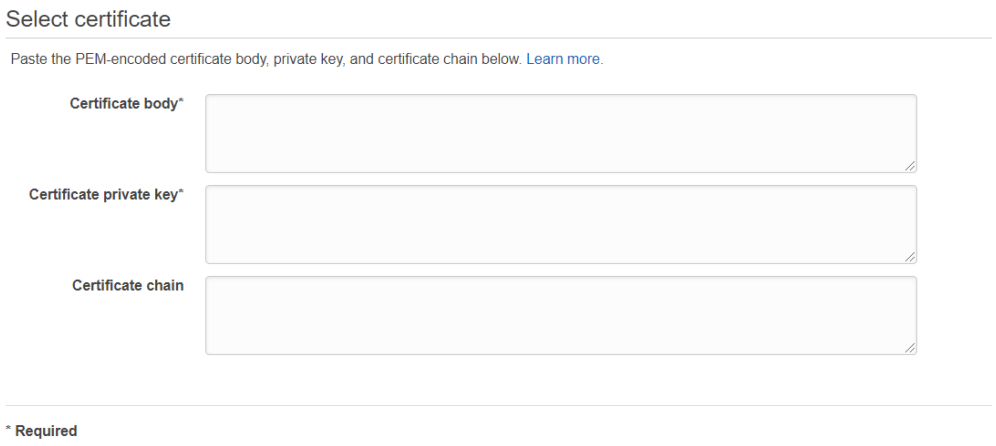
```
echo ${WRITE_RESPONSE} | jq -r ".data | .certificate" > ${COMMON_
NAME}.signed.crt.pem
```

```
echo ${WRITE_RESPONSE} | jq -r ".data | if .ca_chain then .ca_chain[]
else .issuing_ca end" > ${COMMON_NAME}.ca_chain.pem
```

The RE signed certificate is in file `${COMMON_NAME}.signed.crt.pem`. The certificate chain is in file `${COMMON_NAME}.ca_chain.pem`.

2. Import or update the certificate in Amazon Certificate Manager (ACM).

a. Import the self-signed certificate into ACM (for a fresh installation):
   i. Log in to the AWS Console.
   ii. Browse to the Amazon Certificate Manager (ACM).
   iii. Click **Import a certificate**, and then complete the fields as follows:
      - **Certificate body:** Specify the contents of the signed certificate you created earlier. For example, `<your.route53dnsRecordsetName>.crt.pem`
      - **Certificate private key:** Specify the contents of the private key created by the CSR request creation. For example, `<your.route53dnsRecordsetName>.key.pem`
      - **Certificate chain:** Specify the contents of the chain file. For example, `<your.route53dnsRecordsetName>.ca.pem`



b. To update the certificate in ACM (for a current running installation):
   i. Log in to the AWS Console.
   ii. Browse to the Amazon Certificate Manager (ACM).
   iii. Search for your certificate Domain name or ID.
   iv. Select your Certificate ID, and click **Reimport**.
   v. Complete the fields as follows:
      - **Certificate body:** Specify the contents of the new signed certificate you created earlier. For example, `<your.route53dnsRecordsetName>.crt.pem`
      - **Certificate private key:** Specify the new contents of the private key created by the CSR request creation. For example, `<your.route53dnsRecordsetName>.key.pem`
      - **Certificate chain:** Specify the new contents of the chain file. For example, `<your.route53dnsRecordsetName>.ca.pem`

For more details, see https://docs.aws.amazon.com/acm/latest/userguide/import-reimport.html

3. Click **Next**. Optionally, add any tags you wish to the import.
4. Click **Next**, and then, click **Import**.

   After the import, click the arrow next to the certificate ARN value. Note the value to your AWS worksheet for later use. For example:

Configuring the ArcSight Database to Use the Updated RE External Communication Certificate

If you deployed the ArcSight Database with your platform, you need to follow these instructions to update the RE External communication certificate.

1. Run these commands on your database node1 to generate the Kafka Scheduler private key file `kafkascheduler.key.pem` and the certificate signing request file `kafkascheduler.csr.pem`:

```
cd <yourOwnCertPath>/
```

   If you installed using the ArcSight Platform Installer, the default location is `/opt/arcsight-db-tools/cert/`

```
rm -fr kafkascheduler.*.pem issue_ca* *.0
```

```
openssl req -nodes -newkey rsa:2048 -keyout kafkascheduler.key.pem -out kafkascheduler.csr.pem -subj "/C=US/ST=State/L=City/O=Company Inc./OU=IT/CN=kafkascheduler"
```

2. Copy the certificate signing request `kafkascheduler.csr.pem` to your cluster or your bastion or jump host.
3. Run the following commands on your cluster or your bastion host to sign the certificate signing request using your cluster RE certificate:

```
export CDF_APISERVER=$(kubectl get pods -n core -o custom-
columns=":metadata.name"| grep cdf-apiserver)
```

```
export PASSPHRASE=$(kubectl get secret vault-passphrase -n core -o json
2>/dev/null | jq -r '.data.passphrase')
```

```
export ENCRYPTED_ROOT_TOKEN=$(kubectl get secret vault-credential -n core
-o json 2>/dev/null | jq -r '.data."root.token"')
```

```
export VAULT_TOKEN=$(echo ${ENCRYPTED_ROOT_TOKEN} | openssl aes-256-cbc -
md sha256 -a -d -pass pass:"${PASSPHRASE}")
```

```
export COMMON_NAME=kafkascheduler
```

```
export CSR=$(cat ${COMMON_NAME}.csr.pem)
```

```
WRITE_RESPONSE=$(kubectl exec -it -n core ${CDF_APISERVER} -c cdf-
apiserver -- bash -c "VAULT_TOKEN=$VAULT_TOKEN vault write -tls-skip-
verify -format=json RE/sign/coretech csr=\"${CSR}\"") && \
```

```
echo "${WRITE_RESPONSE}" | jq -r ".data | .certificate" > ${COMMON_
NAME}.crt.pem && \
```

```
echo "${WRITE_RESPONSE}" | jq -r ".data | if .ca_chain then .ca_chain[]
else .issuing_ca end" > issue_ca.crt
```

4. Copy the RE signed certificate file `kafkascheduler.crt.pem` and certificate chain file `issue_ca.crt` to database node1 <*yourOwnCertPath*>.
5. Update the Database SSL Configuration.
   a. If you have not already done so, move the following files to database node1 <*yourOwnCertPath*> as root:

```
cd <yourOwnCertPath>/
ls <yourOwnCertPath>/
```

   The output should have the following files:

   - `generated-db-ca.crt`
   - `generated-db-server.crt`

- generated-db-server.key
- generated-db-ca.key
- generated-db-ca.srl
- generated-db-server.csr
- issue_ca.crt
- kafkascheduler.crt.pem
- kafkascheduler.key.pem

If you have not set up the database for SSL mode before and you want to enable SSL now, you may need to generate generated-db-*.*. To do so, continue with substeps 5. b-c. Otherwise, skip to Step 6.

b. For chained CAs, run these commands to split the CAs into individual files:

```
cat issue_ca.crt | awk 'BEGIN {c=0;} /BEGIN CERT/{c++} { print >
"issue_ca_part." c ".crt"}'
```

```
chown -R dbadmin:dbadmin <yourOwnCertPath>
```

c. (Conditional) If your database is SSL enabled, run the following commands on database node1 to update the database SSL configuration:

```
cd /opt/arcsight-db-tools
```

```
./db_ssl_setup --disable-ssl
```

NOTE: If the attempt fails, drop the certificate manually by running these commands:

```
sudo su - dbadmin
```

```
vsql -U <dbadminuser> -w <dbadminpassword> -c "ALTER TLS CONFIGURATION
server CERTIFICATE NULL;"
```

```
vsql -U <dbadminuser> -w <dbadminpassword> -c "DROP CERTIFICATE IF
EXISTS server CASCADE;"
```

i. Enable database SSL for a single issue CA or chained issue CAs:

- For a single issue CA, run this command:

```
./db_ssl_setup --enable-ssl --vertica-cert-path
<yourOwnCertPath>/generated-db-server.crt --vertica-key-path
<yourOwnCertPath>/generated-db-server.key --vertica-ca-path
```

```
<yourOwnCertPath>/generated-db-ca.crt --client-ca-path
<yourOwnCertPath>/issue_ca.crt
```

-or-

- For chained issue CAs, run this command, specifying each CA certificate in the chain one by one, separated by a comma in the `client-ca-path` parameter:

```
./db_ssl_setup --enable-ssl --vertica-cert-path
<yourOwnCertPath>/generated-db-server.crt --vertica-key-path
<yourOwnCertPath>/generated-db-server.key --vertica-ca-path
<yourOwnCertPath>/generated-db-ca.crt --client-ca-path
<yourOwnCertPath>/issue_ca_part.1.crt,
<yourOwnCertPath>/issue_ca_part.2.crt[,...]
```

6. Update the Kafka Scheduler configuration.
   a. On database node1, stop the Kafka Scheduler:

```
cd /opt/arcsight-db-tools/
./kafka_scheduler stop
```

   b. Run the following command on database node1 to configure the schema registry server setting:

```
./schema_registry_setup <FQDN of ArcSight Platform Virtual IP for HA
or single master node> <yourOwnCertPath>/issue_ca.crt
<yourOwnCertPath>/kafkascheduler.crt.pem
<yourOwnCertPath>/kafkascheduler.key.pem
```

   c. (Conditional) If the Kafka Scheduler and database are both SSL enabled, update the Kafka Scheduler SSL setup:
      i. On database node1, delete the Kafka Scheduler:

```
cd /opt/arcsight-db-tools/
./kafka_scheduler delete
```

      ii. On all database nodes, remove the existing Kafka Scheduler SSL configuration.

```
rm -fr /opt/arcsight-db-tools/ssl_default /opt/arcsight-db-
tools/wrk
```

      iii. On database node1, configure the SSL setting for the Kafka Scheduler.

      This method uses the crt and key files gathered or generated in earlier steps. The `issue_ca.crt` file should contain all chained CAs. For the Kafka Scheduler to use SSL, run the following command:

```
./sched_ssl_setup --enable-ssl --sched-cert-path
<yourOwnCertPath>/kafkascheduler.crt.pem --sched-key-path
<yourOwnCertPath>/kafkascheduler.key.pem --vertica-ca-key
<yourOwnCertPath>/generated-db-ca.key --vertica-ca-path
<yourOwnCertPath>/generated-db-ca.crt --kafka-ca-path
<yourOwnCertPath>/issue_ca.crt
```

iv. Run the following command on database node1 to create the Kafka Scheduler:

```
./kafka_scheduler create <aks_nodename1>:9093,<aks_
nodename2>:9093,<aks_nodename3>:9093
```

7. Start the Kafka Scheduler and checker on database node1:

```
./kafka_scheduler start
./kafka_scheduler messages
./kafka_scheduler events
```

Maintaining Certificates

Certificates and their Certificate Authority (CA) have an expiration date; therefore, they need to be renewed prior to expiring in order for the cluster to operate properly.

To better understand the CAs in the cluster, see Securing External Communication with the RE Certificate.

In this section, `${K8S_HOME}` refers to:
On-premises: `/opt/arcsight/kubernetes`
Cloud: `${cdf-deployer path}`

Viewing the CA Validity Dates

Review the content below for information about viewing CA validity dates.

- Internal CA (RIC and RID CA) is reported in the beginning of each `kube-status` run with time/date and days remaining till expiration.
- To view the external CA (RE CA) validity dates, execute the following command on the primary master node. Or, if deployed to the Cloud, execute the command on the jump host.

```
${K8S_HOME}/scripts/cdf-updateRE.sh read | openssl x509 -noout -issuer -
subject -dates
```

Renewing Internal CAs

This information for is for pod communication within the cluster and not for certificates used for external pod communication.

To check if your Internal Certificate Authority is close to expiration, login into CDF Management Portal, which will show a warning if less than 30 days are left till expiration.

Alternatively, you can run the `kube-status.sh` script from `/opt/arcsight/kubernetes/bin` (installation path by default). Expiration date will be reported as the first line in the script output.

To renew internal CAs and dependent certificates:

1. Execute `renewCert`. This action also distributes the renewed CA between the nodes.

   ```
   ${K8S_HOME}/scripts/renewCert --renew -t internal -V 730
   ```

2. Follow the on-screen prompts to:
   a. Generate new certificates.
   b. Distribute them between the nodes using `scp`.
   c. Apply certificates by restarting nodes one by one.

Renewing External CAs

This procedure updates the certificates used by the CDF Management Portal as well as ArcSight capabilities. Changing the certificate by way of the CDF Management Portal, Administration > Certificate, only changes the certificate used by the CDF Management Portal.

To renew external CAs, request that your PKI team generates an intermediate certificate and matching key. Be sure to obtain any higher root certificate authority or a whole chain if more that one level used.

If you cannot get a key from your PKI team, see Using an RE External Communication Certificate Signed by Your Trusted Certificate Authority.

1. Execute `cdf-updateRE.sh`.

```
${K8S_HOME}/scripts/cdf-updateRE.sh write --re-key={New Intermediate Key Name}.pem --re-crt={New Intermediate Certificate Name}.crt
```

If your intermediate certificate is signed by a higher root certificate authority, provide a chain of root CA certificate and intermediate certificate concatenated in one file (keeping the headers) to the "re-crt" parameter. Make sure the intermediate certificate is first in the file, and the root CA certificate is last in the file.

2. Pods of the deployed ArcSight capabilities that perform external communication continue to use the certificates generated by the platform on the pod start up until the pod is restarted.

To understand the pods that perform external communication, see "Understanding Labels and Pods" on page 578.

Renewing External Certificate of Management Portal and Fusion Single-Sign-On Portal

To renew the certificate for portals:

1. Log in to database node1.
2. Follow these steps to stop the Kafka Scheduler and Watchdog:
   a. Change to the database tools directory:

```
cd /opt/arcsight-db-tools
```

   b. Stop the Kafka Scheduler:

```
./kafka_scheduler stop
```

   c. Disable watchdog:

```
script/watchdog.sh disable
```

3. Restart the ArcSight pods:

```
kubectl delete pods --all -n $(kubectl get namespaces --no-headers -o custom-columns=":metadata.name" | grep arcsight-installer)
```

4. Run this command to update the `nginx` certificate:

```
${K8S_HOME}/scripts/cdf-updateRE.sh renewPortals
```

The second command generates the nginx certificate and updates the nginx-invesitgate-secret and nginx-default-secret.

5. Continue to the next step to update the database certificates: Configuring the ArcSight Database to Use the Updated RE External Communication Certificate.

The CDF Doctor Utility

The CDF Doctor utility, supported for both on-premises and BYOK CDF installations, can be used to diagnose and repair CDF issues. You can request a newer version of CDF Doctor through Microfocus Support channels to have enhanced diagnostic and logging capabilities.

CDF Doctor is located at {K8S_HOME}/tools/cdf-doctor.

Running CDF Doctor

For maximum visibility into issues, run CDF Doctor on each problematic node.

To run the CDF Doctor on a problematic node:

1. Enter the following commands:
   ```
   cd $K8S_HOME/tools/cdf-doctor/
   ./cdf-doctor cluster check
   ```
2. When prompted for login credentials:
   a. For username use admin
   b. For password, use your password for the CDF management portal (that is, at https://<your high availability FQDN>:5443)

You can run CDF Doctor on a failed master node by adding the --master parameter to the cluster check run command.

Types of Diagnostic Checks

When run, CDF Doctor will perform an array of diagnostic checks by default. Some checks permit CDF Doctor to repair an issue as soon as it is detected. Default diagnostic checks run by CDF Doctor will check the following components.

- CDF components
- Native components, such docker and kubelet (On-Premises CDF only)

- Kube-system (`etcd`)

Default diagnostic checks are compatible with all CDF versions 2020.08 and later.

Component	Checks...
Docker	Docker status
`kubelet`	• `kubelet` status • whether policy is loaded when SELinux is enforcing • whether `kubelet` runtime data directory is missing • whether `kubelet` certificate files's permission is incorrect • whether `kubelet` certificate is expired • whether swap is off • whether swap is enabled
`Etcd`	`etcd` status
`cdf-apiserver`	`cdf-apiserver` status
`dashboard`	`dashboard` status
`db-backup`	`db-backup` status
`idm`	`idm` status
`mng-portal`	`mng-portal` status
`nginx-ingress`	`nginx-ingress-controller` status
`node`	cluster node status
`pv`	persistent volume status
`registry`	registry status
`suite-config`	FQDN in `suite-conf-cm-` FQDN in `suite-conf-ing-` FQDN in `suite-conf-pod-`
`suite-frontend-ingress`	`suite-frontend-ingress` status

Component	Checks...
`suite-frontend-ui`	`suite-frontend-ui` status
`suite-update`	• FQDN in `suite-upgrade-cm-` • FQDN in `suite-upgrade-ing-`
Vault	• Vault component status • whether node NTP service is enable and synced • cluster nodes time difference • suite metadata folder permission • whether vault policy incorrect (automated fix) • whether pullsecret exists • whether can login to registry • whether registry contains jdbc image • FQDN in `nginx-ingress-controller` deployment • FQDN in idm deployment and ingress • whether suite parameter file is missing • PV info in suite parameter file (fix provided after user confirmation) • FQDN in ingress • FQDN in mng-portal deployment and ingress • FQDN in frontend-ingress deployment • check FQDN in suite-installer-frontend deployment • FQDN in itom-k8s-dashboard deployment and ingress • FQDN in itom-pg-backup-config configmap • FQDN in itom-ingress-pg-backup ingress

Dump File

The dump file provides a quick way to gather information about nodes where CDF is deployed. The file can be used quickly gather and encrypt information to provide for support investigation of issues.

To generate a dump file with check results, run CDF Doctor with the `--encrypt-password` parameter. You can also provide a username and password to get additional dump data from the CDF Management Portal.

The dump file contains the following information:

File Section	Description
OS commands output	Refer to `$K8S_HOME/tools/support-tool/conf/supportdump.config`
Directory content	Refer to `$K8S_HOME/tools/support-tool/conf/supportdump.config`
Files content	Refer to `$K8S_HOME/tools/support-tool/conf/supportdump.config`
Kube-Info	• Docker version and installation status • `kubelet` version and Installation status • Current node infomation Current node information: • Docker containers on current node • Docker images on current node's docker runtime Cluster info: • namespace, pv, pvc, nodes, deployment, service,pod,ingress Pod container information: • pod name • pod namespace • node pod is running on • images pod uses Suite info: • manage portal accessibility • selected features Deployment information • Docker journal logs • Docker images details collected from Docker inspection • cluster dump info collected from `kubectl` cluster dump • pod description • suite-db data • suite metadata

Managing the Database

This section provides information about managing the database.

Monitoring the Database

You can monitor the Database by using commands, or the out-of-the-box Health and Performance Monitoring dashboard included in the component.

Understanding Database Watchdog

Database includes a watchdog, which is configured as a cron job to automatically run once an hour to monitor the database and perform the following operations:

- When it detects a database cluster node is in down state, it will try to restart the node.
- Create the database event ingestion process (Kafka Scheduler) if it is missing.
- Start the database event ingestion process (Kafka Scheduler) if it is stopped.
- Unless there is a policy in place, do not use watchdog to delete reject events.

Monitoring Database Status

Monitor the database status by using the following command:

```
/opt/arcsight-db-tools/db_installer status
```

Monitoring Scheduler Status, Events, and Messages

Monitor the scheduler's status by using the following command:

```
/opt/arcsight-db-tools/kafka_scheduler status
```

Monitor scheduler events by using the following command:

```
/opt/arcsight-db-tools/kafka_scheduler events
```

Monitor scheduler messages by using the following command:

```
/opt/arcsight-db-tools/kafka_scheduler messages
```

Using the Health and Performance Monitoring Dashboard

You can also monitor the status of the database by using the out-of-the-box Health and Performance Monitoring dashboard included in the component. The dashboard includes the following widgets.

Database Event Ingestion Timeline

The Database Event Ingestion Timeline widget represents the rate of event ingestion into the database. This widget measures when the database receives the event data.

As a SOC Manager or an IT Administrator you want to monitor the event ingestion rate into the database. Due to differences in how quickly an event from different sources arrive at the database for storage, the moment when a database stores an event differs from when the event occurred. In this widget, you can monitor when the database receives the event data.

In the Database Event Ingestion widget, you can set the Upper and Medial Threshold values. Yellow represents the EPS values occurring in between the Medial and Upper Thresholds, and red represents the values occurring above the Upper Threshold. Green represents the EPS values occurring below the Medial Threshold.

Removing Rejected Events

For default tenants, use this procedure to ensure there are no rejected events. If `/opt/vertica/data/fusiondb/v_fusiondb_node000*_data/RejectionTableData` is not empty, then reject event exists and you need to take action immediately.

A high volume of rejected events impacts the query performance and occupies disk space, which retention policy cannot reduce. The rejected events will continue to occur until the root cause is resolved.

If you are using watchdog to delete rejected events, be sure a policy is in place, such as *if the reject events utilization is > 1% of the storage then delete the reject events*.

1. Analyze the content of reject_event_file to determine the root cause or save the reject_event_file for further analysis. Without resolving the root cause, the reject event creation will continue.
2. Delete reject events after completed analysis by using the following command:

```
rm -rf /opt/vertica/data/fusiondb/v_fusiondb_node000*_data/RejectionTableData*
```

Understanding the Database Installer Options

To specify an option:

Type `./db_installer <Option_Name>`.

Option Name	Description
install	Installs the database
uninstall	Uninstalls the database and deletes data and users
create-schema	Creates the database schema for Recon/Intelligence
delete-schema	Deletes the Recon/Intelligence database schema
start-db	Starts the database with the `dba_password` specified in `db_credentials.properties`
stop-db	Stops the database
status	Prints the database cluster status

Configuring the Policy for Retaining Data

ArcSight Platform allows you to configure a policy for retaining raw events in the database. The policy for retaining data uses a script, which runs in the primary database node.

Consider the following example to understand how the policy for retaining data works:

If you run the data retention script on 6/30/2019 and the data retention period is set as 90 days, then data older than 04/01/2019 will be deleted.

Configuring Data Retention with Recon and Intelligence Deployed

- "Configuring Analytics Data Retention" below
- "Configuring Event Data Retention" on the next page

Configuring Analytics Data Retention

In Intelligence, you can set a desired event retention period, which should range from 1 to 365 days.

To set the event retention period:

1. In the CDF Management Portal, select **Deployment** > **Deployments**.
2. Click ... (Browse) on the far right and choose **Reconfigure**. A new screen will be opened in a separate tab.
3. Click **Intelligence**.
4. Under **Analytics Configuration**, specify the value for **Analytics Data Retention Period**.

   Note: The default value for **Analytics Data Retention Period** is 90 days.
5. Click **Save**.

Configuring Event Data Retention

For information about using storage groups to organize and retain data, follow the Use Storage Groups to Organize and Retain Data section in the *ArcSight Recon User's Guide*.

Configuring Event Retention with Only Intelligence Deployed

- "Configuring Analytics Data Retention" below
- "Configuring Event Data Retention" below

Configuring Analytics Data Retention

In Intelligence, you can set a desired event retention period, which should range from 1 to 365 days.

To set the event retention period:

1. In the CDF Management Portal, select **Deployment** > **Deployments**.
2. Click ... (Browse) on the far right and choose **Reconfigure**. A new screen will be opened in a separate tab.
3. Click **Intelligence**.
4. Under **Analytics Configuration**, specify the value for **Analytics Data Retention Period**.

   Note: The default value for **Analytics Data Retention Period** is 90 days.
5. Click **Save**.

Configuring Event Data Retention

To enable event retention when only Intelligence is deployed:

1. Login to the primary database node.
2. Use a `vsql` or `db sql` tool of your choice and run the following query:

```
update default_secops_adm.storage_groups set deleteAfter = <number_of_
month_to_keep>;  commit;
```

   Note: In the above query, replace `<number_of_month_to_keep>` by a desired value, which should be greater than 0 or -1.
   The default value for `<number_of_month_to_keep>` is -1, which stores the events forever.

Rebooting Database Cluster

1. Log in to Database node 1.

```
cd /opt/arcsight-db-tools
```

2. Run the following command:

```
./db_installer stop-db
```

3. Reboot all cluster nodes.
4. Log in to Database node 1.

```
cd /opt/arcsight-db-tools
```

5. Run the following command:

```
./db_installer start-db
```

Enabling FIPS Mode on the Database Server

To enable the FIPS mode for the database, you must configure the operating system to FIPS mode.

Note: Some of the following steps are performed in a single node (ArcSight **Database** node1), while some are performed in all nodes. For clarity, when not specified, the step must be executed only on ArcSight **Database** node1.

1. Log in to the database node1.
2. Navigate to the `/opt/arcsight-db-tools` directory.
3. Stop data ingestion:

```
./kafka_scheduler stop
```

4. Stop the database:

```
./db_installer stop-db
```

5. Run the following commands (**to be performed in all nodes**):

```
fips-mode-setup --enable
```

```
reboot
```

6. To verify whether FIPS mode is enabled on the server, run the following command (**to be performed in all nodes**):

   ```
   /usr/bin/fips-mode-setup --check
   ```

   In case the above command fails, you can verify with these alternative commands:

   ```
   sysctl crypto.fips_enabled
   ```

   ```
   cat /proc/sys/crypto/fips_enabled
   ```

7. Run the following command (**to be performed in all nodes**):

   ```
   rm /opt/vertica/lib/libcrypto.* /opt/vertica/lib/libssl.*
   ```

   This command in the database nodes will generate a verification prompt for the deletion of each folder. Enter a "y" to agree, and then hit enter.

8. Verify that Zulu8 RPM package is installed using the following command. The command will return an empty message if Zulu8 is not installed, but will return the filename of the Zulu8 package if installed. If Zulu8 is already installed, skip to Step 11.

   ```
   rpm -qa | grep zulu
   ```

9. (Conditional) If Step 8 returns an empty message, then determine the latest Zulu8 package on the Azul web site (www.azul.com) for Java8 and your OS. Download the file using this command:

   ```
   wget <Zulu8 download URL>
   ```

10. (Conditional) Install the Zulu8 file that you downloaded by running the following command on each node:

    ```
    rpm --nodigest --nofiledigest -i <Zulu8 filename>
    ```

11. Restart the database:

    ```
    ./db_installer start-db
    ```

12. Restart data ingestion:

    ```
    ./kafka_scheduler start
    ```

Disabling FIPS Mode

To disable FIPS mode, run the following commands on each database node.

```
fips-mode-setup --disable
reboot
```

Configuring the Database for MinIO Storage (Examples Only)

Note: The information on this page is provided for illustrative purpose only, is not verified for production use, and no support is provided by Micro Focus for this information. Please utilize MinIO official support resources to assist you with deploying and configuring MinIO.

The setup examples below incorporated MinIO version Release.2021-10-13T00-23-17Z.

The procedures in this section for setting up the MinIO storage solution using MinIO Gateway in NFS mode are for example purposes only. Additionally, the procedures are not necessarily contiguous and provide more than one way of configuring MinIO. For example, the type of certificate used or setting up with or without TLS.

As you proceed with your own MinIO setup, ensure that you thoroughly run the processes that you configure in a test environment before deploying to your operational system. For more information, see ""Understanding Object Storage Options for the ArcSight Database" on page 35."

Server addresses and the like are from our test environment and should not be taken literally.

Setting Up the NFS Server

Example for configuring the NFS Server for MinIO:

1. Log in to the NFS Server and run the following commands:

```
systemctl restart rpcbind
systemctl enable rpcbind
systemctl restart nfs-server
systemctl enable nfs-server
```

2. Create the MinIO NFS directory:

```
mkdir /opt/minio-nfs
chown -R 1999:1999 /opt/minio-nfs
```

```
echo "/opt/minio-nfs 10.000.0.0/16(rw,sync,anonuid=1999,anongid=1999,all_
squash)" > /etc/exports
```

3. Restart the NFS Server:

```
systemctl restart nfs-server
showmount -e
```

Setting Up MinIO - Non-TLS

Example for setting up MinIO without TLS on the NFS server.

1. Log in to the MinIO Server and create the `start-minio.sh` script:

```
start-minio.sh
```

```
#!/bin/bash
NFS_SERVER=<NFS server-IP/FQDN>
NFS_DIR=/opt/minio-nfs
MINIO_DIR=/opt/minio-nfs-data
DATA_DIR=/opt/minio-nfs-data/minio/data

mkdir -p $MINIO_DIR
mount $NFS_SERVER:$NFS_DIR $MINIO_DIR
mkdir -p $DATA_DIR

export MINIO_ROOT_USER=access_key
export MINIO_ROOT_PASSWORD=change_me

./minio gateway nas $DATA_DIR --console-address :42497 &
```

Port 42497 is an example. You need to configure your own unused port. You also need to get your own Minio binary code.

2. Run the `start.minio.sh` script:

```
./start.minio.sh
```

Output example:

```
Output:
echo ======================================
echo "$DATA_DIR has been created"
echo "Data will be placed in $DATA_DIR"
echo "Check output for more information"
echo ======================================
```

```
API: http://10.000.000.000:9000 http://192.168.122.1:9000
http://127.0.0.1:9000
RootUser: access_key
RootPass: change_me
```

```
Console: http://10.000.000.001:42497 http://192.168.122.1:42497
http://127.0.0.1:42497
RootUser: access_key
RootPass: change_me
```

```
Command-line: https://docs.min.io/docs/minio-client-quickstart-guide
$ mc alias set myminio http://10.000.000.001:9000 access_key change_me
```

```
Documentation: https://docs.min.io
```

3. Create the `stop-minio.sh` script:

```
stop-minio.sh
```

```
#!/bin/bash
ps -ef | grep minio | awk '{print $2}' | xargs kill -9
NFS_SERVER=<NFS server-IP/FQDN>
NFS_DIR=/opt/minio-nfs
MINIO_DIR=/opt/minio-nfs-data
DATA_DIR=/opt/minio-nfs-data/minio/data
```

4. Run the `stop-minio.sh` script:

```
./stop-minio.sh
```

Output example:

```
Output:
echo ======================================
echo "$DATA_DIR still contains data"
echo "Clean them if needed"
```

```
echo "$MINIO_DIR is still mounted"
echo "umount it if needed"
echo ======================================
```

Setting Up MinIO Using a TLS-Signed Certificate

Example for configuring a TLS-signed certificate for MinIO.

1. Log in to the MinIO server and create the Sign-CA directory:

```
mkdir Sign-CA
cd Sign-CA
```

2. Generate the self-signed certificate:

```
openssl req -newkey rsa:4096 -sha256 -keyform PEM -keyout ca.key -x509 -
days 3650 -outform PEM -out ca.crt -subj
"/C=<country>/ST=<state>/L=<locality>/O=<organization>/OU=<organizational
unit>/CN=RootCA/emailAddress=<admin@myCompany.com>" -nodes
```

3. Generate a private key for MinIO:

```
openssl genrsa -out private.key 4096
```

4. Update the MinIO server's IP address and FQDN:

```
create openssl.conf ### Update minio server's IP and FQDN
[req]
distinguished_name = req_distinguished_name
x509_extensions = v3_req
prompt = no
```

```
[req_distinguished_name]
C = <country>
ST = <XX>
L = <locality>
O = <organization>
OU = <organizational unit>
CN = <fqdn>
[v3_req]
subjectAltName = @alt_names
[alt_names]
IP.1 = <IP-address>
DNS.1 = <fqdn>
```

5. Create the MinIO signing request:

```
#minio server IP: 10.000.000.001
```

```
openssl req -new -key private.key -out minio.csr -config openssl.conf -
nodes
```

6. Sign the CSR:

```
openssl x509 -req -in minio.csr -CA ca.crt -CAkey ca.key -CAcreateserial
-extensions server -days 3650 -outform PEM -out public.crt -sha256 -
extensions v3_req -extfile openssl.conf
```

7. Run the `start-minio.sh` script:

```
./start-minio.sh
```

```
cp public.crt private.key /root/.minio/certs
cp ca.crt /root/.minio/certs/CAs
```

8. Stop and restart MinIO:

```
./stop-minio.sh
```

```
./start-minio.sh
```

Example output:

```
Output:
========================================
/opt/minio-nfs-data/minio/data has been created
Data will be placed in /opt/minio-nfs-data/minio/data
Check output for more information
========================================
[root@n10-000-000-h001 minio]#
```

```
API: https://10.000.000.001:9000 https://192.168.122.1:9000
https://127.0.0.1:9000
RootUser: access_key
RootPass: change_me
```

```
Console: https://10.000.000.001:42497 https://192.168.122.1:42497
https://127.0.0.1:42497http://127.0.0.1:42497
RootUser: access_key
RootPass: change_me
```

```
Command-line: https://docs.min.io/docs/minio-client-quickstart-guide
$ mc alias set myminio http://10.000.000.000:9000 access_key change_me
```

```
Documentation: https://docs.min.io
```

9. Connect to the database server:

```
scp ca.crt <All-database-nodes>:/tmp
```

10. Run this command on all database nodes:

```
cp /tmp/ca.crt /etc/pki/ca-trust/source/anchors;update-ca-trust
```

Setting Up MinIO Using a TLS Self-Signed Certificate

Example for setting up MinIO without TLS self-signed certificate/

1. Log in to the MinIO server and create the self-sign directory:

```
mkdir self-sign
cd self-sign
```

2. Update the MinIO server's IP address and FQDN:

```
create openssl.conf ### Update minio server's IP and FQDN
[req]
distinguished_name = req_distinguished_name
x509_extensions = v3_req
prompt = no
```

```
[req_distinguished_name]
C = <country>
ST = <XX>
L = <locality>
O = <organization>
OU = <organizational unit>
CN = <fqdn>
```

```
[v3_req]
subjectAltName = @alt_names
```

```
[alt_names]
IP.1 = <IP-address>
DNS.1 = <fqdn>
```

3. Generate the self-signed certificate:

```
openssl req -newkey rsa:4096 -sha256 -keyform PEM -keyout private.key -
x509 -days 3650 -outform PEM -out public.crt -config openssl.conf -nodes
```

4. Run the `start-minio.sh` script:

```
./start-minio.sh
```

```
cp public.crt private.key /root/.minio/certs
cp public.crt /root/.minio/certs/CAs
```

5. Stop and restart MinIO:

```
./stop-minio.sh
```

```
./start-minio.sh
```

Example output:

```
Output:
=======================================
/opt/minio-nfs-data/minio/data has been created
Data will be placed in /opt/minio-nfs-data/minio/data
Check output for more information
=======================================
[root@n10-000-000-h001 minio]#
```

```
API: https://10.000.000.001:9000 https://192.168.122.1:9000
https://127.0.0.1:9000
RootUser: access_key
RootPass: change_me
```

```
Console: https://10.000.000.001:42497 https://192.168.122.1:42497
https://127.0.0.1:42497http://127.0.0.1:42497
RootUser: access_key
RootPass: change_me
```

```
Command-line: https://docs.min.io/docs/minio-client-quickstart-guide
$ mc alias set myminio http://10.000.000.000:9000 access_key change_me
```

```
Documentation: https://docs.min.io
```

6. Connect to the database:

```
scp public.crt <All-database-nodes>:/tmp
```

7. Run this command on all database nodes:

```
cp /tmp/public.crt /etc/pki/ca-trust/source/anchors;update-ca-trust
```

Creating a Bucket and Folder in MinIO

These are the basic steps for creating a bucket and folder in MinIO:

1. Log in to the MinIO console. For example:
   a. Enter the console address in a browser. For example (http://10.000.000.000:42497).
   b. Enter MinIO credentials (access key and password).
2. Create the bucket.
3. Select the new bucket, and create the folder (for example, "data").

To see how the Yaml configuration corresponds to the MinIO bucket, see Yaml Configuration File Example

Yaml Configuration File Example

The following is an example of the Yaml configuration file used for the MinIO setup. For more information, see Using the Configuration Files.

Non-TLS:

```
s3:
 type: preconfigured
 server: <fqdn>
 port: 9000
 url: s3://<bucketName>/<folderName>
 access-key: <access_key>
 tls-enabled: False
 region: <region>
```

TLS:

```
s3:
 type: preconfigured
 server: <fqdn>
 port: 9000
 url: s3://<bucketName>/<folderName>
 access-key: <access_key>
 tls-enabled: True
 region: <region>
```

Configuring MinIO TLS Mode for Database Back Up and Restore

These sections have instructions for configuring MinIO TLS mode to back up and restore the database using a TLS-signed certificate and self-signed certificate:

- Configuring the MinIO Backup Server Using a TLS-Signed Certificate
- Configuring the MinIO Backup Server Using a TLS Self-Signed Certificate

Configuring the MinIO Backup Server Using a TLS-Signed Certificate

Follow these steps to configure the MinIO backup server using a TLS-signed certificate:

1. Log in to the MinIO source server.
2. Change to the Certificate directory:

```
cd Sign-CA
```

3. Generate the private key for MinIO's backup server:

```
openssl genrsa -out backup_minio_private.key 4096
```

4. Create the `backup_minio_openssl.conf` file:

```
[req]
distinguished_name = req_distinguished_name
x509_extensions = v3_req
prompt = no
[req_distinguished_name]
C = <country>
ST = <XX>
L = <locality>
O = <organization>
OU = <organizational unit>
CN = <MinIOBackupfqdn>
[v3_req]
subjectAltName = @alt_names
[alt_names]
IP.1 = <MinioBackupIP-address>
DNS.1 = <MinIOBackupfqdn>
```

5. Create the MinIO backup server signing request:

```
openssl req -new -key backup_minio_private.key -out backup_minio.csr -
config backup_minio_openssl.conf -nodes
```

6. Sign the certificate sign-request using ca.crt from the MinIO source server:

```
openssl x509 -req -in backup_minio.csr -CA ca.crt -CAkey ca.key -
CAcreateserial -extensions server -days 3650 -outform PEM -out backup_
minio_public.crt -sha256  -extensions v3_req -extfile backup_minio_
openssl.conf
```

7. Run this command on the MinIO backup server:

```
./start-minio.sh
```

8. Run these commands on the MinIO server:

```
cd Sign-CA
scp backup_minio_public.crt <backup_minio_server_
IP>:/root/.minio/certs/public.crt
scp backup_minio_private.key <backup_minio_server_
IP>:/root/.minio/certs/private.key
scp ca.crt <backup_minio_server_IP>:/root/.minio/certs/CAs
scp ca.crt <backup_minio_server_IP>:/tmp
```

9. Stop and start the MinIO backup server:

```
./stop-minio.sh
./start-minio.sh
```

10. Log into the Console and create a bucket and folder on MinIO backup server. For more information, see Creating a Bucket and Folder in MinIO.

Configuring the MinIO Backup Server Using a TLS Self-Signed Certificate

Follow these steps to configure the MinIO backup server using a TLS self-signed certificate:

1. Log in to the MinIO backup server and create the self-sign directory:

```
mkdir self-sign
cd self-sign
```

2. Create the `backup_minio_openssl.conf` file:

```
[req]
distinguished_name = req_distinguished_name
x509_extensions = v3_req
prompt = no
[req_distinguished_name]
C = <country>
ST = <XX>
L = <locality>
O = <organization>
```

```
OU = <organizational unit>
CN = <MinIOBackupfqdn>
[v3_req]
subjectAltName = @alt_names
[alt_names]
IP.1 = <MinioBackupIP-address>
DNS.1 = <MinIOBackupfqdn>
```

3. Generate the self-signed certificate:

```
openssl req -newkey rsa:4096 -sha256 -keyform PEM -keyout backup_
private.key -x509 -days 3650 -outform PEM -out backup_public.crt -config
backup_minio_openssl.conf -nodes
```

4. Start MinIO:

```
./start-minio.sh
```

5. Copy the `backup_private.key` and `backup_public.crt` files to the certificate directory:

```
cp backup_private.key backup_public.crt /root/.minio/certs
cp backup_public.crt /root/.minio/certs/CAs/ca.crt
```

6. Stop and restart MinIO:

```
./stop-minio.sh
./start-minio.sh
```

7. log into the Console and create a bucket and folder on MinIO backup server. For more information, see Creating a Bucket and Folder in MinIO.
8. Connect to the database:

```
scp backup_public.crt <All-database-nodes>:/tmp
```

9. Run this command on all database nodes:

```
cp /tmp/backup_public.crt /etc/pki/ca-trust/source/anchors;update-ca-
trust
```

Configuring the Database for Backup

Run these commands on database node1 to configure the database for backup:

```
cd /opt/acsight-db-tools/config
vi backup_restore_cloud_storage_base.ini
change
; cloud_storage_ca_bundle = /home/user/ssl_folder/ca_bundle.pem
```

```
to
cloud_storage_ca_bundle = /etc/pki/tls/certs/ca-bundle.crt
```

You are now ready to proceed with the backup procedures.

Specifying Kafka Scheduler Options

The database uses an event consumer, the **Kafka scheduler**, to ingest events from Transformation Hub's Kafka component. To obtain the event schema from Transformation Hub's Schema Registry component, Kafka scheduler uses the schema registry server setting. When you modify the schema registry settings, you affect Kafka scheduler. Thus, you must stop and delete the scheduler, make your changes to the registry, then create the scheduler again.

To manage Kafka scheduler, enter `./kafka_scheduler <Option_Name>` where you include any of the following options:

Option Name	Description
update	Updates the scheduler
start	Starts the scheduler and begins copying data from all registered Kafka brokers
stop	Stops the scheduler and ends copying data from all registered Kafka brokers
delete	Deletes all registered Kafka instances from the scheduler
status	Prints the following information and log status for a running or stopped scheduler: • Current Kafka cluster assigned to the scheduler • Name and database host where the active scheduler is running • Name, database host, and process ID of every running scheduler (active or backup)
events	Prints event copy progress for the scheduler
messages	Prints scheduler messages

Managing Intelligence

This section provides guidance for managing Intelligence functions and features within the deployment.

Enabling Windowed Analytics

By default, Intelligence is configured to run Analytics in batch mode. When new data is ingested, Analytics is run on both the new and the existing data. Although this process is

beneficial when you first deploy Intelligence (for testing and validation purposes), running Analytics on the entirety of your data on an ongoing basis unnecessarily uses system resources. Instead, you can enable Windowed Analytics.

When you enable Windowed Analytics, you configure Intelligence to run Analytics only on newly ingested data as determined by the date of the last Analytics run and the timestamp of the data. Intelligence identifies the data it has already analyzed, then runs Analytics only on the new data. These results are then aggregated with the existing results to produce updated, current Analytics results for the entire data set.

Windowed Analytics has a positive impact on performance and stability because it allows the system to analyze and aggregate smaller, more consistently sized quantities of data than batch mode, particularly as the total amount of data in your system continues to grow.

Important: After you have validated the initial data ingest and Analytics run for your Intelligence cluster, you might need to ingest and analyze historical data. In this scenario, you must continue to run Analytics in batch mode to ensure that all data is included.

To enable Windowed Analytics:

1. Open a certified web browser.
2. Specify the following URL to log in to the CDF Management Portal: https://<cdf_masternode_hostname or virtual_ip hostname>:5443.
3. Select **Deployment** > **Deployments**.
4. Click ... (Browse) on the far right and choose **Reconfigure**. A new screen will be opened in a separate tab.
5. Click **Intelligence** and disable **Batch Processing**.
6. Click **Save**.

   The first Windowed Analytics run performs a full batch run to establish the baseline for the system going forward. The second and subsequent runs occur as Windowed Analytics.

Configuring the 'Peek-Back' Window for Windowed Analytics

The 'peek-back' window is a best-effort buffer that ensures that delayed or out-of-order data is not missed between Windowed Analytics runs. For more information on configuring the peek-back window, see the *Configuring the 'Peek-Back' Window for Windowed Analytics* section **in the Intelligence User's Guide.**

Running Analytics on Demand

Before you run Analytics on demand, do the following:

- Ensure that Analytics is not already running because running Analytics on demand can cause Analytics in progress to fail.
- If the previous Analytics execution failed, check whether the properties in the Intelligence tab are set correctly. If this does not solve the issue, contact Micro Focus Customer Support.

To run Analytics on demand:

1. Launch a terminal session and log in to the NFS node.
2. Navigate to the following directory:

   ```
   cd <NFSVolume>/interset/analytics
   ```

3. (Conditional) Delete the `blackhawk_down` file, if present. This is an error file and it is generated if the previous Analytics execution fails.

   ```
   rm blackhawk_down
   ```

4. When prompted whether you want to delete the file, specify yes.
5. Execute the following command to delete the latest `AnalyticsStarted.mk` and `AnalyticsCompleted` files:

   ```
   rm -rf AnalyticsStarted-0-<Today's_date>.mk AnalyticsCompleted-0-
   <Today's_date>.mk
   ```

6. When prompted whether you want to delete the files, specify yes.

   After 30 seconds of deletion of the files, Analytics is triggered automatically.

Changing Passwords for a Secure Environment

You can change the passwords for the components during deployment and also at any point after deployment, as needed.

1. Open a certified web browser.
2. Specify the following URL to log in to the CDF Management Portal: https://<cdf_masternode_hostname or virtual_ip hostname>:5443.
3. Select **Deployment** > **Deployments**.
4. Click ... (Browse) on the far right and choose **Reconfigure**. A new screen will be opened in a separate tab.

5. Click **Intelligence** and modify the passwords.
6. Click **Save**.

Changing the Elasticsearch Node Data Path

To change the Elasticsearch node data path, perform the following steps:

1. Launch a terminal session and as a root user, log in to a worker node labeled as **intelligence:yes**.
2. Execute the following commands to scale down the Elasticsearch master node and Elasticsearch data nodes:

```
export NS=$(kubectl get namespaces |grep arcsight|cut -d ' ' -f1)
kubectl -n $NS scale statefulset elasticsearch-master --replicas=0
kubectl -n $NS scale statefulset elasticsearch-data --replicas=0
```

3. (Conditional) To create an Elasticsearch data directory in the NFS server, log in to the server.
4. (Conditional) To create a new Elasticsearch data directory in a worker node labeled as **intelligence:yes**, log in to the node.
5. Execute the following commands to create a new directory:

```
cd <path to create the new directory>
mkdir <new directory in the path>
```

If you are creating a new directory in the NFS server, ensure that the directory is accessible or mounted on all the worker nodes labeled as **intelligence:yes** . The Elasticsearch data directory in the NFS server might impact the system performance.

6. Execute the following command to copy data from the existing directory to the new directory:
   - To copy the data to a worker node labeled as **intelligence:yes**:

     ```
     cp -rf <existing_directory_path> <new_directory_path>
     ```

     For example:

     ```
     cp -rf /opt/arcsight/k8s-hostpath-volume/interset
     /opt/arcsight/testpath/
     ```

     In this example, the existing directory path `/opt/arcsight/k8s-hostpath-volume/interset` and the new directory path is `/opt/arcsight/testpath/`.

- To copy the data to the NFS server:

```
scp -rf <existing_directory_path> root@<ip address or hostname of the
NFS server>:<new_directory_path>
```

7. Execute the following command to change the permissions of the new directory:

```
chown 1999:1999* <new_directory_path>
```

   For example:

```
chown 1999:1999* /opt/arcsight/testpath/
```

8. If you have created a new Elasticsearch directory in a worker node labeled as **intelligence:yes**, then repeat Steps 4 to 7 on all the worker nodes labeled as **intelligence:yes**.
9. Open a certified web browser.
10. Specify the following URL to log in to the CDF Management Portal: https://<cdf_ masternode_hostname or virtual_ip hostname>:5443.
11. Select **Deployment** > **Deployments**.
12. Click ... (Browse) on the far right and choose **Reconfigure**. A new screen will be opened in a separate tab.
13. Click **Intelligence** and provide the new value of the Elasticsearch directory path in the **Elasticsearch Node Data Path to persist data to** field.
14. Click **Save**.
15. Launch a terminal session and as a root user, log in to a worker node labeled as **intelligence:yes**.
16. Execute the following commands to scale up the Elasticsearch master node and Elasticsearch data nodes:

```
export NS=$(kubectl get namespaces |grep arcsight|cut -d ' ' -f1)
kubectl -n $NS scale statefulset elasticsearch-master --replicas=1
kubectl -n $NS scale statefulset elasticsearch-data --replicas=<number_
of_replicas>
```

17. Execute the following curl command on any Kubernetes node and verify the status of the Elasticsearch cluster:

```
curl -k "https://<Elasticsearch_username:Elasticsearch_password>@<ip
address or hostname of the CDF>:31092/_cluster/health"
```

Enabling Elasticsearch to Start on Limited Hardware Sizing

If Elasticsearch is not able to start because of a lack of CPU resources, you can modify the **Elasticsearch Minimum Cores** field in the CDF Management Portal to enable Elasticsearch to start.

1. Open a certified web browser.
2. Specify the following URL to log in to the CDF Management Portal: https://<cdf_ masternode_hostname or virtual_ip hostname>:5443.
3. Select **Deployment** > **Deployments**.
4. Click ... (Browse) on the far right and choose **Reconfigure**. A new screen will be opened in a separate tab.
5. Click **Intelligence**.
6. In the **Elasticsearch Configuration** section, modify the value of the **Elasticsearch Minimum Cores** field.

   For example, for a 0.5 CPU, you can specify the corresponding value in any of the following formats:

   - 500m
   - 0.5
7. Click **Save**.

Updating the Logstash Config Map for Custom Data Identifiers

If you are using custom data identifiers (dids) to identify a specific data type or machine users, then, you must update the `logstash-config-pipeline` config map with custom data identifiers so that you can view the events or explore the raw events corresponding to the anomalies of the custom dids.

1. Open a certified web browser.
2. Specify the following URL to log in to the **CDF Management Portal**: https://<cdf_ masternode_hostname or virtual_ip hostname>:5443.
3. Navigate to **Cluster > Dashboard** to access the Kubernetes Dashboard.
4. Under **Namespace**, search and select the **arcsight-installer-xxxx** namespace.
5. Under **Config and Storage**, click **Config Maps**.
6. Click the filter icon, then search for `logstash-config-pipeline`.

7. Click  and select **Edit**.
8. Add the required mapping corresponding to custom did under the 'filter' section of `logstash-config-pipeline`. For example:

```
if [destinationNtDomain] {
 if [destinationNtDomain] in ['','WORKGROUP','NT SERVICE','NT AUTHORITY']
 {
     mutate {
        replace => {
           "did" => "1"
        }
     }
 }
}
if [destinationUserName] =~ "\$$" {
 mutate
    replace => {
      "did" => "1"
     }
 }
}
```

> If you have upgraded Intelligence, you can update the logstash config map with the custom did mappings used in the previous version of Intelligence, if required. To update, copy the necessary mappings from the logstash-config-pipeline config map that you had backed up prior to the upgrade.

9. Click **Update**.
10. Restart the `interset-logstash` pods:
    a. Launch a terminal session and log in to the master or worker node.
    b. Execute the following command to retrieve the namespace:

```
export NS=$(kubectl get namespaces | grep arcsight|cut -d ' ' -f1)
```

    c. Execute the following commands to restart the interset-logstash pods:

```
kubectl -n $NS scale statefulset interset-logstash --replicas=0
kubectl -n $NS scale statefulset interset-logstash --replicas=3 (set
as per the environment)
```

For more mapping instances for custom dids, contact Micro Focus Customer Support.

Enabling Custom Model Support

This section provides guidance for enabling custom model support in Intelligence.

Introduction

Intelligence provides support for custom machine learning (ML) models. This support enables you to import trained ML models into Intelligence. Intelligence can then use these models to run analytics on the incoming data, that is, detect anomalies and generate risk scores for the associated entities, and display the analytics results in the Intelligence dashboard.

The introduction of this feature enables you to enhance Intelligence with models that provide analytics tailored to your unique environments. It also provides a method for extending Intelligence analytics to address new use cases such as the detection of new patterns of unusual behavior.

Important: Intelligence accepts only those custom models that are in the Predictive Model Markup Language (PMML) format and are based on the data types supported by Intelligence.

Supported Algorithms

The following types of algorithms are supported by Intelligence:

- Classification Algorithms
- Anomaly Detection Algorithms

Classification Algorithms

Intelligence supports those classification algorithms that can be stated as classification problems with two classes of output as follows:

- Anomalous
- Non anomalous

The classes can take any name. A classification algorithm must provide a probability for each class.

You must provide one of the two classes in the **targetClass** parameter while registering the model. This class is used to filter the events that will be considered for determining anomalies.

The following classification algorithms are supported by Intelligence:

- General Regression
- Naïve Bayes

- Neural Network
- Regression
- Rule Set
- Support Vector Machine
- Tree

Anomaly Detection Algorithms

Anomaly Detection algorithms output two values:

- A score on the event.
- A Boolean value indicating if the score is anomalous or not.

These anomaly detection algorithms do not measure the anomalousness of an event, but they give a Boolean value indicating whether the event is anomalous or not. Intelligence interprets a Boolean value of true as an anomalous event and a Boolean value of false as a non anomalous event while calculating the risk scores for the entities associated with the anomalous events.

You must provide one of the two Boolean values in the **targetClass** parameter while registering the model. This value is used to filter the events that will be considered for determining anomalies.

Supported Data Types

Intelligence supports the ingestion and analysis of data of the following data types:

- **Access**
  The Access schema represents events collected from Identity and Access Management (IAM) solutions where users access resources such as servers or fileshares. For more information on the Access schema, see Access in Intelligence Data Types and Schemas.
- **Active Directory**
  The Active Directory schema represents events collected from Identity and Access Management (IAM) solutions that identify successful and failed logins to authentication targets. These authentication targets include domain controllers/servers, resources, and file shares. For more information on the Active Directory schema, see Active Directory in Intelligence Data Types and Schemas.
- **VPN**
  The VPN schema represents events collected from Identity and Access Management (IAM) solutions or from other VPN devices such as Pulse Secure that identify VPN events. For more information on the VPN schema, see VPN in Intelligence Data Types and Schemas.
- **Web Proxy**
  Web Proxy data are raw events that capture network traffic, primarily Web surfing, from a

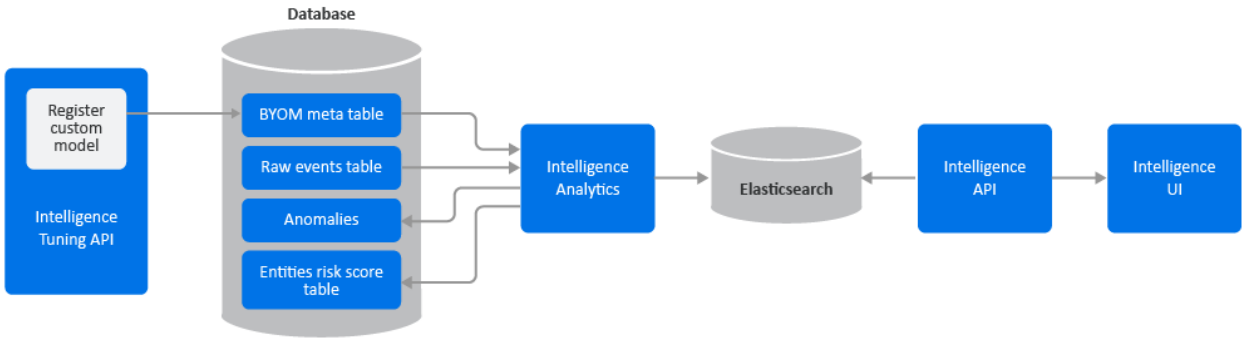
collection of human users. For more information on the Web Proxy schema, see Web Proxy in Intelligence Data Types and Schemas.

- **Repository**
  Repository data are raw events collected from a source control (repository) system. For more information on the Repository data type, see Repository in Intelligence Data Types and Schemas.

For the supported data types, there are corresponding SmartConnectors and FlexConnectors. For more information, see the *Supported Data Types and Supported SmartConnectors/FlexConnectors Types* section in the Technical Requirements Guide. In addition, for the supported data types, Intelligence provides support for new devices which provide data of relevance to the Intelligence analytics models. For more information, see Adding Support for New Devices. You can also refer to the Ingesting Sample CSV Data section to get an understanding of data ingest.

The Custom Model Support Architecture

The following diagram helps you understand the custom model support architecture:



The following table describes the components involved in the custom model support architecture:

Component	Description
Intelligence Tuning API	API that provides a way of registering a custom model with Intelligence. By registering the model, you are importing the model into Intelligence. This API also allows for the management of the custom models.
Database	When a model is registered through the Intelligence Tuning API, the PMML file and the other metadata of the model are stored in the BYOM_meta table of the database. The database also stores the raw events (incoming data from different data sources) in the Raw events table. In addition to events, the database stores the Intelligence analytics data in the Anomalies table and the Entities risk score table.

Component	Description
Intelligence Analytics	Performs the vital task of determining individual behavioral baselines, then discovering and ranking deviations from those baselines. It reads data from the Raw events table and uses the model with the help of the model data in the BYOM meta table to generate the analytics data, that is, anomalies in the Anomalies table and entities' scores in the Entities risk score table. It also stores the analytics data in Elasticsearch.
Elasticsearch	Elasticsearch is an open source, broadly-distributable and easily-scalable enterprise-grade search engine. Elasticsearch houses all the Intelligence analytics results and raw events, and it provides all the data that drives the user interface.
Intelligence API	Intelligence API reads data from Elasticsearch and provides the REST API from which Intelligence UI gathers the Intelligence analytics results and raw events.
Intelligence UI	Provides a rich user interface that allows you to visually explore the Intelligence analytics results and raw events in the Intelligence dashboard.

Understanding the Custom Model Support Process

The end-to-end Custom Model Support process is as follows:

- **Identify your use case** - Identify the use case on which you need to build your custom model. Intelligence supports the Classification and Anomaly Detection algorithms. For more information, see Supported Algorithms.
- **Identify input data columns** - Before you build your model, ensure that the feature column names of the model match the column names present in the **default_secops_adm.events** input data table. This ensures that all the column names in your sample data set on which you are training your custom model match at least a subset of the columns in the **default_secops_adm.events** table. When analytics is run on the incoming data, only the selected column values are considered for arriving at the results. For more information on the event's table column information, see Understanding the Schema for Events.
- **Create and train your model** - Create your model using your data set and the identified column values. Train the model using the selected data set and create a PMML model file that can represent the derived model.
- **Register a custom model** - Import a custom model into Intelligence by registering it through the API. Registering provides a way to import the model's PMML file and provide other metadata associated with the model. You must provide one of the two classes in the **targetClass** parameter while registering the model. This class is used to filter the events that will be considered to determine anomalies. For more information, see the *Registering a Custom Model* section in the ArcSight Intelligence User's Guide.

- **Run analytics** - The registered model is used in analytics and results are derived for it when analytics is run on the next scheduled time or when you run analytics on demand. For more information on running analytics, see Running Analytics on Demand.
- **View the analytics results on the Intelligence dashboard** – Next, you can view the analytics results and explore the underlying raw events in the Intelligence Dashboard. For more information, see the *Understanding the Intelligence Dashboard* section in the ArcSight Intelligence User's Guide.
- **Download CSV and PDF reports** - You can view reports that provide you with further insight into risky entities and their behaviors. For more information, see the *Viewing Reports* section in the ArcSight Intelligence User's Guide.

Additional Tasks on Custom Models

You can perform additional tasks with respect to custom models in Intelligence as follows:

- **Manage the custom models** - You can manage the custom models such as activate or deactivate a model, or tune a model. For more information, see Managing Custom Models.
- **Manage alert templates** - An alert template provides a way to describe an anomaly in the Intelligence UI by using the textual information provided as part of the alert template's meta data. You can customize the alert templates associated with a custom model. For more information, see Managing Alert Templates.

Input Data Table Schema

The custom model that you develop for an Intelligence supported data type must be based on the input data table schema of the database. Before you build your model, ensure that the feature column names of the model match the column names present in the input data table. You can refer to the **default_secops_adm.events** input data table present in Understanding the Schema for Events to identify the columns in the events table that must be used for that model. This ensures that all the column names in your sample data set on which you are training your custom model match at least a subset of the columns in the **default_secops_adm.events** table. When analytics is run on the incoming data and a custom model is used, only the selected column values are considered for arriving at the results.

Before You Proceed

Before you register your models and manage them in Intelligence, ensure the following:

- The models are in the PMML format.
- The models are based on the algorithms supported by Intelligence. For more information on the supported algorithms, see Supported Algorithms.

- The models are based on the data types supported by Intelligence. For more information on the data types supported, see Supported Data Types.
- The models are based on the input data table schema. For more information, see Input Data Table Schema.
- The models are trained.
- SmartConnectors and FlexConnectors are configured for data collection. For more information about data collection, see SmartConnector Installation and User Guide, SmartConnector Configuration Guides, and ArcSight FlexConnector Developer's Guide. You can also refer to the Ingesting Sample CSV Data section to get an understanding of data ingest.

Managing Custom Models

You can import a custom model into Intelligence by registering it through the API. After registering models, you can also manage them.

For more information, see the Managing Custom Models section in the ArcSight Intelligence User's Guide.

Managing Alert Templates

An alert template provides a way to describe an anomaly in the Intelligence UI by using the textual information provided as part of the alert template's meta data. It also helps in associating an anomaly with all the events that triggered it. When you register a model with Intelligence, an anomaly type and an alert template are automatically created for the model. You can customize the created alert templates to suit your needs, create new alert templates, and so on.

For more information, see the *Managing Alert Templates* section in the ArcSight Intelligence User's Guide.

Appendix: Ingesting and Exporting Input Data

This section provides guidance on the following:

- Ingesting sample CSV data to the database input data table.
- Exporting data from the database tables to the CSV format.

Ingesting Sample CSV Data to the Input Data Table

SmartConnectors are applications that collect events from different devices, process them, and send them to the desired destinations.

If SmartConnectors are not available for a particular device of an Intelligence supported data type, you can create FlexConnectors that can read and parse information from the devices and map that information to ArcSight's event schema. FlexConnectors are custom connectors you define to gather security events from log files, databases, and other software and devices. For every FlexConnector that you create, you need to create a corresponding configuration file. A configuration file is a text file containing properties (name, value pairs) that describe how the FlexConnector parses event data.

This section provides guidance on ingesting sample CSV data of a supported data type (for example, Active Directory) to the **default_secops_adm.events** database input data table with the help of FlexConnectors.

Note:

- This section goes on the assumption that SmartConnectors are not available for collecting the sample CSV data and that you must create a FlexConnector.
- This section is intended only for a sample CSV data of the Active Directory data type. If you need to add a new device for any of the Intelligence supported data types, see Adding Support for New Devices.

Configuration File

The configuration file provided in this section is designed only for the sample data set provided here for the Active Directory data type. This configuration file is used by the FlexConnector to parse the CSV data and convert it to the CEF format. The configuration file must be in this format - **<file_name>.sdkfilereader.properties**. For example, **testdata.sdkfilereader.properties**.

Sample CSV Data of the Active Directory Data Type

```
destinationUserName,categoryOutcome,externalId,destinationHostName,deviceRece
iptTime,deviceCustomString5
bennett.merry,Success,4659,OTTAWADC.interset.com,2016-04-01T08:00:04-05:00,
pamila.dankert,Success,4659,NFMC.interset.com,2016-04-01T08:00:18-05:00,
pamila.dankert,Failure,4777,NFMC.interset.com,2016-04-01T08:00:20-05:00,
lakendra.danielson,Success,4634,NFMC.interset.com,2016-04-01T08:00:27-05:00,3
```

Configuration File for the Sample CSV Data

```
delimeter=,
text.qualifier="
comments.start.with=\#
trim.tokens=true
contains.empty.tokens=true

token.count=6
```

```
token[0].name=destinationUserName
token[0].type=String
token[1].name=categoryOutcome
token[1].type=String
token[2].name=externalId
token[2].type=String
token[3].name=destinationHostName
token[3].type=String
token[4].name=deviceReceiptTime
token[4].type=String
token[5].name=deviceCustomString5
token[5].type=String

event.destinationNtDomain=__stringConstant("WIN-MP0VNBBQVSI")
event.categoryBehavior=__stringConstant("/Authentication/Verify")
event.categoryObject=__stringConstant("/Host/Operating System")
event.deviceProduct=__stringConstant("Microsoft Windows")
event.deviceVendor=__getVendor("Microsoft")
event.deviceReceiptTime=__createOptionalTimeStampFromString
(deviceReceiptTime,"YYYY-MM-DDThh:mm:ss.SSSX")
event.destinationUserName=destinationUserName
event.categoryOutcome=categoryOutcome
event.externalId=externalId
event.destinationHostName=destinationHostName
event.deviceCustomString5=deviceCustomString5
```

You can also create or customize the configurations files for other data sets of the supported data types. For more information, see ArcSight FlexConnector Developer's Guide.

FlexConnector Installation and Configuration

To install and configure a FlexConnector, see ArcSight FlexConnector Developer's Guide.

Ensure the following when you install and configure the FlexConnector:

- Select **ArcSight FlexConnector File** as the **Connector Type**.
- When adding the parameters information, specify the following:
  - Select **Log Unparsed Events** as **False**.
  - Provide the absolute path and the CSV file name that the FlexConnector needs to read in the **Log File Name** field.
    For example, c:\temp\sample_data.csv.
  - For the **Configuration File** field, specify only the file name that you used in the configuration file.
    For example, if the configuration file is in this format - **testdata.sdkfilereader.properties**,

then specify only **testdata**. The suffix **.sdkfilereader.properties** is appended automatically. The configuration file name now is **testdata.sdkfilereader.properties**.

- When configuring the destination, select either **CEF File** or **Transformation Hub** as the destination. For more information, see SmartConnector Installation and User Guide.

Post-Installation Tasks

After you install and configure the FlexConnector and before you run the FlexConnector, copy the configuration file in the **ARCSIGHT_HOME\user\agent\flexagent** location.

Sending Data to the Input Data Table

To send data to the input data table, you need to start the SmartConnector/FlexConnector. You can run the SmartConnector/FlexConnector in standalone mode or as a service, depending on the mode you selected during installation.

Running in Standalone Mode

If you have installed the SmartConnector/FlexConnector in the standalone mode, you need to start it manually (periodically or as per your requirement). Also, you need to start the SmartConnector/FlexConnector whenever the host on which it is installed is restarted, because the SmartConnector/FlexConnector is not automatically active when the host is restarted.

Perform the following steps to start the SmartConnector/FlexConnector agent so that it can send the CSV data to the Transformation Hub topic and which will then be loaded to the database events table.

1. Change to the following directory:

```
cd $ARCSIGHT_HOME\current\bin\
```

2. Execute the following command:

```
./arcsight agents
```

Running as a Windows Service

To start or stop the SmartConnector/FlexConnector installed as a service on the Windows platform:

1. Right-click **My Computer**, then select **Manage** from the **Context** menu.
2. Expand the **Services and Applications** folder and select **Services**.
3. Right-click the SmartConnector/FlexConnector service name and select **Start** to run the SmartConnector/FlexConnector or **Stop** to stop the service.

To verify that the SmartConnector/FlexConnector service has started, view the following file:

```
$ARCSIGHT_HOME/logs/agent.out.wrapper.log
```

To reconfigure the SmartConnector/FlexConnector as a service, run the SmartConnectorConfiguration/FlexConnectorConfiguration Wizard again. Open a command window on $ARCSIGHT_HOME/current/bin and run:

```
./runagentsetup
```

Exporting Data from the Database Tables to the CSV Format

You can export data from the database tables to the CSV format by using VSQL and its output format options. These options can be set either from within an interactive vsql session, or through command-line arguments to the vsql command (making the export process suitable for automation through scripting). After setting VSQL options so it outputs the data in a format your target system can read, you can run a query and capture the result in a CSV file. The procedure mentioned here is for some of the VSQL output format options. To know more about the available output format options, see the Database Documentation.

To export data from the database tables:

1. Launch a terminal session and log in to a database node.
2. Change to the following directory:

   ```
   cd /opt/vertica/bin/
   ```

3. Log in as a dbadmin:

   ```
   su dbadmin
   ```

4. (Conditional) To create an output file directly from the command line by passing parameters to vsql, execute the following commands:

   ```
   vsql -U username -F ',' -At -o <outputfile_name> -c "SELECT * FROM
   <table_name>;"
   ```

   where
   -F is used to set the field separator. In this case, because the output is a CSV file, the field separator is ','.

   -At is used to disable the padding and show only the table's tuples in the output file. If you want to show the table headings and the row counts, do not specify t.

   -o is used to send the output to the output file.

   <outputfile_name> is the output file name you need to provide to save the data to, for example, **test.csv**. Ensure that you have write permissions to the output file.

-c is used to run the SQL query and

<table_name> is the table whose data you want to export, for example, **default_secops_adm.events**.

```
[password prompt]
```

5. (Conditional) To create an output file within an interactive vsql session, do the following.
   a. Log in to vsql and specify the password when prompted.

      ```
      vsql
      ```

      ```
      [password prompt]
      ```

   b. Execute the following command to disable padding so as to align the output:

      ```
      \a
      ```

   c. (Optional) Execute the following command to export only the table tuples to the output file:

      ```
      \t
      ```

   d. Execute the following command to set the field separator to export data in the CSV format:

      ```
      \pset fieldsep ','
      ```

   e. Execute the following command to save the output to a file:

      ```
      \o <outputfile_name>
      ```

      where <outputfile_name> is the output file name you need to provide to save the data to, for example, **test.csv**. Ensure that you have write permissions to the output file.

   f. Execute the following command to export data from a database table to the output file you specified in the previous step:

      ```
      select * from <table_name>;
      ```

      where <table_name> is the table whose data you want to export, for example, **default_secops_adm.events**.

   g. Execute the following command to view the data in the output file:

      ```
      \! cat <outputfile_name>
      ```

Adding Support for New Devices

Intelligence supports the ingestion and analysis of data of the following data types:

- Access
- Active Directory
- VPN
- Web Proxy
- Repository

For the supported data types, Intelligence also provides support for new devices that provide data of relevance to the Intelligence analytics models. This section provides information on supporting new devices.

Checklist: Implementation

To add the support for new devices, perform the following tasks in the listed order.

	Task	See
❑	(Conditional) If SmartConnectors are available for the new device, install and configure SmartConnectors for data collection.	SmartConnectors
❑	(Conditional) If SmartConnectors are not available for the new device, install and configure FlexConnectors for data collection.	FlexConnectors
❑	(Conditional) If you have installed and configured FlexConnectors, perform data engineering.	Data Engineering
❑	(Conditional) If you have installed and configured FlexConnectors, perform event categorization.	Event Categorization
❑	Generate SQL Loader Scripts.	SQL Loader Scripts
❑	Update the Intelligence tables required for relations.	Intelligence Tables

SmartConnectors

SmartConnectors are applications that collect events from different devices, process them, and send them to the desired destinations. SmartConnectors are available for the following data types supported by Intelligence:

- Access
- Active Directory
- VPN
- Web Proxy

For more information about the SmartConnectors for the supported data types, see the Supported Data Sources and SmartConnectors/FlexConnectors section. If a new device needs to be supported for any of these data types for which there are corresponding SmartConnectors, then you can configure the SmartConnector for data collection. For more information, see SmartConnector Installation and User Guide and SmartConnector Configuration Guides.

FlexConnectors

If there are no SmartConnectors for a new device of the supported data types, you can create FlexConnectors that can read and parse information from the devices and map that information to ArcSight's event schema. FlexConnectors are custom connectors you define to gather security events from log files, databases, and other software and devices. For the data of repository type, that is, GitHub Enterprise, Bitbucket Server, and Perforce, you can create FlexConnectors to collect the data. For every FlexConnector that you create, you need to create a corresponding configuration file. A configuration file is a text file containing properties (name, value pairs) that describe how the FlexConnector parses event data. For more information about FlexConnectors and the configuration files, see ArcSight FlexConnector Developer’s Guide.

Data Engineering

When a new device is supported and a FlexConnector is configured for it, you must identify data fields that are required, on which Intelligence must run analytics. Data engineering is the process of selecting the fields/columns that is required for Intelligence Analytics. This also entails cleansing the data and filtering it from unwanted information, such as noise.

Perform the following steps for data engineering:

1. Clean up data.
2. Filter data.
3. Normalize data. Perform the following as part of normalizing data:
    a. Ensure that the username is in lowercase.
    b. Set the depth value for filepath.
    c. Perform entity mapping.

For more details on data engineering, contact Micro Focus Customer Support.

Event Categorization

When a new device is supported and a FlexConnector is configured for it, you must perform event categorization. Event Categorization is the process of identifying the type and nature of events and categorizing them into groups. Categorizing events is helpful when customizing SQL Loader Scripts to filter specific types of events. For more information, see Event Categorization WhitePaper.

SQL Loader Scripts

To support a new device of the supported data types, you must update the corresponding loader scripts. For more information, contact Micro Focus Customer Support.

Intelligence Tables

The support of a new device necessitates updating the Intelligence schema tables so that Intelligence analytics can run on the data from the new device. For more information, contact Micro Focus Customer Support.

Securing HDFS for Intelligence

HDFS (Apache Hadoop Distributed File System) is a distributed file system, which is deployed on the worker nodes of the CDF cluster by default. The Intelligence analytics platform uses HDFS as a temporary storage to push analytics data to the ArcSight database. HDFS stores analytics data only when the write process is active.

Intelligence now allows you to secure access to HDFS with SASL (Simple Authentication and Security Layer). To secure HDFS, you can enable and configure Kerberos authentication services. When you configure HDFS to run in a secure mode, Kerberos authenticates each HDFS service and user. For authentication, Intelligence uses the Kerberos protocol, which is built on a trusted third-party encryption server, known as **Key Distribution Center (KDC)**.

The secure data transfer between HDFS and the database is disabled by default. Enabling the secure data transfer between HDFS and the database will increase the run time of the analytics jobs.

Enabling and Configuring Kerberos Authentication

This section provides information on enabling and configuring kerberos authentication for securing HDFS. Perform the tasks in the listed order:

	Task	See
❑	(Conditional) For Linux, configure the Kerberos Key Distribution Centre.	Configuring Kerberos Key Distribution Centre in Linux
❑	(Conditional) For Windows, set up your environment to configure Kerberos KDC.	Setting Up Your Windows Environment to Configure Kerberos KDC
❑	(Conditional) For Windows, create service and user principals for Kerberos ticket generation.	Creating Service Principals for Kerberos Ticket Generation in Windows
❑	Configure HDFS services to use keytabs.	Configuring HDFS Services to Use Keytabs

Configuring Kerberos Key Distribution Centre in Linux

To configure Kerberos Key Distribution Centre (KDC):

1. Install MIT Kerberos on any of the Kubernetes nodes in the CDF cluster. Refer to the open source documentation to perform this step.
2. As a root user, log in to the node where MIT Kerberos is installed, then create a service principal for HDFS:

```
$kadmin.local
$addprinc hdfs/<DATANODE_HOST>
```

3. Generate the keytab for the service principal created in **step 2**:

```
$kadmin.local
$ktadd -k hdfs/<DATANODE_HOST>.keytab hdfs/<DATANODE_HOST>
```

4. As a root user, log in to the node where MIT Kerberos is installed, then create a service principal for HTTP:

```
$addprinc HTTP/<DATANODE_HOST>
```

5. Generate the keytab for the service principal created in **step 4**:

```
$kadmin.local
$ktadd -k HTTP/<DATANODE_HOST>.keytab HTTP/<DATANODE_HOST>
```

6. Repeat steps 2 to 5 for all nodes where HDFS datanodes are active.
7. As a root user, log in to the node where MIT Kerberos is installed, then create a user principal for HDFS:

```
$kadmin.local
$addprinc hdfs
```

Setting Up Windows Environment to Configure Kerberos KDC

The steps provided in this section have been verified on the Windows 2016 server.

To set up your Windows environment to configure Kerberos KDC, do the following:

1. If you have not deployed the Active Directory Domain Controller in your environment, then deploy a Windows server and promote the server as the Active Directory Domain Controller. Refer to the Microsoft documentation to perform this activity.
2. If you have deployed the Active Directory Domain Controller and Intelligence in the same domain, proceed to step 4.
3. If you have deployed the Active Directory Domain Controller and Intelligence in different domains, add the Active Directory Domain Controller DNS entry in the Kubernetes environment:
   a. Log in to the node in the CDF cluster as a root user and run the following command to edit the `DNS-hosts-configmap` file:

```
kubectl edit cm dns-hosts-configmap -n kube-system
```

   Your terminal looks as follows:

```
apiVersion: v1
data:
  dns-hosts-key: ""
kind: ConfigMap
metadata:
  creationTimestamp: 2018-10-19T05:28:05Z
  name: dns-hosts-configmap
  namespace: kube-system
```

   b. Update the DNS entries and save the file. This change will take effect in 20 seconds automatically.

   For example, add the following DNS entries:

```
dns-hosts-key: |
 192.0.2.0 myhost.mydomain.com
 192.0.2.1 myhost.mydomain2.com
```

c. Your terminal looks as follows:

```
apiVersion: v1
 data:
  dns-hosts-key: |
   192.0.2.0 myhost.mydomain.com
   192.0.2.1 myhost.mydomain.com
 kind: ConfigMap
 metadata:
   creationTimestamp: 2018-10-19T05:28:05Z
```

4. (Recommended) Perform the following steps to ensure that you select strong encryption algorithm types for Kerberos in the Active Directory Domain controller:
   a. In **Local Group Policy Editor**, navigate to the following location:

      Local Computer Policy > **Computer Configuration** > **Windows Settings** > **Security Settings** > **Local Policies** > **Security Options**.
   b. Select **Network Security: Configure encryption types allowed for Kerberos**.
   c. Right-click **Network Security: Configure encryption types allowed for Kerberos** and click **Properties**.
   d. In the pop-up window, under the **Local Security Setting** tab, select the following check boxes:
      - AES128_HMAC_SHA1
      - AES256_HMAC_SHA1
      - Future encryption types
   e. Click **Apply** and then click **OK**.
   f. Launch the command prompt in the Active Directory Domain Controller and execute the following command to update the global policy:

      ```
      /gpupdate
      ```

Creating Service Principals for Kerberos Ticket Generation in Windows

To create service principals for Kerberos ticket generation:

1. Create a service principal account for **HDFS** in the Windows Active Directory domain controller:
   a. Click **Active Directory Users and Computers** > **domain name (example: intelligence.lab)** > right-click **Users** > **New** > **User**.
   b. In **New Object - User**, specify your first, last, and full name.
   c. Specify **User logon name** as `hdfs/<DATANODE_HOST>` and click **Next**.

d. Specify and confirm your password. Ensure that you select **Password Never Expires** and click **Next**.

e. Click **Active Directory Users and Computers** > **domain name (example: intelligence.lab)** > right-click **Users (The user created in the above steps)** > **Properties**.

f. Click **Account**, and under **Account Options:**, select all of the following:

- This account supports Kerberos AES 128 bit encryption.
- This account supports Kerberos AES 256 bit encryption.
- Do not require Kerberos preauthentication.

g. Click **Apply** and then click **OK**.

2. Create a service principal account for **HTTP** in the Windows Active Directory domain controller:

a. Click **Active Directory Users and Computers** > **domain name (example: intelligence.lab)** > right-click **Users** > **New** > **User**.

b. In **New Object - User**, specify your first, last, and full name.

c. Specify **User logon name** as `http/<DATANODE_HOST>` and click **Next**.

d. Specify and confirm your password. Ensure that you select **Password Never Expires** and click **Next**.

e. Click **Active Directory Users and Computers** > **domain name (example: intelligence.lab)** > right-click **Users (The user created in the above steps)** > **Properties**.

f. Click Account, and under **Account Options**, select all of the following:

- This account supports Kerberos AES 128 bit encryption.
- This account supports Kerberos AES 256 bit encryption.
- Do not require Kerberos preauthentication.

g. Click **Apply** and then click **OK**.

3. Repeat steps 1 and 2 for all the worker nodes where HDFS datanodes are active.

4. For the service principal account created for **HDFS**, generate the keytabs by running the following commands in the Windows command prompt:

```
ktpass /out hdfs_<DATANODE_HOST>.keytab /princ hdfs/<DATANODE_
HOST>@<Domain name of domain controller> /mapuser <DATANODE_HOST without
domain name>@<Domain name of domain controller> /pass <password> /crypto
all /ptype KRB5_NT_PRINCIPAL
```

5. For the service principal account created for **HTTP**, generate the keytabs by running the following commands in the Windows command prompt:

```
ktpass /out http_<DATANODE_HOST>.keytab /princ http/<DATANODE_
HOST>@<Domain name of domain controller> /mapuser <DATANODE_HOST without
domain name>@<Domain name of domain controller> /pass <password> /crypto
all /ptype KRB5_NT_PRINCIPAL
```

6. Repeat steps 4 and 5 for all the worker nodes where HDFS datanodes are active.

Configuring HDFS Services to Use Keytabs

To configure HDFS services to use keytabs:

1. **For Datanode**
   a. Launch a terminal session and log in to the Kubernetes worker node where the HDFS datanode is active.
   b. Copy the `http<DATANODE_HOST>.keytab` and `hdfs<DATANODE_HOST>.keytab`keytab files from the Windows Active Directory domain controller and paste them in the `/opt/arcsight/k8s-hostpath-volume/interset/hdfs/keytabs` directory of the Kubernetes worker node where the HDFS datanode is active, then rename them as `http.keytab` and `hdfs.keytab`.
   c. Repeat **step a** and **step b** for all the HDFS datanodes that are active in the Kubernetes cluster.
   d. For all the keytab files present in the HDFS datanodes of the Kubernetes cluster, provide the permissions of the users who have privilege to NFS, then navigate to the `/opt/arcsight/k8s-hostpath-volume/interset/hdfs/keytabs` directory and set:

   ```
   chmod 600 *
   chown UID:GID *
   ```

   For example:

   ```
   chmod 600 hdfs.keytab
   chown 1999:1999 hdfs.keytab
   ```

2. **For Namenode**
   a. Launch a terminal session and log in to the Kubernetes node where NFS is created.
   b. Copy the `http<DATANODE_HOST>.keytab` and `hdfs<DATANODE_HOST>.keytab`keytab files from the Windows Active Directory domain controller and paste them in the `/opt/arcsight-nfs/arcsight-volume/interset/hdfs/namenode/keytabs` directory of the Kubernetes node where NFS is created, then rename them as `http.keytab` and `hdfs.keytab`:

> You must generate the above keytab files for the Kubernetes worker node labeled as `intelligence-namenode:yes`.

c. Repeat **step a** and **step b** for all the namenodes active in the Kubernetes cluster.

d. For all the keytab files present in the HDFS datanodes of the Kubernetes cluster, provide the permissions of the users who have privilege to NFS, then navigate to the `/opt/arcsight/k8s-hostpath-volume/interset/hdfs/keytabs` directory and set:

```
chmod 600 *
chown UID:GID *
```

For example:

```
chmod 600 hdfs.keytab
chown 1999:1999 hdfs.keytab
```

Configuring HDFS Security in CDF

This section provides steps to configure or reconfigure HDFS security in the CDF.

> You need not perform this procedure if you already enabled Kerberos Authentication at the time of deploying Intelligence and do not intend to modify the Kerberos details.
> Perform this procedure for any of the following scenarios:
>
> - If you are enabling Kerberos Authentication for the first time.
> - If you need to modify the Kerberos details in the Intelligence tab. In this case, ensure that you first enable and configure Kerberos Authentication with the new Kerberos details before proceeding with this procedure.

To configure or reconfigure HDFS security in CDF:

1. Open a certified web browser.
2. Specify the following URL to log in to the CDF Management Portal: https://<cdf_masternode_hostname or virtual_ip hostname>:5443.
3. Select **Deployment** > **Deployments**.
4. Click ... (Browse) on the far right and choose **Reconfigure**. A new screen will be opened in a separate tab.
5. Click **Intelligence** and specify details under the **Hadoop File System (HDFS) Security** section.

> The Kerberos details that you provide in **Kerberos Domain Controller Server**, **Kerberos Domain Controller Admin Server**, **Kerberos Domain Controller Domain**, and **Default Kerberos Domain Controller Realm** will be considered only if you select **kerberos** in **Enable Authentication with HDFS Cluster**. They are not valid if you select **simple**.

> If you are enabling Kerberos Authentication, then you must enable **Enable Secure Data Transfer with HDFS Cluster**.
> If you disable **Enable Secure Data Transfer with HDFS Cluster**, the database and HDFS will use the same communication standard as Intelligence 6.2.

> **Enable Secure Data Transfer with HDFS Cluster** is disabled by default. If you enable **Enable Secure Data Transfer with HDFS Cluster**, the run time of analytics jobs will increase.
> If you have enabled **Enable Secure Data Transfer with HDFS Cluster**and if you have a non-collocated database cluster, log in to a database node, and copy the RE CA certificate from the CDF master node to /etc/pki/ca-trust/extracted/pem/tls-ca-bundle.pem. Repeat this step on all the database nodes.

6. Click **Save**.
7. The following containers restart:
   - `interset-analytics-xxxxx-xxx`
   - `hdfs-namenode-x`
   - `hdfs-datanode-xxx`
8. (Conditional) If you have modified the value of **Enable Secure Data Transfer with HDFS Cluster**, and if HDFS namenode enters the safe mode when you run analytics, perform the following steps:
   a. Do the following to bring the HDFS namenode up:
      i. Launch a terminal session and log in to the NFS server.
      ii. Navigate to the directory where NFS is created.

      (Conditional) If you have used the ArcSight Platform Installer, navigate to the following NFS directory:

      ```
      /opt/arcsight-nfs/arcsight-volume/interset/hdfs/namenode
      ```

      (Conditional) If you have used the manual deployment method, navigate to the following NFS directory:

      ```
      /<arcsight_nfs_vol_path>/interset/hdfs/namenode
      ```

      for example:

```
/opt/arcsight/nfs/volumes/itom/arcsight/interset/hdfs/namenode
```

iii. Delete the name folder under the namenode directory.

b. Do the following to bring the HDFS datanodes up:

i. Navigate to the following directory:

(Conditional) If you have used the ArcSight Platform Installer, navigate to the following directory:

```
/opt/arcsight/k8s-hostpath-volume/interset/hdfs
```

(Conditional) If you have used the manual deployment method, navigate to the following directory:

```
<arcsight_k8s-hostpath-volume>/interset/hdfs
```

ii. Delete the data folder under the hdfs directory.

iii. Repeat steps i and ii on all the datanodes.

c. Restart the HDFS datanode and namenode containers.

Setting an Encoding Option for the URL

For better data security, Intelligence provides options to encode the Intelligence URL string. Based on your requirement, you can set the limit for the URL string length and then select a preferred URL encoding option.

The supported URL encoding options are:

- **plain**: Does not encode and/or compress the URL string.
- **base64**: Compresses the URL string with **zLib** and encodes the string to **base64**.
- **hash**: Stores the encoded **base64** URL string as **JSON** in **localstorage**. Then, it uses a **hash** of the **base64** encoded URL string as the key values.
- **limitLength**: The URL string uses the **plain** and then **base64** encoding options if either of the encoding options have lesser characters than **urlLimit**. The URL string uses the **hash** encoding option if both the **plain** and **base64** encoding options are above **urlLimit**.

  urlLimit is an integer, which sets the maximum URL length (in characters) for encoding options before using **localstorage**. **urlLimit** is only available for the **limitLength** and **limitAndObscure** encoding options.

- **limitAndObscure**: The URL string uses the **base64** encoding option if it has lesser characters than **urlLimit**. The URL uses the **hash** encoding option if it has more characters than **urlLimit**.

If you do not specify an encoding option, the default encoding option is set as plain.

To set an encoding option for the URL string:

1. Login to the Management portal as the administrator.
   `https://<virtual_FQDN>:5443`
2. Click **CLUSTER** > **Dashboard**. You will be redirected to the **Kubernetes Dashboard**.
3. Under **Namespace**, search and select the `arcsight-installer-xxxx` namespace.
4. Under **Config and Storage**, click **Config Maps**.
5. Click the filter icon, and search for `investigator-default-yaml`.
6. Click  and select **Edit**.
7. In the **YAML** tab, specify the preferred URL encoding option in **`urlEncoding`** and the preferred URL string length limit in **`urlLimit`**.
8. Click **Update**.
9. Restart the `interset-api` pods:
   a. Launch a terminal session and log in to the master or worker node.
   b. Execute the following command to retrieve the namespace:
      ```
      export NS=$(kubectl get namespaces | grep arcsight|cut -d ' ' -f1)
      ```
   c. Execute the following commands to restart the `interset-api` pods:
      ```
      kubectl -n $NS scale deployment interset-api --replicas=0
      kubectl -n $NS scale deployment interset-api --replicas=2
      ```

Intelligence Data Types and Schemas

This section provides detailed information about each data type for Intelligence and how it is used in Intelligence Analytics. For each data type, the following information is included:

- A description of the data type
- The supported SmartConnectors for that type
- The schema (with the mandatory columns identified)

Access

Access data sources: sh (Fileshare), rs (Resource)

The Access schema represents events collected from Identity and Access Management (IAM) solutions where users access resources such as servers or fileshares.

Examples of access events include:

- A user fails to access a network share object `VPM-CFDB01.data.int`
- A user attempts to access shared drive `Network Shares/HR/HR-Policies/`

Examples of IAM products include: Active Directory

The Intelligence Access data type best supports Windows Security Log (or Active Directory) event data.

The Microsoft Windows Security Log contains records of login/logout activity, as well as other security-related events specified in the system's Audit Policy. A System Administrator must enable the Windows Audit feature to allow events to be recorded in the Security Log.

Supported SmartConnectors

The following SmartConnectors are used for the collection and ingestion of Access events:

- SmartConnector for Microsoft Windows Event Log – Native Application and System Event Support
- SmartConnector for Microsoft Windows Event Log – Unified Application and System Event Support

Access Schema

The following table describes the default_secops_adm.Events table columns for Access data.

Column Name	Data Type	Required (Y/N)	Description	Example
deviceReceiptTime	Integer	Y	The time at which the event related to the activity was received.	1592839336200 Equivalent GMT - 2020-06-22 15:22:00
destinatonUserName	Varchar	Y	The user involved in authentication.	john.legget
destinationHostName	Varchar	N	The server handling the authentication.	
filePath	Varchar	N	Path, project, or tag that the resource belongs to.	
fileType	Varchar	N	Type of collection that the resource belongs to, for example, shr	

Column Name	Data Type	Required (Y/N)	Description	Example
fileName	Varchar	N	File, ID, or Object that the resource is mapped to.	
externalId	Varchar	N	Usually a Windows event code (for example, **5140**, **4664**,and so on), but Analytics can be configured to accept other values, including **-1**.	4663
categoryOutcome	Varchar	N	An indicator of whether the authentication was successful. Usually either **success** or **failure** , however, Analytics can be configured to accept other values.	failure

Active Directory

Active Directory data sources: ad

The Active Directory schema represents events collected from Identity and Access Management (IAM) solutions that identify successful and failed logins to authentication targets. These authentication targets include domain controllers/servers, resources, and file shares.

Examples of authentication events include:

- A user fails to log in to YOURDC.yourcompany.com
- A user attempts to access shared drive DEV_102_share

Examples of IAM products include:

- Active Directory

The Intelligence Authentication data type best supports Windows Security Log (or Active Directory) event data.

The Microsoft Windows Security Log contains records of login/logout activity, as well as other security-related events specified in the system's Audit Policy. A System Administrator must enable the Windows Audit feature to allow events to be recorded in the Security Log.

Supported SmartConnectors

The SmartConnector for Microsoft Active Directory Windows Event Log Native is used for the collection and ingestion of Active Directory data.

Active Directory Schema

The following table describes the default_secops_adm.Events table columns for Active Directory data.

Column Name	Data Type	Required (Y/N)	Description	Example
destinationUserName	Varchar	Y	The user involved in authentication. Primary entity for ad data source.	john.legget
categoryOutcome	Varchar	Y	The outcome of the event. One of success or failure.	success
destinationHostName	Varchar	Y	The target involved in the authentication. Typically the domain controller to which the user is authenticating. The secondary entity for an ad data source.	CONTROLLER3.interset.com
externalId	Varchar	Y	Usually a Windows event code (e.g., 4624, 4771, etc.), but Analytics can be configured to accept other values, including -1.	4624
deviceReceiptTime	Integer	Y	The time at which the event related to the activity was received.	1592839336200 Equivalent GMT -2020-06-22 15:22:00
destinationNTDomain	Varchar	N	The domain that contains the user that is affected by the event.	interset
categoryObject	Varchar	N	The type of the object.	/Host/Operating System
categoryBehavior	Varchar	N	The action or behavior associated with the event.	Authentication/Verify
deviceCustomString4	Varchar	N	The string that further explains why the user failed to authenticate. Usually a hexadecimal code, but can be any string.	0xc0000064
sourceGeoLocationInfo	Varchar	N	Combination of the latitude and longitude values separated by a comma.	45.1234, -74.4321

VPN

VPN data source: vpn

The VPN schema represents events collected from Identity and Access Management (IAM) solutions or from other VPN devices such as Pulse Secure that identify VPN events.

Examples of VPN events include:

- A Network Policy Server granted full access to a user
- A user failed to authenticate with a Network Policy Server

Examples of IAM products include:

- Active Directory

The Intelligence Authentication data type best supports Windows Security Log (or Active Directory) event data. It also supports login success and failure event data from the supported VPN devices.

The Microsoft Windows Security Log contains records of login/logout activity, as well as other securityrelated events specified in the system's Audit Policy. A System Administrator must enable the Windows Audit feature to allow events to be recorded in the Security Log.

Supported SmartConnectors

The following SmartConnectors are used for the collection and ingestion of VPN data:

- SmartConnector for Microsoft Network Policy Server File
- SmartConnector for Pulse Secure Pulse Connect Secure Syslog
- SmartConnector for Citrix NetScaler Syslog
- SmartConnector for Nortel Contivity Switch Syslog

VPN Schema

The following table describes the default_secops_adm. Events table columns for VPN data.

Column Name	Type	Required (Y/N)	Description	Example
deviceReceiptTime	Integer	Y	The time at which the event related to the activity was received.	1592839336200 Equivalent GMT - 2020-06-22 15:22:00
sourceUserName	Varchar	Y	The user involved in authentication for Citrix NetScaler device. Primary entity for vpn data source.	john.legget
destinationUserName	Varchar	Y	The user involved in authentication. Primary entity for **vpn** data source.	john.legget
sourceAddressBin	Binary	N Exception: required for IPbased VPN models.	The IP address of the VPN user. Secondary entity	172.1.193.87
sourceGeoCountryCode	Varchar	N Exception: required for countrybased VPN models.	The country the user is authenticating from. Secondary entity	Canada
sourceGeoLatitude	Float	N	The latitude where the VPN connection is initiated.	45.1234
sourceGeoLongitude	Float	N	The longitude where the VPN connection is initiated.	-74.4321
externalId	Varchar	Y	Unique code assigned to a Network Policy Server events. Typically a Windows event code or -1. Analytics can be configured to accept other values.	6272
deviceEventClassId	Varchar	Y	Unique code assigned to a Pulse Secure or Citrix NetScaler event.	AUT24326
deviceAction	Varchar	Y	Unique code assigned to a Nortel event.	OK

Column Name	Type	Required (Y/N)	Description	Example
categoryOutcome	Varchar	Y	The outcome of the event. One of success or failure. For Citrix NetScaler, the outcome is attempt.	success
categoryBehavior	Varchar	Y	The action or behavior associated with the event.	/Authentication/Verify
categoryDeviceGroup	Varchar	Y	The type of events for the device. It is used for Pulse Secure, Citrix NetScaler, and Nortel events.	/VPN
categoryDeviceType	Varchar	Y	The events generated by a device type irrespective of the device group the events belong to. It is used for Citrix NetScaler and Nortel events.	VPN for Nortel Network-based IDS/IPC for Citrix NetScaler
deviceCustomString4	Varchar	N	The string that further explains why the user failed to authenticate. Usually a hexadecimal code, but can be any string. It is used for NPS events with externalId 6273.	18

Web Proxy

Web Proxy data source: pxy

Web Proxy data are raw events that capture network traffic, primarily Web surfing, from a collection of human users.

Examples

- A user accessed the Web site **https://yourcompany.com**
- A user received data from a web destination, **vap3iad3.lijit.com**

Examples of Web Proxy products include:

- Microsoft Internet Security and Acceleration Server (ISA)
- Squid
- Blue Coat Secure Web Gateway

Supported SmartConnectors

The following SmartConnectors are used for the collection and ingestion of Web Proxy data:

- SmartConnector for Microsoft Forefront Threat Management Gateway File
- SmartConnector for Squid Web Proxy Server File
- SmartConnector for Blue Coat Proxy SG Multiple Server File

Web Proxy Schema

The following table describes the default_secops_adm. Events table columns for Web Proxy data.

Column Name	Data Type	Required (Y/N)	Description	Example
deviceReceiptTime	Integer	Y	The time at which the event related to the activity was received.	1592839336200 Equivalent GMT -2020-06- 22 15:22:00
requestMethod	Varchar	Y	The HTTP method of the request.	GET
deviceSeverity	Varchar	Y	The HTTP response status.	400
bytesIn	Integer	Y	Bytes returned to the client in the response.	410235
sourceUserName	Varchar	N	The name associated with the client making the request.	john.legget
destinationHostName	Varchar	N	The host name of the machine the client is trying to connect to.	a-0001.a-msedge.net
bytesOut	Integer	N	The number of bytes the client sent in its request.	690235
requestClientApplication	Varchar	N	The agent string of the Blue Coat devices.	Mozilla/5.0 (Windows NT 5.1; rv:8.0) Gecko/20100101 Firefox/8.0
deviceCustomString1	Varchar	N	The agent string of the Microsoft devices.	Windows Update Agent
deviceVendor	Varchar	N	The device vendor of the client.	Microsoft
deviceProduct	Varchar	N	The device product of the client.	ISA Server

Repository

Repository data source: rp

Repository data are raw events collected from a source control (repository) system.

Examples:

- A user fetched files from a directory **/project_files/linux/tools/**
- A user added files to a directory **/depot/project5/java_source/**

Information in this section pertain to the following repository systems and their versions:

Repository System	Version
GitHub Enterprise	2.21.0
Bitbucket Server	7.5.0
Perforce	2020.1

The repository systems store audit information in log files. The ArcSight FlexConnectors are installed and configured on the repository systems where they read the log files, filter the messages, tokenise them, then populate them in the default_secops_adm.Events table. For each of the repository systems and the specified versions, there is a corresponding configuration file (also referred to as a parser). The configuration file is a text file containing properties (name, value pairs) that describe how the FlexConnector parses event data.

The FlexConnector type that is used to process and parse the repository log files is the ArcSight FlexConnector Regex File.

Configuration Files

The configuration files provided in this section are designed only for the specified versions of the repository systems.

Configuration File for GitHub Enterprise 2.21.0

The configuration file that is used for GitHub Enterprise 2.21.0 is **git.sdkrfilereader.properties**.

```
text.qualifier="
comments.start.with=\#
trim.tokens=true
contains.empty.tokens=true

line.include.regex=(.+)"committer_date":"([^ ]+)(.+)"hostname":"([^,]+)"
(.+)"program":("upload-pack"|"run-hook-postreceive")(.+)"
real_ip":"([^,]+)"(.+)"repo_name":"([^,]+)"(.+)"user_login":"([^,]+)"(.+)
regex=(.+)"committer_date":"([^ ]+)(.+)"hostname":"([^,]+)"(.+)"program":"
([^,]+)"(.+)"real_ip":"([^,]+)"(.+)"repo_
name":"([^,]+)"(.+)"user_login":"([^,]+)"(.+)
token.count=13

token[0].name=CONSTANT1
token[0].type=String
token[1].name=EVENTTIME
```

```
token[1].type=Long
token[2].name=CONSTANT2
token[2].type=String
token[3].name=HOSTNAME
token[3].type=String
token[4].name=CONSTANT2
token[4].type=String
token[5].name=PROGRAM
token[5].type=String
token[6].name=CONSTANT3
token[6].type=String
token[7].name=REALIP
token[7].type=String
token[8].name=CONSTANT4
token[8].type=String
token[9].name=REPONAME
token[9].type=String
token[10].name=CONSTANT5
token[10].type=String
token[11].name=USERNAME
token[11].type=String
token[12].name=CONSTANT6
token[12].type=String

event.deviceVendor=__getVendor("GitHub")
event.deviceProduct=__stringConstant("GitGub Enterprise")
event.deviceVersion=__stringConstant("2.21.0")

event.deviceReceiptTime=__createLocalTimeStampFromSecondsSinceEpoch
(EVENTTIME)
event.destinationUserName=USERNAME
event.deviceCustomString1=__toLowerCase(REPONAME)
event.deviceCustomString1Label=__stringConstant("RepositoryName")
event.deviceAction=__ifThenElse(PROGRAM,"run-hook-post-receive","receive-
pack","upload-pack")
event.sourceAddress=__oneOfAddress(REALIP)
event.destinationHostName=__oneOfHostName(HOSTNAME)
event.name=__ifThenElse(PROGRAM,"run-hook-post-receive","receive-
pack","upload-pack")
event.bytesOut=__safeToInteger(__regexToken(CONSTANT5,".+uploaded_bytes.:
([^,]+)"))
#event.requestMethod=
#event.protocol=
#event.request=

event.categoryObject=__stringConstant("/Host/Resource")
event.categoryBehavior=__stringConstant("/Access")
event.categoryOutcome=__stringConstant("/Attempt")
```

```
event.categorySignificance=__stringConstant("/Informational")
event.categoryDeviceGroup=__stringConstant("Application")
event.categoryDeviceType=__stringConstant("Repository")
```

Configuration File for Bitbucket Server 7.5.0

The configuration file that is used for Bitbucket Server 7.5.0 is **bitbucket.sdkrfilereader.properties**.

```
text.qualifier="
comments.start.with=\#
trim.tokens=true
contains.empty.tokens=true

line.include.regex=(.+)\\|(.+)\\|(.+)\\|([^-]+)\\|(.+)\\|(.+git-upload-
pack.+|.+git-receive-pack.+)\\|(.+)\\|(.+)\\|
(.+)\\|(.+)\\|(.+)\\|(.+)\\|(.+)\\|(.*)
regex=(.+)\\|(.+)\\|(.+)\\|(.*)\\|(.+)\\|(.*)\\|(.+)\\|(.+)\\|(.+)\\|
(.+)\\|(.+)\\|(.+)\\|(.+)\\|(.*)

token.count=14

token[0].name=REALIP
token[0].type=String
token[1].name=PROTOCOL
token[1].type=String
token[2].name=REQUESTID
token[2].type=String
token[3].name=USERNAME
token[3].type=String
token[4].name=EVENTTIME
token[4].type=String
token[5].name=ACTION
token[5].type=String
token[6].name=REQUESTINFO
token[6].type=String
token[7].name=STATUS
token[7].type=String
token[8].name=BYTESREAD
token[8].type=String
token[9].name=BYTESWROTE
token[9].type=String
token[10].name=EXTRAINFO1
token[10].type=String
token[11].name=EXTRAINF02
token[11].type=String
token[12].name=EXTRAINF03
```

```
token[12].type=String
token[13].name=EXTRAINF04
token[13].type=String

event.deviceVendor=__getVendor("BitBucket")
event.deviceProduct=__stringConstant("BitBuket Server")
event.deviceVersion=__stringConstant("7.5.0")

event.deviceReceiptTime=__createOptionalTimeStampFromString
(EVENTTIME,"yyyy-MM-dd HH:mm:ss,sss")
event.destinationUserName=USERNAME
event.deviceCustomString1=__toLowerCase(__regexToken(__regexToken(__split
(ACTION," ",2),"(.*)\\.git(.+)"),".*\/(.+)"))
event.deviceCustomString2=__regexToken(__split(ACTION," ",2),"(\/.+)(\/git-
upload-pack|\/git-receive-pack)")
event.deviceCustomString2Label=__stringConstant("RepositoryName")
event.name=__regexToken(__split(ACTION," ",2),".+\/(.+)")
event.sourceAddress=__oneOfAddress(REALIP)
event.sourceHostName=__oneOfHostName(REALIP)
event.deviceAction=__regexToken(__split(ACTION," ",2),".+\/(.+)")
event.bytesIn=__safeToInteger(BYTESREAD)
event.bytesOut=__safeToInteger(BYTESWROTE)
event.requestMethod=__ifThenElse(__contains
(ACTION,"POST"),"true","POST","GET")
event.requestUrl=__split(ACTION," ",2)

event.categoryObject=__stringConstant("/Host/Resource")
event.categoryBehavior=__stringConstant("/Access")
event.categoryOutcome=__ifThenElse(STATUS,"200","/Success",__ifThenElse
(STATUS,"401","/Denied","/Attempt"))
event.categorySignificance=__stringConstant("/Informational")
event.categoryDeviceGroup=__stringConstant("Application")
event.categoryDeviceType=__stringConstant("Repository")
```

Configuration File for Perforce 2020.1

The configuration file that is used for Perforce 2020.1 is **perforce.sdkrfilereader.properties**.

```
text.qualifier="
comments.start.with=\#
trim.tokens=true
contains.empty.tokens=true

regex=(.+)\\s(.+)\\s(.+)\\s(.+)\\s(.+)\\s(.+)

token.count=6
```

```
token[0].name=EVENTDATE
token[0].type=String
token[1].name=EVENTTIME
token[1].type=String
token[2].name=USER
token[2].type=String
token[3].name=CLIENTIP
token[3].type=String
token[4].name=ACTION
token[4].type=String
token[5].name=RESOURCE
token[5].type=String

event.deviceVendor=__getVendor("Perforce")
event.deviceProduct=__stringConstant("Perforce")
event.deviceVersion=__stringConstant("2020.1")

event.deviceReceiptTime=__createOptionalTimeStampFromString(__concatenate
(EVENTDATE,EVENTTIME),"yyyy/MM/ddHH:mm:ss")
event.destinationUserName=USER

##########################################################################
#1.\/\/([^\/]+)\/([^\/]+)\/([^\/]+).*","/","//",""")
# will return max of depth 4
# __regexTokenFindAndJoin(RESOURCE,"\/\/([^\/]+)?\/?([^\/]+)?\/?
([^\/]+)?","/","//","")
# eg //csvr/A/B/C
# //csrv/main/null
# //csrv/null/null
# //csrv/A/master
#2.\/\/(.*)(?=\/main$|\/null$|\/rel$|\/master$)
#__regexToken(__regexTokenFindAndJoin(RESOURCE,"\/\/([^\/]+)?\/?
([^\/]+)?\/?([^\/]+)?","/","//",""),"\/\/(.*)
(?=\/main$|\/null$|\/rel$|\/master$)")
#eg.returns all info nothign with main/null/rel/master
#3. remove version if any
#__regexToken(__ifGreaterOrEqual(__length(__regexToken(__
regexTokenFindAndJoin(RESOURCE,"\/\/([^\/]+)?\/?([^\/]+)?\/?
([^\/]+)?","/","//",""),"\/\/(.*)
(?=\/main$|\/null$|\/rel$|\/master$)")),"1",__regexToken(__
regexTokenFindAndJoin
(RESOURCE,"\/\/([^\/]+)?\/?([^\/]+)?\/?([^\/]+)?","/","//",""),"\/\/(.*)
(?=\/main$|\/null$|\/rel$|\/master$)"),__
regexTokenFindAndJoin(RESOURCE,"\/\/([^\/]+)?\/?([^\/]+)?\/?
([^\/]+)?","/","//","")),"(.*)[#\/][\\d.]+")
#eg.//crsv/A/12.3
# //crsv/A#1.2
##########################################################################
```

```
event.deviceCustomString1=__ifGreaterOrEqual(__length(__regexToken(__
ifGreaterOrEqual(__length(__regexToken(__
regexTokenFindAndJoin(RESOURCE,"\/\/([^\/]+)?\/?([^\/]+)?\/?
([^\/]+)?","/","//",""),"(\/\/.*)
(?=\/main$|\/null$|\/rel$|\/master$)")),"1",__regexToken(__
regexTokenFindAndJoin(RESOURCE,"\/\/([^\/]+)?\/?([^\/]+)?\/?
([^\/]+)?","/","//",""),"(\/\/.*)(?=\/main$|\/null$|\/rel$|\/master$)"),__
regexTokenFindAndJoin(RESOURCE,"\/\/
([^\/]+)?\/?([^\/]+)?\/?([^\/]+)?","/","//","")),"(.*)[#\/][\\d.]+")),"1",_
_regexToken(__ifGreaterOrEqual(__length(__
regexToken(__regexTokenFindAndJoin(RESOURCE,"\/\/([^\/]+)?\/?([^\/]+)?\/?
([^\/]+)?","/","//",""),"(\/\/.*)
(?=\/main$|\/null$|\/rel$|\/master$)")),"1",__regexToken(__
regexTokenFindAndJoin(RESOURCE,"\/\/([^\/]+)?\/?([^\/]+)?\/?
([^\/]+)?","/","//",""),"(\/\/.*)(?=\/main$|\/null$|\/rel$|\/master$)"),__
regexTokenFindAndJoin(RESOURCE,"\/\/
([^\/]+)?\/?([^\/]+)?\/?([^\/]+)?","/","//","")),"(.*)[#\/][\\d.]+"),__
ifGreaterOrEqual(__length(__regexToken(__
regexTokenFindAndJoin(RESOURCE,"\/\/([^\/]+)?\/?([^\/]+)?\/?
([^\/]+)?","/","//",""),"(\/\/.*)
(?=\/main$|\/null$|\/rel$|\/master$)")),"1",__regexToken(__
regexTokenFindAndJoin(RESOURCE,"\/\/([^\/]+)?\/?([^\/]+)?\/?
([^\/]+)?","/","//",""),"(\/\/.*)(?=\/main$|\/null$|\/rel$|\/master$)"),__
regexTokenFindAndJoin(RESOURCE,"\/\/
([^\/]+)?\/?([^\/]+)?\/?([^\/]+)?","/","//","")))
event.deviceCustomString2=RESOURCE
event.deviceAction=ACTION
event.sourceAddress=__oneOfAddress(CLIENTIP)
event.sourceHostName=__oneOfHostName(CLIENTIP)
event.name=ACTION

event.categoryObject=__stringConstant("Host/Resource")
event.categoryBehavior=__stringConstant("Access")
event.categoryOutcome=__stringConstant("/Attempt")
event.categorySignificance=__stringConstant("/Informational")
event.categoryDeviceGroup=__stringConstant("Application")
event.categoryDeviceType=__stringConstant("Repository")
```

You can also create or customize the configuration files for other versions of the repository systems. For more information, see ArcSight FlexConnector Developer's Guide.

FlexConnector Installation and Configuration

To install and configure a FlexConnector, see ArcSight FlexConnector Developer's Guide.

Ensure the following when you install and configure the FlexConnector:

- Select **ArcSight FlexConnector Regex** File as the **Connector Type**.
- When adding the parameters information, specify the following:
  - Select **Log Unparsed Events** as **False**.
  - Provide the absolute path and the repository log file name that the FlexConnector needs to read in the **Log File Name** field.

    For example:
    c:\temp\sample_data.log
  - For the **Configuration File** field, depending on the repository on which you are installing the FlexConnector, specify only **git**, **bitbucket**, or **perforce**.

    For example, for the GitHub Enterprise repository, you must specify only git. The suffix **.sdkrfilereader.properties** is appended automatically. The configuration file name now is **git.sdkrfilereader.properties**.
- When configuring the destination, select either **CEF File** or Transformation Hub as the destination. For more information, see SmartConnector Installation and User Guide.

Post-Installation Tasks

After you install and configure the FlexConnector and before you run the FlexConnector, copy the desired configuration (parser) files in the **ARCSIGHT_HOME\user\agent\ flexagent** location.

Repository Schema

The following table describes the default_secops_adm.Events table columns for Repository data.

Column Name	Type	Required (Y/N)	Description	Example
deviceAction	Varchar	Y	The action performed on the device.	upload-pack
deviceCustomString1	Varchar	Y	The device involved in the event. Typically a file path. Can be any string identifying a repository. Secondary entity for the rp data source	dev3/rel/hydra
deviceReceiptTime	Integer	Y	The time at which the event related to the activity was received.	1592839336200 Equivalent GMT - 2020-06-22 15:22:00
destinationUserName	Varchar	Y	The user involved in the event. Primary entity.	john.legget
deviceVendor	Varchar	Y	The device vendor of the client.	GitHub

Column Name	Type	Required (Y/N)	Description	Example
deviceProduct	Varchar	N	The device product of the client.	GitHub Server
deviceVersion	Integer	N	The device version.	2.21.0
categoryObject	Varchar	N	The type of the object.	Host/Resource
categoryBehavior	Varchar	N	The action or behavior associated with the event.	/Access
categoryOutcome	Varchar	Y	The outcome of the event.	/Attempt
cateorySignificance	Varchar	N	The significance of the event.	/Informational
categoryDeviceGroup	Varchar	Y	The type of events for the device.	Application
categoryDeviceType	Varchar	N	The events generated by the device type irrespective of the device group the events belong to.	Repository
sourceAddressBin	Varchar	N	The IP address of the user involved in the event.	78.1.198.82
bytesOut/bytesIn	Integer	N	The size of data (in bytes) related to the action performed on the project.	2203

Managing Recon

This section provides guidance for managing Recon functions and features within the deployment.

- "Making Searches Case-insensitive" below
- "Performing a Keyword Search on Raw Event Data" on the next page
- "Configuring and Tuning Event Integrity Checks" on page 769

Making Searches Case-insensitive

By default, Search queries are case-sensitive for full-text searches and field-based ones. You can modify the database to make Search insensitive to case.

As the dbadmin user in the ArcSight Database, execute the following command:

```
 – ALTER DATABASE investigate set DefaultSessionLocale = ‘en_
US@colstrength=secondary’
```

Case-insensitive searches tend to slow Search performance.

Performing a Keyword Search on Raw Event Data

Recon adds a **rawEvent** field, or a subset of event fields, to a text index for use in free-form text search. Users can perform a free-form text search for values only in event fields that are indexed.

- "Understanding Indexed Fields for Free-form Search" below
- "Indexing Event Fields Before Installing the Database" on the next page
- "Indexing Event Fields After Installing the Database" on the next page

Understanding Indexed Fields for Free-form Search

If the **rawEvent** field has a value, the database will tokenize the field's content and store it as indexed text. If the **rawEvent** field is null, search allows you to perform a full-text search on the following columns:

agentDnsDomain	deviceCustomNumber3Label	filePermission
agentHostName	deviceCustomString1	fileType
agentTranslatedZoneURI	deviceCustomString1Label	flexDate1Label
agentZoneURI	deviceCustomString2	flexString1
applicationProtocol	deviceCustomString2Label	flexString1Label
categoryDeviceGroup	deviceCustomString3	flexString2
categoryDeviceType	deviceCustomString3Label	flexString2Label
categoryObject	deviceCustomString4	message
categoryOutcome	deviceCustomString4Label	name
categorySignificance	deviceCustomString5	oldFileId
categoryTechnique	deviceCustomString5Label	oldFileName
cryptoSignature	deviceCustomString6	oldFilePath
destinationDnsDomain	deviceCustomString6Label	oldFilePermission
destinationGeoLocationInfo	deviceDnsDomain	oldFileType
destinationHostName	deviceDnsDomain	rawEvent
destinationNtDomain	deviceDomain	reason

destinationProcessName	deviceEventCategory	requestClientApplication
destinationServiceName	deviceEventClassId	requestContext
destinationTranslatedZoneURI	deviceExternalId	requestCookies
destinationUserId	deviceFacility	requestMethod
destinationUserName	deviceHostName	requestUrl
destinationUserPrivileges	deviceInboundInterface	requestUrlFileName
destinationZoneURI	deviceNtDomain	requestUrlQuery
deviceAction	deviceOutboundInterface	sourceDnsDomain
deviceAssetId	devicePayloadId	sourceGeoLocationInfo
deviceCustomDate1Label	deviceProcessName	sourceHostName
deviceCustomDate2Label	deviceProduct	sourceNtDomain
deviceCustomFloatingPoint1Label	deviceSeverity	sourceProcessName
deviceCustomFloatingPoint2Label	deviceTranslatedZoneURI	sourceServiceName
deviceCustomFloatingPoint3Label	deviceVendor	sourceTranslatedZoneURI
deviceCustomFloatingPoint4Label	deviceVendor	sourceUserId
deviceCustomIPv6Address1Label	deviceZoneURI	sourceUserName
deviceCustomIPv6Address2Label	eventOutcome	sourceUserPrivileges
deviceCustomIPv6Address3Label	externalId	sourceGeoPostalCode
deviceCustomIPv6Address4Label	fileId	sourceGeoRegionCode
deviceCustomNumber1Label	fileName	sourceZoneURI
deviceCustomNumber2Label	filePath	transportProtocol

Indexing Event Fields Before Installing the Database

Before installing the database, you can index event fields that would not otherwise be indexed when the rawEvent field is null. To do so, contact Support Services so they can assist you in modifying the `superschema_vertica.sql` file in the installer.

Indexing Event Fields After Installing the Database

After installing the database, you can index event fields that would not otherwise be indexed when the rawEvent field is null. If there are events in the database, you must drop the text

index and recreate it. The reindexing process might take time, depending on the number of events in the system.

Configuring and Tuning Event Integrity Checks

To validate that the event information in your database matches the content sent from SmartConnectors, run an **Event Integrity Check**. When you run the check, Recon searches the database for verification events received within the specified date range, then runs a series of checks to compare content in the database with information supplied by the verification event. The results of an the Event Integrity Check help you identify whether event data might be compromised. In addition to reviewing the *raw event data* received from SmartConnectors, you can enable Transformation Hub to generate more than 20 *parsed fields* to include in the check.

- "Configuring a SmartConnector to Include a Verification Event for Raw Events" below
- "Enabling Transformation Hub to Generate Verification Events for Parsed Fields" on the next page
- "Improving the Performance of Event Integrity Checks" on the next page

For more information about verification events and running integrity checks, see the Help for Recon.

Configuring a SmartConnector to Include a Verification Event for Raw Events

For a SmartConnector to support event integrity checks, you must enable it include a verification event for each batch of events. This configuration ensures that the connector generates a verification event for the **Raw Event** field in an event at the moment that your environment captures the event.

For this setting...	Enter...
Preserve Raw Event	Yes **NOTE**: When you enable this setting, the size of each event increases, which will require more storage space in your database.
Event Integrity Algorithm	SHA-256
Check Event Integrity Method	Recon

For more information about configuring SmartConnectors, see the following documentation:

- "Configuring Processing" in the *Installation Guide for ArcSight SmartConnectors*
- "Destination Runtime Parameters"

Enabling Transformation Hub to Generate Verification Events for Parsed Fields

The Event Integrity Check can verify the integrity of multiple fields within an event. You must enable Transformation Hub to generate verification events for the parsed fields received from the SmartConnectors. You can configure this setting as you deploy Transformation Hub or at any time after deployment.

1. Log in to the Management Portal.
2. Navigate to **Transformation Hub** > **Stream Processors and Routers**.
3. Enable **Generate verification events for parsed field integrity checks**.
4. For **Verification event batch size**, specify the number of events that you want to be associated with a verification event.

   A lower value indicates fewer associated events need to be included in the batch for integrity checks. However, a lower value will also result in higher resource consumption by generating more verification events.

Improving the Performance of Event Integrity Checks

If Event Integrity Checks consistently fail with notifications about insufficient resources or disk space, your system might not have enough memory or disk space to run the check. To mitigate this issue, you can adjust the event integrity settings.

In the CDF Management Portal, select **Fusion** > **Event Integrity Check Configuration**. Change the following settings as needed:

Event Integrity Task Count **– Lower the Value**

This setting specifies the number of tasks that can run in parallel during an Event Integrity Check. For load balancing, we recommend that you set the number of tasks proportional to the number of nodes in your cluster. For example, three tasks for a three-node cluster. Note that allowing more tasks increases CPU and memory usage but also enables the check to complete in a shorter amount of time.

Event Integrity Chunk Size **– Lower the Value**

This setting specifies the number of events within each data chunk that the Event Integrity Check processes at a time. To reduce the amount of memory and disk space needed to

processes data, you can change the data chunks to a smaller size. However, a smaller chunk size can cause the check to take longer to complete.

Use Event Integrity Resource Pool **– Set to TRUE**

By default, the system uses the general resource pool for event integrity queries. However, if you change this setting to **True**, the system uses the dedicated event integrity resource pool. By moving all queries for the check to a dedicated pool, event integrity queries do not affect other queries and vice versa. For additional tuning of resource pools, it allows detailed tuning furthermore. (add link for db-installer)

Managing Transformation Hub

This section provides guidance for managing Transformation Hub functions and features within the deployment.

Maintaining an On-Premises Transformation Hub

This chapter contains the following sections:

Adding a New Worker Node to an On-Premises Cluster

You can add a new worker node to an existing on-premises cluster.

To add a new worker node:

1. Set up and provision the new node according to the guidelines and system requirements given here. Note the IP address of the new node for use in the following procedures.
2. Modify your NFS server settings to add the new node to the `/etc/exports` file.
3. Run the following command to update the shared volumes:
   `exportfs -ra`
4. Log in to the CDF Management Portal (`https://<ha-address>:5443`).
5. Click **Cluster > Nodes.**
6. Click **+ Add.**
7. Enter values for the pop-up dialog. For host name, use the FQDN of the new node.
8. Click **ADD.**

Now log into the CDF Management Portal and add the appropriate labels to the new worker node.

Uninstalling a Master or Worker Node from an On-Premises Cluster

To uninstall an existing master or worker node from an on-premises cluster, open an SSH connection to the node and run the following commands.

```
cd $k8s-home
./uninstall.sh
```

Then, reboot the node to complete node removal.

When removing the node from the cluster, make sure that the cluster will still have enough resources to host the product workload without the node you are removing . Also, make sure that you have sufficient nodes labeled with the product labels.

Effects on the Cluster

If a worker node is uninstalled, all events data will be stored on the node by default under `/opt/arcsight/k8s-hostpath-volume/th/kafka`.

If a master node is stopped or uninstalled, that node will be reported as unavailable to the cluster. All other functionality, including events processing on the worker nodes, will continue.

From a multi-master cluster with 3 master nodes, you can safely remove only one master node. By removing one of three master nodes you will lose high availability, but the cluster will continue to function. If you remove two of three master nodes, the cluster might become unavailable, and you will then need to set up the cluster from scratch.

Removing a Crashed Worker Node

In case of a worker node failure, do the following:

1. Add a new worker node to replace the failed node before removing the crashed one.
2. Run the following command on one of the healthy nodes to delete the crashed node's IP address from the cluster:

```
kubectl delete node <crashed_node_ip_FQDN>
```

This action needs to be performed manually by the cluster administrator, because there is no way for the cluster to distinguish permanent node failure from temporary network connectivity outage, restart or similar events.

When this command is run, the cluster re-schedules the stateful containers (Kafka, ZooKeeper, routing stream processors) to the remaining machines matching the container requirements (labels, resources).

Adding ZooKeeper Instances

Adding a ZooKeeper instance has the following prerequisites:

- You will need at least 2 available worker nodes already deployed that do not already have a Kafka broker or ZooKeeper instance deployed on them. (Only one ZooKeeper can be installed for each worker node, and there must be an odd number of ZooKeepers --1, 3, 5, 7, and so on. Therefore you need at least 2 additional worker nodes to keep the total number an odd one.)
- Any new node where a ZooKeeper instance is deployed should be labeled `zk:yes` (for on-premises installation) or `zk=yes` (for cloud installation).
- If you plan to deploy both Kafka brokers and ZooKeepers, it is recommended that you perform this procedure to add ZooKeeper instances *before* you have deployed your Kafka brokers.

To add new ZooKeeper instances:

1. Open the CDF Management Portal.
2. Click **... (Browse)** to the right.
3. From the drop-down, select **Reconfigure**. The post-deployment settings page is displayed.
4. Find the field *# of ZooKeeper nodes in the ZooKeeper cluster.*
5. From the field's drop-down, select the new number of ZooKeeper instances.
6. Click **Save**.
7. Verify the new ZooKeeper pods are up and in Running state by running the command:
   `kubectl get pods -n {arcsight_namespace_id} th-zookeeper-x`

Reducing the number of ZooKeeper instances is not currently supported.

Adding a Kafka Broker Instance for Consistency with the Zookeeper

Adding a new Kafka broker has the following prerequisites:

- You will need an available worker node already deployed that does not already have a Kafka broker or ZooKeeper instance deployed on it.
- The new node where a Kafka broker is deployed should be labeled `kafka:yes` (on-premises installation), or `kafka=yes` (for cloud installation).
- If you plan to deploy both Kafka Brokers and ZooKeepers , it is recommended that you perform this procedure to add Kafka brokers after you have deployed your ZooKeepers and they are up and running.

To add a Kafka broker:

1. Open the CDF Management Portal.
2. Click **... (Browse)** to the right.
3. From the drop-down, select **Reconfigure**. The post-deployment settings page is displayed.
4. Find the field *# of Kafka broker nodes in the Kafka cluster.*
5. From the field's drop-down, select the new number of Kafka brokers.
6. Adjust any other related fields as needed. For example, if the topic replication factor is 1, consider increasing it.
7. Click **Save**.
8. Verify that the new Kafka brokers are up and in Running state by running the command: `kubectl get pods -n {arcsight_namespace_id} th-kafka-x`

Next, assign partitions to the new Kafka broker:

1. Connect to Transformation Hub Kafka Manager.

Refer to Connecting to the Kafka Manager for more information.

2. In **Cluster > Transformation Hub > Topics,** click **Generate Partition Assignments.**
3. On the **Confirm Assignments** page, confirm partition assignments for the new broker and click **Generate Partition Assignments.**
4. On the main toolbar, click **Topic > List.**
5. Click **Run Partition Assignments.**
6. On the **Run Assignments** page, confirm partition assignments and click **Run Partition Assignments.**
7. The partition reassignment process begins. On the Reassign Partitions page, under **Status**, check for a date and time of the job completion to verify completion of the task.

Reducing the number of Kafka brokers is not currently supported.

Managing Transformation Hub through ArcMC

After configuring ArcMC to manage your Transformation Hub, you can create topics and routing rules, monitor metrics, and receive notifications about Transformation Hub status through ArcSight Management Center (ArcMC).

Monitored Transformation Hub parameters include CPU usage, memory, event parsing errors, stream processing EPS, and stream processing lag.

To manage a Transformation Hub in ArcMC, add your Transformation Hub as a host to ArcMC. The procedure for adding Transformation Hub as a host is explained in Adding Transformation Hub as a Host to ArcMC.

A single ArcMC can manage only a single Transformation Hub cluster, while a single Transformation Hub can be managed by up to 2 ArcMCs.

Changing Transformation Hub Security Mode

You should decide on a security mode for Transformation Hub prior to deployment and setup. In general, the security mode of systems connected to Transformation Hub (consumers and producers) must be the same as the Transformation Hub security mode.

As of the release of ArcSight Platform 21.1, FIPS is the default security mode. Optional modes include TLS with Client Authentication, as well as TLS with FIPS. A TLS performance impact is a known Kafka behavior. Exact details of the impact will depend on your specific configuration, but could reduce the event rate by half or more.

To protect against unknown clients sending events to Transformation Hub, or changing Avro topic schemas, enabling Client Authentication is recommended.

You can change the Transformation Hub security mode after deployment, but this will cause downtime for your Transformation Hub and associated systems, such as consumers and producers.

You will need to make sure all Transformation Hub-associated systems are re-configured as well. If the security mode change requires that Transformation Hub consumer or Transformation Hub producer restarts, the *producer or consumer must be disconnected from Transformation Hub first.* Consult the appropriate consumer or producer documentation for details.

For an Azure cluster, please run all Azure Cloud Shell (command line) and `kubectl` commands from the authorized jump host. For an AWS cluster, all comments should be run from the bastion.

The process of changing security mode includes the following steps.

Undeploying Transformation Hub will remove all previous configuration settings. Prior to proceeding further, you should make a note of your existing settings and then re-enter these on the pre-deployment configuration page during the re-deployment of the Transformation Hub.

1. Stop SmartConnectors from sending events. This will close connections. See the SmartConnector User Guide for information on stopping SmartConnectors from sending events.

2. Stop all consumers (Logger, ESM, Database Kafka Scheduler) from consuming from topics in Transformation Hub. (There is no need to clear out existing messages from the topics, and the consumers will continue from the last offset later.)
3. Log in to the CDF Management Portal.
4. Click **Administration**.
5. Click the **...** (Browse) icon to the right of the main window.
6. From the drop-down, click **Uninstall**. The post-deployment settings page is displayed.
7. Uninstall the Transformation Hub.
8. Follow the consumer and producer documentation to reconfigure those applications to align their security modes to be the same as Transformation Hub.
9. Redeploy the Transformation Hub with the appropriate security mode configured.
10. Reconnect the consumers and producers to the Transformation Hub.

Migrating the NFS Server to a New Location

The process given here explains how to migrate your NFS server and paths to another location (including changing paths within the same NFS server). During the move, some of exported path pods from the core namespace will incur downtime as they are scaled to zero or temporarily removed. The CDF Management Portal (and all of its features) will not be available during such downtime.

Data will be moved transferred by copying first, so the original location should remain as a backup until the procedure is complete and the cluster successfully operates, with started back pods with new paths and the new NFS server.

This procedure will be executed on your primary master node, with access to the`kubectl` command and the contents of `/opt/arcsight/kubernetes`

The procedures make usage of the `volume_admin.sh` script located in `/opt/arcsight/kubernetes/scripts`

Usage: `./volume_admin.sh <Operation> <Persistent Volume> <Options>`

Where options include:

`reconfigure`: Reconfigure a persistent volume

`search`: Find persistent volume consumers

Preparation

1. Verify that all pods are running correctly with the following command:
   ```
   kubectl get pods --all-namespaces -o wide | awk -F " *|/" '($3!=$4 ||
   $5!="Running") && $5!="Completed" {print $0}'
   ```

2. Verify status of CDF installation with the following command:
   `/opt/arcsight/kubernetes/bin/kube-status.sh`
3. Prepare the new NFS volumes with the same permission set as the existing volumes.

- If you are using a software-controlled NFS, make sure the export policy is configured in the correct order. For example, for NetApp NFS, the RO/RW Access rules are None, Superuser Security types are None, User ID to which anonymous users are mapped equals 1999 (or whatever value you used during initial install).
- For using NFSv4 and later versions, make sure ID mapping (configured in (`/etc/idmapd.conf`) on both the NFS server and all NFS clients (that is, your cluster nodes) use the same domain.
- Verify that UID/GID is correct by manually mounting new NFS mount points and touching a file. Permission should be the same as for touching the file on the old NFS mount points.
- Note that for any changes on the NFS Server to take effect, all pending mounts of mount points should be closed.

4. Get an overview of persistent volumes for your installation with the following command:
`kubectl get pv`

Migration Procedures

The recommended order in which migration should be executed on your persistent volumes is as follows:

1. itom-logging
2. arcsight-installer-xxxxx-db-backup-vol
3. itom-vol
4. db-single
5. arcsight-installer-xxxxx-arcsight-volume

In any of the following commands, `<old_nfs_mount>` and `<new_nfs_mount>` refer to manually-mounted NFS for copying or maintenance procedures, and `<new_nfs_path>` refers to the real path on the NFS server of the mount point for the PV change command.

If any PV change fails, roll back any changes to the old NFS location until the issue is resolved. **Do not leave your cluster in a change-pending state.**

Migrate PV itom-logging

1. Determine the services using the itom-logging PV by running the following command. (Note the number of replicas running for later scaleback, after the NFS migration):
   `/opt/arcsight/kubernetes/scripts/volume_admin.sh search itom-logging`

Note: For fluentd, the YAML definition will include an NFSpath. You will need to mount it on a temporary mount to delete (and later to create) it with the following command:
`kubectl delete -f /<old_nfs_mount>/itom/itom_vol/suite-install/yamlContent/itom-fluentd.yaml`

2. Scale down other services by running these commands:
```
kubectl scale --replicas=0 -n core deployment/idm
kubectl scale --replicas=0 -n core deployment/itom-logrotate-deployment
```
3. Verify all pods of interest are deleted by running this command:
```
/opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide |
awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
```
4. Verify that consumers have been removed from the PV users list:
```
/opt/arcsight/kubernetes/scripts/volume_admin.sh search itom-logging
```
5. Copy NFS data to new mount point:
```
cp -rfp /mnt/<old_nfs_mount>/itom/logging /mnt/<new_nfs_
mount>/itom/logging
```
6. Check the content of mount for any permissions discrepancies. The output of these commands must be identical:
```
ls -l /mnt/<old_nfs_mount>/itom/logging
ls -l /mnt/<new_nfs_mount>/itom/logging
```
7. Authorize the PV change by running this command:
```
/opt/arcsight/kubernetes/scripts/volume_admin.sh reconfigure itom-logging
-t nfs -s <new_nfs_FQDN_or_IP> -p /<new_nfs_path>/itom/logging
```
8. Verify the new NFS path in the configuration by running the following command:
```
kubectl get pv itom-logging -o yaml
```
9. For the previous command, locate the `nfs:` section of the output. It should list the new server and volume.
10. Repeat all the commands you used to scale down or destroy the pods to scale all replicas up or start up related daemonsets.
11. Recreate the daemonset from the YAML with these commands. (Note that this will be still old path until itom_vol PV is migrated.)
```
kubectl create -f /<old_nfs_mount>/itom/itom_vol/suite-
install/yamlContent/itom-fluentd.yaml
kubectl scale --replicas=<value> -n core deployment/idm
kubectl scale --replicas=<value> -n core deployment/itom-logrotate-
deployment
```
12. Verify that consumers have been restored with this command:
```
/opt/arcsight/kubernetes/scripts/volume_admin.sh search itom-logging
```
13. Verify pods are all running:
```
/opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide |
awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
```

14. If all pods are running, verify CDF status:
    `/opt/arcsight/kubernetes/bin/kube-status.sh`

Migrate PV arcsight-installer-xxxxx-db-backup-vol

Some additional checks are omitted from this procedure, but should be run as in the procedure above, to make sure no discrepancies arise.

1. Determine the services using the arcsight-installer-xxxxx-db-backup-vol PV by running the following command. (Note the number of replicas running for later scaleback, after the NFS migration):
   `/opt/arcsight/kubernetes/scripts/volume_admin.sh search arcsight-installer-xxxxx-db-backup-vol`
2. Scale down the necessary deployments:
   `kubectl scale --replicas=0 deployment/itom-pg-backup -n arcsight-installer-xxxxx`
3. Verify that consumers have been removed:
   `/opt/arcsight/kubernetes/scripts/volume_admin.sh search arcsight-installer-xxxxx-db-backup-vol`
4. Copy the NFS data to a new mount point:
   `cp -rfp /mnt/<old_nfs_mount>/itom/db_backup /mnt/<new_nfs_mount>/itom/db_backup`
5. Check the content of mount for any permissions discrepancies. The output of these commands must be identical:
   `ls -l /mnt/<old_nfs_mount>/itom/logging`
   `ls -l /mnt/<new_nfs_mount>/itom/logging`
6. Authorize the PV change:
   `/opt/arcsight/kubernetes/scripts/volume_admin.sh reconfigure arcsight-installer-xxxxx-db-backup-vol -t nfs -s <new_nfs_FQDN_or_IP> -p /<new_nfs_path>/itom/db_backup`
7. Repeat all the commands you used to scale down or destroy the pods to scale all replicas up or start up related daemonsets.
   `kubectl scale --replicas=<value> deployment/itom-pg-backup -n arcsight-installer-xxxxx`

To restore path services, use this command:
`kubectl create -f <PATH>`

8. Verify consumers have been restored with this command:
   `/opt/arcsight/kubernetes/scripts/volume_admin.sh search arcsight-installer-xxxxx-db-backup-vol`

9. Verify pods are all running:
```
/opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide |
awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
```
10. If all pods are running, verify CDF status:
```
/opt/arcsight/kubernetes/bin/kube-status.sh
```

Migrate PV itom-vol

1. Determine the services using the itom-vol PV by running the following command. (Note the number of replicas running for later scaleback, after the NFS migration):
```
/opt/arcsight/kubernetes/scripts/volume_admin.sh search itom-vol
```
2. Delete the YAML-based daemonsets by running these commands:
```
kubectl delete -f /<old_nfs_mount>/itom/itom_vol/suite-
install/yamlContent/kube-registry.yaml
kubectl delete -f /<old_nfs_mount>/itom/itom_vol/suite-
install/yamlContent/itom-fluentd.yaml
```
3. Scale down deployments with these commands. (Note: Make sure you have noted original number of replicas for each deployment.)
```
kubectl scale --replicas=0 -n core deployment/cdf-apiserver
kubectl scale --replicas=0 -n core deployment/idm
kubectl scale --replicas=0 -n core deployment/itom-vault
kubectl scale --replicas=0 -n core deployment/mng-portal
kubectl scale --replicas=0 -n core deployment/kube-registry
kubectl scale --replicas=0 -n core deployment/suite-conf-pod-arcsight-
installer
kubectl scale --replicas=0 -n core deployment/suite-db
kubectl scale --replicas=0 -n core deployment/suite-installer-frontend
```

Note: Any consumer jobs displayed during the listing are just temporary one-time actions and can be deleted by `kubectl delete pod -n core <job_name>`

4. Verify if all Pods are deleted and not in terminating state by running this command:
```
/opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide |
awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
```
5. After make sure PV consumers list is returned empty:
```
/opt/arcsight/kubernetes/scripts/volume_admin.sh search itom-vol
```
6. Copy the NFS data to a new mount point:
```
cp -rfp /mnt/<old_nfs_mount>/itom/itom_vol /mnt/<new_nfs_mount>/itom/itom_
vol
```
7. Check the content of mount for any permissions discrepancies. The output of these commands must be identical:
```
ls -l /mnt/<old_nfs_mount>/itom/logging
ls -l /mnt/<new_nfs_mount>/itom/logging
```

8. Authorize PV change:
   ```
   /opt/arcsight/kubernetes/scripts/volume_admin.sh reconfigure itom-vol -t
   nfs -s <new_nfs_FQDN_or_IP> -p /<new_nfs_path>/itom/itom_vol
   ```
9. Repeat all the commands you used to scale down or destroy the pods to scale all replicas up or start up related daemonsets.
   ```
   #kubectl scale --replicas=<value> -n core deployment/cdf-apiserver
   kubectl scale --replicas=<value> -n core deployment/idm
   kubectl scale --replicas=<value> -n core deployment/itom-vault
   kubectl scale --replicas=<value> -n core deployment/mng-portal
   kubectl scale --replicas=<value> -n core deployment/kube-registry
   kubectl scale --replicas=<value> -n core deployment/suite-conf-pod-
   arcsight-installer
   kubectl scale --replicas=<value> -n core deployment/suite-db
   kubectl scale --replicas=<value> -n core deployment/suite-installer-
   frontend
   kubectl create -f /<new_nfs_mount>/itom/itom_vol/suite-
   install/yamlContent/kube-registry.yaml
   kubectl create -f /<new_nfs_mount>/itom/itom_vol/suite-
   install/yamlContent/itom-fluentd.yaml
   ```
10. To restore path services, use this command:
    ```
    kubectl create -f <PATH>
    ```
11. Verify consumers have been restored with this command:
    ```
    /opt/arcsight/kubernetes/scripts/volume_admin.sh search itom-vol
    ```
12. Verify pods are all running:
    ```
    /opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide |
    awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
    ```
13. If all pods are running, verify CDF status:
    ```
    /opt/arcsight/kubernetes/bin/kube-status.sh
    ```

Migrate PV db-single

1. Determine the services using the db-single PV by running the following command. (Note the number of replicas running for later scaleback, after the NFS migration):
   ```
   /opt/arcsight/kubernetes/scripts/volume_admin.sh search db-single
   ```
2. Scale down the necessary deployments:
   ```
   kubectl scale --replicas=0 -n core deployment/itom-postgresql-default
   ```
3. Verify pods are not stuck in terminating state, and afterward no consumers are displayed:
   ```
   /opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide |
   awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
   #/opt/arcsight/kubernetes/scripts/volume_admin.sh search db-single
   ```
4. Copy the NFS data to a new mount point:
   ```
   cp -rfp /mnt/<old_nfs_mount>/itom/db /mnt/<new_nfs_mount>/itom/db
   ```

5. Check the content of mount for any permissions discrepancies. The output of these commands must be identical:
   ```
   ls -l /mnt/<old_nfs_mount>/itom/logging
   ls -l /mnt/<new_nfs_mount>/itom/logging
   ```
6. Check the content of mount for any permissions discrepancies. The output of these commands must be identical:
   ```
   ls -l /mnt/<old_nfs_mount>/itom/logging
   ls -l /mnt/<new_nfs_mount>/itom/logging
   ```
7. Authorize the PV change by running this command:
   ```
   /opt/arcsight/kubernetes/scripts/volume_admin.sh reconfigure db-single -t
   nfs -s <new_nfs_FQDN_or_IP> -p /<new_nfs_path>/itom/db
   ```
8. Repeat all the commands you used to scale down or destroy the pods to scale all replicas up.
9. Verify consumers have been restored with this command:
   ```
   /opt/arcsight/kubernetes/scripts/volume_admin.sh search db-single
   ```
10. Verify pods are all running:
    ```
    /opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide |
    awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
    ```
11. If all pods are running, verify CDF status:
    ```
    /opt/arcsight/kubernetes/bin/kube-status.sh
    ```

Migrate PV arcsight-installer-xxxxx-arcsight-volume

1. Determine the services using the db-single PV by running the following command. (Note the number of replicas running for later scaleback, after the NFS migration):
   ```
   /opt/arcsight/kubernetes/scripts/volume_admin.sh search arcsight-
   installer-xxxxx-arcsight-volume
   ```
2. Scale down the necessary deployments with the following commands, **in the listed order**. (Your list may vary depending on your Transformation Hub configuration). Note that between each scaledown command, you will run a `get pods` command as shown to make sure the scaledown has finished successfully, before proceeding to the next consumer.
   ```
   kubectl scale --replicas=0 -n arcsight-installer-xxxxx deployment/th-
   kafka-manager
   /opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide |
   awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
   kubectl scale --replicas=0 -n arcsight-installer-xxxxx deployment/th-
   schemaregistry
   /opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide |
   awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
   kubectl scale --replicas=0 -n arcsight-installer-xxxxx deployment/th-web-
   service
   /opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide |
   awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
   kubectl scale --replicas=0 -n arcsight-installer-xxxxx sts/th-routing-
   ```

```
processor-group1
/opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide |
awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
kubectl scale --replicas=0 -n arcsight-installer-xxxxx
deployment/autopass-lm
/opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide |
awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
```

Note: Scaling down can take some time. Please be patient, as this is normal behavior.

3. Run these commands in the listed order:
```
kubectl scale --replicas=0 -n arcsight-installer-xxxxx sts/th-kafka
/opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide |
awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
#kubectl scale --replicas=0 -n arcsight-installer-xxxxx sts/th-zookeeper
/opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide |
awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
```
4. Verify that no consumers are displayed for the PV by running the following command: /opt/arcsight/kubernetes/scripts/volume_admin.sh search arcsight-installer-xxxxx-arcsight-volume
5. Copy the NFS data to a new mount point:
```
cp -rfp /mnt/<old_nfs_mount>/itom/db /mnt/<new_nfs_mount>/itom/db
```
6. Check the content of mount for any permissions discrepancies. The output of these commands must be identical:
```
ls -l /mnt/<old_nfs_mount>/arcsight
ls -l /mnt/<new_nfs_mount>/arcsight
```
7. Authorize the PV change: /opt/arcsight/kubernetes/scripts/volume_admin.sh reconfigure arcsight-installer-xxxxx-arcsight-volume -t nfs -s <new_nfs_FQDN_or_IP> -p /<new_nfs_path>/arcsight
8. Authorize PV change and verify the new server and volume are listed under “nfs:” section in the configuration:
```
opt/arcsight/kubernetes/scripts/volume_admin.sh reconfigure arcsight-
installer-xxxxx-arcsight-volume -t nfs -s <new_nfs_FQDN_or_IP> -p /<new_
nfs_path>/arcsighT
kubectl get pv arcsight-installer-xxxxx-arcsight-volume -o yaml
```
9. Run the scale up commands in the order shown. (After each scaleup, you will run the get pods command as shown to make sure nothing is in the crashing state.)
```
kubectl scale --replicas=<value> -n arcsight-installer-xxxxx
deployment/autopass-lm
/opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide |
awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
kubectl scale --replicas=<value> -n arcsight-installer-xxxxx sts/th-
zookeeper
#/opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide |
```

```
awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
kubectl scale --replicas=<value> -n arcsight-installer-xxxxx sts/th-kafka
/opt/arcsight/kubernetes/bin/kubectl get pods --all-namespaces -o wide |
awk -F " *|/" '($3!=$4 || $5!="Running") && $5!="Completed" {print $0}'
```

10. When all th-zookeeper and th-kafka nodes are in the running state, run these commands to scale up the rest of the PV consumers. (Note that this list may vary depending on your configuration):
```
kubectl scale --replicas=<value> -n arcsight-installer-xxxxx
deployment/th-kafka-manager
kubectl scale --replicas=<value> -n arcsight-installer-xxxxx
deployment/th-schemaregistry
kubectl scale --replicas=<value> -n arcsight-installer-xxxxx
deployment/th-web-service
kubectl scale --replicas=<value> -n arcsight-installer-xxxxx sts/th-
routing-processor-group1
```
11. Log into Kafka manager and verify topic assignment between brokers, and if all brokers are up and running.

Maintaining a Transformation Hub on Azure

You perform maintenance of a Transformation Hub on Azure using the cluster jump host. You can use one of the following methods:

- RDP to log on to the jump host desktop and accessing the CDF portal on port 5443, or,
- Run `kubectl` commands from the jump host CLI.

This chapter contains the following sections:

Enabling Access to Kafka Manager

Kafka Manager is the management tool for maintenance, management, and monitoring of topics, partitions, consumers, and Kafka brokers. It is integrated with Fusion UI and it supports Single Sign-On (SSO). The Fusion permission Manage Kafka is required for a user to access Kafka Manager.

In earlier versions of ArcSight Platform, the Manage Kafka permission was included by default in the Admin and System Admin roles. However, please note that with ArcSight Platform 22.1, the Manage Kafka permission is now included by default in the System Operations Administrator and System Admin roles, but is no longer included in the Admin role. If needed, you can manually add the Manage Kafka permission to the Admin role.

To access Kafka Manager using the default Fusion admin user with the Manage Kafka permission:

1. Follow the steps documented in the Fusion capability section: Creating the First System Admin User.

To access Kafka Manager using a non-default Fusion admin user with the Manage Kafka permission:

> **Note:** Before proceeding, request the default admin to assign the Manager Kafka permission to the non-default admin user.

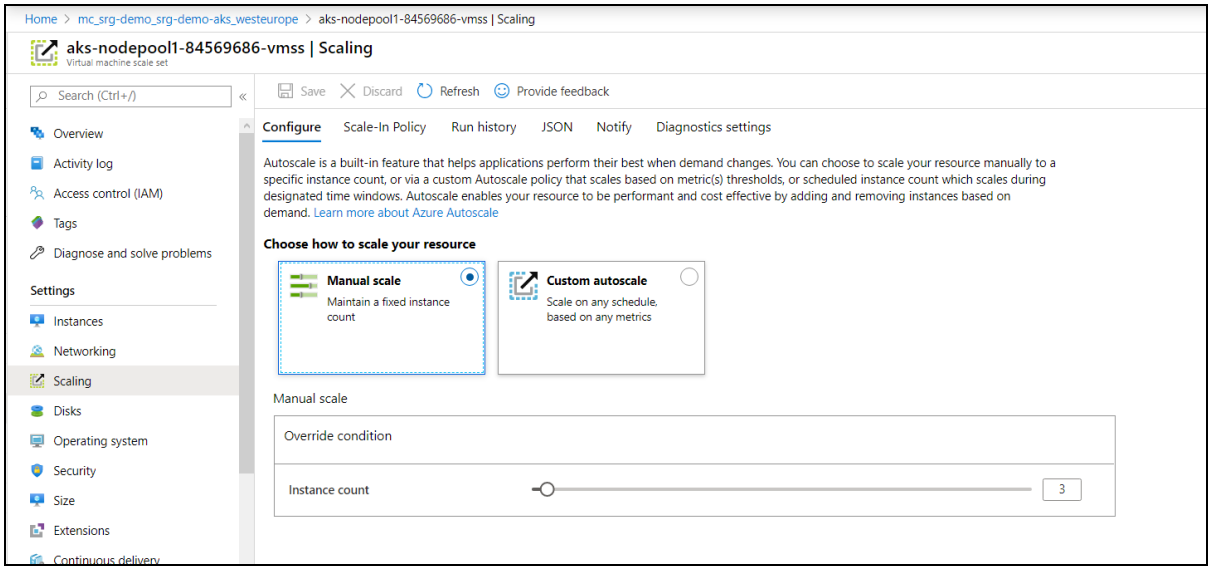
1. Log in to the jump host.
2. Browse to `https://<cdf_external_hostname>/th/cmak`.
3. Log in using the user credentials.
4. Log out by browsing to the Fusion home page: `https://<cdf_masternode_hostname or virtual_ip hostname>/`

Scaling a Cluster Up

You can increase the number of nodes in an Azure cluster by using either the Azure Portal or Azure Cloud Shell. After increasing the number of nodes, you must then label all new nodes.

To add nodes to a cluster using the Azure Portal:

1. In the Azure Portal, locate the Azure Kubernetes resource group. (The AKS resource group name is in the format MC_<your_resource_group>_<aks_name>_<location>.)
2. Open the virtual machine scale set. (The scale set name is in the format *aks-nodepool1-<NUMBER>-vmss.)*
3. Under **Settings**, click **Scaling**.
4. In **instance count**, increase the value to the desired number of nodes in the cluster.



To add nodes to a cluster using the Azure Cloud Shell or jump host CLI:

1. Obtain the AKS resource group name and store it in an environment variable for later usage:
   `CLUSTER_RESOURCE_GROUP=$(az aks show --resource-group <RESOURCE GROUP> --name <AKS NAME> --query nodeResourceGroup -o tsv)`

For example:
`CLUSTER_RESOURCE_GROUP=$(az aks show --resource-group srg-demo --name srg-demo-aks --query nodeResourceGroup -o tsv)`

2. Get the AKS Virtual machine scale set by running the following command:
   `VMSS=$(az vmss list -g $CLUSTER_RESOURCE_GROUP | jq -r .[0].name)`
3. Get the current number of nodes by running the following command:
   `az vmss list -g $CLUSTER_RESOURCE_GROUP | jq -r '.[0].sku.capacity'`
4. Add instances by increasing the value of the `new-capacity` parameter to reflect the new number of instances. For example:
   `az vmss scale --resource-group $CLUSTER_RESOURCE_GROUP --name $VMSS --new-capacity 5`

To label new nodes:

1. Obtain a list of all nodes by running the following command:
   `kubectl get nodes`
2. Select one of the new nodes and label it by running the following command:
   `kubectl label node <virtual_machine_scale_set> role=loadbalancer Worker=label <<label1> <label2>... <labelN>`

Where:

- `role=loadbalancer` and `Worker=label` are mandatory labels.
- `label1` through `...labelN` represent any number of optional labels depending on node usage. For more information about labels, see Labeling Nodes.

For example:
```
kubectl label node aks-nodepool1-84569686-vmss000003 role=loadbalancer
Worker=label zk=yes kafka=yes th-platform=yes th-processing=yes
```

3. Repeat Step 2 for each of the other new nodes.
4. If any of the new nodes contain ZooKeeper and Kafka pods, you must assign new partitions. For more information, see Launch the Kafka Manager.
5. Click the **Topics** list.
6. Click **Generate New Partition Assignments.**
7. Assign the desired topics to the new nodes.
8. Click **Run Partition Assignments.**

Peering Virtual Networks

Peering enables services from different virtual networks to communicate with one another using private IP addresses. This section discusses how to peer two Azure virtual networks; for instance, peering an AKS virtual network with a SmartConnector or other ArcSight product.

You should consult the Azure documentation on peering virtual networks for precise commands. The procedure here is provided as an example. In this peering example, we use the Azure Cloud Shell to peer the virtual network (vnet) demo-vnet, from the resource group srg-demo with the virtual network qaprg-vnet from resource group qaprg.

To set up peering between these two example virtual networks using the Azure Cloud Shell:

1. Obtain the ID for the virtual network demo-vnet from resource group srg-demo, and store it in the variable `vNet1Id`:
   ```
   vNet1Id=$(az network vnet show --resource-group srg-demo --name demo-vnet
   --query id --out tsv)
   ```
2. Obtain the ID for the virtual network qaprg-vnet from resource group qaprg, and store it in the variable `vNet2Id`:
   ```
   vNet2Id=$(az network vnet show --resource-group qaprg --name qaprg-vnet --
   query id --out tsv)
   ```
3. Establish peering for the vnet demo-vnet from resource group srg-demo to remote virtual network (ID in `$vNet2Id`) with the following command:
   ```
   az network vnet peering create --name demo-vnet-to-qaprg-vnet --resource-
   ```

```
group srg-demo --vnet-name demo-vnet --remote-vnet $vNet2Id --allow-vnet-
access
```

Where the `name` parameter is symbolic. You can choose a value for this as desired.

4. To establish a connection, you must establish peering from qaprg-vnet to demo-vnet. Run the following command:
   ```
   az network vnet peering create --name qaprg-vnet-to-demo-vnet --resource-
   group qaprg --vnet-name qaprg-vnet --remote-vnet $vNet1Id --allow-vnet-
   access
   ```
5. To verify the establishment of peering, run the following command:
   ```
   az network vnet peering show --name demo-vnet-to-qaprg-vnet --resource-
   group srg-demo --vnet-name demo-vnet --query peeringState
   ```

Change the name to the same name used in Step 4, and use your vnet and resource group.

6. If peering has been established successfully, then `Connected` is returned.

Configuring Health Probes and Load Balancing Rules for Product Integration

You must configure health probes and load-balancing rules for product integration. These ports include 9093 and, optionally, 9092, as well as 32080/82080.

Add a Health Probe for port 9093

1. On the Azure Portal, find your Azure kubernetes resource group (name format *MC_<you_resource_group>_<aks_name>_<location>*).
2. Open the AKS.
3. Find the entry for the Kubernetes load balancer, then open it.
4. Click **Health probes.**
5. Click **+Add**to add a Kubernetes load balancer health probe.
6. Specify these values:
   - **Name:** Assign a name to the probe.
   - **Protocol:** Select *TCP*.
   - **Port:** Specify 9093.

Add Load Balancing Rule for Port 9093

1. Open the Kubernetes load balancer and click **Load balancing rules.**
2. Click **+ Add** to add a new Kubernetes load balancing rule.

3. Specify these values:
   - **Name:** Assign a name to the rule.
   - **Port:** Specify 9093.
   - **Backend port:** 9093
   - **Health probe:** Select the port 9093 health probe you just created.
   - **Frontend IP Address:** Use the public IP address you prepared earlier.

Add a Health Probe for port 9092

1. On the Azure Portal, locate your Azure Kubernetes resource group (name format *MC_<you_resource_group>_<aks_name>_<location>*).
2. Open the AKS.
3. Locate the entry for the Kubernetes load balancer, then open it.
4. Click **Health probes.**
5. Click **+Add**to add a Kubernetes load balancer health probe.
6. Specify these values:
   - **Name:** Assign a name to the probe.
   - **Protocol:** Select TCP.
   - **Port:** Specify 9092.

Add Load Balancing Rule for Port 9092

1. Open the Kubernetes load balancer and click **Load balancing rules.**
2. Click **+ Add** to add a new Kubernetes load balancing rule.
3. Specify these values:
   - **Name:** Assign a name to the rule.
   - **Port:** Specify 9092.
   - **Backend port:** Specify 9092.
   - **Health probe:** Select the port 9092 health probe you just created.
   - **Frontend IP Address:** Use the public IP address you prepared earlier.

Add a Health Probe for port 32080

1. On the Azure Portal, locate your Azure kubernetes resource group (name format *MC_<you_resource_group>_<aks_name>_<location>*).
2. Open the AKS.

3. Find the entry for the Kubernetes load balancer, and then open it.
4. Click **Health probes.**
5. Click **+Add**to add a Kubernetes load balancer health probe.
6. Specify these values:

- **Name:** Assign a name to the probe.
- **Protocol:** Select *TCP*,
- **Port:** Specify 32080.

Add Load Balancing Rule for Port 32080

1. Open the Kubernetes load balancer and click **Load balancing rules.**
2. Click **+ Add** to add a new Kubernetes load balancing rule.
3. Specify these values:
   - **Name:** Assign a name to the rule.
   - **Port:** Specify 32080.
   - **Backend port:** Specify 32080.
   - **Health probe:** Select the port 32080 health probe you just created.
   - **Session persistence:** Select *Client* or *Client IP and Protocol.* (If one does not work, then try the other.)
   - **Frontend IP Address:** Use the public IP address you prepared earlier.

Removing a Product

To remove a product (capability) from your cluster:

1. Click **Deployment > Deployments.**
2. Click **... (Browse)** on the far right, and then choose **Install**.
3. On the next page, deselect the product you want to remove, and then click **Next**
4. On the **File Storage** page, click **Next**.
5. Update configuration values if needed, and then click **Next**.

After a short wait, the **Configuration Complete** page confirms the change to the cluster.

Uninstalling Installed Products and CDF from Azure

You have several options for uninstallation of CDF and your installed products from Azure. Each of these options is explained in detail below.

- You can uninstall any or all installed products.
- In addition to uninstalling installed products, you can also uninstall CDF but leave your cluster resources in place. Perform this option if you plan to re-use the cluster and re-install CDF later.
  - If CDF was installed using the ArcSight Installer, then use the ArcSight Installer to uninstall CDF.
  - If CDF was installed manually, then manually uninstall CDF.
- You can uninstall products and CDF as above, and then destroy all resources created during platform setup. Only perform this option when the cluster is no longer needed.

To uninstall CDF from your Azure installation using the ArcSight Installer:

The ArcSight Platform Uninstaller simplifies the uninstall procedure.

Prerequisites

Ensure the following before you begin:

- Verify that the ArcSight Platform Installer was used to deploy the original cluster.
- The uninstaller requires all installed software images and the `yaml` configuration file to reside in appropriate locations on the jump host used during the original installation.

You are now ready to begin the uninstallation.

1. On the jump host:
   a. Log in to the jump host and execute the uninstall command from `/opt/arcsight-platform-installer-x.x.x.x/`

   ```
   #./arcsight-install -c /opt/my-install-config.yaml --cmd uninstall
   ```

   b. Execute the following commands to remove these directories from `/opt`.

   ```
   # rm -rf /opt/arcsight/
   ```

   ```
   # rm -rf /opt/arcsight-db-tools/
   ```

   ```
   # rm -rf /opt/containerd/
   ```

   c. Execute the following commands to remove these directories from `/root`.

```
# rm -rf /root/.docker/
```

```
# rm -rf /root/.kube/
```

```
(Conditional) # rm -rf /root/.ssh/
```

d. Manually delete the `/opt/arcsight` directory.

2. On the CDF worker nodes:
   a. Execute the following commands to these current installation directories from `/opt`

```
# rm -rf /opt/arcsight/
```

```
# rm -rf /opt/containerd/
```

   b. Execute the following commands to remove these directories from `/root`

```
# rm -rf /root/.kube/
```

```
(Conditional) # rm -rf /root/.ssh/
```

   c. Manually delete the `/opt/arcsight` directory.
3. On each ArcSight Database node:
   a. Execute the following commands to remove the `arcsight-database` directory as follows:

```
# rm -rf /opt/arcsight-db-tools
```

```
(Conditional) # rm -rf /root/.ssh/
```

4. On the NFS nodes:
   a. Execute the following commands to remove the `arcsight-nfs` directory :

```
# rm -rf /opt/arcsight-nfs
```

   b. Execute the following command to remove all content (related to `/opt/arcsight-nfs`) from the `exports` file:

```
# vi /etc/exports
```

   c. Execute the following command to unexport all directories deleted in the previous step:

```
# exportfs -ra
```

d. (Conditional) Execute the following commands to remove the `.ssh/` directory from `/root`:

```
# rm -rf /root/.ssh/
```

You can now proceed to uninstall your installed products.

To uninstall CDF from your Azure installation manually:

1. On the jump host and all worker nodes :
   a. Execute the uninstall command from `/opt/arcsight/kubernetes/`:

   ```
   # ./uninstall.sh
   ```

   b. Manually delete the `/opt/arcsight` directory.
2. On the jump host, uninstall the ArcSight Database:
   a. Execute the uninstall command from `/opt/arcsight-db-tools`:

   ```
   ./db_installer uninstall
   ```

3. On each ArcSight Database node:
   a. Execute the following command to remove the `/opt/arcsight-db-tools` directory as follows:

   ```
   # rm -rf /opt/arcsight-db-tools
   ```

   b. (Conditional) Execute the following command to remove `.ssh/` directory from `/root`:

   ```
   # rm -rf /root/.ssh/
   ```

You can now proceed to uninstall your installed products.

To uninstall installed products:

If you are also uninstalling CDF, then prior to uninstalling your products, perform the uninstallation of CDF first (using either the ArcSight Installer or manually), and then return here to proceed with uninstalling your products.

1. Log in to the jump host and become `root`.
2. Get the names of all namespaces by running the command:
   `kubectl get namespaces`

For example:

```
kubectl get namespaces
```

```
NAME                          STATUS   AGE
arcsight-installer-blk62   Active   41m
core                          Active   48m
default                       Active   84m
kube-public                   Active   84m
kube-system                   Active   84m
```

3. Delete the product namespaces you wish to delete, and the core namespace by running the command:
   `kubectl delete namespace <namespace name>`

For example:

```
kubectl delete namespace arcsight-installer-blk62
namespace "arcsight-installer-blk62" deleted
kubectl delete namespace core
namespace "core" deleted
```

Your own product namespace will have the name format `arcsight-installer-XXXXX`.

4. Wait for the selected namespaces to be deleted before continuing.
5. Get the names of all PVs (persistent volumes) by running the command:
   `kubectl get pv`

For example:

```
kubectl get pv
NAME                                        CAPACITY   ACCESS MODES   RECLAIM
POLICY   STATUS
arcsight-installer-blk62-arcsight-volume   30Gi       RWX            Retain
        Released
arcsight-installer-blk62-db-backup-vol     1Mi        RWX            Retain
        Released
```

```
db-single                                  10Gi       RWX            Retain
        Released
```

```
itom-logging                               1Mi        RWX            Retain
        Released
```

```
itom-vol                                   5Gi        RWX            Retain
        Released
```

6. Delete all PVs by running the following command for each PV:
   `kubectl delete pv <PV_name>`

```
kubectl delete pv arcsight-installer-blk62-arcsight-volume
```

```
persistentvolume "arcsight-installer-blk62-arcsight-volume" deleted
```

```
kubectl delete pv arcsight-installer-blk62-db-backup-vol
```

```
persistentvolume "arcsight-installer-blk62-db-backup-vol" deleted
```

```
kubectl delete pv db-single
```

```
persistentvolume "db-single" deleted
```

```
kubectl delete pv itom-logging
```

```
persistentvolume "itom-logging" deleted
```

```
kubectl delete pv itom-vol
```

```
persistentvolume "itom-vol" deleted
```

7. Clear the data from your NFS volumes by connecting with SSH and clearing (but **not** deleting) all exported directories.

If you deleted the SSH inbound rule, you will need to add it again to be able to SSH to your NFS.

Disposal of Cluster Resources

The procedures detailed above will leave your cluster resources intact. CDF and applications can be re-installed again on the cluster, either now or in the future, without having to re-create these resources.

If instead the cluster is no longer needed, you can safely destroy all resources created earlier for CDF and your installed applications. Consult the Azure documentation for details on how to destroy resources.

Maintaining a Deployment on AWS

Maintenance of an AWS deployment is performed through the bastion host.

This chapter contains the following sections:

Enabling Access to Kafka Manager

Kafka Manager is the management tool used for maintenance, management, and monitoring of topics, partitions, consumers, and Kafka brokers. It is integrated with Fusion and supports single sign-on (SSO). The Fusion capability (and a Fusion admin account with the Manage Kafka permission) is required for Kafka Manager access.

In earlier versions of ArcSight Platform, the Manage Kafka permission was included by default in the Admin and System Admin roles. However, please note that with ArcSight Platform 22.1, the Manage Kafka permission is now included by default in the System Operations Administrator and System Admin roles, but is no longer included in the Admin role. If needed, you can manually add the Manage Kafka permission to the Admin role.

To access Kafka Manager using the default Fusion admin user with the Manage Kafka permission:

1. Follow the steps documented in the Fusion capability section: Creating the First System Admin User.

To access Kafka Manager using a non-default Fusion admin user with the Manage Kafka permission:

Note: Before proceeding, request the default admin to assign the Manager Kafka permission to the non-default admin user.

1. Log in to your bastion host.
2. Browse to `https://<cdf_external_hostname>/th/cmak`.
3. Log in using the user credentials.
4. To log out, browse to the Fusion home page: `https://<cdf_masternode_hostname or virtual_ip hostname>/`

Refreshing the ECR Credentials in Kubernetes

During the initial CDF installation, the credentials and the URL for the ECR (Elastic Container Repository) are passed through environment variables `ECR_USER_NAME`, `ECR_USER_PASSWORD` and `ECR_URL`. All of these values are then stored inside Kubernetes as Docker secrets for use, in CDF during installation as well by your ArcSight Suite products.

The security policy on `AWS ECR` requires that the `ECR_USER_PASSWORD` is valid for 12 hours after creation. The following scenarios require a refresh of the password used to access the Docker images stored in the ECR when:

- The bootstrap of CDF was performed on port 3000 more than 12 hours before installation
- You need to add a new capability to the suite, images are uploaded but not registered on CDF
- You are performing a suite upgrade

In such cases, replace the stored user password with a freshly generated one

To refresh the credentials:

1. On the bastion, change directory to `arcsight-platform-cloud-installer-22.1.X.X/aws-scripts/scripts`.
2. Generate a new `ecr_credentials` snippet by running the command:

   ```
   ./upload_images_to_ECR --get-ecr-credentials
   ```

3. Add the retrieved values to the environment:

   ```
   source ecr_credentials
   ```

4. Run the following script to create a set of JSON files to be applied to the cluster:

   ```
   ./generate_aws_secret
   ```

   The script has no output, except when one or more required environment variables (`ECR_USER_NAME`, `ECR_USER_PASSWORD` or `ECR_URL`) are empty. This procedure though generates several new files named `secret_xxxx.json` in the directory where you ran the `generate_aws_secret` script.

5. Execute the following command to apply all generated files to Kubernetes:

   ```
   kubectl apply -f secret_core.json
   ```

6. Verify in the CDF, if all required or missing images are available when uploaded to the ECR correctly.

Installing the Kubernetes Metrics Server

The Kubernetes Metrics Server is a service running inside the Kubernetes cluster, which collects various utilization and performance data later used by the ArcSight Management Center (ArcMC).

A detailed guide on how to install Kubernetes Metrics Server service is provided by AWS. It is expected that the version of Kubernetes Metrics Server will evolve as the time goes by.

Removing the Cluster

You can remove the cluster to make it possible to perform a fresh installation of CDF and the ArcSight Suite. As part of the removal, you will remove or clean some cluster-specific resources. However, some of your existing resources are reusable.

Reusable Resources

Many of the resources created during your CDF and ArcSight Suite installation are reusable, and do not need to be removed during the cluster removal. You might find it useful to keep such resources on hand for use with other product suites.

- Launch configuration is not dependent on the CDF installation and is not bound to a VPC. Other users performing installation in the same region could re-use an already existing launch configuration.
- Bastion instance is bound to the VPC created for CDF, and can be used only within this VPC. However, a bastion is a highly re-usable resource for installing and managing other clusters or product suites.
- The Route 53 record set, with its certificate, is not dependent on the installation.

Cluster Removal

As part of removing the cluster, you will perform the following tasks:

- Removal of the Auto Scaling group
- Removal of the EKS control plane
- Cleaning or deleting the EFS/NFS

Each of these procedures is explained below.

Removing the Auto Scaling Group

The AWS Auto Scaling group holds the worker nodes instances. Accordingly, in order to delete the worker nodes, you must delete the Auto Scaling group.

To delete the Auto Scaling group:

1. Run the following command:
   ```
   aws autoscaling delete-auto-scaling-group --force-delete --auto-scaling-group-name <auto-scaling group name from AWS worksheet>
   ```
2. The command has no output, and in the background the deletion instances will start. Check the presence of the Auto-Scaling group by running the following command:
   ```
   aws autoscaling describe-auto-scaling-groups \
   | jq -r '.AutoScalingGroups[] | select(.AutoScalingGroupName=="<auto-scaling group name>") | .AutoScalingGroupName'
   ```

If the group name is returned, the Auto Scaling group and its instances are not fully deleted including its instances. The process can take around 5 minutes to complete.

3. Once the auto-scaling group and worker nodes are removed, you can check the pods by executing this command on the bastion:
   ```
   kubectl get pods -A -o wide
   ```

All pods are shown in the Pending state, as they do not have a host to run on, but the Kubernetes control plane still has the cluster definition.

If desired, you can create another Auto Scaling group with a different launch configuration. All the pods will be deployed and started on the new worker nodes. Remember to add respective targets to Target Groups. Any new worker nodes will receive new instance IDs.

Removing the EKS Control Plane

The Kubernetes control plane holds the definitions of services, daemons, deployments, pods, and other resources, including the fully qualified identifier of Docker images in the registry. To clean the AWS infrastructre for a new installation, this control plane needs to be removed as well.

To remove the EKS control plane:

1. Run the following command:
   ```
   aws eks delete-cluster --name <cluster name from AWS worksheet>
   ```

The command output is very verbose. An example is given below.

2. Verify the cluster has been deleted by running the command:
   `aws eks list-clusters | jq -r '.clusters[] | select(.=="<cluster name from AWS worksheet>")'`

   An empty output indicatesthat the cluster has been deleted.

Example output of a cluster in the process of being deleted:

```
{
   "cluster":{
      "name":"srgdemo-cluster",
      "arn":"arn:aws:eks:eu-central-1:115370811111:cluster/srgdemo-cluster",
      "createdAt":"2020-08-10T12:13:31.748000+02:00",
      "version":"1.20",
      "endpoint":"https://90842F339FC27B9BE1DD0554E508B914.gr7.eu-central-
1.eks.amazonaws.com",
      "roleArn":"arn:aws:iam::115370811111:role/ARST-EKS-Custom-Role",
      "resourcesVpcConfig":{
         "subnetIds":[
            "subnet-0fb2ebb5882c061f0",
            "subnet-0abd7cd806e04c7be",
            "subnet-0f0cac4ec6837abed"
         ],
         "securityGroupIds":[
            "sg-0ce3c569f73737b77"
         ],
         "clusterSecurityGroupId":"sg-0263ae0d4c33decc4",
         "vpcId":"vpc-0143197ca9bd9c117",
         "endpointPublicAccess":false,
         "endpointPrivateAccess":true,
         "publicAccessCidrs":[

         ]
      },
      "logging":{
         "clusterLogging":[
            {
               "types":[
                  "api",
                  "audit",
                  "authenticator",
                  "controllerManager",
                  "scheduler"
               ],
               "enabled":false
            }
         ]
      },
```

```
      "identity":{
         "oidc":{
            "issuer":"https://oidc.eks.eu-central-
1.amazonaws.com/id/90842F339FC27B9BE1DD0554E508B914"
         }
      },
      "status":"DELETING",
      "certificateAuthority":{

"data":"LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUN5RENDQWJDZ0F3SUJBZ0lCQURBT
kJna3Foa2lHOXcwQkFRc0ZBREFWTVJNd0VRWURWUVFERXdwcmRXSmwKY201bGRHVnpNQjRYRFRJd0
1EZ3hNREV3TWpFd01sb1hEVE13TURnd09ERXdNakV3TWxvd0ZURVRNQkVHQTFVRQpBeE1LYTNWaVp
YSnVaWFJsY3pDQ0FTSXdEUVlKS29aSWh2Y05BUUVCQlFBRGdnRVBBRENDQVFvQ2dnRUJBSnZXCnUy
MDVNZFJkWUV1VkliQU1yeFJzKzFSMmtyRlhWUmpZd0ZXQUdIRUY2WmJ6V1F2L2Y0d052MmlxaFM0Q
0lJa2wKVTVvTmtaTzFBaU9USk9Ja1l3UFAwdjRGQkNyVFlvU3BldW1xelhqVFBHU2JFUVJ4OXFVM0
ttTkorUXlSZEhJeQpaTHV6b2tXbXJXSG1TVlRLNUxkZUppN3Z4enoweU1TNzczL01GK3FkcVNML2o
1dHJTNEt2cU5ObVRKMEVVY0hwCjdWNklENnFaSEVxZXdKQjl2cmhPdGFlc05TMFdhVWwwUFU4d3pW
aFVUWUlEMllFTU8rOXFsZEdVQVlWTmo3cVIKMUdXVVNVZVVIUWJqNEViMHg4VGhjcDNPYi9oZUNQW
WZ1Rno5MVRWUUR5enRxaDZtUDQvNXFZaW1QeklkaFh5LwpIdDltVmZ3M0tVemlzMURtNk9VQ0F3RU
FBYU1qTUNFd0RnWURWUjBQQVFIL0JBUURBZ0trTUE4R0ExVWRFd0VCCi93UUZNQU1CQWY4d0RRWUp
Lb1pJaHZjTkFRRUxCUUFEZ2dFQkFDU2pyZG5Xb0N1WTA4c3pqVU5BSHdnbnFtMDgKZlhydkxtVkxz
SHZiZHFSTmorUTJQMFQvVCtFZFRVWFg5SGNia1JwQU5QNTRkNzRQRmJGbzA0K0dmaTYrTHE5UAoyY
lBzZ2o3Mmo4WWx0V0twVHJiNFpKMnhyZXFsWnZ4MVFZNHpZWUhKdDdKZ1RRaU4xQ2JjaFZLR0V6K0
9nQ3ZTClZGMWE2OEJJajlUMFFDNXgzTTJncHdDa1JMOHArbXkzbkp0Z281Q0JHanhGU2ZHNnN3M0Z
MRXdlRHQyc2dOc1UKV2hpQWZGQmtPdUl2OENmMmlwMUZYQ2toWjJxTXdYanU4UzFFc3Z3bUcrSy9v
d3NiOUFLZG5TaVRQVXJSQWdGbwpsVjBrSGVaK1FpSG5wK0t3a1NpbkoyMVpXRUFMVG5GRjBCR3hYM
DhpU1cwM25Kcy9XemRFdTVFWWhUYz0KLS0tLS1FTkQgQ0VSVElGSUNBVEUtLS0tLQo="
      },
      "platformVersion":"eks.1",
      "tags":{
         "user":"user"
      }
   }
}
```


Cleaning or deleting NFS/EFS

Your NFS/EFS is a partially reusable resource. For the EFS you created, you have the following options:

- Leave the existing EFS folder structure intact. A new installation will use a parallel structure. No action needs to be taken.
- Delete and re-create the folder structure during a new installation. The procedure is

discussed below.

- Delete the EFS instance completely. The procedure is discussed below.

To delete the folder structure:

1. Log on to the bastion host.
2. Unmount the EFS file system by running the following command:
   `sudo umount -f /mnt/efs`
3. As a `sudo` user, open the file `/etc/fstab` in a text editor.
4. Locate the following line:
   `fs-5df66605.efs.eu-central-1.amazonaws.com:/ /mnt/efs nfs4 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,_netdev 0 0`
5. Uncomment the line and then save the file.
6. Mount EFS to the bastion by running the following command:
   `sudo mount -a`
7. Change your current working directory to the mount point, in our case `/mnt/efs`.
8. Delete the whole folder structure by running the following command:
   `sudo rm -Rf <parent folder from AWS worksheet>`

To delete the EFS instance (not required for re-installing the CDF bootstrap):

1. Delete the mount targets by running the following command on each configured mount target:
   ```
   aws efs delete-mount-target \
   --mount-target-id <mount target id from AWS worksheet>
   ```
2. Verify the deletion by running the following command:
   ```
   aws efs describe-mount-targets \
   --file-system-id <filesystem Id from AWS worksheet>
   ```

Example output:

```
{
   "MountTargets":[

   ]
}
```

3. Delete the filesystem by running the command:
   `aws efs delete-file-system --file-system-id <filesystem Id from AWS worksheet>`

Next Steps

At this point the filesystem has been deleted. As explained above, some reusable resources will remain.

- The Application Load Balancer, its listeners, and target groups are not dependent on installation. For a new installation, you will need to add new targets to all target groups.
- The VPC tag marking the EKS cluster has been removed, and the required tag `kubernetes.io/cluster/<cluster name>` has been removed as well. Remember to add it before new installation.

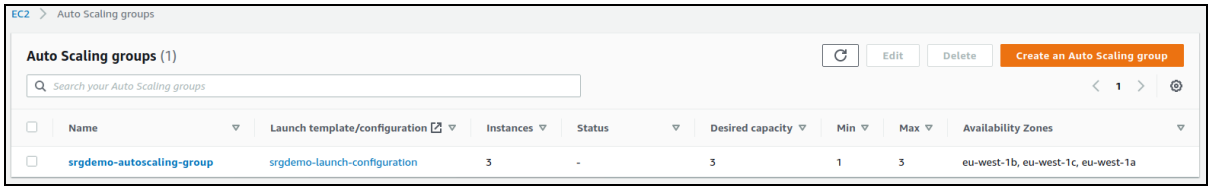
You can now perform a clean installation of a new cluster.

Scaling Up an AWS Cluster

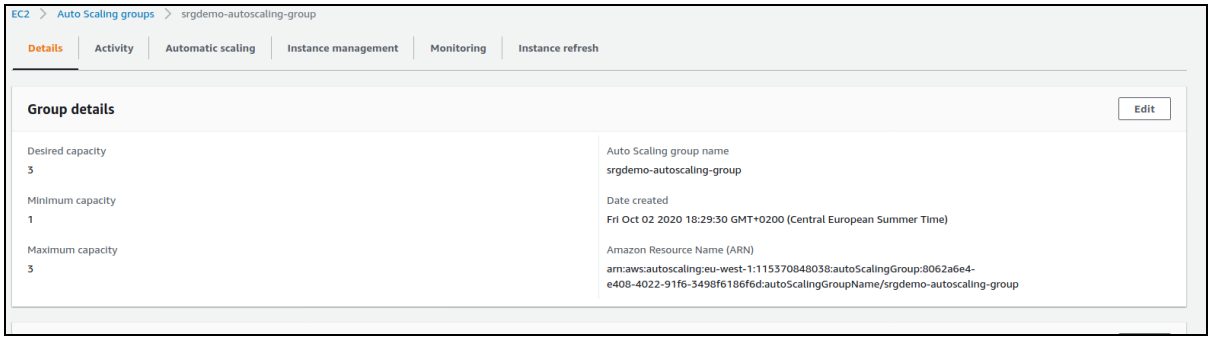
The process of scaling up a cluster involves adding nodes, labeling them as required, and performing some additional configuration. In an AWS cluster, instances are managed by Auto Scaling groups, which include the current number of nodes, as well the upper and lower limits of the nodes for the cluster, and these values must be adjusted to accomodate new instances.

To scale up a cluster using the Web UI:

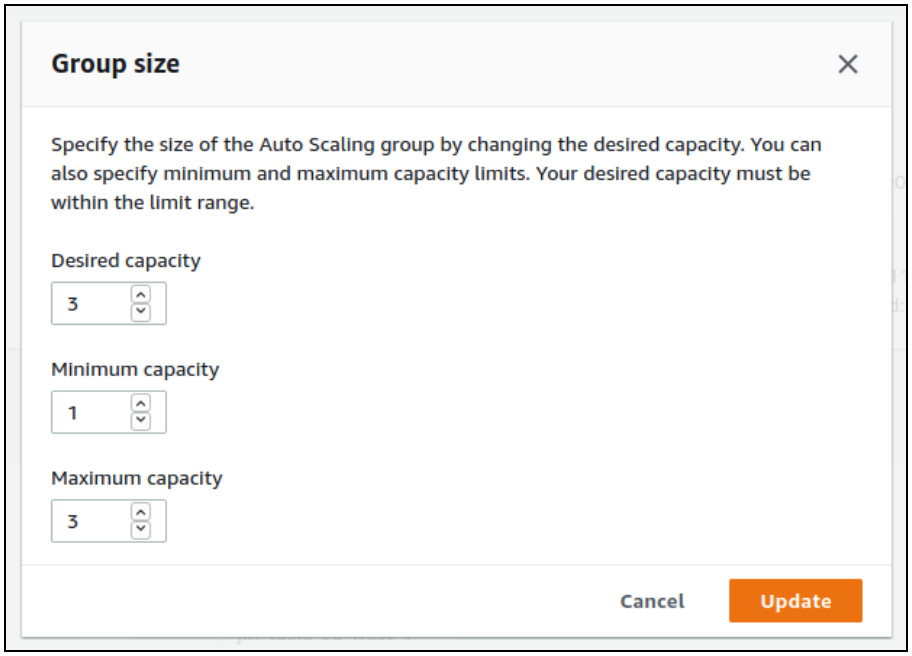
1. Using the Find Services search tool, locate and browse to the EC2 Dashboard.
2. In the left navigation panel, under **Auto Scaling,** click **Auto Scaling Groups**.
3. From the list of **Auto Scaling groups**, click the Auto Scaling group that you previously created.



4. On the **Details** tab, under **Group Details**, click **Edit**.



5. On the **Group Size** dialog, specify values for the cluster's desired, minimum, and maximum capacities.
6. Click **Update**.



7. You are returned to the Auto Scaling group details. The group status is shown as *Updating capacity*.

To scale an AWS cluster up using the CLI:

1. Verify that the new number of nodes will fit into the existing limits set on your Auto Scaling group. Run the following command:
   ```
   aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names
   <your auto-scaling group name> \
   | jq -r '.AutoScalingGroups[0] | "MaxSize: " + (.MaxSize |
   tostring),"DesiredSize: " + (.DesiredCapacity | tostring),"MinSize: " +
   (.MinSize | tostring)'
   ```

Example input and output:

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names
srgdemo-autoscaling-group \
| jq -r '.AutoScalingGroups[0] | "MaxSize:         " + (.MaxSize |
tostring),"DesiredCapacity: " + (.DesiredCapacity | tostring),"MinSize:
  " + (.MinSize | tostring)'
```

```
MaxSize:         3
```

```
DesiredCapacity: 3
```

```
MinSize:         1
```

In this example, both the `DesiredCapacity` as well as the `MaxSize` must be increased.

2. Increase the number of nodes in the Auto Scaling group by running this command (command has no output):
   ```
   aws autoscaling update-auto-scaling-group \
   --auto-scaling-group-name <your auto-scaling group name> \
   --max-size <new maximum size of the cluster> \
   --desired-capacity <desired new number of nodes>
   ```

Example:

In this example, the number of active nodes in our example group `srgdemo-autoscaling-group` is raised to 5, and at the same time the group capacity is increased to 7. The command has no output.

```
aws autoscaling update-auto-scaling-group \
--auto-scaling-group-name srgdemo-autoscaling-group \
--max-size 7 \
--desired-capacity 5
```

3. To verify the updated values, repeat the command shown in Step 1, above.
4. To verify creation of the new nodes, on the bastion host, run the following command until the desired number of nodes is shown in the Ready state:
   `kubetlc get nodes`

Labeling New Nodes

You might need to label the new nodes after they are added and ready to use.

- You will not need to add new labels to new nodes if you have previously extended the set of labels for Launch configuration.
- Otherwise, label the new nodes now.

To verify labels on new nodes:

Run the following command:
`kubectl get nodes --show-labels`

Extending Targets in Target Groups

Your newly created nodes are now labeled. Kubernetes might migrate pods used for installation, management and re-configuration to these new nodes as part of cluster operations. You will now need to perform some tasks for the new nodes that you have previously performed for the old nodes.

1. Retrieve instance IDs from auto-scaling group: Retrieve the IDs of all instances in your cluster. Identify the new instances by comparing with existing list of instance IDs. Note any new instance IDs to your AWS worksheet.
2. Repeat the steps in listed in Target Group 3000 - Get CDF Ingress service node port for 3000 and Add targets to target group 3000 to extend routing for installation portal.
3. Repeat the steps in chapter Configure access to CDF management - Get CDF Ingress service node port for 5443 and Add targets to target group 5443 to extend routing for management portal.
4. Repeat the step in chapter Configure access to re-configuration - Add targets to target group 443.

You might use the web UI or CLI to perform these tasks, as desired. Only apply these procedures to your new nodes.

Additional Steps

For Transformation Hub, see the following procedures:

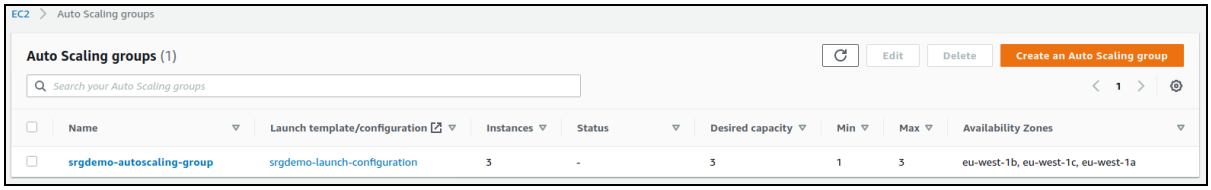
- Adding a Kafka Broker Instance
- Adding a ZooKeeper Instance

Scaling Down an AWS Cluster

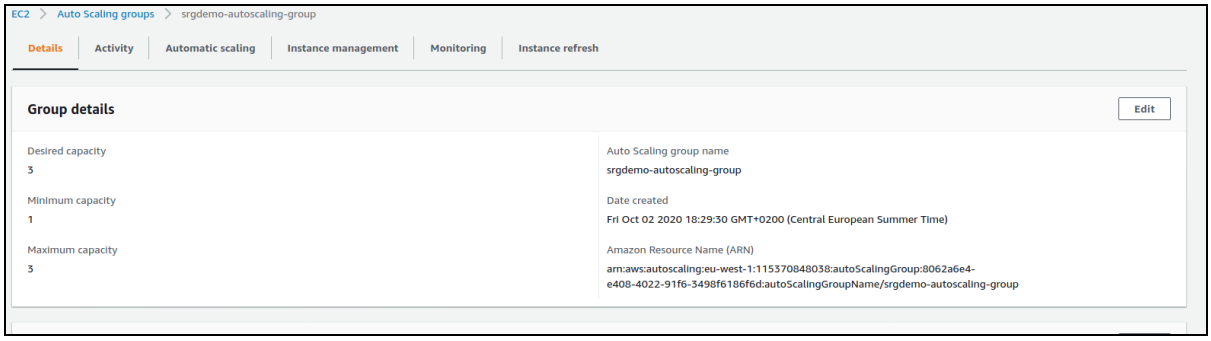
In an AWS cluster, instances are managed by Auto Scaling groups, which include the current number of nodes, as well the upper and lower limits of the nodes for the cluster.

To scale down a cluster using the Web UI:

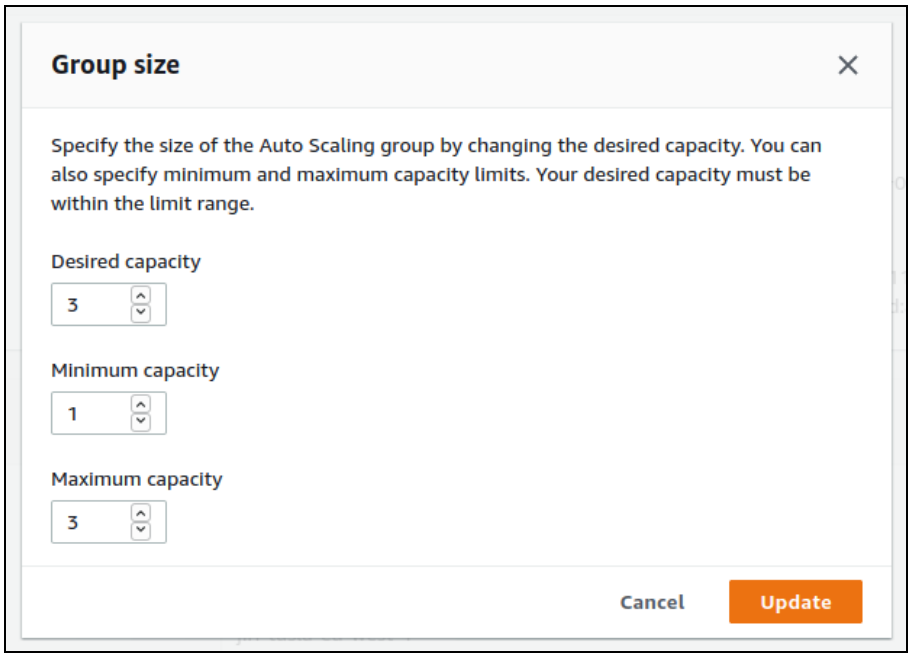
1. Using the Find Services search tool, locate and browse to the EC2 Dashboard.
2. In the left navigation panel, under **Auto Scaling,** click **Auto Scaling Groups**.
3. From the list of **Auto Scaling groups**, click the Auto Scaling group that you previously created.



4. On the **Details** tab, under **Group Details**, click **Edit**.



5. On the **Group Size** dialog, specify new values for the cluster's desired, minimum, and maximum capacities.



6. Click **Update**.
7. You are returned to the Auto Scaling group details. The status is shown as *Updating capacity*.

To scale down the number of nodes in an AWS cluster using the CLI:

1. Verify the existing limits set on your Auto Scaling group. Run the following command:

```
aws autoscaling describe-auto-scaling-groups \
--auto-scaling-group-names <your auto-scaling group name> \
| jq -r '.AutoScalingGroups[0] \
| "MaxSize: " + (.MaxSize | tostring),"DesiredSize: " + (.DesiredCapacity
| tostring),"MinSize: " + (.MinSize \
| tostring)'
```

Example input and output:

```
aws autoscaling describe-auto-scaling-groups \
--auto-scaling-group-names srgdemo-autoscaling-group \
| jq -r '.AutoScalingGroups[0] \
| "MaxSize:         " + (.MaxSize \
| tostring),"DesiredCapacity: " + (.DesiredCapacity \
| tostring),"MinSize:         " + (.MinSize | tostring)'
```

```
MaxSize:         5
```

```
DesiredCapacity: 5
```

```
MinSize:         1
```

In this example, it would be possible to scale down to 3 nodes without any other configuration changes.

2. Decrease the number of nodes in the Auto Scaling group by running the command (command has no output):
   ```
   aws autoscaling update-auto-scaling-group \
   --auto-scaling-group-name <your auto-scaling group name> \
   --desired-capacity <desired new number of nodes>
   ```
3. To verify the updated values, repeat the command shown in Step 1, above.

The AWS will then gracefully stop the removed nodes (2 in our example) and destroy their VMs.

Understanding the Transformation Hub Kafka Manager

The Transformation Hub Kafka Manager enables you to monitor and manage your clusters, topics, and partitions, and perform the following tasks:

- Viewing and managing cluster states, including topics, consumers, offsets, broker nodes, replica distribution, and partition distribution.
- Creating and updating topics.
- Generating partitions and adding partitions to a topic.
- Reassigning partitions to other broker nodes, such as replacing a failed node with a new one.
- Reassigning partition leaders to their preferred broker node after a node temporarily leaves the cluster (for example, in case of a reboot).
- Managing JMX polling for broker-level and topic-level metrics.

Enabling Access to Kafka Manager

Kafka Manager is the management tool used for maintenance, management, and monitoring of topics, partitions, consumers, and Kafka brokers. It is integrated with Fusion and supports single sign-on (SSO). The Fusion permission Manage Kafka is required for a user to access Kafka Manager.

In earlier versions of ArcSight Platform, the Manage Kafka permission was included by default in the Admin and System Admin roles. However, please note that with ArcSight Platform 22.1, the Manage Kafka permission is now included by default in the System Operations Administrator and System Admin roles, but is no longer included in the Admin role. If needed, you can manually add the Manage Kafka permission to the Admin role.

To access Kafka Manager:

1. Browse to `https://<cdf_masternode_hostname or virtual_ip hostname>/th/cmak`
2. Log in with a user account with the Manage Kafka permission and perform the required task.
3. Log out by browsing to the Fusion home page: `https://<cdf_masternode_hostname or virtual_ip hostname>/`

Managing the Kafka Cluster

The **Clusters** page is the Transformation Hub Manager's home page. From here you can modify, disable or delete a cluster from view in the Transformation Hub Manager (the cluster itself is not deleted), or drill down into the cluster for more information.

Location: Clusters

Click the *Cluster Name* link. The Transformation Hub Manager displays the **Cluster Summary** page. For more information, see Viewing Information About a Cluster.

To edit the cluster:

1. Click **Modify**. The Transformation Hub Manager displays the **Update Cluster** page.
2. Update the appropriate fields, and click **Save**.

Editing the cluster is an advanced operation, and normally the cluster should never be edited.

To disable the cluster:

Click **Disable**. Once a cluster has been disabled, a **Delete** button is displayed.

To delete the cluster:

After disabling the cluster, click **Delete**.

Viewing Information About a Cluster

On the **Summary** page, you can view the ZooKeeper processes in your cluster and drill down into its topics and broker nodes for more information.

Location: Clusters > *Cluster Name* > Summary

- "Viewing Information" below
- "Viewing or Editing the Topics" below
- "Viewing or Editing the Broker Nodes" below

Viewing Information

To view information about your cluster:

- If the cluster is not yet open, click **Cluster > List** in the navigation bar. Then click the *Cluster Name* link.
- If the cluster is already open, click **Clusters > *Cluster Name* > Summary**

Viewing or Editing the Topics

To view or edit the topics in your cluster:

Click the **Topics** hyperlink (number of topics) to show the topics in the cluster. For more information, see Managing Topics.

Viewing or Editing the Broker Nodes

To view or edit the broker nodes in your cluster:

Click the **Brokers** hyperlink (number of broker nodes) to show the broker nodes in the cluster. For more information, see Managing Brokers.

Managing Brokers

On the **Brokers** page, you can see an overview of all of your Worker nodes and drill down into a node for more information.

The term *Brokers* refers to nodes running Kafka services (that is, Kubernetes worker nodes, but not master nodes).

Location: Clusters > *Cluster Name* > Brokers

To view the broker nodes in your cluster:

Click **Brokers** in the navigation bar. The **Brokers** page opens.

To see more information about a specific broker:

Click the broker's *Id* link. The *Broker Name* ID opens. For more information, see "Managing Brokers" above

Viewing Broker Details

You can view detailed information about a broker from the *Broker Name* details page.

Location: Clusters > *Cluster Name* > Brokers > *Broker Name*

To view information on a specific broker:

1. Click **Brokers** in the navigation bar.
2. Click the *Broker Name* link. The *Topic Name* page opens.

The following data is displayed.

Summary

In the **Summary** section, you can see an overview of your broker, including the number of topics and partitions located on it.

Metrics

In the **Metrics** section, you can view information about the data flow.

Messages count

In the **Messages** section, you can view a message view chart.

Per Topic Detail

In the **Per Topic Detail** section, you can view topic replication and partition information and drill down to view more information about each topic.

To see more information about a specific topic:

Click the *Topic Name* link in the **Per Topic Details** section. See Viewing Topic Details

Managing Topics

On the **Topics** page, you can run or generate partition assignments, add a new partition, and drill down into individual topics for more information.

Location: Clusters > *Cluster Name* Topic > List

Note: The following default topics are used internally by Transformation Hub and should not be deleted, modified, or used by external data producers or consumers.

```
__consumer_offsets
```

```
 _schemas
```

```
th-arcsight-json-datastore
```

```
th-arcsight-avro-sp_metrics
```

```
th-syslog
```

```
th-arcsight-avro
```

```
mf-event-avro-enriched
```

```
mf-event-avro-esmfiltered
```

```
mf-event-cef-esmfiltered
```

```
th-cef
```

To manage the topics in your cluster:

Click **Topic > List** in the navigation bar.

To view information on a topic:

Click the *Topic Name* link. The *Topic Name* page displays the topic's summary, metrics, consumers, and partitions. See Viewing Topic Details.

To generate partition assignments:

1. Click **Generate Partition Assignments**.
2. Select the topics and broker nodes to reassign.
3. Click **Generate Partition Assignments**.

To assign partitions as generated:

1. Click **Run Partition Assignments**.
2. Select the topics to reassign.

3. Click **Run Partition Assignments**.

To add a partition:

1. From the Topics Summary page, click **Add Partition**.
2. Enter the new number of partitions.
3. Select the topics and broker nodes.
4. Click **Add Partitions**.

Default Topics

Transformation Hub manages the distribution of events to topics, to which consumers can subscribe and receive events from.

Transformation Hub includes the following default topics:

Topic Name	Event Type	Valid Destinations
mf-event-avro-esmfiltered	Filtered Avro events for consumption by ESM.	SmartConnector or Connector in Transformation Hub (CTH).
mf-event-cef-esmfiltered	Filtered CEF events for consumption by ESM.	SmartConnector or Connector in Transformation Hub (CTH).
mf-event-avro-enriched	Event data in Avro format that has been enriched by the Enrichment Stream Processors.	Transformation Hub
th-arcsight-avro	For ArcSight product use only. Event data in Avro format.	Transformation Hub
th-arcsight-avro-sp_metrics	For ArcSight product use only. Routing stream processor operational metrics data.	
th-arcsight-json-datastore	For ArcSight product use only. Event data in JSON format for use by ArcSight infrastructure management	
th-binary_esm	Binary security events, which is the formatconsumed by ArcSight ESM.	SmartConnector
th-cef	CEF event data.	SmartConnector or Connector in Transformation Hub (CTH).
th-cef-other	CEF event data destined for a non-ArcSight subscriber.	
th-syslog	The Connector in Transformation Hub (CTH) feature sends raw syslog data to this topic using a Collector.	Should only be configured as Collector or CTH destination.

In addition, using ArcSight Management Center, you can create new custom topics to which your SmartConnectors can connect and send events.

Topic Data Preservation

Topic data is preserved across Transformation Hub restarts, reinstalls, and upgrades.

- When a Transformation Hub reinstall is performed, all data in a Kafka topic is preserved. No data is lost.
- When the consumer resumes data collection from the topics, the consumer re-starts where it last left off.

No data is lost.

Creating Topics

This method of creating topics does not permit you to specify topic type. As a result, it is strongly recommended that you use ArcMC to create new topics in Transformation Hub.

You can create a new topic on the **Create Topic** page.

Location: Clusters > *Cluster Name* **Topics > Create Topics**

To open the Add Topic page:

Click **Topic > Create** in the navigation bar.

To create a new topic:

1. Fill in values for the **Topic Name,** number of **Partitions**, and **Replication Factor** fields
2. Click **Create**.

For a discussion of field values, consult the Kafka documentation.

The number of custom topics you can create will be limited by Kafka, as well as performance and system resources needed to support the number of topics created.

Creating Routes for Topics

You can use ArcMC to view and create topics , as well as to create *routes*, which direct events into appropriate topics.

A *route* is a rule that directs Transformation Hub to duplicate events that meet certain criteria (filter) from a source topic to the route's destination topic. Rules are defined using event field names and expected values. Only CEF and Avro format events can be routed; binary security events in the `th-binary_esm` topic cannot be routed.

Using ArcMC, you can view, create, edit and delete routes based on CEF fields or Avro schema fields and event metadata. (You must create destination topics before you can route events to them.) For more information., see "Creating a Route" on page 1038.

As a general guideline, `th-arcsight-avro` is no longer a recommended source topic for Avro routing; use `mf-event-avro-enriched` instead.

Tuning the Retention Settings for Topics

Kafka topics occupy storage space on worker nodes where you apply the 'kafka:yes' label. To ensure that the nodes have enough storage space for each topic and other components that use storage on these nodes, you must tune the settings for topic retention. Please note that this procedure does not interrupt the flow of events through the system.

1. Log in to an ArcSight master node as root.
2. To determine the current topic retention storage size for *{your topic name}*, run the following command:

```
kubectl exec -n $( kubectl get pods --all-namespaces | grep kafka-0 | cut -d ' ' -f1 ) th-kafka-0 -- /usr/bin/kafka-configs --bootstrap-server th-kafka-svc:9092 --describe --topic {your topic name} | grep retention.bytes
```

3. To set the retention size for *{your topic name}*, in the following command replace *{number}* with your desired retention size, then run it:

```
kubectl exec -n $( kubectl get pods --all-namespaces | grep kafka-0 | cut -d ' ' -f1 ) th-kafka-0 -- /usr/bin/kafka-configs --bootstrap-server th-kafka-svc:9092 --alter --topic {your topic name} --add-config retention.bytes={number}
```

Deleting a Topic

In order to delete a topic, you must first enable topic deletion, and can then perform the deletion.

Note: Topic deletion is no longer supported in Kafka Manager.

To enable topic deletion:

1. If it has not been defined previously, define the environment variable `arcsight_namespace` by running the following command:
   `export arcsight_namespace=$(kubectl get ns | grep "arcsight" | awk '{print`

```
$1}')
```

2. Edit the Kafka stateful set by running the following command:
   `kubectl edit sts -n $arcsight_namespace th-kafka`
3. Add a new environment variable to the environment section for the `atlas-kafka` container definition in the stateful set.
   `KAFKA_DELETE_TOPIC_ENABLE = "true"`
4. Save the `sts` configuration and exit. The Kafka pods will restart.

To delete a topic:

1. SSH to any Kafka pod.
2. Run the following command:
   `kubectl exec -it -n $arcsight_namespace th-kafka-0 -- bash`
3. Delete the topic by running the following command:
   `kafka-topics --delete --topic $topic_name --bootstrap-server localhost:9092`
4. Verify deletion by running the following command:
   `kafka-topics --list --bootstrap-server localhost:9092`

Viewing Topic Details

You can see details about a topic, including information about the summary, metrics, consumers, and partitions from the *Topic Name* details page.

Location: Clusters > *Cluster Name* Topics > *Topic Name*

To view information on a specific topic:

1. Click **Topic > List** in the navigation bar.
2. Click the *Topic Name* link. The *Topic Name* page opens.

The following data is displayed.

Topic Summary

In the **Topic Summary** section, you view information on the topic's replicas, partitions, and broker nodes.

Metrics

In the **Metrics** section, you can view information about the data flow.

Operations

In the **Operations** section, you can perform a variety of tasks on broker nodes.

To reassign partitions:

Click **Reassign Partitions**.

To update a topic's configuration:

1. Click **Update Config**.
2. Edit the configuration fields.
3. Click **Update Config**.

To specify partition assignments:

1. Click **Manual Partition Assignment**.
2. Select the desired assignments.
3. Click **Save Partition Assignment**.

Partitions by Broker

In the **Partitions by Broker** section, you can see topic partition information and drill down to see details for each broker.

To view details on a broker:

Click the **Broker** link. The **Topic Summary** page displays information on the topic's lag, partitions, and consumer offset.

In Transformation Hub Kafka Manager, users will see different offset values between CEF (Recon or Logger) topics and binary (ESM) topics. In CEF topics, the offset value can generally be associated with number of events that passed through the topic. Each message in a CEF topic is an individual event. However, that same association cannot be made for the ESM topic, as several events are batched into each message.

Consumers consuming from this topic

In the **Consumers consuming from this topic** section, you can drill down to see details on each consumer.

New consumers can take some time to display properly. Give the process time to populate the view with the correct data.

To view details on a consumer:

Click the *Topic Name* link. The Topic Summary page displays information on the topic's lag, partitions, and consumer offset.

Partition Information

In the **Partition Information** section, you can view information about the topic's partitions and drill down for more information about each leader.

To view details on a leader:

Click the **Leader** link. The *Broker Name* ID page displays the broker's summary, metrics, message count, and topic details. See Viewing Broker Details.

Data Redundancy and Topic Replication

When configuring a Transformation Hub, you can specify the number of copies (replicas) of each topic which Transformation Hub should distribute.

Kafka brokers automatically distribute each event in a topic to the number of broker nodes indicated by the topic replication level specified during the Transformation Hub configuration. While replication does decrease throughput slightly, ArcSight recommends that you configure a replication factor of at least 2.

You need at least one node for each replica. For example, a topic replication level of 5 requires at least five nodes; one replica would be stored on each node. The following table illustrates how the replication factor provides redundancy in case of unavailable nodes.

Replication Factor	Number of brokers receiving the event	If one node becomes unavailable...
1	1	Data is lost
2 (or more)	Same as replication factor	• Copies of the event data are still present on other node. • Data is restored to an unavailable node when it becomes available again. • No data is lost unless all nodes become unavailable simultaneously.

When you add new consumers, you don't need to update your producers. The distribution and replication is handled for you. Refer to the Kafka documentation for more information.

Filtering Events for ESM

Transformation Hub is capable of filtering and routing from a source topic of type event-avro to a destination topic of type event-avro. This capability can be used to filter events from a source topic such as `mf-event-avro-enriched` to a destination topic which ESM can consume from, such as `mf-event-avro-esmfiltered`. Both of these are default topics described here.

1. Use ArcSight Smart Connectors or any producer that supports sending Avro formatted events to send the events directly to an event-avro topic. Smart Connectors by default will send Avro formatted events to the `th-arcsight-avro` topic.
2. Filter the events using Transformation Hub's Avro routing rules using ArcMC 2.96 or later. Create a routing rule with an event-avro topic as source topic (such as `mf-event-avro-enriched`) and an event-avro topic as destination topic (such as `mf-event-avro-esmfiltered`). For more information, please refer to the routing section in the ArcMC Administration Guide.

   Earlier versions of Transformation Hub that did not yet support Avro routing rules required using a combination of CEF routing rules and CEF-to-Avro conversion. Using Avro routing rules is a more efficient way to filter Events for ESM, however, so it is now the recommended approach.

   As a general guideline, `th-arcsight-avro` is no longer a recommended source topic for Avro routing; use `mf-event-avro-enriched` instead. For more information, see About Routes.

Managing Consumers

On the **Consumers** page, you can see a list of consumers, view their type, the topics they consume, and drill down into each consumer and topic for more information.

Location: Clusters > *Cluster Name* > Consumers

Viewing the Consumers in Your Cluster

Click **Consumers** in the navigation bar.

Viewing Details on a Specific Consumer

Click the *Consumer Name* link. The *Consumer Name* page displays details about the consumer. You can drill down further for more information, including Consumed Topic Information (such as Partitions Covered % and Total Lag).

Viewing Details on the Topic it Consumes

Click the *Topic Name* link. The *Topic Name* page displays details about the topic. You can drill down further for more information including Consumer Lag, and Consumer Offset and LogSize data by Partition.

Viewing Consumer Details

You can see a information about a consumer and drill down on the topics it consumes from the *Consumer Name* details page.

Location: Clusters > *Cluster Name* Consumer > *Consumer Name*

1. Click Clusters > *Cluster Name* Consumer.
2. Click the *Consumer Name*.

Managing Preferred Replicas

You can update the replicas for each cluster on the **Preferred Replica Election** page.

Location: Clusters > *Cluster Name* > Preferred Replica Election

- "Opening the Preferred Replica Election Page" below
- "Running the Preferred Replica Election for Your Topic" below

Opening the Preferred Replica Election Page

Click **Preferred Replica Election** in the navigation bar.

Running the Preferred Replica Election for Your Topic

Click **Run Preferred Replica Election**.

Managing Partitions

You can reassign partitions for your cluster on the **Reassign Partitions** page.

Location: Clusters > *Cluster Name* > Reassign Partitions

Opening Reassigned Partitions

To open the Reassign Partitions page, click **Reassign Partitions** in the navigation bar.

Reassigning the Partitions

To reassign the partitions for your topic, click **Reassign Partitions**.

Configuring Topic Partitions Based on Number of Consumers

You can scale the consumption rate for a consumer of a topic by adding more consumers to the consumer group. However, when adding new consumers to the consumer group, please consider the topic partition count of the topic you are consuming from. The following table shows the relationship between the number of consumers in a consumer group and data consumption from each partition.

Number of Consumers in Group is...	Consumption from Partitions
A single consumer	Consumer consumes from all partitions in the source topic.
Lower than partition count	Each consumer consumes from a subset of the topic partitions.
Equals partition count	Each consumer consumes from each of the topic partitions.
Exceeds partition count	Each consumer consumes from each of the topic partitions; additional consumers stay idle until new partitions are added to the source topic.

If you change the number of partitions in the source topic to match the consumer group size (same or a multiple) for a given consumer group consumption rate, or add additional consumers in the consumer to match the topic partition count, then the Transformation Hub will automatically re-balance the consumer groups.

Stream Processor Groups

Transformation Hub implements three types of stream processors to process events: routing stream processors, transforming stream processors, and enrichment stream processors.

Routing Stream Processors

Routing stream processors process event data and send it to destinations, based on Transformation Hub routing rules specified in ArcSight Management Center. There are two types of routing stream processors:

- CEF-to-CEF routing stream processing is supported in Transformation Hub 3.4.0 and all previous versions.
- In Transformation Hub 3.4.0 and later versions, Avro-to-Avro routing stream processing occurs between two `event-avro` topics. To use an Avro topic, it should be of the type `event-avro`. You can configure a topic with this type in two ways:
  - Create the topic with type `event-avro` using ArcMC 2.9.6 or later and Transformation Hub 3.4, or,
  - Change the type of an existing topic to `event-avro` using ArcMC 2.9.6 or later.

As a general guideline for routing stream processors, stream processor configurations and routes are refreshed every 60 seconds. Consider this factor when adding, deleting, or editing routing rules using ArcMC.

Transforming Stream Processors

As of ArcSight SmartConnector 8.1, the SmartConnector is capable of sending events to Transformation Hub in the Avro event format from which they can be consumed by Avro formatted event consumers, such as ESM and Database. Earlier versions of the SmartConnector were not capable of this and, as such, would send CEF formatted events to Transformation Hub that then needed to be transformed to Avro format in order to be consumed by Avro formatted event consumers. The following default CEF to Avro or C2AV transforming stream processors work to transform CEF data in the CEF source topic and route it to the dedicated Avro destination topic for use by Avro consumers.

1. The CEF-to-Avro stream processor transforms events from the th-cef topic to the th-arcsight-avro topic.
2. The CEF-to-Avro ESM Filtered Stream Processor transforms events from the mf-event-cef-esmfiltered topic to the mf-event-avro-esmfiltered topic. For more information about filtering events for ESM, see "Filtering Events for ESM" on page 820.

Enrichment Stream Processors

Introduced in Transformation Hub 3.5.0, an enrichment stream processor processes events coming from the selected source topic (by default, `th-arcsight-avro`) by executing enrichment tasks , which include generating a Global ID. Events are then routed to the topic `mf-event-avro-enriched`.

If you are enabling enrichment stream processors, ensure that the Generator ID is enabled.

Use the CDF Management Portal to configure the following aspects of the enrichment stream processor.

Number of enrichment stream processor groups: By default, Transformation Hub has 1 enrichment stream processor group with 2 instances enabled.

Source topic: Choose one of the following source topics according to your deployment needs.

- `th-arcsight-avro:` (default source topic) Use this topic for local ESM event enrichment when ESM is deployed.
- `mf-event-avro-esmfiltered`: Use this topic for global ESM event enrichment when ESM is deployed.

For more information on local and global ESM event enrichment, see below.

Global Event ID Enrichment: Transformation Hub ensures that all the events that passes through the Enrichment Stream processor have a global ID. If the event's global ID value is missing, then a new global ID is assigned to it.

Global Event ID generation enrichment is always enabled. You can also enable Event Integrity enrichment.

Event Integrity Enrichment (ArcSight Recon)

ArcSight Recon can check the integrity of event data, to provide assurance that event data sent by Connectors and other producers through the ingestion pipeline is not modified, and that events are not subsequently lost or deleted.

To achieve this objective, Transformation Hub provides *event integrity enrichment* that publishes summary events (such as M1 or agent:040 Connectors events), about messages that pass through the enrichment source topic. Each summary event will contain a calculated hash of data, a list of fields used to generate the hash, and list of the global event IDs of each message that is summarized. These three pieces of information will enable downstream consumers to verify that message data was not lost or modified.

It's important to tune the number of partitions of the enrichment stream processor source topic before enabling Integrity Events Enrichment. If you change the number of partitions of the source topic after enabling it, you must browse to Kafka Manager's Topics section and do the following:
1. Adjust and match the number of partitions of the Integrity events Enrichment changelog with the source topic number of partitions. The internal topic is named with the following format and pattern: `com.arcsight.th.AVRO_ENRICHMENT_1-integrityMessageStore-changelog.`
2. Restart the TH Web services pod by running the following command:
`kubectl delete pod th-web-service-xxxxxxxxx-yyyyy -n arcsight-installer-yyyyy`

Configuring Event Integrity Enrichment: You can configure event integrity while doing a fresh installation or during an upgrade. Set values of the following parameters accordingly:

- *Generate verification events for parsed field integrity checks*: (Default value: false) If true, a verification event is generated that accompanies a batch of events for checking the integrity of parsed fields in each event. Recon uses this verification event to check event integrity. If true, then specify a value for Verification event batch size as described below.
- *Verification event batch size (4-375)*: (Default value: 256) Specifies the number of events to be associated with a verification event. A lower value indicates fewer associated events need to be included in the batch for integrity checks; however, it will also result in higher resource consumption by generating more verification events.

Event integrity enrichment generates an internal topic named with the following format and pattern com.arcsight.th.AVRO_ENRICHMENT_1-integrityMessageStore-changelog. The setting "# of replicas assigned to each Kafka Topic" setting also applies to it.

If the flow of events is not consistent, and there are long intervals between the reception of events, the feature will check every hour (60 mins) for a summary event that hasn't reached the verification event batch size. If it hasn't been sent for more than 4 hours (240 mins), then it will be sent with the aggregated info of the previous number of events, regardless of whether it reached the verification event batch size.

For more information about verifying event data, see "Checking the Integrity of Event Data" in the User's Guide to ArcSight Recon.

Local and Global ESM Event Enrichment

ESM event enrichment can be configured locally or globally.

Local ESM Event Enrichment: With local ESM event enrichment (the default setting), ArcSight capabilities such as Recon and Intelligence can benefit from ESM Correlation. When local ESM event enrichment is configured:

- ESM reads the topic `mf-event-avro-esmfiltered`, enriches events found there, and stores them in ESM.
- ESM can be configured to send Correlation events to the `th-arcsight-avro` topic.
- Transformation Hub's Event Enrichment Stream Processor reads events from the `th-arcsight-avro` topic, enriches them, and sends them to `mf-event-avro-enriched` for Recon and Intelligence to read.

Global ESM Event Enrichment: With global event enrichment, events enriched by ESM are shared with all other ArcSight capabilities, including Recon and Intelligence. When global ESM event enrichment is configured:

- ESM reads the topic `th-arcsight-avro`, enriches events found there, and stores them in ESM.
- You must configure ESM to send all enriched events and Correlation events to the `mf-event-avro-esmfiltered` topic.
- Transformation Hub's Event Enrichment Stream Processor reads events from `mf-avro-esmfiltered`, enriches them, and sends them to `mf-event-avro-enriched` for Recon and Intelligence to read.

Configuring ESM Event Enrichment: To configure ESM event enrichment:

- For *local* ESM event enrichment, no configuration is needed by default.
- For *global* ESM event enrichment, in the CDF Management Portal, set the source topic for Enrichment Stream Processors to the `mf-event-avro-esmfiltered` topic.

Describing Routing

Each stream processor includes six processing threads. All routes with the same source topic are processed by one *routing stream processor group*. You can scale a processor group independently as load increases by adding more routing processor instances to the group.

You configure routing in ArcMC.

- The number of routing stream processor groups should match the number of source topics they are processing.
- Each routing stream processor group can contain multiple routing stream processors.
- You can configure up to 10 routing stream processor groups on Transformation Hub in the CDF Management Portal, allowing Transformation Hub to support up to 10 source topics.

Tuning Stream Processor Groups

The performance of stream processors is critical to Transformation Hub performance. In general, you can follow these guidelines for tuning stream processors and drive better performance.

- Since all routes which use the same source topic share the same routing stream processor group, adding more source topics can speed up processing.
- Increase the number of source topic partitions to handle high EPS throughput, depending on the CPU and memory resources of each worker node. For example, when the partition number is increased to 60, up to 10 routing (or C2AV) process instances can be used. Each stream processor uses 6 threads by default.
- Where possible, limit the number of routing rules per route.
- If stream processors display a `TimeoutException` in logs, consider overriding the application properties by slightly increasing the following settings, until the exceptions are no longer returned in logs:
  - `max.block.ms` (default is 60000 milliseconds)
  - `delivery.timeout.ms` (default is 120000 milliseconds)

Best Practices for Routing Stream Processors

The following best practices apply to management of routing stream processors.

- By default, Transformation Hub has 1 routing stream processor group. Accordingly, if you create 2 or more routes with different source topics, then make sure to enable more stream processor groups according to the number of source topics used in such routes (this applies to both type of routings: CEF-to-CEF or Avro-to-Avro).
- To enable and increase the number of instances of routing stream processor groups, in the CDF Management Portal, browse to the Reconfigure page. Identify the desired group number; and to enable it, just increase it from 0 to the desired value.
- To support high availability, routing stream processor groups can scale out and down partially. Once a group is enabled, you can increase or decrease the number of instances. However, it might never be reduced to 0, or the source topic mapped to that service group will no longer route until you increase the number of instances above 0.
- Always consider the available resources when enabling more routing stream processor groups.
- C2AV and routing stream processing in Transformation Hub are Kafka Streams applications. By default, Kafka Streams are using at-least-once processing guarantees in the presence of failure. This means that if the stream processing application fails, no data records are lost or will fail to be processed, but some data records maybe re-read and therefore reprocessed.

Therefore, C2AV and routing stream processing is using an at-least-once processing guarantees configuration. In this case, when C2AV/Routing pods are killed abnormally and restarted, the user might see duplicated events.

Stream Processor Deployment Guidelines

Effective stream processors (SPs) deployment is based on workload and some other considerations. Each SP requires CPU and memory, so the more SPs are deployed, the more system resources are used. If no stream processors are needed, then none should not be deployed, in order to conserve resources.

- Do not deploy more than one C2AV/Routing SP per node on small and medium nodes and no more than 3 C2AV/Routing SPs per node on large nodes, in co-located mode.

The table shows recommendations for deployment of SPs. These are recommendations and not requirements, but following the recommendations will avoid overloading a given system and still enable processing of the intended EPS:

Node type	EPS per Enrichment SP	Enrichment SP per worker node (Co-located)	Enrichment SP per worker node (Dedicated)	EPS per Routing SP	Routing SP per worker node (Co-located)	Routing SP per worker node (Dedicated)
Small (VM, 4 cores, 8cpus, 16G RAM)	Up to 10K	0	2	Up to 10K	0	2
Medium (VM, 8 cores, 16 vCPUs, 32G RAM)	Up to 35K	1	3	Up to 50K	1	3
Large (Appliance, 24 cores, 48 vCPUs, 188G RAM)	Up to 70K	4	4	Up to 90K	4	4

The EPS values shown on the table are based on the hardware mentioned on the **Node Type** column. Values shown here may vary depending on the actual performance of the VM/appliance, and the EPS performance of the stream processors may vary.

- On small nodes, we recommend deploying the enrichment SP and routing SP on dedicated nodes, as deploying them in a co-located fashion affects the overall EPS IN capacity of a small cluster.
- On medium nodes, avoid putting more than 1 enrichment SP in the same node. This will depend on how much EPS each enrichment SP has to process. The more EPS, the more CPU consumption there will be.

  - If using the default 2 enrichment SPs, they will be automatically assigned to different nodes.
  - If another SP instance is needed, then assign it to another node where no enrichment nor Routing SP is running.
- On medium nodes, the number of routing SPs deployed in a node follows the same logic as the enrichment SP mentioned above. There is a slight difference when it comes to performance. The more a routing SP needs to filter events, the more CPU consumption there will be. You may find a situation where a routing SP processing less than 1K might use more CPU than a routing SP processing 10K. This will depend on the size of the events and system load of the cluster, among other factors that might affect the overall performance of the routing SP instances.
- On medium nodes, if another enrichment SP/routing SP instance needs to be deployed on a co-located node, then make sure the target node is not running another enrichment SP/routing SP. You can ignore this recommendation for the enrichment SP, but only if the already running enrichment SP is processing less than 10K. Otherwise, just stick to 1 enrichment SP/Routing SP per co-located node.

Dedicated nodes (not running Kafka, Zookeeper or Fusion) enable deploying more enrichment SP and Routing SP instances.

Overriding Application Properties

Each Transformation Hub module (Kafka, Zookeeper, and so on) has many additional properties available, and there may be a need for system administrators to override the default values for some of these properties. This section covers how to override these property values.

Property values (for properties that support overrides) are set by injecting environment variables in the respective container's start-up environment. These variables are read from a user-supplied properties file, in a specific location on the Network File Server (NFS). To see the available properties for override, consult the respective module's published documentation.

Note that in most cases, this feature is not required for normal operation of Transformation Hub, and most likely will be used at the direction of technical support. Not all properties support overrides; please check with technical support before making any changes to your configuration.

- For Kafka, ZooKeeper, and Schema Registry properties, consult the appropriate Confluent documentation.
- The properties for routing processor and stream processor modules are detailed below.

Note: Legacy properties prefixed with `arcsight.eventbroker` will continue to function as they did in previous versions, but as explained below, newly added properties must be prefixed with `arcsight.th`. If two properties of the same name are set with different prefixes, the property with `arcsight.th` will supersede the other one.

Configuring the Values

1. Create a file named `arcsight-env-override.properties` under the folder `<NFS_root_DIRECTORY>/transformationhub/config`.

   The `<NFS_root_DIRECTORY>` path is described in this guide as the NFS root folder (usually `/opt/arcsight/nfs/volumes`). For more information, refer to the section "Creating the NFS Shares" on page 63.

2. Add properties to the file. To each property, add the module prefix from the table below.

Module	Prefix
Kafka	`arcsight.th.kafka.`
Schema Registry	`arcsight.th.schema-registry.`
ZooKeeper	`arcsight.th.zookeeper.`
Routing/C2AV/Enrichment Processor	`arcsight.th.sp.`

3. Delete the pods for which properties were defined, or, alternatively, redeploy Transformation Hub.
4. To verify the changes, search the log file (after the container's status is back to Running) for matching properties.

Routing Processor and Stream Processor Properties

As explained above, prefix these properties with `arcsight.th.sp.` to create an override.

Property Name	Default Value	Description
`RETRIES`	2147483647	The number of retries for broker requests that return a retry-able error.
`RETRY_BACKOFF_MS`	100	The amount of time (milliseconds), before a request is retried. This applies if the retries parameter is configured to be greater than 0.
`RECEIVE_BUFFER_BYTES`	65536	The size of the TCP receive buffer to use when reading data. If the value is -1, the OS default will be used.

Property Name	Default Value	Description
MAX_ PARTITION_ FETCH_BYTES	1048576	The maximum amount of data per-partition the server will return. Records are fetched in batches by the consumer.
MAX_REQUEST_ SIZE	1048576	The maximum size of a request in bytes.
BUFFER_MEMORY	33554432	The total bytes of memory the producer can use to buffer records waiting to be sent to the server.
BATCH_SIZE	16384	the default batch size in bytes when batching multiple records sent to a partition
LINGER_MS	100	the producer will wait for up to the given delay to allow other records to be sent so that the sends can be batched together
HEARTBEAT_ INTERVAL_MS	1000	The expected time (milliseconds) between heartbeats to the consumer coordinator when using Kafka's group management facilities. Heartbeats are used to ensure that the consumer's session stays active and to facilitate rebalancing when new consumers join or leave the group.
MAX_POLL_ INTERVAL_MS	3600000	The maximum delay (milliseconds) between invocations of poll() when using consumer group management
MAX_POLL_ RECORDS	100	The maximum number of records returned in a single call to poll().
SESSION_ TIMEOUT_MS	180000	The timeout (milliseconds) used to detect client failures when using Kafka's group management facility
REQUEST_ TIMEOUT_MS	305000	The configuration controls the maximum amount of time (milliseconds) the client will wait for the response of a request.
CONNECTIONS_ MAX_IDLE_MS	540000	The maximum amount of time (milliseconds) before idle connections are closed.
TH_NUM_ THREADS	6	The number of threads to execute stream processing.

Changing Value Examples

To change the value of ZOOKEEPER_MAX_CLIENT_CNXNS to 65, in ZooKeeper, and to change the value of SCHEMA_REGISTRY_KAFKASTORE_TIMEOUT_MS in the Schema Registry, create a file, <NFS Volume mount>/transformationhub/config/arcsight-env-override.properties, and add the following lines:

```
arcsight.th.zookeeper.ZOOKEEPER_MAX_CLIENT_CNXNS=65
```

```
arcsight.th.schema-registry.SCHEMA_REGISTRY_KAFKASTORE_TIMEOUT_MS=20000
```

Example of verifying the change by searching the log:

```
kubectl -n transformationhub1 logs th-zookeeper-0 | grep ZOOKEEPER_MAX_CLIENT_
CNXNS
Environment override script set: ZOOKEEPER_MAX_CLIENT_CNXNS=65
ZOOKEEPER_MAX_CLIENT_CNXNS=65
```

Transformation Hub Liveness Probes

A *liveness probe* is a Kubernetes feature that can be configured to detect problematic pods. Once detected, Kubernetes will take action to restart a problematic pod. Liveness probes help ensure higher availability of pods as well as a more robust cluster environment. Consult the Kubernetes documentation for a more detailed explanation of liveness probes. Transformation Hub supports these liveness probe types:

- TCP/IP port-socket connection
- HTTP request
- Log scanning

Each container or pod supports the listed liveness probes, with their default parameter values shown.

Container/Pod	Probe	initialDelaySeconds	periodSeconds	timeoutSeconds	failureThreshold
Kafka	tcp socket :9092 and log scanning	240	60	30	3
Zookeeper	tcp socket :2181 and log scanning	240	60	30	3
Web Service	https GET :8080 and log scanning	240	300	30	3
Schema Registry	https GET :8081 config and log scanning	240	300	30	3
Kafka Manager	https GET :9000 and log scanning	240	600	30	3
Routing/Enrichment Processor	log scanning	240	60	30	3
C2AV (CEF-to-Avro) Processor	log scanning	240	60	30	3

Probe parameters are defined as follows:

Parameter	Definition
initialDelaySeconds	Number of seconds after the container has started before liveness probes are initiated. The first probe execution after startup is not until initialDelaySeconds + periodSeconds.
periodSeconds	How often to perform the probe.

Parameter	Definition
`timeoutSeconds`	Number of seconds after which the probe times out.
`failureThreshold`	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up and restarting the pod.

Managing Liveness Probes

To check if a pod has a liveness probe configured:

1. Run:
   `kubectl -n <namespace> describe pod <podname>`
2. Review the output. Look (or `grep`) for the line starting with the string `Liveness...`This will show some of the probe's configuration.

To check for probe failures:

1. Run:
   `kubectl get pods --all-namespaces`
2. If any pod shows 1 or more restarts, run:
   `kubectl -n <namespace> describe pod <podname>`
3. Review any list of events at the end of the output. Liveness probe failures will be shown here.

Configuring Liveness Probes

The default values for liveness probes can be overriden by changing the values of the appropriate properties on the Configuration page.

1. Log in to the CDF Management Portal.
2. Click **Administration**.
3. Click the **...** (Browse) icon to the right of the main window.
4. From the drop-down, click **Reconfigure**. The post-deployment settings page is displayed.
5. Browse the configuration properties list to find the desired property, and specify the new value.
5. Click **Save**.

Configuring Log Scanning Liveness Probes

Log scanning probes scan the application's output for a match to a configured pattern, such as a known error message. If the pattern is found, the pod is restarted.

In addition to the four parameters described in the table above, log scanning probes have two additional properties:

literal	A literal expression for matching against the application's log output.
regex	A regular expression for matching against the application's log output.

- The *literal* property specifies a literal (exact match) search string. If the value matches a portion of the log text, the liveness probe, on its next periodic check, will report a failure and restart the pod.
- The *regex* property is similar, except that a regular expression can be specified for the match. This regex must conform to Java regex rules. To specify a regex escape value within the regex, use 2 backslashes to escape it (`\\`).
- Multiple search patterns can be specified per property, separated by 4 vertical bars (`||||`). A match on any of the patterns will trigger the probe failure.
- There are no default values for these parameters. Log scanning is disabled in the default configuration.
- Matching across multiple rows is not supported. The match must occur on one log line.
- For example, to restart the CEF-to-Avro Routing Stream Processor pod when the value, `Setting stream threads to d` (where `d` could be any single digit), is found in the log, change the configuration property "CEF-to-Avro Routing Stream Processor liveness probes regular expression" to the following value .

```
   Setting stream threads to \\d
```

Verification

To verify that log scanning is configured as intended, review the pod's log and look for entries containing `InputStreamScanner`.

For example, to view the c2av-processor pod log, run:

```
   kubectl -n <namespace> logs th-c2av-processor-0 | more
```

For the previous property example, the corresponding log line would be:

```
   InputStreamScanner: Will scan for RegEx pattern [Setting stream threads to
\d]
```

Backing Up and Restoring

This section provides information about backing up and restoring data and configurations for the following components:

Backing Up and Restoring Configuration Data for Deployed Capabilities

For more information on backing up and restoring Event Data, see Backing Up and Restoring the Arcsight Database

Certain components deployed on the ArcSight Platform use NFS (or EFS when deployed to AWS) to store some of their data, such as Fusion credentials, dashboard widgets, and search preferences. You can configure automatic backups of NFS or EFS as a protection in the event of data corruption or loss. You can restore the backed up data at any time to the same system or a separate one per your requirements. In the event of data corruption or loss, you can use the backed up data and roll back to an available and known good restore point. Backups are carried out at two levels: pod and NFS.

Because data stores are mounted to */<Server mount path>/arcsight-volume*, do not use this same storage device for your NFS backups. Use a local folder on the system or a remote location.

- "Backing Up and Restoring Configuration Data for Deployed Capabilities" above
- "NFS Level Backup" on the next page

Understanding How Pod-level Backup Occurs

Each pod backs up its own data on an hourly basis, placing the data in a **backup staging directory** related to the pod's mount location under */<Server mount path>*/arcsight-volume. This automated backup process ensures that the pod stores a backup of its data in a complete state. The pod retains a maximum of 24 backups in the staging directory. At any given hour, the oldest backup is deleted and a fresh one created. Because the pod backup is stored in the same NFS volume as the pod content, this pod -level backup does not protect you against data loss or corruption of the volume. Thus, you should configure an NFS level backup to ensure data continuity.

The following table lists the impacted pods for each capability:

Capability Name	Pod Name
Fusion	• fusion-metadata-rethinkdb-*-* • fusion-user-management-*-* • fusion-single-sign-on-*-* • fusion-dashboard-web-app-*-* • fusion-metadata-web-app-*-* • fusion-arcmc-web-app-*-* • soar-web-app -*-* • soar-message-broker -*-* • soar-widgets-*-*
Intelligence	• h2-*-* • intelligence-arcsightconnector-api-*-* • intelligence-tenant-control-*-* • intelligence-tuning-api-*-* • interset-analytics-*-* • interset-api-*-* • interset-ui-*-* • searchmanager-api-*-* • searchmanager-engine-*-*
Transformation Hub	• th-c2av-processor-*.* • th-c2av-processor-esm-*.* • th-enrichment-processor-group*-* • th-kafka-*.* • th-kafka-manager-*-* • th-routing-processor-group*-* • th-schemaregistry-*-* • th-web-service-*-* • th-zookeeper-*.*

NFS Level Backup

You must configure an NFS level backup to store the pod-level backup in a reliable, external storage system. For the backup, you must configure a scheduled job to back up the backup staging directory updated by the pod-level backups. The schedule should be less frequent than the hourly interval for pod-level backups.

To set up the NFS level backup or to restore the data, see the following sections based on your environment:

Backing Up and Restoring Configuration Data On AWS

You can use the following information to back up and restore configuration data for deployed capabilities.

Backing Up Configuration Data On AWS

You can back up the configuration data for the deployed capabilities and ArcSight Platform components.

1. Log in to the AWS Backup console at https://console.aws.amazon.com/backup.
2. Attach the *AWSBackupFullAccess* policy to your IAM role.

   For more information, see *Managed policies for AWS Backup* in the AWS documentation.
3. Create an AWS backup plan for the EFS that you created during installation.

   For more information, see *Creating a backup plan* in the AWS Documentation.
4. Follow the onscreen prompts to complete the backup plan for backups at intervals of your choice with appropriate retention periods.
5. After backup creation, in the left-hand pane select **Protected resources** to view a list of Resource IDs and Resource Types.
6. To view associated backups available, click a Resource ID.

Restoring Configuration Data On AWS

When restoring data stores, retain the original directory structure and the pod-level sub-directory structure:

```
/<efs_server mount path>/arcsight-volume
```

The default EFS server mount path is /mnt/efs/*<efs parent folder>*

1. Ensure that you have valid AWS backups.
2. Log in to the AWS Backup console at https://console.aws.amazon.com/backup.

3. In the left-hand pane, select **Protected resources** to view a list of resources designated by Resource Type and Resource ID.
4. To view the available Recovery point IDs, select a given Resource ID.
5. Select a Recovery point ID and then click **Restore**.
6. Keep default values on the page and then click **Restore backup**.
7. In the left-hand pane, drill down to **Jobs** > **Restore Jobs** to view the restore job status.

   The Restore Job creates a new Recovery directory off of the root of the EFS with the recovered contents preserving the original path hierarchy.
8. On the bastion machine used for your AWS setup, run the following command:

   ```
   cd <INSTALLER_LOCATION>/aws-scripts/scripts
   ```
9. To view the restore script options, execute the following command:

   ```
   ./nfs-arcsight-volume-restore.sh -h
   ```

   Use the following parameters:

   -r | --restore-dir
   Specify the AWS restore directory name created in the mount location. This parameter is mandatory.

   -o | --older-backup
   Available pod backups. This parameter is optional.

   -p | --path
   Specify the EFS mount path. This parameter is mandatory.

   -h | --help
   Displays the command options.
10. To restore to the latest pod backup, execute the following command:

    ```
    ./nfs-arcsight-volume-restore.sh -r <restore_dir_in_mount_location> -p
    <mount_path>
    ```

    For example:

    ```
    ./nfs-arcsight-volume-restore.sh -r aws-backup-restore_<TIMESTAMP> -p
    arcsight
    ```

    To restore to an earlier pod backup, use the -o parameter from the Usage Options.
11. To complete the restore process, follow the onscreen instructions.
12. (Conditional) If Transformation Hub is deployed, complete the following steps:

a. Mount and navigate to the EFS backup location.

b. Navigate to the Transformation Hub directory.

   For example:

   ```
   /<efs mount location>/arcsight-volume/transformationhub/config/
   ```

c. Ensure that the *arcsight-volume* is mounted, then navigate to `/transformationhub/config/`.

d. If the file `arcsight-env-override.properties` exists in the backup location (12b), copy it to the *arcsight-volume* directory (12c) and then remove any file properties that do not apply to the restored environment.

13. To get the names of pods to restart, execute the following command:

    ```
    kubectl get pods -n $( kubectl get namespaces | grep arcsight | cut -d '
    ' -f1)
    ```

    Compare the output with the impacted pods listed in the table on page, to know pods names.

14. To restart pods listed in this table, execute the following command:

    ```
    kubectl delete pods -n $( kubectl get namespaces | grep arcsight | cut -d ' ' -
    f1) <space separated pod names>
    ```

    For example:

    ```
    kubectl delete pods -n $( kubectl get namespaces | grep arcsight | cut -d
    ' ' -f1) fusion-user-management-56497c76bb-mdmmz fusion-dashboard-web-
    app-7b864467d5-d2c8v fusion-metadata-rethinkdb-5c69c77756-hxxzg
    ```

15. Remove the recovery directory restored from AWS to the default mount location in step 7 above.

    ```
    cd <default mount path>
    ```

    ```
    sudo rm -rf aws-backup-restore_<TIMESTAMP>
    ```

16. Ensure that all pods display a running status:

    ```
    kubectl get pods --all-namespaces
    ```

17. To verify restored data stores, log in to the associated application.

Backing Up and Restoring Configuration Data On Azure

You can use the following information to back up and restore configuration data for deployed capabilities.

- Backing Up Configuration Data On Azure
- Restoring Configuration Data On Azure

Backing Up Configuration Data On Azure

You can back up the configuration data for the deployed capabilities and ArcSight Platform components.

This procedure explains one possible approach for performing a backup. If you have your own managed backup system and prefer to use it, you can configure it to perform a backup of the *arcsight- volume* instead of using the approach described here.

A maximum of 15 backups folders are available on a given day.

1. SSH to your jump host and become root.
2. Unzip the azure-scripts in the installer location and then navigate to the following location where the backup script resides:

   ```
   cd <INSTALLER_LOCATION>/azure-scripts
   ```

3. Execute the following command to view backup script options:

   ```
   ./nfs-arcsight-volume-backup.sh -h
   ```

   Use the following parameters:

   -s | --source
   Source mount path without the 'arcsight-volume'. This can be either an external or local NFS server mount path. This parameter is mandatory.

   -d | --destination
   Destination path where the NFS backup is to be located. This can be either an external or local NFS server mount path. If not specified, the default location is /nfs/nfs-backup/. This parameter is optional.

   -h | --help
   Displays the command options.

4. Ensure that your jump host has enough space or point to an external NFS server as the destination.
5. To create a NFS backup at an interval of your choice, execute the following command:

```
(crontab -l 2>/dev/null; echo "0 0 * * * <installer_location>/azure-scripts/nfs-arcsight-volume-backup.sh -s <NFS_server:mount_path>")| crontab -
```

Restoring Configuration Data On Azure

When restoring data stores, retain the original directory structure and the pod-level sub-directory structure:

```
/<NFS_server mount path>/arcsight-volume
```

1. Ensure that you have a valid data store backup.
2. Navigate to the following location where the restore script resides:

```
cd <INSTALLER_LOCATION>/azure-scripts
```

3. To view the restore script options, execute the following command:

```
./nfs-arcsight-volume-restore.sh -h
```

Use the following parameters:

-o | --older-backup
Available pod backups. This parameter is optional.

-r | --restore-dir
Available nfs backups. This parameter is optional.

-s | --source
Source mount path of the NFS backup location. This can be either an external or local NFS server mount path. This parameter is mandatory.

-d | --destination
Destination path without the 'arcsight-volume' where the NFS backup is to be restored. This can be either an external or local NFS server mount path. This parameter is mandatory.

-h | --help
Displays the command options.

4. (Conditional) If you restore from your own managed backup system, execute the restore script as follows:

a. Parameter **-s** to specify a source mount path one level above *arcsight-volume*

b. Parameter **-r** to list available sub directories therein that includes *arcsight-volume*

c. Select the index value for *arcsight-volume* to proceed with restore.

5. To restore to the latest NFS backup, execute the following command:

```
./nfs-arcsight-volume-restore.sh -s <NFS_server:mount_path> -d <NFS_
server:mount_path>
```

For -o or -r as parameters, backup index values are made available to choose from upon command execution.

6. To restore from the listed index values, choose an available backup.
7. To complete the restore process, follow the onscreen instructions.
8. (Conditional) If Transformation Hub is deployed, complete the following steps:

a. Mount and navigate to the nfs backup location.

b. Navigate to the Transformation Hub directory.

For example:

```
/<nfs mount location>/<time stamped backup
directory>/transformationhub/config/
```

c. Ensure that the *arcsight-volume* is mounted, then navigate to `/transformationhub/config/`.

d. (Conditional) If the file `arcsight-env-override.properties` exists in the backup location (8b), copy it to the *arcsight-volume* directory (8c), and then remove any file properties that do not apply to the restored environment.

9. To get the names of pods to restart, execute the following command:

```
kubectl get pods -n $( kubectl get namespaces | grep arcsight | cut -d '
' -f1
```

Compare the output with the impacted pods listed in the table on page, to know pods names.

10. To restart pods listed in this table, execute the following command:

```
kubectl delete pods -n $( kubectl get namespaces | grep arcsight | cut -d ' ' -
f1) <space separated pod names>
```

For example:

```
kubectl delete pods -n $( kubectl get namespaces | grep arcsight | cut -d
' ' -f1) fusion-user-management-56497c76bb-mdmmz fusion-dashboard-web-
app-7b864467d5-d2c8v fusion-metadata-rethinkdb-5c69c77756-hxxzg
```

11. Ensure that all Pods display a running status:

```
kubectl get pods --all-namespaces
```

12. To verify restored data stores, log in to the associated application.

Backing Up and Restoring Configuration Data On-Premises

To backup or restore configuration data for deployed capabilities, use the following procedures:

- Backing Up Configuration Data On-Premises
- Restoring Configuration Data On-Premises

Backing Up Configuration Data On-Premises

You can back up the configuration data for the deployed capabilities and ArcSight Platform components.

This procedure explains one possible approach for performing a backup. If you have your own managed backup system and prefer to use it, you can configure it to perform a backup of the *arcsight- volume* instead of using the approach described here.

A maximum of 15 backups folders are available on a given day.

1. SSH to your jump host and become root.
2. Navigate to the following location where the backup script resides:

```
cd <INSTALLER_LOCATION>/installers/cdf/scripts
```

3. Execute the following command to view backup script options:

```
./nfs-arcsight-volume-backup.sh -h
```

Use the following parameters:

-s | --source
Source mount path without the 'arcsight-volume'. This can be either an external or local NFS server mount path. This parameter is mandatory.

-d | --destination
Destination path where the NFS backup is to be located. This can be either an external or local NFS server mount path. If not specified, the default location is /nfs/nfs-backup/. This parameter is optional.

-h | --help
Displays the command options.

4. Execute the following command to create a NFS backup at an interval of your choice; here daily:

```
(crontab -l 2>/dev/null; echo "0 0 * * * <installer_
location>/installers/cdf/script/nfs-arcsight-volume-backup.sh -s <NFS_
server:mount_path>")| crontab -
```

Restoring Configuration Data On-Premises

When restoring data stores, retain the original directory structure and the pod-level sub-directory structure:

```
/<NFS_server mount path>/arcsight-volume
```

1. Ensure that you have a valid data stores backup.
2. Navigate to the following location where the restore script resides:

```
cd <INSTALLER_LOCATION>/installers/cdf/scripts
```

3. To view the restore script options, execute the following command:

```
./nfs-arcsight-volume-restore.sh -h
```

Use the following parameters:

-o | --older-backup
Available pod backups. This parameter is optional.

-r | --restore-dir
Available nfs backups. This parameter is optional.

-s | --source
Source mount path of the NFS backup location. This can be either an external or local NFS server mount path. This parameter is mandatory.

-d | --destination
Destination path without the 'arcsight-volume' where the NFS backup is to be restored. This can be either an external or local NFS server mount path. This parameter is mandatory.

-h | --help
Displays the command options.

4. (Conditional) If you restore from your own managed backup system, execute the restore script as follows:
   a. Parameter **-s** to specify a source mount path one level above *arcsight-volume*
   b. Parameter **-r** to list available sub directories therein that includes *arcsight-volume*
   c. Select the index value for *arcsight-volume* to proceed with restore.
5. To restore to the latest NFS backup, execute the following command:

```
./nfs-arcsight-volume-restore.sh -s /nfs/nfs-backup/ -d <NFS_
server:mount_path>
```

For -o or -r as parameters, backup index values are made available to choose from upon command execution.

6. Follow the onscreen instructions to complete the restore process as appropriate.
7. (Conditional) If Transformation Hub is deployed, complete the following steps:
   a. Mount and navigate to the nfs backup location.
   b. Navigate to the Transformation Hub directory.
      For example:

```
/<nfs mount location>/<time stamped backup
directory>/transformationhub/config/
```

   c. Ensure that the *arcsight-volume* is mounted, then navigate to `/transformationhub/config/`.
   d. (Conditional) If the file `arcsight-env-override.properties` exists in the backup location (7b), copy it to the *arcsight-volume* directory (7c), and then remove any file properties that do not apply to the restored environment.
8. To get the names of pods to restart, execute the following command:

```
kubectl get pods -n $( kubectl get namespaces | grep arcsight | cut -d '
' -f1)
```

Compare the output with the impacted pods listed in the table on page to know pods names.

9. To restart pods listed in this table, execute the following command:

```
kubectl delete pods -n $( kubectl get namespaces | grep arcsight | cut -d ' ' -f1) <space separated impacted pod names>
```

   For example:

```
kubectl delete pods -n $( kubectl get namespaces | grep arcsight | cut -d ' ' -f1) fusion-user-management-56497c76bb-mdmmz fusion-dashboard-web-app-7b864467d5-d2c8v fusion-metadata-rethinkdb-5c69c77756-hxxzg
```

10. Ensure that all pods display a running status:

```
kubectl get pods --all-namespaces
```

11. To verify restored data stores, log in to the associated application.

Backing Up and Restoring the Arcsight Database

You can configure automatic backups of the Arcsight Database to protect against data loss or corruption. The backed up data can be restored anytime to the same or separate system as per your requirement.

The database uses a single communal storage location for all data and for the catalog (metadata). Communal storage is the database's centralized storage location, shared among the database nodes. When you perform a backup, data gets copied to a backup communal storage location that replicates the live communal storage.

- Backing Up the Database
- Restoring the Database
- Managing Your Backups
- Preparing for a Database Recovery

Backing Up the Database

You can manually create or automatically schedule a backup of the database catalog. Follow the steps below to successfully back up the database.

- Backup Overview
- Understanding Backup Terminology
- Prerequisites to Configuring Database Backup
- Preparing the Backup Configuration File
- Backing Up the Database
- Scheduling Automatic Backups

Backup Overview

You can perform a full backup, which is a complete copy of the database catalog, its schemas, tables, and other objects. It provides a snapshot of the database at the time of backup. You can use it for disaster recovery or to restore a damaged or an incomplete database. You can also restore individual objects from a full backup.

If a full backup already exists, then the database backup utility tool backs up new or changed data from the time the full backup was created. You can specify the number of backup snapshots to retain.

Understanding Backup Terminology

- Backups are stored in the following folders in the backup location:
  - **Object Folder:** Consists of database objects files, which contain the actual data stored in the database. Repeated backups copy the new objects that are not in the backup location.
  - **Snapshot folder**: It contains a snapshot of the full catalog of the database at the time of the backup. Catalog contains metadata which is smaller in size than the actual data in the database. Catalog keeps track of all the database objects that were present in the database at the time of the backup snapshot. Many Catalog snapshots will refer to the same object files as the backups are performed more often than the lifespan of the object file. This avoids storing duplicates of object files for each backup. The backup_snapshot portion is defined by the .ini file and the date time strings are automatically appended by the database backup process.
- **Restore point:** Each backup operation records the state of the database at the time of the backup and stores it in the backup archive as a restore point. You can restore to a specific restore point using the `-archive` argument.
- **Restore point limit:** Specifies the number of previous backups that you want to retain in addition to the most recent backup.
- In the backup utility configuration file, you can specify the number of backup snapshots to be retained using **Specify the number of historical backups to retain in addition to the most recent backup**, so that the expired snapshots can be groomed out. When a backup snapshot is groomed out, all associated object files that was being referenced by the snapshot will also be groomed out.

Prerequisites to Configuring Database Backup

Before you configure the database backup, ensure your cloud administrator creates the communal storage backup location.

For an AWS Environment

For an AWS-based deployment, the backup communal storage location must be in the same region as the live database communal storage. The database supports connecting to S3 buckets in AWS using IAM roles. IAM roles are the default access control method for AWS resources. The database uses this method if you do not configure the legacy access control session parameters.

To use an IAM role, the bucket must be in the same region as the mode's database cluster and the role needs to be set with the proper permissions for reading and writing to the S3 bucket. For more information about creating Amazon S3 buckets, see the AWS documentation, Creating a bucket. For more information about IAM roles, see the AWS documentation, IAM Roles for Amazon EC2 and Creating a role to delegate permissions to an AWS service.

For an Azure Environment

The backup communal storage location must be in the same Azure Storage Account as the live database communal storage.

Preparing the Backup Configuration File

A database backup utility is provided to be used to perform backup and restore procedures. To use this utility, it must first be configured. Once configured, it can be used to perform the complete lifecycle of scheduling backups, backup on-demand, managing the backup archive, and restoring from backup.

You must create an S3 bucket or a Blob storage backup folder before configuring the database backup utility.

Run this tool as a root user.

1. Specify the following command from the database scripts path (/opt/arcsight-db-tools/scripts):

   ```
   ./db_backup.sh config
   ```

2. Select the communal storage.
3. Specify the values for the fields based on your requirement. Following are the possible scenarios:
   - For S3 storage:

Scenario	Fields
Using IAM role authentication and S3 settings are saved	a. Enter y to use the IAM role authentication, when prompted. b. Specify the S3 backup bucket name. c. Specify the path to S3 backup folder. d. Specify the path to the locking system. e. Specify the number of historical backups to be retained in addition to the most recent backup.

Not using IAM role authentication and S3 settings are saved	a. Enter n to not use the IAM role authentication, when prompted. b. Specify the S3 server access key. c. Specify the S3 server password. d. Specify the S3 backup server. e. Specify the S3 server backup port. f. Specify if TLS needs to be enabled or disabled. g. Specify the S3 backup server access key. h. Specify the S3 backup server password. i. Specify the S3 backup bucket name. j. Specify the path to S3 backup folder. k. Specify the path to the locking system. l. Specify the number of historical backups to be retained in addition to the most recent backup.
Using IAM role authentication and S3 settings are not saved	a. Enter y to use the IAM role authentication, when prompted. b. Specify the S3 server. c. Specify the S3 server port. d. Specify if TLS needs to be enabled or disabled. e. Specify the S3 backup bucket name. f. Specify the path to S3 backup folder. g. Specify the path to the locking system. h. Specify the number of historical backups to be retained in addition to the most recent backup.
Not using IAM role authentication and S3 settings are not saved	a. Enter n to not use the IAM role authentication, when prompted. b. Specify the S3 server. c. Specify the S3 server port. d. Specify if TLS needs to be enabled or disabled. e. Specify the S3 server access key. f. Specify the S3 server password. g. Specify the S3 backup server. h. Specify the S3 server backup port. i. Specify if TLS needs to be enabled or disabled for the backup server. j. Specify the S3 backup server access key. k. Specify the S3 backup server password. l. Specify the S3 backup bucket name. m. Specify the path to S3 backup folder. n. Specify the path to the locking system. o. Specify the number of historical backups to be retained in addition to the most recent backup.

- For Blob storage:

  The Blob account used for backup must be the same account as used for the live database, otherwise you will observe an error.

Using managed identity to authenticate	a. Specify the account name. b. Enter y to use the managed identity to authenticate with Azure storage container. c. Specify the path to the Blob storage backup folder. d. Specify the path to the locking system. e. Specify the number of historical backups to be retained in addition to the most recent backup.
Not using managed identity to authenticate	a. Specify the account name. b. Enter n to not use the managed identity to authenticate with Azure storage container. c. Specify the account key. d. Specify the path to the Blob storage backup folder. e. Specify the path to the locking system. f. Specify the number of historical backups to be retained in addition to the most recent backup.

Backing Up the Database

To create a new backup, run the following command from the database scripts path (/opt/arcsight-db-tools/scripts):

```
./db_backup.sh backup
```

Scheduling Automatic Backups

Micro Focus recommends that you schedule backups to run every hour. To schedule a backup, use the following command from the database scripts path (/opt/arcsight-db-tools/scripts):

```
./db_backup.sh schedule '<crontab_expression>'
```

For example:

```
./db_backup.sh schedule '0 * * * *'
```

Required parameter:

- crontab_expression

  Specify a crontab expression for the time you want to schedule a backup.

Restoring the Database

You can use the following information to restore a backed up database. This section has the following topics:

- Prerequisites for Restoring the Database
- Restoring a Backup

Prerequisites for Restoring the Database

Before restoring a backup, make a note of the following requirements:

- The database name must match the database name in the backup.
- The number of nodes in the primary subcluster must be equal to the number of nodes that were present in the primary subcluster at the time the backup was taken.
- Database node names must match the names of nodes in the backup.
- Use the same catalog directory location that was used in the database when the backup was taken.
- Use the same port numbers that were used by the database when the backup was taken.
- For object restore, have the same shard subscriptions. If the shard subscriptions have changed, then you can only perform full restore. Shard subscriptions can change when you add or remove nodes or rebalance the cluster.
- The database cannot be restored while it is still running. Run *db_installer.sh* to stop, start, or check the status of the database. If you must stop the database, also run *./scripts/watchdog.sh disable* to disable the watchdog.

You must stop the database to perform a full restore. Run `db_installer.sh stop-db` to stop the database. However, the database must be running to perform an object restore. Run `db_installer.sh start-db` to start the database.

Restoring a Backup

You can restore a full or object backup of a database that has primary and secondary subclusters to a new (target) database. The target database can have both primary and secondary subclusters. However, the backup is restored only to the primary subclusters of the target database.

To restore a backup, use following command from the database scripts path (/opt/arcsight-db-tools/scripts):

```
./db_backup.sh restore
```

You can use the following parameters:

`--archive=<timestamp_value>`	To specify a timestamp of the backup that you want to restore. For example: `./db_backup.sh restore --archive=20211006_205934`
`--restore-objects=<objects>`	To specify the individual objects you want to restore from a full or object-level backup. If you are using wildcards, then use `--include-objects` and `--exclude-objects` instead. For example: `./db_backup.sh restore --restore-objects=default_secops_adm` This parameter is invalid in combination with parameters `--include-objects` and `--exclude-objects`.
`--include-objects=<objects>`	To specify database objects or pattern of objects to restore from a full or object-level backup. Use comma to delimit multiple objects and wildcard patterns. For example: `./db_backup.sh restore --include-objects=default_secops_adm` You cannot use this parameter with `--restore-objects` parameter.
`--exclude-objects=<objects>`	Used along with `--include-objects` option, to specify database objects or pattern of objects you want to remove from the set. Use comma to delimit multiple objects and wildcard patterns. For example: `./db_backup.sh restore --include-objects=default_adm --exclude-objects=default_secops_adm` You cannot use this parameter with `--restore-objects` parameter.

After restore completes, execute the restart commands:

```
./db_installer start-db
./kafka_scheduler start
./scripts/watchdog.sh enable
```

Managing Your Backups

You can use the following information to manage your backups. This section has the following topics:

- Viewing Available Backups
- Quick-Check Backup
- Full-Check Backup
- Deleting a Backup
- Disabling Scheduled Automatic Backups

Viewing Available Backups

To view all the available backups, use the following command from the database scripts path (/opt/arcsight-db-tools/scripts):

```
./db_backup.sh list
```

Quick-Check Backup

You can collect all backup metadata from the backup location specified in the configuration file and compare that metadata to the backup manifest using the following command from the database scripts path (/opt/arcsight-db-tools/scripts):

```
./db_backup.sh quick-check
```

Full-Check Backup

Verify all objects listed in the backup manifest against the filesystem metadata using the following command from the database scripts path (/opt/arcsight-db-tools/scripts):

```
./db_backup.sh full-check
```

Available options:

--report-file=<path or a file name>

Full-Check also includes the steps of Quick-Check.

Deleting a Backup

To delete a backup, use the following command from the database scripts path (/opt/arcsight-db-tools/scripts):

```
./db_backup.sh remove --archive=<timestamp>
```

For example:

```
./db_backup.sh remove --archive=20211006_205934
```

Required options:

- `--archive=<timestamp>`
  To specify a timestamp of the backup you want to remove.

Disabling Scheduled Automatic Backups

To remove a job that runs scheduled backup, use following command from the database scripts path (/opt/arcsight-db-tools/scripts):

```
./db_backup.sh unschedule
```

Backing Up and Restoring the Postgres Database

Some components that are deployed on the ArcSight Platform utilize an embedded Postgres database that is also deployed on the platform. You can configure automatic backups of the Postgres database as a protection in the case of data loss or corruption. The backed up data can be restored anytime to the same or separate system as per your requirement.

Registering the Deployed Capabilities

As part of the installation process, the Management Portal registers the deployed capabilities and the AutoPass function by default. Before backing up the configuration data, though, you should ensure that the capabilities have been registered.

You have to register the capabilty only once and need not to repeat the registration, each time you take the backup.

To register the capability, complete the following steps:

1. "Generate the IDM Token " below
2. "Verifying Application Registration " on the next page
3. "Registering a Capability" on page 857

Generate the IDM Token

The backup service requires an IDM token to authorize the capabilities that you want to register.

1. SSH to the CDF master node.
2. Use **integration_admin** as username and run the following command to fetch the **password**:

   ```
   kubectl exec -it -n core $(kubectl get pods -n core | grep itom-
   postgresql | awk '{print $1}') -c itom-postgresql -- get_secret idm_
   integration_admin_password | cut -d '=' -f2-
   ```

3. Run:

```
curl -k -X POST --data '{
"passwordCredentials": {
"password": "<password>",
"username": "<username>"
},
"tenantName": "provider"
}' -H 'content-type:application/json' https://<itom-management-portal-fqdn>:5443/suiteInstaller/urest/v1.1/tokens
```

Verifying Application Registration

You do not have to re-register the application if it is currently registered. To check which capabilities have been registered, run the following command:

1.
```
curl -k -v -H "Accept: application/json" --header
"X-Auth-Token: <Token>" -X
GET https://<itom-pg-backup-pod-IP>:8443/backupd/api/v1/registry/applications
```

2. Example Output:

```
{
  "_links": {
"self": {
     "href": "/backupd/api/v1/registry/applications",
"class": "collection"
    },
```

```
    "items": [
```

```
{
```

```
"href": "/backupd/api/v1/registry/applications/itom-core",
```

```
"title": "itom-core"
```

```
}
```

```
]
```

```
}
```

```
}
```

Registering a Capability

If the capabilities are not displayed in the registered capabilities list "Verifying Application Registration " on the previous page, then you must register the capability.

1. Run and input the requested information as shown below:

```
curl -k -X PUT --data '
```

```
{
```

```
"services":{
```

```
"<postgresql-server-container-hostname>":{
```

```
"userName":"<application-db-owner-username>",
```

```
"passwordKey":"<application-db-owner-passwordKey-on Vault>",
```

```
"port":"5432",
```

```
"dbName":"<application-db-name>",
```

```
"type":"backup.type.postgres",
```

```
"secure":true}}}'
```

```
-H 'content-type:application/json' -H "Accept: application/json" -H "X-Auth-Token:
```

```
<IDM Token>" https://itom-pg-backupd-pod-IP:8443/backupd/api/v1/registry/applications/<application-name>
```

Where:

- **PostgreSQL Server Container Hostname:** Hostname of the PostgreSQL server container used by the application.
- **Application Database Owner Username:** Application database user name. If the application's database password is not kept in the vault, "postgres" can be used.
- **Application Database Owner PasswordKey on Vault:** Application database password key name in the vault. If the application's database password is not kept in the vault, "defaultdb_user_password" can be used.
- **Application Database Name:** Name of the application database.
- **Secure:** When set to true, the connection to the PostgreSQL service is in secure mode. This is an optional parameter but it should always be set to true to ensure security.
- **ITOM-PG-Backup-Pod-IP:** `kubectl get pods -A -o wide|grep itom-pg-backup` determines the pod's IP address.
- **Application Name:** Descriptive name of the application.

The following applications should be registered for the ArcSight Platform:

Application Name	Time to register	PostgreSQL Server Container Hostname	DB Name	DB User	DB Password Key	Pod Names for Restore Instructions
Autopass	Always	default-postgresql-svc.core	defaultdbapsdb	postgres	defaultdb_user_password	autopass-lm
SOAR	When SOAR is deployed	itom-postgresql.core	soar	postgres	ITOM_DB_DEFAULT_PASSWD_KEY	soar-web-app

Backing up a Capability's Configuration Data

This section presents the data backup process for ArcSight Capabilities on CDF. You can configure all ArcSight capabilities to back up their configuration data on a regular basis. The backed up data can be restored when needed.

Starting the Backup

Use the **db_admin.sh**, backup-restore script, located in the `/opt/arcsight/kubernetes/tools/postgres-backup/` directory to back up data.

Note: Before proceeding with backup steps, ensure that the SOAR UI is accessible or SOAR heartbeat API (https://<CDF HOST>/soar-api/api/heartbeat) returns HTTP 200.

Understanding the backup and restore script

Following are the **db_admin.sh** script options and parameters:

```
./db_admin.sh [Options][Parameters]
```

Options	backup	Start database backup
	status	Obtain database backup status
	restore	Start database restore
Parameters	-t|--type	Perform backup/restore operation
	-u|--user	Specifies the administrator user name
	-p|--password	Specifies the password for administrator
	-l|--location	Identify specific backup/restore operation
	-a|--app	Specifies the appName that want to restore
	-n|--namespace	Namespace that want to backup/restore(default is core)

Running the backup script:

1. Run the following command to get the **backupdApiToken**:

   ```
   ./getRestoreToken
   ```

2. Run the following command:

   ```
   ./db_admin.sh backup
   ```

3. Specify the backup location. The backup data is stored in the following location:

/opt/arcsight-nfs/itom-vol/pg-data-backup/backupd/backups

4. Verify the backup status:

```
./db_admin.sh status -l <location> -t backup
```

5. To see data backup for all the ArcSight Capabilities, run the following command:

```
ls -lh /opt/arcsight-nfs/itom-vol/pg-data-backup/backupd/backups
```

Example output shows data backup for all the ArcSight Capabilities:

```
total 0
drwxr-x---. 3 arcsight arcsight 44 Mar 25 17:04 2021-03-25T17:04:43.807Z
drwxr-x---. 2 arcsight arcsight 27 Mar 25 17:37 2021-03-25T17:37:24.907Z
drwxr-x---. 2 arcsight arcsight 27 Mar 29 08:24 2021-03-29T08:24:14.530Z
drwxr-x---. 2 arcsight arcsight 27 Mar 29 09:36 2021-03-29T09:36:19.605Z
drwxr-x---. 2 arcsight arcsight 27 Mar 29 09:54 2021-03-29T09:54:01.232Z
drwxr-x---. 3 arcsight arcsight 44 Mar 29 13:05 2021-03-29T13:05:46.311Z
drwxr-x---. 4 arcsight arcsight 56 Mar 29 14:12 2021-03-29T14:12:58.071Z
drwxr-x---. 4 arcsight arcsight 56 Mar 30 06:48 2021-03-30T06:48:57.245Z
drwxr-x---. 4 arcsight arcsight 56 Apr  2 14:34 2021-04-02T14:34:55.128Z
drwxr-x---. 4 arcsight arcsight 56 Apr 16 12:17 2021-04-16T12:17:27.006Z
-rw-r-----. 1 arcsight arcsight  0 Mar 25 17:04 backupd.lock
```

The backup data is saved in directories with timestamps. For example, 2021-03-25T17:37:24.907Z. The data stored here is used during the restore procedure of applications.

Restoring a Capability's Configuration Data

You can restore the backed up configuration data for any capability. Use the **db_admin.sh** backup-restore script, located by default in the /opt/arcsight/kubernetes/tools/postgres-backup/ directory.

1. Stop the running application:

```
kubectl scale --replicas=0 deployment <app-deployment-name> -n arcsight-
installer-xxxxx
```

2. Run the following command:

```
./db_admin.sh restore -l <location> -a <application name>
```

3. Check the restore status:

```
./db_admin.sh status -l <restore location>  -t restore
```

Make sure to differentiate the `<restore location>` in the status check from the `<backup location>` in the restore command, as these do not correspond to the same location.

4. If the application's database password is kept in the vault, skip this step. If it is not and it has been backed up with a user other than the database owner (like 'postgres'), you need to run the following commands in the PostgreSQL console.

This procedure needs to be done for the SOAR database.

   a. Log in to the PostgreSQL container console:

```
kubectl exec -it itom-postgresql-default-xxxxxxxxxx-xxxxx -n core -c itom-
postgresql-default --  /bin/bash
```

   b. Get the database user password on the vault:

```
get_secret <vault-key> (postgres user default vault-key=defaultdb_user_
password)
```

   c. Switch to postgres user: `su postgres`

   d. Run the following command and paste the password value in step b.

```
psql –h localhost app-db-name
```

   e. To grant usage privilege to postgres user for the public schema, run the following command:

```
GRANT USAGE ON SCHEMA public TO postgres;
```

   f. To grant create privilege to postgres user for the public schema, run the following command:

```
GRANT CREATE ON SCHEMA public TO postgres;
```

   g. To grant usage privilege to app-db-owner-user for the public schema, run the following command:

```
GRANT USAGE ON SCHEMA public TO app-db-owner-user
```

   For example:

```
soar=# GRANT USAGE ON SCHEMA public TO soar;
```

h. To grant create privilege to app-db-owner-user for the public schema, run the following command:

```
GRANT CREATE ON SCHEMA public TO app-db-owner-user
```

5. Restart the application:

```
 kubectl scale --replicas=1 deployment <app-deployment-name> -n arcsight-
installer-xxxxx
```

The app deployment names are different from the product names, see below for reference.

Application	Deployment Name
Fusion ArcMC	fusion-arcmc-web-app

Checking Backup and Restore Status

Use the **db_admin.sh**, backup-restore script, located in the `/opt/arcsight/kubernetes/tools/postgres-backup/` directory to check the status of backup/restore procedure of all ArcSight Capabilities.

Checking backup/restore status:

1. Run the following command to view the backup/restore status:

```
./db_admin.sh status -l <location>  -t <backup/restore>
```

Where:

<Location>: The encoded directory with timestamps, for example 2021-03-25T17:37:24.907Z.

<Type>: The type of operation to be performed, for example, backup or restore.

For example:

```
./db_admin.sh status -l 2021-03-30T07%3A05%3A02.183Z -t restore
```

2. Specify the IDM token that you received in the Fetching the IDM Token for backupd service section, when prompted.
3. Specify the name of the ITOM Management admin user.
4. Specify the password for the ITOM Management admin user.

Example output shows the in progress restore status for all the ArcSight Capabilities:

```
[INFO] 2021-03-30 07:05:41 :
{
"version": "1",
```

```
"user": "admin",
"mode": "full",
"applications": {
"soar": {
"default-postgresql-svc.core": {
"status": "IN_PROGRESS"
}
}
},
"status": "IN_PROGRESS"
}
```

Chapter 10: Managing Your ArcSight Infrastructure with ArcMC

ArcSight Management Center (ArcMC) is a centralized security management center that manages large deployments of ArcSight solutions such as ArcSight Logger, ArcSight SmartConnectors (Connectors), ArcSight FlexConnectors, and ArcSight Connector Appliance (ConApp) through a single interface.

Whether you have a large deployment of ArcSight or a small shop, ArcMC automates log collection and log management. ArcSight Management Center helps you with centralized management of ArcSight solution, automation of change management, reduction of the resource requirement for security information and event management (SIEM), easy management of large deployments, reduction of the administrative overhead, efficient log traffic management, bandwidth optimization for log collection, support of IT operational analytics. ArcMC also manages the ArcSight deployment through a unified interface.

The following topics are discussed here:

The User Interface

This chapter provides a general overview of the Arcsight Management Center interface. Arcsight Management Center uses a browser-based user interface. Refer to the Arcsight Management Center Release Notes for the latest information on supported browsers.

The following topics are discussed here.

The Menu Bar

The menu bar provides access to the main functional components of Arcsight Management Center. It includes the **Dashboard**, **Node Management, Configuration Management, User Management** and **Administration** menus.

Monitoring Summary

The Monitoring Summary page displays information on all monitored products.

- The aggregated health status for products of each type is displayed in pie graph format, showing total number of nodes, as well as the number corresponding to each status. A summary table shows the same data in percentage format.

- The management panel displays the **Monitoring Summary** table, showing all products which are currently reporting issues.
- The navigation panel enables you to display a monitoring summary for individual product types in the management panel. Click the product type to display the product's monitoring summary.

For more information on viewing and configuring monitoring, see "Dashboard" on page 868.

Node Management

Use **Node Management** to manage any of the following node types:

- Connectors or Collectors
- Hardware or Software Connector Appliances
- Hardware or Software Loggers
- Hardware or Software ArcSight Management Centers
- Transformation Hub

For more information on adding and managing nodes, see "Managing Nodes" on page 906. From the same menu, you can also perform selected management tasks on managed ArcSight products. See "Managing ArcSight Products" on page 939.

Configuration Management

Use **Configuration Management** to create and manage node configurations, synchronization (pushing) of configurations across multiple nodes, and expedite the initial configuration of Loggers. You can manage any of these configuration types:

- Subscriber configurations for:
  - Arcsight Management Center
  - Connectors
  - Connector Appliances
  - Destinations
  - Loggers
  - System administration
- Other configurations are also managed here:
  - Logger Initial configurations
  - Logger event archives
  - Management of Logger peers
  - Management of Transformation Hub

- Bulk Operations
- Generator ID Management
- Management of Deployment Templates

For more information on subscriber configuration management, see "Managing Configurations" on page 991.

For more information on initial configurations, see "Logger Initial Configuration Management" on page 1028.

User Management

User management enables you to manage users across all of your managed nodes. You can create and edit users, user lists, their associations, and roles. You can also check to see if each node complies with a list of authorized users on the managing ArcSight Platform.

For more information about user management, see "Overview" on page 1

Administration

The **Administration** menu contains these items:

- **Backup:** Enables you to back up your current ArcSight Management Center configuration.

  This function isn't available when you deploy ArcMC in the containerized ArcSight Platform.

- **Repositories:** Enables you to manage repositories that store files, such as logs, certificates, and drivers. For more information, see "Managing Repositories" on page 647.
- **Snapshot:** Enables you to take a snapshot image of ArcSight Management Center, to produce logs that are useful in troubleshooting. For more information, see " Snapshots" on page 646.
- **Restore:** Enables you to restore your configuration from a saved backup.

  This function isn't available when you deploy ArcMC in the containerized ArcSight Platform.

- **System Admin:** Describes the system administration tools that enable you to create and manage users and user groups, and to configure security settings for your system. For more information, see "System Administration" on page 1069.
- **Consumption Report:** Generates a report on Logger data consumption for selected managed nodes.

ArcMC Name

You can set a name for your ArcMC during the CDF deployment for Fusion ArcMC.

A valid ArcMC name must meet the following criteria:

- Is a non-empty string
- Is equal to or less than 32 characters long
- It contains characters: A-Za-z0-9 _-

Stats (EPS In/Out)

The **Stats** menu item shows the total Events Per Second (EPS) in and out from all managed connectors (standalone SmartConnectors and connectors running on managed hosts).

Job Manager

The Job Manager shows all deployment jobs processed in a specified time frame. Using the Job Manager, you can identify issues that occurred during deployments.

The Job Manager shows the following for each job:

- **Name of the Job:** The job name (must be smaller than 255 characters).
- **Started By:** The user who ran the job.
- **Type:** Type of job.
- **Start/End Time:** The start and end time of the job.
- **Status:** Job status. If the job has a status of *Failed*, click **Retry** to re-run the job.
- **Details:** Job details.

Hover over any field to display details about the field in a tooltip. Click the Up/Down arrows at the top of any column to sort data by the selected parameter.

To view the Job Manager:

1. On the menu bar, click the Job Manager (notepad) icon  . By default, the Job Manager displays all deployment jobs for the last 5 days. A red numeral on the Job Manager icon, if any, indicates the number of jobs currently in the In-Progress state.

- To change the time frame for job data displayed, specify the date criteria in the date boxes in the upper right corner, then click **Show Results.**You may specify any time frame in the last 180 days (6 months).
- To search for a specific job, specify the search criteria in the **Search** box.
- If a job is in progress, you can click **Refresh** on the menu bar to refresh the display.

Site Map

For ease of accessibility and convenience, the Site Map links to all pages in the Arcsight Management Center UI.

To access the site map: on the main ArcSight Platform toolbar, click **Site Map.** Select the desired link to navigate.

History Management

History management enables you to quickly and easily access previously-navigated pages. History management is available for Node Management, Configuration Management, User Management pages, and for some Administration pages.

In Node Management, the navigation tree shows the full path for any item selected on the tree. Click any node in the path to navigate directly to the corresponding page.

You also can return to any previously-browsed page by clicking the corresponding link in the breadcrumb trail.

In addition, you can use your browser's **Back** and **Forward** buttons to navigate to previously visited pages.

Dashboard

Using Arcsight Management Center, you can monitor the health status of all managed nodes. You can also configure warnings and alerts for issues of importance to you.

> **Note:** In order for products to be monitored, they must be added as nodes to Arcsight Management Center. For more information on managing nodes, see "Managing Nodes" on page 906.

Monitoring is displayed on the **Dashboard > Monitoring Summary** page. Arcsight Management Center automatically monitors all managed nodes.

You can also configure notifications (email, SNMP, and through audit forwarding) about the status of managed nodes.

Monitoring Managed Nodes

Arcsight Management Center monitoring, on the **Dashboard > Monitoring Summary** page, displays the current health history of all managed nodes, both software and hardware.

- Monitored metrics for software nodes (such as Software Logger) include such software parameters as CPU usage, event flow, and disk usage statistics.
- Monitored metrics for hardware appliances (such as Logger Appliance) include both software as well as hardware-related attributes, such as remaining disk space and hardware status.
- Device health related information:
  - Devices have severity associated with them instead of status. Up is equivalent to "HEALTHY" and Down to "FATAL".
  - Sunburst Chart and corresponding breakdown table is enhanced to show the severity instead of status.

You can view a complete list of monitored parameters in "Monitoring Rules Parameters" on page 881 , and use them in creating your own custom rules. These rules breaches will also be displayed on the Health History and Hardware Status panels. Note that the layout and selection of the data panels in the Monitoring Summary is not customizable.

The Monitoring Summary Dashboard

The Monitoring Summary includes a variety of panels that display monitoring information on the health and status of your managed products.

To view the monitoring summary, click **Dashboard > Monitoring Summary**.

Total Number of Nodes

Each tile in the **Total Number of Nodes** panel displays the count of managed nodes of the specified type. These types are defined as follows.

Tile	Count
Devices	Devices which are forwarding events.
ArcMC/CHA	Includes managed ArcMCs and Connector Hosting Appliances, in both hardware and software form factors.
Connectors	Managed Connectors.

Tile	Count
Collectors	Managed Collectors.
Loggers	Managed Loggers (hardware and software form factors).
Nodes	Nodes on the managed Transformation Hub. (Note that if Transformation Hub is upgraded, the Monitoring Summary will not reflect the correct Transformation Hub information until you import the new Transformation Hub certificate into ArcMC. See "Downloading and Importing Host Certificates" on page 1052 for more information.) **Note:** To display Transformation Hub Processing Data users need to turn on the C2AV pod on the Transformation Hub, for more information see the Transformation Hub Administrator's Guide. **Event Parsing Error**, **Stream Processing EPS**, and **Stream Processing Lag** are the metrics that will be available after the C2AV pod is turned on, otherwise only **CPU Usage** and **Memory** under **Broker's Health** will be displayed. **Note:** The stream processors metric name format has changed to SP_Name(SP_Type).

To see the details of a node type, click the title corresponding to the node type. For example, to view the details of all Collectors, click **Collectors**.

Devices by Device Type

The **Devices by Device Type** display shows a color-coded sunburst of the various device types in use across your network. The table shows the total number of active and inactive devices by device product.

The inner ring of the sunburst shows the total devices.

The outer ring of the sunburst shows the total number of device types. For clarity of display, if the number of device types exceeds 1000, the outer ring is not shown.

The **Devices Information for All Device Types** table breaks down the information to display Device Type, Severity (Fatal, Critical, Warning, Healthy), and Total Devices.

To see the details of a device type, click the corresponding tile in the wheel, or its entry in the table.

Note: ArcMC 2.6 and 2.7 Device Monitoring function supports only Connectors 7.3 - 7.7. ArcMC 2.8 and later support Connectors 7.3 and later for Device Monitoring.

Device Configuration for Device Type

The Device Configuration for Device Type page allows you to modify the **Device Product time-out Interval**, **Device age-out Interval**, and **Disable Device Tracking.**

Device Product time-out Interval

The default value is set to 20 minutes, this can be modified. If the selected device type does not send events to the connector during the last 20 minutes, the device type will be marked as Inactive.

Device age-out Interval

The default value is set to 14 days, this can be modified. If the selected device type remains inactive for 14 days, the device type records will be purged from the system.

Disable Device Tracking.

This box can be checked to disable the selected device product family.

Note: If device product monitoring is re-enabled X days later while **Disable Device tracking** is enabled, the aged-out internal should be set to Y days, in which Y comes after X days. This will prevent the selected disable tracking product family device records from being removed of the ArcMC system.

Device Health Metrics

The dashboard displays device health information as severity. The Sunburst Chart shows the Severity as "HEALTHY", "FATAL", "WARNING", or "CRITICAL".

Note: The selection and layout of the panels on the Monitoring Summary is not customizable. You can, however, customize the issues reported for a given node type by creating custom breach rules, which can be viewed on the Severity Issue Summary. See "Monitoring Rules" on page 875

Drilling Down

You can view the details of problematic nodes, then take action to rectify any issues.

To view all details of a problematic node, select it in the upper table. The lower table shows issues associated with that node. Each issue is shown with these identifiers:

- **Metric Type:** Metric assigned to the issue.
- **Metric Name:** Name of the metric.
- **First Occurrence:** Local time of the issue's first occurrence.
- **Last Occurrence:** Local time of the issue's last occurrence.
- **Severity:** Issue severity.
- **Description:** Brief description of the issue.

To view details of nodes by severity:

1. On the menu bar, click **Dashboard > Monitoring Summary**.
2. Click the ring meter corresponding to any of the monitored product types, in the portion of the meter corresponding to the severity you wish to view. (For example, to view all nodes currently with Warning status, click the Warning, or yellow, part of the ring.) The corresponding **Severity Issue Summary** is displayed.
3. On the **Severity Issue Summary** page:

   The upper table shows a list of all problematic nodes, with the following identifiers:

   - **Name:** Node name.
   - **Path:** Path to the node.
   - **Type:** Type of node.
   - **Lead/Breach:** Short summary of the most severe issue reported by the node. The node may be experiencing less severe issues as well.

Details and Health History

To view further health details of a problematic node, including history and status, click **Details**. The data tables show the detailed parameters of the selected node.

The Health History panel will show any rules breaches, including custom rules you have created yourself.

Note: The layout of the panels and selection of the displayed parameters is not customizable.

Data Charts

Each data chart represents values of the parameter over time. Use the drop-down list to change the interval shown from the last 4 hours, the last day, or the last week. Data charts can include any of the metrics shown under the Valid Values for Metric Types table.

Click the data legend to toggle display of the corresponding line from the chart. Hiding some lines may be helpful to clarify a chart with many lines.

ADP License Usage for the Last 30 Days

Your ADP license entitles you to a specified number of managed products and amount of managed traffic. The **ADP License Usage for the Last 30 Days** panel shows your ADP data usage for the previous month.

The graph shows all traffic in your ADP environment.

- Green (the default) indicates that data usage is within your licensed limit.
- Amber indicates periods when your data usage approached your licensed traffic limit.
- Red indicates periods when your data usage exceeded your licensed traffic limit.

The **Active Loggers** indicate the number of ADP Loggers the data from which contributes to the license monitoring report. For more details, you can export the license report to PDF format, which includes data on the last 365 days.

If your ArcMC is enabled as a License Server, the Daily Usage bar chart displays the overall ADP license consumption on a daily basis. The daily license usage is calculated from the managed connectors (version of 7.3.0 or later) and managed ADP loggers based on the following:

- If a Connector is managed by ArcMC, then ArcMC will include its event ingestion from all non-ADP or non-managed source devices in the ADP daily license usage calculation. If a source is also a managed ADP component, the event flow from this source to the managed Connector will not be tracked.
- If an ADP Logger is managed by ArcMC, then ArcMC will include its event ingestion from all non- ADP or non-managed source devices in the ADP daily license usage calculation. If a source is also a managed ADP component, the event flow from this source to the managed ADP Logger will not be tracked.

Each day, ArcMC collects the daily ingestion information from each Connector and ADP Logger. Connectors and Loggers give an accumulated ingestion total when not reachable by ArcMC at the time of ingestion collection (daily at 1:00:00 ArcMC local time by default). This scenario could be caused by any of the following:

- The ADP Logger or Connector was down.
- The ADP Logger or Connector’s server certificate has changed.
- The ADP Logger or Connector was not managed by the ArcMC.

If any managed nodes (Connector, ADP logger) are not reachable during ingestion collection time, the daily consumption of these nodes will be counted and reflected in the consumption number on a daily report, when ArcMC license server has successfully pulled the consumption data from the affected nodes.

Note: Daily ingestion collection only applies to License Server ArcMCs and ArcMCs that are managed by the License Server.

The ingestion report on an individual ADP Logger includes its previous day’s ingestion during the time window of [00:00:00 – 23:59:59] GMT. For license usage calculation, ArcMC collects the previous ADP Logger’s ingestion during the time window of [01:00:00 – 24:59:59] ArcMC local time. The time window used for individual Logger ingestion tracking and ingestion calculation are different; hence, it is not recommended to compare these two reports because they will report different numbers.

To enable the display of ADP license usage:

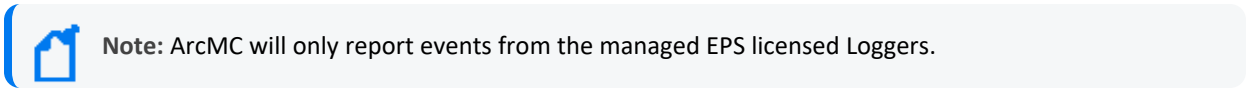
1. Enable ArcMC as an ADP license server. In the ArcMC toolbar, click **ADP License Server,** then click **Yes**.
2. Upload a valid capacity license to the ArcMC on the **License and Upgrade** page.

To export the license report to PDF format:

1. Click **Export License Report.**
2. The PDF is downloaded to your local system.

EPS License Reporting

The customer is considered to be in compliance with the license agreement as long as the MMEPS value indicators remain at the limit or below the purchased license capacity. If 3 or more consecutive MMEPS value indicators exceed their capacity based on the purchased license, they are considered to be out of compliance.



Note: ArcMC will only report events from the managed EPS licensed Loggers.

You can download up to one year license reports in PDF format.

Keystones:

1. **Events per Day (EPD):** Is the total number of events generated in a 24 hour clock period. The clock is calculated based on UTC time starting at 00:00:00 and ending at 23:59:59, regardless of the local times used.
2. **Sustained EPS (SEPS):** Is the event “constant” per second supported by the system within the 24 hour clock period. It stabilizes peaks and valleys and gives a better indication of use
3. **Moving Median EPS (MMEPS):** Is the license usage. It uses the 45 day period SEPS data shifting the calculation window 1 day every 24 hours after the first 45 days. The clock is calculated based on UTC time starting at 00:00:00 and ending at 23:59:59, regardless of the local times used.
4. **License Limit:** Corresponds to the amount of EPS acquired in the license.
5. **Baselining:** The baselining period begins when an EPS licensed product is detected in ArcMC for the first time (day 1), and it continues for the next 45 days. Once ArcMC detects an EPS licensed product, the baseline is set, and it does not change even if the license is redeployed. During this period, the usage will be calculated as the median of the SEPS values available at that moment. MMEPS values are truncated to benefit the customer. For example:

   MMEPS Calculation

Day 1: SEPS of day 1

Day 2: Truncated median of SEPS of days 1 and 2.

Day 3: Truncated median of SEPS of days 1, 2, and 3.

Day 45: Truncated median value of SEPS of days 1 through 45

EPS License Usage Calculation

The usage will be collected from each managed Loggers and ArcMCs once a day.

- **Moving Median Events Per Second (MMEPS):** The median value over the last 45 days.
- **Baselining:** The usage will be calculated as the median of the SEPS values available at that moment. MMEPS values are truncated to benefit the customer.

Host Status Exceptions

This feature lists all the managed nodes that are in either Fatal, Critical or Warning status. To access the monitoring metric details view of a managed node, click **Dashboard > Host Status Exceptions.**

The following fields are displayed in the host status exceptions page:

- Host name: Name of the host.
- Status: Status of the host (Fatal, Critical, Warning).
- Cause: Root cause for hosts to be unhealthy (usually due to being unreachable or triggering a specific rule).
- Type: Type of host.
- Logical Group Path: Host location within ArcMC.

Monitoring Rules

Monitoring rules are defined to generate monitoring warnings for each managed product type. ArcMC includes many preset monitoring rules for your use. You can use these rules as written, or customize them for your own use. In addition, you can create your own custom monitoring rules.

A monitoring rule comprises a set of logical, performance, health, or other criteria. All criteria in the rule are evaluated together to determine the rule's total effect, which generates an alert from ArcMC.

Rules breaches will be displayed in the Warning Severity Issue Summary, which you can view by clicking one of the ring meters on the Monitoring Dashboard.

For example, a rule could check for the number of *input events per second* (criterion #1) that reach a *certain type of device* (criterion #2). Should this number *exceed* (criterion #3) a specified *level* (criterion #4), then a *warning* (*alert*) should be returned.

Breach Function

The breach function checks the backend monitor metric data table. The metric data table is updated every 3 minutes, and the breach check function runs every four minutes at the 45th second. Reducing the rule's time range to a smaller number (e.g. 1 or 2) could result in an undetected breach.

Alerts can be delivered by email or by SNMP, or can be recorded in audit logs. Only when there is new breach detected (i.e. not found on the previous run), ArcMC sends the notification/alert if the notification option is enabled. If the breach keeps coming on the subsequent calls, the alert will only be sent the first time.

For more information on managing and creating rules, see "Managing Rules" on page 880.

Preset Rules

ArcSight Management Center includes preset rules to assist in monitoring. You can use these preset rules as written or customize them as needed for your own use. You can also create custom rules of your own.

By default, ArcMC preset rules are disabled. You must enable a preset rule in order for it to apply and trigger alerts.

Note: For customers with previous versions of ArcMC and who already have a list of existing rules, preset rules included in ArcMC are appended to your existing rules.

To review preset rules:

1. Click **Dashboard > Rules.** The Monitoring Rules summary is shown.
2. To view a rule's settings in detail, in the **Name** column, click the rule name.
3. To enable a disabled preset rule, under **Status**, select **Enable.**

Preset Rules Description

Name	Description	Products			
MM_DD_YYYY_RAID_BATTERY_Failed_ArcMC_ConApp_Logger	Displays a critical alert when the Raid Battery has failed during the last 5 minutes.	ArcMC	ConApp	Logger	
MM_DD_YYYY_POWER_SUPPLY_Failed_ArcMC_ConApp_Logger	Displays a critical alert when the Power supply has failed during the last 5 minutes.	ArcMC	ConApp	Logger	
MM_DD_YYYY_TEMPERATURE_Failed_ArcMC_ConApp_Logger	Displays a critical alert when the temperature reaches a certain level during the last 5 minutes.	ArcMC	ConApp	Logger	
MM_DD_YYYY_POWER_SUPPLY_Degraded_ArcMC_ConApp_Logger	Sends a warning when the power supply has been degraded during the last 5 minutes.	ArcMC	ConApp	Logger	
MM_DD_YYYY_VOLTAGE_Failed_ArcMC_ConApp_Logger	Displays a critical alert when the voltage levels have been failing during the last 5 minutes.	ArcMC	ConApp	Logger	
MM_DD_YYYY_FAN_Failed_ArcMC_ConApp_Logger	Displays a critical alert when the fan has failed during the last 5 minutes.	ArcMC	ConApp	Logger	
MM_DD_YYYY_HARD_DRIVE_Rebuilding_ArcMC_ConApp_Logger	Sends a warning when the hard drive has been rebuilding during the last 5 minutes.	ArcMC	ConApp	Logger	
MM_DD_YYYY_RAID_CONTROLLER_Failed_ArcMC_ConApp_Logger	Displays a critical alert when the RAID controller has failed during the last 5 minutes.	ArcMC	ConApp	Logger	
MM_DD_YYYY_CURRENT_Degraded_ArcMC_ConApp_Logger	Sends a warning when the current has been degraded during the last 5 minutes.	ArcMC	ConApp	Logger	
MM_DD_YYYY_RAID_CONTROLLER_Degraded_ArcMC_ConApp_Logger	Sends a warning when the raid controller has been degraded during the last 5 minutes.	ArcMC	ConApp	Logger	
MM_DD_YYYY_VOLTAGE_Degraded_ArcMC_ConApp_Logger	Sends a warning when the voltage has been degraded during the last 5 minutes.	ArcMC	ConApp	Logger	
MM_DD_YYYY_ALL_EPS_OUT_ArcMC_ConApp_Logger	Displays a critical alert when all outgoing events per second have failed during the last 5 minutes.	ArcMC	ConApp	Logger	
MM_DD_YYYY_HARD_DRIVE_Failed_ArcMC_ConApp_Logger	Displays a critical alert when the hard drive has failed during the last 5 minutes.	ArcMC	ConApp	Logger	

Name	Description	Products			
MM_DD_YYYY_Queue Files Accumulated	Displays a critical alert when files have accumulated in queue during the last 5 minutes.				Connector
MM_DD_YYYY_Full GC	Sends a warning when the garbage collection count is higher than 7 during the last 60 minutes.				Connector
MM_DD_YYYY_Caching	Sends a warning when the connector caching is higher than 100 during the last 5 minutes.				Connector
MM_DD_YYYY_Receiver Down	Sends a warning when the receiver has been down during the last 5 minutes.			Logger	
MM_DD_YYYY_Events Dropped from Cache	Displays a fatal alert when the connector events dropped from cache have been down during the last 5 minutes.				Connector
MM_DD_YYYY_Files Dropped From Cache	Displays a critical alert when the connector files dropped from cache have been down during the last 5 minutes.				Connector
MM_DD_YYYY_Logger Not Receiving Data	Displays a fatal alert when logger hasn't recevied data during the last 30 minutes.			Logger	
MM_DD_YYYY_Storage Disk Usage above 85%	Sends a warning when the storage limit goes over 85% during the last 5 minutes.			Logger	
MM_DD_YYYY_JVM_ MEMORY_ArcMC_ ConApp_Logger	Sends a warning when the jvm memory reaches 800 GB during the last 5 minutes.	ArcMC	ConApp	Logger	
MM_DD_YYYY_Connector Restart	Sends a warning when the connector has restarted more than 5 times during the last 5 minutes.				Connector
MM_DD_YYYY_Memory Red Zone	Displays a critical alert when the Connector JVM memory has gone over 90% during the last 5 minutes.				Connector
MM_DD_YYYY_Memory Yellow Zone	Sends a warning when the Connector JVM memory has gone over 80% during the last 5 minutes.				Connector
MM_DD_YYYY_Events Dropped From Queue	Displays a fatal alert when more than 100 Connector queue events dropped during the last 5 minutes.				Connector

Name	Description	Products			
MM_DD_YYYY_Files Dropping From Queue	Displays acritical alert when Connector files dropped from queue during the last 5 minutes.				Connector
MM_DD_YYYY_RAID_ BATTERY_Degraded_ ArcMC_ConApp_Logger	Sends a warning when the raid battery has been degraded during the last 5 minutes.	ArcMC	ConApp	Logger	
MM_DD_YYYY_ TEMPERATURE_ Degraded_ArcMC_ ConApp_Logger	Sends a warning when the temperature has been degraded during the last 5 minutes in	ArcMC	ConApp	Logger	
MM_DD_YYYY_EPS_OUT_ Connector	Displays a critical alert when the outgoing events per second have been degraded during the last 5 minutes.				Connector
MM_DD_YYYY_FAN_ Degraded_ArcMC_ ConApp_Logger	Sends a warning when the fan's RPMS have failed during the last 5 minutes.	ArcMC	ConApp	Logger	
MM_DD_YYYY_HARD_ DRIVE_Degraded_ArcMC_ ConApp_Logger	Sends a warning when the hard drive has been degraded during the last 5 minutes.	ArcMC	ConApp	Logger	
MM_DD_YYYY_ALL_EPS_ IN_ArcMC_ConApp_ Logger	Displays a critical alert when all incoming events per second have failed during the last 5 minutes.	ArcMC	ConApp	Logger	
MM_DD_YYYY_CPU_ USAGE_ArcMC_ConApp_ Logger	Sends a warning when the cpu usage has exceeded 50% during the last 5 minutes.	ArcMC	ConApp	Logger	
MM_DD_YYYY_QUEUE_ DROP_COUNT_Connector	Sends a warning when Objects droppped from file Queue during the last 5 minutes.				Connector
MM_DD_YYYY_CURRENT_ Failed_ArcMC_ConApp_ Logger	Displays a critical alert when the current has failed during the last 5 minutes.	ArcMC	ConApp	Logger	

Managing Rules

To create a custom rule:

1. Click **Dashboard > Rules.**
2. In the toolbar, click **Add New Rule.**
3. Select values for the rule parameters.
4. Click **Save.**

To edit an existing rule:

1. Click **Dashboard > Rules.**
2. Under **Monitoring Rules**, select the rule you wish to edit.
3. Click **Edit Rule.**
4. Select new values for the rule parameters, as needed.
5. Click **Save.** Alternatively, click **Save As** to save the edited rule with a new name.

When creating or editing rules, the only characters that are allowed for naming them are the following:

- Letters (a-z and/or A-Z)
- Numbers and spaces
- Symbols (only restricted to): **%** _ and -

To export all rules to a text file:

1. Click **Dashboard > Rules.**
2. In the toolbar, click **Export.** Your rules are exported to a local text file called `monitor_breach_rules.properties.` and downloaded locally.

> **Caution:** Do not partially delete a rule from the exported breach rules file. The rules file to be uploaded should have all the properties for all the rules in the file. Before uploading a new breach rules file create a backup of the existing file.

To import a rule:

1. Click **Dashboard > Rules.**
2. In the toolbar, click **Import.** A new window will pop-up, click **Browse**, find the location of the file, select it, and click **Import.**

Global Settings

1. Click **Dashboard > Rules.**
2. In the toolbar, click **Global Settings.** The following settings are displayed: **SNMP Notifications**, **Email Notifications**, and **Audit Notifications.** These settings enable or disable notifications to be sent by ArcMC.

To enable (or disable) a rule:

1. Click **Dashboard > Rules.**
2. In the management panel, under **Monitoring Rules**, select the rule to enable or disable.
3. In the **Rule Name** column, click the rule name.
4. Under **Status**, toggle the status to **Enable** (or **Disable**).
5. Click **Save.**

To delete a rule:

1. Click **Dashboard > Rules.**
2. Under **Monitoring Rules**, select the rule you wish to delete.
3. Click **Delete.**
4. Click **OK** to confirm deletion.

Monitoring Rules Parameters

Monitoring rules are defined by rule parameters. The following table describes monitoring rules parameters and their valid values.

Monitoring Rules Parameters

Parameter	Description
Name	Name of the rule. (Max. length 50 characters)
Metric Type	Criterion being measured. For valid values of Metric Type, see the **Valid Values for Metric Type** table, below. Each metric type has a Value Type constraining the kind of value which may be assigned to it.
Product Type(s)	Managed product type (or types) to which the rule applies. These are automatically selected based on the Metric Type. For example, if you selected a metric type that applied only to hardware, such as Voltage, only products with hardware form factors would be available for selection. You can also deselect types to which to apply the rule, as applicable.

Monitoring Rules Parameters, continued

Parameter	Description
Specific Node Selector	Click **View/Choose**, then select one or more specific nodes to which the rule applies. If none are chosen, then the rule applies to all nodes of the selected Product Types.
Severity	Breach severity. Valid values are Healthy, Warning, Critical and Fatal. Thresholds for each of these values are defined by the administrator.
Aggregation	Aggregation function applied to Metric Type data points. Valid values: • ANY: any value • AVG: average value (numeric values only) • MIN: minimum value (numeric values only) • MAX: maximum value (numeric values only) • SUM: addition of values (numeric values only)
Measurement	A comparison between two criteria. Valid values: • GREATER: One field is greater than the other • LESS: One field is less than the other • EQUAL: One field is equal to the other • NOT_EQUAL: Two fields are unequal
Value	Threshold value for comparison. Valid values are dependent on Metric Type. • Percentage: Number from 1-100 (with no %-sign). • Numeric: Numeric string. • Boolean: true/false (case-insensitive) • Literal Status: Status of the appliance component, and can be one of the following values: *Ok, Degraded, Rebuilding, Failed, Unavailable*.
Notify Me	Select one or more notification mechanisms for alerts about the rule (Email, SNMP, or Audit Forwarding).
Status	If **Enabled**, the rule will apply and produce alerts, as specified in **Notify Me.** (ArcMC rule presets are Disabled by default.)
Time Range	Evaluation interval, in hours and minutes. The total of hours and minutes must not exceed 168 hours (7 days).

Note: Compound rules (AND/OR) are not supported.

Valid Values for Metric Type

Value	Description	Value Type
Description	Brief description of the rule. (Max. length 300 characters.)	What kind of value this is.
For Connector Appliances or Loggers only		
CPU Usage	CPU usage, as a percentage.	Percentage
JVM Memory	Memory of Java Virtual Machine.	Numeric
Disk Read	Number of reads of the disk.	Numeric
Disk Write	Number of writes to the disk.	Numeric
All EPS In	Total Events Per Second in.	Numeric
All EPS Out	Total Events Per Second out.	Numeric
For Connectors only		
Events/Sec (SLC)	Events Per Second (EPS) in (Since Last Checked)	Numeric
EPS In	Events Per Second (EPS) in.	Numeric
EPS Out	Events Per Second (EPS) out.	Numeric
Events Processed	Number of events processed.	Numeric
Events Processed (SLC)	Events processed (Since Last Checked).	Numeric
FIPS Enabled	1= FIPS enabled, 0=FIPS disabled.	Boolean
Command Responses Processed	Number of command responses processed.	Numeric
Queue Drop Count	Queue drop count.	Numeric
Queue Rate (SLC)	Queue rate (Since Last Checked).	Numeric
Active Thread Count	Active thread count.	Numeric
For hardware form factor products only		
Fan	Hardware fan status.	Literal Status
Disk Space	Hardware disk space status. Disk space will be reported as "degraded" if storage reaches 75% of its capacity. Other statuses are not used.	Literal Status
Voltage	Hardware voltage status.	Literal Status
Current	Hardware current status.	Literal Status
Temperature	Hardware temperature status.	Literal Status
Power Supply	Hardware power supply status.	Literal Status

Valid Values for Metric Type, continued

Value	Description	Value Type
RAID Controller	RAID controller status.	Literal Status
RAID Battery	RAID battery status.	Literal Status
Hard Drive	Hard drive status.	Literal Status
For Loggers Only		
Storage Group Usage	Current storage group usage, in bytes.	Numeric
Storage Group Capacity	Current storage group capacity, in bytes.	Numeric
For Transformation Hubs Only		
Transformation Hub All Bytes In	All bytes received by the Transformation Hub cluster.	Numeric
Transformation Hub All Bytes Out	All bytes transmitted by the Transformation Hub cluster. Note that due to the replication of each topic, Bytes Out will always exceed Bytes In.	Numeric
Transformation Hub Disk Usage	Disk usage of Transformation Hub's individual nodes.	Numeric
Transformation Hub Memory Usage	Memory usage of Transformation Hub's individual nodes.	Numeric
Transformation Hub SP EPS	Count of events per second received by Transformation Hub's Stream Processor.	Numeric
Transformation Hub SP Error	Count of events per second wating to be processed received by Transformation Hub's Stream Processor which produced an error.	Numeric
Transformation Hub SP Lag	Count of events per second waiting to be received by Transformation Hub's Stream Processor.	Numeric
For Collectors Only		
Collector CPU Load Average	Average load of Collector CPU.	Numeric
GC Count	Count of Java garbage collection.	Numeric
Restart Count	Number of restarts.	Numeric
Total Memory	Total JVM memory.	Numeric
Used Memory	JVM memory in use.	Numeric

Rule Verification

It is possible to create syntactically valid rules that return confusing or meaningless alerts. For example, you could create a syntactically valid rule to trigger an alert if CPU usage is below 101%, but this rule would not return useful alerts (since it would alert you constantly).

Always verify your rules to ensure that they return meaningful values, to help you best detect problems and issues.

Note: Custom Polling Intervals: ArcSight Management Center uses three polling intervals (4 hours, 1 day, and 1 week) associated with metric data archive types across ArcSight products. These intervals can be adjusted for proper usage, if required.

It is strongly recommended that you adjust these intervals only if you fully understand the impact of the changes.

Polling intervals can be specified in the file `logger.properties` using a text editor.

- 4-hour data (minimum allowed interval 1 minute):

  `monitoring.data.poll.4hour.cron=10 0/3 * * * ?`

  This property indicates a poll at 3 minute intervals.
- 1-day data (minimum allowed interval 5 minutes):

  `monitoring.data.poll.1day.cron=15 0/10 * * * ?`

  This property indicates a poll at 10 minute intervals.
- 1-week data (minimum allowed interval 1 hour):

  `monitoring.data.poll.1week.cron=20 2 */2 * * ?`

  This property indicates a poll at 2 hour intervals.

After making the changes and saving the edited file, a server restart is required for the changes to take effect.

Custom Rules Examples

Shown here are examples of custom monitoring rules.

Example 1: Warning Breach

This example specifies the following Warning condition:

“Generate a Warning breach if the average CPU usage of any ArcMC in the past 30 minutes is greater than 70%.”

Name: ArcMC Warning

Metric Type: CPU Usage

Product Type: ArcMCs

Severity: Warning

Aggregation: AVG

Measurement: GREATER

Value: 70

Timespan: 30 minutes

Example 2: Critical Breach

Example 2 specifies the following Critical condition:

“Generate a Critical breach if the Power Supply fails on any Logger Appliance in the past hour.”

***Name:** Logger Warning*

Metric Type: Power Supply

Product Type: Loggers

Severity: Critical

Aggregation: ANY

Measurement: EQUAL

Value: Failed

Timespan: 60 minutes

Device Rule Management

Device Rule Management involves creating, editing and deleting rules specifically for devices. The operation of creating, editing and deleting rules is different than what is done for other entities. Rules are created on the Device List page. The contents of the rule are the same as those of the exiting rule.

The Device List page is where you manage rules. This page has two tabs: Devices and Manage Rules.

Device Inactive Notification

When ArcMC detects an inactive device, (time out value can be defined by the customer on the Device UI page, default value is set to 20 minutes), the internal defined device inactive rule is triggered, and an alert is sent out via snmp, email, and audit log.

There are two options for users who don’t want to receive device inactive notifications:

1. Keep the device on “active” status: Review the connector’s device event status and configure a proper interval value for Device Product time-out interval on **Dashboard > Monitoring Summary> Devices UI** page.
2. Contact support to disable device inactive notifications.

Managing Devices

About

From the Devices page you can add one or more devices to a new rule or add one or more devices to an existing rule.

The Lead Breach column describes the Lead Breach for a device. The Severity column describes the severity of a device. Severity is defined when creating a rule. The # of Rules column describes the number of rules applied to the devices.

Procedure

Location: Dashboard > Monitoring summary > Devices count indicator > Devices page

To add one or more devices to a new rule

1. Select the desired device or devices.
2. Click **Add New Rule**.
3. From the Add New Rule dialog, specify the necessary information.

   Device rules support "EPS out" and "Bytes out" measurements.

To add one or more devices to an existing rule

1. Select the desired device or devices.
2. Click **Add to Existing Rule**.
3. From the Add to Existing Rule dialog, specify the existing rule.

See also

- "Device Rule Management" on the previous page
- "Managing Device Rules" on the next page

Managing Device Rules

About

The Manage Rules page lists of all the rules and options: Disable, Enable, Delete and Edit an existing rule. The multi-selection option is available for Disable, Enable and Deleting the Rules. You can Edit one rule at a time.

A device that has stopped sending events will be marked as "Fatal" and there is no rule to change that. The timeout value for each device product is configurable and documented.

Procedure

Location: Dashboard > Monitoring summary > Devices count indicator > Manage Rules tab

1. Click **Manage Rules**.
2. From the Rules Details page, specify the desired management option.

See also

- "Device Rule Management" on page 886
- "Managing Devices" on the previous page

Configuring Email Notifications

Email notifications will inform recipients about monitored nodes being down or out of communications.

Note: Email alerts do not include issues with connectors or Collectors. However, containers may be the subject of email alerts.

Before configuring email notifications, ensure that values are specified for your SMTP settings under **Administration > System Admin > System > SMTP.** For more information on SMTP settings, see "SMTP" on page 1074.

Once configured, email notifications must be configured for each of the notification rules you wish to trigger an alert.

To configure email notifications:

1. In a text editor, open the file .../`userdata/arcmc/logger.properties`. (If the file does not exist, you can create it in a text editor. When creating the file, ensure that it is owned by the non-root user.)

2. Add a new line with the new property named `monitoring.notification.emails` and a value equal to a comma-separated list of email addresses of all administrators you intend to receive notifications. For example, this value would send email alerts to address1@example.com and address2@example.com:

   ```
   monitoring.notification.emails=address1@example.com,
   address2@example.com
   ```

3. Save the modified `logger.properties` file.
4. Restart the ArcMC web process.
5. In the rules editor, open the notification rule you wish to trigger an email alert, and under **Notify Me,** select *Email*.

Example Email Notification

An example of the email sent to recipients is shown here.

`<URI>` refers to the URI of a problematic node.

NodeN is the hostname of a problematic node.

This information is found on the **Hosts** tab under Node Management.

```
Subject: <Email title>
The following nodes are either down or not reachable from ArcSight Management
Center:
```

`//Default/<URI>/<Node1>`

`//Default/<URI>/<Node2>`

Configuring SNMP Notifications

SNMP notifications will send SNMP traps about monitored nodes being down or out of communications.

To configure SNMP notifications on ArcMC appliance:

1. Under **Administration > System Admin > System > SNMP**, enable SNMP. Then, specify values for port, SNMP version, and other required settings for your SNMP environment.
2. In the rules editor, open the notification rule you wish to trigger an SNMP alert, and under **Notify Me,** select *SNMP*. Repeat for each rule you wish to trigger an SNMP alert.

Enabling SNMP on Software ArcSight Platform

Software ArcMC does not include UI controls for SNMP configuration. Instead, take these steps to configure Software ArcMC for SNMP notifications and monitoring.

To enable SNMP notifications on a software host:

1. Make sure following RPM packages are installed on the system: `net-snmp, net-snmp-utils, net-snmp-libs, lm_sensors-libs.`

2. Enable the SNMP service by entering: `chkconfig snmpd on`

3. Start the SNMP service by entering: `service snmpd start`

4. In a text editor, create a file `/opt/arcsight/userdata/platform/snmp.properties` with the following parameters, Items in angle brackets <>indicate you should substitute values appropriate for your own environment.

```
snmp.enabled=true
```

```
snmp.version=V3
```

```
snmp.port=161
```

```
snmp.v3.authprotocol=SHA
```

```
snmp.v3.authpassphrase=<password>
```

```
snmp.v3.privacyprotocol=AES128
```

```
snmp.v3.privacypassphrase=<password>
```

```
snmp.user=<SNMP username>
```

```
snmp.community=public
```

```
snmp.system.location=<SNMP location>
```

```
snmp.system.name=ArcMC Node 247
```

```
snmp.system.contact=<your support email address>
```

```
snmp.trap.enabled=true
```

```
snmp.trap.version=V3
```

```
snmp.trap.port=162
```

```
snmp.trap.nms=<IP address of NNMI>
```

```
snmp.trap.user=<SNMP trap user name>
```

```
snmp.trap.community=public
```

```
snmp.trap.v3.authprotocol=SHA
```

```
snmp.trap.v3.authpassphrase=<password>
```

```
snmp.trap.v3.privacyprotocol=AES128
```

```
snmp.trap.v3.privacypassphrase=<password>
```

---

5. Give the file permission: 644 and owner: arcsight.

6. Copy the file `ARCSIGHT-EVENT-MIB.txt` file from `$ARCSIGHT_HOME/current/arcsight/aps/conf/` to location `/usr/share/snmp/mibs`. Give the file permission: 644 and owner: root:root.

7. Run the script `arcsight_snmpconf` script as a root user, as follows:

```
<ArcSight_Home>/current/arcsight/aps/bin/arcsight_snmpconf <ArcSight_Home>/userdata/platform/snmp.properties trap
```

8. Run the script a second time, as follows:

```
<ArcSight_Home>/current/arcsight/aps/bin/arcsight_snmpconf <ArcSight_Home>/userdata/platform/snmp.properties poll
```

This script will setup `/etc/snmp/snmpd.conf` file and restart the SNMP service.

9. Restart SNMP services: `service snmpd restart`

Note: To preserve the `SNMP V3 Trap oldEngineID` persistent in software ArcMC, set the `$ARCMC_HOME/userdata/platform/snmp_persist/snmpapp.conf` file to be immutable:

```
#chattr +i $file_path_of_snmpapp.conf
```

Follow the steps below to create the `snmpapp.conf` file if it does not exist in the snmp_persist folder:

a) In a text editor, create a file `<ARCSIGHT_HOME>/userdata/platform/snmp_persist/snmpapp.conf` with the following entry: `oldEngineID $VALUE`

$VALUE: copy the value from the `oldEngineID` entry to `/var/lib/net-snmp/snmpd.conf`
For example: `oldEngineID 0x80001f888011b5336c8d41895f00000000`

b) Give the file permission 600:
```
chmod 600 <ARCSIGHT HOME>/userdata/platform/snmp_persist/snmpapp.conf
```

c) Set the owner:
If arcmc is installed as root user: `# chown root:root <ARCSIGHT_HOME>/userdata/platform/snmp_persist/snmpapp.conf`

If arcmc is installed as arcsight user: `#chown arcsight:arcsight <ARCSIGHT_HOME>/userdata/platform/snmp_persist/snmpapp.conf`

d) Set immutable:
```
chattr +i <ARCSIGHT_HOME>/userdata/platform/snmp_persist/snmpapp.conf
```

10. In the rules editor, open the notification rule you wish to trigger an SNMP alert, and under **Notify Me,** select *SNMP*. Repeat for each rule you wish to trigger an SNMP alert.

Topology View

The Topology View displays your end-to-end data flow in browseable format. Shown are the logical relationships between network devices (event producers), connectors and Collectors, and their destinations in each of your ArcMC locations.

As your environment scales to thousands of source devices, you can use logical groupings (locations) to model subsystems, and datacenters can quickly trace issues and drill down on details.

To display the Topology View, click **Dashboard > Topology View.**

The left column highlights the current topology view. The available views are based on the locations defined in ArcMC.

Each of monitor icons represents a Device Product type, and the bubbles on the left of each monitor icon indicate the number of devices for each Device Product type.

The severity status of each item in the topology view is indicated by its color. Item status may be Healthy (green), Fatal (red), Critical (amber), Warning (yellow), or Unknown (gray).

The status indicates the severity as reported by the managed product. Hovering over the device product show more details of the severity status. Clicking on any of the severity levels opens the device details filtered by that product type and severity combination.

The **Devices** area shows any devices which are forwarding events in your network.

- To view the EPS (events per second) traffic to and from a device, mouse over the device.

The **Connectors/Collectors** area shows connectors and Collectors in the current topology view, specific to the location.

- To view the EPS (events per second) traffic to and from a connector, and get an overview of the connector status, mouse over the connector. Also shown are name, Device Type, Status, Path, Rule Violation (if any), Version, and ArcMC Managed.
- To drill down and view the health of the connector in detail, including health history, click the connector.
- In some cases, such as immediately following adding a connector node, an unmanaged connector may be displayed. This will be replaced with the connector data within a few collection cycles as data from the new connector is collected.
- Connectors displayed with the  symbol are included in a different location from the one currently selected for viewing.

Note: Transformation Hub drill-down mode is ArcMC-location specific.

The **Destinations** area shows connector destinations.

- To drill down and view the health of an ArcMC-managed destination in detail, click the destination.

The Topology View refreshes automatically once per minute. (You can toggle automatic data refresh with the **Auto Refresh** control.) To refresh the view manually, click **Refresh** in the toolbar.

The **Export** button allows users to export the devices list, status, last reported, eps, event size, connector, and customer URI into a CSV file.

When exporting the devices list, users can choose between **Use stored data** and **Real time device query** information. These options are displayed from a drop-down after clicking the **Export** button.

Note: The **Real time device query** option might take some time to be completed.

You can also toggle the display of legends for the graphic with the **Legends** control.

Click **Deployment View** to show your environment's Deployment View.

Note: If any are present, unmanaged connectors (or other nodes) in your network are noted as such in the Topology View. ArcMC will have no visibility into unmanaged connectors, nor any visibility of traffic from those nodes. Various scenarios for such views, and the results of each scenario, are detailed here. To get the most complete and accurate picture of your network, you are strongly encouraged to use ArcMC to manage all connectors which are part of your logical topology.

Deployment View

The Deployment View shows the physical relationships between network devices (event producers), connectors, their hosts, and their destinations in each of your ArcMC locations.

To display the Deployment View, click **Dashboard > Deployment View.**

The left column highlights the current deployment view. The available views are based on the physical hosts.

Each of the monitor icons represents a Device Product type, and the bubbles on the left of each monitor icon indicate the number of devices for each Device Product type.

The severity status of each item in the topology view is indicated by its color. Item status may be Healthy (green), Fatal (red), Critical (amber), Warning (yellow), or Unknown (gray).

The status indicates the severity as reported by the managed product. Hovering over the device product shows more details of the severity status. Clicking on any of the severity levels opens the device details filtered by that product type and severity combination.

The **Devices** area shows any devices which are forwarding events in your network.

- To view the EPS (events per second) traffic to and from a device, mouse over the device.

  The **Connectors/Collectors** area shows connectors and Collectors in the current topology view.
- To view the EPS (events per second) traffic to and from a connector, and get an overview of the connector status, mouse over the connector. Also shown are name, Device Type, Status, Path, Rule Violation (if any) and ArcMC Managed.
- To drill down and view the health of the connector in detail, including health history, click the connector.
- In some cases, such as immediately after adding a connector node, an unmanaged connector may be displayed. This will be replaced with the connector data within a few collection cycles as data from the new connector is collected.
- Connectors displayed with the  symbol are included in a different location from the one currently selected for viewing.

  The **Destinations** area shows connector destinations.
- To drill down and view the health of an ArcMC-managed destination in detail, click the destination.

The Topology View refreshes automatically once per minute. (You can toggle automatic data refresh with the **Auto Refresh** control.) To refresh the view manually, click **Refresh** in the toolbar.

The **Export** button allows users to export the devices list, status, last reported, eps, event size, connector, and customer URI into a CSV file.

When exporting the devices list, users can choose between **Use stored data** and **Real time device query** information. These options are displayed from a drop-down after clicking the **Export** button.

Note: The **Real time device query** option might take some time to be completed.

You can also toggle the display of legends for the graphic with the **Legends** control.

Click **Topology View** to show the topological relationships in your environment.

Prerequisites for Instant Connector Deployment

The following are prerequisites for Instant Connector Deployment.

- You must set up one or more deployment templates.
- Instant Connector Deployment is supported for accounts using SSH key authentication, but not supported for SSH with passphase authentication. To enable SSH key authentication, the SSH key needs to be set up between a non-root user of ArcMC and a user of the remote host that will be used for deployment.
- In addition, it is strongly suggested you consult the Configuration Guide for the connector you plan to deploy before deployment, to understand any special considerations or features of the connector being installed.
- For more information regarding Connector destinations, please see the Smart Connectors User's Guide.
- The below prerequisites are not present by default on Linux 8.x, unlike in previous Linux versions (e.g. Linux 6.x and 7.x). Perform the following steps for RHEL/CentOS 8.1 on the machine where the ArcMC is or will be installed, and in the target Linux host (the VM where the Connector/Collector will be deployed):

  a. Install python2:

  For RHEL/CentOS 7.x:
  ```
  sudo yum install -y python2
  ```

  For RHEL/CentOS 8.x:
  ```
  sudo dnf install -y python2
  ```

  b. Create a symlink:

  ```
  sudo ln -s /usr/bin/python2 /usr/bin/python
  ```

  c. Install libselinux-python package:

  For RHEL/CentOS 7.x:
  ```
  sudo yum install -y libselinux-python
  ```

For RHEL/CentOS 8.x:
`sudo dnf install -y libselinux-python`

> **Note:** If the `yum/dnf` command fails when installing `libselinux-python` on RHEL/CentOS, follow the steps below:
> - Download `libselinux-python-2.8-6.module_el8.0.0+111+16bc5e61.x86_64.rpm`
> - Install the package:
> `rpm -i libselinux-python-2.8-6.module_el8.0.0+111+16bc5e61.x86_64.rpm`

Additional Requirements For Windows Platforms

The following additional items are required for Instant Connector Deployment on Windows platforms.

- Only the local admin account is supported for deployment.
- The following preparatory steps are required when deploying on a Windows VM.

1. Enable PowerShell 4.0 or later.

https://www.microsoft.com/en-us/download/details.aspx?id=40855

2. Enable and configure PowerShell Remoting, with CredSSP authentication.

- Download the "ConfigureRemotingForAnsible.ps1" file:
  - https://github.com/ansible/ansible/blob/devel/examples/scripts/ConfigureRemotingForAnsible.ps1
- Open Power Shell as Administrator and run the following command:
  - `ConfigureRemotingForAnsible.ps1 -EnableCredSSP`

3. Enable TLS 1.2.

Instant Connector Deployment

Instant Connector Deployment enables rapid installation of connectors or Collectors where you need them in your environment. You perform Instant Connector Deployment right from the Deployment View.

Before proceeding, ensure you have met all the prerequisites for performing Instant Connector Deployment.

To instantly deploy a connector or Collector:

1. Click **Dashboard > Deployment View.**
2. In the **Connectors/Collectors** column label, click +, then select **Add Connector** or **Add Collector.**

3. On the **Add Connector** (or **Add Collector**) dialog, specify values for the connector to be added. Any fields marked with an asterisk (*) are required. Note that your selected deployment template may populate some fields automatically, but you may overwrite the values in these fields, if needed, for a particular deployment. **Exception:** you may only use the latest version of the connector you have uploaded to the repository when you set up deployment templates. You can add multiple destinations for each connector if needed.
4. To add multiple hosts to the Host list, in the Host drop-down, click Add Host, then select or specify the name of each host.

- **Collector Hostname:** The Collector hostname must match the hostname of the remote machine. If the remote machine does not have proper DNS /hostname setup correctly, specify the IP address of the remote machine as the hostname.
- **Collector Destination:** A Collector's destination must be the th-syslog topic on your ArcMC-managed Transformation Hub.
- **ArcSight SecureData Add-On Enablement:** To enable the ArcSight SecureData Add-on during deployment, under **Global Fields**, set **Format Preserving Encryption** to *Enabled*. For more information on enabling the SecureData Add-On, see "SecureData Encryption" on page 905.

4. To add multiple connectors (or Collectors) of the same type, click **Clone**. Then specify the information unique to the new connector (or Collector). When deploying multiple connectors, if any specified parameters (such as port number) are invalid, the deployment of all connectors in the job will fail.
4. Click **Install.** The connector or Collector is deployed. Alternatively, click **Add** to add more connectors to the deployment job.

Note: Instant Connector Deployment (including Collectors) is not supported from RHEL/CentOS 6.9 to a remote Windows host.

You can track and manage deployment jobs and issues using the Job Manager.

Note: If you later connect to a host where Connectors were installed through Instant Deployment, and run the Connector setup wizard from the command line, you should run agent setup from `$ARCSIGHT_HOME/current/bin` by setting the mode with option, `-i`, such as: `./runagentsetup.sh -i console` or `./runagentsetup.sh -i swing`, where options are swing, console, silent, and so on. For more information on options, see the Smart Connectors User's Guide.

Deployment on Linux Platform Using Non-root User

Follow these steps to install a connector/collector using non-root user through instant deployment feature.

Step 1

Option 1: Provide blanket sudo rights to non-root users:

1. Edit the sudoers file on the remote machine where the connector/collector will be deployed:

- Open the sudoers file:

```
# visudo
```

- Locate the following lines in the file:

```
## Allow root to run any commands anywhere root ALL=(ALL) ALL
```

2. Provide blanket sudo rights to non-root user below the previously mentioned line.

```
<non-root-user> ALL= (ALL) NOPASSWD:ALL
```

3. Save the file.
4. Specify this non-root user and password in the instant deployment job.

Option 2: Provide rights to non-root user to execute specific set of commands as mentioned below:

1. Edit the sudoers file on the remote machine where the connector/collector will be deployed:

- Open the sudoers file:

```
# visudo
```

- Locate the following lines in the file:

```
## Allow root to run any commands anywhere root ALL=(ALL) ALL
```

2. Add special rights to the non-root user below the previously mentioned line:

```
<non-root-user> ALL=(ALL) NOPASSWD: /bin/chown root\:root <connector_install_
dir>/current/config/agent/arc_<service_internal_name>, /bin/mv <connector_install_
dir>/current/config/agent/arc_<service_internal_name> /etc/init.d/, /bin/chmod 755
/etc/init.d/arc_<service_internal_name>, /bin/rm -rf /etc/init.d/arc_<service_internal_
name>
```

Note: <connector_install_dir> and <service_internal_name> should match exactly what the user will be entering in the instant deployment job. Provide these 4 commands in the sudoers for every connector/collector installation that will be done from ArcMC through this non-root user.

3. Save the file.
4. Specify this non-root user and password in the instant deployment job.

Step 2

Option 1: Use the Home user path.

The folder will be automatically created.

Option 2: Use an alternative path.

For non-root installation, users need to create the folder:

```
mkdir <path to folder>
```

Grant full permissions:

```
chmod 777 <path to folder>
```

Troubleshooting

This section describes possible scenarios in which users might encounter issues during the instant deployment of Connectors/Collectors.

Job does not start

Issue: Job does not start during a deployment(Connector/Collector) and no error message is displayed.

Possible solution: When the Job does not start and the status displayed is "Not Started", the possible reason is that the ArcMC has an 8.0 OS version or higher, and the python and associated library (libselinux) are not installed in the VM.

Job start but fails in the "Copy Installer" step

Issue: When a Job starts but fails in the "Copy Installer" step it will display the following message: "Aborting, the target uses SELinux but python bindings (libselinux-python) aren't installed!". This is related to a problem with the target host (where the Connector/Collector is going to be installed), the python or the SELinux are not installed there.

Possible solution: Go to the target host and install python and the SELinux library.

If the SSH certificate changes...

If the connector VM is redeployed, its SSH certificate will change and will no longer be able to use Instant Connector Deployment to deploy connectors to the VM. In this case, take the following steps to re-enable Instant Connector Deployment to the re-deployed VM.

1. Connect to the ArcMC's VM.
2. Change to the directory `/home/<non root user>/.ssh`
3. Open the file `known_hosts`.
4. Delete the line with the IP or hostname of the Connector's VM.
5. Save the file.

Deploying a Connector in Transformation Hub (CTH) (Standalone ArcMC)

A Connector in Transformation Hub (CTH) moves the security event normalization, categorization, and enrichment of connectors processing to the Docker containers environment of Transformation Hub, while reducing the work done by the Collector.

Ensure you have added a Transformation Hub host for a supported version (3.0 or later) before adding any CTHs. Transformation Hub 3.0 and later can have a maximum of 50 CTHs. Earlier versions can have up to 10 CTHs.

For a fresh installation, we provide 50 ports to support 50 of the CTHs.

If upgrading to Transformation Hub 3.1, you automatically get 50 ports for CTHs based on the new Transformation Hub images.

Note: CTHs cannot be configured with SecureData encryption. By default, CTH is set as TLS + CA.

To update the CTH port range:

1. Open `logger.properties` for editing.

   Create the file if it does not exist.

   ```
   /opt/arcmc/userdata/arcmc/logger.properties
   ```

   ```
   chown <non-root user>:<non-root user> logger.properties
   ```

   ```
   chmod 660 logger.properties
   ```

2. Add the following information to `logger.properties`.

```
# ============================================================
# CTH port range
# ============================================================
configuration.cth.end.port=39050
```

For Transformation Hub 3.3 and later use:

```
configuration.cth.end.port.post.th.32=32150
```

3. Restart the web process.

To deploy a CTH:

Note: To use the Global ID feature, Generator ID Manager has to be enabled in the ArcMC so that Generator ID can be set on the CTH.

1. Click **Dashboard > Deployment View.**
2. In the **Transformation Hub** column, click the managed Transformation Hub, then click the **+** icon.
3. On the **Deploy CTH** dialog, in **CTH Name**, specify a name for the CTH.

   The name must be smaller than 256 characters.
4. Under Acknowledgment mode, click the down arrow, then select the Acknowledgment mode for this CTH. (none/leader/all)

   The mode you select affects the safety of stored events in case of immediate system failure.

Acknowledgment Mode	Description
none	Acknowledgment off The producer will not wait for any acknowledgment from the server. The record will be immediately added to the socket buffer and considered sent. No guarantee can be made that the server has received the record in this case, and the retries configuration will not take effect (as the client won't generally know of any failures). The offset given back for each record will always be set to -1.
leader	Leader mode on The leader will write the record to its local log but will respond without awaiting full acknowledgment from all followers. In this case, if the leader fails immediately after acknowledging the record but before the followers have replicated it, the record will be lost.
all	All acknowledgments on The leader will wait for the full set of in-sync replicas to acknowledge the record; guaranteeing that the record will not be lost if at least one in-sync replica remains alive (strongest available guarantee). This is equivalent to the `acks=-1` setting.

5. Under **Destination Topics**, click the down arrow, then select one or more destination topics (CEF, Avro, or binary) for the CTH.
6. Select the corresponding ESM version. This is required for CTH to support Global ID when sending events to ESM 7.2
7. Click **Deploy**.

   Note: Please allow a few minutes after deploying or updating the CTH for the new values to be displayed.

   The CTH deployment job status can be viewed in Job Manager.

Once deployed, the CTH displays in Node Management on the Connectors tab, and in the Topology and Deployment View drill-down under the source topic.

Note: Destination topics must always be grouped the same for multiple CTHs. For example, if a CTH is sending events to both th-cef and th-esm topics, then any other CTH that sends events to one of these topics must also send events to the other topic, or events will be duplicated.

Editing a CTH

To edit a CTH:

1. Click **Dashboard > Deployment View.**
2. In the **Transformation Hub** column, click the managed Transformation Hub, then click the edit (pencil) icon.
3. On the **CTH Parameters** dialog, modify the name or destination topics, as needed.
4. Click **Redeploy**. The CTH is re-deployed. The job progress can be viewed in Job Manager.

Undeploying CTHs

To undeploy one or more CTHs:

1. Click **Dashboard > Deployment View.**
2. Click on the Transformation Hub box to drill down.
3. Click the edit (pencil) icon.
4. On the **CTH Parameters** dialog, click **X** next to any CTHs to be undeployed.
5. Click **Redeploy**. The job progress can be viewed in Job Manager.

Deploying Collectors

This section provides information about deploying Collectors, Non-TLS, TLS, and FIPS deployment.

1. Under **Dashboard > Deployment View**, click the  icon next to the **Connectors/Collectors** label and click **Add Collector.**
2. From the **Add Collector** window, under add collector details select the collector template.

Non-TLS Collectors Deployment

For Non-TLS Collector deployment:

1. Follow the steps in "Deploying Collectors" above section, and select the Syslog Daemon Collector template.
2. Scroll down to **Destination > Destination Template** and select the TH Collector Template.
3. Under **Kafka Broker Host(s):Port(s)**, confirm that port number is 9092, otherwise, change it accordingly.

4. Set the **Kafka Broker on SSL/TLS** flag to false.
5. Click **Install**.

TLS Collectors Deployment

For TLS Collector deployment:

1. Follow the steps in "Deploying Collectors" on the previous page section, and select the Syslog NG Daemon Collector template.
2. Scroll down to **Destination > Destination Template** and select the TH Collector Template.
3. Under **Kafka Broker Host(s):Port(s)** confirm that port number is 9093, otherwise, change it accordingly.
4. Set the **Kafka Broker on SSL/TLS** flag to true.
5. Click **Install**.

FIPS Collectors Deployment

FIPS can be enabled on Collectors either during the deployment of the Collector or while creating the Collector Configuration Template.

1. Follow the steps in "Deploying Collectors" on the previous page section.
2. From the **Add Collector** window, under add collector details select the collector template. Scroll down to the **Global Fields** section, and select **Enabled** from the **Enable FIPS mode** drop-down.
3. Replicate steps 2 through 4 in the "TLS Collectors Deployment" above section.
4. Click **Install**.

Post Deployment Collector Property Update

FIPS

1. Go to **Configuration Management > Bulk Operations.**
2. Click on the **Collector** tab, select one of the Collectors from the Manage Collectors table, and click **Properties.**
3. In the **Collector Property Update** window, click the  icon next to **Property List** and search for `fips.enable`.
4. Click **Edit** and set the Value to `true`.

> **Note:** When setting the `fips.enable` property to true, you need to modify the property's `agents[0].destination[0].params` bootstraphosts parameter port value to 9093, as well

as change the usessl parameter (SSL/TLS) value to `true`.

5. Click **Save**.

Non-TLS and TLS

For Non-TLS and TLS the process remains the same for steps 1 to 2 listed in "Post Deployment Collector Property Update" on the previous page In the **Collector Property Update** window, click the icon next to Property List and search for the `agents[0].destination[0].params` property. See the table below for the correct values.

Property	Parameter	Non-TLS	TLS
agents[0].destination[0].params	bootstraphosts	port value: 9092	port value: 9093
agents[0].destination[0].params	usessl	false	true

SecureData Encryption

To enable SecureData encryption, you must provide the SecureData server details in the Deployment Template for a connector.

Note: CTHs cannot be configured with SecureData encryption.

If any proxy settings are required, these must also be provided in the Deployment Template.

To explicitly specify that no proxy be used for the SecureData client, no parameters are needed in the Deployment Template. In addition, edit the file `/etc/profile.d/proxy.sh` (or its equivalent on Windows VM) and add/edit the line “export no_proxy and export NO_PROXY” with your SecureData server details.

If your SecureData client needs a certificate, then upload the valid certificate to ArcMC's cacerts repository when creating the deployment template.

After all settings are configured, and a connection is ensured from the connector host to the SecureData server, you can deploy the connector using the Instant Connector Deployment process.

Warning: SecureData settings may only be updated once. Once encryption is turned on, it may not be turned off. Make sure you wish to use encryption before activating it.

Managing Nodes

A *node* is a networked ArcSight product that can be centrally managed through ArcSight Management Center. Each node is associated with a single networked host which has been assigned a hostname, an IP address, or both.

Node types can include any of the following ArcSight products:

Management of Fusion ArcMC is not supported.

- Connector Appliances or Software Connector Appliances
- Logger Appliances or Software Loggers
- Containers, connectors, or Collectors
- Other ArcSight Management Centers, either software or Connector Hosting Appliances
- Transformation Hub

A single host, such as a single deployed Transformation Hub, can comprise multiple nodes for management purposes. In addition, a node can be in a parent or child relationship with other nodes.

You can perform any of the following node management tasks:

- View managed nodes by location, by host, or by node type.
- Add, view, edit, and delete locations for hosts.
- Add nodes from a host, import hosts from a CSV file, view and delete hosts, view all hosts in a location, update software on hosts, move hosts to different locations, and scan hosts for new connectors or containers.

For more information on adding hosts, see "About Adding a Host" on page 921.

The following topics are discussed here.

Node Management

To manage nodes, on the menu bar, click **Node Management > View All Nodes.** The Node Management UI displays. The Node Management UI comprises two panels:

- The left side displays the navigation tree.
- The right side displays the management panel, enabling you to perform management operations on items selected in the navigation tree.

The Navigation Tree

The navigation tree organizes managed nodes into a hierarchy, and comprises the following:

- **System**: Displays the entire set of nodes managed by Arcsight Management Center.
- **Location**: Individual locations are displayed under **System**, listed in the order in which they were added. Locations are logical groupings you can use to organize a list of hosts. For more information, see "Locations" on page 919.
- **Host**: Each location branch shows all hosts assigned to that location, listed by hostname, in the order in which they were added. For more information, see "Hosts" on page 920.
- **Node Types**: Each host branch shows all managed nodes associated with that host. A node can be any of the following types:
- **Connector Appliance or Software Connector Appliance:** Each Connector Appliance (hardware or software) is shown as a separate node.
- **Logger Appliance or Software Logger:** Each Logger (hardware or software) is shown as a separate node.
- **ArcSight Management Center:** Each ArcSight Management Center (hardware or software) is shown as a separate node.
- **Container:** If the host includes any containers, each is shown as a node.
- **Connector**: If a container node contains a connector, the connector is shown under the container node in which it is contained.
- **Collector**: If a container node contains a Collector, the Collector is shown under the container node in which it is contained.
- **Transformation Hub:** A managed Transformation Hub is shown as a node.

Since items in the tree are organized hierarchically, each item in the tree includes all branches displayed below it. For example, a **Location** branch includes all hosts assigned to that location. Click the wedge icon to toggle the view of any branch and any items included in the branch.

The Management Panel

Select an item in the navigation tree to display its details on one of the tabs in the central management panel. For example, to display the details of a host shown in the navigation tree, select the host in the tree. The management panel to the right of the tree will display details and controls pertaining to selected host.

Management Tabs

The tabs displayed in the management panel depend on the type of item selected in the navigation tree. The management tabs displayed will show detailed information associated with the selected item, depending on its position in the hierarchy.

Selected Item Type in Navigation Tree	Tabs Shown in Management Panel
System	Locations, Hosts, Containers, Connectors, Collectors, ConApps, Loggers, ArcMCs, TH Nodes
Location	Hosts, Containers, Connectors, Collectors, ConApps, Loggers, ArcMCs, TH Nodes
Host	Containers, Connectors, Collectors, ConApps, Loggers, ArcMCs, TH Nodes
Node	Connectors, Collectors, ConApps, Loggers, ArcMCs, TH Nodes

For example, if you selected a location item from the navigation tree, the **Hosts, Containers**, **Connectors, Collectors, ConApps**, **Loggers ArcMCs** and **TH Nodes** tabs would be shown. Each tab would display the items of the named type associated with the selected location, including details on those items.

Working with Items in the Management Panel

Selecting One or Multiple Items: To select an item from a list of items in the management panel, click the item. Use Shift+Click to select multiple adjacent list items, or Ctrl+Click to select multiple non-adjacent items.

Column Settings: Click the gear icon to change column settings:

- **Sorting:** To sort data by a column, select either **Sort Ascending** or **Sort Descending**.
- **Column Display:** To change the columns displayed in a table, select **Columns**. Then toggle one or more columns to display.
- **Filter:** To filter a list of items, select **Filters**. Then specify one or more filter criteria to display items matching those criteria.

Refreshing a List: To refresh the data in a list, click **Refresh** in the upper right corner.

Tab Controls

These controls are commonly displayed on all tabs in the management panel:

- **Toolbar Buttons:** Toolbar buttons enable operations related to the items on the tab.
- **Items Table:** Items corresponding to the tab header are displayed in a table. For example, locations are listed in tabular format on the Locations tab.
- **Bulk Operations Buttons:** On most tabs, bulk operations buttons enable you to perform operations on one or more items. Select one or multiple items in the list, then click the button to perform the indicated operation. For example, to delete multiple items such as hosts, select one or more hosts on the **Hosts** tab, then click **Delete**. The selected hosts would be deleted.

In addition, each tab may have controls individual to that item type. For example, the **Connectors** tab includes controls related to the management of connectors (see "Managing Connectors" on page 964).

The Locations Tab

The **Locations** tab displays all locations defined in Arcsight Management Center. The **Locations** tab includes these buttons:

Add Location	Adds a new location. For more information, see "Adding a Location" on page 919
Delete	Deletes one or more selected locations from ArcMC. For more information, see "Deleting a Location" on page 920

The **Locations** table displays these parameters for each location.

- **Name:** Location name.
- **Number of Hosts:** Number of hosts assigned to the location.
- **Action:** Drop-down includes a control for editing a location. For more information on editing a location, see "Editing a Location" on page 919.

For more information on managing locations, see "Locations" on page 919.

The Hosts Tab

The **Hosts** tab displays all hosts associated with the location selected in the navigation tree. The **Hosts** tab includes these buttons:

Add Host	Adds a host. Available on the **Hosts** tab when a location is selected in the navigation tree. For more information on adding a host, see "About Adding a Host" on page 921.
Move	Moves selected hosts to a new location. For more information, see "Moving a Host to a Different Location" on page 1055

Update Agent	Updates the ArcMC Agent on selected hosts. If the Agent is not currently installed, this button will install the Agent. For more information, see "Updating (or Installing) the ArcMC Agent " on page 1.
Delete	Deletes selected hosts from ArcMC. For more information, see "Deleting a Host" on page 1055

The **Hosts** table displays these parameters for each host:

- **Hostname:** Fully qualified domain name (FQDN) or IP address of the host. The hostname must match the hostname in the host's SSL certificate. (If IP address was used to add the host, then the certificate will match the IP address used.)
- **Path:** Path to the host.
- **Agent Version:** Version number of the Arcsight Management Center Agent running on the host.
- **Issues:** Status of any issues associated with the host. Possible indicators include:
  - *None:* No issues are associated with the host.
  - *Internet connection Not Present:* The host is currently not reachable by internet connection. Displayed when ArcMC is not able to connect to the Marketplace for retrieving parser upgrade versions. If the user environment needs a proxy server for an internet connection, configure the logger.properties file. If the user environment is an appliance, save the DNS settings on the **System Admin > Network** page.
  - *Valid Marketplace Certificate Not Found in ArcMC*: Displayed when the Marketplace certificate does not match the one found in ArcMC's trust store.
  - *Host Certificate Mismatch:* The hostname does not match the hostname in the SSL certificate. For instructions on downloading and importing certificates for the host, see "Downloading and Importing Host Certificates" on page 1052.
  - *ArcMC Agent Out of Date:* The host's Agent version cannot be upgraded from the managing ArcMC, or the Arcsight Management Center cannot communicate with the Arcsight Management Center Agent on the managed node. You may need to manually install the ArcMC Agent. For requirements and instructions, see "Installing the Agent" on page 1
  - *ArcMC Agent Stopped:* The Agent process on the host has been stopped.
  - *ArcMC Agent Upgrade Recommended:* The host's Agent version is older than the one on the managing ArcMC. An Agent upgrade is recommended.
  - *ArcMC Agent Uninstalled:* The Agent on the host has been uninstalled.
  - *ArcMC Agent Down:* The Agent on the host is not running.
  - *Error in REST Authentication*: The Transformation Hub node lacks the ArcMC certificate, ArcMC session ID, or ArcMC URL and port. To resolve this issue:

  - Make sure the user has the permission rights for the Transformation Hub operations.
  - Make sure the valid ArcMC certificate (with FQDN and .crt extension) is present in the Transformation Hub's location: /opt/arcsight/k8s-hostpath-volume/th/arcmccerts
  - Make sure that the ArcMC URL is updated with correct FQDN and port in ArcSight Installer > Transformation Hub Configuration > ArcMC_Monitoring field.
  - Note that each time the user replaces the ArcMC certificate to the TH's location, the TH's webservice pod has to be restarted for the new certificate to be read and updated in the trust store.
- **Model**: If the host is an appliance, this shows the ArcSight model number of the appliance. If the host is not an appliance, the label *Software* is shown.
- **Type**: Type of installation, either ArcMC Appliance or Software.
- **Version:**Version number of the software on the host.
- **Action:** Drop-down shows controls for executing host management tasks, which include:
  - Scanning a host
  - Downloading certificate details
  - Updating host credentials

For more information on host management, see "Hosts" on page 920.

The Containers Tab

The **Containers** tab displays all containers associated with the item selected in the navigation tree. For example, if you selected a location in the tree, since locations include hosts, the **Containers** tab would display all containers associated with all hosts in the selected location. The **Containers** tab includes these buttons:

Properties	This operation previously performed on this tab, is now performed on the new **Bulk Operations** page.
Certificates	Manage certificates on selected containers. For more information, see "Managing Certificates on a Container" on page 959.
FIPS	Enable or disable FIPS on selected containers. For more information, see "Enabling FIPS on a Container" on page 956.
Upgrade	Upgrades all connectors in selected containers. For more information, see "Upgrading All Connectors in a Container" on page 953.
Credentials	Manage credentials on selected containers. For more information, see "Changing Container Credentials" on page 952.

Logs	Manage logs on selected containers. For more information, see "Viewing Container Logs" on page 955.
Restart	Restart all connectors in selected containers. For more information, see "Restarting a Container" on page 955.
Delete	Deletes the selected containers from Arcsight Management Center. For more information, see "Deleting a Container" on page 952.

The **Containers** table includes the following columns:

- **Name:** Name of the container.
- **Path:** Path to the container.
- **Issues:** Status of any issues associated with the container.
- **Port:** Port number through which the container is communicating.
- **Framework Ver:** Framework version number of the container.
- **Parser Ver:** Parser version number of the container.
- **Status:** Status of the container. Possible values for container status are:
  - *Improper configuration: Initial default state.*
  - *Initializing connection:* The connector has a resolvable URL, but Arcsight Management Center has not logged in to the connector yet.
  - *Down*: There was an exception trying to execute the login command.
  - *Unauthorized:* The login command was executed, but login has failed.
  - *Connecting:* The login is in progress.
  - *Connected:* The login was successful.
  - *Empty:* Login successful, but the container doesn't have connectors.
  - *Initialized:* Login successful and the container has connectors.
  - *Unknown:* No information on status. To resolve, manually SSH to the system and restart the container.
- **Last Check:** Date and time of last status check.
- **Action:** Drop-down shows a variety of controls for executing container management tasks, which include:
  - Edit Container
  - Send Container Command
  - Add Connector
  - Run Logfu

- Download Certificate
- Display Certificates
- Deploy (to ArcExchange)
- Run FlexConnector Wizard

For more information on container management, see "Upgrading All Connectors in a Container" on page 953

The Connectors Tab

The **Connectors** tab displays all connectors associated with the item selected in the navigation tree. For example, if you selected a container in the navigation tree, the **Connectors** tab would show all connectors in the selected container. For the details on managing connectors, see "Managing Connectors" on page 964.

The Connectors tab will also show any deployed CTHs.

The **Connectors** tab includes these buttons, which perform operations on one or more selected connectors:

Add Connector	(Only shown when a container is selected in the navigation tree.) Adds a connector to the selected container.
Runtime Parameters	Edit the runtime parameters on selected connectors. For more information, see "Editing Connector Parameters" on page 967.
Destinations	Sets the destinations of selected connectors. For more information, see "Managing Destinations" on page 969.
Parameters	Sets parameters for selected connectors. For more information, see "Editing Connector Parameters" on page 967.
Delete	Deletes connectors from ArcSight Management Center. For more information, see "Deleting a Connector" on page 977.

The **Connectors** table displays the following parameters for each connector:

- **Name:** Name of the connector.
- **Path:** Path to the connector.
- **Group Name:** Name of the group for the connectors and CTHs. For connectors that have not been assigned to any group and older connector types no group name will be displayed.
- **Type:** Type of connector.
- **EPS In:** Events per second received by the connector.
- **EPS Out:** Events per second sent by the connector to its destination.

- **Cache:** Connector cache size. For more information on cache files, see the Smart Connectors User Guide.
- **Last Check:** Date and time of the last status check.
- **Action:** Drop-down shows a variety of controls for executing connector management tasks. These include:
  - Send Connector Command
  - Share a connector to ArcExchange
  - Edit a FlexConnector

For more information on connector management, see "Managing Connectors" on page 964.

The Connector Summary Tab

To view a single connector in detail, click the connector in the navigation tree. The toolbar on the summary tab includes the following buttons for operations on the connector:

Connector Command	Sends a command to the connector. For more information, see "Sending a Command to a Connector" on page 977.
Remove Connector	Removes the connector. For more information, see "Deleting a Connector" on page 977.
Run Logfu	Run Logfu diagnostics on the connector. For more information, see "Running Logfu on a Connector" on page 978.
Share	Shares the connector through ArcExchange. For more information, see "Sharing Connectors in ArcExchange" on page 981.

Tables below the toolbar show connector specifics, including basic connector data, parameters, and connector destinations. These tables include the following columns:

Connector Data

- **Type:** Type of connector.
- **Status:** Connector status.
- **Input Events (SLC):** Total number of events received by the connector since it was last checked (generally once per minute).
- **Input EPS (SLC):** Events per second received by the connector since it was last checked (generally once per minute).
- In addition, the columns to the right include tools for editing a connector, editing runtime parameters, adding a failover destination, and sending a destination command.

Connector Parameters

Click **Connector Parameters** to toggle display of this table. The **Connector Parameters** table includes:

- Click  to edit parameters.
- **Parameters:** Parameters can include connector network port, IP address , protocol, and other information.
- **Value:** Parameter value.

Table Parameters (WUC Connectors Only)

WUC connectors (only) display these parameters.

- **Domain Name:** Connector domain name.
- **Host Name:** Connector host name.
- **User Name:** Connector user name.
- **Security Logs:** Indicates whether security events are collected.
- **System Logs:** Indicates whether system events are collected.
- **Application:** Indicates whether application events are collected from the Common Application Event Log.
- **Custom Log Names:** List of custom application log names, if any.
- **Microsoft OS Version:** Microsoft operating system for the connector.
- **Locale:** Connector locale.

Destinations

Click **Destinations** to toggle display of this table. The **Destinations** table includes:

- Click  to add additional destinations.
- **Name:** Destination name.
- **Output Events (SLC):** Total number of events output by the connector to the destination since it was last checked (generally once per minute).
- **Output EPS (SLC):** Events per second output by the connector to the destination since it was last checked (generally once per minute).
- **Cached:** Total number of events cached to be transmitted to the destination.
- **Type:** Destination type. Destination types are described in the SmartConnector User's Guide.
- **Location:** Location of the destination.
- **Device Location:** Location of the device on which the destination is located.

- **Comment:** Comments on the destination.
- **Parameters:** Destination-specific parameters, such as IP address , port, and protocol.
- **Action Buttons:** Action buttons enable destination management tasks, such as editing the destination, editing the runtime parameters, adding a new failover destination, sending destination commands and removing the destination.

For more information on managing connectors, see "Managing Connectors" on page 964.

The ConApps Tab

The **ConApps** tab displays all hardware and software Connector Appliances associated with the item selected in the navigation tree. For example, if you selected **System** in the navigation tree, the **Connector Appliances** tab would display all Connector Appliances in Arcsight Management Center; if you selected a Location, the tab would display all Connector Appliances in the selected location.

The **Connector Appliances** tab includes the following button, which operates on one or more selected Connector Appliances:

Set Configuration	Sets the configuration for selected Connector Appliances. For more information, see "Setting a Configuration on ConApps" on page 941

The **Connector Appliances** table displays these parameters for each Connector Appliance:

- **Name:** Name of the Connector Appliance.
- **Path:** Path to the Connector Appliance.
- **Port:** Port number through which the Connector Appliance is communicating.
- **Version:** Software version of the Connector Appliance.
- **Status:** Status of the Connector Appliance.
- **Last Check:** Date and time of last status check.
- **Action:** Drop-down shows a variety of controls for executing Connector Appliance management tasks, including the following:
  - Rebooting
  - Shutting down
  - Editing or removing a configuration

For more information on Connector Appliance management, see "Managing Connector Appliances (ConApps)" on page 939.

The Loggers Tab

The **Loggers** tab displays all hardware and software Loggers associated with the item selected in the navigation tree. For example, if you selected **System** in the navigation tree, the **Loggers**

tab would display all Loggers in Arcsight Management Center; while if you selected a Location, you would see all Loggers in that location.

The **Loggers** tab includes the following buttons, which perform operations on one or more selected Loggers:

Set Configuration	Sets the configuration for selected Loggers. For more information, see "Setting a Configuration on Loggers" on page 950.
Upgrade Logger	Upgrades selected Loggers. For more information, see "Upgrading a Logger " on page 948

The **Loggers** table displays these parameters for each Logger:

- **Name:** Name of the Logger.
- **Path:** Path to the Logger.
- **Port:** Port number through which the Logger is communicating.
- **Version:** Software version of the Logger.
- **Top Storage Use:** Displays the most used storage group and its percentage of storage.
- **Status:** Status of the Logger.
- **Last Check:** Date and time of last status check.
- **Action:** Shows controls for executing Logger management tasks, including the following:
  - Rebooting
  - Shutting down
  - Editing or removing a configuration

The ArcMCs Tab

The **ArcMCs** tab displays all Software ArcSight Management Centers and ArcSight Management Center Appliances associated with the item selected in the navigation tree. For example, if you selected **System** in the navigation tree, the **ArcMCs** tab would display all managed ArcSight Management Centers; while if you selected a Location, you would see all ArcMCs in that location.

The **ArcMCs** tab includes the following buttons, which perform operations on one or more selected ArcMCs:

Set Configuration	Sets the configuration for selected ArcMCs. For more information, see "Setting a Configuration on Managed ArcMCs" on page 945
Upgrade ArcMC	Upgrades selected ArcMCs. For more information, see "Upgrading ArcMC" on page 943

The **ArcMCs** table displays these parameters for each ArcMC:

- **Name:** Name of the Arcsight Management Center.
- **Path:** Path to the Arcsight Management Center.
- **Port:** Port number through which the Arcsight Management Center is communicating.
- **Version:** Software version of the Arcsight Management Center.
- **Status:** Status of the Arcsight Management Center.
- **Last Check:** Date and time of last status check.
- **Action:** Shows controls for executing ArcMC management tasks, including the following:
  - Rebooting
  - Shutting Down
  - Editing a configuration

For more information on managing other ArcSight Management Centers in Arcsight Management Center, see "Managing Other ArcSight Management Centers" on page 942.

The TH Nodes Tab

ArcMC can only manage a single Transformation Hub. However, the single managed Transformation Hub may have any number of Transformation Hub nodes, each of which can be managed and monitored by ArcMC. When you add a Transformation Hub as a host to ArcMC, you add all of its nodes.

The **TH Nodes** tab displays all Transformation Hub nodes present in the managed Transformation Hub. For example, if you selected **System** in the navigation tree, the **TH Nodes** tab would display all managed Transformation Hub nodes; while if you selected a location, you would see all Transformation Hub nodes in that location.

The tab displays these parameters for each managed Transformation Hub node:

- **Name:** Name of the Transformation Hub node.
- **Port:** Port number through which the Transformation Hub node is communicating.
- **Type:** Type of Transformation Hub node.
- **Last Check:** Date and time of last status check.

For more information on managing Transformation Hub in Arcsight Management Center, see "Managing Transformation Hub" on page 1036.

The Collectors Tab

The **Collectors** tab displays all Collectors associated with the item selected in the navigation tree. For example, if you selected a container in the navigation tree, the **Collectors** tab would show all Collectors in the selected container.

The **Collectors** table displays the following parameters for each connector:

- **Name:** Name of the Collector.
- **Port:** Collector port.
- **Type:** Type of Collector.
- **Syslog Lines Received:** Events Received.
- **Custom Filtering:** Messages filtered out.
- **Status:** Collector status.
- **Last Check:** Date and time of the last status check.

For the details on managing Collectors, see "Bulk Operations" on page 1042.

Locations

A *location* is a logical grouping of hosts. The grouping can be based on any criteria you choose, such as geographical placement or organizational ownership. Locations are a useful way to organize a set of hosts.

For example, you could group all hosts in New York separately from hosts in San Francisco and assign them to locations named “New York” and “San Francisco”. Similarly, you could group hosts in a location named “Sales” and others in the location “Marketing”.

A location can contain **any number** of hosts. For information on adding hosts to locations, see "About Adding a Host" on page 921.

Note: Arcsight Management Center includes one location by default (called *Default*) but you may add any number of locations. The name of the Default location may be edited, and the location itself may be deleted.

Adding a Location

You can add any number of locations.

To add a location:

1. Click **Configuration Management > Bulk Operations**.
2. In the navigation tree, click **System** and click the **Location** tab.
3. Click **Add.**
4. Specify the name of the new location, and click **Save**.

Editing a Location

You can edit the name of a location.

To edit a location:

1. Click **Configuration Management > Bulk Operations**.
2. In the navigation tree, click **System**, then click the **Location** tab.
3. On the **Locations** tab, choose a location to rename.
4. Click **Edit**.
5. Specify the new name of the location, and click **Save**. The location is renamed.

Viewing All Locations

You can see all the locations that exist in Arcsight Management Center.

To view all locations:

1. Click **Node Management**.
2. In the navigation tree, click **System**, then click the **Locations** tab to view all locations.

Deleting a Location

When you delete a location from Arcsight Management Center, any hosts in the location (and their associated nodes) are also deleted.

> **Tip:** If you want to delete a location but still want to keep its hosts in Arcsight Management Center, relocate the hosts before deleting the location. See "Moving a Host to a Different Location" on page 1055.

To delete a location:

1. Click **Configuration Management > Bulk Operations**.
2. In the navigation tree, click **System**, then click the **Location** tab.
3. On the **Location** tab, choose one or more locations to delete.
4. Click **Delete**.
5. Click **Yes** to confirm deletion. The selected locations are deleted.

Hosts

A *host* is a networked system associated with a unique IP address or hostname. A host can be an ArcSight appliance, or a system running an ArcSight software product, such as Software Logger.

For information on adding hosts to manage, see "About Adding a Host" on the next page.

About Adding a Host

After a host is added to Arcsight Management Center, ArcSight products on the host becomes *nodes,* and can be managed. For example, adding a host running Connector Appliance with 4 containers would add 5 nodes to Arcsight Management Center: the Connector Appliance itself, and each container.

Prerequisites for Adding a Host (for each Host Type)

Connection Information for Adding a Host

Host Type	Required Information
Appliance with Local Connectors (includes ArcSight Management Center Appliance, Connector Appliance, or Logger Appliance (L3XXX))	• Hostname (FQDN) or IP address . Hostname or IP must be resolvable by ArcSight Management Center: either through DNS for a hostname, or directly for an IP address. If hostname is used, the hostname entered must match the hostname from the host's SSL certificate. (If the FQDN fails to resolve, restart the web service.) • Authentication credentials (username and password) for logging into the host. If the host is configured for external authentication, such as LDAP or RADIUS, use the external authentication credentials, if possible, or use the fall back credentials. **Note:** See "Node Authentication Credentials" on page 924 for more information about authentication credentials.
	• Authentication credentials (username and password) for any local containers. If the appliance includes multiple containers, then the credentials for each container must be identical. For example, if the username and password for one container managed by a Connector Appliance is *myusername* and *mypassword*, then *myusername* and *mypassword* must be the credentials for all local containers managed by the same Connector Appliance.
Appliance without Local Connectors (includes Logger Appliance (non-L3XXX))	• Hostname (FQDN) or IP address . Hostname or IP must be resolvable by ArcSight Management Center: either through DNS for a hostname, or directly for an IP address. If hostname is used, the hostname entered must match the hostname from the host's SSL certificate. (If the FQDN fails to resolve, restart the web service.) • Authentication credentials (username and password) for logging into the host. If the host is configured for external authentication, such as LDAP or RADIUS, use the external authentication credentials, if possible, or use the fall back credentials. **Note:** See "Node Authentication Credentials" on page 924 for more information about authentication credentials.

Connection Information for Adding a Host, continued

Host Type	Required Information
Software Form Factor (includes Software ArcSight Management Center, Software Connector Appliance, or Software Logger)	• Hostname (FQDN) or IP address. Hostname or IP must be resolvable by ArcSight Management Center: either through DNS for a hostname, or directly for an IP address. If hostname is used, the hostname entered must match the hostname from the host's SSL certificate. (If the FQDN fails to resolve, restart the web service.) • Authentication credentials (username and password) for logging into the host. If the host is configured for external authentication, such as LDAP or RADIUS, use the external authentication credentials if possible, or use the fall back credentials. **Note:** See "Node Authentication Credentials" on page 924 for more information about authentication credentials. • Port number assigned to the product.
Connector (includes SmartConnectors of all types)	• Hostname (FQDN) or IP address. Hostname or IP must be resolvable by ArcSight Management Center: either through DNS for a hostname, or directly for an IP address. (If the FQDN fails to resolve, restart the web service.) • Authentication credentials (username and password) for the connector. **Note:** See "Node Authentication Credentials" on page 924 for more information about authentication credentials. • Optionally, specify an inclusive port range separated by a hyphen (such as 9004-9008) to scan a port range for all connectors. **Note:** If the port range includes multiple connectors, then the credentials for each connector in the range must be identical. For example, if the username and password for one connector in the range was *myusername* and *mypassword*, then *myusername* and *mypassword* must be the credentials for every connector in the port range. **Note:** Prior to adding a software-based SmartConnector as a host, you must prepare the Smart Connector as explained in SmartConnectors on ArcMC.

Connection Information for Adding a Host, continued

Host Type	Required Information
Collector	• Hostname (FQDN) or IP address . Hostname or IP must be resolvable by ArcSight Management Center: either through DNS for a hostname, or directly for an IP address. (If the FQDN fails to resolve, restart the web service.) • Authentication credentials (username and password) for the Collector. **Note:** See "Node Authentication Credentials" on the next page for more information about authentication credentials. • Optionally, specify an inclusive port range separated by a hyphen (such as 48080-48088) to scan a port range for all Collectors. **Note:** If the port range includes multiple Collectors, then the credentials for each Collector in the range must be identical. For example, if the username and password for one connector in the range was *myusername* and *mypassword*, then *myusername* and *mypassword* must be the credentials for every Collector in the port range.
Transformation Hub - Non-Containerized Deployment	• Hostname (FQDN) or IP address. Hostname or IP must be resolvable by ArcSight Management Center: either through DNS for a hostname, or directly for an IP address. (If the FQDN fails to resolve, restart the web service.) • Port number for the Transformation Hub (default 32080) • In order to add Transformation Hub as a host, the active user must belong to an ArcMC permission group with rights to do so. By default, the admin user has such rights. **Note:** Prior to performing the Add Host process, you need to generate the ArcMC certificate with complete FQDN and download the .crt file, then copy the certificate file to your Kubernetes master node. See Preparing to Add Transformation Hub as a Host for details on this process.

Transformation Hub - Container Deployment Foundation (CDF)	• Virtual FQDN or Virtual IP (VIP) address.
	VIP must be resolvable by ArcSight Management Center: either through DNS for a hostname, or directly for a VIP address. (If the FQDN fails to resolve, restart the web service.)
	• Port number for the Transformation Hub (default 32080)
	• The following Kubernetes cluster parameters:
	  • Cluster Port (default 443)
	  • Cluster Username and Password
	  • Contents of the certificate file. For more details, see here.
	• In order to add Transformation Hub as a host, the active user must belong to an ArcMC permission group with rights to do so. By default, the admin user has such rights.

- **An SSL Certificate:** An SSL certificate must be generated for any of the following host types to be managed:
  - Connector Appliance or Software Connector Appliance
  - Logger Appliance or Software Logger
  - Transformation Hub (any version)
  - ArcSight Management Center Appliance or Software ArcSight Management Center

The hostname in the certificate must match the hostname you are adding to Arcsight Management Center. For more information on generating certificates for these host types, consult the ArcSight Administrator's Guide for each product. (If a host to be added already has a certificate installed, you can use the existing certificate, as long as the hostname on the certificate matches the hostname of the host you are adding.)

> **Note:** If the hostname does not match the hostname in the SSL certificate, you can regenerate a matching certificate by doing one of the following:
>
> - For a hardware appliance, in **System Admin > Network**, click the **NICS** tab. Under **Host Settings**, note the entry in the Hostname field. (This is the value you should use to add the host to Arcsight Management Center.) Click **Restart Network Service.** Then, in the navigation menu, under **Security**, pick **SSL Server Certificate**. Click **Generate Certificate**. A new certificate will be generated that matches the hostname from the **NICS** tab.
> - For software form factor, in **System Admin > SSL Server Certificate**, under **Enter Certificate Settings**, verify that the hostname from the NICS tab noted previously is entered in the **Hostname** field. Then, click **Generate Certificate**. A new certificate will be generated that matches the hostname from the **NICS** tab.

- **Check for Agent Installation:** Check the table under "Installing the Agent" on page 1 to determine if the ArcMC Agent needs to be installed on a host prior to adding it to ArcMC. For some host types, the Agent will be installed automatically upon adding a host.

  > **Note:** Perl is required for the automatic installation of the ArcMC Agent. Ensure that Perl is installed on the host prior to attempting to add the host to ArcMC.

Node Authentication Credentials

ArcSight Management Center authenticates to each managed node each time it communicates with the node, using the node's authentication credentials—that is, username and password—you supply when first adding the host. If the host includes connectors or containers, then authentication credentials must also be supplied for these as well. (Exception: Transformation Hub does not require authentication credentials for individual nodes.) As a result, valid credentials for each node are required when adding a host.

Determining a Node's Credentials:

Consult the system administrator for each managed node to determine its current login credentials. Each ArcSight product ships with a default set of credentials. However, for optimal security, it is expected that the default credentials are changed as soon as possible by the administrator, so the default credentials may no longer be valid for authentication.

- For default credentials for ArcSight products, consult the relevant product administrator's guide. (For SmartConnector default credentials, consult the SmartConnector User's Guide, available from the Micro Focus Community.)
- Some products can be configured by administrators to use external authentication, in which case the external authentication credentials or fallback credentials should be provided when adding the host to Arcsight Management Center. (SmartConnectors may not be configured for external authentication.)

Changed or Expired Credentials

If the username or password on a node are changed (or expire) any time after the node is added to ArcSight Management Center, then the node will no longer be managed. However, it will still appear in the list of managed nodes. For example, on some hosts, passwords are set to expire automatically after some time period, which would prevent successful authentication by Arcsight Management Center using the node's initial credentials. To avoid this issue, you may wish to use node credentials that do not expire. To continue management of node on which the credentials have changed or expired, use the Update Host Credentials feature.

Dynamic Credentials

If authentication credentials are configured to change dynamically (such as with RADIUS one-time passwords), then instead of providing external authentication credentials, you can provide the credentials of a local user on the managed node who is permitted to use fallback authentication. Arcsight Management Center will then try to authenticate to the managed node using the external authentication method first, and if this fails, it will try to authenticate to the managed node using the local user credentials.

Managing SmartConnectors on ArcMC

ArcMC can remotely manage previously-installed, software-based SmartConnectors; however, the remote management feature is disabled on software SmartConnectors by default.

You can install several SmartConnectors on a single host if supported by the hardware. ArcSight certifies a maximum of 4 SmartConnectors on Windows hosts and 8 on Linux hosts.

To manage software-based SmartConnectors with ArcSight Platform, you need to enable remote management on each connector, as follows:

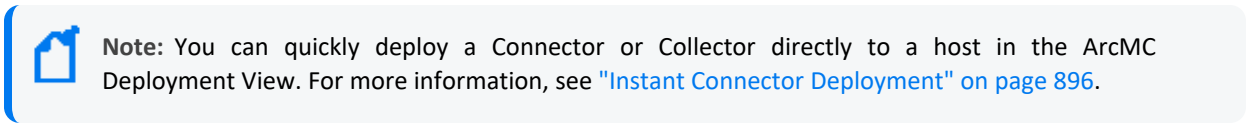
1. In a text editor, in the installation directory for the SmartConnector, open the file `/<install_dir>/user/agent/agent.properties.`
2. Add the line: `remote.management.enabled=true`
3. If desired, customize the connector's listening port. The default is 9001. To change this value, add the line: `remote.management.listener.port=<port_number>`, where `<port_number>` is the new port number.
4. Save the file.
5. Restart the SmartConnector for changes to take effect.

Adding a Host

Before adding a host, ensure that you have the required information for the host on hand. For more information, see "Prerequisites for Adding a Host (for each Host Type)" on page 921.

To add a host to ArcMC:

1. Click **Node Management**.
2. In the navigation tree, select a location to which you plan to add the host.
3. On the **Hosts** tab, click **Add Host.**
4. On the **Add a new Host** dialog, in **Hostname/IP,** specify either the hostname or IP address of the host.
5. In **Type,** select the type of node from the drop-down list.
6. Specify values for the required settings. (See About Adding a Host for the specific information required, based on the different type of nodes.)
   - In **Host Credentials** or **Connector Credentials**, specify the username and password required for authentication.
   - In **Port**, if required, specify the value of the port on which Arcsight Management Center will connect to the host.
7. Click **Add**. The host is added to Arcsight Management Center.



Note: You can quickly deploy a Connector or Collector directly to a host in the ArcMC Deployment View. For more information, see "Instant Connector Deployment" on page 896.

Adding a Host with Containers

When you add a host that includes containers (such as Connector Appliance), Arcsight Management Center also attempts to retrieve the SSL certificates from any containers that reside on the host, and add each container as a separate node. Containers on the remote host

can be managed only if Arcsight Management Center can authenticate using the certificates and supplied credentials. When the certificates are retrieved, you are prompted to import them into Arcsight Management Center.

Note: On Arcsight Management Center Appliance, all local containers are added automatically as hosts of type Software Connector.

Adding Transformation Hub Non-Containerized (THNC) as a Host

To add THNC as a managed host:

In the THNC server:

1. During the THNC setup script, add the arcmc host when the option is prompted. For example: hostname:443.
2. Get a copy of the ArcMC server certificate, with the extension *.crt from the system where ArcMC is running.
3. Copy the ArcMC certificate file and paste it on /opt/arcsight/th/current/cert/webservice/ directory.
4. Restart the THNC services.

In ArcMC:

1. Go to **Node Management > View All nodes**
2. From the left navigation tree, select the location where you want to add the THNC.
3. Click **Add Host**.
4. In the **Hostname/IP** field, type the fully qualified name of the THNC.
5. In the **Type** field, select **Transformation Hub - Non-Containerized Deployment.**
6. In the **Port** field, type 8080 and click **Add**.

Importing Multiple Hosts

To quickly and easily add multiple hosts in bulk, you can import a comma-separated values (CSV) file that lists the names and required attributes of the hosts to be added.

Note: Arcsight Management Center 1.0 used a slightly different file format for importing connector hosts. That file format is not supported by Arcsight Management Center 2.1. Use the file format described here instead.

Prerequisites for Importing Multiple Hosts

The following prerequisites apply to importing hosts.

- **Add Host Prerequisites:** Any prerequisites for the Add Host process also apply to importing multiple hosts by a CSV file. See "Prerequisites for Adding a Host (for each Host Type)" on page 921.
- **Valid CSV File:** Ensure the values in your CSV file are valid and correct. An import hosts job will fail immediately upon receiving an invalid or incorrect value. The CSV file format is described under "CSV File Format" below.
- **Stop the Agent 1.0 Process:** In addition, if any of the hosts to be imported are running the Arcsight Management Center 1.0 Agent, stop the Agent process on each such host before the import. (This is not needed for later versions of the ArcMC Agent.)

CSV File Format

The CSV (comma-separated value) file requires the following header line to be its first line:

```
location,hostname,type,host username,host password,connector
username,connector password,connector container name,port/port range,collector
username,collector password, collector port/port range
```

Each subsequent line represents one host to be imported. Each line must include values for the following comma-separated fields for each host:

```
<Location>, <Hostname>,<Host Type>,<Host Username>,<Host Password>,
<Connector Username>,<Connector Password>,<Connector Container
Name>,<Port/Port Range>,<Collector Username>,<Collector Password>,<Collector
Port/Port Range>
```

Note: The column `connector container name` (for instances in which users edit a container) has been added to the CSV file when importing or exporting hosts. If users don't want to import the values of this field, they can leave it blank. This applies for ArcMC versions 2.9.4 and later.

Collector port information will be exported as a single port. If more than one port is present, they will be exported individually. For example:

```
Default,example.com,Collector,,,,,collector,,48098
Default,example.com,Collector,,,,,collector,,48099
```

For importing hosts, users can import the Collector port information in a range or individually. For example:

```
Default,example.com,Collector,,,,,collector,,2001
Default,example.com,Collector,,,,,collector,,2002
```

```
Default,example.com,Collector,,,,,collector,,2001-2002
```

Some host types require values for all fields, and some are optional. An optional field with no value specified must still include a comma to represent the empty field.

Note: Only US ASCII characters are supported for import.

Host Field Values

Valid values for host fields are detailed in the following table. An asterisk (*) indicates a required field. An optional field with no value specified must still include a comma to represent the empty field.

Field	Description
Location*	Location to which the host will be assigned.
Hostname*	Hostname (FQDN) or IP address of the host. • FQDN or IP must be resolvable by ArcSight Management Center: either through DNS for a hostname, or directly for an IP address. • If hostname is used, the hostname entered must match the hostname from the host's SSL certificate. • For a hardware appliance, DNS must be configured on the managing appliance (**System Admin > DNS**).
Host Type*	Host type. Valid (case-insensitive) values are: • `appliance_with_local_connectors:` includes ArcSight Management Center Appliance, Connector Appliance and Logger Appliance (L3XXX) • `appliance_without_local_connectors:` includes Logger Appliance (non-L3XXX). • `software_form_factor:` includes Software ArcSight Management Center, Software Connector Appliance or Software Logger. • `software_connector:` includes all connectors and Collectors. • `Collector_software_connector:` indicates that connector and Collector reside on the same host. • `Collector:` includes all Collectors.
Host Username/ Password*	User name and password used to authenticate to the host. **Note:** See "Node Authentication Credentials" on page 924 for more information about authentication credentials.
Connector Username/ Password	Username and password used to authenticate to the connector. Required for hosts of type Appliance with Local Connector and Software Connector; otherwise optional. **Note:** See "Node Authentication Credentials" on page 924 for more information about authentication credentials.
Connector Container Name	Name of the container. For example: Syslog Container or SmartConnector Container.

Field	Description
Port/Port Range	Starting port or port range for connector scan. Valid values: • Port number • Port range • Comma-separated port numbers (for example, 9000,9004,9007) Notes: • *For software form factors*, port is required. • *For appliance form factors*, to add all local containers, leave the field blank. However, if any port numbers are entered, then certificates will be downloaded only for the specified port numbers, and only those containers will be imported. • *For connectors,* either a port or port range is required. If using port range, specify an inclusive port range, using a hyphen between starting and ending port. For example, a specified port range of 9001-9003 would scan ports 9001, 9002, and 9003. **Note:** If the port range includes multiple connectors, then the credentials for each connector in the range must be identical. For example, if the username and password for one connector in the range was *myusername* and *mypassword*, then *myusername* and *mypassword* must be the credentials for every connector in the port range.
Collector Username/ Password	Username and password used to authenticate to the Collector. **Note:** See "Node Authentication Credentials" on page 924 for more information about authentication credentials.
Port/Port Range	Port or port range for Collector scan. Valid values: • Port number • Port range • Comma-separated port numbers (for example, 9000,9004,9007)

An example of a valid import file, importing two hosts, is shown here:

```
location,hostname,type,host_username,password1,connector_
username,password2,port/port range,username,password3,port/port range
```

```
CorpHQ,hostname.example.com,software_connector,username,password,connector__
username,connector_password,9001-9005,collector_username,collector_
password,9006
```

```
EMEA,hostname2.example.com,appliance_without_local_connectors,
logger_user,logger_pword,,,,,
```

In this example, the first line would represent the required header line, the second line a Software Connector, and the third line would represent a Logger Appliance.

Import Hosts Procedure

Only a single Import Hosts job may be executed at one time.

Note: Importing Transformation Hub host in ArcMC is not supported. Please add Transformation Hub host to ArcMC through the "Adding a Host" on page 926 process.

To import hosts from a CSV file:

Note: Before beginning the import, stop the Agent processes on any hosts running version 1.0 of the ArcMC Agent.

1. Create and save your CSV file in a text editor.
2. Log into Arcsight Management Center.
3. Select **Node Management > Import Hosts.** The Import Hosts wizard starts.
4. Click **Browse**, and browse to the location of your hosts CSV file.
5. Click **Import**. The hosts are imported as a background job.

   If the CSV file is valid, connector certificates are retrieved automatically so that Arcsight Management Center can communicate with each connector in a container. The Upload CSV wizard lists the certificates. (To see certificate details, hover over the certificate.).

Automatic installation of the ArcMC Agent may increase the time required for the Import Hosts job.

- Select **Import the certificates...**, then click **Next** to import the certificates and continue.
- Select **Do not import the certificates...**, then click **Next** if you do not want to import the certificates. The Upload CSV wizard does not complete the upload CSV process.

  Note: The Import Hosts wizard does not complete the upload if certificate upload failed for any of the connectors in a container, or if any of the certificates failed to import into the trust store.

1. The Import Hosts job executes.

Import Hosts Job Logs

Arcsight Management Center logs the results of all Import Hosts jobs. Each job produces a new log, named `import_hosts_<date>_<time>.txt`, where `<date>` and `<time>` are the date and time of the import hosts job.

- For Software ArcSight Management Center, logs are located in the directory `<install_dir>/userdata/logs/arcmc/importhosts.`

- For ArcSight Management Center Appliance, logs are located in the directory `opt/arcsight/userdata/logs/arcmc/importhosts`.

Log Format

Each entry in the log will show the success or failure of each host import attempt, in the following format:

```
<User initiating job>, <CSV filename>, <Time of import host job
start>,<Hostname>,<Success/failure result>
```

For example:

```
admin, my_csv_file.csv, Tue Apr 08 14:16:58 PDT 2015, host.example.com, Host
added successfully
```

If the import hosts job has failed due to one or more invlaid entries in the CSV file, the result file will show the parsing error details with the line number and error.

For example:

```
Line [1] has [connector password] field empty. [connector password] field is
required for this host type.
```

Exporting Hosts

Exporting hosts from an Arcsight Management Center will create a CSV list of hosts managed by that Arcsight Management Center. (Password information is not included in this file.)

After adding passwords for each host to the file, you can then import this list of hosts into another Arcsight Management Center, using the Import Hosts feature described under "Importing Multiple Hosts" on page 927

Exporting hosts is most useful when you are reassigning management of hosts from one ArcMC to another.

For example, consider two ArcSight Management Centers, called ArcMC East and ArcMC West. ArcMC East currently manages 50 hosts. However, you are consolidating management of all hosts to the new ArcMC West. To do this quickly and easily, you would export the hosts from ArcMC East into a CSV file. Then, you would add an additional entry for ArcMC East to the CSV file.

After adding in password data for each host, you would import the resulting CSV file into ArcMC West. At the end of the process, all of ArcMC East’s hosts, and ArcMC East itself, would be managed by ArcMC West.

To export hosts in Arcsight Management Center:

1. Select **Node Management > Export Hosts.**
2. All hosts managed by the Arcsight Management Center are exported to the local CSV file (`exporthosts.csv`).
3. Optionally, open the file in a CSV editor. Add the password information for each host to the CSV file, then save the file.

Viewing All Hosts

You can see all the hosts managed by Arcsight Management Center, or view hosts by location.

To view all hosts:

1. Click **Node Management**.
2. In the navigation tree, click **System**. (To view by location, click the location you wish to view.)
3. Click the **Hosts** tab. All managed hosts are displayed.

Viewing Managed Nodes on a Host

You can view all the managed nodes on a host, by host type.

To view managed nodes on a host:

1. Click **Node Management**.
2. In the navigation tree, click the location to which the host is assigned. Then, click the host.
3. Click the appropriate tab to view the node types for the managed host: **Containers, Connectors, Connector Appliances, Loggers,** or **ArcMCs**.

Generator ID Manager

Every event generated by an ArcSight component will have a unique Global Event ID. This will help in identifying the events in case the same event is seen in multiple ArcSight components like Logger, ESM, and Transformation Hub.

- "Generator ID Management" on the next page
- "Setting Up Generator ID Management" on the next page
- "Getting Generator ID for Non-managed Nodes" on the next page
- " Setting Generator IDs on Managed Nodes" on the next page

Generator ID Management

This feature allows users to generate an ID to assign it to a non-managed product. Each assigned Generator ID should be unique for the ArcSight environment.

Setting Up Generator ID Management

1. On the top right side of the screen, click **Generator ID Manager.**
2. Select **Yes** to enable Generator ID Management in ArcMC.
3. Specify the numeric values between 1 and 16383 for the Generator ID range (**Start** and **End**) and click Save. ArcMC will set the generator ids for itself if not set already.
4. Restart all ArcMC processes to continue.

Getting Generator ID for Non-managed Nodes

1. Go to **Configuration Management** and select **Generator ID Management**.
2. Click **Assign a Generator ID.**
3. Select the **Event Producer Type**. Other fields are optional, click **Assign.**
4. Copy the ID by clicking the copy to clipboard icon and Click **OK**. A list of generated IDs will be displayed.

Setting Generator IDs on Managed Nodes

ArcMC will automatically set the generator IDs for each managed node when performing the following actions, if ArcMC is enabled as a Generator ID Manager:

Connectors

- Adding a Host version 7.11 or later.
- Scanning a Host
- Adding a Connector to a Container
- Connector upgrade to version 7.11 or later.
- Instant Deployment

Note: Multiple host deployment is disabled when the Generator ID Manager flag is enabled.

Logger

- Remote Upgrade: Upgrade from and to Logger version 6.7 or later.
- Adding a Host version 6.7 or later.

ArcMC

- Remote Upgrade: Upgrade from and to ArcMC version 2.9 or later.
- Adding a Host version 2.9 or later.
- Scanning a Host.

Transformation Hub

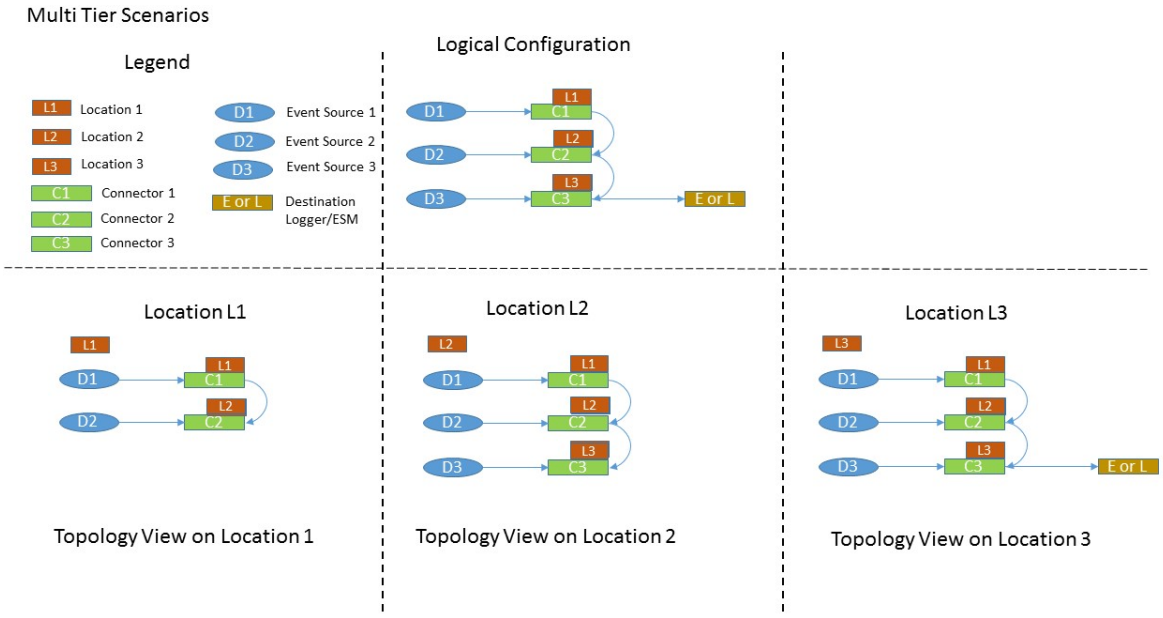
- Deploy CTH

The Topology View and Unmanaged Devices

This section details various scenarios for the inclusion of devices not managed by ArcMC in your network, and the effect of each scenario on the ArcMC Topology View. Particularly when connectors (or Collectors) are chained together in a multi-tier configuration, unmanaged products can block the view from their immediate downstream neighbor.

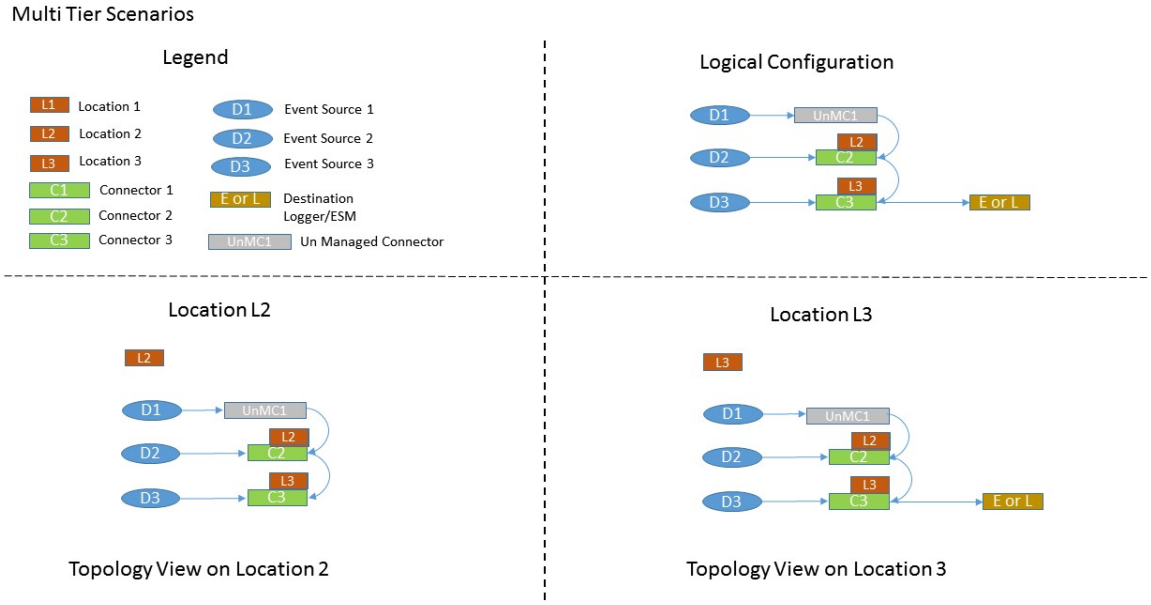
Scenario 1: No Unmanaged Devices

In this scenario, no umanaged products are included in the network. As a result, the ArcMC Topology view is unimpeded and gives an accurate picture of the logical topology as viewed from any location.



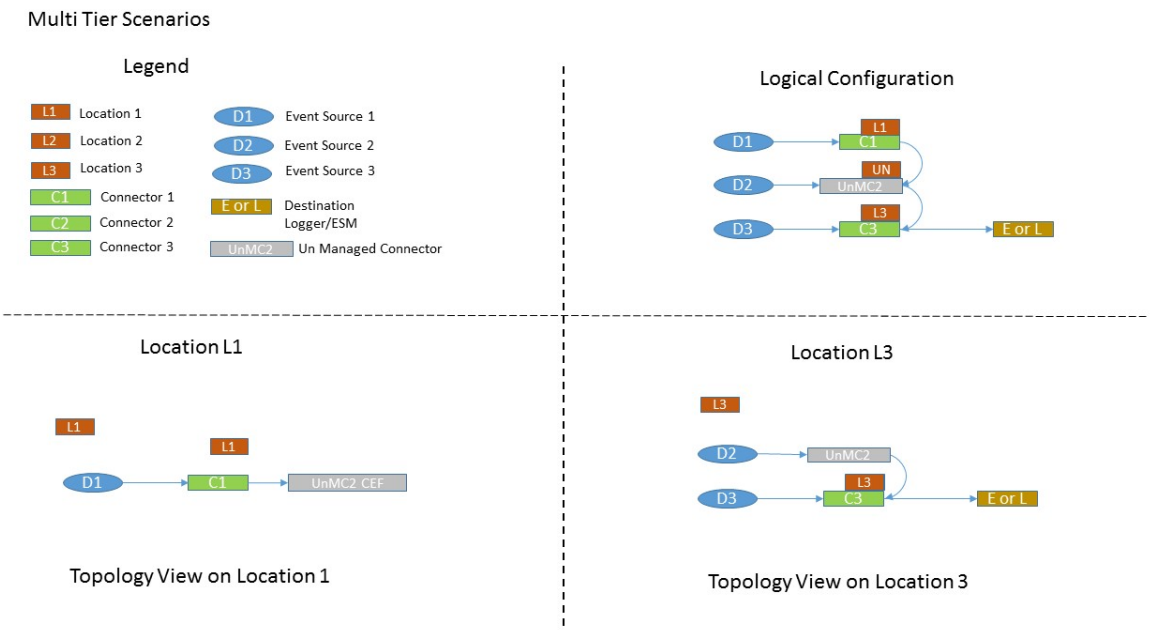
Scenario 2: Unmanaged Connector in Location L1

This scenario shows an unmanaged connector in location L1 and the results on the Topology View as seen from locations L2 and L3. No view is seen from L1, since it does not include any managed nodes. The view at the other downstream locations is as expected.



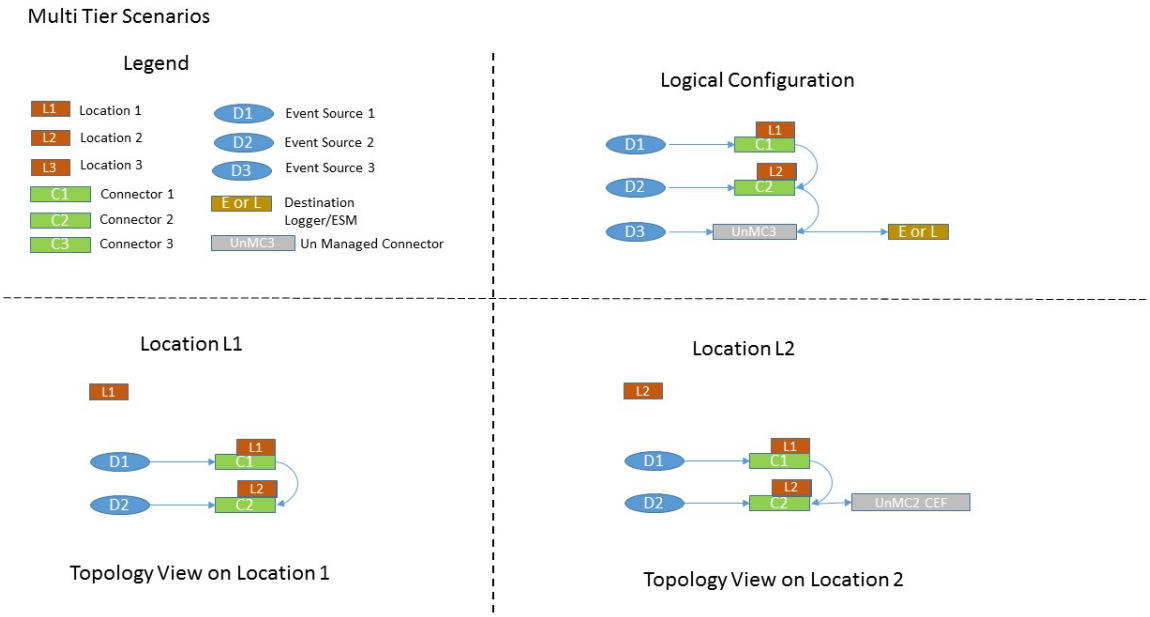
Scenario 3: Unmanaged Connector in Location L2

In this scenario, an unmanaged connector is located in Location L2 and chained to connectors in locations L1 and L2. This blocks the Topology view of L1 as seen from L3. In addition, the destination Logger or ESM shows no traffic from L1.



Scenario 4: Unmanaged Connector in Location L3

In this scenario, an unmanaged connector is in Location L3. This impedes an accurate Topology view of location 3. In fact, no traffic from locations L1 and L2 is shown for the destination Logger/ESM.



To get the most complete and accurate topological view, you are strongly encouraged to use ArcMC to manage all supported connectors (or Collectors) included in your logical topology.

Logger Consumption Report

The Logger Consumption Report includes information on your Logger data consumption. You can choose which managed Logger 6.1 (or later) nodes to include in the report.

To generate a Logger Consumption report:

1. Click **Administration > Application > Consumption Report.**
2. Use the **Add** and **Remove** arrows to add or remove nodes from the **Available Nodes** column to the **Selected Nodes** column.
3. Click **Run Report.** The report is generated for the selected nodes.
4. Click **+** to expand the data on any node to view licensing specifics.
5. To export the license report to PDF, click **Export to PDF.**
6. Specify a time range for the report.
7. Click **OK** to exit the report.

Report Data

The report displays the licensed value and actual value for data consumption by managed Loggers.

Value	Description
Licensed Consumption	Shows the data consumption to which your license entitles you. For individual ADP Loggers, the license limit will be shown as *Not Applicable*, since ArcMC tracks the overall data limit, not those of individual Loggers. **Note:** If an ADP Logger is managed by a version of ArcMC earlier than 2.5, then the license limit will be incorrectly shown in the report as *Unlimited*.
Actual Consumption	Shows the current value of data consumption. Click the value to display the Consumption Chart, which shows data consumption in detail.
Status	Click any status hyperlink to view individual Logger data for the last 30 days. Status values are shown as follows: *OK* if the actual value is less than or equal to the license value. *In Violation* indicates that the actual value exceeds the license value, which constitutes a violation of the terms of your license. Your license permits you a number of violations for each 30-day period, which is shown on the *Violations Last 30 Days* line. Click any hyperlink to view individual Logger data for the last 30 days.

Exporting PDF Reports

You can export up to 5 MB of data in PDF reports, this is the default size.

Follow the steps below to increase the limit of data to be exported perform:

1. Log in to the CDF Management Portal. See "Accessing the CDF Management Portal" on page 680 for more information.
2. From the left menu select **Deployment > Deployments**.
3. Click **... (Browse)** on the far right and choose **Reconfigure**. A new screen will open in a separate tab.
4. Select the **Fusion** tab and scroll down to the **ArcMC Configuration** section to specify the desired value for the "**Maximum Exported PDF Report Size** parameter.
5. Click **Save**. The ArcMC pod will be restarted

Managing ArcSight Products

ArcSight Management Center enables management tasks on a variety of ArcSight products, including the following:

- Hardware and Software Connector Appliances
- Hardware and Software Arcsight Management Centers
- Hardware and Software Loggers
- Containers
- Software connectors
- Transformation Hub

This chapter discusses the remote management of these products.

Managing Connector Appliances (ConApps)

You can perform any of the following management tasks on managed Connector Appliances or Software Connector Appliances using Arcsight Management Center:

- Reboot or shut down.
- Edit or remove a configuration.
- Set a configuration on one (or multiple) Connector Appliances.

Note: Not all Connector Appliance functionality is manageable through Arcsight Management Center. For a complete discussion of Connector Appliance features, see the Connector Appliance Administrator's Guide.

Rebooting a ConApp

To remotely reboot a managed Connector Appliance:

1. Click **Node Management.**
2. In the navigation tree, click **System**.
3. In the management panel, click **ConApps**.
4. In the list of Connector Appliances, locate the Connector Appliance to be rebooted.
5. In the **Action** drop-down of the Connector Appliance, select **Reboot ConApp**.
6. Click **Next** to confirm reboot.
7. The Connector Appliance is rebooted. Click **Done**.

Shutting Down a ConApp

To remotely reboot a managed Connector Appliance:

1. Click **Node Management.**
2. In the navigation tree, click **System**.
3. In the management panel, click **ConApps**.
4. In the list of Connector Appliances, locate the Connector Appliance to be shut down
5. In the **Action** drop-down of the Connector Appliance, select **Shutdown ConApp**.
6. Click **Next** to confirm shutdown.
7. The Connector Appliance is shut down. Click **Done**.

Editing or Removing a Configuration for a ConApp

You can edit a configuration on, or remove property values of a list configuration from, a managed Connector Appliance.

Editing or removing a configuration will overwrite the node's current configuration. This may make the node non-compliant with its current subscriptions.

To edit or remove a configuration on Connector Appliance:

1. Click **Node Management**.
2. In the navigation tree, click **System**.
3. In the management panel, click **ConApps**.
4. In the list of Connector Appliances, locate the desired Connector Appliance.
5. In the **Action** drop-down of the Connector Appliance, select **Edit/Remove Config.** The Update Configurations wizard is launched.
6. Review the dialog box, and click **Next**.
7. Follow the prompts to complete the wizard.
8. When the wizard is complete, click **Done**.

> **Note:** In order to edit a backup configuration on a Connector Appliance node, the node must have a scheduled backup to begin with.

Setting a Configuration on ConApps

You can set a configuration on one or multiple Connector Appliances using the Set Configuration wizard.

- For list configurations, use the Set Configuration wizard to append property values to an existing configuration on multiple Connector Appliances. Only new values will be appended. For more information on list configurations, see "List Configurations" on page 994.
- For non-list configurations, use the Set Configuration wizard to overwrite the configuration on multiple Connector Appliances.

> **Caution:** Setting a configuration on one or multiple Connector Appliances may make each Connector Appliance node non-compliant with its current subscriptions.

To set a configuration on one or more Connector Appliances:

1. Click **Node Management**.
2. In the navigation tree, click **System**.
3. In the management panel, click **Connector Appliances**.
4. In the list of Connector Appliances, select one or more Connector Appliances.
5. Click **Set Configuration**. The Set Configuration wizard is launched.
6. Review the dialog box, then click **Next**.
7. Follow the prompts to complete the wizard.

- Click **Add Row** to add a new Property to a list configuration, then specify values as needed.

8. The configuration is set on the selected Connector Appliances. Click **Done**.

Managing Other ArcSight Management Centers

You can perform any of the following management tasks on managed Software ArcSight Management Centers or Arcsight Management Center Appliances:

- Reboot or shut down.
- Edit or remove a configuration.
- Remotely upgrade an ArcSight Management Center.
- Set a configuration on one (or multiple) ArcSight Management Centers.

Rebooting an ArcMC

To remotely reboot a managed ArcSight Management Center:

1. Click **Node Management.**
2. In the navigation tree, click **System**.
3. In the management panel, click **ArcMCs**.
4. In the list of ArcSight Management Centers, locate the ArcSight Management Center to be rebooted.
5. In the **Action** drop-down of the ArcMC, select **Reboot ArcMC**
6. Click **Next** to confirm reboot.
7. The ArcSight Management Center is rebooted. Click **Done**.

Shutting Down an ArcMC

To remotely shut down a managed ArcSight Management Center:

1. Click **Node Management.**
2. In the navigation tree, click **System**.
3. In the management panel, click **ArcMCs**.
4. In the list of ArcSight Management Centers, locate the ArcSight Management Center to be shut down.
5. In the **Action** drop-down of the ArcMC, select **Shutdown ArcMC**.

6. Click **Next** to confirm shutdown.
7. The ArcSight Management Center is shut down. Click **Done**.

Editing or Removing a Configuration for ArcMC

You can edit a configuration on, or remove property values of a list configuration from, a managed ArcSight Management Center.

Editing or removing a configuration will overwrite the node's current configuration. This may make the node non-compliant with its current subscriptions.

To edit or remove a configuration on ArcSight Management Center:

1. Click **Node Management**.
2. In the navigation tree, click **System**.
3. In the management panel, click **ArcMCs**.
4. In the list of ArcSight Management Centers, locate the desired ArcSight Management Center.
5. In the **Action** drop-down, select **Edit/Remove Config**. The Update Configurations wizard is launched.
6. Review the dialog box, then click **Next**.
7. Follow the prompts to complete the wizard.
8. When the wizard is complete, click **Done**..

Note: In order to edit a backup configuration on an ArcMC node, the node must have a scheduled backup to begin with.

Upgrading ArcMC

Remote upgrades from ArcMC 2.9.2 to ArcMC 2.9.3 require a hot-fix to be applied, this hot-fix file that needs to be uploaded depends on the form factor, as follows:

Form Factor	Upgrade File Name	Comments
Appliance	arcmc-2190.enc	This file needs to be uploaded before remotely upgrading an ArcMC appliance from version 2.9.2 to 2.9.3
Software	arcmc-sw-2190-remote.enc	This file needs to be uploaded before remotely upgrading an ArcMC software from version 2.9.2 to 2.9.3

Note: The upgrade file name must not be changed.

To upload the file from the master ArcMC:

1. Download the hotfix file and store it in a secure network location. The file name should always be “arcmc-2190.enc” or “arcmc-sw-2190-remote.enc” depending on the form factor.
2. Click **Administration > Repositories.**
3. Select **Upgrade Files** from the navigation tree.
4. In the management panel, click **Upload.**
5. Click **Choose File** and browse to your hot fix file, then click Submit. The file is now uploaded.
6. Continue with the normal procedure outlined in the "Remote Upgrade Using Node Management" below

In ArcMC, you can remotely upgrade any of the following managed ArcMC types and versions.

Form Factor	Upgrade File Name	Can Upgrade From...	Can Upgrade To...	Comments
Appliance	`arcmc-<build number>.enc`	ArcMC version 2.0 or later	Any later ArcMC version.	
Software	`arcmc-sw-<build number>-remote.enc`	ArcMC version 2.1	Any later ArcMC version.	Remote operating system upgrade is not supported for software ArcMC, and, if required, must be performed manually.

Remote Upgrade Using Node Management

Remote upgrade first requires that you upload the appropriate file to your ArcMC repository first. You can then apply the upgrade file to managed ArcMCs.

To upload the upgrade file to your repository:

1. Download the ArcMC upgrade file for the upgrade version, as outlined in the table above, and store it in a secure network location.
2. Click **Administration > Repositories.**
3. In the navigation tree, pick **Upgrade Files.**
4. In the management panel, click **Upload**.
5. Click **Choose File** and browse to your upgrade file, then click **Submit**. The file is uploaded.

To remotely upgrade one or more managed ArcMCs:

1. Click **Node Management**.
2. In the navigation tree, click **System**.
3. In the management panel, click **ArcMCs**.
4. In the list of ArcMCs, select one or more ArcMCs for upgrade. (You may select only the form factor appropriate for the upgrade file type, as outlined above.)
5. Click **Upgrade ArcMC**. The Upgrade wizard is launched.
6. Review the dialog box, then click **Next**.
7. Follow the prompts to complete the wizard.
8. When the wizard is complete, click **Done**.

Setting a Configuration on Managed ArcMCs

You can set a configuration on one or multiple ArcSight Management Centers using the Set Configuration wizard.

- For list configurations, use the Set Configuration wizard to append property values to an existing configuration on multiple ArcSight Management Centers. Only new values will be appended. (For more information on list configurations, see "The Configurations Table" on page 993.).
- For non-list configurations, use the Set Configuration wizard to overwrite the configuration on multiple ArcSight Management Centers.

> **Caution:** Setting a configuration on one or multiple ArcSight Management Centers may make each ArcSight Management Center node non-compliant with its current subscriptions.

To set a configuration on one or more ArcSight Management Centers:

1. Click **Node Management**.
2. In the navigation tree, click **System**.
3. In the management panel, click **ArcMCs**.
4. In the list of ArcSight Management Centers, select one or more ArcSight Management Centers for which to set a configuration.
5. Click **Set Configuration**. The Set Configuration wizard is launched.
6. Review the dialog box, then click **Next**.
7. Follow the prompts to complete the wizard.

- Click **Add Row** to add a new Property to a list configuration, then specify values as needed.

8. The configuration is set on the selected ArcSight Management Centers. Click **Done**.

Managing SmartConnectors on ArcMC

ArcMC can remotely manage previously-installed, software-based SmartConnectors; however, the remote management feature is disabled on software SmartConnectors by default.

You can install several SmartConnectors on a single host if supported by the hardware. ArcSight certifies a maximum of 4 SmartConnectors on Windows hosts and 8 on Linux hosts.

To manage software-based SmartConnectors with ArcSight Platform, you need to enable remote management on each connector, as follows:

1. In a text editor, in the installation directory for the SmartConnector, open the file `/<install_dir>/user/agent/agent.properties.`
2. Add the line: `remote.management.enabled=true`
3. If desired, customize the connector's listening port. The default is 9001. To change this value, add the line: `remote.management.listener.port=<port_number>`, where `<port_number>` is the new port number.
4. Save the file.
5. Restart the SmartConnector for changes to take effect.

Managing Loggers

You can perform any of the following management tasks on managed Logger Appliances or Software Loggers using Arcsight Management Center.

- Reboot or shut down.
- Edit or remove a configuration.
- Set a configuration on one (or multiple) Loggers.
- Remotely upgrade a Logger.

> **Note:** Not all Logger functionality is manageable through Arcsight Management Center. For a complete discussion of Logger features, see the Logger Administrator's Guide.

Rebooting a Logger

To remotely reboot a managed Logger:

1. Click **Node Management.**
2. In the navigation tree, click **System**.
3. In the management panel, click **Loggers**.
4. In the list of Loggers, locate the Logger to be rebooted.
5. In the **Action** drop-down of the Logge, click **Reboot Logger**.
6. Click **Next** to confirm reboot.
7. The Logger is rebooted. Click **Done**.

Shutting Down a Logger

To remotely shut down a managed Logger:

1. Click **Node Management.**
2. In the navigation tree, click **System**.
3. In the management panel, click **Loggers**.
4. In the list of Loggers, select the Logger to be shut down.
5. In the **Action** drop-down of the Logger, select **Shut Down Logger**.
6. Click **Next** to confirm shut down.
7. The Logger is shut down. Click **Done**.

Editing or Removing a Configuration for a Logger

You can edit a configuration on, or remove property values of a list configuration from a managed Logger.

Editing or removing a configuration will overwrite the node’s current configuration. This may make the node non-compliant with its current subscriptions.

To edit or remove a configuration on a managed Logger:

1. Click **Node Management**.
2. In the navigation tree, click **System**.
3. In the management panel, click **Loggers**.

4. In the list of Loggers, locate the desired Logger.
5. In the **Action** drop-down of the Logger, select**Edit/Remove Config**. The Update Configurations wizard is launched.
6. Review the dialog box, then click **Next**.
7. Follow the prompts to complete the wizard.
8. When the wizard is complete, click **Done**..

> **Note:** In order to edit a backup configuration on a Logger node, the node must have a scheduled backup to begin with.

Upgrading a Logger

Remote upgrades to Logger 7 require a previous hot-fix to be applied if Logger's versions are any of the following:

• 6.7.0.8242

• 6.7.1.8253

• 6.7.1.8257

• 6.7.1.8262

The hot fix file to be uploaded is **preupgrade-logger-20190924.enc**. The name should be kept as is.

To upload the file from the master ArcMC:

1. Download the hotfix file and store it in a secure network location. The file name should always be preupgrade-logger-20190924.enc depending on the form factor.
2. Click **Administration > Repositories.**
3. Select **Upgrade Files** from the navigation tree.
4. In the management panel, click **Upload.**
5. Click **Choose File** and browse to your hot fix file, then click **Submit.** The file is now uploaded.
6. Continue with the normal procedure outlined in the "To remotely upgrade one or more managed Loggers:" on the next page

In ArcMC, you can remotely upgrade any of the following managed Logger types.

Form Factor	Upgrade File Name	Can Upgrade From Version...	Can Upgrade To Version...	Comments
Appliance	`logger-<build number>.enc`	6.0 or later	6.1 or later	The filename format for the remote upgrade file for Logger Appliance is `logger-<build number>.enc`
Software	`logger-sw-<build number>-remote.enc`	6.0 or later	6.1 or later	• The filename format for the remote upgrade file for software Logger is `logger-sw-<build number>-remote.enc` • Remote operating system upgrade is not supported for software Logger, and, if required, must be performed manually.

Note: Upgrading to Logger version 6.0 requires ArcMC Agent 1167.1 or later to be running on the managed Logger. Upgrade the Agent on the managed Logger before performing the upgrade to Logger 6.0.

To upload the upgrade file to your repository:

1. Download the Logger upgrade file for the upgrade version, as outlined in the table above, and store it in a secure network location.
2. Click **Administration > Repositories.**
3. In the navigation tree, pick **Upgrade Files.**
4. In the management panel, click **Upload.**
5. Click **Choose File** and browse to your upgrade file, then click **Submit**. The file is uploaded.

To remotely upgrade one or more managed Loggers:

1. Click **Node Management**.
2. In the navigation tree, click **System**.
3. In the management panel, click **Loggers**.
4. In the list of Loggers, select one or more Loggers. (You may only select one form factor type to upgrade.)
5. Click **Upgrade Logger**. The Upgrade wizard is launched.
6. Review the dialog box, then click **Next**.
7. Follow the prompts to complete the wizard.
8. When the wizard is complete, click **Done**.

Setting a Configuration on Loggers

You can set a configuration on one or multiple Loggers using the Set Configuration wizard.

- For list configurations, use the Set Configuration wizard to append property values to an existing configuration on multiple Loggers. Only new values will be appended. For example, if you had a common group of users on three Loggers, you could use the Set Configuration wizard to add the same new user to all three Loggers with a single action. (For more information on list configurations, see "The Configurations Table" on page 993.)
- For non-list configurations, use the Set Configuration wizard to overwrite the configuration on multiple Loggers.

> **Caution:** Setting a configuration on one or multiple Loggers may make each Logger node non-compliant with its current subscriptions.

To set a configuration for one or more Loggers:

1. Click **Node Management**.
2. In the navigation tree, click **System**.
3. In the management panel, click **Loggers**.
4. In the list of Loggers, select one or more Loggers for which to set a configuration.
5. Click **Set Configuration**. The Set Configuration wizard is launched.
6. Review the dialog box, and click **Next**.
7. Follow the prompts to complete the wizard.
   - Click **Add Row** to add a new Property to a list configuration, then specify values as needed.
8. The configuration is set on the selected Loggers. Click **Done**.

Managing Containers

A *container* is a single Java Virtual Machine (JVM) that can run up to four connectors. The exact number of connectors depends on your current service agreement and the type of connector.

Containers may run on ArcMCs, on Connector Appliances, and on L3XXX model Loggers. The number of containers that can be run at one time is based on the product license. Check under **System Admin > License & Update** for this information.

Scanning a managed host will ensure all currently running containers on the host (and the connectors associated with them) are accurately inventoried. For more information, see "Scanning a Host" on page 1053.

Note: A connector of any of the following types must be the single connector running in its container:

- Trend Micro Control Manager (TMCM)
- Syslog
- Windows Unified Connector (WUC)

Note: For Microsoft Windows Event Log (WINC), only one connector can be created on an ArcMC appliance.

Viewing All Containers

You can view all containers managed in Arcsight Management Center.

To view all containers:

1. Click **Node Management**
2. In the navigation tree, click **System**. (Alternatively, to view containers on a specific host, select the host from the navigation tree.)
3. Click the **Containers** tab to display the containers.

Viewing Connectors in a Container

You can see all the connectors in a container.

To view connectors in a container:

1. Click **Node Management**.
2. In the navigation tree, navigate to the container whose connectors you wish to view.
3. Click the tree branch corresponding to the container.
4. Click the **Connectors** tab. The connectors in the container are displayed.

Editing a Container

The default name for a container is *Container N*, where N is a sequential number that indicates the order in which the container was added. However, you can edit a container's default name.

To edit a container:

1. Click **Node Management**.
2. In the navigation tree, navigate to the host with container you wish to rename.

3. In the list of containers, locate the container you wish to edit.
4. In the **Action** drop-down of the container, click **Edit Container.**
5. In **Name,** specify the new container name, then click **Next**.
6. Click **Done**. The container is renamed.

Deleting a Container

When you delete a container, the connectors that it contains are also deleted.

To delete a container:

1. Click **Node Management**.
2. In the navigation tree, navigate to the host on which the container resides.
3. Click the **Containers** tab.
4. On the **Containers** tab, select one or more containers to delete.
5. Click **Delete**.
6. Click **OK** to confirm deletion. The selected containers are deleted.

Note: Containers on appliances can't be deleted.

Changing Container Credentials

You can change the user name and password associated with each container.

Caution: A container's default user name is `connector_user` and the default password is `change_me`. ArcSight strongly recommends that for optimal security, you should change each container's credentials to a non-default value before deploying it to production.

To change container credentials:

1. Click **Node Management**.
2. In the navigation tree, navigate to the host on which the container resides.
3. Click the **Containers** tab.
4. On the **Containers** tab, select one or more containers for which to change the credentials.
5. Click **Credentials**.
6. Follow the instructions in the wizard to update the credentials for the selected containers.

Sending a Command to a Container

You can run commands on a container to configure memory settings, pull an OPSEC certificate, generate a key, or restart the container.

To run a command on a container:

1. Click **Node Management**.
2. In the navigation tree, navigate to the host on which the container resides.
3. Click the **Containers** tab.
4. In the **Action** drop-down of the container, click **Send Container Command.** The Send Command wizard starts.
5. From the drop-down list, select the command you want to send, then click **Next**.
6. Specify appropriate values for the parameters and then click **Done**.

Upgrading All Connectors in a Container

You can upgrade all connectors in a container to a specific parser or framework version number.

Before Performing the Upgrade

Prior to performing a container upgrade, you will need to follow the steps below:

For connectors running 32-bit with version < 7.11, it is required to perform 32-bit to-64 bit migration before upgrading to a connector running >=7.11.

32-bit to 64-bit container migration.

1. Upgrade the appliance you currently have (a G8 migrated from conapp to ArcMC) to the latest ArcMC build.
2. Back up the container using the repositories page.
3. Emergency restore the container to the 64 bit connector AUP.
4. Restore the backup to the container using the repositories page.
   - **Case #1:** G8 appliances: model >= 6500 and restoreToVersion >= 7.9.0.8084 OR if connector is restored to any version less than 7.9.0, the connector will get restored to 32 bit, if connector is restored to any version greater than 7.9.0, the connector will get restored to 64 bit.

- **Case #2:** G9 appliances: model >= 6600 and restoreToVersion >= 7.2.1.7714.0 if connector is restored to any version less than 7.2.1 - restore should not be allowed, if connector is restored to any version greater than 7.2.1, the connector will get restored to 64 bit.

> **Note:** The above Emergency Restore to perform 32-bit to 64 bit connector migration does not support Appliances running on C5500 model.

To upload a version file to your repositories.

You can use a connector AUP file of the new parser or framework version in your ArcMC repository. If you opt to use this method, you will need to upload the version file to your repository as follows:

1. Click **Administration > Repositories.**
2. In the navigation tree, pick **Upgrade Files.**
3. In the management panel, click **Upload**.
4. Click **Choose File** and browse to your connector AUP file, then click **Submit**. The file is uploaded.

- Alternatively, instead of using a parser AUP file from the repository, you can download and use parser files from the ArcSight Marketplace. (Framework files are not available from the Marketplace.) Create your administrative account on the ArcSight Marketplace. If you have not created your Marketplace account, you are given an opportunity to sign up for an account during the parser upgrade process.

To perform the parser or framework upgrade on all connectors in a container:

> **Note:** Parser Remote Upgrade on Connector in Transformation Hub (CTH) is not supported from ArcMC. Parsers on CTH are updated during the Transformation Hub releases.

1. Click **Node Management.**
2. In the navigation tree, navigate to the host on which the container resides.
3. Click the **Containers** tab.
4. On the **Containers** tab, select one or more containers to upgrade.
5. Click **Upgrade**.
6. On the upgrade page, under **Select Upgrade Type,** choose either **Parser upgrade** or **Framework upgrade.**
7. Under **Select Upgrade Version,** from the drop-down list, choose the version to which you want to upgrade the selected containers. (You can select the number of parser upgrade

versions displayed in the drop-down as described in "Configuring ArcMC Parser Upgrades" on page 387)

a. For a parser upgrade, if the selected parser version is from the Marketplace and not the local repository, save your Marketplace credentials in ArcMC. This is a one-time task unless you wish to update these credentials.

8. Click **Upgrade**. The upgrade is performed on all containers.

Restarting a Container

Restarting a container will restart all the connectors in the container. You can restart multiple containers in bulk.

To restart one or more containers:

1. Click **Node Management.**
2. In the navigation tree, navigate to the host on which a container resides.
3. Click the **Containers** tab.
4. On the **Containers** tab, select one or more containers to restart.
5. Click **Restart**.
6. Click **Yes** to confirm restart. The selected containers are restarted.

Viewing Container Logs

You can retrieve and view the log files for one or more containers. The log files are in `.zip` format.

Container logs must be uploaded to the Logs repository before they can be viewed. For instructions on how to upload logs, see "Uploading a File to the Logs Repository" on page 648.

To retrieve and view container logs:

1. Click **Node Management**.
2. In the navigation tree, navigate to the host on which the container resides.
3. Click the **Containers** tab.
4. On the **Containers** tab, select one or more containers for which to view logs.
5. Click **Logs**.
6. Click **Next** to begin the **Retrieve Container Logs** process. When complete, click **Done**.
7. Click **Administration** > **Repositories.**

8. In the left panel, click **Logs**.
9. In the management panel, click  to retrieve the log files (in `.zip` format) you want to view.

Deleting a Container Log

You can delete unneeded container logs as necessary.

To delete a container log file:

1. Click **Administration > Repositories.**
2. In the left panel, click **Logs.**
3. In the management panel, on the list of logs, click  next to the log file you want to delete.
4. Click **OK** to confirm deletion.

Enabling FIPS on a Container

FIPS mode is supported on local, and remote connectors and Collectors running version 4.7.5 or later, but certain connectors do not support FIPS mode. For information about which connectors do not support FIPS mode, see the Installing FIPS-Compliant SmartConnectors document. Before enabling FIPS on a container that contains connectors running as a service, review the caveats listed in that document.

FIPS is disabled by default on ArcSight Management Center, but can be enabled as described under "FIPS 140-2" on page 1. After FIPS is enabled on the appliance, you can enable FIPS on a container. Any FIPS-compliant connector in that container (or one which is added later) will automatically communicate in FIPS mode.

- If the connector destination is ArcSight Manager, Connector Management automatically imports the ArcSight Manager certificate into its trust store and applies it to the container.
- However, if the connector destination is Logger, the Logger certificate must be uploaded manually and applied to the container.

A FIPS Suite B certificate must be uploaded manually, regardless of the connector destination, as described in under “Enabling FIPS Suite B on a Container”, below.

You enable or disable FIPS using the same procedure.

To enable or disable FIPS mode on a container:

1. Click **Node Management**.
2. In the navigation tree, navigate to the host on which the container resides.
3. Click the **Containers** tab.
4. On the **Containers** tab, select one or more containers for which to enable FIPS.
5. Click **FIPS**.
6. Follow the instructions in the wizard to update FIPS status.

Check that the appropriate CA certificates are in the trust store so that the connectors in the container can validate their configured destinations successfully. If necessary, add the appropriate certificates to the container.

Note: A 32-bit FIPS connector enabled cannot be remotely managed if it is installed on a 64-bit Linux system.

Enabling FIPS Suite B on a Container

Managed connectors can communicate in FIPS Suite B mode with their destination. A FIPS Suite B certificate must be imported manually and applied to the container, regardless of the connector destination.

Before you perform the following procedure, make sure FIPS mode is enabled on ArcSight Management Center, as described in "FIPS 140-2" on page 1.

To enable FIPS Suite B on a container:

1. Export the certificate for the connector destination (either ArcSight Manager or Logger) to a temporary directory. For example, on ArcSight Manager, from $ARCSIGHT_HOME/current/bin, specify the following command: `./arcsight runcertutil -L -n mykey -r -d /opt/arcsight/manager/config/jetty/nssdb -o /tmp/managercert.cer`
2. Upload the certificate from the temporary directory to the CA Certs Repository, as described in "CA Certs Repository" on page 649.
3. Enable FIPS on the container as described above.
4. Add the certificate on the container, as described in "Managing Certificates on a Container" on page 959 .
5. Click **Node Management**.
6. In the navigation tree, navigate to the host on which the container resides.
7. Click the **Containers** tab.

8. On the **Containers** tab, select one or more containers for which to enable FIPS Suite B.
9. Click **FIPS**.
10. Follow the instructions in the wizard to update FIPS Suite Bstatus.

Adding a Connector to a Container

Each container may hold up to 4 connectors.

To add a connector to a container:

1. Click **Node Management**.
2. In the navigation tree, navigate to the container to which you wish to add a connector.
3. On the **Connectors** tab, click **Add Connector**. The **Connector Setup** wizard starts.
4. Click **Next**, then follow the prompts to set up the new connector.

Note: Always change the default credentials of any new connector to non-default values. For more information, see "Changing Container Credentials" on page 952.

Running Logfu on a Container

The **Logfu** utility is a diagnostic tool that parses ArcSight logs to generate an interactive visual representation of the information contained within the logs. When event flow problems occur, it can be useful to have a visual representation of what happened over time.

To run Logfu on a container:

1. Click **Node Management**.
2. In the navigation tree, navigate to the host on which the container resides.
3. Click the **Containers** tab.
4. On the **Containers** tab, locate a container on which to run Logfu.
5. In the **Action** drop-down of the container, click **Run Logfu.**
6. The Logfu progress window is displayed as system data logs are retrieved and analyzed. Data is then displayed by **Group**, **Field**, and **Chart**.
   - In the **Group** box, choose which type of data you would like to view. The **Group** box lists all connectors within the chosen container, plus many other types of data such as memory usage and transport rates.
   - Then, choose one of the Group box **data points**. Depending on which data point you chose, a list of fields appears in the Field box below.

- Select a **field** to view. A graphic chart appears in the Chart box, providing rate and time information. The key at the bottom of the Chart box defines the data points mapped in the chart.
- To choose a different data point for analysis, click **Reset Data**.

7. When complete, close the display window.

Managing Certificates on a Container

Connectors require a Certificate Authority (CA) issued or self-signed SSL certificate to communicate securely with a destination. The Certificate Management wizard, available from the **Containers** tab, helps you add and remove certificates on a container. Using the wizard, you can:

- Add a certificate to a container.
- Add certificates in bulk, enabling multiple containers at once.
- Enable or disable a demo certificate on a container that is in non-FIPS mode only.
- Add a CA Certs file on a container that is in non-FIPS mode only.
- Remove a certificate from a container.

From the **Containers** tab and the **Connectors** tab, you can view details about the certificates applied to a container. See "Viewing Certificates on a Container" on page 962.

For information about resolving invalid certificates, see "Resolving Invalid Certificate Errors" on page 963.

Adding CA Certificates to a Container

You can add a single CA certificate to a container that is in FIPS mode or non-FIPS mode.

Note: Whenever you enable or disable FIPS mode on a container, check that the required certificates are present in the trust store and add them if necessary.

Click the icon next to the container name to see the type of certificate applied to it. Click **Display Certificates** from the action drop down to see the list of available certificates on the container.

Before you perform the following procedure, make sure the certificate you want to add is loaded in the CA Certs repository.

To add a single CA certificate to a container:

1. Click **Node Management**.
2. In the navigation tree, click **System**.

3. Click the **Containers** tab.
4. On the **Containers** tab, select one or more containers to which you wish to add certificates.
5. Click **Certificates**. The Certificate Management wizard starts.
6. Review the dialog box, then click **Next**.
7. Under **Choose an Action**, select **Add Certificate**, then click **Next**.
8. Follow the instructions in the wizard to add the certificate.

   If a container is down or a connector is running an older build, the wizard reports errors in the progress bar and on the Summary page.

Removing CA Certificates from a Container

You can remove CA certificates from a container when they are no longer needed. When you remove a CA certificate, the certificate is removed from the container's trust store; but it is **not** deleted from the repository.

Caution: Use caution when deleting certificates. When you delete a certificate on a container but the connector destination is still using that certificate, the connector can no longer communicate with the destination.

To remove CA certificates from a container:

1. Click **Node Management**.
2. In the navigation tree, click **System**.
3. Click the **Containers** tab.
4. On the **Containers** tab, select one or more containers to which you wish to remove certificates.
5. Click **Certificates**. The **Certificate Management** wizard starts.
6. Review the dialog box, then click **Next**.
7. Under **Choose an Action,** select **Remove certificate**, then click **Next**.
8. Select one or more certificates from the certificate list, then click **Next**. The certificates are removed from the list of certificates and no longer used. When you remove a certificate from a container in FIPS mode, the container restarts automatically.
9. The Certificate Management wizard displays the certificates that are removed successfully in a comma-separated list. Certificates that cannot be removed are shown in a comma-separated list together with a reason why the certificate removal failed.

Adding a CA Certs File to a Container

You can add a CA Certs file to any container that is in non-FIPS mode.

Caution: When you apply a CA Certs file, the entire trust store on the container is overwritten. All previously-added certificates are overwritten.

Before you follow the procedure below, make sure that the CA Certs file you want to add is loaded in the CA Certs repository.

To add a CA Certs file to a non-FIPS mode container:

1. Click **Node Management**.
2. In the navigation tree, click **System**.
3. Click the **Containers** tab.
4. On the **Containers** tab, Select one or more non-FIPS mode containers to which you wish to add a CA Certs file.
5. Click **Certificates**. The **Certificate Management** wizard starts.
6. Review the dialog box, then click **Next**.
7. Under **Choose an Action,** select **CA Cert (Legacy)**.
8. Follow the instructions in the wizard.

   After the CA Certs file has been added to a container, the container restarts automatically.

Enabling or Disabling a Demo Certificate on a Container

You can use the demo certificate on a container for testing purposes. By default, the demo certificate on a container is disabled. You can enable the demo certificate temporarily for testing purposes on a container that is non-FIPS mode.

Note: Enable a *demo* certificate on a container in non-FIPS mode for testing purposes only. Using a demo certificate in a production environment is a serious security issue because the demo certificate is not unique.

To enable or disable a demo certificate on a non-FIPS mode container:

1. Click **Node Management**.
2. In the navigation tree, click **System**.
3. Click the **Containers** tab.

4. On the **Containers** tab, Select one or more non-FIPS mode containers for which you wish to enable or disable a CA Certs file.
5. Click **Certificates**. The **Certificate Management** wizard starts.
6. Review the dialog box, then click **Next**.
7. Under **Choose an Action**, select **Demo CA (Legacy)**, then click **Next**.
8. Follow the instructions in the Certificate Management wizard.

   After you add the demo certificate on a container, the container restarts automatically.

Adding Multiple Destination Certificates to a Container

You can add multiple destination certificates to a container, whether in FIPS mode or not.

Note: Whenever you enable or disable FIPS mode on a container, check that the required certificates are present in the trust store and add them if necessary.

Click the  icon to display a list of the certificates available on the container.

Note: In the event that importing destination certificates for Transformation Hub fails due to changes in the certificate, please proceed to **remove** and then **add** the destination from the Connector as explained in "Removing Destinations" on page 972 and "Adding a Primary Destination to a Connector" on page 970.

To apply multiple destination certificates to a container:

1. Click **Node Management**.
2. In the navigation tree, click **System**.
3. Click the **Containers** tab.
4. On the **Containers** tab, containers for which you wish to add multiple destination certificates.
5. Click **Certificates**. The **Certificate Management** wizard starts.
6. Review the dialog box, then click **Next**.
7. Under **Choose an Action**, select **Import destination certificates** to add a certificate.
8. Follow the instructions in the wizard to complete the process.

Viewing Certificates on a Container

You can display a list of the CA certificates applied to a container and view the details for a particular certificate in the list. To view certificates on a container,

- On the **Containers** tab, in the **Action** drop-down for the container whose certificates you want to view, select **Display Certificates.**
- On the **Connectors** tab, click **Certificates** at the top of the page.

The Certificate List wizard displays the certificates applied to a container. To see details of a certificate, select the certificate, then click **Next** at the bottom of the page.

Resolving Invalid Certificate Errors

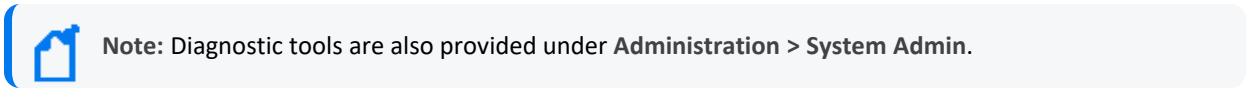
If no valid CA certificates exist for the connectors in the container, resolve the invalid certificate error as follows:

To resolve the invalid certificate error:

1. Select the container in the navigation tree.
2. Click the **Containers** tab. The error message is displayed.
3. In the **Action** drop-down of the container showing the issue, select **Download Certificates.**
4. Follow the instructions in the wizard to download and import the valid certificates.

Running Diagnostics on a Container

You can run diagnostics on a container.



To run diagnostics on a container:

1. Click **Node Management**.
2. In the navigation tree, navigate to the host on which the container resides.
3. Click the **Containers** tab.
4. On the **Containers** tab, select one or more containers for which to run diagnostics.
5. In the **Action** drop-down, click **Run Logfu**. The Diagnostics wizard starts.
6. Select the action you want to take on the selected container:
   - Select **Edit a configuration file** to edit a file in the `user/agent` folder on the container with the extension `.properties`, `.csv`, or `.conf`.
   - Select **Edit a user file** to edit any file (except binary files, such as `.zip`, `.jar`, or `.exe`) in the `user/agent` folder on the container.
7. From the list of available files, select the file you want to edit. The file displays in the Edit File panel. Make your edits, then click **Next** to save your edits and restart the container.

> **Note:** When you click **Next**, Arcsight Management Center saves the updated file in the `user/agent` folder on the container. The original file is overwritten.

8. Click **Done** to close the Diagnostics wizard.

Managing Connectors

A *connector* (also known as a *SmartConnector*) is an ArcSight software component that collects events and logs from various sources on your network. A connector can be configured on Arcsight Management Center, on a Logger platform with an integrated Connector Appliance, or installed on a computer on your network, managed remotely. For a complete list of supported connectors, go to the ArcSight Customer Support site.

> **Note:** The maximum number of selected entries when managing Connectors/Collectors is 50.

Procedures for managing connectors are described below.

Viewing All Connectors

You can see all currently managed connectors.

To view all connectors:

1. Click **Node Management**.
2. Click **System** in the navigation tree.
3. In the management panel, click the **Connectors** tab. All connectors display on the **Connectors** tab in the management panel.

Adding a Connector

Prerequisites

Before you add a connector, review the following important information.

- Make sure that the container, host, and location to which you want to add the connector exist in Arcsight Management Center. If any of these elements do not exist, create them.
- Follow the configuration best practices described in "Configuration Suggestions for Connector/Collector Types" on page 986.

  If you are configuring the Check Point OPSEC NG Connector, see "Configuring the Check Point OPSEC NG Connector" on page 987 and refer to the SmartConnector Configuration Guide for Check Point OPSEC NG.

If you are configuring a database connector that requires the MS SQL Server Driver for JDBC, follow instructions in "Adding the MS SQL Server JDBC Driver " on page 990.

> **Caution:** This connector type has special requirements concerning JDBC and authentication setup. Refer to the SmartConnector Configuration Guide for Microsoft SQL Server Multiple Instance Audit DB for this important information before installing the connector.

- If you are adding a software-based connector, make sure that the username and password for the connector match the username and password for the container to which you are adding the connector. If necessary, refer to "Changing Container Credentials" on page 952.

  > **Caution:** Each connector's default user name is `connector_user` and the default password is `change_me`. A connector with these default values still in place should be considered non-secure. ArcSight strongly recommends that for optimal security, you should change each connector's credentials to non-default values before deploying the connector to production.

- File-based connectors use the Common Internet File System (CIFS) or Network File System (NFS). These stipulations apply when creating a local connector to run as part of ArcMC.
  - On a Windows system, a CIFS share needs to be configured before you add a file-based connector.
  - For all other connectors, an NFS mount needs to be established before a file-based connector can be added. In addition, when entering the connector parameters, specify the configuration file name without an extension in the **Configuration File** field. The extension `.sdkrfilereader.properties` is appended automatically.
- For detailed information about individual connector parameters, refer to the specific ArcSight SmartConnector Configuration Guide for the type of connector chosen. The configuration guide also describes how to set up the source device for use with the connector

To add a connector:

> **Tip:** If you are adding a connector for the Check Point FW-1/VPN-1 system, see a more detailed procedure in "Configuring the Check Point OPSEC NG Connector" on page 987.

1. Click **Node Management**.
2. In the navigation tree, browse to the host on which the connector will reside.
3. In the management panel, click the **Containers** tab.
4. On the **Containers** tab, locate the container where you will assign the connector.
5. In the **Action** drop-down, click **Add Connector.** The Connector Setup wizard starts.
6. Review the dialog box, then click **Next**.

7. Select a connector type from the pull-down list of available types, then click **Next**.
8. Specify basic parameters for the connector. Parameters vary based on the connector type. (Hover over a field for more information on a field.) When all fields have been entered, click **Next**.

   Note: When entering parameters that include a file path, specify the path in POSIX format (for example, `/folder/filename`).

   For file-based connectors on Windows systems, specify the name of the CIFS mount point you created for the connector. (You need to specify `/opt/mnt/CIFS_share_name`.)

   Some connectors include table parameters. For example, the Microsoft Windows Event Log includes parameters for each host in the domain and one or more log types (security, application, system, directory service, DNS, file replication, and so on). You can import table parameters from a CSV file that was exported from another connector, as long as you export it and import it from the same containers. If the CSV file was exported from a different container, you need to change the secret parameters, such as the password, which appear in obfuscated format in the CSV file to plain text before you import the CSV file.

   Note: For connectors that query Microsoft Active Directory to detect devices (for example, Microsoft Windows Event Log - Unified), if the "Network Security: LDAP Server Signing Requirements" policy is set to "Signing Required" on the Domain Controller, Arcsight Management Center will be unable to connect to the Active Directory or browse for devices. You see an error when selecting **Windows Host Browser** as the connector device browser type.
9. Select a primary destination for the connector and specify destination-specific parameters on the following page(s), then click **Next**. Destinations can be:
   - ArcSight Logger SmartMessage (encrypted)
   - ArcSight Manager (encrypted)
   - CEF Syslog (plaintext, that is, unencrypted)

   Note: FIPS Suite B certificates are not retrieved automatically and must be uploaded manually.

   To see certificate details, hover over the certificate.
   - Select **Import the certificate to the connector from the destination**, then click **Next** to import the certificate and continue.
   - Select **Do not import the certificate to the connector from the destination, and then click Next** if you do not want to import the certificate. The destination will not be added.
10. Specify connector details:

Parameter	Description
Name	A descriptive name for this connector.
Location	The location of the connector (such as the hostname).
Device Location	The location of the device that sends events to the connector.
Comment	Additional comments.

11. When complete, click **Done**.

Editing Connector Parameters

ArcSight supports a large number of connector types to gather security events from a variety of sources, including syslog, log files, relational databases, and proprietary devices. Accordingly, configuration parameters vary widely depending on the type of connector being configured.

You can edit parameters (simple and table) for a specific connector, or for multiple connectors of the same type at the same time.

Note: The maximum number of selected entries when managing Connectors/Collectors is 50.

Updating Simple Parameters for a Connector

The following procedure describes how to update simple parameters for a specific connector.

To update parameters for a specific connector:

1. Click **Node Management**.
2. In the navigation tree, browse to the connector you wish to update.
3. In the management panel, the **Connector** summary tab displays.
4. On the **Connector** tab, next to **Connector Parameters,** click .
5. Modify parameters as necessary, then click **Next**.

   Note: When editing parameters that include a file path, specify the path in POSIX format (for example, `/folder/filename`).

6. When complete, click **Done**. The updated parameters display in the **Connector Parameters** table of the Connector summary tab.

Updating Table Parameters for a Connector

Certain connectors, such as the Microsoft Windows Event connector, have table parameters. You can update the table parameters for a specific connector when necessary.

To update table parameters for a specific connector:

1. Click **Node Management**.
2. In the navigation tree, browse to the connector you wish to update. In the management panel, the **Connector** summary tab displays.
3. On the **Connector** summary tab, next to **Table Parameters,** click  .
4. Modify parameters as necessary and then click **Next**.
   - To add more rows of parameter information, click the **Add Row** link.
   - You can use an Excel-compatible program to prepare a comma-separated values text file with the information and click the **Import File** button to load the entire table at once. The file needs to be in the same format as the rows shown on the Update Table Parameters page, and it needs to include a header row with parameter labels in the order shown on that page. For fields that require check box values, specify True or False as the value. An example is shown below.

	A	B	C	D	E	F
1	Domain Name	Host Name	User Name	Password	Security Logs	System Logs
2	test	1.1.1.1	admin	password	TRUE	FALSE
3	test2	1.1.1.1.1	admin	password	TRUE	FALSE

5. When complete, click **Done**. The updated table parameters display in the Table Parameters section of the Connector page.

Note: You can import a CSV file that was exported from another connector as long as you export and import the CSV file from the same container. If the CSV file was exported from a different container, you need to change the secret parameters, such as the password, which appear in obfuscated format in the CSV file to plain text before you import the CSV file.

Updating Simple and Table Parameters for Multiple Connectors

If you have multiple connectors of the same type, you can change the simple and table parameters for all the connectors at the same time.

To edit parameters for multiple connectors of the same type:

1. Click **Node Management**.
2. In the navigation tree, select the host where the connectors reside.
3. In the management panel, select the connectors whose parameters you want to update.
4. Click **Parameters**. The Update Connect Parameters wizard starts.
5. Review the dialog box, then click **Next**.

6. Follow the instructions in the wizard.
   - You can choose to modify the simple parameters for all the selected connectors at once or modify the simple parameters per connector.
   - If the connectors have table parameters, the table parameters are displayed so that you can modify them. If you have many table parameters to modify for multiple connectors, you can import the parameters from a CSV file. You can also export the table parameters to a CSV file for use as a backup or to import on another Connector Appliance.

     Note: When you update parameters for connectors of different versions, the newer connectors might have additional parameters. In this case, only those parameters shared by all connectors are displayed for updating.

7. Click **Done** when complete.

Managing Destinations

Connectors can forward events to more than one destination, such as ArcSight Manager and ArcSight Logger. You can assign one or more destinations per connector. You can assign multiple destinations to a connector and specify a failover (alternate) destination in the event that the primary destination fails.

The following procedures describe how to perform these actions on a specific connector or for multiple connectors at the same time:

- Add a primary or failover destination
- Edit destination parameters and destination runtime parameters
- Remove destinations
- Re-register destinations
- Manage alternate configurations for a destination
- Send a command to a destination

Note: Compared to the standalone Connector destination list, ArcMC Appliance on-board Connector does not cover the following three options: CEF file, CSV file and Raw Syslog

Note: In the event that the Transformation Hub certificate changes, the Connectors that had that Transformation Hub as a destination will be lost. To re-enable them follow the steps under ----new section---

Adding a Primary Destination to a Connector

When you add a primary destination to a connector, you need to specify details for the destination, such as the destination hostname and port used.

To add a primary destination to a connector:

1. Click **Node Management**.
2. In the navigation tree, browse to the connector to which you wish to add a destination. In the management panel, the **Connector** summary tab displays.
3. On the **Connector** summary tab, next to **Destinations**, click + . The Add Destination wizard starts.
4. Follow the steps in the wizard. You can either select an existing destination or add a new destination. If you are adding a new destination, select the destination type and specify parameters for the destination. Destination types are described in the SmartConnector User's Guide.

   Note: For containers running 5.1.2.5823 and later, Arcsight Management Center retrieves the certificate for the ArcSight Manager destination automatically and displays the certificate summary.

   For containers running 5.1.2 and earlier, upload the certificate on the container and then add the destination.

   FIPS Suite B certificates are not retrieved automatically and must be uploaded manually.

   To see certificate details, hover over the certificate.

   - Select **Import the certificate to the connector from the destination**, then click **Next** to import the certificate and continue.
   - Select **Do not import the certificate to the connector from the destination** and click **Next** if you do not want to import the certificate. The destination will not be added.
5. Click **Done** when complete.

Adding a Failover Destination to a Connector

Each destination can have a failover destination in case the connection with the primary destination fails.

Tip: UDP connections cannot detect transmission failure. Use Raw TCP for CEF Syslog destinations.

To add a failover destination to a connector:

1. Click **Node Management**.
2. In the navigation tree, browse to the connector to which you wish to add a destination. In the management panel, the **Connector** summary tab displays.
3. On the **Connector** summary tab, in the **Destinations** table, click  . The Add Destination wizard starts.
4. Follow the steps in the wizard to select from available destinations and specify the destination details.

   Note: FIPS Suite B certificates are not retrieved automatically and must be uploaded manually.

   To see certificate details, hover over the certificate.

   - Select **Import the certificate to the connector from the destination**, then click **Next** to import the certificate and continue.
   - Select **Do not import the certificate to the connector from the destination** and click **Next** if you do not want to import the certificate. The destination will not be added.
5. Click **Done** when complete.

Adding a Primary or Failover Destination to Multiple Connectors

You can add a primary or failover destination to several connectors at the same time.

To add a primary or failover destination to multiple connectors:

1. Click **Node Management**.
2. In the navigation tree, browse to the container where the connectors reside.
3. In the management panel, click the **Connectors** tab.
4. From the list of connectors, select all connectors to which you wish to assign a destination.
5. Click + **Destinations**.
6. Review the dialog, then click **Next**.
7. Under **Choose an Option**, select **Add a destination**, and then click **Next**.
8. Select between creating a new destination or selecting an existing destination, then click **Next**.
   - If you choose to **create a new destination**, select the destination type and then provide the destination parameters. Destination types are described in the SmartConnector User's Guide.

- If you choose to **select an existing destination**, select a destination from the list.

  Note: Arcsight Management Center retrieves the ArcSight Manager certificate for the destination automatically and displays the certificate summary.

  FIPS Suite B certificates are not retrieved automatically and must be uploaded manually.

  To see certificate details, hover over the certificate.

  - Select **Import the certificate to the connector from destination**, then click **Next** to import the certificate and continue.
  - Select **Do not import the certificate to the connector from the destination** and click **Next** if you do not want to import the certificate. The destination will not be added.

9. Define the destination function by choosing between a primary or failover destination.
   - If you choose **Primary destination**, click **Next** to update the configuration.
   - If you choose **Failover destination**:
     a. Select the primary destination that applies to your failover.
     b. Check the box in the table header to modify all of the displayed connectors.
     c. Click **Next** to update the configuration.
10. Click **Done** when complete.

Removing Destinations

You can remove a destination from a connector at any time. Each connector must have at least one destination; as a result, you may not remove all destinations from a connector.

To remove destinations from one or more connectors:

1. Click **Node Management**.
2. In the navigation tree, browse to the container where the connectors reside.
3. In the management panel, click the **Connectors** tab.
4. From the list of connectors, select all connectors to which you wish to remove a destination.
5. Click **Destinations**.
6. Review the content, and then click **Next**.
7. Under **Choose an Option**, select **Remove a destination, then click Next**.
8. Follow the instructions in the wizard, and click **Done** when complete.

Re-Registering Destinations

At certain times, you might need to re-register the destinations for one or more connectors; for example, after you upgrade ESM, or if a Logger appliance or ESM appliance becomes unresponsive.

To re-register destinations for one or more connectors:

1. Click **Node Management**.
2. In the navigation tree, browse to the container where the connectors reside.
3. In the management panel, click the **Connectors** tab.
4. From the list of connectors, select all connectors to which you wish to assign a destination.
5. Click **Destinations**.
6. Review the content, and then click **Next**.
7. Under **Choose an Option**, select **Re-register destinations,** and then click **Next**.
8. Follow the instructions in the wizard and click **Done** when complete.

Editing Destination Parameters

The following procedures describe how to edit destination parameters for a specific connector and how to edit destination parameters for multiple connectors.

Note: When enabling the demo CA for one or more connectors, use the Certificate button, instead of editing the ESM destination.

To edit destination parameters for a connector:

1. Click **Node Management**.
2. In the navigation tree, browse to the connector to which you wish to edit destination parameters. In the management panel, the **Connector** summary tab displays.
3. In the **Destinations** table, click  next to the destination you want to edit to display the **Edit Destination Parameters** page.
4. Make your changes, then click **Next**.
5. Click **Done** when complete.

To edit destination parameters for multiple connectors:

1. Click **Node Management**.
2. In the navigation tree, browse to the container where the connectors reside.
3. In the management panel, click the **Connectors** tab.
4. From the list of connectors, select all connectors for which you wish to edit destination parameters.
5. Click **Destinations.**
6. Review the content, and then click **Next**.
7. Under **Choose an Option**, select **Edit a destination**, then click **Next**.
8. Follow the instructions in the wizard and click **Done** when complete.

Editing Destination Runtime Parameters

The runtime parameters for a destination enable you to specify advanced processing options such as batching, time correction, and bandwidth control. The parameters you can configure are listed in "Destination Runtime Parameters " on page 1055. The user interface automatically displays the parameters valid for a destination.

The following procedures describe how to edit the runtime parameters for a specific connector and how to edit the runtime parameters for multiple connectors at the same time.

To edit destination runtime parameters for *a* connector:

1. Click **Node Management**.
2. In the navigation tree, browse to the connector for which you wish to edit destination runtime parameters. In the management panel, the **Connector** summary tab displays.
3. On the **Connector** summary tab, in the **Destinations** table, click next to the destination whose runtime parameters you want to edit.
4. Under **Add Alternate Configurations**, click  next to the alternate configuration that you want to edit.

   If you have not set up alternate configurations, click  next to the **Default**. For more information about alternate configurations, see "Managing Alternate Configurations " on the next page.
5. Specify or update values for the listed parameters, then click **Save**.

To edit destination runtime parameters for *multiple* connectors at the same time:

1. Click **Node Management**.
2. In the navigation tree, browse to the container where the connectors reside.
3. In the management panel, click the **Connectors** tab.
4. From the list of connectors, select all connectors for which you wish to edit destination runtime parameters.
5. Click **Runtime Parameters** to open the wizard.
6. Follow these steps in the wizard to edit the runtime parameters:
   a. Select the destinations whose runtime parameters you want to modify.
   b. Select the configurations to be affected (default or alternate configurations).
   c. Select the group of parameters you want to modify (for example, batching, cache, network, processing).
   d. Modify the parameters.

Managing Alternate Configurations

An *alternate configuration* is a set of runtime parameters that is used instead of the default configuration during a specified portion of every day. For example, you might want to specify different batching schemes (by severity or size) for different times of a day. You can define more than one alternate configuration per destination, and apply them to the destination for different time ranges during the day. For example, you can define a configuration for 8 a.m. to 5 p.m. time range and another configuration for the 5 p.m. to 8 a.m. time range.

By default, a configuration labeled **Default** is applied to a destination. Any subsequent configurations you define are labeled **Alternate#1**, **Alternate#2**, and so on. The default configuration is used if the time ranges specified for other alternate configurations do not span 24 hours. For example, if you specify an alternate configuration, **Alternate#1** that is effective from 7 a.m. to 8 p.m., the **Default** configuration is used from 8 p.m. to 7 a.m.

If you need to apply the same alternate configuration for multiple destinations, you need to define an alternate configuration (with the same settings) for each of those destinations.

Defining a New Alternate Configuration

The process of defining a new alternate configuration includes first defining the configuration, then editing it to specify the time range for which that configuration is effective.

To define an alternate configuration:

1. Click **Node Management**.
2. In the navigation tree, browse to the connector for which you wish to edit destination runtime parameters. In the management panel, the **Connector** summary tab displays.
3. On the **Connector** summary tab, in the **Destinations** table, click  .
4. Under **Add Alternate Configurations,** click **Add**.
5. Specify or update values for the listed parameters.
6. Click **Save**. If this is the first alternate configuration you defined, it is saved as Alternate#1. Subsequent configurations are saved as Alternate#2, Alternate#3, and so on.

   To specify the effective time range for which the configuration you just defined, edit the configuration you just defined using the following procedure, "Editing an Alternate Configuration" below.

Editing an Alternate Configuration

In addition to editing an alternate configuration to change parameter values, you can edit it to specify the time range for which it is effective.

To edit an alternate configuration:

1. Click **Node Management**.
2. In the navigation tree, browse to the connector for which you wish to edit destination runtime parameters. In the management panel, the **Connector** summary tab displays.
3. On the **Connector** summary tab, in the **Destinations** table, click  .
4. From the list of alternate configurations, select the alternate configuration that you want to edit, then click  .
5. Specify or update values for the listed parameters, including the time range in the From Hour/To Hour.
6. Scroll down to the end of the page and click **Save**.

Editing Alternate Configurations in Bulk

If you need to update the same parameters in multiple alternate configurations, follow the procedure described in "Editing Destination Runtime Parameters" on page 974.

Sending a Command to a Destination

You can send a command to a connector destination.

To send a command to a destination on a connector:

1. Click **Node Management**.
2. In the navigation tree, browse to the connector for which you wish to send a command. In the management panel, the **Connector** summary tab displays.
3. On the **Connector** summary tab, in the **Destinations** table, click  .
4. Select the command you want to run, then click **Next**.
5. Specify values for the parameters that the user interface displays, then click **Finish**.

Deleting a Connector

To delete one or more connectors:

1. Click **Node Management**.
2. In the navigation tree, browse to the container where the connectors reside.
3. In the management panel, click the **Connectors** tab.
4. From the list of connectors, select all the connectors you want to delete.
5. Click **Delete**.
6. Click **OK** to confirm deletion.
7. Reboot the Connector Appliance or Logger system that each connector was associated with.

Note: You can also delete a specific connector from its **Connector** summary tab. Click  at the top of the tab to delete the connector.

Sending a Command to a Connector

You can send a command to a connector.

To send a command to a connector:

1. Click **Node Management**.
2. In the navigation tree, browse to the connector to which you wish to send a command. In the management panel, the **Connector** summary tab displays.
3. On the **Connector** summary tab, click **Connector Command.**
4. From the **Command Type** drop-down list, select the command you want to send to the connector, then click **Next**.

Running Logfu on a Connector

Run Logfu on a connector to parse ArcSight logs and generate an interactive visual representation of the information contained within the logs.

To run Logfu on a connector:

1. Click **Node Management**.
2. In the navigation tree, browse to the connector to which you wish to run Logfu. In the management panel, the **Connector** summary tab displays.
3. On the **Connector** summary tab, click **Run Logfu.**
4. The Logfu progress window is displayed as system data logs are retrieved and analyzed. Data is then displayed by **Group**, **Field**, and **Chart**.
   - In the **Group** box, choose a data type to view. The **Group** box lists all connectors within the chosen container, plus many other data types, such as memory usage and transport rates.
   - Next, choose one of the **Group** box **data points**. Depending on which data point you chose, a list of fields appears in the **Field** box below.
   - Select a **field** to view. A graphic chart appears in the **Chart** box, providing rate and time information. The key at the bottom of the **Chart** box defines the data points mapped in the chart.
   - To choose a different data point for analysis, click **Reset Data**.
5. When complete, close the Logfu display window.

Changing the Network Interface Address for Events

Arcsight Management Center has multiple network interfaces. By default, the connector determines which network interface address is used for events displayed in the ArcSight Console or Logger, but typically uses `eth0`.

To use a specific network interface address for events, add the parameter `connector.network.interface.name` to the Connector's `agent.properties` file. For example, to use the IP address for `eth1`, specify the following parameter:

```
connector.network.interface.name=eth1
```

Developing FlexConnectors

FlexConnectors are custom, user-designed *SmartConnector*s that can read and parse information from third-party devices and map that information to ArcSight's event schema.

Arcsight Management Center provides a FlexConnector Development wizard that enables you to quickly and easily develop a FlexConnector by creating a parser file, and enables you to test and package your new FlexConnector before deploying it. The wizard generates regular expressions and provides event field mapping suggestions automatically so you do not need to be an expert in regular expression authoring, parser syntax, or ArcSight event schema.

Use the FlexConnector Development wizard to develop FlexConnectors for simple log files. For complex log files, use the FlexConnector SDK (available from the ArcSight Customer Support site)

The FlexConnector Development wizard supports Regex Files, Folder Follower, and Syslog (Daemon, File, Pipe) FlexConnectors only.

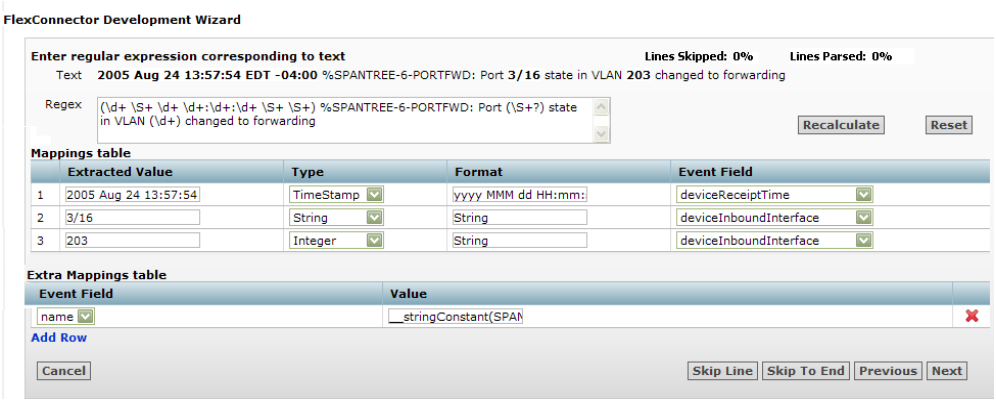
The FlexConnector Development wizard does not support the extra processors property or multiple sub-messages. If you need these features, use the FlexConnector SDK to create your FlexConnector.

Caution: A FlexConnector that you develop with the FlexConnector Development wizard might perform more slowly than an ArcSight *SmartConnector*.

To develop a FlexConnector:

1. Click **Node Management**.
2. In the navigation tree, browse to the container where you wish to develop the connector.
3. In the management panel, click the **Connectors** tab.
4. On the **Connectors** tab, in the Action drop-down, click **Edit FlexConnector**. The FlexConnector Development wizard is launched.
5. Provide the vendor and product name of the device for which you are creating a FlexConnector, then click **Next**.
6. Select the data source type, then click **Next**:
   - Select **Syslog** to create a Syslog FlexConnector to read events from Syslog messages.
   - Select **File** to create a FlexConnector to parse variable-format log files using regular expressions (ArcSight FlexConnector Regex File) or to parse variable-format log files in batch mode (ArcSight FlexConnector Folder Follower).

7. Upload a sample log file for the data source type you selected in the previous step, then click **Next**.
8. The wizard finds the first unparsed line in the log file, generates a regular expression to match and extract tokens from that line, and displays the suggested field mappings for each extracted token in the Mappings table.



Note: The mappings are displayed in descending order of probability (based on ArcSight training data). You can change the mappings by selecting from the list.

The percentage of parsed lines in the file is shown in the top right of the panel. You can use this percentage to estimate where you are in the log file. The percentage of unparsed lines skipped in the file is also shown in the top right of the panel.

- To change the regular expression in the **Regex** box and recalculate the mappings, edit the expression and then click the **Recalculate** button.You can set the regular expression back to the suggested value by clicking the **Reset** button.
- Field mappings that do not correspond directly to the extracted tokens in the unparsed line of the log file are displayed in the Extra Mappings table. You can change the Event Field and provide a token operation. To add a new Event Field, click **Add Row**.

  You can use extra mappings to:

  - Remap an extracted token to a different Event Field in addition to the existing mapping. For example, you can add an Event Field with the value `$3` where `$3` is the third token in the list of suggested mappings.
  - Map a modified token or combination of tokens to an Event Field. For example, you can add an Event Field with the value `__operation($1,$3)`.
  - Map an Event Field to a constant string or integer. For example, you can add an Event Field with the value `__stringConstant(constant)`.

    For a list of the token operations used when tokens are mapped to ArcSight event fields, refer to the FlexConnector Developer's Guide (available from the ArcSight Customer Support site).

9. Click **Next** to save the mapping to the parser file and display the next unparsed line in the log file.

   After all unparsed lines in the log file have corresponding regular expressions and mappings, the wizard displays the parser file for review.
10. Review the parser file and make changes, if necessary, directly in the Review Parser File panel.
11. Click **Next** to save and package the parser file.
12. Select how you want to deploy the FlexConnector:
    - Select **Deploy parser to existing connector in container**, then click **Next** to use the parser file with an existing connector. Click **Done** to close the FlexConnector wizard and re-display the **Container** tab.

      Note: The **Deploy parser to existing connector in container** option displays only if the container already contains a connector of the same type.
    - Select **Add new connector to container,** and then click **Next** to add the parser as a new connector. Follow the steps to add the connector to the container.

You can share FlexConnectors with other users. See "Sharing Connectors in ArcExchange" below.

Editing FlexConnectors

After you have developed a FlexConnector with the FlexConnector wizard and have deployed it in a container, you can edit the FlexConnector to make changes to the parser file when needed.

The FlexConnector Edit wizard is available on the **Connectors** tab in the **Action** drop-down.

Click **Edit Connector** in the **Action** drop-down for the FlexConnector to open the wizard, then edit the parser file.

Caution: Only edit a FlexConnector that is created with the FlexConnector wizard. Editing manually-created FlexConnectors might produce unpredictable results.

Sharing Connectors in ArcExchange

You can share FlexConnectors and parser overrides with other users.

A FlexConnector is a custom connector that you define to gather security events from log files, databases, and other software and devices. You can share the following FlexConnector types:

- Syslog FlexConnectors (to read events from syslog messages)
- Log File FlexConnectors (to read fixed-format log files)
- Regular Expression Log File FlexConnectors (to read variable-format log files)
- Regular Expression Folder Follower FlexConnectors (to read variable-format log files recursively in a folder)
- Regular Expression Multiple Folder Follower FlexConnectors (to read events in real time or batch mode from multiple folders)
- XML FlexConnectors (to read events recursively from XML-based files in a folder)

A parser override is a file provided by ArcSight used to resolve an issue with the parser for a specific connector, or to support a newer version of a supported device where the log file format changed slightly or new event types were added. You can share parser overrides for all connector types that use a parser.

To share a FlexConnector or parser override, you need to package and upload it to ArcExchange on the ArcSight online community (Protect 724) or to your local machine. You can also download a FlexConnector or parser override that you need from ArcExchange or from your local machine and add it to a container.

Note: ArcExchange will not be able to reach the ArcSight Protect724 Community if access is attempted through a proxy server.

Packaging and Uploading Connectors

Before uploading your FlexConnector or parser override to Protect 724 or to your local computer, you need to package it into a zip file (called an AUP package) using the upload wizard.

A FlexConnector AUP package contains the connector properties file, categorization file, connector parameters, and a manifest file with all the metadata on the package required for successful deployment. Metadata includes information about the AUP package, such as the package type, connector type, connector description, and so on. You can create only one AUP package per connector per device type. You can package a FlexConnector in Basic or Advanced mode. In **Basic** mode:

- The wizard packages the FlexConnector properties file automatically. If the wizard finds more than one properties file, you are prompted to select the file you want to package.
- The wizard packages the categorization file automatically *only* if it can be determined based on the device vendor and product information found in the properties file.
- The wizard does not package connector parameters. You are prompted to configure the connector when it is downloaded and deployed.

In **Advanced** mode:

- The wizard packages the FlexConnector properties file automatically. If the wizard finds more than one properties file, you are prompted to select the file you want to package. (Same as Basic mode.)
- The wizard packages the categorization file automatically if it can be determined based on the device vendor and product information found in the properties file. If the categorization file cannot be determined, you are prompted to select the categorization file you want to package from the list of files found in the container.
- The wizard displays connector parameters so you can configure the ones you want to display and set the default values you want to provide during connector deployment (download). The parameters you do not configure for display are pre-configured with the current values and will not be displayed during connector deployment.

A parser override package contains the parser override properties file and the manifest file only.

Follow the steps below to package and upload a FlexConnector or parser override.

- To upload to ArcExchange, you must have a valid username and password for Protect 724.
- Make sure that you have configured network settings under **Administration** > **System Admin** > **Network** and that Arcsight Management Center can communicate with the Protect 724 server.

To package and upload a FlexConnector or parser override:

1. Click **Node Management**.
2. In the navigation tree, browse to the connector for which you wish to upload a package. In the management panel, the **Connector** summary tab is displayed.
3. On the **Connector** details page, click  .The upload wizard is launched.
4. Click **Next** and follow the steps in the wizard to:
   a. Select the type of AUP package you want to create for the selected connector. Arcsight Management Center scans the container and displays the relevant files that can be packaged.
   b. For a FlexConnector, select **Basic** to create a default package or select **Advanced** to customize the package to meet your needs.
   c. If the connector contains several properties files, you are prompted to select the properties file you want to package. Certain connectors, such as syslog connectors, can have more than one parser override folder, in this case, you are prompted to select the folder you want to package.
   d. If you previously selected Advanced mode for a FlexConnector, and the categorization file cannot be determined, you are prompted to select the categorization file you want

to package from a list of files found in the container.

> **Note:** Categorization files are not packaged for parser overrides.

e. If you previously selected Advanced mode for a FlexConnector, select the configuration parameters you want to display when the connector is deployed and then provide default values for these parameters. Parameters you do not select are pre-configured with the current values.

   If any advanced connector parameters were previously modified from their defaults, the wizard displays these parameters so that you can select which ones you want to be configured automatically during deployment.

   > **Note:** Configuration parameters are not displayed for parser overrides. If the connector has table parameters, they are not displayed during packaging. However, when the connector is downloaded to a container, you are prompted to provide values for all the table parameters.

f. Provide a description of the AUP package and instructions on how to configure the device used by the connector.

g. Provide the vendor, product, and version of the device used by the connector.

   If the wizard can determine the vendor, product, and version of the device, the information is displayed in the fields provided. You can change the information to meet your needs.

h. Upload the created AUP package to ArcExchange or to your local machine. You will require a username and password for the Micro Focus Community.

Downloading Connectors

You can download a FlexConnector or parser override that is available from ArcExchange on the Micro Focus Community or from your local computer. You download a FlexConnector or parser override directly to a container.

You can download only one FlexConnector per container using the download wizard. However, there is no limit to the number of parser overrides you can download to a container.

- When downloading a parser override to a container, the download wizard overwrites any existing parser override with the same name in the container without prompting for confirmation. To avoid overwriting an existing parser override, send a **Get Status** command to the existing parser override to check the parser information before you download a new one. For information on sending a Get Status command, refer to "Sending a Command to a Connector" on page 977.
- Always back up the container to the Backup Files repository before downloading a connector or parser override so you can revert to the previous configuration if the download produces unexpected results.

Follow the steps below to download a FlexConnector or parser override to a container.

To download to ArcExchange, you must have a valid username and password for Protect 724. Also, make sure that you have configured network settings under **Administration** > **System Admin** > **Network** and that the appliance can communicate with the Protect 724 server.

To download a FlexConnector or parser override:

1. Click **Node Management**.
2. In the navigation tree, browse to the host on which the container resides.
3. In the management panel, click the **Containers** tab.
4. From the list of containers, locate the container into which you want to download the connector. In the **Action** drop-down, select **Run FlexConnector Wizard.**
5. Click **Next** and follow the steps in the wizard to:
   a. Select whether you want to download the connector from ArcExchange on Protect 724 or from your local computer.
   b. Select the AUP package you want to download.

      On the Micro Focus Community, you can search for a parser override or FlexConnector AUP package using a keyword or a combination of keywords.

      Note: You can only download a parser override package to a container that has a connector of the same type as the package. You can download only one FlexConnector per container using the download wizard. If the container already contains a FlexConnector of the same type as the one you want to download, you can replace the existing FlexConnector with the one you are downloading, but you cannot create a new one.

   c. For a FlexConnector, provide connector configuration parameters, if needed.

      Pre-configured and advanced parameters are deployed automatically with the values that were packaged; you are not prompted to configure these parameters. The configurable parameters are displayed with suggested defaults, which you can modify if

necessary. The table parameters are displayed with no configured values, you have to provide the values manually, as needed.

d. Add or select a destination for the connector.

   If you are downloading the connector to a container that has an existing connector of the same type, you are *not* prompted for a destination.

The wizard copies the properties and categorization files to the appropriate locations and also installs the zip file for the AUP package in the `user/agent/deployedaups` folder on Arcsight Management Center to keep track of the deployment history.

After a successful download, the container is restarted automatically.

Configuration Suggestions for Connector/Collector Types

The following table provides configuration suggestions for different types of connectors or Collectors.

Connector/Collector Type	Effects of Limited Usage
Syslog	Due to the nature of UDP (the transport protocol typically used by Syslog), these Connectors/Collectors can potentially lose events if the configurable event rate is exceeded. This is because the connector delays processing to match the event rate configured, and while in this state, the UDP cache might fill and the operating system drops UDP messages. **Note**: Do not use the **Limit CPU Usage** option with these connectors because of the possibility of event loss.
SNMP	Similar to Syslog connectors, when the event rate is limited on SNMP connectors, they can potentially lose events. SNMP is also typically UDP-based and has the same issues as Syslog.
Database	Because connectors follow the database tables, limiting the event rate for database connectors can slow the operation of other connectors. The result can be an event backlog sufficient to delay the reporting of alerts by as much as minutes or hours. However, no events will be lost, unless the database tables are truncated. After the event burst is over, the connector might eventually catch up with the database if the event rate does not exceed the configured limit.

Connector/Collector Type	Effects of Limited Usage
File	Similar to database connectors, file-based connectors *follow* files and limiting their event rates causes an event backlog. This can eventually force the connector to fall behind by as much as minutes or hours, depending on the actual event rate. The connectors might catch up if the event rate does not exceed the configured rate.
Asset Scanner	All connectors on ArcSight Platform run as a service (not as an application). Therefore, asset scanner connectors running onConnector Appliance are *not* supported in Interactive mode. To run the asset scanner connector in Interactive mode, install the connector on a standalone system and manage it as a software-based connector.
Proprietary API	The behavior of these connectors depends on the particular API, (for example, OPSEC behaves differently than PostOffice and RDEP). But in most cases, there will be no event loss unless the internal buffers and queues of the API implementation fill up. These connectors work much like database or file connectors.

Included FlexConnectors

ArcSight Arcsight Management Center Connector Applianceincludes these prototype FlexConnectors:

- ArcSight FlexConnector File
- ArcSight FlexConnector ID-based Database
- ArcSight FlexConnector Multiple Database
- ArcSight FlexConnector Regular Expression File
- ArcSight FlexConnector Regular Expression Folder File
- ArcSight FlexConnector Simple Network Management Protocol (SNMP)
- ArcSight FlexConnector Time-based Database
- ArcSight FlexConnector XML File

You can use these prototypes to develop your own FlexConnectors, and these can be shared with other users. Refer to "Sharing Connectors in ArcExchange" on page 981.

For more information, consult the FlexConnector Developer’s Guide, available from ArcSight Customer Support.

Configuring the Check Point OPSEC NG Connector

The Check Point FW-1/VPN-1 OPSEC NG connector can operate in clear channel or sslca mode.

Note: The following stipulations apply to configuring the Check Point OPSEC NG Connector:

- This procedure is supported only for ArcSight connector release 4.6.2 or later.
- A hostname is called an Application Object Name on Check Point. A password is a Communication Activation Key on Check Point.

To configure a connector to operate in sslca mode:

On the Check Point SmartDashboard:

1. Create an OPSEC Application Object using the Check Point SmartDashboard. You need to provide these parameters when creating the application object.

Parameter	Description
Name	A meaningful name for the application object you are creating; for example, ArcSightLea-1. This name is used to pull the OPSEC certificate.
Host	The hostname of the ArcSight Management Center system managing the connector.
Client Entities	Select LEA.
Secure Internal Communication	If a DN string is not present, initialize the communication by providing an activation key. The activation key is used when the certificate is pulled. This is the SIC Name. Click **Communication** > **Initialize**.

   After the object is created, note down the following information, which you will need to provide when continuing configuration.

   - *SIC Name:* DN string that you obtain after initializing communication as described below.
   - *SIC Entity Name:* Double-click the Check Point Gateway name in the SmartDashboard to view its general properties. The SIC Entity Name is the SIC string configured in the general properties window.
   - Check Point IP address or hostname.

2. Pull the Check Point certificate.

   To do so, run the `Pull OPSEC Certificate` command on the container to which you are adding the connector. For detailed information about running a command on a container, see "Sending a Command to a Container" on page 953. You need to provide this information when running the command:

Parameter	Description
Server hostname or IP address	The name or IP address of the Check Point server.
Application object name	The OPSEC Application object name you specified in the previous step. This parameter is case sensitive.
Password	The activation key you entered when creating the OPSEC application object in the previous step.

If the certificate is pulled successfully, a message similar to this is displayed:

```
OPSEC SIC name (CN=ArcSightLea-1,0=cpfw1.5ad8cn) was retrieved and stored
in /opt/arcsight/connectors/<container
name>/current/user/agent/checkpoint/<name>. Certificate was created
successfully and written to "/opt/arcsight/connectors/<container
name>/current/user/agent/checkpoint/ArcSightLea-1.opsec.p12".
```

Note down the OPSEC SIC Name (CN=ArcSightLea-1,0=cpfw1.5ad8cn in the above example) and the file name (ArcSightLea-1.opsec.p12 in the above example).

Tip: If the certificate is not pulled successfully, check to ensure that the Application object name you specified is correct (including the case) and the container on which you are running the command is up and running.

3. Install Policy on the LEA client for the Check Point Gateway using the SmartDashboard.

On Connector Appliance:

1. Add a Check Point connector by following instructions described in "Adding a Connector" on page 964. You need to provide the following information.

Parameters	Values to input
Type	Check Point FW-1/VPN-1 OPSEC NG
Connection Type	SSLCA

Parameters	Values to input
Connector Table Parameters	Server IP: The IP address of the Check Point server. Server Port: The port on the server that listens for SSLCA connections. Use the default value 18184. OPSEC SIC Name: The name you noted in "Create an OPSEC Application Object using the Check Point SmartDashboard. You need to provide these parameters when creating the application object. " on page 988. OPSEC SSLCA File: The name you noted after pulling the certificate in "Pull the Check Point certificate." on page 988. OPSEC Entity SIC Name: The name you noted in "Create an OPSEC Application Object using the Check Point SmartDashboard. You need to provide these parameters when creating the application object. " on page 988.

2. An error similar to the following is displayed.

   ```
   -1:[X] Unable to connect to the Lea Server[10.0.101.185] -1:1 connection test failed!
   ```

   Select the **Ignore warnings** check box, then click **Next**.
3. Continue to configure the rest of the connector as described under "Adding a Connector" on page 964.

Adding the MS SQL Server JDBC Driver

When you install and configure database connectors that use Microsoft SQL Server as the database, a JDBC driver is required. This driver does not ship pre-installed; you need to install it before configuring database connectors on the appliance.

To install a JDBC Driver:

1. From the Microsoft web site, download the MS SQL Server JDBC Driver to a computer that can access Arcsight Management Center.
2. Run the setup program to install the driver.
3. Follow the instructions in "Uploading Files to a Repository" on page 655 to add the `sqljdbc.jar` file.

   Tip: The name of the `jar` file may be different from that of some JDBC driver versions. Different versions of the JDBC driver are required for different SQL Server database versions; be sure to use the correct driver for your database.

   The new driver file is added to the repository, as shown in the following example.

   After you have installed the JDBC driver, you need to upload the driver file to the containers that will hold the SQL Server database Connectors. Follow the instructions in "Uploading Files to a Repository" on page 655.

After the driver file has been uploaded to a container, follow the instructions in "Adding a Connector" on page 964 to add a connector that requires a JDBC driver.

Adding the MySQL JDBC Driver

When you install and configure database connectors that use MySQL as the database, a JDBC driver is required. This driver does not ship pre-installed. Install it before configuring database connectors on the appliance.

To install a JDBC Driver:

1. From the MySQL web site, download the MySQL JDBC Driver to a computer that can access Arcsight Management Center.

   http://dev.mysql.com/downloads/connector/j/5.0.html
2. Extract the driver.
3. Follow the instructions in "Uploading Files to a Repository" on page 655 to add the `mysql-connector-java-x.x.x-bin.jar` file. The new driver file is added to the repository.

   After you have installed the JDBC driver, you need to upload the driver file to the containers that will hold the MySQL database Connectors. Follow the instructions in "Uploading Files to a Repository" on page 655.

   After the driver file has been uploaded to a container, follow the instructions in "Adding a Connector" on page 964 to add a connector that requires a JDBC driver.

   Note: For both **Adding the MS SQL Server JDBC Driver** and **Adding the MySQL JDBC Driver** cases, make sure to use the default path `/lib` when uploading the JDBC driver for on-board connector installation. For Instant Connector Deployment installation use `user/agent/lib`.

Managing Configurations

A *configuration* is a group of related appliance or software settings and their associated values, which applies to one or more node types. A configuration created for a node can be pushed to nodes of the same type managed by Arcsight Management Center, assuring uniformity across a group of nodes.

Configurations come in these kinds:

- A *subscriber* configuration is for the routine management of multiple managed ArcSight products. You can easily assign values to, propagate, and maintain the same settings across multiple nodes of the same type, including connectors, Collectors, Connector Appliances, Loggers, or other ArcMCs.

- A *initial* configuration is for the rapid, uniform setup of multiple ArcSight Loggers (only). Use an initial configuration to expedite the initial deployment of ArcSight Loggers to a production environment.

Configuration management tasks include:

- *Configuration Creation:* A configuration for a node type can be created (as well as edited or deleted) in Arcsight Management Center.
- *Configuration Import:* A configuration can be created directly on a managed node, exported, then imported into Arcsight Management Center for sharing with nodes of the same type.
- *Configuration Push:* A configuration can be *pushed* from ArcMC to managed nodes. This copies the configuration from ArcMC and changes the settings on each destination node.
- *Subscriptions:* Managed nodes can be *subscribed* to a subscriber configuration, so they can receive a new or updated configuration pushed from Arcsight Management Center.
- *Compliance Checks:* Check whether the settings and their values on a managed node match the ones for a configuration type specified in Arcsight Management Center. If so, the node is said to be in *compliance* with the configuration.
- *Comparisons:* Compare two configurations of the same type quickly, with a field by field breakdown of each setting, its value, and any differences. You can compare the values of a configuration on a subscriber node to the values of the baseline or reference configuration on an ArcMC which manages it. You can also compare two configurations of the same type on a single ArcMC.

For example, a typical workflow for a subscriber configuration might work as follows: you can create a suitable DNS configuration for an appliance, specifying primary DNS server, secondary DNS server, and search domains for the appliance. (See "Destination Configuration Types" on page 1011.) You can then push your DNS configuration to subscribing appliances, and so ensure that DNS settings for all subscribed nodes are configured identically with a single action.

If you later updated the configuration to use a new primary DNS server, you could push the new configuration to all subscribers, and all of them would be updated for the new DNS server with one action.

At any time, you could verify any managed node’s compliance with the configuration to determine if its settings were assigned the desired values.

The following topics are discussed here.

Configuration Management

To create or manage configurations, on the menu bar, click **Configuration Management.** To manage a specific configuration type, select the configuration type from the sub-menu.

For example, to access subscriber configurations for Loggers, click **Configuration Management > Subscriber Configurations > Logger Configurations.**

The Configurations Table

The **Configurations** table lists all currently available subscriber configurations in ArcSight Management Center. Each listed configuration, whether it was created in ArcSight Management Center or imported from an existing node, is considered the baseline copy of that configuration, for pushing to managed nodes. The table includes the following columns.

- **Name:** The name of the configuration.
- **Type:** The type of configuration.
- **Last Edited By:** The most recent user to edit the configuration.
- **Compliance:** An aggregation of the status of the individual subscribers to that configuration.
  - *Compliant* indicates that all subscribers are in compliance.
  - *Non-Compliant* indicates that at least one subscriber is out of compliance.
  - *Unknown* indicates that the compliance status for one or more subscribers cannot be determined (for example, because connectivity to one or more subscribers is not available).

    Tip: You can check the individual compliance of each subscriber on the **Subscribers** tab.

Click any column header to sort the **Configurations** table by that column.

To view the details of any configuration, click its name in the list. The **Details** and **Subscribers** tabs will display additional information.

Tip: To select multiple items from any list, Shift+Click or Ctrl+Click while selecting.

The Details Tab

The **Details** tab shows the specifics of the configuration, including any configured attributes and their values.

Configuration Name

Each configuration has a unique name. A configuration may be up to 255 characters in length.

General

General details describe the basics of the configuration, as follows:

- **Configuration Type:** The type of the configuration. For details of configuration types, see " Subscriber Configuration Types" on page 1005.
- **Last Edited By:** The most recent user to edit the configuration.

Properties

A *property* is a group of one or more settings for the configuration. For example, for the NTP Server configuration, the property includes two settings: Enable as NTP Server (a Boolean value indicating whether to enable the product as an NTP server), and NTP Servers (a list of NTP servers).

The specific parameters included in each property are pre-defined for each configuration type. ArcSight Management Center prompts for values of each setting when the property is selected. Each parameter must be assigned a valid value corresponding to its data type. For instance, if the data type is integer, you must specify an integer value. A red asterisk (*) indicates a required parameter.

List Configurations

A configuration type that can include more than one property is known as a *list configuration*. A list configuration represents a configuration with multiple instances of data values of the same kind. Each instance is known as a *property*.

For example, the Connector Map File configuration could include information on multiple map files. Each Property would represent a different map file (with different values for file path and content).

Note: A pushed list configuration will override any existing configuration of the same type on the managed node. To *append* data to an existing configuration, use the bulk management tools (**Set Configuration**)

For a description of supported configuration types, the parameters associated with each type, and their data types, see "The Configurations Table" on the previous page.

The Subscribers Tab

The **Subscribers** list shows all managed nodes currently eligible to receive the configuration. (The list is empty if no hosts have been added yet.)

The tab includes these operations buttons:

Add Subscribers	Adds subscribers to the existing configuration.
Push	Pushes the configuration to one or more selected subscribers.

Check Compliance	Checks the compliance of all subscribers with the baseline configuration.
Unsubscribe	Removes one or more selected subscribers from the subscriber list.

The list includes the following columns:

- **Path:** The path of the subscribing node, consisting of location/hostname/node type.
- **Type:** The type of subscribing node.
- **Last Pushed At:** The time and date of the most recent push to the subscriber.
- **Last Push Status:** The status of the most recent push to the subscriber.
  - *Succeeded*: The configuration push was successful.
  - *Failed*: Hover over the link to determine the reason for the push failure. An error message is displayed to help in remediation of the issue. For more information, see "Push Remediation" on page 1002.
  - *Unknown:* Initial status before the subscriber has received any pushes.
- **Last Compliance Check:** The date and time of the most recent compliance check.
- **Compliance:** Whether the node is in compliance with the configuration.
  - *Compliant* indicates the node is in compliance. The values for *all* settings associated with the configuration type match the values from the configuration.
  - *Non-Compliant* indicates the node is out of compliance. One or more values for the settings associated with the configuration type do not match the values from the configuration. Hover over *No* to show the cause of the node's non-compliance.
  - *Unknown* indicates either that the node's compliance could not be determined at the time of the most recent compliance check, or that the node has not yet undergone a compliance check.

Non-Compliance Reports

You can determine why a compliance status is Non-Compliant.

For a compliance status of *Non-Compliant,* click the status to display the **Configuration Comparison** dialog, which compares all setting values for the configuration on ArcMC and on the managed node.

Click **Push Configuration** to push the configuration to the managed node in order to make it Compliant.

Creating a Subscriber Configuration

You can create a subscriber configuration for pushing to any subscribed nodes.

Note: The following subscriber configuration types cannot be created in ArcMC, but can only be imported from managed nodes:

- Logger Storage Group
- Logger Filter
- Logger ESM Forwarder, Connector Forwarder, TCP Forwarder, UDP Forwarder
- Authentication External

For more information on importing a configuration from a managed node, see "Importing a Subscriber Configuration" on the next page.

To create a configuration:

1. Click **Configuration Management > Subscriber Configurations > All Configurations.**

   Tip: To filter for a specific subscriber configuration type, select the desired configuration type from the **Subscriber Configurations** sub-menu.

2. Under **Configurations**, click **New**.
3. On the **Details** tab, select a configuration type from the **Configuration Type** drop-down list. (Only the appropriate configuration types are shown in the drop-down list.)
4. In **Configuration Name,** specify a name for the configuration. (Configuration names must be unique and may be up to 255 characters in length.)
5. Specify values for any required parameters, which are indicated with a red asterisk (*).

   Note: For a description of valid parameters for each configuration type, and the data type associated with each, see " Subscriber Configuration Types" on page 1005.

6. Optionally, add values for any optional parameters.
7. Optionally, to add an additional property for a list configuration: click **Add Property**, then specify values for the prompted parameters. Repeat adding properties as needed to completely define the configuration.
8. Click **Save**.

Editing a Subscriber Configuration

You can modify or delete values for a subscriber configuration. (You may not edit a configuration currently being pushed.)

To edit a configuration:

1. Click **Configuration Management > Subscriber Configurations > All Configurations.**

   > **Tip:** To filter for a specific subscriber configuration type, select the desired configuration type from the **Subscriber Configurations** sub-menu.

2. From the **Configurations** table, click the name of the configuration to be edited.
3. On the **Details** tab, click **Edit**.
   - Edit the general settings as needed.
   - Optionally, to add an additional property for a list property, click **Add Property**, then specify values for the prompted parameters. Repeat adding properties as needed to completely define the configuration.
   - Optionally, to delete a property from the configuration, click **Delete Property.**
4. When complete, click **Save**. After saving, if the configuration has any subscribers, you are prompted to push the updated configuration to the subscribers.

Deleting a Subscriber Configuration

A deleted subscriber configuration is no longer available for pushes to subscribers. You may not delete a configuration currently being pushed.

To delete a subscriber configuration:

1. Click **Configuration Management > Subscriber Configurations > All Configurations.**

   > **Tip:** To filter for a specific subscriber configuration type, select the desired configuration type from the **Subscriber Configurations** sub-menu.

2. From the **Configurations** table, select one or more configurations to be deleted.
3. Click **Delete.**
4. Click **Yes** to confirm deletion.

Importing a Subscriber Configuration

A subscriber configuration created on a managed node may be imported into ArcMC, for editing and pushing to other nodes of the same type.

For example, you can define a configuration on a managed Connector Appliance, then import the configuration into ArcMC. The imported configuration may then be edited and pushed to

other managed Connector Appliances, just the same as you would with a configuration you originally created in ArcMC.

Note: If configuration import to the localhost fails, restart the web service on the localhost.

To import a subscriber configuration from a managed node:

1. Click **Configuration Management > Subscriber Configurations > All Configurations.**

   Tip: To filter for a specific subscriber configuration type, select the desired configuration type from the **Subscriber Configurations** sub-menu.

2. Under **Configurations**, click **Import**.
3. On the **Choose a Node** dialog, select the node from which you wish to import the configuration.
4. Click **Continue**.
5. On the **Import Configuration** dialog:
   a. Select a configuration type for the imported configuration from the **Type** drop-down list. (The entries in the list depend on the configuration types which apply to the node chosen in Step 3.)
   b. In **Name**, specify a name for the imported configuration.
6. Click **Import**. The configuration is imported into ArcMC and is shown in the **Configurations** table.

   Note: In order to import a backup configuration from a Connector Appliance, Logger, or ArcMC node, the node must have a scheduled backup to begin with.

Managing Subscribers

A *subscriber* is a managed node to which a configuration may be pushed. A subscriber to which a configuration is pushed will receive and process the pushed configuration and apply it to the managed node, so that the managed node's settings are the same as the settings specified in the configuration.

Each node can subscribe to *only one* configuration of each configuration type.

For example, a Logger appliance could subscribe to one Logger Storage Group configuration, but the same appliance could also subscribe to a Logger Filter configuration as well as a Logger Transport Receiver configuration.

Viewing Subscribers

To view subscribers for a configuration:

1. Click **Configuration Management > All Configurations.**
2. From the list of configurations, locate the configuration for which you wish to view subscribers.
3. Click the name of the configuration.
4. Click the **Subscribers** tab. The current subscribers are displayed.

Adding a Subscriber

A subscriber (that is, a subscribed node) can receive a pushed configuration.

To subscribe a node to a configuration:

1. Click **Configuration Management > All Subscriber Configurations.**

   Tip: To filter for a specific subscriber configuration type, select the desired configuration type from the **Subscriber Configurations** sub-menu.

2. From the **Configurations** table, click the name of the configuration to which you wish to add subscribers.
3. Click the **Subscribers** tab.
4. Click **Add Subscribers.**
5. On the **Add Subscribers** dialog, select a node to add as a subscriber. The list of potential subscribers is determined by the selected configuration type. To select multiple nodes for subscription, Ctrl+Click each node.

   Note: A node may only subscribe to one configuration of each type; for example, one DNS configuration.

   If you attempt to add a subscriber which is already subscribed to a configuration of the same type, the following message is displayed: *No available subscribers have been found for the selected configuration.*

6. Click **Add Subscribers.**
7. Click **OK** to confirm completion. The subscriber is added to the recipients for the configuration.

Unsubscribing a Subscriber

After being unsubscribed, a node can no longer receive a pushed configuration.

To remove a subscriber from a configuration:

1. Click **Configuration Management > All Subscriber Configurations.**

   Tip: To filter for a specific subscriber configuration type, select the desired configuration type from the **Subscriber Configurations** sub-menu.

2. From the **Configurations** table, click the name of the configuration from which you wish to remove subscribers.
3. Click the **Subscribers** tab.
4. Select one or more subscriber from the list of subscribers.
5. Click **Unsubscribe**.
6. Click **OK** to confirm. The selected subscribers are unsubscribed.

Pushing a Subscriber Configuration

A pushed subscriber configuration synchronizes the configuration from ArcMC to all or a selection of the configuration's subscribers. Pushing must be performed manually.

When selecting subscribers, only valid potential subscribers for the configuration are shown. For example, if pushing a Logger configuration, which only applies to Loggers, only managed Loggers would be shown as potential subscribers, not Connector Appliances or ArcMCs.

Note: If a configuration push to the localhost fails, restart the web service on the localhost.

To push a subscriber configuration to all subscribers:

1. Select **Configuration Management > All Subscriber Configurations.**

   Tip: To filter for a specific subscriber configuration type, select the desired configuration type from the **Subscriber Configurations** sub-menu.

2. From the **Configurations** table, select a configuration to be pushed.
3. Click **Push**.
4. Click **Yes** to confirm the push. The configuration is pushed to all subscribers of the selected configuration. A compliance check is automatically performed on each subscriber.

To push a subscriber configuration to selected subscribers:

1. Select **Configuration Management > Subscriber Configurations > All Configurations.**

   Tip: To filter for a specific subscriber configuration type, select the desired configuration type from the **Subscriber Configurations** sub-menu.

2. From the **Configurations** table, select a configuration to be pushed, and click the name of the configuration.
3. On the **Configuration Details and Subscribers** page, click the **Subscribers** tab.
4. On the **Subscribers** tab, select one or more subscribers to which to push the configuration.
5. Click **Push**.
6. Click **Yes** to confirm the push. The configuration is pushed to the selected subscribers. A compliance check is automatically performed on each recipient.

Push Validation

During a push to subscribers, the configuration is automatically validated by ArcSight Management Center. Validation ensures that a pushed configuration contains appropriate, meaningful values for all settings. If any configuration values are found to be invalid, the push will fail, and an error message will be returned. Hover over the subscriber's entry on the **Subscribers** tab, in the **Push Status** column, to show the cause of the failed push. In addition, a compliance check is automatically performed after the push.

Common Causes for Push Failure

A push to a subscriber may fail for any number of reasons. These may include:

- **Validation Failure:** A push with invalid content will fail. Verify that your configuration includes valid setting values for the configuration type.
- **Lack of Connectivity:** Network or system issues can cause disrupt connectivity to a subscriber. Verify connectivity with the subscriber.
- **Agent Not Running on Host :** Verify that the ArcMC Agent process is active on the subscribing node. (This does not apply to connectors or Collectors, which do not require the Agent.)
- **Privileges on Subscribing Host**: In order to push a subscription, the ArcSight Platform user (specified by the user credentials) must have privileges to view, edit, or delete configuration settings on the subscriber nodes.
- **Expired License:** An expired host license will cause a push to the host to fail.

Push Remediation

If a push to a subscriber fails, you may be able to remedy the failure by following these steps:

1. Select the configuration from the **Configurations** table.
2. Click the **Subscribers** tab and choose the subscriber to which the push failed.
3. The **Last Push Status** will show *Failed*. Hover over this link to view the error message associated with the push failure.

After viewing the error message, you can take the appropriate steps on the managed node to address the issue. Resolution may require direct or remote access to the node outside of ArcSight Management Center.

After the issue is resolved, you can retry the failed configuration push.

Checking Subscriber Compliance

A subscribed node is in *compliance* with a configuration if the settings for the node match those assigned to the configuration in ArcSight Management Center.

The configuration listed in the managing ArcSight Management Center is considered the baseline copy of the configuration.

For example, you create an SMTP configuration in ArcSight Management Center named *Sample SMTP Configuration,* with these values assigned:

- Primary SMTP Server: *Mailserver1*
- Secondary SMTP Server: *Mailserver2*
- Outgoing Email Address: *admin@example.com*

A node would be in compliance with this configuration if the values for its primary and secondary SMTP servers, and outgoing email address, matched the values in *Sample SMTP Configuration*.

If any one of these values were different (for example, if a node had a primary SMTP Server of *CorporateMail1*) the node would be out of compliance.

You can manually check the compliance of all subscribers to a configuration.

To manually check subscriber compliance for a configuration:

1. Click **Configuration Management > Subscriber Configurations > All Configurations.**

   Tip: To filter for a specific subscriber configuration type, select the desired configuration type from the **Subscriber Configurations** sub-menu.

2. In the **Configurations** table, select the configuration to be checked for compliance.
3. Click **Check Compliance**. All subscribers to the selected configuration are checked for compliance.
   - On the **Configurations** table, the **Compliance** column shows the aggregated compliance of all subscribers.
   - On the **Subscribers** tab for the configuration:
     - The **Last Compliance Check** column is updated to show the most recent check.

       Automatic compliance checks will run every 12 hours. So this will be the date and time of the latest automatic check.

     - The **Compliance** column indicates the individual compliance of each node.

Comparing Configurations

You can compare two configurations of the same type to verify whether they contain the same settings. The following two comparisons are possible:

- **Comparing two configurations on a single ArcMC.** You can compare two configurations of the same type on a single ArcMC. For example, you could compare the settings for two different SMTP configurations.
- **Comparing the configuration on a subscriber to the same configuration on its managing ArcMC.** You can quickly check to see how the settings for a configuration on a subscribing node differs from the same configuration on its managing ArcMC.

To compare two configurations of the same type on one ArcMC:

1. Click **Configuration Management.**
2. Select **All Configurations.**
3. In the list of configurations, select two configurations.
4. Click **Compare**.

The **Configuration Comparison** dialog shows each setting for the configuration and the current value for each compared item in the **Status** column.

To print the comparison as a PDF report, click **Export to PDF.**

To compare the configuration on a subscriber to the same configuration on its managing ArcMC:

1. Click **Configuration Management.**
2. Select **All Configurations.**
3. In the configurations list, select the configuration you wish to compare between ArcMC and the subscriber.
4. Under **Configuration Details & Subscribers**, click the **Subscribers** tab.
5. In the **Compliance** column, click the status link.

The **Configuration Comparison** dialog shows each setting for the configuration and the current value for each compared item.

Optionally, if the subscriber is Non-compliant with the configuration on its managing ArcMC, click **Push Configuration** to push the configuration to the subscriber (which will make it compliant).

To export the comparison as a PDF report, click **Export to PDF.**

Configuration Management Best Practices

Configuration management is a powerful tool for managing multple ArcSight products. You can easily implement configurations across managed products with just a few actions.

- **Node management versus Configuration Management:** Use ArcSight Management Center's node management tools for the administration of individual nodes and their day-to-day operations. However, for consistent and wide-ranging changes to the data or settings of managed nodes, use configuration management if the appropriate configuration exists. For example, to change DNS settings across multiple managed nodes, it would be faster and easier to create the configuration in ArcMC and push it out to managed nodes, than to individually change the settings across multiple devices.
- **Implementing data settings across multiple appliances or products in bulk:** Use the Bulk Management (**Set Configuration**) tools to implement data settings across multiple appliances or products. For example, you can quickly configure all of your appliances to use the same hardware settings (such as SMTP server) with a single platform (in this case, SMTP) configuration applied to managed nodes. (Pushing will overwrite any existing data.)
- **Compliance versus Non-Compliance:** If configuration compliance is not relevant to your configuration management, use the bulk management tools under Node Management to manage your node settings. A bulk push can also be performed under Configuration Management.

Subscriber Configuration Types

The following section lists the available subscriber configuration types, the parameters associated with each, their data types, and a brief description of what the parameter represents. When assigning values to parameters:

- Each parameter's value must be of the data type indicated (for example, the String data type indicates that you must specify a string for the value).
- *Required* parameters are marked with an asterisk (*) and must be assigned a value. A configuration missing a value for a required parameter cannot be saved or pushed.
- *Read-only* parameters cannot be edited in ArcSight Platform.
- For security reasons, all password parameters are displayed with obfuscation.

> **Tip:** For details of each entry field, in edit mode, hover over the field label and view its descriptive tooltip.

Connector Configuration Types

Connector configurations set values for settings on containers, connectors, or Collectors. The available connector configuration types are listed here.

BlueCoat Connector Configuration

A BlueCoat Connector configuration defines settings for one or more BlueCoat connectors. The configuration is only pushed to a target if a BlueCoat connector exists.

To push a BlueCoat Connector configuration from ArcSight Platform to a managed node that already has values defined for all fields listed here, then specify values for all fields in the pushed configuration. Default values may be used if necessary.

BlueCoat Connector Configuration Parameters

Parameter	Data Type	Description
Row Number*	Integer	Row number of the table parameter to which the configuration is pushed.
Log File Wildcard*	String	Log file wildcard.
Log File Type*	String	Log file type. Valid values are: • `main` • `im` • `ssl` • `streaming`

BlueCoat Connector Configuration Parameters, continued

Parameter	Data Type	Description
Processing Mode	String	Processing mode. Valid values are Batch and Real time.
Post-Processing Mode	String	Post-processing mode. Valid values are: • `RenameFileInTheSame Directory` • `PersistFile` • `DeleteFile`
Mode Options	String	Mode options. Required if Post-Processing Mode is chosen as `RenameFileInTheSame Directory`
Processing Threshold	Integer	Interval, in hours, after which the log file will be marked as processed.
Processing Limit	Integer	Number of files that can be read in the directory at the same time.

FIPS Configuration

A FIPS configuration enables or disables FIPS mode on a container.

Note: After pushing a FIPS configuration, the destination container will be restarted.

FIPS Configuration Parameters

Parameter	Data Type	Description
Enabled*	Boolean	If **Yes**, FIPS is enabled on the container.

Map File Configuration

A map file configuration defines the path and content of one or more container map files. Each Path/Content pair represents a single map file. To include multiple files, add multiple Properties to the configuration.

- When pushed, the configuration deletes all `*.properties` files in the `\map` directory on the target, then adds the list of map files to the target, replacing any existing map files.
- If the configuration contains an empty list, all `*.properties` files are deleted.

Note: If importing and uploading a map configuration file, convert the downloaded CSV file into a .properties file before uploading.

Uploading Map Files Larger Than 1 MB

a. Log in to the CDF Management Portal. See "Accessing the CDF Management Portal" on page 680 for more information.
b. From the left menu select **Deployment > Deployments.**
c. Click **... (Browse)** on the far right and choose **Reconfigure**. A new screen will open in a separate tab.
d. Select the **Fusion** tab
e. Scroll down to the **ArcMC Configuration** section, and specify the desired value for the **Maximum In-memory Buffer Size** parameter.
f. Click **Save**. The ArcMC pod will be restarted.

Map File Configuration Parameters

Parameter	Data Type	Description
Path*	String	Path to the map file.
Content*	String	Content of the map file.

Parser Override Configuration

A parser override configuration defines the path and content of one or more container parser override files.

Each Path/Content pair represents a single parser override file. To include multiple files, add multiple Properties to the configuration.

- When pushed, the configuration deletes all `*.properties` files in the `\fcp` directory on the target, then adds the list of parser override files to the target, replacing any existing parser override files.
- If the configuration contains an empty list, all `*.properties` files are deleted.

Parser Override Configuration Parameters

Parameter	Data Type	Description
Path*	String	Path to the parser override file.
Content*	String	Content of the parser file.

Syslog Connector Configuration

A Syslog connector configuration defines values for one or more Syslog connectors. The configuration is only pushed to the target node if a Syslog connector exists.

Syslog Connector Configuration Parameters

Parameter	Data Type	Description
Port*	Integer	Syslog connector port.
Protocol*	Enum	Protocol of the syslog connector (either UDP or Raw TCP).

Windows Unified Connector (WUC) External Parameters Configuration

A WUC External Parameters connector configuration defines the external parameters for one or more WUC connectors. The configuration is only pushed to the target node if a WUC connector exists.

Limitations to WUC External Parameters Configurations

A WUC external parameters configuration has the following limitations:

- Domain user password is not supported as a WUC configuration parameter. Instead, domain user password must be managed individually for each WUC host.
- WUC connectors are not FIPS-compliant.
- If you wish to push a WUC configuration from ArcMC to a managed node that already has values defined for all fields listed here, then you must specify values for all fields in the pushed configuration. Default values may be used if necessary.

  WUC External Parameters Configuration Parameters

Parameter	Data Type	Description
Domain Name*	String	Windows domain name.
Domain User*	String	Windows domain user name.
Active Directory Host	String	Hostname for the Active Directory server, if one is used. ◦ If specified, values for User, User Password, Base DN, Protocol, and Port must be specified in subsequent entries.
Active Directory Use	String	Username for the AD server. ◦ Required if a value is provided for Active Directory Host.
Active Directory User Password	String	Password for AD server. ◦ Required if a value is provided for Active Directory Host.
Active Directory Base DN	String	Base DN of the Active Directory. ◦ Required if a value is provided for Active Directory Host.

WUC External Parameters Configuration Parameters, continued

Parameter	Data Type	Description
Active Directory Protocol	String	Protocol for Active Directory. ◦ Required if a value is provided for Active Directory Host.
Active Directory Port	String	Port for Active Directory. ◦ Required if a value is provided for Active Directory Host.
Global Catalog Server	String	Hostname for the Global Catalog server, if one is used. ◦ If specified, values for User Name, User Password, and Base DN must be specified in subsequent entries.
Global Catalog User Name	String	Username for the GC server. ◦ Required if a value is provided for Global Catalog server.
Global Catalog User Password	String	Password for the GC server. ◦ Required if a value is provided for Global Catalog server.
Global Catalog Base DN	String	Base DN of the GC server. ◦ Required if a value is provided for Global Catalog server.
WEF Collection*	String	Indicates if Windows Event Format collection is enabled. Valid values are: ◦ Disabled ◦ Enabled (use Active Directory for sources) ◦ Enabled (do not use Active Directory for sources) **Note:** WEF collection is only supported for Connector versions 6.0.6 or later. Otherwise, compliance checks for checks for WUC External Parameters configurations will always fail.

Windows Unified Connector (WUC) Internal Parameters Configuration

A WUC Internal Parameters connector configuration defines the internal parameters for one or more WUC connectors. The configuration is only pushed to the target if a WUC connector exists.

Limitations to WUC Internal Parameters Configurations

A WUC internal parameters configuration has the following limitations:

- Domain user password is not supported as a WUC configuration parameter. Instead, domain user password must be managed individually for each WUC host.
- WUC connectors are not FIPS-compliant.
- If you wish to push a WUC configuration from ArcMC to a managed node that already has

values defined for all fields listed here, then you must specify values for all fields in the pushed configuration. Default values may be used if necessary

WUC Internal Parameters Configuration Parameters

Parameter	Data Type	Description
Enable GUID Translation*	Boolean	If true, Globally Unique Identifier translation is enabled.
Enable SID Translation*	Boolean	If true, Security Identifier translation is enabled.
Enable SID Translation Always*	Boolean	If true, SID translation is used even for events Windows does not translate.
FCP Version	Integer	File Control Protocol version number.
Global Catalog Port	Integer	Port used by Global Catalog server.
Global Catalog Security Protocol	Enum	Security protocol used by Global Catalog server.
Host Browsing Threads Sleep Time	Integer	Time in milliseconds between host browsing queries.
Inactivity Sleep Time	Integer	Time in milliseconds to sleep if no events are retrieved from the configured hosts
Log Rotation Check Interval	Integer	Time in milliseconds to wait before checking for log rotation.
Reconnect Interval	Integer	Time in milliseconds after which the connection to a previously down host is to be retried.
Rotation Retry Count	Integer	Number of times to check that log has been rotated.
Rotation Retry Interval	Integer	Interval in milliseconds for rotation retry.
Sleep Time	Integer	Time, in milliseconds, to sleep before collecting more events from hosts (-1 means disable sleep time).
Thread Count	Integer	Number of threads to use for the connector.

ArcMC/Connector Appliance Configuration Types

ArcMC/Connector Appliance configurations set values for settings on Software ArcSight Management Centers, ArcSight Management Center Appliances, and hardware or software Connector Appliances. The currently available ArcMC/Connector Appliance configuration type is listed here.

ArcMC/Connector Appliance Configuration Backup Configuration

An ArcMC/Connector Appliance Configuration Backup configuration sets values for scheduled configuration backups of ArcSight Management Center or Connector Appliance. Backup content includes all backup data.

After a push, the web process is automatically restarted on the subscriber.

For this configuration type, no automatic compliance checks will be performed. You must check compliance manually. The following limitation applies:

- This Configuration is not supported if the Backup Server platform is CentOS 7.4.

Note: You can neither create nor import settings related to a one-time configuration backup.

ArcMC/Connector Appliance Configuration Backup Parameters

Parameters	Data Type	Description
Backup Server IP Address*	String	IP address of the remote system where the backup will be saved.
Port*	Integer	Port of the remote system. Default value is 22.
Base Remote Directory*	String	Destination directory on the remote system. Must be manually created on remote system prior to push. After a push, the destination host name is appended to this, to give it a unique value across all nodes.
User*	String	User name on destination.
Password*	String	Password on the destination. (Obfuscated.)
Days of the Week*	List of comma-separated strings	Comma-delimited list of days of the week on which the backup will be performed. Valid values are *Su, M, T, W, Th, F, Sa*.
Hours of Day*	List of comma-separated integers	Comma-delimited list of hours of the day at which the backup will be performed. Valid values are integers from 0 to 23, where 0 is 12:00 midnight. For example, a value of 14 would correspond to 2 PM.

Destination Configuration Types

A destination configuration sets values for ESM destination settings on Connectors/Collectors. The available destination configuration types are listed here.

Destination Configuration Parameters

A Destination Configuration Parameters configuration defines values and behavior for destination configuration parameters.

Note: Destination Configuration Parameters configurations can only be imported from managed Collectors/Connectors, not created in ArcMC. See "Importing a Subscriber Configuration" on page 997 for more information.

For a description of the parameters for this configuration type, see "Destination Runtime Parameters " on page 1055.

Networks and Zones

A Networks and Zones configuration defines values and behavior for ArcSight ESM networks and zones. For more information on ESM networks and zones, consult the ArcSight Console documentation. For Connector Network and Zone Configuration information see the Smart Connector's User Guide.

Note: So as not to interfere with ESM connector management, ArcMC will not push Network and Zones AUPs to a connector's ESM destination folder.

Networks and Zones Configuration Parameters

Parameter	Data Type	Description
Configuration Name*	String	Name of the configuration.

Networks CSV Content*	CSV	Comma-separated Value (CSV) file. Click **Upload** to upload a valid CSV file, or click **Download** to download an existing file.
		Creating a CSV File
		The CSV must include the literal header line:
		`#Type,Name,Parent Group URI,Customer URI`
		Then, each line describes a Network. Each line must comprise values for the following fields, and end with a hard return (no white spaces). Begin the first of these network lines with the # character before Type.
		`<Type>,<Name>,<Parent Group URI>,<Customer URI>`

Networks and Zones Configuration Parameters, continued

Parameter	Data Type	Description

Zones CSV Content*	CSV	Comma-separated Value (CSV) file. Click **Upload** to upload a valid CSV file, or click **Download** to download an existing file.
		Creating a CSV File
		The CSV must include the literal header line:
		`#Name,Start Address,End Address,Parent Group URI,Network URI`
		Then, each line describes a Zone. Each line must comprise values for the following fields, and end with a hard return (no white spaces). Begin the first of these zone lines with the # character before Name.
		`<Name>,<Start Address>,<End Address>,<Parent Group URI>,<Network URI>`

Logger Configuration Types

Logger configurations set values for settings on hardware and software Loggers. The available Logger configuration types are listed here.

Logger Configuration Backup Configuration

A Logger configuration backup configuration sets values for scheduled configuration backups of hardware and software Logger to a remote system. The following limitation applies:

- This Configuration is not supported if the Backup Server platform is CentOS 7.4.

Note: You can neither create nor import settings related to a one-time configuration backup.

Logger Configuration Backup Configuration Parameters

Parameter	Data Type	Description
SCP Port*	String	Port of the remote system. Default value is 22.
Backup Server IP Address*	String	IP address of the remote system where the backup will be saved.
Username*	String	User name on destination.
Password*	String	Password on destination. (Obfuscated.)

Logger Configuration Backup Configuration Parameters, continued

Parameter	Data Type	Description
Base Remote Directory*	String	Destination directory on the remote system. After a push, the destination host name is appended to this, to give it a unique value across all nodes. When using a Logger appliance, some settings need to be configured in the `/etc/hosts` file. For more information, please refer to the *Configuring Hosts for the Appliance* chapter in the Logger Installation Guide.
Days of the Week*	List of comma-separated strings	Comma-delimited list of days of the week on which the backup will be performed. Valid values are *Su, M, T, W, Th, F, Sa*.
Hours of Day*	List of comma-separated integers	Comma-delimited list of hours of the day at which the backup will be performed. Valid values are integers from 0 to 23, where 0 is 12:00. For example, a value of 14 would correspond to 2 PM.
Backup Content*	String	Type of content to be included in the backup. Valid values are: • *All:* includes all backup data. • *Report_Content_Only*: includes only report data.

Logger Connector Forwarder Configuration

A Logger Connector Forwarder configuration sets values for one or more connector forwarders on a Logger (version 6.1 or later). Each forwarder in the configuration is represented by a different Property.

Note: Logger Connector Forwarder configurations can only be imported from managed Loggers, not created in ArcMC. See "Importing a Subscriber Configuration" on page 997 for more information.

Logger Connector Forwarder Configuration Parameters

Parameter	Data Type	Description
Forwarder Name*	String	Display name of the forwarder
Filter Type*	Enum	Filter type that was selected while creating a forwarder on logger. Valid types are *Unified* or *Regex*.
Query	String	Used to filter events that the forwarder will forward.
Unified Query Filters	String	Select from the default and user-defined Unified filters on the source Logger. Only visible if Filter Type is Unified.

Logger Connector Forwarder Configuration Parameters, continued

Parameter	Data Type	Description
Regular Expression Filters	String	Select from the default and user-defined Regex filters on the source Logger. Only visible if Filter Type is Regex.
Start Time	DateTime	Optional start of time range for selection.
End Time	DateTime	Optional end of time range for selection.
IP/Host*	String	IP address or host name of the destination that will receive forwarded events.
Port*	Integer	Port number on the destination that will receive forwarded events. Ensure this port is open on the destination.
Enable*	Boolean	If **Yes**, the forwarder is enabled.
Connection Retry Timeout*	Integer	Time, in seconds, to wait before retrying a connection.
Source Type*	Integer	Source Type. Valid values: • Apache HTTP Server Access • Apache HTTP Server Error • IBM DB2 Audit • Juniper Steel-Belted Radius • Microsoft DHCP Log • Other

Logger ESM Forwarder Configuration

A Logger ESM Forwarder configuration sets values for one or more ESM destinations on a Logger (version 6.1 or later). Each destination in the configuration is represented by a different Property.

Note: Logger ESM Forwarder configurations can only be imported from managed Loggers, not created in ArcMC. See "Importing a Subscriber Configuration" on page 997 for more information.

Logger ESM Forwarder Parameters

Parameter	Data Type	Description
Parameter	Data Type	Description
Forwarder Name*	String	Display name of the forwarder
Filter Type*	Enum	Filter type that was selected while creating a forwarder on logger. Valid types are *Unified* or *Regex*.
Query	String	Used to filter events that the forwarder will forward.

Logger ESM Forwarder Parameters, continued

Parameter	Data Type	Description
Unified Query Filters	String	Select from the default and user-defined Unified filters on the source Logger. Only visible if Filter Type is Unified.
Regular Expression Filters	String	Select from the default and user-defined Regex filters on the source Logger. Only visible if Filter Type is Regex.
Start Time	DateTime	Start of time range for selection.
End Time	DateTime	End of time range for selection.
IP/Host*	String	IP address or host name of the destination that will receiveforwarded events.
Port*	Integer	Port number on the destination that will receive forwarded events. Ensure this port is open on the destination.
Enable	Boolean	If **Yes**, the forwarder is enabled.

Logger Filter Configuration

A Logger Filter configuration sets values for one or more saved searches on a Logger.

Each filter in the configuration is represented by a different Property.

Note: Logger Filter configurations can only be imported from managed Loggers, not created in ArcMC. See "Importing a Subscriber Configuration" on page 997 for more information.

Logger Filter Configuration Parameters

Parameter	Data Type	Description
Filter Name*	String (Read-only)	Name of the filter.
Filter Category	String	Category of filter. Valid values are **Shared**, **System** and **SearchGroup**.

Logger Filter Configuration Parameters, continued

Parameter	Data Type	Description
Filter Type*	String	Type of filter. Valid values are **RegexQuery** or **UnifiedQuery**.
Query*	String	Query string.
Permission Group	String	Permission group which with the Logger filter is associated. When the configuration is pushed: • If the permission group is not present on the target Logger, the permission group will be created during the push. • If the permission group of the same name is already present on the target, but has different rights, the rights of the permission group on the target Logger will not be overwritten, and the association between the filter and the permission group will be removed.

Logger SmartMessage Receiver Configuration

A Logger SmartMessage Receiver sets values for one or more for SmartMessage Receivers.

A SmartMessage Receiver configuration pushed to a target overwrites any existing SmartMessage receivers on the target; other types of receivers such as UDP and TCP are not affected.

Logger SmartMessage Receiver Configuration Parameters

Parameter	Data Type	Description
Receiver Name*	String	Name of the receiver.
Enabled*	Boolean	If **Yes**, SmartMessage reception is enabled.
Encoding*	String	Encoding type. Valid values are: • `UTF-8` • `US-ASCII`

Logger Storage Group Configuration

A Logger Storage Group configuration sets values for one or more Logger storage groups.

Note: Logger Storage Group configurations can only be imported from managed Loggers, not created in ArcMC. See "Importing a Subscriber Configuration" on page 997 for more information.

Logger Storage Group Configuration Parameters

Parameter	Data Type	Description
Storage Group Name*	String (Read-only)	Name of the storage group. • The pushed configuration must contain the same number of storage groups as configured on the Logger. • The names of the storage groups in the pushed configuration must match the names of storage groups on the Logger.
Maximum Age (Days)*	Integer	Maximum age of events in storage, in days.
Maximum Size (GB)*	Integer	Maximum size of the storage group, in gigabytes. • The cumulative size of all storage groups must not be greater than the storage volume size on the Logger.

Logger TCP Forwarder Configuration

A Logger Connector Forwarder configuration sets values for one or more TCP forwarders on a Logger (version 6.1 or later). Each forwarder in the configuration is represented by a different Property.

Note: Logger TCP Forwarder configurations can only be imported from managed Loggers, not created in ArcMC. See "Importing a Subscriber Configuration" on page 997 for more information.

Logger TCP Forwarder Configuration Parameters

Parameter	Data Type	Description
Forwarder Name*	String	Display name of the forwarder
Filter Type*	Enum	Filter type that was selected while creating a forwarder on logger. Valid types are *Unified* or *Regex*.
Query	String	Used to filter events that the forwarder will forward.
Unified Query Filters	String	Select from the default and user-defined Unified filters on the source Logger. Only visible if Filter Type is Unified.
Regular Expression Filters	String	Select from the default and user-defined Regex filters on the source Logger. Only visible if Filter Type is Regex.
Start Time	DateTime	Optional start of time range for selection.
End Time	DateTime	Optional end of time range for selection.
IP/Host*	String	IP address or host name of the destination that will receive forwarded events.
Port*	Integer	Port number on the destination that will receive forwarded events. Ensure this port is open on the destination.
Enable*	Boolean	If **Yes**, the forwarder is enabled.

Logger TCP Forwarder Configuration Parameters, continued

Parameter	Data Type	Description
Preserve System Timestamp*	Boolean	If **Yes**, the timestamp showing original event receipt time is preserved.
Preserve Original Syslog Sender*	Boolean	If **Yes**, event is sent as is, without inserting Logger's IP address in the hostname (or equivalent) field of the syslog event.
Connection Retry Timeout*	Integer	The time, in seconds, to wait before retrying a connection.

Logger Transport Receiver Configuration

A Logger Transport Receiver configuration sets values for one or more UDP, TCP, CEF UDP, or CEF TCP receivers.

Note: In Logger documentation, a *Transport Receiver* is referred to as simply a *Receiver*.

A pushed Transport Receiver type configuration will overwrite any existing UDP, TCP, CEF UDP, or CEF TCP receiver. Any other type of receivers, such as SmartMessage receivers, are not affected.

Logger Transport Receiver Configuration Parameters

Parameter	Data Type	Description
Receiver Name*	String	Name of the receiver.
Receiver Type*	String	Receiver type. Valid values are: • UDP • TCP • CEF UDP • CEF TCP
Receiver Name*	String	Name of the receiver.

Logger Transport Receiver Configuration Parameters, continued

Parameter	Data Type	Description
Port*	Integer	Port number. Must be a non-zero positive number. Ensure this port is open on the destination.
Enabled*	Boolean	If **Yes**, transport reception is enabled.
Encoding*	String	Encoding type. Valid values are: • `UTF-8` • `Shift_JIS` • `EUC-JP` • `EUC-KR` • `US-ASCII` • `GB2312` • `UTF-16BE` • `Big5` • `GB18030` • `ISO-8859-1` • `Windows-1252` For CEF UDP and CEF TCP receivers, only UTF-8 and US-ASCII apply. **Caution:** Selection of the wrong encoding for a CEF receiver will cause a push failure.

Logger UDP Forwarder Configuration

A Logger Connector Forwarder configuration sets values for one or UDP forwarders on a Logger. Each forwarder in the configuration is represented by a different Property.

Note: Logger UDP Forwarder configurations can only be imported from managed Loggers, not created in ArcMC. See "Importing a Subscriber Configuration" on page 997 for more information.

Logger UDP Forwarder Configuration Parameters

Parameter	Data Type	Description
Forwarder Name*	String	Display name of the forwarder
Filter Type*	Enum	Filter type that was selected while creating a forwarder on logger. Valid types are *Unified* or *Regex*.
Query	String	Used to filter events that the forwarder will forward.
Unified Query Filters	String	Select from the default and user-defined Unified filters on the source Logger. Only visible if Filter Type is Unified.

Logger UDP Forwarder Configuration Parameters, continued

Parameter	Data Type	Description
Regular Expression Filters	String	Select from the default and user-defined Regex filters on the source Logger. Only visible if Filter Type is Regex.
Start Time	DateTime	Optional start of time range for selection.
End Time	DateTime	Optional end of time range for selection.
IP/Host*	String	IP address or host name of the destination that will receive forwarded events.
Port*	Integer	Port number on the destination that will receive forwarded events. Ensure this port is open on the destination.
Enable*	Boolean	If **Yes**, the forwarder is enabled.
Preserve System Timestamp*	Boolean	If **Yes**, the timestamp showing original event receipt time is preserved.
Preserve Original Syslog Sender*	Boolean	If **Yes**, event is sent as is, without inserting Logger's IP address in the hostname (or equivalent) field of the syslog event.

SecureData Configuration

A SecureData configuration sets values for the SecureData encryption client on a managed Logger.

SecureData Configuration Parameters

Parameter	Data Type	Description
Server*	String	SecureData server IP address.
Port*	String	SecureData server port.
Auth Identity*	String	SecureData authentication identity
Shared Secret*	String	SecureData shared secret
Event Fields*	String	Comma-separated list of event fields to be encrypted. Default data for event fields will be populated from the connector bin file uploaded in the repository. If there is no such file, then the default field will be defined by ArcMC.

System Admin Configuration Types

System Admin configurations set values for system administrative settings. The available System Admin configuration types are listed here.

Authentication External

An Authentication External configuration defines values and behavior for a hardware or software system requiring authentication to an external server, such as LDAP or RADIUS.

After changing the Authentication Method on a host, you must delete the host from ArcMC, then re-add it using Node Management.

Note: Authentication External configurations can only be imported from managed Loggers, not created in ArcMC. See "Importing a Subscriber Configuration" on page 997 for more information.

Authentication External Configuration Parameters

Parameter	Data Type	Description
Authentication Method*	String	System authentication method.
Allow Local Password Fallback for Default Admin Only*	Boolean	If **Yes**, the authentication server will fall back to local passwords for authentication for administrators.
Allow Local Password Fallback for All Users*	Boolean	If **Yes**, the authentication server will fall back to local passwords for authentication for all users.
LDAP Server Hostname[port]*	String	LDAP server hostname and port.
LDAP Backup Server Hostname [port]	String	LDAP backup server hostname and port.
LDAP Server Request Timeout (seconds)	Integer	LDAP server request timeout, in seconds.
RADIUS Server Hostname[port]	String	RADIUS server hostname and port.
RADIUS Backup Server Hostname [port]	String	RADIUS backup server hostname and port.
RADIUS Shared Authentication Secret	String	RADIUS authentication shared secret.
RADIUS Server NAS IP Address	String	RADIUS server Network Access Server IP address .
RADIUS Request Timeout (seconds)	Integer	RADIUS server request timeout, in seconds.
RADIUS Retry Request	Integer	Number of times to retry RADIUS server requests.
RADIUS Protocol	String	Type of RADIUS protocol.

Authentication Local Password

An Authentication Local Password configuration defines a hardware or software system's local password options and behavior.

Authentication Local Password Configuration Parameters

Parameter	Data Type	Description
Enable Account Lockout*	Boolean	If **Yes**, account lockouts are enabled after an incorrect password entry.
Lock Out Account after N Failed Attempts*	Integer	Number of failed attempts before lockout.
Remember Failed Attempts For (seconds)*	Integer	Time, in seconds, between failed attempts that will trigger a lockout.
Lockout Account for (minutes)*	Integer	Time, in minutes, that the account will be locked out.
Enable Password Expiration*	Boolean	If **Yes**, password expiration is enabled
Password Expires in (days)*	Integer	Interval, in days, after which a password expires.
Notify User (Days Before Expiration)*	Integer	Days before password expiration that the user is notified.
Users Exempted from Password Expiration Policy	List of comma-separated strings	Comma-separated list of users whose passwords will never expire.
Enforce Password Strength*	Boolean	If **Yes**, password strength is enforced.
Minimum Length (characters)*	Integer	Minimum number of password characters.
Maximum Length (characters)*	Integer	Maximum number of password characters.
Numeric [0-9]*	Integer	Minimum number of numeric password characters.
Upper Case [A-Z]*	Integer	Minimum number of uppercase password characters.
Lower Case [a-z]*	Integer	Minimum number of lowercase password characters
Special [1$^*...]*	Integer	Minimum number of special password characters.
Password Must Be At Least*	Integer	Minimum number of characters a new password must differ from the user's previous password.
Include "Forgot Password" link on Login Screen*	Boolean	If **Yes**, a link is provided where the user can recover a password.

Authentication Session

An Authentication Session configuration defines values for a hardware or software system's authentication sessions.

Authentication Session Configuration Parameters

Parameter	Data Type	Description
Max Simultaneous Logins Per User*	Integer	Maximum number of simultaneous logins per user. If Max Simultaneous Logins/User is set to 1, it is required to have at least another admin user, otherwise the admin user will not be able to log in.
Logout Inactive Session After (seconds)*	Integer	Inactivity session timeout, in seconds.
Disable Inactive Account After (days)*	Integer	Number of days of inactivity after which an account will be disabled.

DNS Configuration

A DNS Configuration defines values for a hardware appliance's Domain Name Service.

DNS Configuration Parameters

Parameter	Data Type	Description
Primary DNS*	String	Primary DNS server.
Secondary DNS	String	Secondary DNS server.
DNS Search Domains	List of comma-separated strings	Comma-separated list of DNS search domains.

FIPS Configuration

A FIPS configuration enables or disables FIPS mode on a managed node.

Note: After pushing a FIPS configuration, the destination node will be restarted.

FIPS Configuration Parameters

Parameter	Data Type	Description
Enabled*	Boolean	If **Yes**, FIPS is enabled on the node.

Network Configuration

A Network Configuration defines values for a hardware appliance's default gateway setting.

Note: Values for these network settings cannot be changed through ArcSight Management Center: hostname, IP addresses for the network interfaces, static routes, /etc/hosts file, and time settings.

Network Configuration Parameters

Parameter	Data Type	Description
Default Gateway*	String	Default network gateway.

NTP Configuration

An NTP Configuration defines values for a hardware appliance's Network Time Protocol.

NTP Configuration Parameters

Parameter	Data Type	Description
Enable as NTP Server*	Boolean	If **Yes**, the system is enabled as an NTP server.
NTP Servers*	List of comma-separated strings	Comma-separated list of NTP servers. Required even if **Enable as NTP Server** is false.

SMTP Configuration

An SMTP Configuration defines values for a hardware or software system's Simple Mail Transfer Protocol.

SMTP Configuration provides for authentication and security. This is implemented through the primary STMP server port, primary username, primary password, primary certificate, backup STMP server port, backup username, backup password, and backup certificate fields, along with the primary STMP server, backup STMP server, and outgoing email address fields.

SMTP Configuration Parameters

Parameter	Data Type	Description
Primary SMTP Server*	String	Primary SMTP server.
Secondary SMTP Server	String	Secondary SMTP server.
Outgoing Email Address*	String	Outgoing email address.
Enable Auth/TLS	Boolean	Enable/Disable secure authenticated mode of communication with SMTP server
Primary SMTP Server Port	Integer	Primary SMTP Server Port. Required if Auth/TLS is enabled.

SMTP Configuration Parameters, continued

Parameter	Data Type	Description
Primary SMTP Server Username	String	Primary SMTP Server Username. Required if Auth/TLS is enabled.
Primary SMTP Server Password	String	Primary SMTP Server Password. Required if Auth/TLS is enabled.
Primary SMTP Server Certificate Content	String	Upload Primary SMTP Server Certificate. Required if Auth/TLS is enabled.
Secondary SMTP Server Port	Integer	Secondary SMTP Server Port. Required if Auth/TLS is enabled.
Secondary SMTP Server Username	String	Secondary SMTP Server Username. Required if Auth/TLS is enabled.
Secondary SMTP Server Password	String	Secondary SMTP Server Password. Required if Auth/TLS is enabled.
Secondary SMTP Server Certificate Content	String	Upload secondary SMTP Server Certificate. Required if Auth/TLS is enabled.

SNMP Poll Configuration

An SNMP Poll Configuration defines values for a hardware appliance's Simple Network Management Protocol monitoring. ArcSight Platform supports V2c and V3 of SNMP.

SNMP Poll Configuration Parameters

Parameter	Data Type	Description
Status	Boolean	If **Yes**, SNMP polling is enabled.
Port*	Integer	SNMP port.
SNMP Version*	String	Version of SNMP supported.Valid values are v2c and v3.
Community String	String	SNMP community string. Required for V2c only.
Username	String	Authentication username. Required for V3 only.
Authentication Protocol*	String	Authentication protocol. Valid values are MD5 and SHA. Required for V3 only.
Authentication Passphrase	String	Authentication passphrase. Required for V3 only.
Privacy Protocol	String	Privacy protocol. Valid values are DES and AES128. Required for V3 only.
Privacy Passphrase	String	Privacy passphrase. Required for V3 only.
System Name	String	Name of the SNMP system.
Point of Contact	String	Point of contact.
Location	String	System location.

SNMP Trap Configuration

An SNMP Trap Configuration defines values for a hardware appliance's Simple Network Management Protocol notifications. ArcSight Platform supports V2c and V3 of SNMP.

Note: In previous versions of ArcSight Platform, an SNMP Trap configuration was known as an SNMP Configuration.

SNMP Trap Configuration Parameters

Parameter	Data Type	Description
Status	Boolean	If **Yes**, SNMP polling is enabled.
NMS IP Address	String	IP address of network management server.
Port*	Integer	SNMP port.
SNMP Version*	String	Version of SNMP supported.Valid values are v2c and v3.
Community String	String	SNMP community string. Required for V2c only.
Username	String	Authentication username. Required for V3 only.
Authentication Protocol*	String	Authentication protocol. Valid values are MD5 and SHA. Required for V3 only.
Authentication Passphrase	String	Authentication passphrase. Required for V3 only.
Privacy Protocol	String	Privacy protocol. Valid values are DES and AES128. Required for V3 only.
Privacy Passphrase	String	Privacy passphrase. Required for V3 only.

Logger Initial Configuration Management

A *Logger initial configuration* is intended for the rapid, uniform setup of multiple ArcSight Loggers of the same model number and software version. Use a Logger initial configuration to expedite the initial deployment of Loggers to a production environment. Initial configuration management is supported on Logger version 6.1 or later.

A Logger initial configuration is not created in ArcMC. Instead, a suitable initial configuration is created on a managed Logger and imported into ArcMC. The configuration may then be pushed to other managed Loggers of the same model and software version number.

The following attributes are shown for each initial configuration:

Attribute	Description
Imported Init-Config Name	Name of the imported initial configuration.
Product Type	Type of Logger to which the configuration may be pushed: either Logger (appliance) or SWLogger (software)
Source Host	IP address of the host from which the configuration was imported.
Imported On	Date of import.
Imported By	User who imported the configuration.
SW Version	Software version of the configuration.
Source Model	For appliances, the model number of the source host Logger. (For software Logger, this is shown as Software.)

You can perform the following initial configuration management tasks:

- Import an Initial Configuration
- Push an Initial Configuration
- View the Initial Configuration Event History
- Delete an Initial Configuration

Importing a Logger Initial Configuration

An initial configuration created on a managed Logger (of version 6.1 or later) may be imported into ArcSight Platform, for editing and pushing to other Loggers.

ArcMC can store up to 30 initial configurations.

To import an initial configuration from a Logger of version 6.1 or later:

1. Click **Configuration Management > Logger Initial Configurations.**
2. Under **Configurations**, click **Import**.
3. On the **Import Initial Configuration** dialog, in **Name**, specify a name for the configuration you wish to import.
4. Under **Source Host URI**, select the node from which you wish to import the configuration.
5. Click **Import**. The configuration is imported into ArcSight Platform and is shown in the **Configurations** table.
6. Optionally, if you wish to push the imported configuration to managed nodes, when prompted to push, click **Yes**.

Note: An initial configuration is not created in ArcMC. Instead, create the initial configuration on a managed Logger, then import it into ArcMC for pushing to other managed Loggers.

Pushing a Logger Initial Configuration

You can push a Logger initial configuration to selected managed Loggers of version 6.1 or later. The destination Loggers must be of the same software version (and, for hardware appliances, model number) as the Logger on which the initial configuration was created.

The push process overwrites the settings on the destination Loggers.

Pushing a Logger initial configuration must be performed manually.

Note: Before performing a push, ensure that the destination Logger's storage volume is set up, and that it exceeds that of any source Logger.

To push an initial configuration to one or more managed Loggers of version 6.1 or later:

1. Click **Configuration Management > Logger Initial Configurations.**
2. From the **Configurations** table, select a configuration to be pushed.
3. Click **Push**.
4. On the **Make Selections for Push** dialog, under **Available Nodes**, the nodes eligible for receiving a push are displayed by location. Browse to the recipient node and click **Add**. The selected node is shown under **Selected Nodes.** (To select multiple nodes to receive a push, Ctrl+click each selected node.)
5. Click **Push**.
6. Click **Yes** to confirm the push and change settings on the destinations. The configuration is pushed to the selected destination nodes.

Tip: In order to correctly view push status, click **Refresh**, even if the status is shown as In Progress.

Push Results on a Destination Logger

The results of a push of an initial configuration on a given setting of a destination Logger are dependent on the setting, as shown in the following table.

Setting on Destination	Result After Push
• Archive storage settings • Audit logs • ESM destinations • Event archives • Finished tasks • Forwarders • Peer Loggers	**Blank:** These settings will be blank on the destination, even if they are included in the pushed initial configuration. Also, all configurations on the destination Logger related to these settings will also be blanked.
• Alerts • User-created receivers (RFSFileReceiver, FileTransfer, FolderFollowerReceiver)	**Disabled:** These settings are disabled on the destination Logger, but are editable through the destination Logger's UI.
• Hosts file • Groups • Users	**Copied From Source:** These values are copied from the initial configuration and overwritten on the target. This may include user credentials that the Logger uses to authenticate to ArcMC, which could break the management link between ArcMC and the destination Logger (which requires these credentials). If an overwrite of these credentials occurs, to enable management, delete the host from ArcMC, then re-add the Logger as a host (with the new credentials).
• All other settings	**Copied From Source:** Values are copied from the initial configuration and overwritten on the target.

Deleting a Logger Initial Configuration

A deleted initial configuration is no longer available for pushes. You may not delete a configuration currently being pushed.

To delete an initial configuration:

1. Click **Configuration Management > Logger Initial Configurations.**
2. From the **Logger Initial Configurations** table, select one or more configurations to be deleted.
3. Click **Delete.**
4. Click **Yes** to confirm deletion.

Event History

The **Event History** list records all imports, pushes, and deletes transactions related to initial configuration pushes. Each event in the history displays the following information:

Column	Description
Init-Config Name	Initial configuration's name.
Author	User who performed the action.
Event Type	Type of event recorded for the initial configuration. Event types include Push, Import, and Delete.
Event Occurrence	Local date and time of the event.
Source Host	URI of the host on which the initial configuration was created.
Destination URI for Push	If the event is of type Push, this is the URI of the destination node to which the initial configuration was pushed.
Event Status	Status of the event. Status types include: • In-progress: the transaction is still in progress. • Successful: the transaction succeeded. • Failed: the transaction failed. Click the failed status to view an indication of the failure reason.

To search for a specific event by any of these criteria, click the drop-down in the corresponding column header. Then, in **Filters**, select or specify the specific criterion for which you wish to show events. Only events matching the filter will be displayed in the **Event History** list.

For example, to see all pushes, in the **Event Type** column, click the header drop-down. Then, in **Filters**, select *Push*.

Managing Logger Event Archives

Logger Event Archives enable you to save the events for any day in the past (not including the current day). In ArcSight Platform, you can view Logger Event Archives on managed Loggers, and perform management tasks including loading, unloading, and indexing archives.

Logger Event Archive management is only available for managed Loggers of version 6.2 or later.

For complete information on managing Logger Event Archives, see the Logger Administrator's Guide.

The following parameters are shown on the Logger Event Archives list:

Parameter	Description
Peers	For Loggers, the number of peers of the Logger. To see the Logger's peers in detail, click the number shown.
Event Status	The status of a current archiving job, where status is one of the following values: • *Loading:* The archive is being loaded on the managed Logger. • *Loaded*: The archive is currently loaded on the managed Logger. • *Unloading*: The archiving job is currently executing. • *Archived*: The archiving job is complete. • *Failed*: The archiving job was not successful.
Index Status	The status of a current indexing job, where status is one of the following values. • *None:* No indexing status is available. • *Pending*: The indexing job is about to begin. A pending job can be canceled by clicking in the **Cancel** column of the table. • *Indexing*: The indexing job is in process. • *Indexed*: The indexing job is complete. • *Failed*: The indexing job was unsuccessful.
Cancel	Click the **X** to cancel a pending indexing job before it begins.

To view Logger event archives:

1. Under **Configuration Management**, select **Logger Event Archive.**
2. On the **Event Archive List** tab, select your criteria to search for Logger Event Archives on managed Loggers.
3. Select a Start and End Date, then select one or more Loggers to search.
4. Click **Search**. All Logger Event Archives matching the search criteria are listed in hierarchical format: by managed Logger, then by Storage Group, and finally by Event Archive.

To toggle the view open or closed, click **Expand** or **Collapse**.

Managing Event Archives

You can perform two management tasks on managed Loggers related to event archives: loading (or unloading) archives, and indexing them.

To load an event archive:

1. On the Event Archive List, select an archive to load.
2. Click **Load Archive**. The selected operation will be performed. The status of the job will be shown in the **Event Status** column.

To index an Event Archive:

1. On the Event Archive List, select an archive to index.
2. Click **Index Archive**. The selected archive will be indexed. The status of the indexing job will be shown in the **Index Status** column.

Viewing Load/Unload History

You can also view your Logger event archive load, unload, and indexing history. This displays the actions taken in ArcMC to view Logger Event Archives.

To view Logger event archive load/unload history:

1. Under **Configuration Management**, select **Initial Configurations > Logger Event Archive.**
2. Click the **Archive Load/Unload History** tab. The activity history is displayed.

Managing Logger Peers

Managed Loggers can be peered with any number of other Loggers. You can manage the peer relationship between Loggers in ArcMC. ArcSight recommends that, if possible, all peer Loggers be managed by ArcMC.

You can view peers; add or remove peers to a Logger; and import, edit, push, and delete peer groups. A *peer group* is a named set of Loggers you can use to organize and administer sets of Loggers more easily.

Note: For more information about Logger peering, please refer to the ArcSight Logger Administrator's Guide.

Viewing Peers or Peer Groups

You can view the peers of a Logger managed by ArcMC, as long as the Logger is version 6.1 or later.

To view peered Loggers in ArcMC:

1. Select **Configuration Management > Logger Peers.** The **Logger Peers** table is displayed with all managed Loggers of version 6.1 or later.
2. To view the Loggers peered to a specific Logger in the list , in the **Peer Loggers** column, click the link indicating the number of peers. The filterable **Peer Loggers** dialog lists all the

Logger's peers.

3. To view peer groups in ArcMC, click **View Peer Groups.**

Adding or Removing Peers

You can add peers to, or remove peers from, a Logger managed by ArcMC, as long as the managed Logger is version 6.1 or later.

Note: If you remove a Logger not managed by ArcMC as a peer, you will not be able to add it back to the group unless you import the peer group including the Logger into ArcMC, or you add the removed Logger to ArcMC management.

To add peers to, or remove peers from, a Logger:

1. Select the Logger whose peers you wish to edit from the **Logger Peers** table.
2. Click **Edit Peers.**
3. All currently peered Loggers are shown.
   a. To add one or more peers, click **Add Peers**. Then, in the **Add Peers** dialog, select the Loggers to be added as peers. Optionally, to create a new peer group in ArcMC, in **Peer Group Name**, specify a name for the peer group. Then, click **Add**.
   b. To remove one or more Loggers as peers, select the Loggers to remove, and click **Remove Peers.** Click **Yes** to confirm removal as peers.

Note: For this release, Logger peering is supported using user name and password, not authorization code.

Importing a Peer Group

You can import Logger peer groups into ArcMC. Importing a peer group is only supported on Loggers version 6.1 or later.

To import a peer group from a Logger (of version 6.1 or later):

1. Select **Configuration Management > Logger Peers.**
2. Click **View Peer Groups.**
3. Click **Import Peers.**
4. On the **Select Peer** dialog, select a managed Logger. (The selected Logger will also be part of the imported peer group.) Then, click **Next**.
5. On the **Select Peer (of the Target)** dialog, select one or more peers to import into ArcMC.

6. In **Peer Group Name**, specify a name for the selected peer group.
7. Click **Import**. The selected peer group is imported into ArcMC.

Edit a Peer Group

You can edit a peer group, including the name, peered Logger hostname, and group members.

To edit a peer group:

1. Select**Configuration Management > Logger Peers.**
2. Click **View Peer Groups.**
3. Click the name of the peer group you wish to edit.
4. On the **Edit Peer Group** dialog, edit the peer group as needed. You can edit the peer group name, and add or remove peers from the group.
5. Click **Save**. Alternatively, click **Save As...** to save the peer group under a new name.

Pushing a Peer Group

You can push a peer group to one or multiple managed Loggers of version 6.1 or later. The Loggers in the group will become peered with the managed Loggers to which you pushed the group.

To push a peer group:

1. Click **Configuration Management > Logger Peers.**
2. Click **View Peer Groups.**
3. From the table, select a peer group to push.
4. Click **Push**.
5. On the **Destination Loggers** dialog, select one or more destination Loggers to which to push the peer group.
6. Click **Push**. The peer group is pushed to the destination Loggers.

Deleting a Peer Group

You can delete a peer group from ArcMC.

To delete a peer group:

1. Click **Configuration Management > Logger Peers.**
2. Click **View Peer Group.**

3. From the list of peer groups, select a group to delete.
4. Click **OK** to confirm deletion.

Managing Transformation Hub

You can use ArcMC to perform management and monitoring of Transformation Hub. These functions include adding topics, managing routes, and status monitoring.

About Topics

A *topic* is a metadata tag that you can apply to events in order to categorize them. Transformation Hub ships with several pre-set topics, and you can define any number of additional topics as needed.

A topic includes these components:

- **Name:** The name of the topic.

  Note: ArcSight Avro is the displayed name for the type name event-avro for the ArcSight avro topic.

- **Topic Type:** The type of topic CEF (routable) Arcsight Avro (routable) BINARY (not routable) RAW (not routable) SYSLOG (not routable).
- **Partition Count:** A segment of a topic. There can be one or more partitions for each topic. The number of partitions limits the maximum number of consumers in a consumer group.
- **Replication Factor:** The number of copies of each partition in a topic. Each replica is created across one Transformation Hub node. For example, a topic with a replication factor of 3 would have 3 copies of each of its partitions, across 3 Transformation Hub nodes.

You can only use ArcMC to add topics (not delete them). The **Edit** option is only available for topics with a *null* topic type (topics not created by ArcMC. e.g. Kafka manager) and it allows the user to modify the **Topic Type** value.

To set the type for existing topics (only for topics not created by ArcMC. e.g. Kafka manager), users can access the **List of Topics** page located in **Configuration Management > Transformation Hub > Topics.** This page will display detailed information for Topic Name, Topic Type, Routable Topic, Partitions Count, and Replication Factor. This option is only available for Transformation Hub 3.4+.

For more information on managing topic partitions and replication, please see "Managing Topics" on page 812.

Adding a Topic

To add a topic:

1. Click **Configuration Management > Transformation Hub.**
2. On the Transformation Hub Configurations page, click **Topics > + Add.**
3. On the Add New Topic dialog, in **Topic Name**, specify a name for the new topic.
4. In **Topic Type**, select the type for the new topic.

> **Note:** For Transformation Hub 3.4 users must select the topic type when adding the new topic. This option is disabled for Transformation Hub 3.3 or earlier.

5. In **# of Partitions,** specify the number of partitions the topic will have.
6. In **Replication Factor**, specify the number of copies that will be made for each partition.
7. Click **Save**.

> **Best Practice:** When creating a topic, use a value for replication factor of at least 2. In addition, the number of partitions should be equal to the number of consumers which will be subscribed to the topic (now and in future). If ArcSight Database will be a consumer, the number of partitions should be a multiple of the number of Database nodes.

About Routes

A *route* is a method of retrieving events in a topic that meet certain criteria and then copying them into a new topic. Use routes to filter events into your topics for your own requirements, such as selecting a group of events for more detailed examination.

A route comprises these components:

- **Name:** Name of the route.
- **Routing Rule:** A logical filter that defines criteria by which events will be categorized into topics. The criteria are defined in terms of CEF and Avro fields for Transformation Hub 3.4 and later, and CEF only for Transformation Hub 3.3 and earlier.
- **Source Topic:** The topic being filtered for events which match the routing rule.
- **Destination Topic:** The topic to which a copy of an event matching the routing rule should be copied. (A copy of the event will remain in the source topic.)
- **Description:** A short description of the route.

You can add, edit, or delete routes in ArcMC. Routes only apply to CEF and Avro topics for Transformation Hub 3.4 and later, and CEF only for Transformation Hub 3.3 and earlier. Routes created to or from a binary topic (such as th-binary_esm) will not function.

Every Avro routing created in Transformation Hub 3.4 and earlier using th-arcsight-avro as source topic will be automatically overridden after upgrading to Transformation Hub 3.5. As a general guideline, th-arcsight-avro is no longer recommended as a source topic for Avro routing, since enrichment stream processors were added as intermediate layer between th-arcsight-avro and mf-event-avro-enriched in Transformation Hub 3.5 . This makes the mf-event-avro-enriched topic the new primary source topic for Recon's database scheduler (replacing th-arcsight-avro). As a result, the routing starting point should start from the mf-event-avro-enriched topic to benefit from event enrichment.

Creating a Route

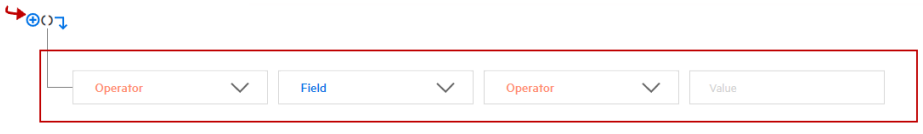
Before creating a route, ensure that your source and destination topics already exist. If not, create them before creating a route that uses them.

To create a route:

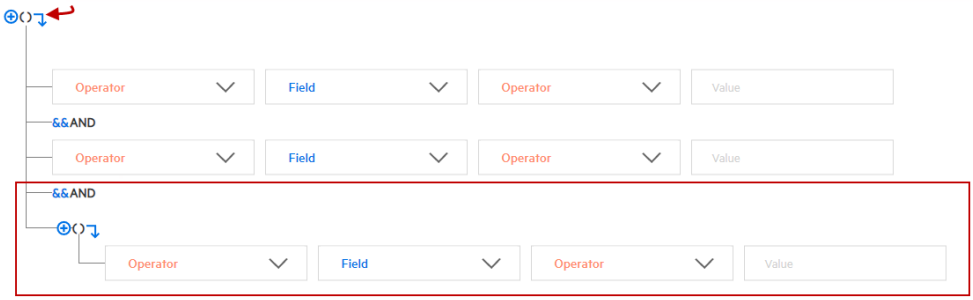
1. Click **Configuration Management > Transformation Hub.**
2. On the Transformation Hub Configurations page, click **Routes > +Add**.
3. In **Route Name**, specify a name for the route.
4. From the **Source Topic** drop-down list, select the topic from which events will be filtered.
5. From the **Destination Topic** drop-down list, select the destination to which events will be copied.
6. In **Description**, specify a short description of the route.
7. Under **Add Routing Rule**, use the Rule Editor to define the criteria for the routing rule.

Note: Only routable topics are displayed in the drop-down list for both **Source Topic** and **Destination Topic** when adding a new route. This option is only available for Transformation Hub 3.4.

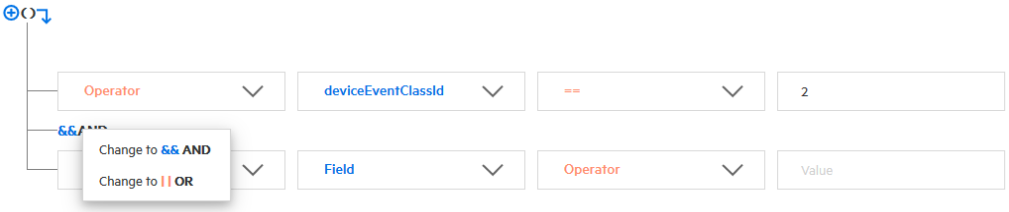
- Define a criterion by using the drop-downs to select a **Field, Operator,** and **Value** as a filter. **Fields** and **Operators** are based on the **Source Topic** type.
- Click + to add a new conjunction (& AND, || OR), or the right arrow to add a dependent conjunction. Then define any new required criteria as needed.



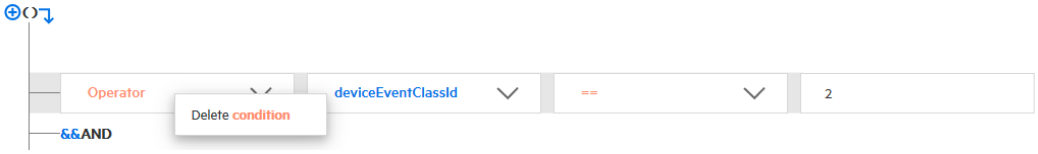
- You can create nested conjunctions by clicking the right arrow at the same level as the current conjunction.



- To change a conjunction, right-click the conjunction and select your choice from the drop-down menu.



- To delete a conjunction, right-click the conjunction and pick Delete. Note that deleting a conjunction will delete all the criteria associated with the deleted conjunction.



Note: If users have more than two source topics in routing, they need to increase service group through the CDF UI for the routing configuration. For more information please see the Administrator's Guide for the ArcSight Platform.

The rule is shown in the rule field as you construct it. When the rule is complete, click **Save**.

Editing a Route

To edit a route:

1. Click **Configuration Management > Transformation Hub > Routes.**
2. On the Transformation Hub Configurations page, select the route to edit, then click **Edit**.
3. Edit the route as needed, then click **Save**.

Deleting a Route

To delete a route:

1. Click **Configuration Management > Transformation Hub > Routes.**
2. On the Transformation Hub Configurations page, select one or more routes to delete, then click **Delete**.
3. Click **Yes** to confirm deletion.

Deployment Templates

A *deployment template* is a pre-set collection of settings and parameters for a connector or Collector.

When you deploy that connector or Collector type using the Instant Connector Deployment process, and specify a deployment template, all of the settings you have predefined in the template are applied during the deployment.

You may specify any number of deployment templates for each connector type.

Note: During the deployment process, you are prompted to use the predefined template settings, but may choose to overwrite any of the predefined template settings to custom-fit a particular deployment.

Managing Deployment Templates

You should be familiar with the settings for connectors and Collectors before managing deployment templates. These settings are described in detail in the Smart Connector User's Guide.

Prior to managing any deployment templates, first upload the appropriate 64-bit connector or Collector installer file to your ArcMC repository. Only the Linux and Windows 64-bit installers are supported. The installer contains a list of currently supported connectors or Collectors and is used in the creation of the connector or Collector list in ArcMC. This upload only needs to be done in preparation to manage deployment templates.

To upload the installer file to ArcMC:

1. Download the connector or Collector installer file to a secure network location.
2. In ArcMC, click **Administration > Application > Repositories.**
3. In the navigation menu, click **Upgrade Files.**

4. Click **Upload**.
5. Under **Upload Upgrade Repository**, click **Choose File.** Then, browse to and select the installer file you previously downloaded.
6. Click **Submit**. The installer file is uploaded to ArcMC.

Additional Files

Note that some connector types may require additional, supplementary files to function correctly, such as Windows DLLs. Such files are not included in the connector installer file.

If additional files are required for a connector type, you must also upload these files to an ArcMC repository before attempting to deploy them using the Instant Connector Deployment process. After uploading the installer file as described, upload additional files (in ZIP format) to the following repositories:

File Type	Repository
SecureData server certificate (Certificate_FPE)	cacert. **Note:** The certificate must be Base 64 encoded. For Linux plarforms (only), it must include the .pem extension.
Windows DLL, JavaLibrary	JDBC Drivers
FlexParsers	Flex Connectors

You will be able to specify the location of these additional files when you create the deployment template.

To create a deployment template:

Click **Configuration Management > Deployment Templates.**

1. In the navigation menu, from the list of supported connectors or Collectors, select the type of connector/Collector for which you wish to create a template.
2. In the management panel, click **New**.
3. To clone a template from an existing template of the same type, click **+ New/Clone**.

   To clone a template, select one from the **Copy from** dropdown and the values are populated based on the selected template instance.
4. Specify values for any required settings (marked with an asterisk *), as well as any settings you wish to apply to all connectors or Collectors of that type when using Instant Connector Deployment. (**Note:** Spaces in file or path names are not supported.)
5. If additional files are needed for operation, such as a Voltage server certificate, under **File Table Fields**, specify values for file name, type, and any other required fields. If more than 1 additional file is needed, click **Add Row**, then specify the details of the additional file.

Repeat for additional files as needed.

6. Click **Save**.

ArcSight SecureData Add-On Enablement: To enable the ArcSight SecureData Add-on during deployment, under **Global Fields**, set **Format Preserving Encryption** to *Enabled*. Note that only a single instance of the add-on is supported on Windows clients. If you wish to move the add-on to a new location, you must first uninstall the previously installed client before launching Instant Connector Deployment.

To delete a deployment template:

1. In the navigation menu, browse to the template you wish to delete. (Templates are sorted by connector/Collector type.)
2. In the management panel, select the template and click **Delete**. Click **Yes** to confirm deletion.

Bulk Operations

The following topics are discussed here.

Location

The **Location** tab displays all locations defined in Arcsight Management Center. The **Location** tab includes these buttons:

Add	Adds a new location. For more information, see "Adding a Location" on page 919
Edit	Edits the name of a location. For more information, see "Editing a Location" on page 919
Delete	Deletes one or more selected locations from ArcMC. For more information, see "Deleting a Location" on page 920

The **Manage Locations** table displays these parameters for each location.

- **Name:** Location name.
- **Number of Hosts:** Number of hosts assigned to the location.

Host Tab

The Host tab displays all hosts associated with the location selected in the navigation tree. The Hosts tab includes these buttons:

Update Credentials	Updates the host's credentials. For more information, see "Updating Host Credentials" on page 1052
Download Certificate	Downloads the host's current certificates. For more information, see "Downloading and Importing Host Certificates" on page 1052
Scan Host	Scans each port on non-appliance based hosts. For more information, see "Scanning a Host" on page 1053
Move	Moves selected hosts to a new location. For more information, see """Moving a Host to a Different Location" on page 1055
Delete	Deletes selected hosts from ArcMC. For more information, see "Deleting a Host" on page 1055

The **Hosts** table displays these parameters for each host:

- **Hostname:** Fully qualified domain name (FQDN) or IP address of the host. The hostname must match the hostname in the host's SSL certificate. (If IP address was used to add the host, then the certificate will match the IP address used.)
- **Path:** Path to the host.
- **Issues:** Status of any issues associated with the host. Possible indicators include:
  - *None:* No issues are associated with the host.
  - *Internet connection Not Present:* The host is currently not reachable by internet connection. Displayed when ArcMC is not able to connect to the Marketplace for retrieving parser upgrade versions. If the user environment needs a proxy server for an internet connection, configure the logger.properties file. If the user environment is an appliance, save the DNS settings on the **System Admin > Network** page.
  - *Valid Marketplace Certificate Not Found in ArcMC*: Displayed when the Marketplace certificate does not match the one found in ArcMC's trust store.
  - *Host Certificate Mismatch:* The hostname does not match the hostname in the SSL certificate. For instructions on downloading and importing certificates for the host, see "Downloading and Importing Host Certificates" on page 1052.
  - *Error in REST Authentication*: The Transformation Hub node lacks the ArcMC certificate, ArcMC session ID, or ArcMC URL and port. To resolve this issue:
    - Make sure that the CDF Cluster has been configured correctly with the appropriate ArcMC details. For more information, please see "Configuring ArcMC to Manage a Transformation Hub" on page 471.
    - Note that each time the user replaces the ArcMC certificate to the TH's location, the TH's webservice pod has to be restarted for the new certificate to be read and updated in the trust store.
- **Model:** If the host is an appliance, this shows the ArcSight model number of the appliance. If the host is not an appliance, the label *Software* is shown.

- **Type:** Type of installation, either ArcMC Appliance or Software.
- **Version:**Version number of the software on the host.

Container Tab

The Containers tab includes the **Properties** button, it allows you to modify the properties of Containers.

The **Containers** table includes the following columns:

- **Name:** Name of the container.
- **Path:** Path to the container.
- **Issues:** Status of any issues associated with the container.
- **Port:** Port number through which the container is communicating.
- **Framework Ver:** Framework version number of the container.
- **Parser Ver:** Parser version number of the container.
- **Status:** Status of the container. Possible values for container status are:
  - *Improper configuration: Initial default state.*
  - *Initializing connection:* The connector has a resolvable URL, but Arcsight Management Center has not logged in to the connector yet.
  - *Down*: There was an exception trying to execute the login command.
  - *Unauthorized:* The login command was executed, but login has failed.
  - *Connecting:* The login is in progress.
  - *Connected:* The login was successful.
  - *Empty:* Login successful, but the container doesn't have connectors.
  - *Initialized:* Login successful and the container has connectors.
  - *Unknown:* No information on status. To resolve, manually SSH to the system and restart the container.
- **Last Check:** Date and time of last status check.

Collector Tab

The **Collector** tab displays all Collectors associated with the item selected in the navigation tree. For example, if you selected a host in the navigation tree, the **Collectors** tab would show all Collectors associated with that host.

A Collector is a standalone System component in charge of processing efficiency improvements and the collection of raw data.

Note: The maximum number of selected entries when managing Connectors/Collectors is 50.

The **Collectors** tab includes the following buttons, which operates on one or more selected Collectors:

Properties	Update the properties of the selected Collectors. For more information, see "Updating Collector Properties" on the next page
Retrieve Logs	Retrieves Collector logs. For more information, see "Retrieving Collector Logs" on the next page
Update Parameters	Update the parameters of the selected Collectors. For more information, see "Updating Collectors Parameters" on the next page
Destinations	Manage Collector destinations. For more information, see "Updating Collector Destinations" on page 1047
Credential	Manage Collector credentials. For more information on managing Collector credentials, see "Updating Collector Credentials" on page 1047
Restart	Restart the selected Collectors. For more information on restarting Collectors, see "Restarting Collectors" on page 1047.
Delete	Deletes the selected Collectors. For more information, see "Deleting Collectors" on page 1048

The **Collectors** table displays the following parameters for each connector:

- **Name:** Name of the Collector.
- **Port:** Collector port.
- **Type:** Type of Collector.
- **Syslog Lines Received:** Number of events received.
- **Custom Filtering:** messages filtered out.
- **Status:** Collector status.
- **Version:** Collector version.
- **Last Check:** Date and time of the last status check.

Transformation Hub Tab

The **Transformation Hub** table includes the following columns:

- **Transformation Hub:** Name of the Transformation Hub.
- **Host:** Name of the host.
- **Port:** Port number through which the Transformation Hub is communicating.
- **Last Check:** Date and time of the last status check.

For more information on connector management, see "Managing Connectors" on page 964

Updating Collector Properties

To update Collector properties:

1. Click **Configuration Management > Bulk Operations**.
2. On the **Manage Collectors** page, select the item you wish to manage.
3. Click **Properties**.
4. On the **Property Update** page, click **Edit**
5. Edit the Collector properties as needed.
6. To add a new property, specify the property, a value for the property, and click the check mark.
7. When complete, click **Save**.

Retrieving Collector Logs

To retrieve Collector logs:

1. Click **Configuration Management > Bulk Operations**.
2. On the **Bulk Opeartions** page, select one or more items for which you wish to retrieve logs.
3. Click **Retrieve Logs**.
4. Follow the wizard prompts to zip the selected logs into a single file.
5. To view the logs, on the main menu bar, click **Admin > Repositories.** The log zip file is stored in the *Logs* respository.

Updating Collectors Parameters

To update Collector parameters:

1. Click **Configuration Management > Bulk Operations**.
2. On the **Manage Collectors** page, select one or more items for which you wish to update parameters.
3. Click **Update Parameters**.
4. On the **Collector Parameter Update** page, specify values for the parameters, as needed.
5. Click **Save**. The parameters are updated for the selected items.

Updating Collector Destinations

To update Collector destinations:

1. Click **Configuration Management > Bulk Operations**.
2. On the **Manage Collectors** page, select one or more items for which you wish to update destinations.
3. Click **Update Destinations**.
4. On the **Collector Destination Update** page, specify values for the destinations, as needed.
5. Click **Save**. The destinations are updated for the selected items.

Updating Collector Credentials

To update Collector credentials:

1. Click **Configuration Management > Bulk Operations**.
2. On the **Manage Collectors** page, select one or more items for which you wish to update credentials.
3. Click **Credential**.
4. On the **Collector Credential Update** page, specify values for passwords, as needed. (The username is fixed as *collector*.)
5. Click **Save**. The passwords are updated for the selected Collectors.

Note: Updating Collector credentials from ArcMC does not update the actual credentials, just the credentials ArcMC uses to authenticate.

Restarting Collectors

To restart one or more Collectors:

1. Click **Configuration Management > Bulk Operations.**
2. On the **Manage Collectors** page, select one or more items which you wish to restart.
3. Click **Restart.**
4. Click **Yes** to confirm restart. The Collectors are restarted.

Deleting Collectors

To delete Collectors:

1. Click **Configuration Management > Bulk Operations**.
2. On the **Manage Collectors** page, select one or more items which you wish to delete.
3. Click **Delete**.
4. Click **Yes** to confirm delete. The items are deleted.

Enabling SecureData Encryption on Managed Connectors

SecureData can be enabled as part of the Instant Connector Deployment process. However, you can also enable SecureData encryption on connectors or containers you already manage in ArcMC.

To enable SecureData encryption on connectors or containers you already manage in ArcMC:

1. Ensure that the remote VM can communicate with the SecureData server. If not, edit the hosts file or configure DNS to enable communication.
2. If there is a certificate for the SecureData server, make sure it is successfully imported to the remote VM.
3. Ensure proxy settings allow the SecureData client to communicate with the SecureData server.
4. Install the SecureData client manually on the remote VM where the connectors reside.
5. Finally, in ArcMC, select the connectors or containers. Perform the Modify Property operation and provide the necessary SecureData and proxy details.

Prerequisites for Addition of SecureData Client to Multiple Containers

The following are prerequisites for the addition of the SecureData client to multiple containers.

- The process should be performed by an account with which the Connector was installed.

  Note: If this user was a non-root user, that user must have access to the directory on the destination host with all permissions.

  The process must have a dedicated port numbered higher than 1024.

Bulk SecureData client install is supported for accounts using SSH key authentication, but not supported for SSH with passphase authentication. To enable SSH key authentication, the SSH key needs to be set up between a non-root user of ArcMC and a user of the remote host.

- You should consult and review the Format Preserving Encryption Environment Setup Guide for proxy settings.
- All the selected container host machines need to have same SSH credentials (username:password).
- The voltage client install path on all the selected containers hosts must be the same.
- You can only push voltage client in bulk to all the container hosts that are on the same platform e.g. all Linux, or all Windows.
- The below prerequisites are not present by default on RHEL/CentOS 8.x, unlike in previous RHEL/CentOS versions (e.g. RHEL/CentOS 6.x and 7.x). Perform the following steps for RHEL/CentOS 8.1 on the machine where the ArcMC is or will be installed, and in the target RHEL/CentOS host (the VM where the Connector/Collector will be deployed):

  a. Install python2:

  For RHEL/CentOS 7.x:
  ```
  sudo yum install -y python2
  ```
  For RHEL/CentOS 8.x:
  ```
  sudo dnf install -y python2
  ```

  b. Create a symlink:

  ```
  sudo ln -s /usr/bin/python2 /usr/bin/python
  ```

  c. Install libselinux-python package:

  For RHEL/CentOS 7.x:
  ```
  sudo yum install -y libselinux-python
  ```
  For RHEL/CentOS 8.x:
  ```
  sudo dnf install -y libselinux-python
  ```

  Note: If the `yum/dnf` command fails when installing `libselinux-python` on RHEL/CentOS, follow the steps below:
  - Download `libselinux-python-2.8-6.module_el8.0.0+111+16bc5e61.x86_64.rpm`
  - Install the package:
  ```
  rpm -i libselinux-python-2.8-6.module_el8.0.0+111+16bc5e61.x86_64.rpm
  ```

Additional Requirements For Windows Platforms

For Windows platforms, only the local admin account is supported for the bulk-addition of the SecureData client.

In addition, the following preparatory steps are required when deploying on a Windows VM.

1. Enable PowerShell 4.0 or later.

https://www.microsoft.com/en-us/download/details.aspx?id=40855

2. Enable and configure PowerShell Remoting, with CredSSP authentication.

- Download the "ConfigureRemotingForAnsible.ps1" file:
  - https://github.com/ansible/ansible/blob/devel/examples/scripts/ConfigureRemotingForAnsible.ps1
- Open Power Shell as Administrator and run the following command:
  - `ConfigureRemotingForAnsible.ps1 -EnableCredSSP`

3. Enable TLS 1.2.

Adding SecureData to Multiple Containers

You can add the SecureData encryption client to multiple containers at once. The following limitations apply:

- The selected containers must meet all prerequisites for adding SecureData.
- All selected container hosts must have the same user credentials (username and password), and must be the same platform (that is, all Windows or all Linux.)
- The SecureData client installation path on all container hosts will be the same.
- If a certificate is needed, upload the required certificate before proceeding to **Repositories > CA Certs.**

Note: CTHs cannot be configured with SecureData encryption.

To add SecureData encyption to multiple containers:

1. Click **Configuration Management > Bulk Operations**
2. On the **Container** tab, select the containers to which you wish to add SecureData encryption.
3. Click **Properties**.
4. On the Container Property Update dialog, click **Edit**.
5. in the **Property List** column, click the **Settings** icon, then search for any values with `fpe` in the name. Change or specify values for these properties as follows.

Property	Description
`fpencryption.enabled`	If true, SecureData (Format Preserving) Encryption is enabled. Once enabled, encryption parameters cannot be modified. A fresh installation of the connector will be required to make any changes to encryption parameters.
`fpencryption.host.url`	URL of the SecureData server

Property	Description
`https.proxy.host`	Proxy SecureData server (https)
`https.proxy.port`	Proxy port
`fpencryption.user.identity`	SecureData identity
`fpencryption.shared.secret`	SecureData shared secret
`fpencryption.event.fields`	Comma-separated list of fields to encrypt.
`fpencryption.voltage.installdir`	Absolute path where the SecureData client needs to be installed

6. Select Install SecureData Client.
7. To use SSH key-based authentication to Linux container hosts (only), select **SSH Key.**

Note: SSH key applies to Linux hosts only. If the SSH Key check box is selected for Windows hosts, the update will fail.

8. If needed, from the **SecureData Cert** drop-down, select a previously-uploaded certificate for SecureData.
9. In **Username** and **Password**, specify the common user credentials for all selected container hosts. (Password is not needed if SSH is enabled in Step 7.)
10. Click **Save.**

The SecureData client is pushed to the selected containers, and each one is restarted. To see if the encryption properties were updated successfully, wait on this page. The Job Manager shows the status of client installation on the containers.

Updating Transformation Hub Cluster Details in ArcMC

When you upgrade the Transformation Hub cluster to the latest version, and if you choose to manage the whole cluster with ArcMC, you need to update the cluster details in ArcMC. Doing this enables you to deploy CTHs on the latest version of the Transformation Hub cluster.

Note: Make sure that the **Cluster Username**, **Cluster Password**, and **Certificate** information, correspond to the upgraded version of the Transformation Hub.

To update Transformation Hub Cluster Details in ArcMC:

1. Click **Configuration Management > Bulk Operations**
2. Click the **Host** tab.
3. Select the Transformation Hub host.
4. Click **Update Cluster Details.**

5. In the **Hostname** field, type the fully qualified name of the TH.
6. In the **Cluster Port** field, type **443**.
7. In the **Cluster Username** field type the TH username.
8. In the **Cluster Password** field type the TH password.
9. SSH to the Transformation Hub and go to: `/opt/arcsight/kubernetes/scripts/`
10. Run the following script to generate the certificate: `cdf-updateRE.sh`
11. Copy the certificate name ca.crt (be sure to include from ----- BEGIN CERT to END CERT -----), navigate to the GUI, paste it on the **Cluster Certificate** field and click **Save.**

Updating Host Credentials

ArcSight Platform relies on a host's login credentials to connect and authenticate to the managed host. You specify these credentials when adding the host to ArcSight Platform for management. If these credentials ever change, the management link between ArcSight Platform and the host will be broken.

However, you can update the credentials ArcSight Platform uses to authenticate to a managed host, which will prevent the management link from being broken.

Updating host credentials on ArcSight Platform does not change the actual credentials on the managed host. You will need to change those on the host directly, either immediately before or immediately after performing this operation. Updating credentials will only update the credentials that ArcSight Platform uses to authenticate to the host.

To update host credentials:

1. Click **Configuration Management > Bulk Operations**.
2. Click the **Host** tab.
3. Select the host you want to update, click **Update Credentials**.
4. In **Username** and **Password**, specify the new credentials that ArcSight Platform will use to connect to the host.
5. Click **Save.**

Downloading and Importing Host Certificates

In case of a mismatch between the hostname and the hostname in the SSL certificate, you can download and import the host’s current certificates.

To download and import host certificates:

1. Click **Configuration Management > Bulk Operations**.
2. Click the **Hosts** tab.
3. Select the desired host.
4. Click **Download Certificate**.
5. Click **Import** in the wizard and then Click **Done**.

Scanning a Host

Scanning a host will inventory all currently running containers on the host and the connectors associated with them.

To ensure accuracy and currency of container inventory, you will need to manually scan for new containers in any of the following circumstances:

- Additional containers or connectors are added to a remote host after it has been added to Arcsight Management Center.
- Containers and connectors are removed from a remote host managed in Arcsight Management Center.
- Any containers which were down when the initial, automatic scan was performed have since come back up.
- The license for a managed ArcSight Management Center (managed by another ArcSight Management Center) is upgraded to increase the number of licensed containers.

Any host that includes containers is scanned automatically when first added to ArcSight Management Center.

You can manually scan any host types that can run containers. These types include:

- Connector Appliances
- Loggers (L3XXX models only)
- ArcSight Management Center Appliances
- Connectors

The Scan Process

A host scan retrieves information on all CA certificates from any running containers on the host. The containers on the remote host can be managed only if Arcsight Management Center can authenticate using the certificates and the credentials. You are prompted to import any retrieved certificates into the Arcsight Management Center trust store.

A manual scan will be discontinued if any of the following are true:

- Any containers on a scanned Connector Appliance host are down.
- If you choose *not* to import any certificates that are retrieved.
- Authentication fails on any of the containers.

Note: When a Collector and connector are intended to run on the same host, add the Collector to ArcMC first, before the connector. Then perform a scan host to correctly detect the connector.

To manually scan a host:

1. Click **Configuration Management > Bulk Operations**.
2. In the navigation tree, select the location to which the host has been assigned.
3. Click the **Host** tab.
4. Select the host you want to scan, click **Scan Host**. The Host Scan wizard starts.
5. Specify values for the parameters in the following table, then click **Next**.

Parameter	Description
Starting Port	The port number on the host on which Arcsight Management Center starts scanning for containers.
Ending Port	The port number on the host on which Arcsight Management Center ends scanning for containers.
Connector Username	The Connector user name to authenticate with the host.
Connector Password	The password for the Connector you provided.
Collector Username	The Collector user name to authenticate with the host.
Collector Password	The password for the Collector you provided.

6. Connector certificates are retrieved automatically so that the Arcsight Management Center can communicate with each connector in a container. The Host Scan wizard lists the certificates. (To see certificate details, hover over the certificate.)
   - To continue the scan, select **Import the certificates**, then click **Next** to import the certificates and continue.
   - Otherwise, select **Do not import the certificates,** and then click **Next**. The Host Scan wizard discontinues the scan.

Moving a Host to a Different Location

You can assign one or more hosts to a new location. When you move a host, any nodes associated with it are also moved. For example, if you moved a Connector Appliance to a new location, all of its containers and managed connectors would also be moved to the new location.

To move one or more hosts:

1. Click **Configuration Management > Bulk Operations**.
2. Click the **Hosts** tab.
3. Select one or more hosts to move.
4. Click **Move**.
5. Follow the prompts in the **Host Move** wizard.The selected hosts are reassigned to their new locations.

Deleting a Host

When you delete a host, any nodes associated with the host are also deleted. Deleting a host removes its entry from ArcSight Management Center, but otherwise leaves the host machine unaffected.

Note: Use caution when deleting a host. Deleting a host will delete its associated nodes from any node list, association, peers listing, or subscribers listing that includes those nodes.

To delete one or more hosts:

1. Click **Configuration Management > Bulk Operations.**.
2. Select one or more hosts to delete.
3. Click **Delete**.
4. Click **Yes** to confirm deletion. The host (and any associated nodes) are deleted.

Destination Runtime Parameters

The following table describes configurable destination parameters. The parameters listed in the table are not available for all destinations. The user interface automatically displays the parameters valid for a destination. For step-by-step instructions on updating the runtime parameters of a destination, see "Editing Connector Parameters" on page 967.

Parameter	Description
Batching	Connectors can batch events to increase performance and optimize network bandwidth. When activated, connectors create blocks of events and send them when they either (1) reach a certain size or (2) the time window expires, whichever occurs first. You can also prioritize batches by severity, forcing the connector to send the highest-severity event batches first and the lowest-severity event batches later.
Enable Batching (per event)	Create batches of events of this specified size (5, 10, 20, 50, **100**, 200, 300 events).
Enable Batching (in seconds)	The connector sends the events if this time window expires (1, 5, 10, 15, 30, 60).
Batch By	This is **Time Based** if the connector should send batches as they arrive (the default) or **Severity Based** if the connector should send batches based on severity (batches of Highest Severity events sent first).
Time Correction	The values you set for these fields establish forward and backward time limits, that if exceeded, cause the connector to automatically correct the time reported by the device.
Use Connector Time as Device Time	Override the time the device reports and instead use the time at which the connector received the event. This option assumes that the connector will be more likely to report the correct time. (**No** | Yes)
Enable Device Time Correction (in seconds)	The connector can adjust the time reported by the device `Detect Time`, using this setting. This is useful when a remote device's clock isn't synchronized with the ArcSight Manager. This should be a temporary setting. The recommended way to synchronize clocks between Manager and devices is the NTP protocol. The default is **0**.
Enable Connector Time Correction (in seconds)	The connector can also adjust the time reported by the connector itself, using this setting. This is for informational purposes only and allows you to modify the local time on the connector. This should be a temporary setting. The recommended way to synchronize clocks between Manager and connectors is the NTP protocol. The default is **0**.
Set Device Time Zone To	Ordinarily, it is presumed that the original device is reporting its time zone along with its time. And if not, it is then presumed that the connector is doing so. If this is not true, or the device isn't reporting correctly, you can switch this option from Disabled to GMT or to a particular world time zone. That zone is then applied to the time reported. Default: **Disabled**.
Device Time Auto-correction	
Future Threshold	The connector sends the internal alert if the detect time is greater than the connector time by `Past Threshold` seconds.
Past Threshold	The connector sends the internal alert if the detect time is earlier than the connector time by `Past Threshold` seconds.
Device List	A comma-separated list of the devices to which the thresholds apply. The default, `(ALL)`, means all devices.
Time Checking	These are the time span and frequency factors for doing device-time auto-correction.
Future Threshold	The number of seconds by which to extend the connector's forward threshold for time checking. The default is **5 minutes** (300 seconds).

Parameter	Description
Past Threshold	The number of seconds by which to extend the connector's rear threshold for time checking. Default is **1 hour** (3,600 seconds).
Frequency	The connector checks its future and past thresholds at intervals specified by this number of seconds. Default is **1 minute** (60 seconds).
Cache	Changing these settings will not affect the events cached, it will only affect new events sent to the cache.
Cache Size	Connectors use a compressed disk cache to hold large volumes of events when the ArcSight Manager is down or when the connector receives bursts of events. This parameter specifies the disk space to use. The default is **1 GB** which, depending on the connector, can hold about 15 million events, but it also can go down to **5 MB**. When this disk space is full, the connector drops the oldest events to free up disk cache space. (5 MB, 50 MB, 100 MB, 150 MB, 200 MB, 250 MB, 500 MB, 1 GB, 2.5 GB, 5 GB, 10 GB, 50 GB.)
Notification Threshold	The size of the cache's contents at which to trigger a notification. Default is **10,000**.
Notification Frequency	How often to send notifications after the Notification Threshold is reached. (1 minute, 5 minutes, **10 minutes**, 30 minutes, 60 minutes.)
Network	
Heartbeat Frequency	This setting controls how often the connector sends a heartbeat message to the destination. The default is **10 seconds**, but it can go from **5 seconds** to **10 minutes**. Note that the heartbeat is also used to communicate with the connector; therefore, if its frequency is set to **10 minutes**, then it could take as much as 10 minutes to send any configuration information or commands back to the connector.
Enable Name Resolution	The connector tries to resolve IP addresses to hostnames, and hostnames to IP addresses , if required and if the event rate allows. This setting controls this functionality. The Source, Target and Device IP addresses , and Hostnames might also be affected by this setting. By default, name resolution is enabled (**Yes**).
Name Resolution Host Name Only	Default: **Yes**.
Name Resolution Domain From E-mail	Default: **Yes**.
Clear Host Names Same as IP Addresses	Default: **Yes**.
Limit Bandwidth To	A list of bandwidth options you can use to constrain the connector's output over the network. (**Disabled**, 1 kbit/sec to 100 Mbits/sec.)

Parameter	Description
Transport Mode	You can configure the connector to cache to disk all the processed events it receives. This is equivalent to pausing the connector. However, you can use this setting to delay event-sending during particular time periods. For example, you could use this setting to cache events during the day and send them at night. You can also set the connector to cache all events, except for those marked with a very-high severity, during business hours, and send the rest at night. (**Normal** | Cache | Cache (but send Very High severity events).
Address-based Zone Population Defaults Enabled	This field applies to v3.0 ArcSight Managers. This field is not relevant in ESM v3.5 because the system has integral zone mapping. Default: **Yes**.
Address-based Zone Population	This field applies to v3.0 ArcSight Managers. This field is not relevant in ESM v3.5 because the system has integral zone mapping.
Customer URI	Applies the given customer URI to events emanating from the connector. Provided the customer resource exists, all customer fields are populated on the ArcSight Manager. If this particular connector is reporting data that might apply to more than one customer, you can use Velocity templates in this field to conditionally identify those customers.
Source Zone URI	Shows the URI of the zone associated with the connector's source address. (Required for ESM v3.0 compatibility.)
Source Translated Zone URI	Shows the URI of the zone associated with the connector's translated source address. The translation is presumed to be NAT. (Required for ESM v3.0 compatibility.)
Destination Zone URI	Shows the URI of the zone associated with the connector's destination address. (Required for ESM v3.0 compatibility.).
Destination Translated Zone URI	Shows the URI of the zone associated with the connector's translated destination address. The translation is presumed to be NAT. (Required for ESM v3.0 compatibility.)
Connector Zone URI	Shows the URI of the zone associated with the connector's address. (Required for ESM v3.0 compatibility.)
Connector Translated Zone URI	Shows the URI of the zone associated with the connector's translated address. The translation is presumed to be NAT. (Required for ESM v3.0 compatibility.)
Device Zone URI	Shows the URI of the zone associated with the device's address. (Required for ESM v3.0 compatibility.)
Device Translated Zone URI	Shows the URI of the zone associated with the device's translated address. The translation is presumed to be NAT. (Required for ESM v3.0 compatibility.)

Parameter	Description
Field Based Aggregation	This feature is an extension of basic connector aggregation. Basic aggregation aggregates two events if, and only if, all the fields of the two events are the same (the only difference being the detect time). However, field-based aggregation implements a less strict aggregation mechanism; two events are aggregated if only the selected fields are the same for both alerts. It is important to note that field-based aggregation creates a new alert that contains only the fields that were specified, so the rest of the fields are ignored. Connector aggregation significantly reduces the amount of data received, and should be applied only when you use less than the total amount of information the event offers. For example, you could enable field-based aggregation to aggregate “accepts” and “rejects” in a firewall, but you should use it only if you are interested in the count of these events, instead of all the information provided by the firewall.
Time Interval	Select a time interval, if applicable, to use as a basis for aggregating the events the connector collects. It is exclusive of Event Threshold. (**Disabled**, 1 sec, 5 sec, and so on, up to 1 hour.)
Event Threshold	Select a number of events, if applicable, to use as a basis for aggregating the events the connector collects. This is the maximum count of events that can be aggregated; for example, if 150 events were found to be the same within the time interval selected (that is, contained the same selected fields) and you select an event threshold of 100, you then receive two events, one of count 100 and another of count 50. This option is exclusive of Time Interval. (**Disabled**, 10 events, 50 events, and so on, up to 10,000 events.)
Field Names	Specify one or more fields, if applicable, to use as the basis for aggregating the events the connector collects. The result is a comma-separated list of fields to monitor. For example, "eventName,deviceHostName" would aggregate events if they have the same event- and device-hostnames. Names can contain no spaces and the first letter must not be capitalized.
Fields to Sum	Specify one or more fields, if applicable, to use as the basis for aggregating the events the connector collects.
Preserve Common Fields	Choosing Yes adds fields to the aggregated event if they have the same values for each event. Choosing **No**, the default, ignores non-aggregated fields in aggregated events.
Filter Aggregation	Filter Aggregation is a way of capturing aggregated event data from events that would otherwise be discarded due to an agent filter. Only events that would be filtered out are considered for filter aggregation (unlike Field-based aggregation, which looks at all events). Connector aggregation significantly reduces the amount of data received, and should be applied only when you use less than the total amount of information the event offers.
Time Interval	Select a time interval, if applicable, to use as a basis for aggregating the events the connector collects. It is exclusive of Event Threshold. (**Disabled**, 1 sec, 5 sec, and so on, up to 1 hour.)
Event Threshold	Select a number of events, if applicable, to use as a basis for aggregating the events the connector collects. This is the maximum count of events that can be aggregated; for example, if 150 events were found to be the same within the time interval selected (that is, contained the same selected fields) and you select an event threshold of 100, you then receive two events, one of count 100 and another of count 50. This option is exclusive of Time Interval. (**Disabled**, 10 events, 50 events, and so on, up to 10,000 events.)
Fields to Sum	(Optional) Select one or more fields, if applicable, to use as the basis for aggregating the events the connector collects.

Parameter	Description
Processing	
Preserve Raw Event	For some devices, a raw event can be captured as part of the generated alert. If that is not the case, most connectors can also produce a serialized version of the data stream that was parsed/processed to generate the ArcSight event. This feature allows the connector to preserve this serialized "raw event" as a field. This feature is disabled by default since using raw data increases the event size and therefore requires more database storage space. You can enable this by changing the **Preserve Raw Event** setting. The default is **No**. If you choose **Yes**, the serialized representation of the "Raw Event" is sent to the destination and preserved in the Raw Event field.
Turbo Mode	You can accelerate the transfer of a sensor's event information through connectors by choosing one of two “turbo” (narrower data bandwidth) modes. The default transfer mode is called **Complete**, which passes all the data arriving from the device, including any additional data (custom, or vendor-specific). **Complete** mode does indeed use all the database performance advances of ArcSight ESM v3.x. The first level of Turbo acceleration is called **Faster** and drops just additional data, while retaining all other information. The **Fastest** mode eliminates all but a core set of event attributes, in order to achieve the best throughput. The specific event attributes that apply to these modes in your enterprise are defined in the self-documented `$ARCSIGHT_HOME/config/connector/agent.properties` file for the ArcSight Manager. Because these properties might have been adjusted for your needs, you should refer to this file for definitive lists. Only scanner connectors need to run in **Complete** mode, to capture the additional data. **Note**: Connector Turbo Modes are superseded by the Turbo Mode in use by the ArcSight Managers processing their events. For example, a Manager set to **Faster** will not pass all the data possible for a connector that is set for the default of **Complete**.

Parameter	Description
Enable Aggregation (in seconds)	When enabled, aggregates two or more events on the basis of the selected time value. (**Disabled**, 1, 2, 3, 4, 5, 10, 30, 60) The aggregation is performed on one or more matches for a fixed subset of fields: • Agent ID • Name • Device event category • Agent severity • Destination address • Destination user ID • Destination port • Request URL • Source address • Source user ID • Source port • Destination process name • Transport protocol • Application protocol • Device inbound interface • Device outbound interface • Additional data (if any) • Base event IDs (if any) The aggregated event shows the event count (how many events were aggregated into the displayed event) and event type. The rest of the fields in the aggregated event take the values of the first event in the set of aggregated events.
Limit Event Processing Rate	You can moderate the connector's burden on the CPU by reducing its processing rate. This can also be a means of dealing with the effects of event bursts. The choices range from **Disabled** (no limitation on CPU demand) to **1 eps** (pass just one event per second, making the smallest demand on the CPU). Note: The effect of this option varies with the category of connector in use, as described in the connector Processing Categories table below.
Fields to Obfuscate	
Store Original Time in	**Disabled** or Flex Date 1.
Enable Port-Service Mapping	Default: **No**.
Enable User Name Splitting	Default: **No**.

Parameter	Description
Split File Name into Path and Name	Default: **No**.
Generate Unparsed Events	Default: **No**.
Preserve System Health Events	Yes, **No**, or Disabled.
Enable Device Status Monitoring (in minutes)	**Disabled** or 1, 2, 3, 4, 5, 10, 30, 60, or 120 minutes.
Filters	
Filter Out	NA
“Very High Severity” Event Definition	NA
“High Severity” Event Definition	NA
“Medium Severity” Event Definition	NA
“Low Severity” Event Definition	NA
“Unknown Severity” Event Definition	NA
Payload Sampling	(When available.)
Max. Length	Discard, 128 bytes, **256 bytes**, 512 bytes, 1 kbyte
Mask Non-Printable Characters	Default: **False**.

Special Connector Configurations

Certain connectors require additional configuration when used with Arcsight Management Center. This appendix describes the additional configuration. For general information about installing connectors, see "Adding a Connector" on page 964.

The following topics are discussed here:

Microsoft Windows Event Log - Unified Connectors

The SmartConnector for Microsoft Windows Event Log - Unified is not part of a FIPS-compliant solution. When you add a Windows Event Log - Unified connector, be sure the container is not FIPS-enabled in order for the connector to collect events.

When adding a Windows Event Log - Unified connector, follow the specific instructions in the SmartConnector configuration guide for entering parameters, entering security certifications when using SSL, enabling audit policies, and setting up standard user accounts.

There are currently two parser versions for the Microsoft Windows Event Log - Unified SmartConnector.

- Parser Version 0 is generally available with each SmartConnector release
- Parser Version 1 is available with the Microsoft Windows Monitoring content

The Microsoft Windows Event Log - Unified SmartConnector configured for you during initial configuration uses Parser Version 1.

Detailed Security Event mappings for this parser version can be found in Security Event Mappings: SmartConnectors for Microsoft Windows Event Log - Unified with Parser Version 1 (MSWindowsEventLogUnifiedMappingsParserVersion1.pdf), available on ArcSight Protect724.

When you install additional Microsoft Windows Event Log Unified connectors, they are installed with the generally available base parser version (Parser Version 0). Mappings for the base parser version are available with each SmartConnector release (Security Event Mappings: SmartConnectors for Microsoft Windows Event Log) and can be found on Protect724, along with the SmartConnector configuration guide. You must use Parser Version 1 if you want the default Windows Monitoring content to work. For details see the SmartConnector Configuration Guide for Microsoft Windows Event Log - Unified, or SmartConnector Configuration Guide for Microsoft Windows Security Events - Mappings.

Note: The pre-bundled SmartConnector for Microsoft Windows Event Log - Unified installed using the First Boot Wizard is installed with Parser Version 1. Any Windows Event Log - Unified connectors you add using the connector configuration wizard are installed with Parser Version 0 (the base parser).

Change Parser Version by Updating Container Properties

A parser is a SmartConnector component that specifies how to parse the information contained in the device raw events, and how to map it to ArcSight security event schema fields. Parsers can be in the form of property files, map files, or CSV files. Each SmartConnector has its own parser or set of parsers.

Multiple parser versions enables each SmartConnector parse raw events in many different ways to generate ArcSight security events with appropriate mappings. The SmartConnector for Microsoft Windows Event Log -- Unified, supports two parser versions: Base Parser and Parser Version 1.

With multiple parser versions:

- One SmartConnector build supports multiple parser versions.
- Users can configure their connectors to use the available parser versions of their choice, depending on their event mapping requirements.
- Users can reconfigure connectors to use the appropriate parser version as needed.

Multiple parser versions currently are supported only for the SmartConnector for Microsoft Windows Event Log -- Unified. This functionality is not supported for user-developed ArcSight FlexConnectors.

Each SmartConnector has its own internal `fcp.version` parameter setting to represent its current parser version. The default value for the fcp.version parameter is the base (or default) parser version, which is Parser Version 0. Each SmartConnector can support a total of 8 parser versions. The `fcp.version` parameter values range from 0 through 7. Microsoft Windows Unified SmartConnector supports parser versions 0 and 1.

Be sure that when you have content with new mappings, you change the parser version to match that content.

To update container properties (located in the agent.properties file) to change the parser version being used when mapping events:

1. Click **Manage** from the top-level menu bar.
2. Select a navigation path.
3. Select the container whose properties you want to update. You can select multiple containers.
4. Click **Properties**.
5. Follow the instructions in the wizard to update connector properties.

   The `fcp.version` parameter value 0 designates the base parser. To use parser 1, change the `fcp.version` parameter value to 1. For example:

   ```
   agents[0].fcp.version=1
   ```

SSL Authentication

If you choose to use SSL as the connection protocol, you must add security certificates for both the Windows Domain Controller Service and for the Active Directory Server. Installing a valid certificate on a domain controller permits the LDAP service to listen for, and automatically

accept, SSL connections for both LDAP and global catalog traffic. With the First Boot Wizard installation of the connector, the certificates are already imported for you. If you add Windows Event Log - Unified connectors, see the SmartConnector Configuration Guide for Microsoft Windows Event Log - Unified for instructions.

Database Connectors

The following database connectors are available for installation with ArcSight Express:

- IBM SiteProtector DB*
- McAfee ePolicy Orchestrator DB*
- McAfee Vulnerability Manager DB*
- McAfee Network Security Manager DB*
- Microsoft SQL Server Audit Multiple Instance DB*
- Oracle Audit DB
- Symantec Endpoint Protection DB*
- Trend Micro Control Manager NG DB*
- Snort DB*

*These connectors extract events from an SQL Server or MySQL databases, which requires a JDBC driver. See "Add a JDBC Driver" on the next page for instructions.

All of these database connectors require the following information when being added to ArcSight Express; some connectors require additional parameters, such as event types or polling frequency.

Parameter	Description
Database JDBC Driver	If you are using an ODBC DRIVER, select 'sun.jdbc.odbc.JdbcOdbcDriver' driver. For JDBC drivers, select the 'com.microsoft.sqlserver.jdbc.SQLServerDriver' driver.
Database URL	If you are using an ODBC DRIVER, enter: 'jdbc:odbc:<ODBC Data Source Name>, where the <ODBC Data Source Name> is the name of the ODBC data source you just created. If you are using a JDBC DRIVER, enter: 'jdbc:sqlserver://<MS SQL Server Host Name or IP Address>:1433;DatabaseName=<MS SQL Server Database Name>,' substituting actual values for <MS SQL Server Host Name or IP Address> and <MS SQL Server Database Name>.
Database User	Specify the login name of the database user with appropriate privilege.
Database Password	Specify the password for the SiteProtector Database User.

Add a JDBC Driver

The IBM SiteProtector DB, McAfee ePolicy Orchestrator DB, McAfee Vulnerability Manager DB, McAfee Network Security Manager DB, Microsoft SQL Server Audit Multiple Instance DB, Symantec Endpoint Protection DB, and Trend Micro Control Manager NG DB connectors extract events from a SQL Server database. For information about and to download the MS SQL Server JDBC Driver, see the Microsoft web site.

Note: Different versions of the JDBC driver are required for different SQL Server database versions; be sure to use the correct driver for your database version. The name of the jar file may be different for some JDBC driver versions.

The SmartConnector for Snort DB extracts events from a MySQL database.

After downloading and extracting the JDBC driver, upload the driver into the repository and apply it to the appropriate container or containers, as follows:

1. From ArcSight Express, select **Setup > Repositories**.
2. Select **JDBC Drivers** from the left pane and click the **JDBC Drivers** tab.
3. Click **Upload to Repository**.
4. From the **Repository File Creation Wizard**, select **Individual Files**, then click **Next**.
5. Retain the default selection and click **Next**.
6. Click **Upload** and locate and select the *.jar* file you downloaded.
7. Click **Submit** to add the specified file to the repository and click **Next** to continue.
8. After adding all files you require, click **Next**.
9. In the **Name** field, specify a descriptive name for the zip file (`JDBCdriver`, for example). Click **Next**.
10. Click **Done** to complete the process; the newly added file is displayed in the **Name** field under **Add Connector JDBC Driver File**.
11. To apply the driver file, select the driver `.zip` file and click the up arrow to invoke the **Upload Container Files** wizard. Click **Next**.
12. Select the container or containers into which the driver is to be uploaded; click **Next**.
13. Click **Done** to complete the process.

Configuration guides for the database connectors supported with ArcSight Express can be found on the microfocus.com website. The individual configuration guides that provide setup information and mappings for the applications listed below can be found on Micro Focus Community:

- IBM SiteProtector DB
- McAfee ePolicy Orchestrator DB
- McAfee Vulnerability Manager DB (formerly FoundScan)
- McAfee Network Security Manager DB
- Microsoft SQL Server Multiple Instance Audit DB
- Oracle Audit DB
- Symantec Endpoint Protection DB
- Trend Micro Control Manager DB
- Snort DB

API Connectors

The following API connectors are available for installation with ArcSight Express. They require a client and authentication credentials, as well as configuring the events types to be sent to the connector by the device.

- Cisco Secure IPS SDEE
- Sourcefire Defense Center eStreamer

For Cisco Secure IPS SDEE, if you want the SmartConnector to validate the Cisco IPS sensor's authentication certificate, obtain the authentication certificate from the IPS sensor and import it to the appliance.

For Sourcefire Defense Center eStreamer, add an eStreamer client, create an authentication certificate, and select event types to be sent to the connector.

See the individual configuration guides for these connectors for instructions.

Follow the instructions in "Uploading Certificates to the Repository" in the Connector Management for ArcSight Express 4.0 User's Guide to import the trusted certificates to ArcSight Express.

Configuration guides for the API connectors supported with ArcSight Express can be found on the microfocus.com website, as well as the individual configuration guides that provide setup information and mappings for the applications listed below.

- Cisco Secure IPS SDEE
- Sourcefire Defense Center eStreamer

File Connectors

File-based connectors use the Network File System (NFS) or the Common Internet File System (CIFS).

The following File connector is available for installation with ArcSight Express:

- Blue Coat Proxy SG Multiple Server File

See the configuration guide for device setup, parameter configuration, and mappings information for the SmartConnector for Blue Coat Proxy SG Multiple Server File.

File-based connectors use the Network File System (NFS) or the Common Internet File System (CIFS). For the file-based connectors on a Windows system, configure a CIFS share before you add the connectors.

For information on creating a CIFS Mount or an NFS Mount, see "Managing a Remote File System" in the Connector Management for ArcSight Express 4.0 User's Guide.

Syslog Connectors

If you selected Syslog Daemon during initial installation with the First Boot Wizard, the Syslog Daemon connector has already been installed.

You can add a Syslog File, Pipe, or Daemon connector in a new container. Syslog connectors for the following devices are available with ArcSight Express:

- Cisco PIX/ASA Syslog
- Cisco IOS Router Syslog
- Juniper Network and Security Manager Syslog
- Juniper JUNOS Syslog
- UNIX OS Syslog

Be sure your device is set up to send syslog events. See your device documentation or the SmartConnector Configuration Guide for device configuration information; the guide also includes specific device mappings to ArcSight event fields as well as further information needed for configuration if you are installing the Pipe or File connectors. Mappings in the SmartConnector for UNIX OS Syslog configuration guide apply to all syslog connectors. Specific mappings per device are documented in the configuration guide for the device.

Configuration guides for these syslog connectors supported with ArcSight Express can be found on the microfocus.com website:

- Cisco PIX/ASA Syslog
- Cisco IOS Syslog
- Juniper JUNOS Syslog
- Juniper Network and Security Manager Syslog
- UNIX OS Syslog

System Administration

This chapter describes the System Administration tools that enable you to create and manage users and user groups, configure SMTP and other system settings, network, storage, and security settings for your system.

This chapter includes information on the following areas of system administration:

System

From the System tab, you can configure system specific settings such as network settings (if applicable) and SMTP.

System Reboot

To reboot or shutdown your system:

1. Click **Administration > Setup > System Admin** from the top-level menu bar.
2. Click **System Reboot** in the **System** section.
3. Select from the following options:

Button	Description
Reboot	Your system reboots in about 60 seconds. The reboot process normally takes 5-10 minutes, during which time the system is unavailable.
Reboot in 5 Minutes	Your system reboots after a 5-minute delay. The reboot process normally takes 5-10 minutes, during which time the system is unavailable.
Shutdown	Automatically shuts down (powers off) the system.

Note: Each of the above actions can be cancelled. “Reboot” and “Shutdown” allow for cancellation within **60 seconds**. “Reboot in 5 Minutes” can be cancelled within **300**

seconds.

4. Click **Reboot**, **Reboot in 5 Minutes**, or **Shutdown** to execute the chosen action.

Network

System DNS

The **System DNS** tab allows you to edit the DNS settings and to add DNS search domains.

To change DNS settings:

1. Click **Administration > Setup > System Admin** from the top-level menu bar.
2. Click **Network** in the **System** section.
3. In the **System DNS** tab, specify new values for the IP address of the primary and secondary DNS servers, or edit the list of search domains.

   To add a new domain, click the icon. To remove a domain, click the icon. To change the search order of domains, select a domain name, and click the up or down arrow until the domain is in the desired position.
4. Click **Save**.
5. Click **Restart Network Service** to put the changes into effect.

Hosts

The **Hosts** tab allows direct editing of your system's `/etc/hosts` file. You can specify data in the System Hosts text box or import it from a local file.

To change the Hosts information:

1. Click **Setup > System Admin** from the top-level menu bar.
2. Click **Network** in the **System** section, then click the **Hosts** tab.
3. In the **System Hosts** text box, specify hosts information (one host per line) in this format:

   `<IP Address> <hostname1> <hostname2> <hostname3>`

   When editing your `etc/hosts` file, ensure that the IP address specified each host is unique and not duplicated across hosts. A single IP address can be associated with multiple hostnames, but the same IP address may not be used for multiple hosts.

   To import information from a file, click **Import from Local File,** and locate the text file on

the computer from which you are accessing your system.

4. Click **Save**.

NICs

The **NICs** tab enables you to set the IP addresses for the network interface cards (NICs) on your system. Additionally, you can configure the hostname and default gateway for your system.

To set or change the NICs settings:

1. Click **Setup > System Admin** from the top-level menu bar.
2. Click **Network** in the **System** section.
3. In the **NICs** tab, specify the following settings. To edit the IP address , subnet mask, or speed/duplex of an NIC, select the NIC and click **Edit** above the NIC Name list.

Setting	Description
Default Gateway	The IP address of the default gateway.
Hostname	The network host name for this system. Make sure that your DNS can resolve the host name you specify to your system's IP address . Performance is significantly affected if DNS cannot resolve the host name. This name must be identical to the domain specified in the Certificate Signing Request, described in "Generating a Certificate Signing Request (CSR)" on page 1. **Note:** If you previously used a self-signed or CA-signed certificate on this system and are now changing its host name, you must regenerate a new self-signed certificate or CSR. Once obtained, the new certificate should be uploaded to ensure that the connectors which communicate with your system are able to validate the host name. For more information about generating a CSR, see "Generating a Certificate Signing Request (CSR)" on page 1.
Automatically route outbound packets (interface homing)	When this option is enabled (checked box), the response packets are sent back on the same system interface on which the request packets had arrived. Enabling this option can improve performance as the routing decisions do not need to be made (using the default gateway information and static routes) to send packets out from your system. If you have static routes configured, they are ignored when this feature is enabled. When this feature is disabled (unchecked box), the static routes (if configured) are used to determine the interface through which the response packets should leave your system. If you configure only one network interface, this setting does not provide any additional benefit.

Setting	Description
IP Address	The IP address for each network interface card (NICs) in your system. **Add NIC Alias** You can create an alias for any listed NIC. To do so: a. Highlight the NIC for which you want to create an alias. b. Click **Add**. c. Create an alternative IP address for the alias. d. Click **Save**. You can identify the alias from its original by an appended colon alongside a digit indicating the number of aliases you have created on a particular NIC. **Notes:** • You cannot alter the speed of an IP alias. • You can create as many aliases as you choose.
Subnet Mask	The subnet mask associated with the IP address you entered for an NIC.
Speed/Duplex	Select a speed and duplex mode, or let your system determine the network speed automatically: Auto (recommended) 10 Mbps - Half Duplex 10 Mbps - Full Duplex 100 Mbps - Half Duplex 100 Mbps - Full Duplex 1 Gbps - Full Duplex

4. Click **Save**.
5. Click **Restart Network Service** to put the changes into effect.

Static Routes

You can specify static routes for the NICs on your system.

To add, edit, or delete a static route:

1. Click **Setup > System Admin** from the top-level menu bar.
2. Click **Network** in the **System** section.
3. In the **Static Routes** tab:
   - To add a new static route, click **Add**.
   - To edit or delete an existing route, select the route first, then click **Edit** or **Delete**.

When adding or editing a static route, you need to configure these settings.

Setting	Description
Type	Whether the static route is to a Network or a Host
Destination	The IP address for the static route destination
Subnet Mask	The subnet mask if you specify a network as the destination
Gateway	The IP address of the gateway for the route

4. Click **Save**.

Time/NTP

The **Time/NTP** tab enables you to configure system time, date, local timezone, and NTP servers. Micro Focus strongly recommends using an NTP server instead of manually configuring the time and date on your system.

To set or change the system time, date, or time zone manually:

Caution: If you manually set the date and time settings and are also using an NTP service, the date and time entered manually cannot be more than 16 minutes ahead of or behind the time that the NTP server is providing. If the manually entered time is more than 16 minutes different from the NTP server time, then the NTP service will fail to start.

1. Click **Setup > System Admin** from the top-level menu bar.
2. Click **Network** in the **System** section.
3. In the **Time/NTP** tab, configure these settings.

Setting	Description
Current Time Zone	The time zones appropriate to your system's location. To change this setting, click **Change Time Zone**... Local times zones follow the Daylight Saving Time (DST) rules for that area. Greenwich Mean Time (GMT) + and - time zones are DST agnostic. For example, the America/Los Angeles time zone varies by an hour compared with GMT when DST goes into and out of effect. • Pacific Standard Time (PST) = GMT-8 • Pacific Daylight Time (PDT) = GMT-7
Current Time	The current date and time at the system's location. To change this setting, click **Change Date/Time...** and then specify the current date and time.

4. The Time Zone change requires that you reboot the appliance. However, the Current Time change takes effect immediately.

To configure your system as an NTP server or for using an NTP server for your system:

1. Click **Setup > System Admin** from the top-level menu bar.
2. Click **Network** in the **System** section.
3. Click the **Time/NTP** tab.
4. Under **NTP Servers**, configure these settings.

   To add a new NTP server, click the icon. To remove a server, click the icon. To change the order in which the NTP servers should be used, select a server and click the up or down arrow until the NTP server is in the desired position.

Setting	Description
Enable as an NTP server	Check this setting if this system should be used as an NTP server.
NTP Servers	Specify the host name of an NTP server. For example, time.nist.gov. Micro Focus recommends using at least two NTP servers to ensure precise time on your system. To specify multiple NTP servers, type one server name per line. Notes: • An ArcSight system can serve as an NTP server for any other ArcSight system. • If System A serves as an NTP server for System B, System B needs to list System A in its **NTP Servers** list. • Use the **Test Servers** button to verify the status of the servers entered into the **NTP Servers** box.

5. Click **Save**.

   Tip: You may need to scroll down to view the **Save** button and **Restart NTP Service**.

6. Click **Restart NTP Service** to put the changes into effect.

SMTP

Your system uses the Simple Mail Transfer Protocol (SMTP) setting to send email notifications such as alerts and password reset emails.

To add or change SMTP settings:

1. Click **Administration > Setup > System Admin.**
2. Click **SMTP** in the **System** section and specify these settings.

Setting	Description
Enable SMTP Auth Mode	Enable/Disable secure authenticated mode of communication with SMTP server.
Primary SMTP Server	Mandatory. The IP address or hostname of the SMTP server that will process outgoing email.
Primary SMTP Server Port	Primary SMTP Server Port. Required if SMTP Auth Mode is enabled.
Username	Primary SMTP Server Username. Required if SMTP Auth Mode is enabled.
Password	Primary SMTP Server Password. Required if SMTP Auth Mode is enabled.
Upload Cert File SMTP Primary	Upload Primary SMTP Server Certificate. Required if SMTP Auth Mode is enabled.
Backup SMTP Server	Mandatory. The IP address or hostname of the SMTP server that will process outgoing email in case the primary SMTP server is unavailable.
Ba ckup SMTP Server Port	Secondary SMTP Server Port. Required if SMTP Auth Mode is enabled.
Username	Secondary SMTP Server Username. Required if SMTP Auth Mode is enabled.
Password	Secondary SMTP Server Password. Required if SMTP Auth Mode is enabled.
Upload Cert File SMTP Backup	Upload secondary SMTP Server Certificate. Required if SMTP Auth Mode is enabled.
Outgoing Email Address	The email address that will appear in the From: field of outbound email.

3. Click **Save**.

   For more information on SMTP Configuration, see Connecting to Your SMTP Server.

License & Update

This page displays license information, the version of the components, and the elapsed time since Arcsight Management Center was last rebooted. From here, you can update Arcsight Management Center and apply a license.

Updating the Appliance

To update your Arcsight Management Center:

1. Download the update file from the Micro Focus Support site at https://softwaresupport.softwaregrp.com/ to the computer from which you can connect to Arcsight Management Center.
2. Click **Administration > Setup > System Admin** from the top-level menu bar.

3. Click **License & Update** in the **System** section.
4. Click **Browse** to locate the file.
5. Click **Upload Update**.

   An “Update In Progress” page displays the update progress.
6. Once the update has completed, the Update Results page displays the update result (success/failure) and whether the update requires a reboot. If the update requires a reboot, the Arcsight Management Center reboots automatically.

Updating the License File

To update a license file:

1. Download the license update file from the Micro Focus Support site at https://softwaresupport.softwaregrp.com/ to the computer from which you can connect to the Arcsight Management Center with your browser.
2. Log in to the Arcsight Management Center user interface using an account with administrator (upgrade) privileges.
3. Click **Administration > System Admin.**
4. Click **License & Update** in the **System** section.
5. Browse to the license file you downloaded earlier, and click **Upload Update**.

   An “Update In Progress” page displays the update progress.

After the update has completed, the Update Results page displays the update result (success/failure). If you are installing or updating a license, a reboot is required.

Note: After updating the license file, refresh the browser to see the current list of enabled features.

Process Status

The **Process Status** page lists all processes related to your system and enables you to view the details of those processes and start, stop, or restart them.

To view the Process Status page:

1. Click **Administration > Setup > System Admin.**
2. In **System** section, click **Process Status**.
3. To view the details of a process, click the icon to the left of the process name.

4. To start, stop, or restart a process, select the process and click **Start**, **Stop**, or **Restart** at the top of the **Processes** list.

System Settings

If you did not select Arcsight Management Center to start as a service during the installation process, you can do so using the **System Settings** page.

To configure Arcsight Management Center to start as a service:

1. Click **Administration > Setup> System Admin.**
2. Click **System Settings** in the left panel.
3. From under **Service Settings**, choose the appropriate option:
   - Start as a Service
   - Do not start as a Service
4. Click **Save**.

SNMP

SNMP (Simple Network Management Protocol) can be used to monitor the health of your appliance. ArcSight Platform supports versions 2c and 3 of SNMP.

SNMP Configuration

You can configure SNMP polling and notifications. If SNMP polling is configured, a manager station can query the SNMP agent residing on the ArcSight Platform. The information retrieved provides detailed information at the hardware and operating system level.

To configure SNMP polling:

1. In the main menu bar, click **Administration > Setup> System Admin**
2. In the navigation tree, under **System**, click **SNMP**.
3. On the **SNMP Poll Configuration** tab, ensure **Enabled** is selected.
   - For **Port**, the default is *161* but can be any available port. Ensure the specified port is open on your firewall.
   - For **SNMP version**, select *V2c* or *V3,*
     - If *V2c* is selected, specify a community string of between 6 and 128 alphanumeric, underscore, and dash characters.

- If *V3* is selected, specify the username (alphanumeric lower-case string of 4-16 characters, which must begin with an alphabetic characters and may include underscores), authentication protocol, authentication passphrase (4 to 256 characters), privacy protocol, and privacy passphrase (4 to 256 characters).

4. Click **Save**.

If an SNMP destination is configured, ArcSight Platform can send notifications for a limited set of events (see "Viewing SNMP System Information" below

SNMP notifications differ from those sent by connectors, which are for a generic ArcSight event. The notifications listed here are specific to a single event, making them easier for understanding by a network management system.

To configure the destination for SNMP notifications:

1. In the main menu bar, click **Administration > System Admin**
2. In the navigation tree, under **System**, click **SNMP**.
3. On the **SNMP Destination** tab, ensure **Enabled** is selected. Then, specify values for the other parameters that match your existing NMS SNMP settings.
   - For Port, specify *162*. (Note: Specifying a non-default port may cause a brief delay. Give the process time to complete.)
   - For SNMP version, select *V2c* or *V3,*and then specify values for the prompted settings.
4. Click **Save**

Viewing SNMP System Information

SNMP notifications are viewable in any MIB browser. The following SNMP notifications are supported:

- **Application**
  - Login attempt failed
  - Password change attempt failed
  - User account locked
  - Reboot command launched
  - Manual backup failed
  - Enable FIPS mode successful
  - Disable FIPS mode successful
  - Enable FIPS mode failed
  - Disable FIPS mode failed

- **Platform**
  - CPU Usage
  - Memory Usage
  - Disk Almost Full
  - Fan Failure
  - Power Supply Failure
  - Temperature Out of Range
  - Ethernet Link Down

To view system notifications in an MIB browser:

On your appliance:

You can download the ArcSight MIB file and other standard Net-SNMP MIB files using the following URLs:

- `https://<system_name_or_ip>/platform-service/ARCSIGHT-EVENT-MIB.txt`
- `https://<system_name_or_ip>/platform-service/DISMAN-EVENT-MIB.txt`
- `https://<system_name_or_ip>/platform-service/HOST-RESOURCES-MIB.txt`
- `https://<system_name_or_ip>/platform-service/IF-MIB.txt`
- `https://<system_name_or_ip>/platform-service/UCD-SNMP-MIB.txt`

In any standard MIB browser:

1. Load the MIB in the browser.
2. Specify the address and port number of the SNMP agent—your appliance, in this case.
3. Configure the community string that is set on your appliance.
4. Initiate the SNMP WALK operation of the OID from the browser.
5. Once the SNMP data is returned, interpret it based on the information described earlier in this section.

MIB Contents

Notifications are written to the following modules of the MIB file:

Module	Notification Types
HOST-RESOURCES-MIB	Standard hardware parameters.
IF-MIB	Objects for network interfaces.
IP-MIB	IP and ICMP implementations.
DISMAN-EVENT-MIB	Event triggers and actions for standard network management.

Troubleshooting

The following troubleshooting tips might be helpful in resolving issues with yourCDF cluster or product license:

- "Troubleshooting Your Cluster" below
- "Troubleshooting Issues with Your Product License" on page 1083

More troubleshooting information is available on the Micro Focus support portal.

Troubleshooting Your Cluster

The following troubleshooting tips may be helpful in resolving issues with the CDF cluster.

Issue	Description
Installation of master nodes fails	During installation, installation of Master Nodes can fail with the error: `Unable to connect to the server: context deadline exceeded` Ensure that your `no_proxy` and `NO_PROXY` variables include valid virtual IP addresses and hostnames for each of the master and worker nodes in the cluster, as well as the NFS server.
Installation times out	During installation, the process may time out with the error: `Configure and start ETCD database` Ensure your `no_proxy` and `NO_PROXY` variables include correct Master Node information.

Issue	Description
During sudo installation, worker node fails to install	During the Add Node phase, if one or more of the worker nodes fails to install and the log shows the following error message: `[ERROR] : GET Url: https://itom-vault.core:8200/v1/***/PRIVATE_KEY_CONTENT_{hostname}_{sudo user}, ResponseStatusCode: 404` You can take the following steps to rectify the issue: 1. Click **Cancel** to return to the version selection screen. 2. Proceed through the installation screens again (all previous data is preserved). 3. On the **Add Node** screen, where you added the Worker Node data, remove the worker node which failed by clicking on the **Delete** icon. 4. Click **Add Node** and add the node again. 5. Click **Next** and proceed with the installation.
Cluster list empty in Kafka Manager	If cluster list is empty in the Kafka Manager UI, delete the existing Kafka Manager pod and try the UI again after a new Kafka Manager pod is back to the Running state.
Worker nodes out of disk space and pods evicted	If the worker nodes run out of disk space, causing the pods on the node to go into Evicted status, try one of the following steps: • Fix the disk space issue by adding an additional drive or contact Micro Focus support to receive help remove unnecessary files. • On the node where the low disk space occurred, run the following command: `{install dir} /kubernetes/bin/kube-restart.sh` For information on adjusting the eviction threshold, see "'Updating the CDF Hard Eviction Policy" on page 371."

Kafka fails to start up; fails to acquire lock or corrupted index file found	Many scenarios can cause a failure for Kafka to start up and report either `Failed to acquire lock` or `Corrupted index file found.`
	Workaround: To resolve this on the problematic Kafka node:.
	1. Go to the directory:
	`cd /opt/arcsight/k8s-hostpath-volume/th/kafka/`
	2. Find the file `.lock`, and delete it.
	3. Search for all index files:
	`find . -name "*.index" | xargs ls -altr`
	4. Delete all the corrupted index files
	5. Restart the affected Kafka pod.

Issue	Description
Slow network or slow VM response during upgrade causes delay or failure of web services operations	An intermittent issue has been observed with web service pod startup, during the upgrade toTranformation Hub 3.3, that correlates with slow network and/or slow VM response. The pod startup gets blocked or delayed, leading to various issues, such as failing to create new topics and/or failing to register the new schema version. One error seen in the web service log file is, `"Thread Thread[vert.x-eventloop-thread-0,5,main] has been blocked for 5715 ms, time limit is 2000".` The workaround is to restart the web service pod.
Arcsight database rejects new sessions because the maximum sessions limit is reached	You might observe the following error in the logs: [Vertica][VJDBC](4060) FATAL: New session rejected due to limit, already 125 sessions active **Workaround:** Do one of the following: • Delete the active open sessions to ensure that the total number of active sessions is within the specified maximum limit. • Refer to the Arcsight Database documentation to configure the maximum sessions.
ArcSight Database fails to restart	If the database fails to start, you can run a set of commands to recover the last known good set of data and restart the database. For example, the database might not restart after an unexpected shutdown. Please consult your database administrator for the commands to run.
Multiple node failures	Here are some considerations when handling node failures on 3 or more worker nodes. • A cluster with 3 masters and 3 or more worker nodes should have at least 2 or more master and worker nodes running (quorum) to work properly in high availability. • As a general rule in terms of data loss prevention, no more than `TOPIC_REPLICATION_FACTOR` minus 1 worker nodes can be down at any time • Handling failures and stability if Worker nodes go down: ◦ Resume the stability of the cluster as follows: • Repair or replace any down worker nodes or replace with new ones • Delete any pods which are in “Terminating” state (this is the expected behavior for stateful pods in Kubernetes when nodes are down). ◦ Wait until the pod startup sequence is completed. The cluster should resume normal operation. ◦ Repair any issues on the lost nodes, the cluster should return to Running state
Second upgrade fails or some resources aren't really upgraded after it	In some cases, a second upgrade may fail completely or fail to upgrade resources. If this is encountered, run the following command: `kubectl delete deployment suite-upgrade-pod-arcsight-installer -n `kubectl get namespaces | grep arcsight-installer | awk '{print $1}'`` Wait until the suite-upgrade-pod-arcsight-installer is deleted, then begin the second upgrade again.

Issue	Description
CDF deployment fails on servers running VMWare V Motion	Installation of CDF may fail on virtual machines running the VMWare product VMotion. If this occurs, run the installation of CDF again but disable VMotion on all CDF virtual machines.
After adding or reducing stream processor instances, Kafka Manager fails to show accurate consumer information for topics	After adding or reducing the number of stream processor instances, Kafka Manager may fail to show correct consumer information for some topics. To get the most current consumer information, restart the Kafka Manager pod with the command: `kubectl delete pod -n arcsight-installer-XXX th-kafka-manager-XXX` Then reconnect to the Kafka Manager UI.
Kafka Manager not displaying members of the consumer group	When a new member is added to a consumer group, Kafka Manager must be restarted in order to display the new members. This applies to Logger, ESM, Vertica Scheduler, SOAR, and Intelligence.
New partition source topics not correctly displayed in Kafka Manager	Changes to the partition source topics in Kafka Manager may take up to 5 minutes to refresh and display correctly.

Troubleshooting Issues with Your Product License

This section provides guidance on issues that you might encounter related to your product license.

System Fails to Recognize a License Change

When you install or update a license, some components might not recognize the update because of cached data. You should to wait an hour to ensure that the update propagates across the system.

Conflicting Indicators about Your License

It's possible Autopass indicates that your license is valid but the product behaves as if the the license has expired or the pods fail to work. To check the status of a license, you can run the following command:

```
/opt/arcsight/k8s-hostpath-volume/<product>/autopass/license.log
```

For example, for Transformation Hub, run:

```
/opt/arcsight/k8s-hostpath-volume/th/autopass/license.log
```

The system responds with the following messages:

License status	Message
Valid license	`<product> licensed capacity: <eps number>`
Not installed	`ERROR: No valid license key was found. Please install a valid license key or contact Micro Focus Customer Support for instructions on how to get one`
Expired	`"<errorMessage>No license is found in Memory ..."`

Send Documentation Feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this computer, click the link above and an email window opens with the following information in the subject line:

Feedback on Administrator's Guide for ArcSight Platform 22.1.2 (ArcSight Platform 22.1.2)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to documentation-feedback@microfocus.com.

We appreciate your feedback!