

---

# **Micro Focus Fortify Application Defender**

Software Version: 19.1.0

## **On-Premises Installation Guide**

Document Release Date: March 2018

Software Release Date: March 2018



## Legal Notices

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

<https://www.microfocus.com>

## Warranty

The only warranties for products and services of Micro Focus and its affiliates and licensors ("Micro Focus") are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

## Restricted Rights Legend

Confidential computer software. Except as specifically indicated otherwise, a valid license from Micro Focus is required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

## Copyright Notice

© Copyright 2016 - 2019 Micro Focus or one of its affiliates

## Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

## 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>

# Contents

Preface .....	5
Contacting Micro Focus Fortify Customer Support .....	5
For More Information .....	5
About the Documentation Set .....	5
Change Log .....	6
Getting Started .....	7
Intended Audience .....	7
Installation Overview .....	7
Overview: Installing a Single Instance .....	8
Overview: Deploying a Cluster .....	10
Hardware Requirements .....	11
Software Requirements .....	13
Application Defender Installation Package .....	14
Installing Application Defender .....	15
Installing a Single Fortify Application Defender Instance .....	15
[Optional] Encrypting Sensitive Values .....	19
Post Encryption Clean-up .....	20
Redeploying an Installation that Includes Encrypted Values .....	21
Deploying a Fortify Application Defender Cluster .....	21
Configuring a Fortify Application Defender Cluster .....	22
[Optional] Encrypting Sensitive Values .....	27
Post Encryption Clean-up .....	28
Redeployment: Encrypted Values Used .....	28
Advanced Installation Notes .....	29
SMTP Email Server Authentication .....	29
Java Keystore .....	29
All Docker-Compose Files .....	30
Vertica Database .....	33
Postgres Database (Optional) .....	33
Infrastructure Virtual Machine Services .....	34
Application Virtual Machine Services .....	35
Scaling Fortify Application Defender On-Premises Services .....	36
Fortify Application Defender System Hardening .....	37

Logging Policy .....	37
Upgrading from an Earlier Version .....	39
Regenerating Docker Compose Files .....	39
Upgrade Docker Images .....	39
Additional References .....	41
Send Documentation Feedback .....	42

## Preface

### Contacting Micro Focus Fortify Customer Support

If you have questions or comments about using this product, contact Micro Focus Fortify Customer Support using one of the following options.

#### **To Manage Your Support Cases, Acquire Licenses, and Manage Your Account**

<https://softwaresupport.softwaregrp.com>

#### **To Call Support**

1.844.260.7219

### For More Information

For more information about Fortify software products:

<https://software.microfocus.com/solutions/application-security>

### About the Documentation Set

The Fortify Software documentation set contains installation, user, and deployment guides for all Fortify Software products and components. In addition, you will find technical notes and release notes that describe new features, known issues, and last-minute updates. You can access the latest versions of these documents from the following Micro Focus Product Documentation website:

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

## Change Log

The following table lists changes made to this guide.

<b>Software Release-Version</b>	<b>Change</b>
19.1.0	Changed: <ul style="list-style-type: none"> <li>• Support for Docker engine version to 18.09.2 and later</li> </ul>
18.20	Added: <ul style="list-style-type: none"> <li>• Support for encrypting sensitive values in the <code>appdefender.properties</code> file</li> <li>• Agent support for Tomcat 9 and IIS 10</li> </ul> Updated: <ul style="list-style-type: none"> <li>• Docker engine and kernel support for 18.09 and later</li> </ul>
17.1	Added: <ul style="list-style-type: none"> <li>• Support for Docker version 17.05.0</li> <li>• SMTP email server authentication</li> </ul>
16.3	Added: <ul style="list-style-type: none"> <li>• <code>rsyslog_defender</code> service to consume logs from all application services</li> <li>• A new script to print the status of all Fortify Application Defendercontainers</li> <li>• Support for Docker 1.12</li> </ul>

# Getting Started

This document provides administrators with instructions on how to install and run Micro Focus Fortify Application Defender as either a single instance or clustered deployment.

This section contains the following topics:

- [Intended Audience](#) ..... 7
- [Installation Overview](#) ..... 7
- [Hardware Requirements](#) ..... 11
- [Software Requirements](#) ..... 13
- [Application Defender Installation Package](#) ..... 14

## Intended Audience

This document is designed for anyone who wants to deploy Fortify Application Defender on premises. Fortify provides this content based on the assumption that the reader has a basic understanding of hardware and server management.

## Installation Overview

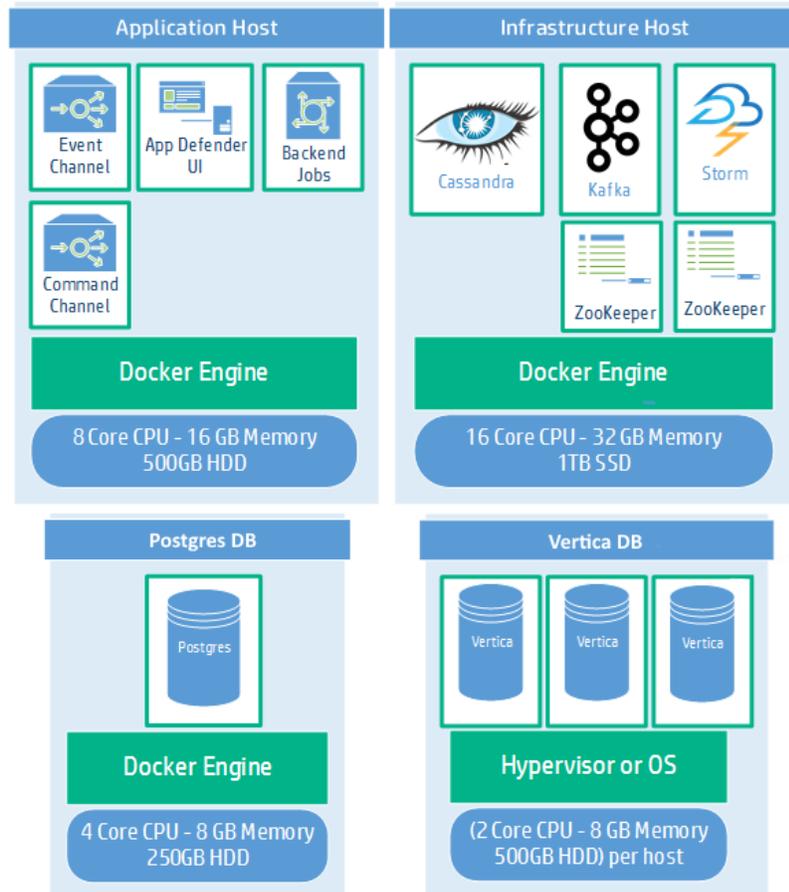
You can install Fortify Application Defender on premises as either a single instance, or a clustered deployment. Of these two options, a clustered deployment provides higher availability, reliability, and scaling.

This section contains the following topics:

- [Overview: Installing a Single Instance](#) ..... 8
- [Overview: Deploying a Cluster](#) ..... 10

## Overview: Installing a Single Instance

The following diagram illustrates a Fortify Application Defender on-premises environment. The minimum recommended deployment consists of an application host, an infrastructure host, a Postgres host, and three Vertica hosts.



### Application Host Components

The application host components are described in the following table.

Component	Description
<b>Event Channel</b>	Secure communication channel between the Fortify Application Defender agent and the service used by agents to send events to the service.
<b>Application Defender UI</b>	Website that all Fortify Application Defender users use to access protect, manage, message and alert functionality.
<b>Backend</b>	Component used to manage and schedule internal back-end jobs,

Component	Description
<b>Jobs</b>	such as reports.
<b>Command Channel</b>	Secure communication channel between the Fortify Application Defender agents and the service used to exchange commands.
<b>rsyslog_defender</b>	syslog container to consume logs from Fortify Application Defender app services. This includes logs for the following services: <ul style="list-style-type: none"> <li>• ui_customer</li> <li>• command_channel</li> <li>• backend_jobs</li> <li>• Edge</li> </ul>

### Infrastructure Host Components

Component	Description
<b>Apache Cassandra</b>	Open-source distributed database that Fortify Application Defender uses to store intermediate data for alerts.
<b>Apache Kafka</b>	Stateless distributed queue used for reports, events, and activity stream processing.
<b>Apache Storm</b>	Distributed real-time stream computation system. Fortify Application Defender uses Storm topologies for notifications, reporting, alerting, reconciliation, and writing events to Vertica.
<b>Apache Zookeeper</b>	Service used to maintain configuration information, naming, distributed synchronization, and group services used by Kafka and Storm.

### Postgres Database Component

Component	Description
<b>Postgres</b>	Object-relational database that stores Fortify Application Defender user data.

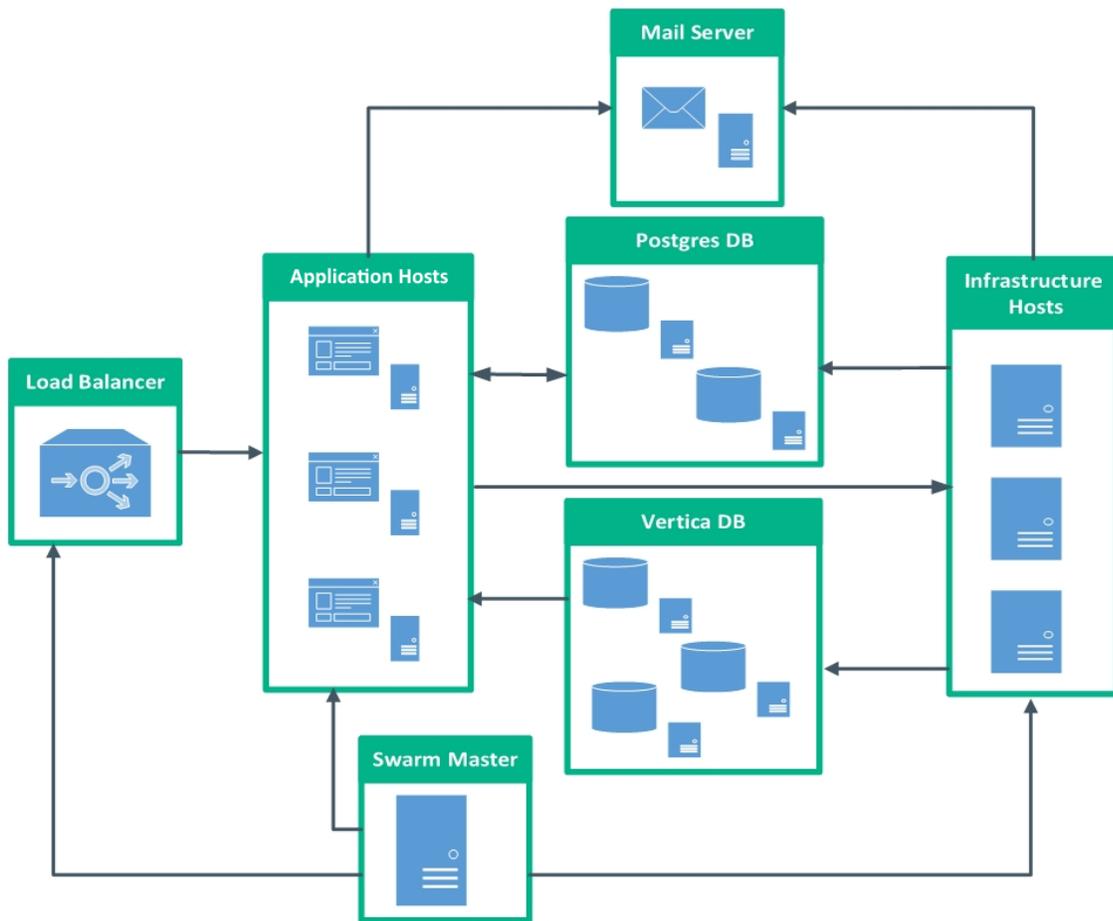
### Vertica Database Component

Component	Description
<b>Vertica</b>	Columnar database that stores event data for Fortify Application Defender. Clustering

Component	Description
	the Vertica hosts makes it easier to scale up the entire system later. Fortify recommends that you have a minimum of three clustered Vertica hosts. Fortify supports, but does not recommend, a single instance.

### Overview: Deploying a Cluster

The following diagram illustrates the clustered Fortify Application Defender on-premises deployment. To perform a clustered deployment, you must have experience with clustering networks.



### Clustered Deployment Components

Component	Description
<b>Application Hosts</b>	List of the nodes designated for application components. Node information includes the following properties: <ul style="list-style-type: none"> <li>• Number assigned to the host (numeric range: <b>1-255</b>).</li> <li>• IP Address</li> </ul>

**Clustered Deployment Components, continued**

Component	Description
	<ul style="list-style-type: none"> <li>• Hostname</li> </ul>
<b>Infrastructure Hosts</b>	<p>List of the nodes designated for infrastructure components. Node information includes following properties:</p> <ul style="list-style-type: none"> <li>• Number assigned to the host (numeric range: <b>1-255</b>).</li> <li>• IP Address</li> <li>• Hostname</li> </ul>
<b>Load Balancer</b>	Load balancer host. Front-end node that provides the interface with which users and agents are to interact.
<b>Mail Server</b>	Application that sends and receives email from local users (users within the same domain) and remote senders.
<b>Postgres Database</b>	Database server. Its primary function is to store data securely and allow for retrieval at the request of other software applications.
<b>Swarm Master</b>	Manages the resources for the entire cluster.

**Hardware Requirements**

An on-premises Installation of Fortify Application Defender requires the hardware described in the following table, and a cluster of three or more Vertica instances.

**Note:** While you can create an installation with a single Vertica instance, Fortify strongly recommends that you deploy a Vertica cluster of three or more instances. Otherwise, because your data is not replicated when you install a single Vertica instance, you risk losing security event data. Moving from a single-node deployment to a clustered Vertica deployment later requires manual data migration.

Component	CPU	Memory	Hard Drive
Application	8 cores	16 GB	500 GB HDD
Infrastructure	16 cores	32 GB	1 TB SSD
Postgres database	4 cores	8 GB	Single Instance: 250 GB HDD Clustered: 500 GB HDD

<b>Component</b>	<b>CPU</b>	<b>Memory</b>	<b>Hard Drive</b>
Vertica (x3)	2 cores	8 GB	500 GB HDD per host

For additional Vertica requirements, see ["Additional References" on page 41](#).

To begin installation, see ["Installing a Single Fortify Application Defender Instance" on page 15](#).

To begin installation, see ["Deploying a Fortify Application Defender Cluster" on page 21](#).

## Software Requirements

The following software requirements apply to both single and clustered installations, except where noted.

### Network Connection

All of the Fortify Application Defender hosts (application, infrastructure, Postgres, and Vertica) need to communicate with each other. Communication ports on the Fortify Application Defender apps server must be open to allow all application servers access to the Fortify Application Defender service. For additional networking and service details, see the diagram in "[Installation Overview](#)" on page 7.

### Docker Hub

A Docker Hub account is needed to access Fortify Application Defender docker images. To gain access to the required Docker repositories, provide your Docker Hub account username to your Fortify Application Defender account team or Fortify technical support representative.

### Firewall Rules

Firewalls on all machines must be configured to allow communication across hosts.

### SMTP Server (mail)

Fortify Application Defender sends an email notifications to each user in the system. Provide a reference to the SMTP server for Fortify Application Defender to use. For more information, see "[Advanced Installation Notes](#)" on page 29.

### Vertica Database Cluster

- Use Vertica documentation to install a Vertica cluster. For links to the Vertica site, see "[Additional References](#)" on page 41.
- Firewall rules must allow application and infrastructure host access.

**Note:** Single node installations are not as reliable and require data migration to grow your Fortify Application Defender installation into a cluster later. This, and other limitations of single-node installations, make them less suitable for use in production environments.

### Postgres Database

- Fortify recommends that you use a Postgres container. To use a Postgres container to start Fortify Application Defender, see the [Installing a Single Fortify Application Defender Instance](#) section for alternate installation information.
- If you have an existing Postgres database, use the `appdefender.sql` file generated when using the generation script to create the appdefender database.

### Linux Machines

Install the following software on your Linux machines:

- **RHEL 7 or CentOS 7:** Kernel version 3.10 or later
- **Docker-engine:** version 18.09.2 or later
- **Docker-compose:** version 1.7.0
- **Python:** version 2.7.11
- **Java:** Openjdk version 7 or 8

For more information about Docker, Postgres, or Vertica, see "[Additional References](#)" on page 41.

## Fortify Application Defender License

You should have received an email that contains your license key and instructions on how to redeem the keys. If you have not received the email, contact Micro Focus Fortify support (<https://softwaresupport.softwaregrp.com>).

## Additional Settings for Clustered Installations

- Network time protocol (ntp) should be running on all nodes.
- Update `/etc/hosts` file with the required IP - host mapping on all nodes.

## Application Defender Installation Package

The Application Defender installation package (AppDefender\_XX.XX.zip) contains the following files:

File Name	Purpose
CertGeneration.tar.gz	Files needed to auto-generate java keystore files.
generate-compose-yaml.py	Use an existing database or use this script to create docker-compose files, haproxy template, and the database creation script.
appdefender.properties (sample)	Used as an argument with the <code>generate-compose-yaml.py</code> script to create different App Defender services.
2018SecurityContent.zip	Package used to populate the App Defender service with the latest security content.
Fortify Application Defender On-Premises Installation Guide	This document.
Vertica OEM license	An open license for Vertica that includes technical support.
ArcSight Enterprise Security Manager (ESM) content	ArcSight Enterprise Security Manager enables Application Logging, Application Protection-specific dashboards, and ESM use cases.

## Installing Application Defender

Fortify offers two on-premises installation options. A single instance installation allows you to start an instance of Fortify Application Defender. It supports a specific number of users and agents determined by your hardware configuration. A clustered deployment [Deploying a Fortify Application Defender Cluster](#) supports high availability, continuous operations or zero downtime deployment, and replication. It scales to support any number of Fortify Application Defender agents.

Begin by selecting the installation that best matches your needs:

<a href="#">Installing a Single Fortify Application Defender Instance</a>	15
<a href="#">[Optional] Encrypting Sensitive Values</a>	19
<a href="#">Post Encryption Clean-up</a>	20
<a href="#">Deploying a Fortify Application Defender Cluster</a>	21
<a href="#">Configuring a Fortify Application Defender Cluster</a>	22
<a href="#">[Optional] Encrypting Sensitive Values</a>	27

## Installing a Single Fortify Application Defender Instance

The single instance on-premises installation is intended for users who have a small number of agents or who need a proof-of-value installation.

Perform the following steps in order:

1. Copy the entire installation package to a folder in your `opt` directory (for example: `/opt/appdefender`) on a Linux machine.

**Note:** You must have read, write, and execute privileges to install Fortify Application Defender.

2. Generate the Java Keystore, as follows:
  - a. Run the `build-stores.sh` script.
  - b. At the prompt, enter one of the following two server certificate options:
    - o For self-signed server certificate generation, enter **1**.  
Self-signed certificate scripts are used with trial or pilot installations.

```
#export PATH=$PATH:$JAVA_HOME/bin
#cd CertGeneration
#chmod 755 build-stores.sh
#chmod 755 server-root-self-signed.sh
```

```
#sh build-stores.sh
#<Press Enter>
```

- o If you have a server certificate signed by valid certificate authority (CA), enter **2**.
- c. Copy the signed server certificate (`server.crt`), server private key (`server.key`), CA intermediate Root cert (`server.int.crt`), and CA Root cert (`server.root.crt`) into the third-party folder.

**Note:** Use the file names provided in parentheses. Rename your files if necessary.

```
#export PATH=$PATH:$JAVA_HOME/bin
#cd CertGeneration
#chmod 755 build-stores.sh
#sh build-stores.sh
#<Type 2><Press Enter>
```

- d. Enter a passphrase (at least six characters long) for the keystore.
- e. Press **ENTER**.

Both options generate the following three files, which are required to start the Fortify Application Defender service:

- `keystore.jks`
  - `truststore.jks`
  - `itemstore.jks`
3. Update the `appdefender.properties` file with the required parameters, as shown in the following example.

```

deploy:single
lb_host:10.100.100.100
apps_host:[['1', '10.100.100.100', 'applications']]
infrastructure_host:[['1', '10.100.100.101', 'infrastructure']]
apps_host_mac_address:00-aa-bb-cc-dd-ee
appdefender_registry:appdefender
defender_logs:/opt/defenderlogs
defender_data:/opt/defenderdata
initial_user_email:test@hpe.com
initial_user_first_name:Application
initial_user_last_name:Defender
initial_tenant_domain:defender.com
initial_tenant_name:SingleInstance
mail_from:single@defender.com
mail_host:mail.server.com
mail_port:25
mail_username:mailusername@hpe.com
mail_password:mailpassword
postgres_ip:10.100.100.121
postgres_dbname:appdefender
postgres_user:username
postgres_password:postgrespassword
vertica_ip:10.100.100.131
vertica_dbname:docker
vertica_user:username
vertica_password:verticapassword
keystore_path:/opt/hpad/serverkeys/keystore.jks
keystore_password:keystorepassword
truststore_path:/opt/hpad/serverkeys/truststore.jks
truststore_password:keystorepassword
itemstore_path:/opt/hpad/serverkeys/itemstore.jks
itemstore_password:keystorepassword
license_file_dir:/opt/hpad/license
haproxy_config_location:/opt/hpad/haproxy/haproxy.tpl
docker_folder:/home/defender/onprem/docker
docker_version:1.12+
version:17.1
syslog:disable

```

**Note:** The `appdefender.properties` file is stored as clear text. Since the file includes user names and passwords, you can encrypt sensitive `appdefender.properties` data. For instructions, see "[\[Optional\] Encrypting Sensitive Values](#)" on [page 19](#).

- Next, run the following to update the contents of the `appdefender.properties` file with the required details to generate the compose files, environment files, template file, and database creation script. (To display the help contents and parameter definitions, run the `generate-compose-yaml.py` script with the `-h` parameter.)

```

#chmod 755 generate-compose-yaml.py
#python generate-compose-yaml.py -h

```

```
#python generate-compose-yaml.py appdefender.properties
```

This script generates the files in listed in the following table and places them in the appdefender directory. These files start the Fortify Application Defender service.

### Application Defender Directory Files

File	Definition
applications.env	Contains the environment variables used to start Fortify Application Defender components.
applications.yml	Contains the service description to start Fortify Application Defender application containers.
create-db.sql	Used to create the Fortify Application Defender database.
haproxy.tpl	Temporary file used to create the HAProxy configuration file (haproxy.cfg).
infrastructure.env	Contains the environmental variables used to start Fortify Application Defender infrastructure components.
infrastructures.yml	Contains the service description to start Application Defender application containers.
postgres.yml	If a Postgres container is being used to start the Application Defender service, this file contains information used in bringing up the postgres container.
optional.yml	File that contains the service description for optional services such as storm_ui.
privacy_scripts.env	Optional file created if you encrypted sensitive property values. It includes the encryption key used to decrypt properties.
privacy_scripts.yml	Optional file created if you encrypted sensitive property values.
Shell script for individual hosts	In a single instance installation, this script copies corresponding shell scripts to their respective hosts and runs them.

- To log in to your Docker Hub account, run the following command on both the application and infrastructure hosts:

```
#docker login
```

6. Start the Postgres database, either as a standalone Postgres database or as a Postgres container. Fortify recommends that you start it as a Postgres container.

To start the database as a standalone Postgres database:

- a. Copy the `create-db.sql` file to the database host.
- b. Log in to the Postgres database server using valid credentials.
- c. Run the `create-db.sql` script to create the Fortify Application Defender database.

To start the database as a Postgres container, run the following:

```
#docker-compose -f postgres.yml up -d
```

7. Start the database migrations to create the initial schema, as follows:
  - a. To create the database schemas required for Fortify Application Defender, copy the `infrastructures.yml` and `infrastructures.env` files to your infrastructure host in a folder (for example: `/opt/defender/`).
  - b. Execute following command from the infrastructure host:

```
#docker-compose -f infrastructures.yml up -d db_migrations
```

**Note:** Starting the database migrations container also starts the Cassandra container.

8. [Optional] Run the following to start the `rsyslog_defender` service to consume the Fortify Application Defender service logs:

```
#docker-compose -f applications.yml up -d rsyslog_defender
```

9. To start the customer user interface and create the initial Fortify Application Defender version, execute the following command from the application host:

```
#docker-compose -f applications.yml up -d ui_customer
```

10. Delete the `zookeeper` folder, which you can find in the default `AppDefender` folder (specified in `appdefender.properties`).
11. To start the infrastructure components, run the following command from the infrastructure host:

```
#docker-compose -f infrastructures.yml up -d
```

12. To start all application components, run the following command from application host:

```
#docker-compose -f applications.yml up -d
```

13. Navigate to `https://<application_url>:8443` and reset the password.

## [Optional] Encrypting Sensitive Values

By default, YML files, ENV files, and the `appdefender.properties` files are created in clear text. The `appdefender.properties` file includes your Postgres and Vertica credentials and other sensitive data. If you want to encrypt the sensitive values stored in these files, follow these steps:

1. Complete `privacy-scripts.env` with the sensitive properties and encryption key:

```
VERTICA_USER=
```

```

VERTICA_PASSWORD=
POSTGRES_USER=
POSTGRES_PASSWORD=
DB_KEY=
PROPERTIES_KEY=

```

The `DB_KEY` and `PROPERTIES_KEY` are used to encrypt sensitive values. These values can be only 16, 24, or 32 long and can contain only alphanumeric characters and the following special characters:

```
! . , > < _
```

**Important!** Make sure that you maintain these keys in a safe place.

2. Run the docker container to encrypt the properties, as follows:

```
docker-compose -f privacy-scripts.yml up
```

**Note:** The encrypted properties will be displayed on the console. Make sure that you store them locally for later use.

3. To set values for the properties in the `appdefender.properties` file, provide the encrypted values from the previous step for the following:

```

postgres_user:
postgres_password:
vertica_user:
vertica_user:
db_key:

```

4. Create a property named `'properties_key.'` Specify the value from `PROPERTIES_KEY` in the `privacy-scripts.env` file as its value.
5. If this is a new Fortify Application Defender installation (not an upgrade), provide values for the other properties in the `appdefender.properties` file (`postgres_ip`, `postgres_dbname`, and so on).
6. Return to [Step 3](#) of the single instance installation procedure, and complete the process.

After you finish encrypting the sensitive values, remove the sensitive information from your hard drive (see ["Installing a Single Fortify Application Defender Instance" on page 15.](#))

## Post Encryption Clean-up

If you encrypted sensitive values, you must do the following to remove the sensitive information from your hard drive:

1. Delete the disk properties encryption key.
2. Delete the `appdefender/properties_encryption.env` file.
3. Delete the `properties_key` property from the `appdefender.properties` file. This property is now stored in RAM in running containers and is no longer needed in the file.

## Redeploying an Installation that Includes Encrypted Values

To redeploy an installation that includes encrypted values:

1. Add `properties_key` property to the `appdefender.properties` file.
2. Regenerate the Docker compose files (including `appdefender/properties_encryption.env`).
3. After you redeploy, delete the encryption keys again.

## Deploying a Fortify Application Defender Cluster

A clustered on premises Fortify Application Defender deployment requires at least six nodes; three application nodes and three infrastructure nodes. The advantages of a clustered installation include:

- Fault tolerance
- Replication
- Rolling restart for upgrades
- Horizontal scaling, which allows application components to handle more users and agents
- Faster event processing

**Note:** To deploy a cluster, you must have experience configuring and managing clustered networks. If this is your first Fortify Application Defender installation, Fortify recommends that you install a single instance. (See ["Installing a Single Fortify Application Defender Instance" on page 15](#) .)

### Application and Infrastructure Node Requirements

Nodes	Descriptions
Application nodes (x3)	<p>You must have at least three application nodes. Application nodes are used to start the following application components:</p> <ul style="list-style-type: none"> <li>• edge</li> <li>• command-channel</li> <li>• haproxy - The host with which agents and users interact. Designate one node for haproxy.</li> <li>• scheduler</li> <li>• ui-customer</li> <li>• rsyslog_defender</li> </ul>
Infrastructure nodes (x3)	<p>Infrastructure nodes start components for event processing, alerting, and reporting. You must have at least three infrastructure nodes for clustering. The generation script, <code>generate-compose-yaml.py</code>, generates the file required to configure the Docker daemon and start the Fortify Application Defender service from information</p>

**Application and Infrastructure Node Requirements , continued**

Nodes	Descriptions
	<p>saved in the <code>appdefender.properties</code> file.</p> <p>The following services run on infrastructure nodes:</p> <ul style="list-style-type: none"> <li>• Zookeeper</li> <li>• Kafka</li> <li>• Storm</li> <li>• Cassandra</li> <li>• Topologies</li> <li>• Database migrations</li> </ul>

**Configuring a Fortify Application Defender Cluster**

To configure a Fortify Application Defender cluster, do the following:

1. Copy the installation package to a Linux machine.
2. Designate one node as the swarm master. The swarm master controls the cluster and deploys containers across the swarm cluster. Copy the entire installation package to any single folder (for example: `/opt/appdefender`).

**Note:** You must have read, write, and execute privileges to install Fortify Application Defender.

3. Generate the Java Keystore, as follows:
  - a. Run the `build-stores.sh` script.
  - b. Enter one of the following server certificate options at the prompt:
    - For self-signed server certificate generation, enter **1**.  
Self-signed certificate scripts are used with trial or pilot installations.

```
#export PATH=$PATH:$JAVA_HOME/bin
#cd CertGeneration
#chmod 755 build-stores.sh
#chmod 755 server-root-self-signed.sh
#sh build-stores.sh
#<Press Enter>
```

- For server certificate signed by valid certificate authority (CA), enter **2**.  
If you use certificates signed by a third-party CA, then copy server certificate signed by CA (`server.crt`), server private key (`server.key`), CA intermediate Root cert (`server.int.crt`) and CA Root cert (`server.root.crt`) to the third-party folder.

**Note:** Use the filenames provided here in parentheses.

```
#export PATH=$PATH:$JAVA_HOME/bin
#cd CertGeneration
#chmod 755 build-stores.sh
#sh build-stores.sh
#<Type 2>
<Press Enter>
```

- c. Enter a passphrase that is at least six characters long for the keystore.
- d. Press **ENTER**.

Both options generate the following three files, which are required to start the Fortify Application Defender service:

- keystore.jks
- truststore.jks
- itemstore.jks

4. Update the `appdefender.properties` file with the required parameters, as shown in the following example.

```
deploy:cluster
lb_host:10.100.100.101
apps_host:['1', '10.100.100.101', 'c02app01'], ['2', '10.100.100.102', 'c02app02'], ['3', '10.100.100.103', 'c02app03']
infrastructure_host:['1', '10.100.100.111', 'c02infra01'], ['2', '10.100.100.112', 'c02infra02'], ['3', '10.100.100.113', 'c02infra03']
apps_host_mac_address:00-aa-bb-cc-dd-ee
appdefender_registry:appdefender
defender_logs:/opt/defenderlogs
defender_data:/opt/defenderdata
initial_user_email:test@defender.com
initial_user_first_name:Application
initial_user_last_name:Defender
initial_tenant_domain:defender.com
initial_tenant_name:ClusterDefender
mail_from:cluster@defender.com
mail_host:mail.server.com
mail_port:25
mail_username:mailusername@hpe.com
mail_password:mailpassword
postgres_ip:10.100.100.121
postgres_dbname:appdefender
postgres_user:username
postgres_password:postgrespassword
vertica_ip:10.100.100.131
vertica_dbname:docker
vertica_user:dbadmin
vertica_password:verticapassword
keystore_path:/opt/hpad/serverkeys/keystore.jks
keystore_password:keystorepassword
truststore_path:/opt/hpad/serverkeys/truststore.jks
truststore_password:keystorepassword
itemstore_path:/opt/hpad/serverkeys/itemstore.jks
itemstore_password:keystorepassword
license_file_dir:/opt/hpad/license
haproxy_config_location:/opt/hpad/haproxy/haproxy.tmp1
docker_folder:/home/defender/onprem/docker
docker_version:1.12+
version:17.1
syslog:disable
```

**Note:** The `appdefender.properties` file is stored as clear text. Since the file includes user names and passwords, you may want to encrypt sensitive `appdefender.properties` data.

For instructions, see "[\[Optional\] Encrypting Sensitive Values](#)" on page 27. If you do encrypt your sensitive data, make sure that you then return to the next step and complete this procedure.

- To generate the Compose files, environment files, the template file, and the database creation script, update the `appdefender.properties` file with the required values. To display the help contents and parameter definitions, run the `generate-compose-yaml.py` script with the `-h` parameter, as follows:

```
#chmod 755 generate-compose-yaml.py
#python generate-compose-yaml.py -h
#python generate-compose-yaml.py appdefender.properties
```

This generates the following files in the `appdefender` directory. These files start the Fortify Application Defender service.

### Application Defender Directory Files

File	Definition
<code>applications.env</code>	Contains the environment variables used to start Fortify Application Defender components.
<code>applications.yml</code>	Contains the service description to start Fortify Application Defender application containers.
<code>create-db.sql</code>	Use this file to create the Fortify Application Defender database. If you are installing a single instance deployment, copy and execute this file to the Postgres host.
<code>haproxy.tpl</code>	Copy this file to the application host.
<code>infrastructure.env</code>	Contains the environmental variables used to start Fortify Application Defender infrastructure components.
<code>infrastructure.yml</code>	Contains the service description to start Fortify Application Defender infrastructure containers.
<code>postgres.yml</code>	If a Postgres container is used to start the Fortify Application Defender service, copy this file to the Postgres database host.
<code>privacy_scripts.env</code>	An optional file created if you encrypted sensitive property values. It includes the encryption key used to decrypt properties.
<code>privacy_scripts.yml</code>	An optional file created if you encrypted sensitive property values.

**Application Defender Directory Files, continued**

File	Definition
Shell script for individual hosts	Clustered Installation - Additional shell scripts are generated under <b>hostShellScripts</b> folder. The scripts generated are created using the hostname provided in <code>appdefender.properties</code> . Copy and run the script on corresponding hosts.

6. The generation script creates a shell script that configures the Docker daemon and starts the consul and Swarm container. Run the generated shell scripts on their respective docker nodes. Following is a summary of the commands executed by shell script on individual host to create and start the Swarm cluster.

For a link to Docker documentation, see ["Additional References" on page 41](#).

After you create your Swarm cluster, you must configure it. The following steps provide the information you need to configure your Swarm cluster for use with Fortify Application Defender.

7. The generation script creates a shell script based on the settings in the `appdefender.properties` that configure the Docker daemon. Options configured using the shell script start the Swarm cluster daemon. The following sample shows the configuration for one host: Pl.

Sample Configuration: `docker.conf` (folder: `/etc/systemd/system/docker.service.d/docker.conf`)

```
[Service]
ExecStart=
ExecStart=/usr/bin/docker daemon -H fd:// -H tcp://10.100.100.111 -H
unix:///var/run/docker.sock --cluster-
store=consul://10.100.100.111:8500 --cluster-
advertise=10.100.100.111:2375 -g /var/lib/docker --label
com.defender.server="infrastructure"
```

After updating Docker configuration, the shell script reloads the configuration and restarts the Docker daemon.

8. The discovery backend is used to authenticate Swarm managers and nodes within the cluster. To start a Swarm cluster you must set up a discovery backend. The `docker.conf` shell script used also starts a docker container for consul.

Sample Run Command:

```
docker run -d --name=consul_dockernode01 -v /opt/newConsul/./data -e
constraint:node==dockernode01 -p 8300:8300 -p 8301:8301 -p
8301:8301/udp -p 8302:8302 -p 8302:8302/udp -p 8400:8400 -p 8500:8500 -
p 172.17.0.1:53:53/udp appdefender/consul -server -advertise
10.100.100.111 -join 10.100.100.111 -bootstrapp;
```

9. Once the discovery backend is running, a shell script starts the Swarm agent on each Docker node.

Sample Run Command:

```
docker run -d --name=swarm_dockernode01 --addr=10.100.100.111:2375
consul://10.100.100.111/swarm;
```

10. Start the Swarm Manager with replication.

Sample Run Command:

```
docker run -d -p 10.100.100.111:3375:2375 --name=manager_dockernode01
swarm manage --replication --advertise 10.100.100.111:3375
consul://10.100.100.111:8500/swarm
```

11. Log in to your Docker Hub account, and then run the following command on both the application and infrastructure hosts:

```
#docker login
```

**Important!** You must run this command from the server on which the Swarm manager is started and all Compose files are stored.

12. Start the Postgres database as either a standalone Postgres database or as a Postgres container. (Fortify recommends that you start the database as a Postgres container.) Do one of the following:

- To start the database as a standalone Postgres database, run the `create-db.sql` script to create the Fortify Application Defender database after login to the Postgres database server with valid credentials.
- To start the database as a Postgres container, copy the `postgres.yml` and `create-db.sql` files to the Docker host where you plan to run the Postgres container, and then run the following command:

```
#docker-compose -f postgres.yml up -d
```

13. Execute Database Migrations to Create the Initial Schema

Run all the Application Defender related `docker-compose` commands after pointing to the designated port for Swarm Manager, for example, port 3375 in this case.

```
#export DOCKER_HOST=<ip-address>:3375
```

The `db_migrations` creates database schemas required for Application Defender and starts Cassandra cluster containers. Execute following command from swarm manager:

```
#docker-compose -f infrastructures.yml up -d db_migrations
```

14. [Optional] Start `rsyslog_defender` service to consume Application Defender service logs:

```
#docker-compose -f applications.yml up -d rsyslog_defender
```

15. To distribute application containers across hosts:
  - a. Copy the required license and keystore files to all the hosts designated for applications in the folders mentioned in the `appdefender.properties` file.
  - b. Copy the generated `haproxy.tmp1` file to the host designated as the load balancer (`lb_host`) under the folder configured in the `appdefender.properties` file.
16. Start the customer user interface to create the initial Fortify Application Defender version, and then run the following command from the application host:

```
#docker-compose -f applications.yml up -d ui_customer
```

17. To start the infrastructure components, run the following command from the infrastructure host:

```
#docker-compose -f infrastructures.yml up -d
```

18. To start all application components, run the following command from the application host:

```
#docker-compose -f applications.yml up -d
```

19. Navigate to `https://<application_url>:8443` in your browser, and then use the reset password link to reset the password for the first user.

## [Optional] Encrypting Sensitive Values

By default, `.yml` files, `.env` files, and the `appdefender.properties` files are created in clear text. The `appdefender.properties` file includes your PostgreSQL and Vertica credentials and other sensitive data. If you want to encrypt the sensitive values stored in these files, follow these steps:

1. Complete `privacy-scripts.env` with sensitive properties and encryption key:

```
VERTICA_USER=  
VERTICA_PASSWORD=  
POSTGRES_USER=  
POSTGRES_PASSWORD=  
DB_KEY=  
PROPERTIES_KEY=
```

`DB_KEY` and `PROPERTIES_KEY` are used for encrypting sensitive values. These values should contain only alphanumeric symbols and the following special characters: `!,><_`

Possible key lengths: 16, 24, 32.

**Important!** Maintain keys in a safe place.

2. Run the docker container to encrypt the properties.

The encrypted properties will be displayed on the console. You should store them locally for later use.

```
docker-compose -f privacy-scripts.yml up
```

3. Fill `appdefender.properties`.

Insert encrypted values from Step 2 into:

```
postgres_user:  
postgres_password:  
vertica_user:  
vertica_password:  
db_key:
```

4. Create new property 'properties\_key' and fill it with the value from `PROPERTIES_KEY` in `privacy-scripts.env`

5. If this is a new App Defender installation and not an upgrade, the other properties will be empty and should be completed (e.g. `postgres_ip`, `postgres_dbname`, ...).

6. Return to [Step 3](#) of Configure an Application Defender Cluster to complete the process.

After completing the process, follow the [Post Encryption Clean-Up](#) steps below.

### Post Encryption Clean-up

If you followed the procedure to encrypt sensitive values, you will need to remove sensitive information from your hard drive to complete the process.

1. Delete the properties encryption key.
2. Delete `appdefender/properties_encryption.env`.
3. Delete `properties_key` field from the `appdefender.properties` file. This field is no longer necessary as it is stored in RAM in running containers.

### Redeployment: Encrypted Values Used

To redeploy an installation that includes encrypted values:

- a. Add `properties_key` field to `appdefender.properties` file.
- b. Regenerate docker-compose files (including `appdefender/properties_encryption.env`).
- c. After redeployment, delete encryption keys again.

## Advanced Installation Notes

The advanced installation notes are intended for experienced Fortify Application Defender users.

<a href="#">SMTP Email Server Authentication</a> .....	29
<a href="#">Java Keystore</a> .....	29
<a href="#">All Docker-Compose Files</a> .....	30
<a href="#">Vertica Database</a> .....	33
<a href="#">Postgres Database (Optional)</a> .....	33
<a href="#">Infrastructure Virtual Machine Services</a> .....	34
<a href="#">Application Virtual Machine Services</a> .....	35

### SMTP Email Server Authentication

If you want to access the SMTP email server using authentication, provide the appropriate values for `mail_username` and `mail_password` in the `appdefender.properties` file before you run the `generate-compose-yaml.py` script:

```
mail_username: <abc@abc.com>
```

```
mail_password: <password>
```

If you do not want to authenticate the mail server, leave these fields empty.

### Java Keystore

This section contains the following topics:

All Fortify Application Defender communication takes place on a secure channel. To get this working, Fortify Application Defender needs three keystore files. Trial and pilot installations must use the [Self-signed Server Certificate](#) script. If you use certificates signed by a third party, use a [Server Certificate Signed by Valid Certificate Authority](#).

#### Self-signed Server Certificate

The script provided in the package gives an option to create a self-signed server certificate chain and agent certificate chain to be used with Fortify Application Defender.

The included scripts:

`server-root-self-signed.sh` - This script generates the certificate chain for the Fortify Application Defender server. Execute this script only when creating a self-signed server certificate.

`build-stores.sh` - This script generates the agent certificate chain and the final java keystore files used for the Fortify Application Defender service. After executing this script, the following jks files are generated in the CertGeneration folder:

- `keystore.jks` - Contains the server certificate chain which includes the Intermediate ROOT certificate and ROOT certificate.
- `truststore.jks` - Contains `trustedCertEntry` for the Intermediate agent, ROOT agent, and server ROOT certificate.
- `itemstore.jks` - Contains the agent certificate chain, `trustedCertEntry` for ROOT certificate and `trustedCertEntry` for the ROOT agent.

## Server Certificate Signed by Valid Certificate Authority

If you are using a certificate signed by a valid CA, copy the signing authority's ROOT certificate and Intermediate ROOT certificate to `CertGeneration>thirdparty` folder and rename the files if necessary:

- The server certificate should be named `server.crt` (example: `qa_appdefender_com.crt` renamed to `server.crt`)
- The server Private key should be named `server.key` (example: `qa_appdefender_com.crt` renamed to `server.key`)
- The CA Intermediate ROOT certificate should be named `server.int.crt` (example: `Digicert_int.crt` renamed to `server.int.crt`)
- The CA ROOT certificate should be named `server.root.crt` (example: `Digicert_root.crt` renamed to `server.root.crt`)

## All Docker-Compose Files

The `generate-compose-yaml.py` script generates the docker-compose files, database creation scripts, and the proxy template file. These files are used to start the Fortify Application Defender service. This script uses the `appdefender.properties` file as input and generates the files needed to start the Fortify Application Defender service.

### Update the `appdefender.properties` file

The following describes the `appdefender.properties` file settings that need to be updated:

#### Docker Compose Setting Field Descriptions

Setting	Description
<code>appdefender_registry</code>	The Docker registry that stores Fortify Application Defender images
<code>apps_host</code>	List of the nodes designated for application components. Node information includes following properties: <ul style="list-style-type: none"> <li>• Number assigned to the host (numeric range: <b>1-255</b>).</li> </ul>

**Docker Compose Setting Field Descriptions, continued**

Setting	Description
	<ul style="list-style-type: none"> <li>• IP Address</li> <li>• Hostname</li> </ul>
mail_username	Required for SMTP authentication. For more information, see page 29.
mail_password	Required for SMTP authentication.
apps_host_mac_address	The MAC address of the host machine running docker for the applications. Use the MAC address provided for the Fortify Application Defender license download.
defender_data	Directory on individual hosts to persist application defender data.
defender_logs	Directory on individual hosts to persist application defender logs.
deploy	Based on the kind of deployment, this setting should be either single or cluster.
infrastructure_host	<p>List of the nodes designated for infrastructure components. Node information includes the following properties:</p> <ul style="list-style-type: none"> <li>• Number assigned to the host (numeric range: <b>1-255</b>).</li> <li>• IP Address</li> <li>• Hostname</li> </ul>
initial_tenant_domain	Domain of the tenant, for example, microfocus.com.
initial_tenant_name	Name of the tenant, for example, Micro Focus.
initial_user_email	Email address of the first user in the system.
initial_user_first_name	Name of the first user in the system.
initial_user_last_name	Surname of the first user in the system.
lb_host	Load balancer host. The front end node which provides the interface for users and agents to interact with.

**Docker Compose Setting Field Descriptions, continued**

Setting	Description
logs_data_dir	Directory on host machine that contains the logs and data.
SMTP server	<p>Fortify Application Defender needs an SMTP server to send emails to users:</p> <p>mail_from - A valid email address as the sender of all automated emails.</p> <p>mail_host - A valid mail host address.</p> <p>mail_port - Default port = <b>25</b></p> <p>mail_username: - Required for SMTP authentication. For more information on SMTP Authentication, see <a href="#">Advanced Installation Notes</a>.</p> <p>mail_password: - Required for SMTP authentication.</p>
Postgres Container	<p>A Docker container for the Postgres database is created unless the following properties are set to configure Fortify Application Defender to use an external Postgres database:</p> <p>postgres_ip</p> <p>postgres_dbname</p> <p>postgres_user</p> <p>postgres_password</p>
Vertica Database	<p>A Vertica database is required. Use the properties below to configure Fortify Application Defender to use the Vertica database:</p> <p>vertica_ip</p> <p>vertica_dbname</p> <p>vertica_user</p> <p>vertica_password</p>
Keystore, Truststore, and Itemstore Configuration	<p>Configure a keystore, truststore, and itemstore must be configured as per the Administrator Guide:</p> <p>keystore_path</p> <p>keystore_password</p> <p>truststore_path</p> <p>truststore_password</p> <p>itemstore_path</p>

**Docker Compose Setting Field Descriptions, continued**

Setting	Description
	itemstore_password
License File Directory	Provide a path to license file directory. license_file_dir
haproxy_config_location	High Availability Proxy Configuration Location - Directory where haproxy.tmp1 file is copied to the load balance host (lb_host). This is an optional field.
version	Version refers to the Application Defender docker containers hosted on the registry. This is an optional field.
docker_folder	Folder where the docker files are saved. This is an optional field.
docker_version	Version of docker used to install Fortify Application Defender. This value should be "1.12+" unless you are installing a previous version. This establishes the docker start command as "dockerd -D" or "docker daemon -H".
Syslog: (enable/disable)	This setting redirects application logs to the rsyslog_defender container.

**Vertica Database**

A columnar database that Fortify Application Defender uses to store event data.

**Vertica Database Port Descriptions**

Service Defined	Host Port Number	Container Port Number	Service Description
vertica	5433	Standalone	Used as a persistent data store for security and monitor events.

**Postgres Database (Optional)**

Use this option if you plan to use a Postgres container instead of the separate database installation.

Execute following command on the Postgres host:

```
#docker-compose -f postgres.yml up -d
```

## Postgres Database Port Descriptions

Service Defined	Host Port Number	Container Port Number	Service Description
postgres	5432	5432	Relational database that Fortify Application Defender uses to store application state, agent state, user data, and agent binaries.

## Infrastructure Virtual Machine Services

To start all the infrastructure components on infrastructure virtual machine, execute the following command:

```
#docker-compose -f infrastructure.yml up -d
```

## Infrastructure Virtual Machine Port Descriptions

Service Defined	Host Port	Container Port Number	Service Description
cassandra	9042, 7000	9042, 7000, 7001, 7199, 9160	A highly available distributed database that Fortify Application Defender uses to store intermediate data for alerts.
kafka	9092	9092	A stateless distributed queue used for event stream processing.
zookeeper	2181, 2888, 3888	2181, 2888, 3888	A service for maintaining configuration, naming, distributed synchronization, and group services that Kafka uses.
storm_nimbus	6627	6627	A distributed real-time stream processing technology. Fortify Application Defender uses five Storm topologies for notifications, reporting, alerting, and writing events to Vertica and Reconciliation.
storm_supervisor		6700, 6701, 6702,	Storm

**Infrastructure Virtual Machine Port Descriptions, continued**

<b>Service Defined</b>	<b>Host Port</b>	<b>Container Port Number</b>	<b>Service Description</b>
		6703	
storm_ui	8080	8080	Provides a clean UI to check the status of Storm processes.
topologies	n/a	n/a	n/a

**Application Virtual Machine Services**

To start all the application components on applications virtual machine, issue the following command:

```
#docker-compose -f applications.yml up -d
```

**Application Virtual Machine Port Descriptions**

<b>Service Defined</b>	<b>Host Port Number</b>	<b>Container Port Number</b>	<b>Service Description</b>
backend_jobs		8080	Component used to schedule reports.
command_channel	random	8080	Secure communication channel between Fortify Application Defender agents and service for exchange of commands.
consul		8300, 8400, 8500, 8600 8301, 8302	
db_migrations			Used to migrate existing Fortify Application Defender data in case of upgrades or to create database schemas in case of new installations.
edge	random	4321	Secure communication channel between Fortify Application Defender agent and service that agents use to

**Application Virtual Machine Port Descriptions, continued**

<b>Service Defined</b>	<b>Host Port Number</b>	<b>Container Port Number</b>	<b>Service Description</b>
			send events to the service.
haproxy		1936, 8443, 8444, 4321	Load balancer that provides access to different services provided by Fortify Application Defender.
registrator			
ui_customer	random	8080	The website used by all Fortify Application Defender users to access protect, manage, messaging and alerting functionality.
Rsyslog_defender	514, 1999	514, 1999	rsyslog_defender container used to consume Fortify Application Defender service logs.

## Scaling Fortify Application Defender On-Premises Services

Docker Compose files allow you to scale different services based on load and number of agents, as follows:

- To scale instances of the customer user interface, execute following command:

```
#docker-compose -f applications.yml scale ui_customer=2
```

- To scale instances of the Command Channel:

```
#docker-compose -f applications.yml scale command_channel=2
```

- To scale process for backend jobs:

```
#docker-compose -f applications.yml scale backend_jobs=2
```

- To scale instances of Edge:

```
#docker-compose -f applications.yml scale edge=2
```

## Fortify Application Defender System Hardening

Fortify Application Defender is a complex, multi-process solution with a big-data architecture. The distributed nature of the solution increases the attack surface, especially to malicious insiders. In addition to proper patch management policies, strict access controls, and secure server configurations, Fortify recommends the following to reduce your attack surface and increase security of your Fortify Application Defender deployment:

change to:

- Protect the `appdefender.properties`, `applications.env`, and `infrastructures.env` files by restricting who can access them and read their contents. Fortify recommends at least file system level access controls to ensure only authenticated users with sufficient entitlement can access these files.
- The Fortify Application Defender installation provides a container with the Storm user interface to monitor storm processes as well as perform topology administration. Malicious users with access to the Storm UI can disable storm topologies and prevent event storage, analysis, or visualization in the Fortify Application Defender server. Fortify recommends that you disable `storm_ui` if you are not using it:

```
#docker stop storm_ui
```

- Fortify Application Defender has a three tier architecture:
  - a. Application - Presentation tier
  - b. Infrastructure - Logic tier
  - c. Databases - Data tier

Users and agents only interact with the application layer. Fortify recommends that you configure your firewall to provide access to only these machines.

- Follow the instructions that Docker provides to secure your Docker daemon and secure Swarm Cluster deployment. For more information, see ["Additional References" on page 41](#).

## Logging Policy

The following logging policy table provides information about each of the Fortify Application Defender services.

Svc #	Docker Image	Data Location	Log	Container Log Rotation Policy	Internal Daemon Rotation Policy
1	ui-customer		Docker Container Folder e.g. <code>/home/defender/docker/containers/&lt;container_id&gt;/</code>	max-size: "50m"max-file: "9"	
2	ui-internal		Docker Container Folder e.g. <code>/home/defender/docker/containers/&lt;container_id&gt;/</code>	max-size: "50m"max-file: "9"	

3	backend-jobs		Docker Container Folder e.g. /home/defender/docker/containers/<container_id>/	max-size: "50m"max-file: "9"	
4	command-channel		Docker Container Folder e.g. /home/defender/docker/containers/<container_id>/	max-size: "50m"max-file: "9"	
5	edge		Docker Container Folder e.g. /home/defender/docker/containers/<container_id>/	max-size: "50m"max-file: "9"	
6	topologies		Docker Container Folder e.g. /home/defender/docker/containers/<container_id>/	max-size: "50m"max-file: "9"	
7	db-migrations		Docker Container Folder e.g. /home/defender/docker/containers/<container_id>/	max-size: "50m"max-file: "9"	
8	Zookeeper	\$defender_data/zookeeper	\$defender_logs/zookeeper	max-size: "50m"max-file: "9"	autopurge.purgeInterval=24 autopurge.snapRetainCount=10
9	Kafka	defender_data/kafka	\$defender_logs/kafka	max-size: "50m"max-file: "9"	log.retention.hours=168
10	Storm-nimbus		\$defender_logs/storm_nimbus	max-size: "50m"max-file: "9"	100 MB 9 Files
11	Storm-supervisor		\$defender_logs/storm_supervisor	max-size: "50m"max-file: "9"	100 MB 9 Files
12	Storm-ui		\$defender_logs/storm_ui	max-size: "50m"max-file: "9"	100 MB 9 Files
13	Cassandra	\$defender_data/cassandra	\$defender_logs/cassandra	max-size: "50m"max-file: "9"	20 MB 20 files
14	Consul	\$defender_data/consul	Docker Container Folder e.g. /home/defender/docker/containers/<container_id>/	max-size: "50m"max-file: "9"	
15	Registrar		Docker Container Folder e.g. /home/defender/docker/containers/<container_id>/	max-size: "50m"max-file: "9"	
16	haproxy		Docker Container Folder e.g. /home/defender/docker/containers/<container_id>/	max-size: "50m"max-file: "9"	
17	Swarm		Docker Container Folder e.g. /home/defender/docker/containers/<container_id>/	max-size: "50m"max-file: "9"	
18	Postgres			max-size: "50m"max-file: "9"	
19	Vertica			max-size: "50m"max-file: "9"	
20	Syslog			max-size: "50m"max-file: "9"	

## Upgrading from an Earlier Version

If you are upgrading from a previous version of Fortify Application Defender on premises, you must first regenerate the Docker Compose files, and then upgrade the Docker images.

### Regenerating Docker Compose Files

To regenerate the Docker Compose files:

1. Get the latest version of the Application Defender installation package from the Fortify Application Defender portal.
2. Update the `appdefender.properties` file to be used for updated or new properties. For details, see ["Deploying a Fortify Application Defender Cluster" on page 21](#). In the **Version** field (for example, if you are upgrading to version 19.1.0, use version 19.1.0).
3. Run the generation script with the updated properties file. This will generate Compose files to start the Fortify Application Defender instance.

### Upgrade Docker Images

Fortify Application Defender has two image categories; Docker *Infrastructure* images and Docker *Application* images. All infrastructure Docker images are updated on an as-needed basis. All application images are updated monthly.

To update the Docker images:

1. Use the following command to stop all Fortify Application Defender containers:

```
# docker stop $(docker ps -a -q)
```

2. Use the following command to remove all Fortify Application Defender containers:

```
# docker rm -vf $(docker ps -a -q)
```

3. To start the Fortify Application Defender instance, do one of the following:

- If `rsyslog_defender` is enabled for centralized logging, start the `rsyslog_defender` container, as follows:

```
#docker-compose -f infrastructures.yml up -d db_migrations
#docker-compose -f applications.yml up -d rsyslog_defender
#docker-compose -f infrastructures.yml -f applications.yml up -d ui_
customer
#docker-compose -f infrastructures.yml -f applications.yml up -d
```

- If `rsyslog_defender` is *not* enabled for centralized logging, do the following:

```
#docker-compose -f infrastructures.yml up -d db_migrations
#docker-compose -f infrastructures.yml -f applications.yml up -d ui_
```

```
customer  
#docker-compose -f infrastructures.yml -f applications.yml up -d
```

4. Use the following command to start the Storm user interface to troubleshoot Storm topologies that have been submitted:

```
#docker-compose -f infrastructures.yml -f applications.yml -f  
optional.yml up -d storm_ui
```

## Additional References

For assistance in configuring the recommended hardware components in your Fortify Application Defender on-premises installation see the documentation listed in the following table.

Software Component	Documentation URL
Docker Compose	<a href="https://docs.docker.com/compose/install/">https://docs.docker.com/compose/install/</a>
Docker Control and configure with systemd	<a href="https://docs.docker.com/engine/admin/systemd/">https://docs.docker.com/engine/admin/systemd/</a>
Docker Engine	<a href="https://docs.docker.com/engine/installation/ubuntu/linux/">https://docs.docker.com/engine/installation/ubuntu/linux/</a>
Docker Hub Account	<a href="https://hub.docker.com/">https://hub.docker.com/</a>
Docker Protect the daemon socket	<a href="https://docs.docker.com/engine/security/https/">https://docs.docker.com/engine/security/https/</a>
Docker Swarm Configuration	<a href="https://docs.docker.com/swarm/plan-for-production/">https://docs.docker.com/swarm/plan-for-production/</a>
Docker Swarm for TLS	<a href="https://docs.docker.com/swarm/configure-tls/">https://docs.docker.com/swarm/configure-tls/</a>
Postgres	<a href="http://www.postgresql.org/docs/9.4/static/index.html">http://www.postgresql.org/docs/9.4/static/index.html</a>
Vertica	<p>Version 8.1.x:</p> <p><a href="https://my.vertica.com/docs/7.1.x/HTML/#Authoring/InstallationGuide/Other/InstallationGuide.htm%3FTocPath%3DInstallation%2520Guide%7C_____0">https://my.vertica.com/docs/7.1.x/HTML/#Authoring/InstallationGuide/Other/InstallationGuide.htm%3FTocPath%3DInstallation%2520Guide%7C_____0</a></p> <p><a href="https://my.vertica.com/docs/Hardware/HP_Vertica%20Planning%20Hardware%20Guide.pdf">https://my.vertica.com/docs/Hardware/HP_Vertica%20Planning%20Hardware%20Guide.pdf</a></p> <p>Version 9.1.x:</p> <p><a href="https://www.vertica.com/documentation/vertica/9-1-x/">https://www.vertica.com/documentation/vertica/9-1-x/</a></p>

# 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 On-Premises Installation Guide (Fortify Application Defender 19.1.0)**

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 [FortifyDocTeam@microfocus.com](mailto:FortifyDocTeam@microfocus.com).

We appreciate your feedback!