NetIQ by opentext

NetIQ Cloud Bridge Agent

Installation and Administration Guide

February 2024

Legal Notice

Contents

About This Book

The *Installation and Administration Guide* provides conceptual information about the NetIQ Cloud Bridge Agent. This book includes conceptual information and step-by-step guidance for common tasks.

Intended Audience

This book provides information for individuals responsible for installing, configuring, and managing one or more Cloud Bridge Agents in a software-as-a-service (SaaS) environment. A working knowledge of network operations, network security, and cloud SaaS technologies is assumed.

Additional Documentation

For the most recent version of this guide and other NetIQ Cloud Bridge Agent documentation resources, visit the NetIQ Identity and Access Management Services Documentation web page (https://www.microfocus.com/documentation/identity-and-access-management/iam-services/).

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 OpenText Support at https://www.microfocus.com/support-and-services/.

1 Introduction to Cloud Bridge

In OpenText software-as-a-service (SaaS) or hybrid environments, Cloud Bridge is a data transfer bridge between applications in the cloud and data sources on premises. The Cloud Bridge Agent (CBA) is the on-premises entity that responds to the collection and fulfillment commands and directs them to the proper data source within the multi-tenant Cloud Bridge service for execution. An on-premises administrator installs the Cloud Bridge Agent and configures it with the necessary service parameters, including locally-persisted and protected credentials for on-premises and third-party data sources.

Understanding the Benefits of Cloud Bridge

Cloud Bridge provides the following benefits:

- Secure communication
- Simple credential management
- High availability

Cloud Bridge Secures Communication

Cloud Bridge simplifies and secures communication between SaaS applications, such as Identity Governance as a Service or Advanced Authentication as a Service, and on-premises identity sources and applications, such as Active Directory or Identity Manager. The Cloud Bridge Agent communicates with the SaaS applications through a secure messaging service outside the corporate firewall. This messaging service is adaptable for various workloads and provides guaranteed delivery of messages. No VPN is needed and all Cloud Bridge Agent connections are outbound connections to a well-defined port. Data is protected both in transit and at rest.

In a common scenario, you might have both on-premises and SaaS products interacting with Cloud Bridge. You have both on-premises NetIQ Identity Manager and SaaS licenses for Advanced Authentication and Identity Governance. Your employees need to log in to their SaaS accounts as well as their on-premises applications. Your employees are authenticated through the Advanced Authentication SaaS service, which communicates with on-premises identity sources through a Cloud Bridge messaging layer.

After your Cloud Bridge Agent is installed and running in your on-premises environment, it begins sending heartbeat messages. The OpenText SaaS operations team sets up the necessary data protection features and monitors the health of your installed Agent.

Cloud Bridge Manages Credentials

The credential management feature in the Cloud Bridge Agent ensures that the credentials for a target data source never leave your network. The Agent associates the credentials with the service configuration on demand.

Cloud Bridge Ensures High Availability

High availability capabilities in Cloud Bridge 1.9.0 or later also help you meet your organizational goals for operational performance. After you configure your environment to specify your preferred sites and CBA instances, when a planned or unplanned shutdown takes place, failover to the specified CBA site and instance occurs automatically and with minimal service interruption. As part of the failover process, the Cloud Bridge Client loads active service configurations previously used by the primary CBA instance into a new target CBA instance, enabling consuming applications to quickly resume their collection, provisioning, and other activities.

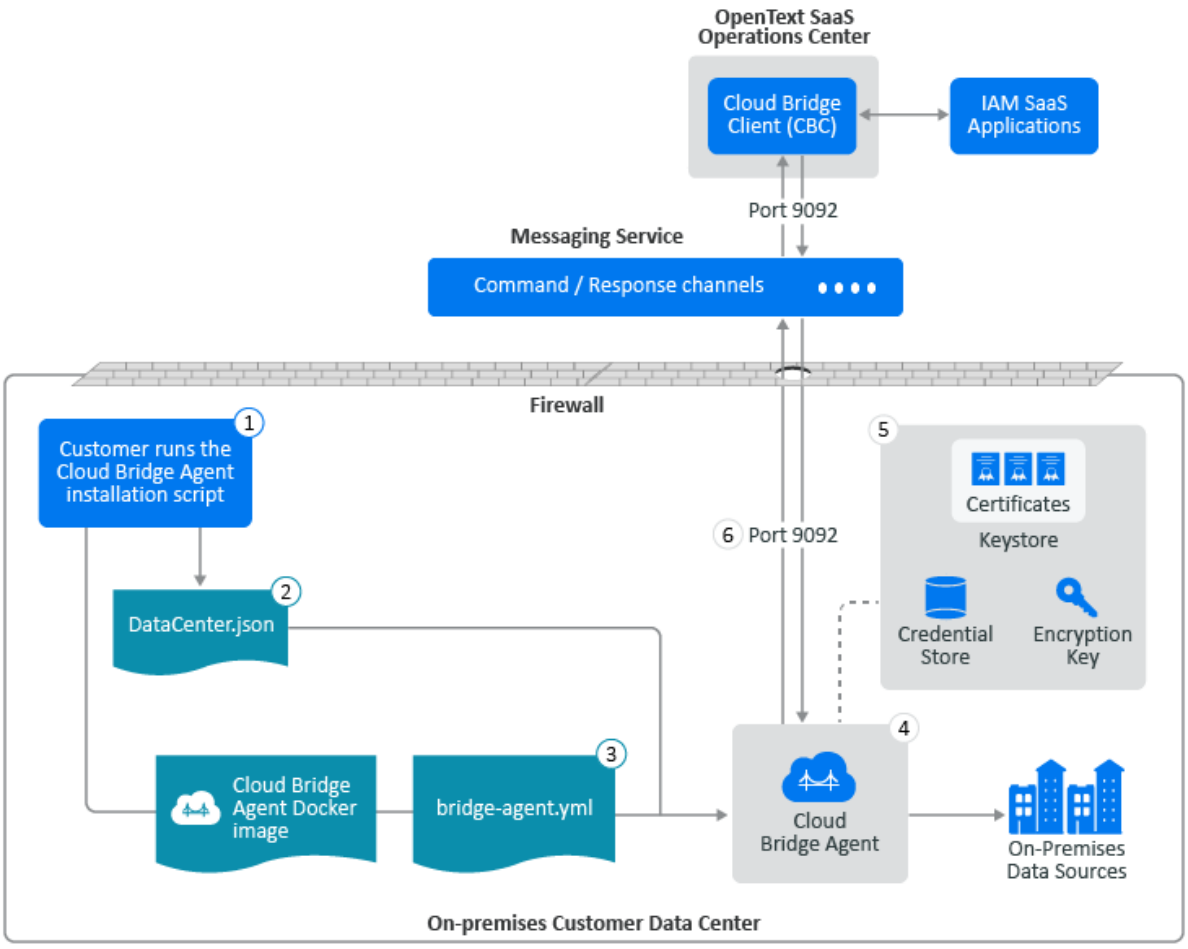
The high availability architecture means that more instances of the CBA simply require you to run additional CBA containers without any orchestration software or databases needed. You can set up as many failover instances as your organization requires, and all CBA instances can be in an active state concurrently. Not only do CBA administrators have the ability to view each CBA's instance configuration and the current target CBA instance, but the SaaS operations team also monitors your CBA instances. For more information about planning for high availability, see "Planning for High Availability" on page 13.

How the Cloud Bridge Agent Works

Your Cloud Bridge data center is configured as part of your OpenText SaaS product licensing (for example, Identity Governance as a Service) based on the information you provide in the technical questionnaire. **Data centers** are a conceptual representation of your Cloud Bridge Agent instance. You install one or more Agents on your local systems, then configure data sources and data source connections as needed to connect to your on-premises data sources. If you need to collect data from multiple data centers, you will need to install a Cloud Bridge Agent in each on-premises data center.

The following diagram illustrates the standard Cloud Bridge Agent configuration in your on-premises environment, regardless of the other components you might have installed.

Figure 1-1 *Overview of Cloud Bridge Agent Configuration On Premises*



For the numbered components on the above diagram, see the following additional information:

1. Requires access to the AWS download site to download the CBA package on port 443. The customer runs a custom curl/bash command. The curl portion downloads the container image and the main part of the installation script and then executes it.
2. DataCenter.json – Contains customer-specific data center information. Defines the connection information between the CBC and a single CBA instance.
3. bridge-agent.yml – Contains configuration settings for the Cloud Bridge Agent.
4. Host server – CBA runs in a Docker or Podman container on this server.
5. Secure communication components – Configurable on the CBA host server.
6. Port 9092 – Facilitates the secure outbound data transfer from the on-premises CBA to the Cloud Bridge API in the Open Text AWS cloud (`*.aws.confluent.cloud`).

2 Planning Your Cloud Bridge Agent Environment

Before you can install the Cloud Bridge Agent (CBA) on premises, the SaaS operations team must have granted you the privilege to use Cloud Bridge.

IMPORTANT: You should not attempt to install the Cloud Bridge Agent in a production environment until after you have successfully deployed the CBA in a test environment.

Understanding Requirements

Review the following sections and ensure that your environment meets all requirements before you install the Cloud Bridge Agent (CBA).

Access Prerequisites

You must have administrator privileges to install the Cloud Bridge Agent. In addition, you must have the following rights:

- Access to the installer of a supported Linux operating system. For more information, see "Hardware and Software Requirements" on page 12.
- Access to the relevant repositories to install or upgrade the operating system and the Docker or Podman container.
- Access to outbound TCP port 9092 to allow outbound TLS communication from the on-premises Cloud Bridge Agent to the Cloud Bridge API in the OpenText AWS cloud (`*.aws.confluent.cloud`).

  IMPORTANT: You must use DNS filtering because OpenText cannot provide specific hosts.

- Access to the AWS download site to download the Cloud Bridge package on port 443.
- Access rights for each server on which you plan to install the CBA for high availability.

If you are using Identity Governance with Cloud Bridge, you must also have:

- Access from your on-premises Cloud Bridge Agent to the on-premises authentication directory or directories for Advanced Authentication and other applications from which you expect to collect data as part of the Identity Governance collection processes
- Internal access to your Cloud Bridge Agent using a supported browser on port 8080

IMPORTANT: Advanced Authentication can use the following user lookup attributes by default for authentication purposes to Advanced Authentication and Identity Governance:

- Active Directory: mail, sAMAccountName, and userPrincipalName
- eDirectory: cn and mail

Every Advanced Authentication repository has a configured Base DN. The value found on the specified Base DN for a specific user must be unique for the authentication service to work. This means that repeated values for a given user trying to authenticate will result in an authentication error.

Hardware and Software Requirements

This section provides the minimum hardware and software requirements for each server on which you install the Cloud Bridge Agent.

NOTE: We recommend hosting the CBA on a dedicated server. If your organization has a dedicated server with sufficient resources, it is possible for the CBA to co-exist with other containers. However, ensure that it does not have to compete for resources.

If you do install the Cloud Bridge Agent on the same server with another on-premises product, the installation order is not important.

Table 2-1 *Cloud Bridge Agent Server Requirements*

Category	Minimum Requirement
Processors	4 CPUs
Memory	16 GB of RAM
Hard disk space	200 GB **NOTE:** If you are using non-standard Linux partitioning and you are partitioning `/var/tmp`, ensure that at least 16 GB of the required 200 GB are allocated to this partition.
VM environment	(Conditional) If you plan to install the CBA on a virtual machine, VMware ESX 6.7 or later.
Operating system	One of the following Linux operating systems: ◆ Ubuntu 18.04 LTS Server Edition or later ◆ RHEL Server 8.3 ◆ SUSE Linux Enterprise Server 15.1 or later **NOTE:** Ensure that wget and awk are installed before you attempt to deploy the Cloud Bridge Agent container.
Container	One of the following: ◆ Docker 19.03.*x* or later ◆ Podman 1.6.4 or later

Category	Minimum Requirement
Browser	One of the following to access the Cloud Bridge Agent console: ◆ Google Chrome (latest version) ◆ Firefox (latest version)

NOTE: You must configure a DNS entry for your LDAP server in order to use a secure connection. You will not be able to connect over SSL (port 636) if you are using a host file.

Planning for High Availability

You can ensure that your organization provides reliable services to your employees without downtime by planning for and configuring high availability in Cloud Bridge 1.9.0 or later.

Understanding High Availability in Cloud Bridge

Cloud Bridge addresses both instance and site failover scenarios as follows:

- **Instance failover:** If the hardware or container hosting the CBA fails, Cloud Bridge can immediately switch to another CBA instance with the same data center configuration that has access to the same data sources.
- **Site failover:** If an entire data center location becomes inoperative due to a catastrophic event, Cloud Bridge can immediately switch to a CBA instance at another site that has access to the same data sources.

When you install a CBA, you specify the following properties that are saved to the `bridge-agent.yml` configuration file for the CBA:

- **Instance ID** - A unique identifier for that instance
- **Site Weight** - Specifies the priority (Primary, Secondary, or Backup) of the site in relation to other data center sites

  In the event of failure, all CBA instances within a Primary site take precedence over CBA instances with a Secondary site configuration, and the Secondary site instances take precedence over CBA instances with a Backup site configuration.
- **Instance Weight** - Specifies the priority (Primary, Secondary, or Backup) of the instance in relation to other CBA instances

  In the event of failure, all CBA instances (within the same site) with a Primary instance weight take precedence over CBA instances with a Secondary instance configuration, and the Secondary instances take precedence over CBA instances with a Backup instance configuration.

Cloud Bridge calculates the configured site weight and instance weight into a single `haWeight` value. When a shutdown takes place, whether planned or unplanned, Cloud Bridge uses a simple numerical comparison of the weight values from all active CBA instances to determine which CBA instance becomes the target CBA instance. Cloud Bridge fails over to the highest weighted instance that it finds. You can use whatever combination of weights you choose.

For a site, the weight values are as follows:

- Primary = 30
- Secondary = 20
- Backup = 10

For an instance, the weight values are as follows:

- Primary = 5
- Secondary = 3
- Backup = 1

So, for example, if you configured nine CBA instances in total, with a single CBA instance at each possible weight, failover would occur in the following order:

Table 2-2 *Configuration Combinations*

Order	Site	Instance	Calculated haWeight Value
1.	Primary (30)	Primary (5)	35
2.	Primary (30)	Secondary (3)	33
3.	Primary (30)	Backup (1)	31
4.	Secondary (20)	Primary (5)	25
5.	Secondary (20)	Secondary (3)	23
6.	Secondary (20)	Backup (1)	21
7.	Backup (10)	Primary (5)	15
8.	Backup (10)	Secondary (3)	13
9.	Backup (10)	Backup (1)	11

NOTE: ◆Cloud Bridge does not prevent you from configuring multiple CBA instances with the same weight values. If multiple instances have the same weight, the Cloud Bridge Client selects the *first* instance with that weight that it detects.

- Secondary and Backup sites and instances provide additional layers for failover in larger environments, and you can have multiple sites and instances at each level if needed. However, you do not need to assign both Secondary and Backup site and instance roles if your organization does not need this level of complexity.

Understanding Cloud Bridge Communication in Failover Scenarios

Each active CBA sends a heartbeat message when it starts up and every 30 seconds thereafter. The Cloud Bridge Client monitors these heartbeats and is able to quickly detect an instance outage.

- In a scenario where an administrator intentionally shuts down a CBA instance for maintenance, re-hosting, and so on, the CBA sends a heartbeat immediately to the Cloud Bridge Client, allowing failover to a new target to be performed instantly.
- In an unplanned shutdown scenario where the target CBA instance is no longer able to communicate its heartbeat to the Cloud Bridge Client, the Client has an agent monitoring task that detects when a target CBA has not communicated within a configured timeframe (with a default of one minute). It then attempts to ping the CBA. If the ping attempt fails, the Cloud Bridge Client marks the target CBA as unresponsive and initiates the CBA target selection process.

Whether the shutdown was planned or unplanned, after the CBC selects a new target CBA from the current list of active CBA instances, the CBA sends a ping command to all CBA instances to inform them of the new CBA target selection. The CBC also loads the current data source configurations into the new target CBA. All data collection sessions that were active at the time of the shutdown fail, and new commands and collections are routed to the new target CBA.

If no active and initialized CBA is available, the CBC marks the data center as unconnected and uninitialized. All command traffic that is sent to the data center receives an immediate “no agent available” error response.

If a non-target CBA instance is shut down, whether intentionally or not, the CBC simply removes it from the list of active agent instances.

Understanding Encryption IV and Key Values

Cloud Bridge uses initialization vectors (IVs) and keys to encrypt and decrypt data for protection of customer data and data source connection passwords stored in the CBAs. For more information, see TechTarget (https://www.techtarget.com/whatis/definition/initialization-vector-IV) and Wikipedia (https://en.wikipedia.org/wiki/Initialization_vector).

Each instance of the CBA configuration in your high availability environment must use the same encryption IV and Key values. Using the same values on all CBA instances ensures that no disruptions will occur during failovers.

If you create new encryption IV and Key values for the CBA configurations, the set of credentials for the CBAs (which were encrypted using the older keys) become invalid. At that point you must reset the password values. The CBA console indicates which credential sets are invalid.

For information about how to set the same encryption IV and Key values on secondary and backup CBA instances after you have installed your primary CBA instances, see “Installing Secondary and Backup CBA Instances” on page 23.

Essential Components of Successful Failover

The following are critical elements of successful configuration of the CBA high availability components:

- Each CBA instance must have a unique Instance Id.
- Each instance of the CBA configuration must use the same encryption IV and Key values. For more information, see “Understanding Encryption IV and Key Values” on page 15.
- Cloud Bridge does not prevent you from configuring multiple CBA instances with the same weight values. If multiple instances have the same weight, the Cloud Bridge Client selects the *first* instance with that weight that it detects.
- To enable effective failover, it is important to ensure that the set of data source credentials that are stored in the high availability CBAs are correct and consistent. After you have set up credentials for your first CBA, you can export and import credentials for the remaining CBAs.

NOTE: It is possible to run a single CBA instance using a pre-1.9.0 CBA `bridge-agent.yml` file. The Instance Id will default to an empty string, and the Site Weight and Instance Weight values will default to Backup. However, we do not recommend this scenario except for continuity during a CBA upgrade process.

Planning for Failover

Before you begin preparing servers to install the Cloud Bridge Agent, you should determine the following:

- If you have more than one data center, which one will be the primary site, and which ones will serve as secondary and backup sites in the event of catastrophic site failure
- Number of CBA servers you plan to install in each data center
- The priority (primary, secondary, or backup) of each CBA server you plan to install
- The naming convention you will use to identify each CBA instance

We recommend that you create a spreadsheet similar to the following example to record your decisions. Ensure that you keep this document up to date as you make changes in your environment.

Table 2-3 *High Availability Planning Spreadsheet*

Data Center Site Name	Site Priority	Instance Id	Instance Priority
Houston data center	Primary	Hou_Instance_1	Primary
		Hou_Instance_2	Secondary
		Hou_Instance_3	Backup
Provo data center	Secondary	Pro_Instance_1	Primary
		Pro_Instance_2	Secondary
		Pro_Instance_3	Backup
Cambridge data center	Backup	Cam_Instance_1	Primary
		Cam_Instance_2	Secondary
		Cam_Instance_3	Backup

Recommended Installation Order

If you have a large organization with multiple servers and sites, the most efficient method for you to set up your environment for successful failover is as follows:

1. Install your first CBA instance, saving your encryption key and ID for reuse.
2. Install all subsequent CBA instances using the encryption key and ID that you copied from your first CBA instance.
3. Configure your first CBA instance with the credentials for all data sources you plan to use, then export those credentials to a file for reuse.

   NOTE: We recommend that you verify all data source credentials are correct before you export them for reuse.

4. Import the data source credentials to each secondary and backup instance one at a time.

3 Installing the Cloud Bridge Agent

Before you install and configure your Cloud Bridge Agent host servers, ensure that you have reviewed the server requirements and planned your environment for failover. For more information, see the following topics:

- “Understanding Requirements” on page 11
- “Planning for High Availability” on page 13

Cloud Bridge requires that you install required components in a specific order. You must set up the CBA server with a supported Linux operating system and a supported Docker or Podman container before you can install the Cloud Bridge Agent. For more information, see the following topics:

- “Setting Up the Agent Server” on page 19
- “Installing Docker or Podman” on page 20
- “Installing the Cloud Bridge Agent” on page 21
- “Upgrading Your CBA Environment” on page 24

Setting Up the Agent Server

Before you install the Cloud Bridge Agent, you must set up a Linux server with a supported operating system and a Docker or Podman environment. You can use either a virtual machine or a physical server. The following steps assume you are performing the most common installation, where the Cloud Bridge Agent is running on premises as a guest virtual machine.

IMPORTANT: If your current environment requires the Cloud Bridge Agent to be running in a cloud environment such as AWS, Azure, or GCP, you do not need to download an installer image, because you can create a new instance directly with those operating systems.

To set up the Agent server:

1. Download an installer image of a supported operating system. For more information, see the documentation for the selected operating system.
2. Create a new virtual machine that meets all specified requirements. For more information, see the documentation for the selected virtual environment.
3. Install your preferred operating system on your newly created virtual machine.
4. Verify that your server has access to the internet as well as to your internal authentication repositories and applications.
5. Depending on your chosen operating system, complete additional steps to install Docker or Podman as follows:
   - “Setting Up Ubuntu with Docker” on page 20
   - “Setting Up SUSE Linux with Docker” on page 20
   - “Setting Up RedHat Linux with Podman” on page 20

Installing Docker or Podman

After you install a supported operating system on your Cloud Bridge Agent host server, you must complete additional steps to install a Docker or Podman environment for the Agent.

Setting Up Ubuntu with Docker

Set up Ubuntu with Docker using the instructions provided on the Docker documentation site. If any of the commands fail, it is most likely because of formatting issues. We recommend that you copy the lengthy commands directly from the online Ubuntu Docker documentation (https://docs.docker.com/engine/install/ubuntu/).

After you have set up your Docker environment, you can proceed with installing the Cloud Bridge Agent. For more information, see “Installing the Cloud Bridge Agent” on page 21.

Setting Up SUSE Linux with Docker

Set up SUSE Linux with Docker using the instructions provided on the Docker documentation site. If any of the commands fail, it is most likely because of formatting issues. We recommend that you copy the lengthy commands directly from the online OpenSUSE, Docker, and Docker-Compose documentation (https://documentation.suse.com/sles/15-SP2/html/SLES-all/cha-docker-installation.html).

After you have set up your Docker environment, you can proceed with installing the Cloud Bridge Agent. For more information, see “Installing the Cloud Bridge Agent” on page 21.

Setting Up RedHat Linux with Podman

Complete the following steps to set up RedHat Linux with Podman.

1. Ensure that your RHEL 8.3 Server is registered with RedHat.
2. Check the installed Podman version. Run:

   ```
   podman version
   ```

3. (Conditional) If you already have Podman 1.6.4 or later, you do not need to install it.
4. (Conditional) If you do not have Podman installed, run:

   ```
   yum install podman
   ```

5. After you have set up your Podman environment, you can proceed with installing the Cloud Bridge Agent. For more information, see “Installing the Cloud Bridge Agent” on page 21.

Installing the Cloud Bridge Agent

This section provides instructions for installing the Cloud Bridge Agent (CBA) in a new environment. To upgrade an existing CBA environment, see "Upgrading Your CBA Environment" on page 24.

Before you can install the Cloud Bridge Agent, you must have already installed a supported Linux operating system and a Docker or Podman environment on the host server where you plan to install the CBA. For more information, see:

- "Setting Up the Agent Server" on page 19
- "Installing Docker or Podman" on page 20

You can install the Cloud Bridge Agent anywhere on the Docker or Podman host server, but we recommend that you use a standard installation location for each CBA instance. Wherever you install the CBA, the installation script installs an `agent` directory. The `agent` directory contains the scripts, the `.env` file containing the encryption Key and IV values, and additional directories as follows:

- `agent/conf` contains the `bridge-agent.yml` and `DataCenter.json` files
- `agent/log` holds the rolling log files
- `agent/bridgelib` is where you put additional `connector.jar` files (or updated connectors)

Installing Your Primary CBA Instance

You can install a single CBA instance in your environment, but we recommend also installing secondary and backup CBA instances to ensure high availability for your users. Depending on the size of your organization, you might need to set up more than one site. For more information, see "Planning for High Availability" on page 13.

IMPORTANT: Ensure that you download the installation script within the time window that the SaaS operations team specified for your installation. After this time, the script will no longer be available for download and you will have to request another script.

---

1. After you receive the CBA download instructions from the SaaS operations team, open a command line and navigate to the folder where you want to install the CBA.
2. Copy and paste the provided `curl` command, then press **Enter**.

   This command downloads and runs the installation script specific to your organization.
3. At the prompt, specify the desired role for the CBA instance as follows:
   - Enter `0` if you want the instance to be the Primary instance. If you just press **Enter**, the installer defaults to Primary.
   - Enter `1` for Secondary.
   - Enter `2` for Backup.

   NOTE: If you need to make any changes to your CBA instance or site settings at a later time, you can rerun the installation script. For more information, see "Updating High Availability and Other CBA Settings" on page 41.

   ---
4. At the prompt, specify the desired role for the CBA site as you did for the CBA instance.

The installer displays a generated Instance ID for the CBA consisting of the host name and random letters, but you can change this ID to a more meaningful name.

5 (Optional) Type your desired Instance ID and press **Enter** to save it.

The installer then checks whether Docker or Podman is installed and displays the version.

6 At the prompt, enter the user name for the CBA administrator (`cbagent`) and set a password.

You will use these credentials to log in to the CBA console and add credentials for your data source.

The script then installs the Cloud Bridge Agent. When installation is complete, the CBA comes up and sends a heartbeat to the SaaS operations center.

7 Log in to Advanced Authentication and perform the following steps:

7a Configure an external repository to an on-premises LDAP source. For more information, see "Adding a Cloud Bridge External Repository" (https://www.netiq.com/documentation/advanced-authentication-64/tenant-administrator-guide/data/t4donma2ncp4.html) in the *Advanced Authentication SaaS Administration Guide*.

IMPORTANT: If you are setting up a Cloud Bridge external repository in Advanced Authentication for an IGA tenancy, use the format *TENANT_ID*_AA_ER for the external repository name, where *TENANT_ID* is in uppercase.

7b Copy the data source connection credential ID from the Advanced Authentication UI. You will need this credential ID in Cloud Bridge.

8 (Conditional) If you have Identity Governance as a Service, log in to Identity Governance, configure a data source connection, then copy the unique ID for that connection. For more information, see "Collecting Data Using Cloud Bridge" (https://www.microfocus.com/documentation/identity-governance-and-administration/igaas/user-guide/t4h6p598xpxq.html) in the *Identity Governance as a Service User and Administration Guide*.

9 (Conditional) If you have other NetIQ SaaS products that you want to use with Cloud Bridge, log in to those products and configure data source connections as described in the documentation for those products.

10 In a supported browser, go to the Cloud Bridge Agent URL: `http://localhost (CBA IP address or DNS name):8080`.

11 Log in to the Cloud Bridge Agent console using the following credentials:

User name: `cbagent`

Password: The password that you set when you ran the installation script

12 (Optional) On the **Dashboard** tab, verify that the site and instance roles you set for the CBA during installation are correct.

NOTE: The **Instance Id** is the generated identifier or the name that you specified for the *current* CBA instance.

The **Target Id** is the name of the *primary* CBA instance in a high availability environment. It is the instance with which the Cloud Bridge Client communicates. The Instance Id and the Target Id might be the same if you are currently viewing the primary CBA instance or if you have only one CBA instance in your installation.

---

13 Click the **Data Source Management** tab and add your data source connection credentials. For more information, see "Add Credentials for Data Source Connections" on page 27.

14 (Optional) In Advanced Authentication, click the **Test** button to verify that the data source connection works.

After you have installed your primary CBA instance, consider installing additional CBA instances for high availability. For more information, see "Installing Secondary and Backup CBA Instances" on page 23.

For more information about using Cloud Bridge with other OpenText SaaS products, see the following resources:

- *Identity Governance and Administration as a Service Quick Start* (https://www.microfocus.com/documentation/identity-governance-and-administration/igaas/quick-start/quick-start.html)
- *Identity Governance as a Service User and Administration Guide* (https://www.microfocus.com/documentation/identity-governance-and-administration/igaas/user-guide/front.html)
- *Advanced Authentication SaaS Administration Guide* (https://www.netiq.com/documentation/advanced-authentication-64/tenant-administrator-guide/data/bookinfo.html)
- Other product documentation on the Identity and Access Management Services Documentation web page (https://www.microfocus.com/documentation/identity-and-access-management/iam-services/)

Installing Secondary and Backup CBA Instances

After you have installed your primary CBA instance, you can install secondary and backup instances following the same steps that you used for your primary CBA instance. For more information, see "Installing Your Primary CBA Instance" on page 21.

Each instance of the CBA configuration in your high availability environment must use the same encryption IV and Key values. Using the same values on all CBA instances ensures that no disruptions will occur during failovers. For more information about encryption keys and IVs, see "Understanding Encryption IV and Key Values" on page 15.

NOTE: If you have a scenario where you need to back up credentials on an existing CBA and plan to reinstall the CBA, you should make note of the original key and IV from the old CBA before you uninstall it. When you import the credentials into the newly installed CBA, you will need to provide the key and IV from the old CBA.

To reuse the encryption key and IV from the first CBA instance:

1 On your primary CBA instance, locate and open the `.env` file. Copy the encryption key and IV from this file.

2 Install your secondary CBA instance.

The installation script creates the `/agent` directory on the secondary instance, but the instance has its own encryption key and IV at this point.

3 On the secondary CBA instance:

3a Enter the following command to stop the instance:

```
sh <CBA_install_dir>/agent/stop.sh
```

3b Open the `agent/.env` file:

```
vi <CBA_install_dir>/agent/.env
```

3c Replace the KEY and IV entries in the `.env` file with the KEY and IV values from the primary CBA instance.

4 Because the second instance created the Docker container on the installation, it must be replaced for the new key and IV to be used. Enter the following commands to replace the pod definition with the updated key and IV properties:

```
sh <CBA_install_dir>/agent/remove.sh
```

```
sh <CBA_install_dir>/agent/create.sh
```

5 Start the secondary instance back up with the start script:

```
sh <CBA_install_dir>/agent/start.sh
```

The second instance initializes, and Cloud Bridge recognizes both CBAs as running in a high availability system.

NOTE: If you need to change your CBA instance or site settings at any point, you can rerun the installation script and make updates. For more information, see "Updating High Availability and Other CBA Settings" on page 41.

Upgrading Your CBA Environment

You can upgrade an existing Cloud Bridge Agent instance by running the installation script on the host server. The installation script detects that an older version is already installed and prompts you to upgrade.

However, *before you upgrade*, you should determine:

- Whether to replace the connectors you currently have installed on the CBA host server. For more information, see "Planning to Upgrade Connectors" on page 24.
- Whether you want to make changes in your environment for high availability. For more information, see "Planning for High Availability Before Upgrade" on page 25.

Planning to Upgrade Connectors

Before you begin upgrading your CBA environment, review your existing connectors and take the appropriate action as outlined below. When you install the Cloud Bridge Agent, the connector `.jar` files that reside in the `<CBA_install_dir>/agent/bridgelib/` folder on the CBA host server will take precedence over the connector `.jar` files that are included in the `/collectors` folder.

IMPORTANT: Work with your SaaS support team to address any questions or concerns you have about upgrading connectors.

Consider the following:

- If you have older patched connectors in the `/bridgelib` folder and the newer CBA has better built-in `.jar` files, you should remove the outdated files in the `/bridgelib` folder before you install the CBA. Remove any copies of dist-collectors, daas-ldap, or any other updated `.jar` files provided by your SaaS support team.
- If your `/bridgelib` folder contains files such as a supporting `.jar` file for JDBC or other custom collector `.jar` files, you should leave those files alone.
- If you have older connectors in your `/bridgelib` folder that are working well and you do not want to risk upgrading them, you might actually choose to keep an older version in `/bridgelib` and not use the updated version included with the CBA in the `/collectors` folder.

Planning for High Availability Before Upgrade

Even if you have a small organization with a single data center, we recommend that you install and configure one or more secondary or backup CBA servers for failover. For more information to help you plan, see "Planning for High Availability" on page 13.

Consider the following possible high availability upgrade scenarios:

- You want to upgrade your environment with your *current* CBA as your primary CBA instance. For more information, see "Upgrading with Your Current CBA as the Primary Instance" on page 25.
- You want to upgrade your current CBA instance, but want to use a *different* server for your primary CBA. For more information, see "Upgrading with a New CBA as the Primary Instance" on page 26.

NOTE: You can reconfigure your CBA instances as needed after installation by rerunning the installation script on the appropriate servers. For more information, see "Updating High Availability and Other CBA Settings" on page 41.

Upgrading with Your Current CBA as the Primary Instance

If you want to upgrade your CBA environment and assign your current CBA as your primary instance, complete the following steps:

1. Run the CBA installation script on your CBA server. For more information, see "Installing Your Primary CBA Instance" on page 21.
2. At the upgrade prompt, enter `y`.

   IMPORTANT: Cloud Bridge saves your CBA credentials when you upgrade. If you respond `n` at the upgrade prompt and do a fresh installation, you lose those credentials.

   ---

3. Set the instance and site to `Primary`.

   Wait for the server to restart.
4. Run a fresh installation for each secondary and backup CBA that you want to use in your environment. Ensure that you use the encryption key and ID from your first CBA instance for all secondary and backup CBA instances. For more information, see "Installing Secondary and Backup CBA Instances" on page 23.

5 Configure your primary CBA instance with the credentials for all data sources you plan to use, then export those credentials to a file. For more information, see “Add Credentials for Data Source Connections” on page 27.

NOTE: We recommend that you verify that all data source credentials are correct before you export them for reuse.

---

6 Import the data source credentials to each secondary and backup CBA instance one at a time.

Upgrading with a New CBA as the Primary Instance

If you are upgrading your CBA environment and plan to use a new host server as your primary CBA instance, complete the following steps:

1 Configure the host server that you plan to use as your primary CBA instance with all of the CBA prerequisites. For more information, see “Understanding Requirements” on page 11 and “Setting Up the Agent Server” on page 19.

2 Run the CBA installation script on the new host server and set the instance and site to `Primary`. Since this is a *new* installation, there is no prompt to upgrade. For more information, see “Installing Your Primary CBA Instance” on page 21.

   Wait for the server to restart.

3 Run the CBA installation script on your *existing* CBA server, enter `y` at the upgrade prompt, and set the instance and site to `Secondary` or `Backup` as appropriate for your environment.

4 Run a fresh installation for any additional secondary and backup CBAs that you want to use in your environment. Ensure that you use the encryption key and ID from your first CBA instance for all secondary and backup CBA instances. For more information, see “Installing Secondary and Backup CBA Instances” on page 23.

5 Configure your primary CBA instance with the credentials for all data sources you plan to use, then export those credentials to a file. For more information, see “Add Credentials for Data Source Connections” on page 27.

NOTE: We recommend that you verify that all data source credentials are correct before you export them for reuse.

---

6 Import the data source credentials to each secondary and backup CBA instance one at a time.

4 Configuring Cloud Bridge

After you install your Cloud Bridge Agent (CBA) and configure your external repository and data source connection, you must add credentials for the data source connection in the CBA UI.

View Cloud Bridge Agent Details

Cloud Bridge Agent > Dashboard

Use this page to view details of the current CBA instance, including its site and instance priority in your high availability environment.

You cannot make any changes to the CBA on this page, but you can rerun the installation script if you need to make changes. For more information, see Updating High Availability and Other CBA Settings.

Add Credentials for Data Source Connections

Cloud Bridge Agent > Data Source Management

Use this page to add, view, modify, or delete credentials for your data source connections. You can also export and import credentials from one CBA to another to reuse them in a high availability environment.

Click column headings to change the item sort order, or click the hamburger menu on any column to access additional viewing and filtering options:

- On the **Columns** tab, select or deselect the columns you want to display
- On the **General** tab, pin or autosize any or all columns
- On the **Filter** tab, use Boolean operators to filter the data

Click anywhere outside the dialog box to close it so you can view the data you wanted.

To undo your changes at any time and revert to the default view, click the hamburger menu to reopen the dialog box and click **Reset**.

NOTE: Credential import and export are not available in pre-1.9.0 Cloud Bridge versions. So, if you are upgrading from Cloud Bridge 1.8.1, you must manually enter credentials for your first CBA instance. You can then export and import credentials to additional CBA instances.

To add new credentials:

1. Click the plus (+) sign.
2. In the **Unique ID** field, paste the data source connection ID you copied from the application to which you want to connect. For example, for Advanced Authentication, paste the data source connection credential Id you copied in Step 7b on page 22.
3. (Optional) In the **Description** field, specify a unique description of the data source to help you easily identify it.
4. In the **Username** field, specify the service account ID. For LDAP accounts, specify the full DN. For example, `CN=svc-id-admin,CN=Users,DC=support,DC=test`.

   NOTE: This field is limited to 255 characters.

5. In the **Password** field, enter the password for the service account.

   NOTE: This field is limited to 255 characters.

6. In the **Ordinal** field, specify the appropriate ordinal for the authentication method. This value varies depending on whether you are entering credentials for the IDM AE Permission collector or one of the SCIM collectors and fulfillers. For information about ordinals, see the documentation for the data source for which you are adding credentials.
7. Click **Create**.
8. Repeat these steps to add credentials for other identity or application collectors in Identity Governance or external repositories in Advanced Authentication.

You can change these credentials anytime if your service account or password has changed. Click the credential name in the **Unique Id** column to open the Update Credential window where you can make changes.

To export or import existing credentials:

1. (Conditional) If this is the first CBA in your environment and you plan to install additional CBAs for high availability, export your data source credentials as follows:
   - **1a** Click the **Export** icon.
   - **1b** Provide a name for the credentials file with a `.json` extension, then click **Download**.
2. (Conditional) If you previously saved data source credentials and want to reuse them now, import them as follows:
   - **2a** Click the **Import** icon.
   - **2b** Click **Select File to Import** and locate the `.json` file you previously saved to the Downloads folder, then click **Import**.

IMPORTANT:
- If you recreate your Advanced Authentication external repositories, your Data Source Connection credential ID will change and you will need to delete the previous credential and add the new credential.
- If you delete and recreate any in-use data source connection credential ID in Identity Governance, the associated collector will stop communicating through the CBA and you will need to enter the new credential in the CBA using the steps above.

View Connector Information

Cloud Bridge Agent > Connectors

Use this page to view detailed information about the connectors that are loaded on the CBA instance. You cannot make changes to connectors on this page, but you can customize your view of the page.

Click column headings to change the item sort order, or click the hamburger menu on any column to access additional viewing and filtering options:

- On the **Columns** tab, select or deselect the columns you want to display

  For example, this page displays six columns by default, but you can add several optional columns such as Loaded Index, Local Checksum, or Data Source Service.
- On the **General** tab, pin or autosize any or all columns
- On the **Filter** tab, use Boolean operators to filter the data

Click anywhere outside the dialog box to close it so you can view the data you wanted.

To undo your changes at any time and revert to the default view, click the hamburger menu to reopen the dialog box and click **Reset**.

Configure CBA Logging

Cloud Bridge Agent > Logger Configuration

The default CBA logging settings are appropriate for standard installations. However, you can customize these settings if necessary to:

- Add or remove loggers using the plus (+) sign or trashcan icon
- Edit the logging level for any logger to collect more or less data
- Reset all loggers to their default collection levels
- Clear logs
- Modify logging filters to limit what the CBA console displays
- Customize the default view

To customize your view of this page:

- Click column headings to change the item sort order
- Click the hamburger menu on any column to access additional viewing and filtering options:
  - On the **Columns** tab, select or deselect the columns you want to display
  - On the **General** tab, pin or autosize any or all columns
  - On the **Filter** tab, use Boolean operators to filter the data
- Click anywhere outside the dialog box to close it so you can view the data you wanted

To undo your changes at any time and revert to the default view, click the hamburger menu to reopen the dialog box and click **Reset**.

To change the logging level of a logger file:

1. In the Package/Class column, click the name of the logger.
2. In the Update Logger dialog box, select the appropriate log level.
3. Click **Update**.

5 Configuring Secure Communication for Your Cloud Bridge Agent

This section provides an introduction to the essential components of secure communication, instructions for configuring HTTPS connections in your Cloud Bridge Agent environment, and reference information about the CBA Transport Layer Security (TLS) policy.

Understanding the Components of Secure Communication

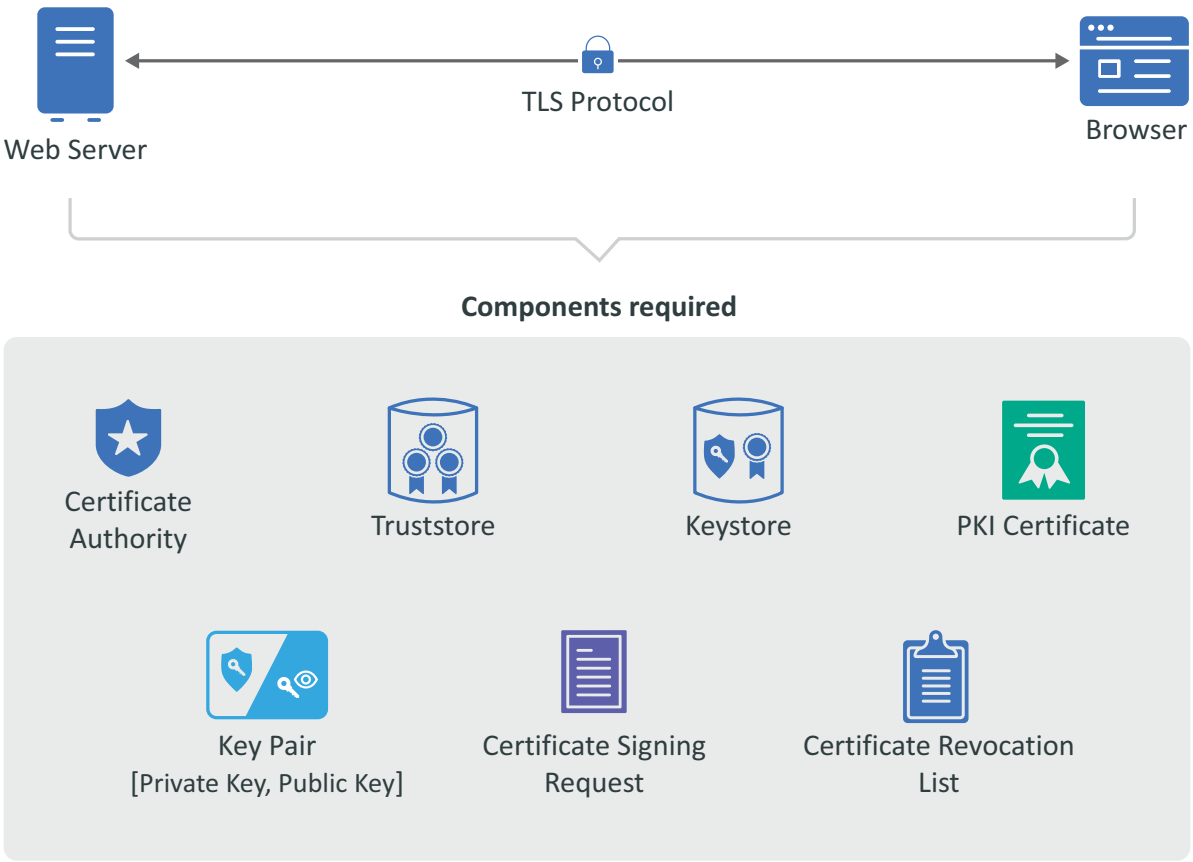
Configuring secure communication between your OpenText products, infrastructure products, and any administrative tools requires that you have a good understanding of the components that allow the secure communication to occur. OpenText products use industry standards of X.509 certificates, public key infrastructure (PKI), and transport layer security (TLS).

This section provides a basic introduction to these components. For more detailed information, see the following resources:

- Internet X.509 Public Key Infrastructure Certificates and Certificate Revocation List (CRL) Profile (https://datatracker.ietf.org/doc/html/rfc5280)
- The Transport Layer Security (TLS) Protocol Version 1.2 (https://datatracker.ietf.org/doc/html/rfc5246)
- The Transport Layer Security (TLS) Protocol Version 1.3 (https://datatracker.ietf.org/doc/html/rfc8446)

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

Figure 5-1 *Components of Secure Communication*



The secure communication occurs between a server and a client. In Figure 5-1, the web server represents the server and the browser represents the client. The following are important terms that you need to understand to create a secure connection between a server and a client.

- **Certificate Authority:** 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 generates self-signed certificates through other products such as openSSL, eDirectory, and Active Directory. You can create self-signed certificates through the certificate authorities in these other products to use in test environments.

  NOTE: A security best practice is to use a well-known CA to issue certificates in production environments.

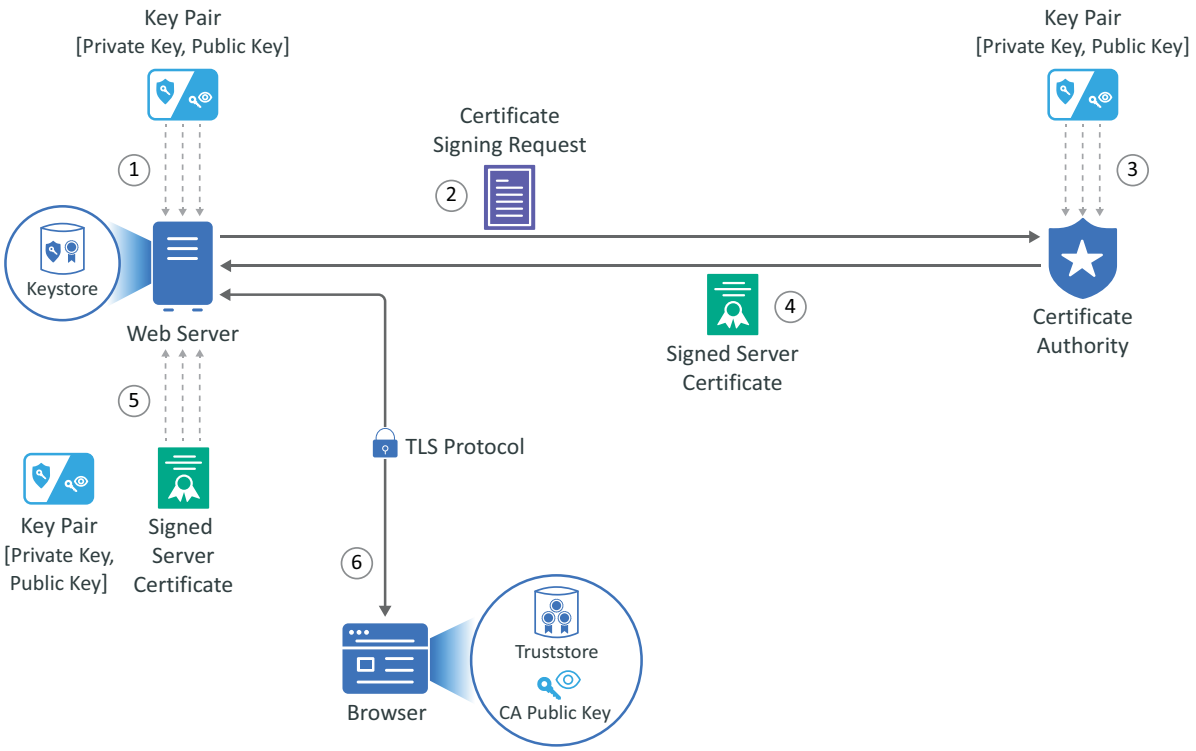
- **Public Key Infrastructure (PKI) Certificates:** Digital certificates that the CA issues that prove ownership of the certificate. The CA can issue certificates for users, applications, or devices. PKI certificates contain the following information:
  - Version number
  - Unique serial number

    - CA digital signature and algorithm used
    - Validity period
    - Certificate usage
    - Subject name, URL, and email address
    - Public and private keys (sometimes only the public key)
- **Key Pair:** 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 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 (CRL):** A list that the CA creates and manages that contains a list of unique serial numbers that it has revoked. The CA uses the certificate revocation list (CRL) to deny requests from any user, application, or device that has a serial number listed on the CRL.
- **Certificate Signing Request (CSR):** 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 tool 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 tool to set and manage the TrustStore passwords.
- **Transport Layer Security (TLS) Protocol:** The secure protocol created by all of the components defined in this section. It allows the server and client to communicate securely using certificates and key pairs to prove identity on the server and client.

Example of Establishing Secure Communication for a Web Server

When you install a web server, the communication is not secure by default. Or, 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 communication with any client.

Figure 5-2 *Obtaining a Signed Server Certificate from a Well-known Certificate Authority*



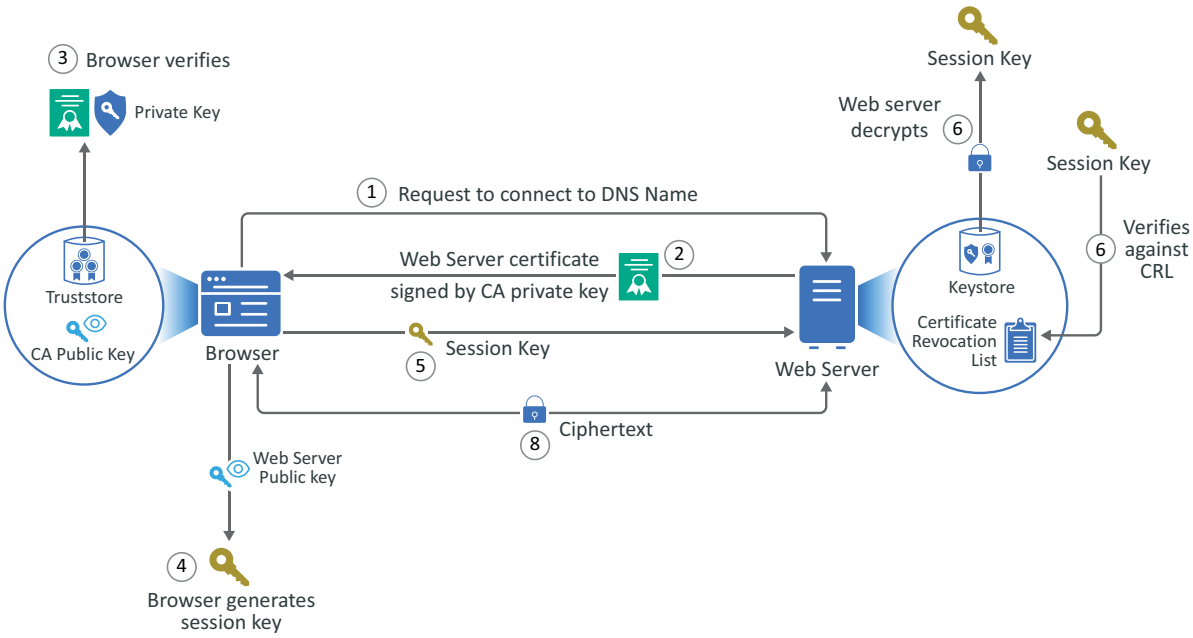
In this example, Adam the administrator requests a signed server certificate from the well-known CA and uses the certificate to establish secure communication with a client that is a web application.

1. Adam generates a key pair on the web server using keytool. Adam uses 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 identify the web server.
2. Adam sends the CSR containing the web server's information to a well-known CA such as DigiCert.
3. The CA uses the CSR to generate a server certificate for the web server. 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 accesses 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. The browser now trusts any communication coming from this web server. For more information, see "Example of a Secure Handshake for the Client" on page 35.

Example of a Secure Handshake for the Client

The following example shows how a server receives a certificate from a well-known certificate authority (CA) to be able to communicate securely with any client. This example shows 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.

Figure 5-3 *A Browser Establishes a Secure Communication Channel to a Web Server*



In this example, Adam the administrator logs in to the administration console that is a web application. Every action except for Adam entering the URL of the web browser happens automatically between the browser and the web server. No user interaction is required.

1. Adam enters the URL into the browser. The browser sends a request to connect to the fully qualified DNS name 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 accesses 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 (CRL). 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, then uses the session key to encrypt data, and finally sends the data back in ciphertext. This secure communication continues until the session ends.

Enabling HTTPS Connections to Your Cloud Bridge Agent

In some instances, a company's security policy for on-premises applications requires that all HTTP connections be protected by the HTTPS protocol. This section describes how to generate a self-signed HTTPS certificate and import it into a Java keystore, and how to enable the Cloud Bridge Agent (CBA) to use that keystore to secure HTTP connections to the CBA user interface and APIs. The use of a self-signed certificate should be a temporary solution, and OpenText recommends that you take steps to replace the initial, self-generated certificate with a purchased certificate from a reputable certificate authority as soon as possible.

Prerequisites

Before you enable HTTPS for the Cloud Bridge Agent, verify that you have taken the following steps:

1. You have received and executed the installation script for the Cloud Bridge Agent.
2. The Cloud Bridge Agent is up and running on the Docker or Podman host server.
3. You have the appropriate rights to the host server file system to add and edit some files.
4. You know the location of the Cloud Bridge Agent installation on the host server.
5. The host server has Java 11 keytool installed. You can check your Java version using the `java -version` command.

Enabling HTTPS

The process of enabling HTTPS includes the following steps:

1. Generate the new certificate and keystore.
2. Enable HTTPS connections.
3. Verify HTTPS connectivity.

Generate the New Certificate and Keystore

When you install the Cloud Bridge Agent, the `agent/conf/` directory is created below the installation location <*`CBA_install_dir`*>.

From the installation location:

1. Use the `cd` command to navigate to the <*`CBA_install_dir`*>`/agent/conf` directory.
2. Using keytool:
   - **2a** Generate the new certificate and Java keystore.

     ```
     keytool -genkey  -keyalg RSA -alias localhost -keystore
     bridge.keystore -storepass changeme -validity 360 -keysize 2048
     ```

   - **2b** Enter the appropriate information when prompted.

     When completed, a new file named `bridge.keystore` is created in the directory.
3. (Optional) If you are requesting a Certificate Authority (CA) signed (paid) certificate:
   - **3a** Using keytool, generate a key pair and use it to create a certificate signing request (CSR):

```
keytool -certreq -alias localhost -keystore bridge.keystore -
storepass changeme -file mycert.csr
```

The CSR contains the fully qualified DNS name of the server, the key pair, and other such information to help identify the web server.

3b Send the CSR to a well-known CA such as DigiCert.

The CA uses the CSR to generate a server certificate for the web server. 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. The CA sends the signed web server certificate back to you.

3c When you get the real certificate from the CA, import it into the `bridge.keystore` using keytool.

When a browser accesses 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. The browser now trusts any communication coming from this web server.

For more information about certificates, see "Understanding the Components of Secure Communication" on page 31.

Enable HTTPS Connections

To allow HTTPS connections to the CBA user interface and REST endpoints, you must edit the <*CBA_install_dir*>`/agent/conf/bridge-agent.yml` file. You must also update the container definition in the `create.sh` script file to allow connections to the HTTPS ports.

1 Edit the `bridge-agent.yml` file:

1a We recommend that you copy the `bridge-agent.yml` file to something like `bridge-agent.yml.bak` in case you need to restore the file.

1b For both the `applicationConnectors:` and `adminConnectors:` sections in the file, remove the comment indicators (#) for the `– type:`, `port`, and `keystore` entries.

1c Save the changes.

2 Edit the `create.sh` file:

2a Use the `cd` command to return to the <*CBA_install_dir*>`/agent/` directory.

2b Edit the `create.sh` script file:

2b1 Duplicate the entry for port mapping `-p "8080:8080" -p "8081:8081" \`

2b2 Copy this line into the file. Change the ports in this new line from `"8080:8080"` to `"8443:8443"` and `"8081:8081"` to `"8444:8444"`.

These will be the HTTPS connection ports.

3 Stop the Cloud Bridge Agent, recreate the container with the new ports mapped, then start the Cloud Bridge Agent.

3a From the <*CBA_install_dir*>`/agent` directory, stop the Agent:

```
sh stop.sh
```

3b Remove the bridge-agent container:

```
sh remove.sh
```

3c Recreate the container:

```
sh create.sh
```

3d (Conditional) If the `sh create.sh` command did not start the container, start the container:

```
sh start.sh
```

Verify HTTPS Connectivity

You should now be able to connect to the Cloud Bridge Agent user interface using the following URL:

`https://<hostname>:8443/`

Understanding the Cloud Bridge Agent TLS Security Policy

This section provides reference information for the Cloud Bridge Agent Transport Layer Security (TLS) policy. TLS is the successor to Secure Sockets Layer (SSL).

TLS Settings

Java ships with a security policy codified in the `java.security` policy file. It is typically located in the `conf/security` folder relative to `$JAVA_HOME`. Cloud Bridge containers are based upon the CAFapi Java 11 image. The CAFapi Java 11 container image disables weak TLS cipher suites in the `disableWeakTlsAlgorithms.patch` patch file. This is an excellent, if strict, security posture and meets the OpenText security requirements. However, some customers have had difficulty connecting with other resources on their respective networks. So, we have developed and tested a relaxed or "intermediate" security policy. This policy restores some of the ciphers disabled by the CAFapi team and further introduces key length limitations. The patch in its entirety appears below, but the following sections provide additional explanation.

Terminology

This section uses the following terms:

- RSA (Rivest Shamir Adleman)
- ECC (Elliptic Curve Cryptography)
- Symmetric Cryptography (for example, AES)
- Asymmetric Cryptography (Public/Private Key pair)
- DH & ECDH (Diffie-Hellman & Elliptic-Curve Diffie-Hellman)
- Hash Function (for example, SHA1, SHA256, SHA512)
- CA (Certificate Authority)
- Host Certificate (Certificate used for a web server)

Disabled Algorithms

Ideally, we would like to restrict communication to TLS 1.3. However, we must also make it possible to use the TLS 1.2 protocol safely, which causes us to exclude cipher suites that are available but should not be used.

Highlights

1. Remove the TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256 cipher from the list.
2. Increase RSA key size: “RSA keySize < 2048”

Key Lengths

- RSA less than 2048-bit
- Diffie-Hellman less than 1024-bit
- Elliptic Curve less than 224-bit

The following algorithms are disallowed and will not successfully negotiate the "handshake" process:

- Camellia 128-bit with Cipher Block Chaining (CBC)
- AES 256-bit with CBC
- AES 128-bit with CBC
- TLS DH DSS with AES 256-bit GCM SHA384 (under translation...)

Intermediate Java Security Policy

You can achieve “intermediate” level security by disabling the following suites and enforcing acceptable key sizes:

```
--- java.security  2021-10-05 06:34:16.000000000 -0500
+++ java.security.intermediate  2022-02-15 14:22:30.878689700 -0600
@@ -753,25 +753,20 @@
 #   jdk.tls.disabledAlgorithms=MD5, SSLv3, DSA, RSA keySize < 2048, \
 #       rsa_pkcs1_sha1, secp224r1
jdk.tls.disabledAlgorithms=SSLv3, TLSv1, TLSv1.1, RC4, DES, MD5withRSA, \
-    DH keySize < 1024, EC keySize < 224, 3DES_EDE_CBC, anon, NULL, \
+    RSA keySize < 2048, DH keySize < 1024, EC keySize < 224, 3DES_EDE_CBC,
anon, NULL, \
     CAMELLIA_128_CBC, AES_256_CBC, AES_128_CBC, \
     DES40_CBC, RC4_40, CAMELLIA_256_CBC, DES_CBC, \
     SEED_CBC, RC4_56, RC4_128, IDEA_CBC, RC2_CBC_40, \
     TLS_DH_DSS_WITH_AES_128_GCM_SHA256, \
-    TLS_DH_DSS_WITH_AES_256_GCM_SHA384, \
-    TLS_DH_RSA_WITH_AES_128_GCM_SHA256, \
-    TLS_DH_RSA_WITH_AES_256_GCM_SHA384, \
+    TLS_DH_DSS_WITH_AES_256_GCM_SHA384, \
     TLS_DHE_DSS_WITH_AES_128_GCM_SHA256, \
```

```
     TLS_DHE_DSS_WITH_AES_256_GCM_SHA384, \
     TLS_DHE_PSK_WITH_AES_128_GCM_SHA256, \
     TLS_DHE_PSK_WITH_AES_256_GCM_SHA384, \
-    TLS_RSA_WITH_AES_256_GCM_SHA384, \
-    TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384, \
-    TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384, \
+    TLS_RSA_WITH_AES_256_GCM_SHA384, \
     TLS_DHE_DSS_WITH_AES_256_GCM_SHA384, \
     TLS_RSA_WITH_AES_128_GCM_SHA256, \
     TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256, \
-    TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256, \
     TLS_DHE_DSS_WITH_AES_128_GCM_SHA256, \
     TLS_EMPTY_RENEGOTIATION_INFO_SCSV, \
     include jdk.disabled.namedCurves
```

Additional Resources

For more information about configuring secure communication, see the following resources:

- The ROBOT Attack (https://www.robotattack.org)
- Testing for Weak SSL TLS Ciphers Insufficient Transport Layer Protection (https://github.com/OWASP/www-project-web-security-testing-guide/blob/master/v41/4-Web_Application_Security_Testing/09-Testing_for_Weak_Cryptography/01-Testing_for_Weak_SSL_TLS_Ciphers_Insufficient_Transport_Layer_Protection.md)
- Security/Server Side TLS (https://wiki.mozilla.org/Security/Server_Side_TLS)
- Testing TLS/SSL encryption anywhere on any port (https://wiki.mozilla.org/Security/Server_Side_TLS)

6 Managing the Cloud Bridge Agent

After the Cloud Bridge Agent (CBA) has been installed and configured, it does not typically require a lot of administrative maintenance, and the OpenText SaaS operations team has visibility into the overall health of your CBA. This section provides some basic information to help you maintain your CBA environment, and some troubleshooting tips for common issues.

Uninstalling the Cloud Bridge Agent

To uninstall a Cloud Bridge Agent (CBA) instance:

1. Log in to the host server where the CBA instance is installed.
2. Open a command line and run the following commands:
   a. `cd <CBA_install_dir>`
   b. `sh agent/stop.sh`
   c. `sh agent/remove.sh`
   d. `rm -rf agent/`

Updating High Availability and Other CBA Settings

When you run the CBA installation script the first time, it is saved to the installation folder on the CBA instance. To make changes to any CBA settings, for example to change instance or site role in a high availability environment, you can rerun the `cb_agent.sh` installation script on the affected CBA instances.

To rerun the CBA installation script:

1. Log on to the host server of the CBA instance whose settings you want to change.
2. Run the following command:

   ```
   bash cb_agent.sh 2>&1 | tee -a install.log
   ```

3. Make the desired changes.

   For more information about high availability settings, see "Planning for High Availability" on page 13.

   The CBA instance restarts and applies your changes.

Repeat these steps for all CBA instances whose settings you need to change.

Troubleshooting Common Issues

To begin troubleshooting an error, you can check the main log file for the Cloud Bridge Agent. This file is located at `agent\conf\daas_remote.log`. You can adjust the logging level in the CBA console so this log collects the appropriate amount of data for your environment.

The `daas-remote.yml` file also allows you to configure logging levels. In the second section of the file, appenders can control the format of the log, the number of files, the size, and its rotation behavior.

NOTE: Avoid changing the log level for the `org.apache.kafka` file because it is very verbose and does not provide useful information.

Upgrade Does Not Remove Old Images

The Cloud Bridge Agent upgrade process does not automatically remove older images from Docker or Podman in your environment. This allows you to keep images until you are sure you no longer need them. Perform the following steps when you are ready to clean up old images to free up disk space.

1. Log in to the CBA host server where you need to free up space.
2. Open a command line and run the appropriate command for your environment:
   - Docker: `docker image prune [options]`
   - Podman: `podman image prune [options]`

For more information about available options:

- See the Docker documentation (https://docs.docker.com/engine/reference/commandline/image_prune/)
- See the Podman documentation (https://docs.podman.io/en/latest/markdown/podman-image-prune.1.html)

Podman RHEL 8.3 Does Not Automatically Restart the Container After a Server Restart

To fix this issue and ensure that your container automatically runs on RHEL 8.3 Podman after a server restart, complete the steps in the following procedure. For more information, see https://docs.podman.io/en/latest/markdown/podman-generate-systemd.1.html.

1. Stop your container:

   ```
   podman stop bridge-agent
   ```

2. Use the `podman generate systemd` command to create your system file:

   ```
   podman generate systemd --restart-policy=always --files --name bridge-agent
   ```

3. Locate the file with the name `container-bridge-agent.service`, then copy the file to the system folder:

```
cp container-bridge-agent.service /etc/systemd/system/container-bridge-agent.service
```

4. Enable the service:

```
systemctl enable container-bridge-agent.service
```

5. Start the service:

```
systemctl start container-bridge-agent.service
```

6. Check the status:

```
systemctl status container-bridge-agent.service
```

7. Restart your server to test it. After a server restart, if you run the command `podman ps` and the above steps work, you should find your bridge-agent container running.