Micro Focus
Fortify Application Defender

Software Version: 20.3.0

On-Premises Installation Guide

Document Release Date: October 13, 2020

Software Release Date: September 2020

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
20.3.0	Updated: Updated the installation process. Removed separate installation flows for single and Cluster installations. Removed: Installing a Single Fortify Application Defender Instance Installing a Clustered Fortify Application Defender Instance
19.4.0	Added: Support for Secure LDAP
19.3.0	Added: LDAP configuration instructions.

Getting Started

This document provides instructions on how to install and run Micro Focus Fortify Application Defender.

This section contains the following topics:

Intended Audience

This document provides information on deploying Fortify Application Defender on premises. To deploy Fortify Application Defenderyou should have experience installing and configuring Docker containers. In addition, you should have a basic understanding of hardware and server management.

For information on using the software, consult the program Help system.

Hardware Requirements

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. If you only install a single Vertica instance, your data is not replicated and you risk losing security event data.

Component	CPU	Memory	Hard Drive
Application	8 cores	16 GB	500 GB HDD
Infrastructure	16 cores	32 GB	1 TB SSD

Component	**CPU**	**Memory**	**Hard Drive**
Postgres database	4 cores	8 GB	500 GB HDD
Vertica	2 cores	8 GB	500 GB HDD per host Fortify recommends a minimum of three Vertica instances in a production environment.

For additional Vertica requirements, see "Additional References" on page 40.

Software Requirements

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

Network Connection

All 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. Application Defender is supplied with Docker swarm, soit uses the Docker overlay networking subsystem to creates a distributed network among multiple Docker daemon hosts. The network works with host-specific networks so that connected containers can securely communicate with each other.

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. Your Application Hosts should be able to pull images from the Docker Hubs. For additional port information, see "Additional Installation Notes" on page 34.

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 34.

Vertica Database Cluster

- Use Vertica documentation to install a Vertica cluster. For links to the Vertica site, see "Additional References" on page 40.
- 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 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 40.

Fortify Application Defender License

You will receive 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).

Application Defender Installation Package

The Application Defender installation package 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)0.
`appdefender.properties` (sample)	Used as an argument with the `generate-compose-yaml.py` script to create different App Defender services.
`SecurityContent<Release_Number>.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.
Fortify AppDefender <Version_Number> License.txt	List of 3rd Party component licenses.
Welcome to Fortify Software Products_AppDef.pdf	Welcome document with any last-minute notices.
EULA.pdf	End-User License Agreement.
Agents	Folder that contains the Application Defender Agents

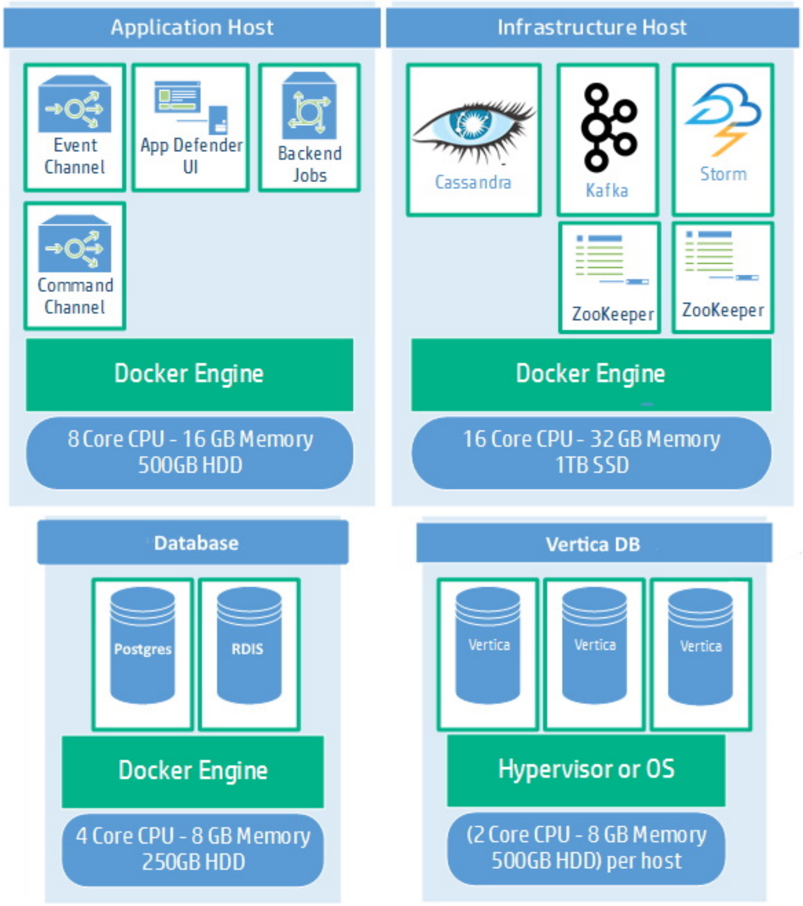
On-Premises Environment

You can install Fortify Application Defenderas a single instance or in a cluster.

Single-Instance Installation

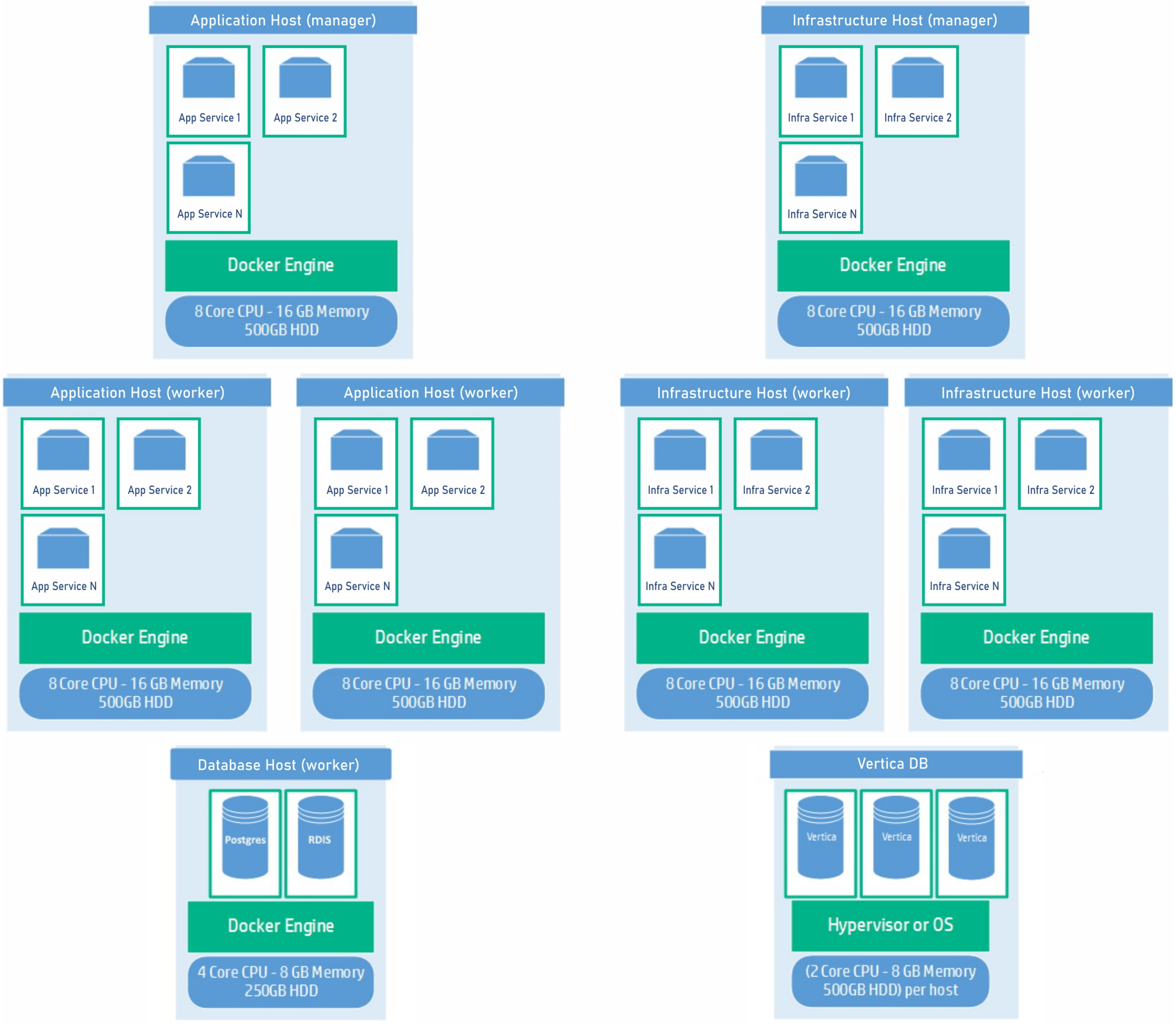
The following diagram illustrates a Fortify Application Defender on-premises environment. The minimum deployment consists of:

- An application host (swarm manager)
- An infrastructure host (swarm manager)
- A Postgres host (swarm worker)
- Three Vertica hosts (not part of a swarm cluster)
- An email server (not provided)



Clustered-Instance Installation

The following diagram illustrates a clustered Fortify Application Defender on-premises installation. In most cases, one instance of each application service in use is run on each node. They are not tightly bound so you can scale nodes and services independently.


Application Host (manager)
App Service 1
App Service 2
App Service N
Docker Engine
8 Core CPU - 16 GB Memory 500GB HDD
Infrastructure Host (manager)
Infra Service 1
Infra Service 2
Infra Service N
Docker Engine
8 Core CPU - 16 GB Memory 500GB HDD
Application Host (worker)
App Service 1
App Service 2
App Service N
Docker Engine
8 Core CPU - 16 GB Memory 500GB HDD
Application Host (worker)
App Service 1
App Service 2
App Service N
Docker Engine
8 Core CPU - 16 GB Memory 500GB HDD
Infrastructure Host (worker)
Infra Service 1
Infra Service 2
Infra Service N
Docker Engine
8 Core CPU - 16 GB Memory 500GB HDD
Infrastructure Host (worker)
Infra Service 1
Infra Service 2
Infra Service N
Docker Engine
8 Core CPU - 16 GB Memory 500GB HDD
Database Host (worker)
Postgres
RDIS
Docker Engine
4 Core CPU - 8 GB Memory 250GB HDD
Vertica DB
Vertica
Vertica
Vertica
Hypervisor or OS
(2 Core CPU - 8 GB Memory 500GB HDD) per host

Deployment Hosts

Host	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

Deployment Hosts, continued

Host	Description
	• 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
Database Hosts	Postgres and RDIS database server. Its primary function is to store data securely and allow for retrieval at the request of other software applications.
Vertica Hosts	Columnar database that stores event data for Fortify Application Defender.

Application Host Services

The services you run on the application hosts are described in the following table.

Service	Image Name Service Name Host Port Number Container Port Number	Description
Application Defender UI	**Image Name**: ui_customer **Service Name**: applications_ui_customer **Host Port Number**: 8443 **Container Port Number**: 8080	Website used to access Application Defender's functionality.
Command Channel	**Image Name**: command-channel **Service Name**: applications_command_channel **Host Port Number**: 8444 **Container Port Number**: 8080	Secure communication channel between Fortify Application Defender agents and the service used to exchange commands.
Backend Jobs	**Image Name**: backend-jobs **Service Name**: applications_backend_jobs	Component used to manage and schedule internal back-end jobs, such as reports.

Service	**Image Name** **Service Name** **Host Port Number** **Container Port Number**	Description
Event Channel	**Image Name**: edge **Service Name**: applications_edge **Host Port Number**: 4321 **Container Port Number**: 4321	A secure communications channel between the Fortify Application Defender agent and the service used by agents to send events to the service.
Rsyslog (rsyslog)	**Image Name**: rsyslog **Service Name**: applications_rsyslog **Host Port Number**: 514, 1999 **Container Port Number**: 514, 1999	syslog container that consumes logs from Fortify Application Defender app services. This includes logs for the following services: • ui_customer • command_channel • backend_jobs • Edge

Infrastructure Host Services

Service	**Image Name** **Service Name** **Host Port Number** **Container Port Number**	Description
Apache Cassandra	**Image Name**: cassandra **Service Name**: infrastructures_cassandra	Open-source distributed database that Fortify Application Defender uses to store intermediate data for alerts.
Apache Kafka	**Image Name**: Kafka **Service Name**: infrastructures_kafka	Stateless distributed queue used for reports, events, and activity stream processing.
Apache Storm	**Image Name**: storm-nimbus, storm-	Distributed real-time stream

Service	Image Name Service Name Host Port Number Container Port Number	Description
	supervisor, topologies **Service Name**: infrastructures_storm_nimbus, infrastructures_storm_supervisor, infrastructures_topologies **Host Port Number**: 6627 **Container Port Number**: 6700, 6701, 6702, 6703, 6627	computation system. Fortify Application Defender uses the following Storm topologies for notifications, reporting, alerting, reconciliation, and writing events to Vertica. • storm-nimbus • storm-supervisor • topologies
Apache Zookeeper	**Image Name**: zookeeper **Service Name**: infrastructures_zookeeper **Host Port Number**: 2181, 2888, 3888 **Container Port Number**: 2181, 2888, 3888	Service used to maintain configuration information, naming, distributed synchronization, and group services used by Kafka and Storm.
Database Migration Script (db-migrations)	**Image Name**: db-migrations **Service Name**: infrastructures_db_migrations	A script, executed at system start-up, that's used to generate or update database schemas. It should be executed and exit with a return code of 0.

Database Host Services

Note: You can use the Postgres database included in the postgres container, or use a pre-existing Postgres database. If you choose to use your own Postgres database, see Postgres (Optional Installation) in "Additional Installation Notes" on page 34.

Service	**Image Name** **Service Name** **Host Port Number** **Container Port Number**	Description
Postgres	**Image name :** postgres **Service name:** postgres_postgres **Host Port Number:** 5432 **Container Port Number:** 5432	Object-relational database that stores Fortify Application Defender user data.
Redis	**Image name :** redis **Service name:** redis_redis **Container Port Number:** 6379	An in-memory data structure used to store live user sessions to the Portal.

Vertica Database

Component	**Host Port Number** **Container Port Number**	Service Description
Vertica	**Host Port**: 5433 **Container Port**: Standalone	Used as a persistent data store for security and monitor events.

Email Server

Note: Email server is not provided.

Component	Description
Email Server	Application that sends and receives email from local users (users within the same domain) and remote senders. Application Defender On Premises does not include an email server; configure own email server.

Installation

Beginning with Application Defender version 20.3.0, we no longer distinguish between a single install and a cluster install. In addition, the installation has been simplified, utilizing Docker swarm technology to manage the installation of the various components.

This guide provides the steps required to install the product, but requires a basic understanding of Docker, specifically installation and configuration concepts. If you are new to Docker, please read the official Docker documentation before attempting to install this product.

This section contains the following topics:

Before You Begin

Installing and configuring Application Defender requires experience with Docker. If you are new to Docker or need more information, consult the following topics in the official Docker documentation to help you better understand Application Defender's installation and configuration options.

- Swarm Mode
- Deploying Services to a Swarm
- Swarm mode mounting routing mesh
- Use of Overlay Networks
- Docker secrets
- Overview on Services
- Workers
- Managers

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

Prepare the Environment

Perform the following steps in order:

1. Copy the entire installation package to a folder in your `opt` directory (for example: `/opt/appdefender`) on application's hosts.
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 a 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`

Note: If you use an internal insecure image registry, in the `appdefender.properties` file set the value of the `appdefender_registry` to `<hostname>:<port>` and rerun the `generate-compose-yaml.py` script.

3. Update the `appdefender.properties` file with the required parameters. Each of the parameters is defined in the Properties table below the example.

```
app_manager_host:10.10.10.100
apps_host_mac_address:001122aabbcc
appdefender_registry:appdefender
defender_data:/opt/appdefender/data
initial_user_email:John.Smith@corpdomain.com
```

```
initial_user_first_name:John
initial_user_last_name:Smith
initial_tenant_domain:corpdomain.com
initial_tenant_name:Corp-Tenant
mail_from:John.Smith@corpdomain.com
mail_host:smtp.corpdomain.com
mail_port:
mail_username:
mail_password:
postgres_user:postgresusername
postgres_password:postgrespassword
redis_password:password
vertica_ip:10.10.10.200
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
version:20.3.0
syslog:enable
db_key:1111qqqq2222wwww
ldap_enabled:true
scale:1
```

Properties

The following table lists allowable entries for the `appdefender.properties` file.

Parameter	**Description**
app_manager_host	IP address of the swarm manager node that holds application services .
apps_host_mac_address	MAC address of the host machine running Docker for the applications. This must be the same MAC address used for license generation.
appdefender_registry	The docker registry where the Application Defender images are stored. Default is "appdefender".
docker_folder	This is the location on the host machine where Docker stores all its files. Defaults to: `/var/lib/docker`.

Parameter	Description
ip_documentation:	URL of the Application Defender documentation server. Defaults to `/documentation`for local documentation.
defender_data	The directory on the host machine where the data will be stored.
initial_user_email	Email address of the initial Application Defender user. Use an email address that the user has access to. You will need this address to retrieve a reset password link that will be required for first log on to the system.
initial_user_first_name	First name of the initial Application Defender user.
initial_user_last_name	Last name of the initial Application Defender user.
initial_tenant_domain	The domain of the tenant, e.g., corp.com.
initial_tenant_name	Name of the initial tenant.
mail_from	A valid email address for the sender of all automated emails.
mail_host	The mail server address.
mail_port	Email server port. Default is 25.
mail_username	Username for SMTP authentication.
mail_password	Password for email account.
postgres_user	Valid username to connect to Postgres database.
postgres_password	Password to connect to Postgres database.
postgres_ip	IP Address of Postgres host when using a standalone Postgres database. Default is Postgres container.
postgres_dbname:	Postgres database name when using a standalone Postgres database. Default value is appdef.
vertica_ip	IP address of Vertica host.
vertica_dbname	Vertica database name to be used with Application Defender.

Parameter	Description
vertica_user	Valid username to connect to Vertica database.
vertica_password	Valid password to connect to Vertica Database.
redis_password	Valid password to connect to Redis database.
keystore_path	Path to the keystore file where Application Defender is being started.
keystore_password	Valid password for keystore.
truststore_path	Path to the truststore file where Application Defender is being started.
truststore_password	Valid password for truststore.
itemstore_path	Path to the itemstore file where Application Defender is being started.
itemstore_password	Valid password for itemstore.
license_file_dir	Valid path to license files on host machine.
version	The version of Docker containers to be used to start Application Defender instance. If no version is specified, the latest version will be used.
syslog	Set to "enable" in order to integrate Application Defender with Syslog server. Default is "disable".
db_key	A random string of length 16, 24, or 32 characters. These characters will be used to encrypt sensitive data in your Postgres database. **Note:** Keep this key in a secure place. If lost, there is no way to restore the Postgres database.
ldap_enabled	Set to "true" in order to integrate Application Defender with a corporate LDAP server. Default is "false".
scale	Scaling coefficient. Use a value of 1 for a single instance (all containers will be deployed once), for two instances of each container, use a value of 2 and so on. Scaling coefficient

Parameter	Description
	defines number of replicas for specific services.

> **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. Execute the `generate-compose-yaml.py` script with the `-h` parameter to display help content and parameter definitions.

```
#python generate-compose-yaml.py -h
```

5. Execute the `generate-compose-yaml.py` script with the `appdefender.properties` file as a parameter to generate compose files, environment files, and the secrets generation script.

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

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.
`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.
`rdis.yml`	File that contains the information required to bring up the RDIS container.
`create-secrets.sh`	Bash script to generate Docker Secrets in the Application Defender installation directory.

> **Note:** All files, except `create-secrets.sh`, will be located in the `<install dir>/appdefender/` folder.

6. Move all .jks files (`keystore.jks`, `trustore.jks`, and `itemstore.jks`) to the folders specified in the respective `keystore_path`, `truststore_path`, and the `itemstore_path` properties.
7. Create the folder specified in the `license_file_dir` property and copy your server and application licenses there.
8. On the database host, create a `postgres`folder in the directory specified in the `defender_data` property.

Enforce Firewall Rules

On each worker host:

1. Add Worker firewall rules:

```
systemctl start firewalld;
systemctl enable firewalld;
firewall-cmd --add-port=2376/tcp --permanent;
firewall-cmd --add-port=7946/tcp --permanent;
firewall-cmd --add-port=7946/udp --permanent;
firewall-cmd --add-port=4789/udp --permanent;
firewall-cmd --reload;
systemctl restart docker
```

2. Docker Login:

```
#docker login
```

On each manager host:

1. Add manager firewall rules.

```
systemctl start firewalld;
systemctl enable firewalld;
firewall-cmd --add-port=2376/tcp --permanent;
firewall-cmd --add-port=2377/tcp --permanent;
firewall-cmd --add-port=7946/tcp --permanent;
firewall-cmd --add-port=7946/udp --permanent;
firewall-cmd --add-port=4789/udp --permanent;
firewall-cmd --reload;
systemctl restart docker;
```

2. Docker login.

```
Docker login
```

Initialize Swarm Cluster

1. Initialize Swarm cluster on Application Manager host.

```
docker swarm init;
```

2. Execute the following to get the commands and tokens required to add nodes.

   For Worker nodes:

```
docker swarm join-token worker; # add worker command
```

   For Manager nodes:

```
docker swarm join-token manager; # add manager command
```

3. Execute the command you obtained on the database host to add a Worker to the cluster.

```
docker swarm join --token <worker-token> <host-ip>:<port>
```

   Note: To add additional Worker nodes, repeat this step on every node.

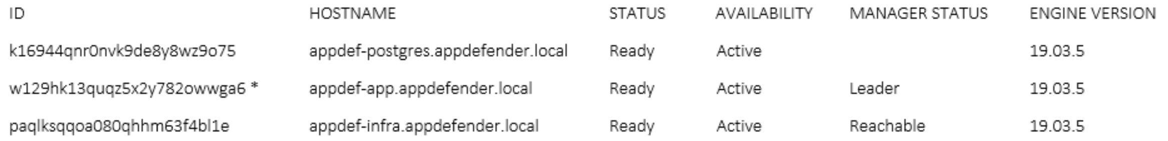
4. Execute the command you obtained on the infrastructure host to add a Manager to the cluster.

```
docker swarm join --token <manager-token> <host-ip>:<port>
```

5. Run the following command on the Application Manager host to ensure all nodes were added to the cluster. The list command provides a list of nodes with their IDs, IPs, and hotnames.

```
docker node ls
```

   Sample output:



```
ID                            HOSTNAME                          STATUS   AVAILABILITY   MANAGER STATUS   ENGINE VERSION
k16944qnr0nvk9de8y8wz9o75     appdef-postgres.appdefender.local Ready    Active                          19.03.5
w129hk13quqz5x2y782owwga6 *   appdef-app.appdefender.local      Ready    Active         Leader           19.03.5
paqlksqqoa080qhhm63f4bl1e     appdef-infra.appdefender.local    Ready    Active         Reachable        19.03.5
```

6. Add constraint labels to all nodes in the cluster. These labels will be used to determine which service can be deployed on which node. Use the node IDs acquired in the previous step.

```
docker node update --label-add host-type=app <app-node-id>
docker node update --label-add host-type=infra <infra-node-id>
docker node update --label-add host-type=db <db-node-id>
```

Creating Secrets, Overlay Network, and Run Services

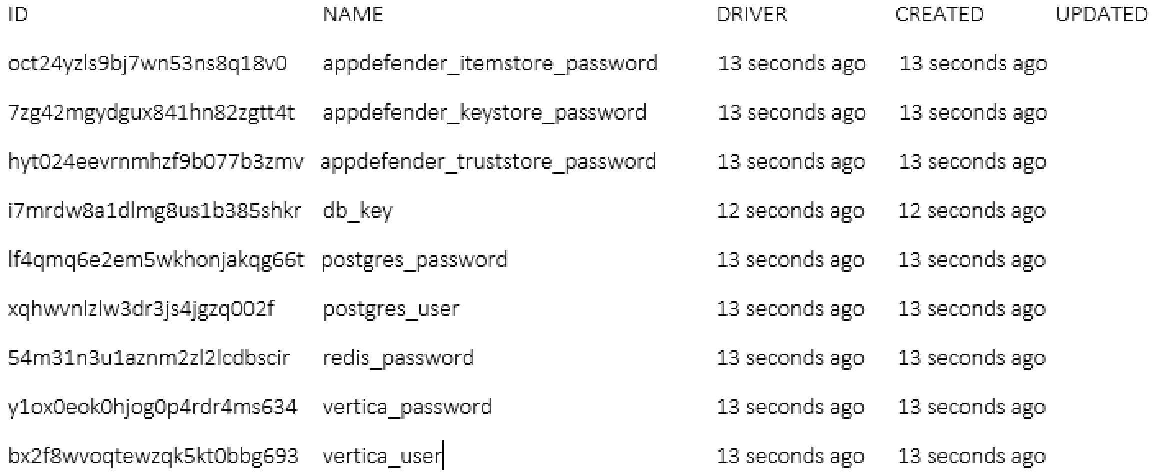
1. Navigate to the Application Manager host installation folder and run the create secrets script.

   ```
   ./create-secrets.sh
   ```

2. Check the list of secrets.

   ```
   docker secrets ls;
   ```

   Sample output:

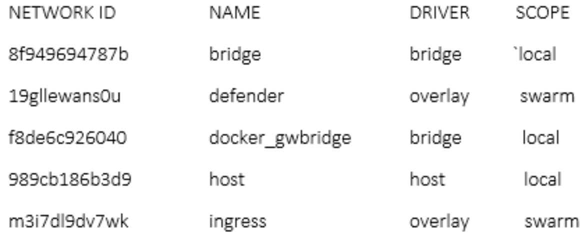
   

   ```
   ID                          NAME                              DRIVER           CREATED          UPDATED
   oct24yzls9bj7wn53ns8q18v0   appdefender_itemstore_password    13 seconds ago   13 seconds ago
   7zg42mgydgux841hn82zgtt4t   appdefender_keystore_password     13 seconds ago   13 seconds ago
   hyt024eevrnmhzf9b077b3zmv   appdefender_truststore_password   13 seconds ago   13 seconds ago
   i7mrdw8a1dlmg8us1b385shkr   db_key                            12 seconds ago   12 seconds ago
   lf4qmq6e2em5wkhonjakqg66t   postgres_password                 13 seconds ago   13 seconds ago
   xqhwvnlzlw3dr3js4jgzq002f   postgres_user                     13 seconds ago   13 seconds ago
   54m31n3u1aznm2zl2lcdbscir   redis_password                    13 seconds ago   13 seconds ago
   y1ox0eok0hjog0p4rdr4ms634   vertica_password                  13 seconds ago   13 seconds ago
   bx2f8wvoqtewzqk5kt0bbg693   vertica_user                      13 seconds ago   13 seconds ago
   ```

3. Create an overlay network for service communication.

   ```
   docker network create -d overlay defender
   ```

4. Run the `docker network ls` command. The defender network should appear in the list.
   Sample output:

   ```
   NETWORK ID     NAME              DRIVER    SCOPE
   8f949694787b   bridge            bridge    `local
   19gllewans0u   defender          overlay    swarm
   f8de6c926040   docker_gwbridge   bridge     local
   989cb186b3d9   host              host       local
   m3i7dl9dv7wk   ingress           overlay    swarm
   ```

5. Navigate to the `<install dir>/appdefender/` directory on the Application Manager host, and start the following services in order:
   a. Start the Postgres container and make sure it is running without errors.

      ```
      docker stack deploy --with-registry-auth -c postgres.yml postgres;
      docker stack ps postgres;
      ```

> **Note:** Skip this step if you are using a standalone Postgres server.

b. Start the Redis container and make sure it is running without errors.

```
docker stack deploy --with-registry-auth -c redis.yml redis;
docker stack ps redis;
```

c. Deploy infrastructure services.

```
docker stack deploy --with-registry-auth -c infrastructures.yml
infrastructures;
docker stack ps infrastructures;
```

> The database migration script will be executed as part of the infrastructure services startup. This script will connect to the Postgres and Vertica databases to create relevant schemas. This operation may take a few minutes. Before moving on, make sure the process completed without errors (for example, there should be no connectivity or credential issues).

d. Deploy Application Services.

```
docker stack deploy --with-registry-auth -c applications.yml
applications;
docker stack ps applications;
```

e. (optional) Start the Storm UI to troubleshoot Storm topologies that have been submitted.

```
docker stack deploy --with-registry-auth -c optional.yml optional;
docker stack ps optional;
```

6. (optional) Run a service that provides insight into Docker Swarm cluster on the Web UI.

```
docker service create --name=viz \
--publish 8090:8080 \
--constraint=node.role==manager \
--mount=type=bind,src=/var/run/docker.sock,dst=/var/run/docker.sock
/appdefender/visualizer;
```

7. Protect `appdefender.properties`, `applications.env`, and `infrastructures.env` files according to recommendation in "Fortify Application Defender System Hardening" on page 36.

Upgrading from 20.3.0 or Later

If you are updating from Fortify Application Defender version 20.3.0 or later, follow these directions to upgrade to the latest release.

1. Update the `version`property in the `appdefender.properties` file to the current release number.
2. Regenerate the compose files using the following Python script.

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

3. Delete all stacks.

   ```
   docker stack rm applications;
   docker stack rm infrastructures;
   docker stack rm optional;
   docker stack rm postgres;
   docker stack rm redis;
   ```

4. Run all the stacks again in the `appdefender` directory.

   ```
   docker stack deploy --with-registry-auth -c postgres.yml postgres
   docker stack deploy --with-registry-auth -c redis.yml redis
   docker stack deploy --with-registry-auth -c infrastructures.yml infrastructures
   docker stack deploy --with-registry-auth -c applications.yml applications
   docker stack deploy --with-registry-auth -c optional.yml optional
   ```

Upgrading from 20.2.X or Earlier Release

If you are updating from a version of Fority Application Defender earlier than 20.3.X, you will need to update the `appdefender.properties` file and generate new stacks.

> **Note:** If you have been using the properties encryption feature, sensitive information like your database username and passwords in the `appdefender.properties` file are encrypted. You need to replace these encrypted values with plain text values as Fortify now uses the docker secrets function as a more secure and robust solution.

1. Clean up the old environments on all application and infrastructure hosts:
   a. Delete obsolete files.

   ```
   rm -rf <appdef_installation_dir>/appdefender/
   rm -rf <defender_data>
   rm -rf <defender_logs>
   ```

   b. Stop and remove all containers, images, and volumes.

   ```
   sudo docker stop $(sudo docker ps -aq)
   sudo docker rm $(sudo docker ps -aq)
   sudo docker rm -f$(sudo docker images -aq)
   docker volume rm $(docker volume ls -a)
   ```

2. Edit the properties file:
   a. Remove deprecated properties.

   ```
   - lb_host
   - apps_host
   - infrastructure_host
   - haproxy_config_location
   - docker_folder
   - deploy
   - defender_logs
   - postgres_ip
   ```

```
- postgres_dbname
- postgres host if you are using postgres container (not an external
postgres service)
```

b. Add new properties.

```
- app_manager_host
- redis_password
- scale
```

c. Update version number property`(Set to 20.3.0 or later (current version))`

3. Upload a new `generate-compose-yaml.py` script to the Application Manager Host just as you did in Step 1 above.
4. Modify the `appdefender.properties` file on the Application Manager Host.
5. Regenerate compose files with the Python script.

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

6. Enforce firewall rules. (See Enforce Firewall Rules on page 1.)
7. Initialize Swarm cluster. (See Initialize Swarm cluster on page 1.)
8. Create secrets and an overlay network for service communication (see Create Secrets, Overlay Network, and Run Services on page 1.)
9. Run all stacks.
10. Remove opened passwords and usernames from hosts.

Scaling the Cluster

As your requirement change, you can add additional nodes and services. Nodes and services are scaled separately. You can add as many nodes as you require to your environment. See Scaling the Cluster on page 29 for instructions on adding additional nodes to your environment.

Manual Scaling

Manual scaling allows you to choose the services you need to scale and specify how many replicas you require. The disadvantage to manually scaling services is that each time you redeploy Application Defender, you will need to manually scale the services to the state they were in before the redeployment.

You can manually scale services using the `docker service scale` command. For example, to scale the UI Customer service to create 3 replicas you would use the following command.

```
docker service scale applications_ui_customer=3;
```

Scaling can be applied to the services listed in the Services topic.

> **Note:** To disable automatic scaling, remove the `scale` parameter from the `appdefender.property` file.

Automatic Scaling

Services can be scaled automatically based on the scaling coefficient set using the scale property in `appdefender.properties`. The scaling coefficient sets the number of replicas created for each services. By default, the coefficient is set to 1. If you increase the value of the scale parameter and redeploy Application Defender, each service will be replicated the number of times you specify.

If you do not require automatic scaling, remove the scale property from the `appdefender.properties` file.

Services

The following services can be scaled automatically or manually:

- UI Customer – the service responsible for the UI.
- Command Channel – the service responsible for status and settings exchange with agenst.
- Backend Jobs – the service responsible for periodically executing backend jobs, such as sending reports and gathering statistics.
- Edge – the service responsible for gathering events from agents.
- Kafka – the message broker for events and alerts.
- Storm Supervisor – the service responsible for reports generation, events persisting, and alert generation.

Nodes

If you need additional nodes to run the services you require, you can add additional nodes to the cluster. Based on the type of service you need, add infrastructure or application nodes.

To add an additional node:

1. Navigate to the swarm manager node.
2. Run `docker join-token worker` to obtain the `docker swarm join` command for the worker node.
3. Navigate to the node you want to join.
4. Enforce worker node firewall rules. See Enforce Firewall Rules on page 1.
5. Run the `docker swarm join` command on the new node.
6. Navigate to the swarm manager node.
7. Run `docker node ls` in order to verify the node has been added.
8. Use the node id from the `docker node ls` output to add node constraintto the node. Run one of

the following commands based on the node type.

For application nodes:

```
docker node update --label-add host-type=app <app-node-id>
```

For infrastructure nodes:

```
docker node update --label-add host-type=infra <infra-node-id>
```

9. The node is now ready, but services currently running will not be automatically added to the new node. For more information, see Add Service to a Node.

Add Service to a Node

Services that are already running are not added to the new node. You will need to manually add the service tasks to the new node. There are two ways to add a service to a node.

- "Recreate the Stack" below
- "Reset the Stack" on the next page

Recreate the Stack

One way to add a service to a service node is to delete the current one and create a replacement.

To recreate an app node, issue the following commands.

```
docker stack rm applications;
docker stack deploy --with-registry-auth -c applications.yml applications;
```

To recreate an infrastructure node, issue the following commands.

```
docker stack rm infrastructures;
docker stack deploy --with-registry-auth -c infrastructures.yml
infrastructures;
```

Reset the Stack

Alternatively, you can scale a specific service to 1 replica and then reset it to multiple using the following commands.

```
docker service scale applications_ui_customer=1;
docker service scale applications_ui_customer=3;
```

Whether you choose to recreate or reset the stack, all service tasks will be placed across each node of the cluster.

Docker Cluster Commands

The following Docker commands can be used to inspect your setup to ensure everything is set up correctly and running.

To list the nodes in a cluster.

```
docker node ls;
```

To lists the networks on the machine and ensure the defender network was created.

```
docker networks ls;
```

To list running services and to check the number of replicas.

```
docker service ls;
```

To list service logs for all tasks.

```
docker service logs <service-name>;
```

To check service configuration (for advanced Docker swarm users).

```
docker service inspect <service-name>;
```

To list service tasks in order to check their state.

```
docker service ps;
```

To check task state and configuration.

```
docker inspect <task-id>; # output could provide an error
```

To run a service based on public image which provides insight into Docker swarm cluster via the web UI interface.

```
docker service create --name=viz \
--publish 8090:8080 \
--constraint=node.role==manager \
--mount=type=bind,src=/var/run/docker.sock, dst=/var/run/docker.sock \
appdefender/visualizer;
```

Note: Accessible on port 8090 on each cluster node. Use a firewall to close the service port to the Internet. You can check the UI and API services logs in the `rsyslog_defender/` folder specified in `defender_logs`.

Additional Installation Notes

This section provides additional configuration considerations .

Integrating LDAP Servers

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

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.

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 blank.

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 Java Keystore script. If you use certificates signed by a third party, use a Java Keystore.

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`)

Standalone Postgres Database (Optional)

If you decide to use a standalone Postgres Database installation rather than the Postgres container provided by Fortify Application Defender, you will need to do the following:

1. Make sure that the network is configured properly and all Fortify Application Defender hosts can reach the Postgres database.
2. Create a user with CREATEDB privilage.
3. Create a database using the user you just created.
4. Edit the following properties in the `appdefender.properties` file:

   postgres_ip: (defaults to postgres container) IP address of Postgres host in case standalone Postgres database

   postgres_dbname: (defaults to "appdef") Postgres database name to be used for Application Defender in case standalone Postgres database

   postgres_user: Valid username to connect to Postgres database

   postgres_password: Valid password to connect to Postgres database

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 service rm optional_storm_ui
```

- The Fortify Application Defender installation includes an open source container that is used solely for visualizing the Application Defender container status. A malicious user who gets access to the Web UI could get a full picture of Application Defender's cluster nodes and deployed services. Fortify suggests you stop the service when not in use. To stop the service:

  ```
  docker service rm viz
  ```

- 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.

- Docker Swarm technology allows you to access any published port using any host IP in the cluster. Because of this, any firewall rule you apply should be applied to every node in the cluster.
- Follow the instructions that Docker provides to secure your Docker daemon and secure Swarm Cluster deployment. For more information, see "Additional References" on page 40.

Logging Policy

Protect the appdefender.properties, applications.env, and infrastructures.env according to recommendation in <Application Defender System Hardening>

The following logging policy tables provide information about each of the Fortify Application Defender services.

Application Services

All Application Services use rsyslog as a logging driver. Rsyslog stores all logs in a default docker volume location. To find the exact path, execute the `next` command and note `Mountpoint` parameter. In most cases, it will be “`/var/lib/docker/volumes/applications_rsyslog_logs/_data`”`docker inspectapplications_rsyslog_logs`
`ls < Mountpoint >`

Svc #	Docker Image	Data Location Log Internal Daemon Rotation Policy	Container Log Rotation Policy
1	ui-customer	**Log**: Rsyslog Volume Folder e.g. /var/lib/docker/volumes/applications_rsyslog_logs/_data/ui_customer	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"

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 20.3.0)

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!