Micro Focus
Fortify Application Defender

Software Version: 19.4.0

On-Premises Installation Guide

Document Release Date: December 2019
Software Release Date: December 2019

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

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

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.

Software Release-Version	Change
19.4.0	Added: Support for Secure LDAP
19.3.0	Added: LDAP configuration instructions.
19.2.0	Changed: Updated to make instructions more concise and easier to follow.
19.1.0	Changed: • Support for Docker engine version to 18.09.2 and later
18.20	Added: • Support for encrypting sensitive values in the `appdefender.properties` file • Agent support for Tomcat 9 and IIS 10 Updated: • Docker engine and kernel support for 18.09 and later
17.1	Added: • Support for Docker version 17.05.0 • SMTP email server authentication
16.3	Added: • rsyslog_defender service to consume logs from all application services • A new script to print the status of all Fortify Application Defendercontainers • Support for Docker 1.12

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

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.

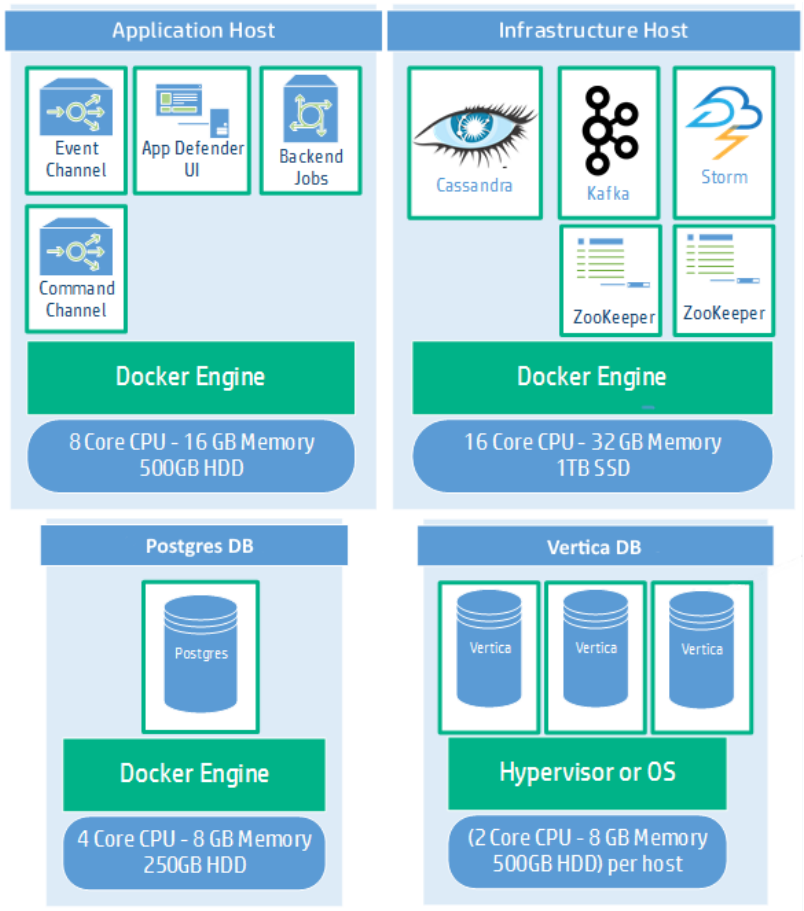
Installation Overview

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

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, three Vertica hosts, and an email server.



Application Host Components

The application host components are described in the following table.

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

Component	Description
Jobs	such as reports.
Command Channel	Secure communication channel between the Fortify Application Defender agents and the service used to exchange commands.
rsyslog_ defender	syslog container to consume logs from Fortify Application Defender app services. This includes logs for the following services: • ui_customer • command_channel • backend_jobs • Edge

Infrastructure Host Components

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

Postgres Database Component

Component	Description
Postgres	Object-relational database that stores Fortify Application Defender user data.

Vertica Database Component

Component	Description
Vertica	Columnar database that stores event data for Fortify Application Defender. Clustering

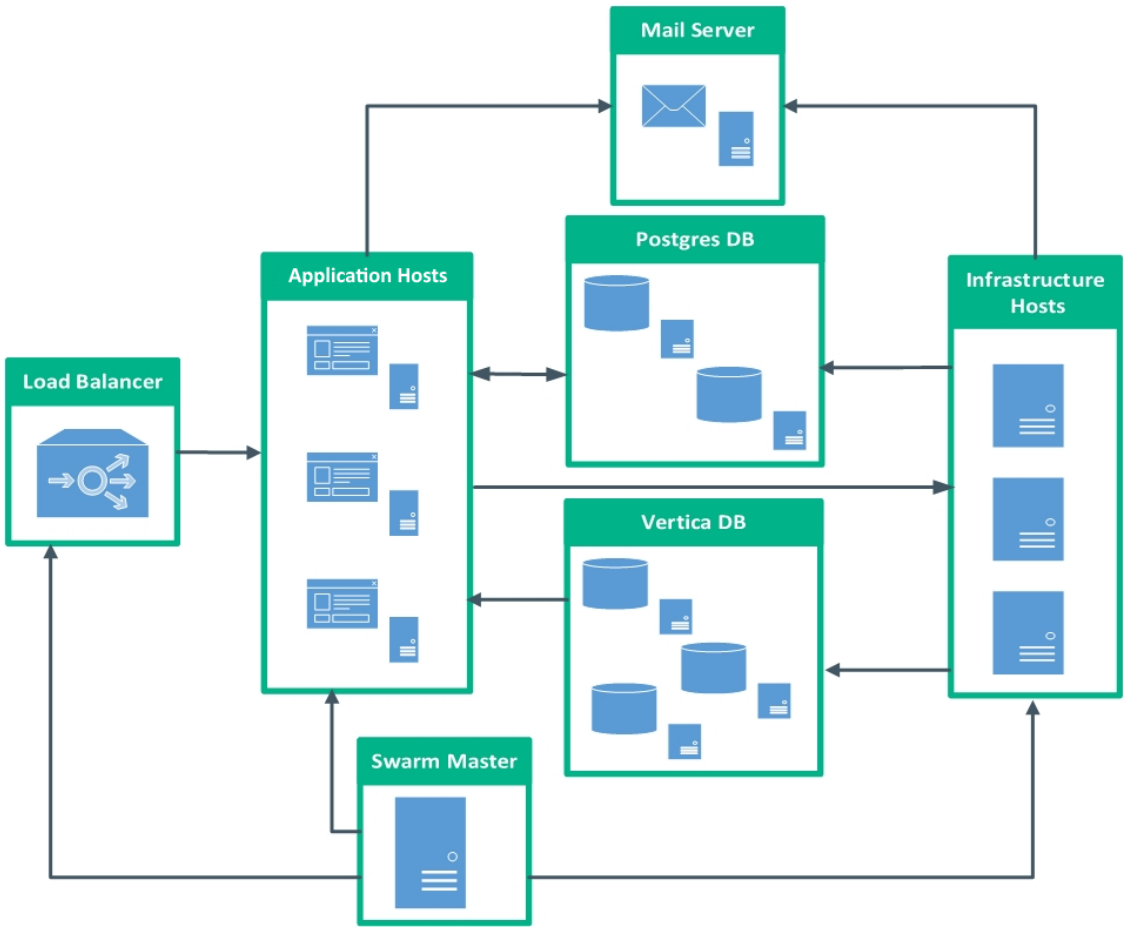
Component	Description
	the Vertica hosts makes it easier to scale up the entire system later. Fortify recommends that you have a minimum of three clustered Vertica hosts. Fortify supports, but does not recommend, a single instance.

Email Server Component

Component	Description
Email Server	Application that sends and receives email from local users (users within the same domain) and remote senders.

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
Application Hosts	List of the nodes designated for application components. Node information includes the following properties: • Number assigned to the host (numeric range: **1-255**). • IP Address • Hostname
Infrastructure Hosts	List of the nodes designated for infrastructure components. Node information includes following properties: • Number assigned to the host (numeric range: **1-255**). • IP Address • Hostname
Load Balancer	Load balancer host. Front-end node that provides the interface with which users and agents are to interact.
Mail Server	Application that sends and receives email from local users (users within the same domain) and remote senders.
Postgres Database	Database server. Its primary function is to store data securely and allow for retrieval at the request of other software applications.
Swarm Manager	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
Swarm Manager	1 core	4 GB	100 GB HDD

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
Vertica (x3)	2 cores	8 GB	500 GB HDD per host

For additional Vertica requirements, see "Additional References" on page 42.

To begin installation, see "Installing a Single Fortify Application Defender Instance" on page 15.

To begin installation, see "Installing a Clustered Fortify Application Defender Instance" on page 20.

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.

Docker container network communication

If you're using RHEL, you may need to add the following entries to your `/etc/sysctl.conf` file in order for Docker containers to communicate correctly on the network:

```
net.ipv4.conf.all.forwarding=1
net.ipv6.conf.all.forwarding=1
net.bridge.bridge-nf-call-iptables=1
net.bridge.bridge-nf-call-ip6tables=1
```

Note: If you receive a docker warning about `net.bridge.bridge-nf-call-iptables` and `bridge-nf-call-ip6tables` being disabled, you will need to make calls to `firewall-cmd` to allow communication on the needed ports. For information on enabling `bridge-nf-call-iptables`, consult the Oracle Container Runtime for Docker User's Guide.

SMTP Server (mail)

Fortify Application Defender sends an email notification to each user in the system. Provide a reference to the SMTP server for Fortify Application Defender to use. For more information, see "Additional Installation Notes" on page 30.

Vertica Database Cluster

- Use Vertica documentation to install a Vertica cluster. For links to the Vertica site, see "Additional References" on page 42.
- 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.
- To create a database schema run the `db_migrations` Docker container.

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 42.

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).

Additonal 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`	Used to generate docker-compose files (.yml and .env files), haproxy template file, and Bash scripts (for Clustered deployment).
`appdefender.properties` (sample)	Used as an argument with the `generate-compose-yaml.py` script to create different App Defender services.

File Name	Purpose
`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 singe 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 supports high availability, zero downtime deployment, and replication. It scales to support any number of Fortify Application Defender agents.

Select the installation that best matches your needs:

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.

Configuring a Single Fortify Application Defender

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:
   - 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>
     ```

   - 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:00aabbccddee
appdefender_registry:appdefender
defender_logs:/opt/appdefender/logs
defender_data:/opt/appdefender/data
initial_user_email:appdefender@microfocus.com
initial_user_first_name:Application
initial_user_last_name:Defender
initial_tenant_domain:appdefender.com
initial_tenant_name:SingleInstance
mail_from:single@appdefender.com
mail_host:mail.server.com
mail_port:25
mail_username:mailusername@microfocus.com
mail_password:mailpassword
postgres_ip:10.100.100.121
postgres_dbname:appdefender
postgres_user:postgresusername
postgres_password:postgrespassword
vertica_ip:10.100.100.131
vertica_dbname:appdefender
vertica_user:verticausername
vertica_password:verticapassword
keystore_path:/opt/appdefender/serverkeys/keystore.jks
keystore_password:keystorepassword
truststore_path:/opt/appdefender/serverkeys/truststore.jks
truststore_password:keystorepassword
itemstore_path:opt/appdefender/serverkeys/itemstore.jks
itemstore_password:keystorepassword
license_file_dir:/opt/appdefender/license
haproxy_config_location:/opt/appdefender/haproxy/haproxy.tmpl
docker_folder:/var/lib/docker
docker_version:1.12+
version:19.2.0
syslog:enable
```

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 "Post Installation Steps" on page 27.

Note: If you provided an incorrect SMTP server address or the SMTP server is not accessible to the Application Defender environment, you may not be able to complete the first login after deployment.

4. Next, run the following to generate the compose files, environment files, and template file. To display 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 listed in the following table and places them in the appdefender directory. These files start the Fortify Application Defender services.

Note: If you use an internal insecure image registry:

a. In the `appdefender.properties` file, set the value of the `appdefender_registry` setting to `<hostname>:<port>` and re-run the `generate-compose-yaml.py` script.

b. For all hostShellScript files created, add the following argument to the service command: `--insecure-registry=<hostname>:<port>`

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.
`haproxy.tmpl`	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.

Application Defender Directory Files, continued

File	Definition
Shell script for individual hosts	In a clustered installation, this script copies corresponding shell scripts to their respective hosts and runs them.

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

```
#docker login
```

Startup a Fortify Application Defender

1. 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. Log into the Postgres database.
   b. Create the user specified in the `appdefender.properties` file.
   c. Create the database specified in the `appdefender.properties` file.
   d. Ensure the user you created has the necessary database permissions.

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

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

2. 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 the 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.

Note: If a Postgres container is used, Fortify recommends you disable unauthenticated login after migrations are complete:

Edit the Postgres config file:

`<appdef-data-folder>/postgres/pg_hba.conf`
change:

```
host all all 0.0.0.0/0 trust
```
to:

```
host all all 0.0.0.0/0 md5
```

3. [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
```

4. 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
```

5. Delete the zookeeper folder, which you can find in the default AppDefender folder (specified in `appdefender.properties`).
6. Start the infrastructure components by running the following command from the infrastructure host:

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

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

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

8. Navigate to `https://<application_url>:8443` and follow the Reset Password procedure for the user specified in the `initial_user_email` property in `appdefender.properties`. Check your mailbox for the Reset Password link.

Note: There is a bug in recent versions of Docker where some published ports don't show up when you run `ps` after container deployment. For example, the haproxy or rsyslog_defender containers may not appear to have an published ports, yet the ports may actually be functioning. Continue with the installation process.

Post Installation

To complete your deployment, please follow the "Post Installation Steps" on page 27.

Installing a Clustered Fortify Application Defender Instance

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)	You must have at least three application nodes. Application nodes are used to start the following application components: • edge • command-channel • haproxy - The haproxy host runs on a single node in your network. This is the host where agents and users interact. • scheduler • ui-customer • rsyslog_defender
Infrastructure nodes (x3)	Infrastructure nodes start components for event processing, alerting, and reporting. You must have at least three infrastructure nodes for clustering. The generation script, `generate-compose-yaml.py`, generates the file required to configure the Docker daemon and start the Fortify Application Defender service from information saved in the `appdefender.properties` file. The following services run on infrastructure nodes: • Zookeeper • Kafka • Storm • Cassandra • Topologies • Database migrations

Configuring a Fortify Application Defender Cluster

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

1. Copy the installation package to each allocated host machine.
2. Designate one node as the Swarm Manager. The Swarm Manager 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','app01'],['2','10.100.100.102','app02'],['3','10.100.100.103','app03']]
infrastructure_host:[['1','10.100.100.111','infra01'],['2','10.100.100.112','infra02'],['3','10.100.100.113','infra03']]
apps_host_mac_address:00aabccddee
appdefender_registry:appdefender
defender_logs:/opt/appdefender/logs
defender_data:/opt/appdefender/data
initial_user_email:appdefender@microfocus.com
initial_user_first_name:Application
initial_user_last_name:Defender
initial_tenant_domain:appdefender.com
initial_tenant_name:ClusterInstance
mail_from:single@appdefender.com
mail_host:mail.server.com
mail_port:25
mail_username:mailusername@microfocus.com
mail_password:mailpassword
postgres_ip:10.100.100.121
postgres_dbname:appdefender
postgres_user:postgresusername
postgres_password:postgrespassword
vertica_ip:10.100.100.131
vertica_dbname:appdefender
vertica_user:verticausername
vertica_password:verticapassword
keystore_path:/opt/appdefender/serverkeys/keystore.jks
keystore_password:keystorepassword
truststore_path:/opt/appdefender/serverkeys/truststore.jks
truststore_password:keystorepassword
itemstore_path:opt/appdefender/serverkeys/itemstore.jks
itemstore_password:keystorepassword
license_file_dir:/opt/appdefender/license
haproxy_config_location:/opt/appdefender/haproxy/haproxy.tmpl
docker_folder:/var/lib/docker
docker_version:1.12+
version:19.2.0
syslog:enable
```

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 "Installing a Clustered Fortify Application Defender Instance" on page 20. If you do encrypt your sensitive data, make sure that you then return to the next step and complete this procedure.

5. To generate the Compose files, environment files, and the template file, 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
`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.

Application Defender Directory Files, continued

File	Definition
`haproxy.tmpl`	Copy this file to the application host where you will run haproxy.
`infrastructure.env`	Contains the environmental variables used to start Fortify Application Defender infrastructure components.
`infrastructure.yml`	Contains the service description to start Fortify Application Defender infrastructure containers.
`postgres.yml`	If a Postgres container is used to start the Fortify Application Defender service, copy this file to the Postgres database host.
`privacy_scripts.env`	An optional file created if you encrypted sensitive property values. It includes the encryption key used to decrypt properties.
`privacy_scripts.yml`	An optional file created if you encrypted sensitive property values.
Shell script for individual hosts	Clustered Installation - Additional shell scripts are generated under **hostShellScripts** folder. The scripts generated are created using the hostname provided in `appdefender.properties`. Copy and run the script on corresponding hosts.

Configuring a Swarm Cluster

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 hosts to create and start the Swarm cluster.

For a link to Docker documentation, see "Additional References" on page 42.

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.

1. 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.
   a. Run each script from `/hostShellScripts/` folder on its respective host, starting with the infrastructure nodes.
   b. Start infrastructure node (`infra01 - 10.100.100.111`) and run "`infrastracture1.sh`" there.
   c. Start the other Infrastructure nodes.

d. After Infrastructure nodes are done, do the same on application machines (run `applications1.sh` on `app01 - 10.100.100.101`).

Script explanation:

The following sample shows the configuration for one host after the script has been executed:

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 the Docker configuration, the shell script reloads the configuration and restarts the Docker daemon. 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;
```

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/swar
```

Finally, it starts 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
```

2. To check if all cluster nodes are working fine, execute the following command:

```
docker -H tcp://any_of_your_infrastracture_nodes_ip:3375 info
```

Note: If you cloned your machines using a virtual machine management tool, you need to delete the current Docker ID and restart Docker. In order to change Docker ID, just remove this file `/etc/docker/key.json` and restart the Docker daemon.

Swarm managers use Docker ID's to identify cluster nodes, so if you don't adjust ID's nodes, it won't work properly.

Start a Fortify Application Defender Cluster

1. 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.

2. 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.

   To start Postgres as a standalone database:

   a. Log into the Postgres database.

   b. Create the user specified in the `appdefender.properties` file.

   c. Create the database specified in the `appdefender.properties` file.

   d. Ensure the user you created has the necessary database permission.

   To start the database as a Postgres container, copy the `postgres.yml` 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
```

3. 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 the following command from swarm manager:

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

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

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

5. 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.tmpl` file to the host designated as the load balancer (`lb_host`) under the folder configured in the `appdefender.properties` file.

6. 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
```

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

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

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

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

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

Note: If you provided an incorrect SMTP server address or the SMTP server is not accessible to the Application Defender environment, you may not be able to complete the first login after deployment.

Post Installation

To complete your deployment, please follow the "Post Installation Steps" below.

Post Installation Steps

After performaing a Fortify Application Defender single or clustered installation, follow these post-installation steps to complete your deployment.

[Optional] Encrypting Sensitive Values

By default, .yml files, .env files, and the `appdefender.properties` files are created in clear text. The `appdefender.properties` file inludes your Postgress 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.

Integrating LDAP Servers

If you would like access to your LDAP users, you can integrate your LDAP server or servers with Application Defender.

To Integrate an LDAP server:

1. Click the **Administer** tab and then click the **LDAP Configurations** button.

   The Add LDAP Configuration screen appears.
2. Fill in the Basic Server Properties of the Add LDAP Configuration screen as follows:
   a. Server Name: Type a name of your choice to identify the LDAP server.
   b. Server URL: Type the URL address for the LDAP server.
   c. Base DN: Paste in the base distinguishedName.
   d. Bind User DN: Paste in the Bind User distinguisedName.
   e. Bind User Password: Type in the Bind User password.
3. Fill in the User Lookup Schema section of the Add LDAP Configuration screen as follows:
   a. User firstname attribute: Type the LDAP attribute name that should align with this one.
   b. User lastname attribute: Type the LDAP attribute name that should align with this one.
   c. Groupname attribute: Type the LDAP attribute name that should align with this one.
   d. User username attribute: Type the LDAP attribute name that should align with this one.
   e. User email attribute: Type the LDAP attribute name that should align with this one.
   f. Group member attribute: Type the LDAP attribute name that should align with this one.
4. Click the **Test Connection** button. If the connection fails, double check your work and try again.
5. Repeat these steps for additional LDAP server integrations.

Note: If you are using LDAP over SSL/TLS (LDAPS), you must install the LDAP server certificate to the Trusted Certificate Authority on the Applications host machine.

Redeployment: Encrypted Values Used

To redeploy an installation that includes encrypted values:

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

Additional Installation Notes

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

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

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
`appdefender_registry`	The Docker registry that stores Fortify Application Defender images
`apps_host`	List of the nodes designated for application components. Node information includes following properties: • Number assigned to the host (numeric range: **1-255**). • IP Address • Hostname

Docker Compose Setting Field Descriptions, continued

Setting	Description
mail_username	Required for SMTP authentication.
mail_password	Required for SMTP authentication.
`apps_host_mac_address`	The MAC address of the host machine running docker for the applications. The MAC address must match the MAC address specified in the license file.
`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`	List of the nodes designated for infrastructure components. Node information includes the following properties: • Number assigned to the host (numeric range: **1-255**). • IP Address • Hostname
`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.
`logs_data_dir`	Directory on host machine that contains the logs and data.
SMTP server	Fortify Application Defender needs an SMTP server to send emails to users: `mail_from` - A valid email address as the sender of all automated emails.

Docker Compose Setting Field Descriptions, continued

Setting	Description
	`mail_host` - A valid mail host address. `mail_port` - Default port = **25** `mail_username:` - Required for SMTP authentication. For more information on SMTP Authentication, see Advanced Installation Notes. `mail_password:` - Required for SMTP authentication.

Postgres Container	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:
	`postgres_ip`
	`postgres_dbname`
	`postgres_user`
	`postgres_password`

Vertica Database	A Vertica database is required. Use the properties below to configure Fortify Application Defender to use the Vertica database:
	`vertica_ip`
	`vertica_dbname`
	`vertica_user`
	`vertica_password`

Keystore, Truststore, and Itemstore Configuration	Configure a keystore, truststore, and itemstore must be configured as per the Administrator Guide:
	`keystore_path`
	`keystore_password`
	`truststore_path`
	`truststore_password`
	`itemstore_path`
	`itemstore_password`

License File Directory	Provide a path to license file directory. `license_file_dir`

Docker Compose Setting Field Descriptions, continued

Setting	Description
`haproxy_config_location`	High Availability Proxy Configuration Location - Directory where `haproxy.tmpl` 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 Host Services

To start all the infrastructure components on infrastructure hosts, 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,	Part of Storm framework architecture.

Infrastructure Virtual Machine Port Descriptions, continued

Service Defined	Host Port	Container Port Number	Service Description
		6703	
storm_ui	8080	8080	Provides a clean UI to check the status of Storm processes.
swarm	2375, 3375, 53		
topologies	n/a	n/a	Part of Storm framework architecture.

Application Host Services

To start all the application components on application hosts, issue the following command:

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

Application Host Port Descriptions

Service Defined	Host Port Number	Container Port Number	Service Description
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, 53	8300, 8400, 8500, 8600, 8301, 8302	Registry server.

Application Host Port Descriptions, continued

Service Defined	Host Port Number	Container Port Number	Service Description
db_ migrations			Used to migrate existing Fortify Application Defender data in case of upgrades or to create database schemas in case of new installations. After the job is complete, the container goes down.
edge	random	4321	Secure communication channel between Fortify Application Defender agent and service that agents use to send events to the service.
haproxy		1936, 8443, 8444, 4321	Load balancer that provides access to different services provided by Fortify Application Defender.
registrator			Service that monitors and registers components in Consul.
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.
swarm	2375, 3375, 53		

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:

- 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 42.

Logging Policy

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

Svc #	Docker Image	Data Location Log Internal Daemon Rotation Policy	Container Log Rotation Policy
1	ui-customer	Log: Docker Container Folder e.g. /home/defender/docker/containers/<container_id>/	max-size: "50m"max-file: "9"
2	ui-internal	Log: Docker Container Folder e.g. /home/defender/docker/containers/<container_id>/	max-size: "50m"max-file: "9"
3	backend-jobs	Log: Docker Container Folder e.g. /home/defender/docker/containers/<container_id>/	max-size: "50m"max-file: "9"
4	command-channel	Log: Docker Container Folder e.g. /home/defender/docker/containers/<container_id>/	max-size: "50m"max-file: "9"
5	edge	Log: Docker Container Folder e.g. /home/defender/docker/containers/<container_id>/	max-size: "50m"max-file: "9"
6	topologies	Log: Docker Container Folder e.g. /home/defender/docker/containers/<container_id>/	max-size: "50m"max-file: "9"
7	db-migrations	Log: Docker Container Folder e.g. /home/defender/docker/containers/<container_id>/	max-size: "50m"max-file: "9"
8	Zookeeper	Data Location: $defender_data/zookeeper Log: $defender_logs/ Internal Daemon Rotation Policy: autopurge.purgeInterval=24 autopurge.snapRetainCount=10	max-size: "50m"max-file: "9"
9	Kafka	Data Location: defender_data/kafka Log: $defender_logs/kafka Internal Daemon Rotation Policy: log.retention.hours=168	max-size: "50m"max-file: "9"
10	Storm-nimbus	Log: $defender_logs/storm_nimbus Internal Daemon Rotation Policy: 100 MB 9 Files	max-size: "50m"max-file: "9"
11	Storm-supervisor	Log: $defender_logs/storm_supervisor Internal Daemon Rotation Policy: 100 MB 9 Files	max-size: "50m"max-file: "9"
12	Storm-ui	Log: $defender_logs/storm_ui Internal Daemon Rotation Policy: 100 MB 9 Files	max-size: "50m"max-file: "9"
13	Cassandra	Data Location: $defender_data/cassandra Log: $defender_logs/cassandra Internal Daemon Rotation Policy: 20 MB 20 files	max-size: "50m"max-file: "9"

14	Consul	Data Location: $defender_data/consul Log: Docker Container Folder e.g. /home/defender/docker/containers/<container_id>	max-size: "50m"max-file: "9"
15	Registrator	Log: Docker Container Folder e.g. /home/defender/docker/containers/<container_id>/	max-size: "50m"max-file: "9"
16	haproxy	Log: Docker Container Folder e.g. /home/defender/docker/containers/<container_id>/	max-size: "50m"max-file: "9"
17	Swarm	Log: 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 "Installing a Clustered Fortify Application Defender Instance" on page 20. Update the **Version** field (for example, if you are upgrading to version 19.4.0, use version 19.4.0).
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	https://docs.docker.com/compose/install/
Docker Control and configure with systemd	https://docs.docker.com/engine/admin/systemd/
Docker Engine	https://docs.docker.com/engine/installation/ubuntulinux/
Docker Hub Account	https://hub.docker.com/
Docker Protect the daemon socket	https://docs.docker.com/engine/security/https/
Docker Swarm Configuration	https://docs.docker.com/swarm/plan-for-production/ https://docs.docker.com/swarm/install-manual/
Docker Swarm for TLS	https://docs.docker.com/swarm/configure-tls/
Postgres	http://www.postgresql.org/docs/9.4/static/index.html
Vertica	Version 8.1.x: https://my.vertica.com/docs/7.1.x/HTML/#Authoring/InstallationGuide/Other/InstallationGuide.htm%3FTocPath%3DInstallation%2520Guide%7C_____0 https://my.vertica.com/docs/Hardware/HP_Vertica%20Planning%20Hardware%20Guide.pdf Version 9.1.x: https://www.vertica.com/documentation/vertica/9-1-x/

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 19.4.0)

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.

We appreciate your feedback!