



Virtual Firewall Container

User Guide

August 22, 2018

Copyright 2018 by Qualys, Inc. All Rights Reserved.

Qualys and the Qualys logo are registered trademarks of Qualys, Inc. All other trademarks are the property of their respective owners.

Qualys, Inc.
919 E Hillsdale Blvd
4th Floor
Foster City, CA 94404
1 (650) 801 6100

Table of Contents

Preface	4
About Qualys.....	4
Qualys Support	4
Introduction	5
How it works.....	5
Deployment options	6
Basic Docker Deployment.....	6
Clustered Docker Deployment.....	7
Mixed Deployment	8
Get Started	9
Provision QVFC WAF Image	9
Locate QVFC image.....	9
Install QVFC Image in Local Registry	10
List Docker Images	10
Upload QVFC Image in Local Registry.....	10
Launch Docker Image	11
Basic launch	11
Run Container As a Daemon.....	11
Container Name.....	11
Port Exposure	11
Network Selection.....	12
Volume Association (Linux Socket for Docker Daemon)	12
Add System Constraints	13
List Running Containers	13
Stop a Running Container.....	13
Resume/Restart a Stopped Container	13
Remove a Container	14
Set up Access to Docker Services.....	14
Bind Local Linux Socket	14
Connect to Remote Docker host(s) using TCP/IP	15
Configure Docker Integration using Qualys WAF App.....	15
Provide WAF Configuration Option on Startup.....	15
Access WAF CLI in a docker container	16
Create a Dynamic Docker Server Pool.....	16
Assign the Dynamic Server Pool to a Web Application	17
Configure Dynamic Docker Server Pool.....	17
Enable Health Checks	18

Preface

Welcome to Qualys Cloud Platform! In this guide, we'll show you how to integrate Qualys Web Application Firewall (WAF) with Docker.

About Qualys

Qualys, Inc. (NASDAQ: QLYS) is a pioneer and leading provider of cloud-based security and compliance solutions. The Qualys Cloud Platform and its integrated apps help businesses simplify security operations and lower the cost of compliance by delivering critical security intelligence on demand and automating the full spectrum of auditing, compliance and protection for IT systems and web applications.

Founded in 1999, Qualys has established strategic partnerships with leading managed service providers and consulting organizations including Accenture, BT, Cognizant Technology Solutions, Deutsche Telekom, Fujitsu, HCL, HP Enterprise, IBM, Infosys, NTT, Optiv, SecureWorks, Tata Communications, Verizon and Wipro. The company is also a founding member of the [Cloud Security Alliance \(CSA\)](#). For more information, please visit www.qualys.com

Qualys Support

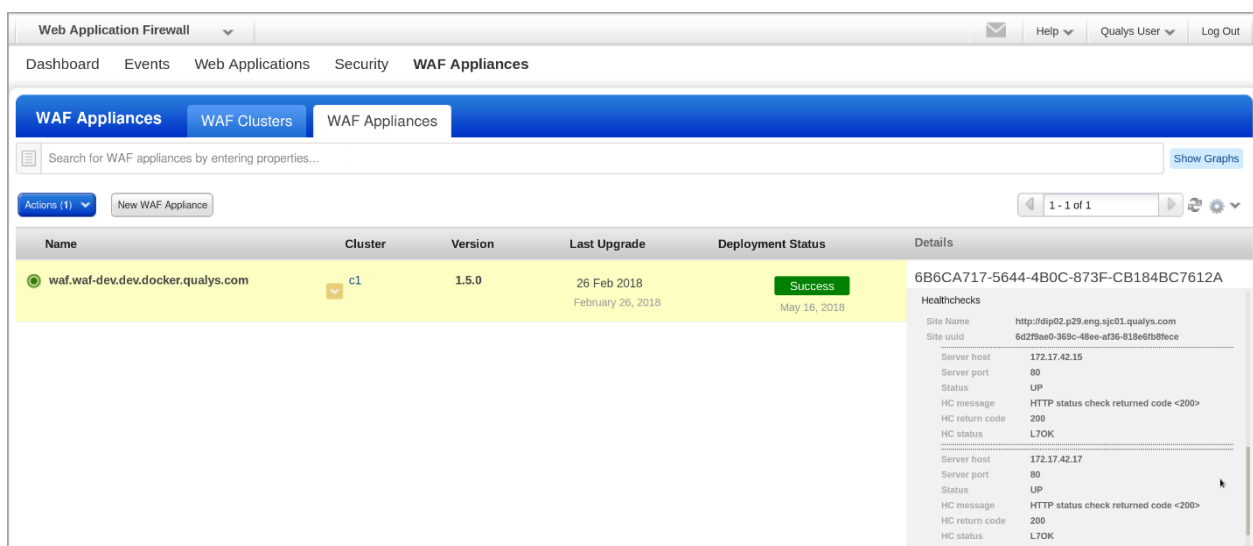
Qualys is committed to providing you with the most thorough support. Through online documentation, telephone help, and direct email support, Qualys ensures that your questions will be answered in the fastest time possible. We support you 7 days a week, 24 hours a day. Access support information at www.qualys.com/support/

Introduction

The Qualys Virtual Firewall Container (QVFC) is now available for Docker, in addition to the traditional Qualys Virtual Firewall Appliance (QVFA), extending the list of supported virtualized platforms with the likes of Amazon, Azure, Google, Hyper-V and VMware solutions. QVFC fully supports Docker and integrates with the Docker Host. WAF integration with Docker offers the following benefits along with continuous protection for your applications in the container:

- Monitors and analyzes multiple Docker hosts
- Tracks containers running on multiple hosts in real time
- Allows you to easily find container status, availability

It's easy to deploy QVFC on Docker, get deployment status and monitor security events. In the screenshot below, you'll see QVFC detected and imported 2 containers running web applications.



The screenshot shows the Qualys WAF Appliance monitoring interface. At the top, there's a navigation bar with 'Web Application Firewall' and tabs for 'Dashboard', 'Events', 'Web Applications', 'Security', and 'WAF Appliances'. Below this, there's a search bar and a table of WAF appliances. The table has columns for Name, Cluster, Version, Last Upgrade, Deployment Status, and Details. One appliance is listed: 'waf.waf-dev.docker.qualys.com' with cluster 'c1', version '1.5.0', last upgrade '26 Feb 2018', and deployment status 'Success'. The details panel on the right shows health checks for two server hosts, both with status 'UP' and return code '<200>'.

Name	Cluster	Version	Last Upgrade	Deployment Status	Details
waf.waf-dev.docker.qualys.com	c1	1.5.0	26 Feb 2018 February 26, 2018	Success May 16, 2018	6B6CA717-5644-4B0C-873F-CB184BC7612A Healthchecks Site Name: http://dip02.p29.eng.sjc01.qualys.com Site uuid: 6d278ae0-369c-48ee-af36-818e6fb8f8ce Server host: 172.17.42.15 Server port: 80 Status: UP HC message: HTTP status check returned code <200> HC return code: 200 HC status: L7OK Server host: 172.17.42.17 Server port: 80 Status: UP HC message: HTTP status check returned code <200> HC return code: 200 HC status: L7OK

WAF Appliance Monitoring Containers with Web Apps

How it works

QVFC is capable of automatically detecting containers on a docker host, and scaling WAF's server pool dynamically. It means when a backend container is crashed, stopped or removed on/from a docker host, QVFC can detect this change and remove the container from its endpoints list - or the other way around.

Qualys WAF integration with Docker not only works with QVFC but also with the regular Qualys Virtual Firewall Appliance (QVFA) on any type of virtualization platform, as long as it is able to reach out to docker host services (exposed through HTTP sockets), and of course to backend containers running on the host.

Note that you need to ensure that network devices such as routers, switches or firewalls are properly configured in order to allow seamless connectivity between QVFA and backend containers. Without proper configuration, QVFA will not be able to communicate with backend containers even if the application is up and running.

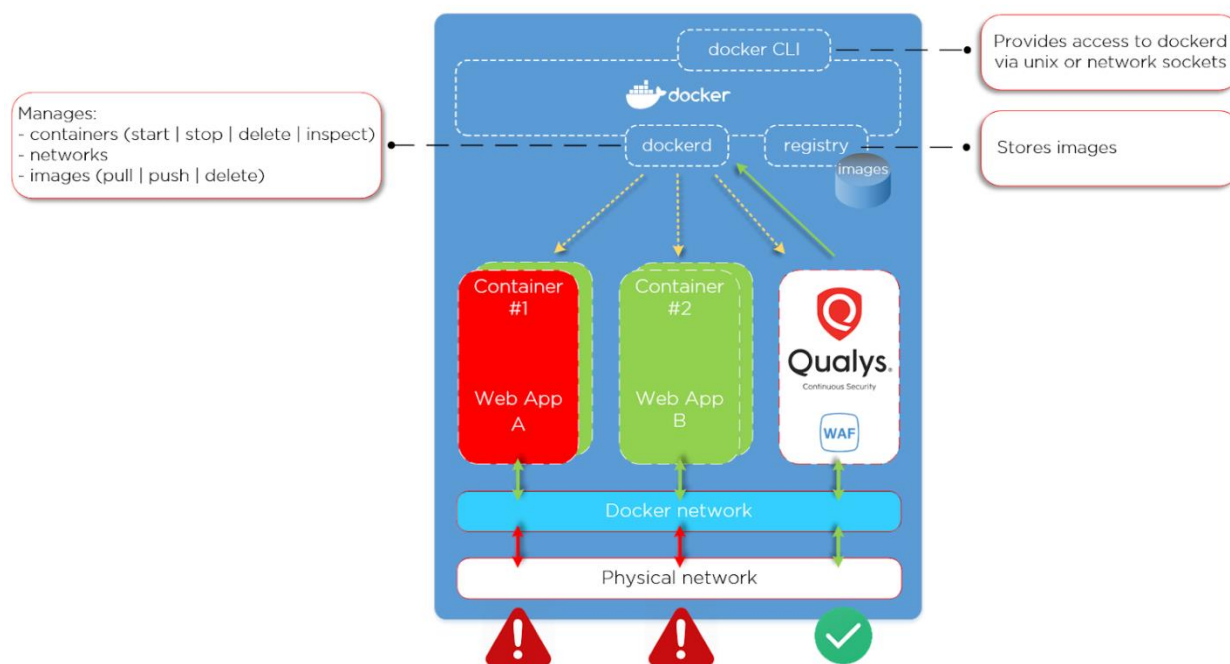
Deployment options

We recommend Basic, Clustered and Mixed docker deployment options. You can pick from these deployment options with the docker platform.

Basic Docker Deployment

A basic docker deployment consists of a single docker host running several backend containers and QVFC. In that type of scenario, only a single docker network is needed, and application containers do not need to be exposed to the external network. All the incoming client-side traffic is routed through QVFC which ensures that only the legitimate/safe traffic reaches server-side.

In this deployment model, QVFC sits in the same host, processes client-side HTTP/S traffic and forwards legitimate requests to web applications A and B served by containers 1 and 2 (server-side).

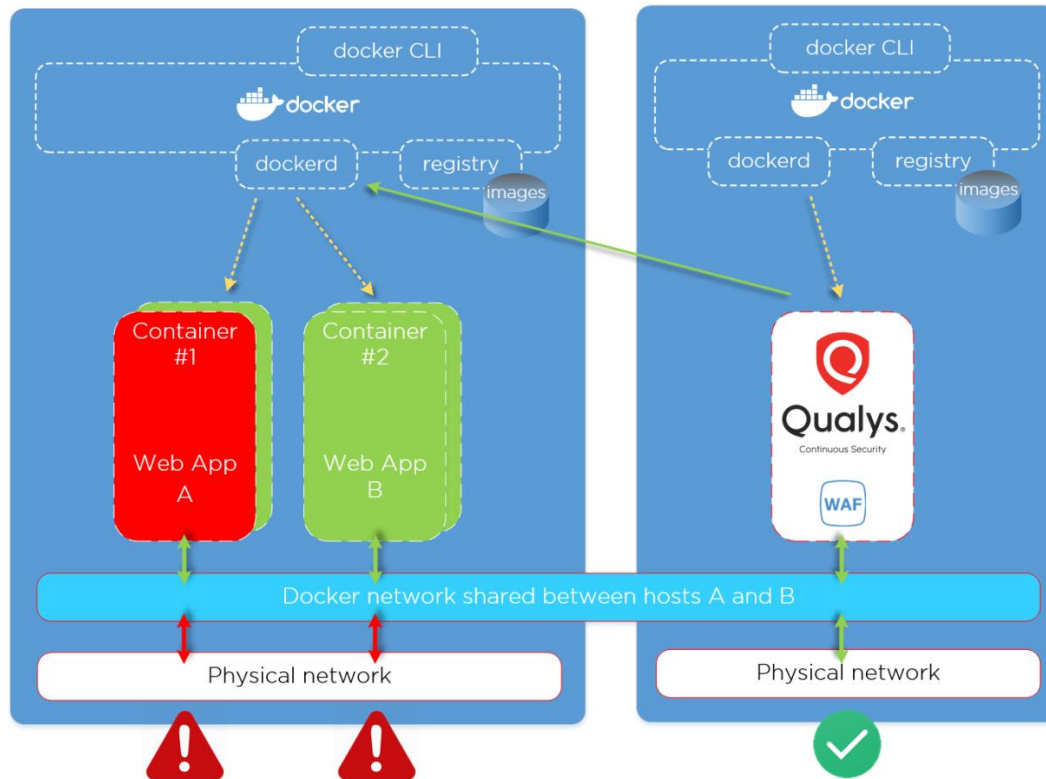


Basic Docker Deployment Overview

Containers 1 and 2 are NOT exposed to the public. They can't be reached directly from external networks; whereas QVFC exposes ports HTTP (tcp:80) and/or HTTPS (tcp:443) to the public.

Clustered Docker Deployment

Docker host can be part of a cluster (using Docker Swarm or Kubernetes for example), in which case several docker hosts are linked together. In that type of scenario, QVFC runs in a docker host which is distinct from the one bearing backend containers. You must ensure that QVFC can connect to the backend containers over a docker private network.

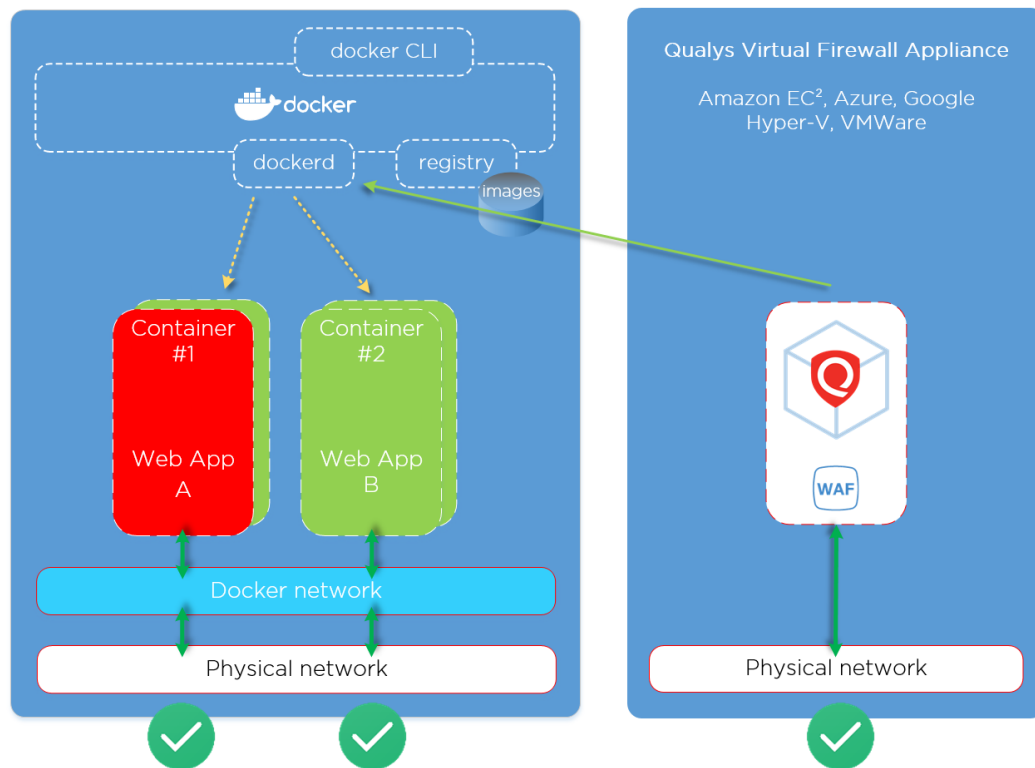


Clustered Docker Deployment Overview

Qualys WAF may be connected to Docker host A services in order to dynamically scan available containers and detect their IP internal addresses within docker virtual network.

Mixed Deployment

As long as network connectivity requirements are fulfilled, Qualys WAF may be deployed on any kind of virtualization platform in order to benefit from the docker integration feature. QVFA queries docker API the same way as QVFC does in order to retrieve the list of available containers.



Mixed Docker Deployment Overview

Qualys WAF may be connected to Docker host A services in order to dynamically scan available containers. In this case, Qualys WAF cannot reach web application containers unless the ports of the containers are exposed. As long as exposed port cannot be shared or accessible from external network, several docker hosts can be deployed and linked to a WAF appliance.

Note that any combination of deployments can be used since a WAF cluster can be composed of QVFC and QVFA nodes. Our only recommendation is to keep deployments as simple as possible in order to keep administration and maintenance easy and efficient.

Get Started

We will walk you through the steps of spinning a WAF container in a docker host and configuring it to inspect the HTTP/S traffic.

[Provision QVFC WAF Image](#)

[Launch Docker Image](#)

[Set up Access to Docker Services](#)

[Configure Docker Integration using Qualys WAF App](#)

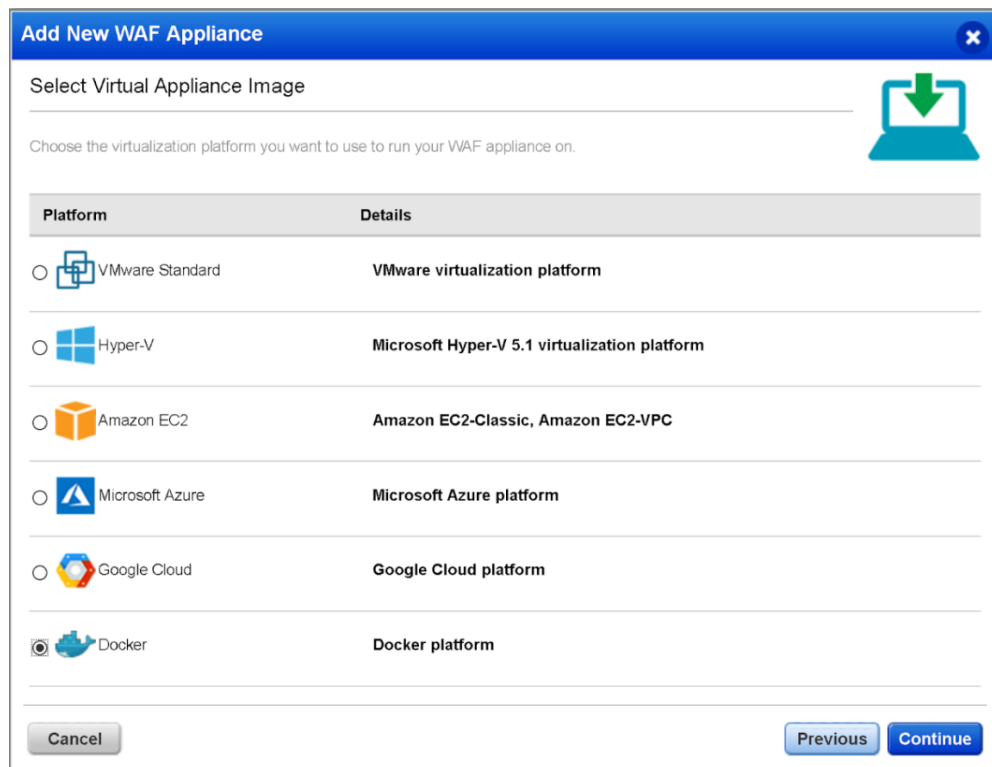
Provision QVFC WAF Image

To create a WAF container, you need to install the QVFC image into your local docker registry.

Locate QVFC image

Download the image from Qualys WAF UI. To do that, follow these steps:

- 1) Go to WAF Appliances screen and click New WAF Appliance.
- 2) Select an existing cluster or create a new one, and select Docker image.
- 3) Click Continue to start downloading QVFC.
- 4) Save the downloaded archive file to your docker host file system; or copy the download URL of the image if your docker host can reach download URL.



Add New WAF Appliance

Select Virtual Appliance Image

Choose the virtualization platform you want to use to run your WAF appliance on.

Platform	Details
<input type="radio"/> VMware Standard	VMware virtualization platform
<input type="radio"/> Hyper-V	Microsoft Hyper-V 5.1 virtualization platform
<input type="radio"/> Amazon EC2	Amazon EC2-Classic, Amazon EC2-VPC
<input type="radio"/> Microsoft Azure	Microsoft Azure platform
<input type="radio"/> Google Cloud	Google Cloud platform
<input checked="" type="radio"/> Docker	Docker platform

Qualys WAF UI

Install QVFC Image in Local Registry

After locating the image, the next step is to load it into the local docker registry. Use the “docker load --input <name of the archive file>” command on CLI to install the image from the archive to the local docker registry.

For example, if your archive file name is waf-prod-version.image.tar.gz, type:

```
$ docker load --input ./waf-prod-version.image.tar.gz
```

If you can directly load docker image from your local docker host, you may use this command for downloading and importing docker image in a single command line:

```
$ curl https://download-site/download-path/waf-prod-version.image.tar.gz | docker load
```

Each of those commands may take few minutes before completing.

List Docker Images

You can type the following command to verify whether the image is properly registered:

```
$ docker image
```

This command lists all the available images in the registry with image name and version (tags).

\$ docker images			
REPOSITORY	TAG	IMAGE ID	CREATED
waf-prod	1.5.0-1	8ec2160731e4	12 days ago
waf-dev	1.5.0-10	564bb7d87480	11 weeks ago
centos	6.9	e071bce628ba	6 months ago

Docker Images

Upload QVFC Image in Local Registry

- 1) Locate your QVFC repository URL or download the image archive from Qualys Platform.
- 2) Install QVFC into the local registry using the docker load command on CLI.

Note that you may have several versions of WAF docker images stored into your registry. We recommend you to delete the old image after installing new ones. Type the “docker rmi” command along with the tag name and version of the WAF docker image that you want to delete. For example,

```
$ docker rmi waf-prod:tag-and-version
```

where waf is the name of the tag and prod is the Qualys docker image version to delete.

Updating an image has no impact on existing containers. Once a container is started it will not receive image updates. This does not mean containers never upgrade. Qualys WAF is designed to be self-updatable, either through the CLI using the “system upgrade” command, or using Qualys UI.

In cases where upgrades affect Linux kernel/firmware, we recommend simply replacing the image.

Launch Docker Image

Basic launch

Running a docker image is usually done using the run command. Review the docker manual for more details about the docker run command.

Run Container As a Daemon

Docker allows launching a container for any image stored in the registry. The “run” command comes with many options for tweaking and customizing the startup. Type the “docker run -d” command along with the tag name and version of the QVFC image of the container instance. For example,

```
$ docker run -d waf-prod:tag-and-version
```

You can use the option -d to let the docker container run as a daemon process. This is recommended as launching a docker container in daemon mode prevents shut down of the container upon closing the command line interface. When you choose this option, docker prints container’s UUID as soon startup is completed. Without -d option, startup traces are dumped into a current shell, which can be useful for debugging, but not suitable for production. CTRL+C breaks and stops container execution.

Container Name

You can label containers to identify them. Use the --name option followed by the chosen container name when launching the container. For example, to name a container to Qualys-WAF, simply type:

```
$ docker run --name Qualys-WAF waf-prod:tag-and-version
```

Port Exposure

Each container may expose 0 or several ports externally. Ports on containers are exposed so that containers on a docker network can communicate with each other or with external hosts. In most cases, QVFC will expose port 80 and 443 to the public. In order to expose a port, docker needs to know which container port need to be exposed and possibly through a translated port on the docker host itself. We recommend avoiding complex port mappings for easier deployment and avoiding configuration issues.

In order to expose ports 80 and 443, add the following option to your docker run command:

```
$ docker run -p 80:80 -p 443:443 waf-prod:tag-and-version
```

This command creates a new WAF container and publishes the container ports 80 and 443 to the docker host. The traffic received on docker ports 80 and 443 will be forwarded to the published ports of the new WAF container.

Note: Docker host ports can be exposed only once. You cannot spin up 2 Qualys WAF (or any other applications) using same host port. For example, following commands give error.

```
$ docker run --name WAF -p 80:80 -p 443:443 waf-prod:tag-and-version
```

```
$ docker run --name WEB -p 80:8080 -p 443:443 my-custom-application
```

In case QVFC is reached by a load balancer (a docker load balancer for instance), client-side ports may not need to be exposed to anyone, since only the load balancer should be granted.

Similarly, applications protected by Qualys WAF do not need to expose any ports if QVFC and backend containers protected by WAF share the same network.

Network Selection

If your docker host does not run several networks, you can jump to the next step.

Several networks may be used/virtualized by docker host and it may be important to select the appropriate one. Once again, network selection needs to be set up at startup and cannot be tweaked easily once the container goes live. To select the correct network configuration, make sure to identify the network to use. The following command lists all the available networks..

```
$ docker network ls
```

NETWORK ID	NAME	DRIVER	SCOPE
dee5a244329f	bridge	bridge	local
d022468ac842	host	host	local
6c9b30431174	none	null	local
399b197f4a29	qualys_int	bridge	local

Note that the output may contain only a few networks when no custom networks are configured. In the run command, you can specify the name of the network for the WAF container. For example, to specify the “qualys_int” network for the container, add the following option to the docker run command:

```
$ docker run --network="qualys_int" waf-prod:tag-and-version
```

With some modifications, you can use the `--network` command to copy the current network configurations of an existing container to a new container. For example, to copy the current network configurations from an existing container named WAF1 to a new container named WAF2, type:

```
$ docker run --network="container:WAF1" --name WAF2 waf-prod:tag-and-version
```

Note that WAF1 does not need to be a live, running container.

More documentation about docker network is available here: <https://docs.docker.com/network/>.

Volume Association (Linux Socket for Docker Daemon)

It is possible to mount/bind/link local file system and docker container file system. Those options need to be specified at startup. Like port virtualization (port mapping), docker host file system entry needs to be mapped to a virtualized container file system entry. This option is used to bring robustness and persistence to file system even when the container is deleted.

Note that we do not recommend tweaking container file system and map docker host files/folders with container files/folders.

Here is a use case where you may need to expose docker services (available from a Linux socket) to a WAF container. Docker services are exposed using Linux socket (available locally and secured using classic Linux ACL) or using TCP/IP connection (available remotely from external hosts).

When a docker command is run, the CLI reaches the docker services using either the Linux socket or the URL provided. In this use case, the Linux socket is chosen (by default, the socket used is /var/run/docker.sock), we may want to grant docker service access to WAF container and thus bind /var/run/docker.sock Linux socket to container's file system.

In order to mount the volume, we will use the option “-v /var/run/docker.sock:/var/run/docker.sock”. The docker.sock file is the Linux socket on which the daemon listens to by default. The container use the Linux socket to communicate with the docker host.

You can use Linux ACL permissions to grant read/write access to the WAF container users/processes on that Linux socket. Or you may use the standard Linux commands such as groups or chmod to let container's processes use the Linux socket.

Add System Constraints

Without any specified options, containers created can grab ALL docker host system resources (RAM, CPU, ...) and this may be required to limit system resources consumption. Limits can be set on container startup.

Read quotas and limits available in docker configuration documentation such as <https://docs.docker.com/engine/reference/run/#cpu-quota-constraint>.

List Running Containers

To list running containers, docker provides the “docker ps” command. The option -a lists all the containers that are stopped or suspended.

```
$ docker ps -a
```

The output contains all containers (alive only if -a is not specified) with their respective startup image and version/tag, containers status and UUID.

Stop a Running Container

To stop a running container, use the stop command with the container UUID or name:

```
$ docker stop 83ecf4ff703b
```

```
$ docker stop WAF2
```

Resume/Restart a Stopped Container

You can use the restart command to start a stopped container. When you stop a container, the data in the container is still available on the docker host disks as long container is not deleted. For example, to start a stopped container named WAF2, run the following command:

```
$ docker restart WAF2
```

To restart a container with UUID 83ecf4ff703b, type:

```
$ docker restart 83ecf4ff703b
```

Remove a Container

We recommend to first stop the container by running the docker stop command and then run the docker rm command to remove the stopped container. You can use --force to forcefully remove the container. Following commands will remove containers stopped after the docker stop command.

```
$ docker rm 83ecf4ff703b
```

```
$ docker rm WAF2
```

Set up Access to Docker Services

As mentioned earlier, enabling a full integration with docker and unleashing the power of backend automation involves querying docker services, and you will want to provide QVFC/QVFA access to those docker services. On a single docker host, services are usually exposed using a Linux socket and the inode needs to be shared with and mounted to a container's file system at startup.

See “-v” startup option for mounting volumes into a container at startup. In case the WAF container (or WAF appliance) is not running on the same docker host, docker services must be exposed to WAF using HTTP web services

Please check for docker configuration in order to allow that option on your docker hosts:
<https://docs.docker.com/config/daemon/#configure-the-docker-daemon>.

Bind Local Linux Socket

Use the -v option to share inode used for Linux socket by docker daemon. Ensure that **Linux ACL** is properly configured so that this inode can be **read and or written** by WAF container. Now specify the socket inode location to WAF container either from startup option, or through WAF CLI using the “set” command. We recommend the latter option.

Startup option would be -e WAF_DOCKERD_URLS=unix:///var/run/docker.sock in order to define an environment variable on QVFC.

Otherwise, log in to WAF CLI and use the set command for setting docker daemon endpoints:

```
qualys waf # set
```

Syntax: set KEY=VALUE

Valid keys:

- proxy_url
- waf_service_url
- registration_code
- sem_syslog_addr
- waf_ssl_passphrase
- dockerd_urls

Here is an example for setting environment value for the WAF container:

```
qualys waf # set dockerd_urls=unix:///var/run/docker.sock
```

```
qualys waf # save
```

This saves the settings. Make sure to restart Agent for changes to take effect (agent restart).

```
qualys waf # agent restart
```

Agent 'restart' command queued. Enter 'agent status' to get current & queued job status(es).

Connect to Remote Docker host(s) using TCP/IP

Once docker services are exposed in HTTP over TCP/IP connections, Qualys WAF may reach this docker daemon as long as WAF knows TCP/IP endpoint (see CLI option `dockerd_urls` or `WAF_DOCKERD_URLS` WAF container startup environment value) and can reach those TCP/IP endpoints (check network configuration).

See [Mixed Deployment](#).

Configure Docker Integration using Qualys WAF App

Provide WAF Configuration Option on Startup

For easier deployment, you can use configuration script to set up Qualys WAF containers. Use CLI only for debug purpose or network advanced setup. While registering a WAF appliance, you need to provide registration code and other properties as appropriate using the variable below:

Variable	Description
WAF_REGISTRATION_CODE	(Required) in order to associate that container and your Qualys WAF subscription and your WAF cluster
WAF_SSL_PASSPHRASE	(Required if the appliance protects a site communicating over SSL) If your web application's primary or secondary base URL uses the HTTPS protocol, the Qualys Cloud Platform portal protects the private key by encrypting it with a 64 byte dedicated passphrase. This way, it's not accessible in clear on the Qualys Platform. This WAF_SSL_PASSPHRASE needs to be set on the appliance, for decrypting the key. Enter the passphrase in this format: WAF_SSL_PASSPHRASE=passphrase
WAF_SERVICE_URL	(Required) The URL of the Qualys Cloud Platform hosting your Qualys account. This can be set in CLI using set WAF_SERVICE_URL=...
WAF_PROXY_URL	(Required if a proxy is required for the WAF cluster to access the Qualys Cloud Platform) If the WAF needs to connect to the Qualys Cloud Platform through an HTTP proxy, please input the URL of the proxy. Enter the proxy URL in this format: PROXY_URL=proxy_url
WAF_SEM_SYSLOG_ADDR	(Optional) Configure this option for exporting log to external syslog server.
WAF_DOCKERD_URLS	(Optional) Configure this option to enable container integration with one or more docker hosts.

To specify a configuration key on startup, use option `-e` in your docker run command. This will expect `{key=value}` parameter to be set as environment value. WAF container will check those keys on startup and, if properly defined, will use them for configuration automation.

As an example, this command will spin up a new WAF container and associate it with a specified WAF cluster.

```
$ docker run -d \
-e WAF_REGISTRATION_CODE=056E28BA-C498-41C6-9FFA-AF543CD2B5A6 \
-e WAF_SERVICE_URL=https://your.rns.on.qualys.guard.com \ waf-prod:tag-and-version
```

Access WAF CLI in a docker container

You can identify a running container by UUID or name of the container. To connect to a container by UUID, use the docker exec command in conjunction with -ti option, which allows input/output data in your shell.

```
$ docker exec -ti 0dd4ce77b483a182ac01c2e49f629865100269a38841d4ef0ae3537ea1bfd6d2 /bin/waf-shell
```

```
$ docker exec -ti 0dd4ce77b483a182ac01c2e49f629865100269a38841d4ef0ae3537ea1bfd6d2 /bin/waf-shell
qualys waf # help

Commands (type help <command>):
=====
agent  deregister  help      passwd  routes  setup    ssh      unset
ca     diag        ifconfig  reboot  save    show     status   viewlog
core   exit        network   reset   set     shutdown sysinfo   waf

qualys waf #
```

WAF-Shell

In order to get the full benefits of docker integration once WAF sensors are properly linked to docker hosts, one or more dynamic server pools should be configured.

Create a Dynamic Docker Server Pool










Qualys WAF provides an option for creating dynamic server pools, which can fetch all the backend containers one WAF appliance should protect in a docker environment. The main difference between Regular and Dynamic docker server pools is that Regular server pool requires an exhaustive list of pool members or web container names running in the docker host; while with dynamic docker server pool, one does not have to specify any pool member nor container name. WAF appliance automatically detects web applications each time WAF container queries the docker services.

- 1) Go to Web Applications > Web servers.
- 2) Click the New Web Servers button.
- 3) Assign a Server Pool Name and click the Continue button.
- 4) Set the Docker platform toggle button to ON. Upon selecting this option, the Configuration is set to dynamic server pool server.
- 5) In Docker properties section, enter the name of the Docker image. All the web containers on a docker platform that are created from this image will be part of this server pool.
- 6) Click Save to save the dynamic server pool.

Assign the Dynamic Server Pool to a Web Application

Once dynamic server pool is enabled, you can assign it to a Web Application.

- 1) Go to Web Applications.
- 2) Click New Web Application or edit an existing application.
- 3) In the Applications tab, go to Web Servers section and select the Server Pool that you created.

		Copy of http://dip02.p29.eng.sjc01.qualys.com	1	roundrobin
		Copy of Docker cluster for art-hq.intranet.qualys.com:5001/rns0/httpd	dynamic	roundrobin
		Docker cluster for art-hq.intranet.qualys.com:5001/rns0/httpd	 dynamic	roundrobin
		http://dip02.p29.eng.sjc01.qualys.com	1	roundrobin

Dynamic Server Pool

Configure Dynamic Docker Server Pool

On each WAF appliance, docker services will be queried (every 15 seconds if no configuration updates or system updates are pending) and every docker containers matching the filters will become pool members across all web applications that use this dynamic server pool.

The mandatory filter is the base docker image name used to spin up the application. Please note that docker image tag/version must not be added to image name but full image name is required like, my.repository.com:12345/path/to/image or docker.io/wordpress or docker.io/httpd.

You may filter backend containers in a dynamic docker server pool based on a container name when several web servers are deployed on a docker host based on the same docker image but provides different applications. This filter is mandatory in such a scenario.

For example, for container A, started from a Tomcat docker image running application “abc”, and container B started from that same docker image and running another application “xyz” ; both containers may be secured using Qualys WAF but security policies may not be actually the same. In such a scenario, you must define two separate applications and assign each different dynamic server pools.

The main difference between these 2 server pools should be the container name filter. Note the name filter string will be searched across all container names and if found, the container will be added to the server pool. For example, if name's filter is set to AAA then a container named BBBAACCC will be considered as valid. Existing parameters for server pools do not change, the port is still required (port applications are exposed on) and HTTP/HTTPS selector need to be properly configured. Load balancing algorithm is also still customizable.

Web Servers Creation

Turn help tips: On | Off Launch help

Step 2 of 3

1 Web Servers Details

2 Configuration

3 Review And Confirm

Web Servers configuration

Application Servers (*) REQUIRED FIELDS

Here you define the server-side properties. Select a port, protocol (http or https), and a list of servers along with a load-balancing method.

Port*

80

Protocol HTTPS

OFF

Docker platform

ON

Docker properties

Docker image*

docker.io/elasticsearch

Docker name

(optional)

Load-balancing

roundrobin

Cancel

Previous Continue

Create Server Pool

Enable Health Checks

As dynamic pool members can be short-lived, it is not possible to get the list and count of pool members. Pool member count will be set to “dynamic” on the UI in order not to highlight an information that is quite volatile.

You may have more feedback regarding web servers running in containers connected by WAF using Health Checks. Enable this health check option on the WAF site deployed in order to get a snapshot view of pool member reached for each web application deployed.

- 1) Go to WAF Appliances > WAF Appliances.
- 2) Select a WAF appliance in the list to see Health Checks statuses for deployed web applications even if server pool used is dynamic.

In the following screen provided, WAF appliance has detected and successfully connected to 2 containers running web application to secure. We have IP used for those pool members 172.17.42.15 and 172.17.42.17 that belongs to virtual docker network configured 172.17.42.0/24. WAF appliance needs to be able to reach and connect to that network 172.17.42.0/24 in order to forward traffic to web application servers.

Virtual Firewall Container

18

Web Application Firewall
Help
Qualys User
Log Out

Dashboard
Events
Web Applications
Security
WAF Appliances

WAF Appliances
WAF Clusters
WAF Appliances

Search for WAF appliances by entering properties...
Show Graphs

Actions (1)
New WAF Appliance
1 - 1 of 1

Name	Cluster	Version	Last Upgrade	Deployment Status	Details
waf.waf-dev.dev.docker.qualys.com	c1	1.5.0	26 Feb 2018 February 26, 2018	Success May 16, 2018	6B6CA717-5644-4B0C-873F-CB184BC7612A Healthchecks Site Name http://dip02.p29.eng.sjc01.qualys.com Site uuid 6d2f9ae0-369c-48ee-af36-818e6fb8fcec <hr/> Server host 172.17.42.15 Server port 80 Status UP HC message HTTP status check returned code <200> HC return code 200 HC status L7OK <hr/> Server host 172.17.42.17 Server port 80 Status UP HC message HTTP status check returned code <200> HC return code 200 HC status L7OK

Health Check Status