Container Security

Sensor Deployment Guide

September 4, 2019

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

Table of Contents

About this Guide

Welcome to Qualys Container Security! We'll help you get acquainted with the Qualys solutions for securing your Container environments like Images, Containers and Docker Hosts using the Qualys Cloud Security Platform.

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 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 online support information at www.qualys.com/support/.

About Container Security Documentation

This document provides information on deploying the sensor on MAC, CoreOS, and various orchestrators and cloud environments.

For information on using the Container Security UI to monitor vulnerabilities in Images, Containers, and Registries, refer to the Qualys Container Security User Guide.

For information on using the Container Security API, refer to the Qualys Container Security API Guide.

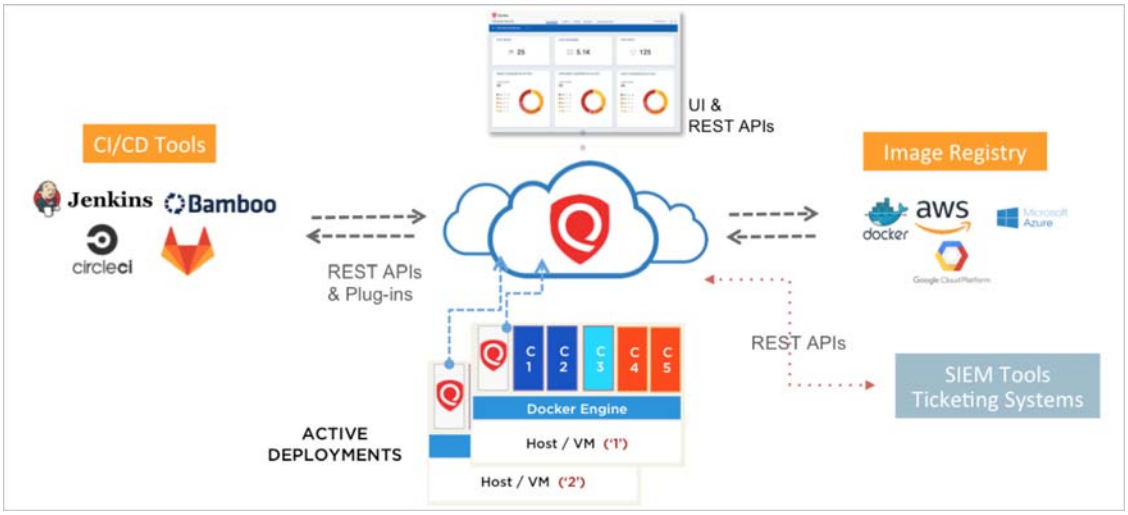
For information on deploying the sensor in CI/CD environments refer to:

- Qualys Vulnerability Analysis Plugin for Jenkins

- Qualys Vulnerability Analysis Plugin for Bamboo

Container Security Overview

Qualys Container Security provides discovery, tracking, and continuously protecting container environments. This addresses vulnerability management for images and containers in their DevOps pipeline and deployments across cloud and on-premise environments.



With this version, Qualys Container Security supports

- Discovery, inventory, and near-real time tracking of container environments

- Vulnerability analysis for images and containers

- Vulnerability analysis for registries

- Integration with CI/CD pipeline using APIs (DevOps flow)

- Uses 'Container Sensor' – providing native container support, distributed as a docker image

Qualys Container Sensor

The sensor from Qualys is designed for native support of Docker environments. Sensor is packaged and delivered as a Docker Image. Download the image and deploy it as a Container alongside with other application containers on the host.

The sensor is docker based, can be deployed on hosts in your data center or cloud environments like AWS ECS, Azure Container Service or Google Container Service. Sensor currently is only supported on Linux Operating systems and requires docker daemon of version 1.12 and higher to be available.

Since they are docker based, the sensor can be deployed into orchestration tool environments like Kubernetes, Mesos or Docker Swarm just like any other application container.

Upon installation, the sensor does automatic discovery of Images and Containers on the deployed host, provides a vulnerability analysis of them, and additionally it monitors and reports on the docker related events on the host. The sensor lists and scans registries for vulnerable images. The sensor container runs in non-privileged mode. It requires a persistent storage for storing and caching files.

Currently, the sensor only scans Images and Containers, for getting a vulnerability posture on the Host, you would require Qualys Cloud Agents or a scan through Qualys Virtual Scanner Appliance.

Note: Qualys Container Security does not support scanning images in Docker-in-Docker setup as the sensor cannot listen to events generated by the inner Docker.

What data does Container Security collect?

The Qualys Container Security sensor fetches the following information about Images and Containers in your environment:

- **Inventory of Images and Containers** in your environment from commands such as **`docker ps`** that lists all containers.

- **Metadata information** about Images and Containers from commands such as **`docker inspect`** and **`docker info`** that fetches low level information on docker objects.

- **Event information** about Images and Containers from the docker host for docker events like created, started, killed, push, pull, etc.

- **Vulnerabilities** found on Images and Containers. This is the output of the vulnerability management manifests run for identifying vulnerability information in Images and Containers. This is primarily software package listing, services running, ports, etc.

For example, package manager outputs like **`rpm -qa`**, **`npm`**. This is supported across various Linux distributions (CentOS, Ubuntu, CoreOS, etc) and across images like Python, NodeJS, Ruby, and so on.

Get Started

Follow the steps to get started with Container Security.

Qualys Subscription and Modules required

You would require “Container Security” (CS) module enabled for your account. Additionally, in order to get vulnerabilities for the hosts that run the containers, you would need to enable Vulnerability Management (VM), either via Scanner Appliance or Cloud Agent.

System support

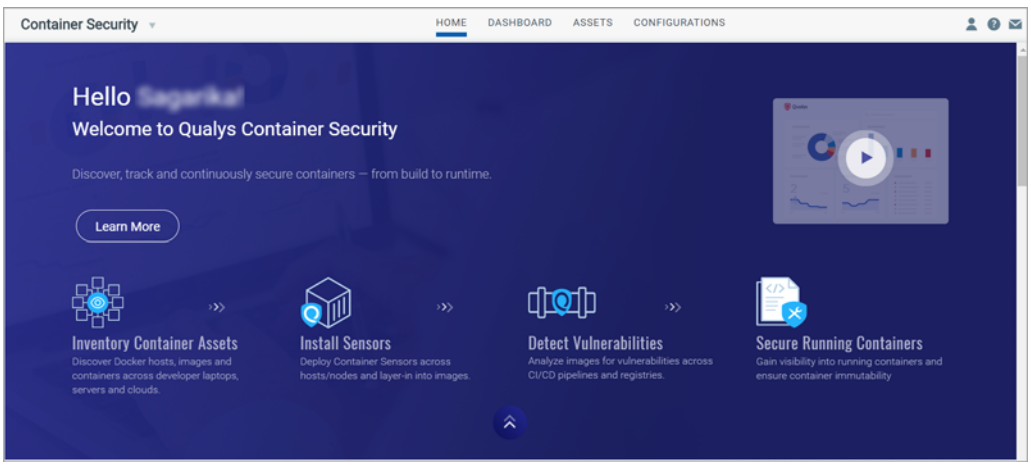
Container Security supports these systems running Docker version 1.12 or later.

- Ubuntu
- Red Hat Enterprise Linux
- Debian
- CentOS
- MAC
- CoreOS

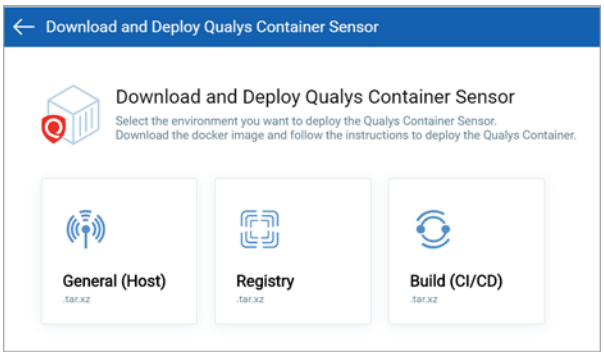
Deploying Container Sensor

Log into your Qualys portal with your user credentials. Select Container Security from the module picker.

As a first time user, you’ll land directly into the Home page.



Go to Configurations > Sensors, and then click Download to download the sensor tar file. You can see various sensor types:



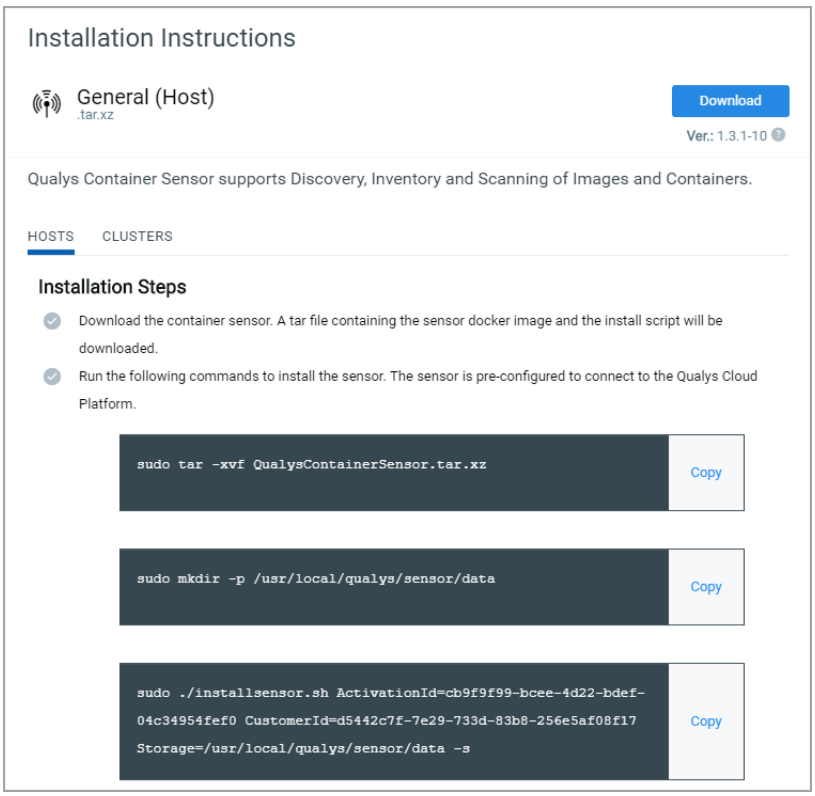
General (Host) Sensor: Scan any host other than registry / build (CI/CD).

Registry Sensor: Scan images in a registry (public / private).

Build (CI/CD) Sensor: Scan images on CI/CD pipeline (Jenkins / Bamboo).

For Registry you need to append the install command with **--registry-sensor** or **-r**

For CI/CD you need to append the install command with **--cicd-deployed-sensor** or **-c**



Download the QualysContainerSensor.tar.xz file and run the commands generated directly from the screen on the docker host. Note the requirements for installing the sensor, the sensor needs a minimum of 1 GB persistent storage on the host.

A quick overview of the “installsensor.sh” script command line parameters options:

- **ActivationId** : Activation Id for the container sensor, auto-generated based on your subscription.

- **CustomerId** : Qualys subscription's customerId, auto-generated based on your subscription.

- **Storage** : Directory where the sensor would store the files. Default: /usr/local/qualys/sensor/data. Create it if not already available or you can specify a custom directory location.

- **ImageFile** : Location of the Sensor ImageFile, defaults to the local directory [Optional]

- **LogLevel** : Configuration to set the logging level for sensor, accepts 0 to 5 [Optional]

- **HostIdSearchDir** : Directory to map the marker file created by Qualys Agent or Scanner appliance on the host, update if modified [Optional]

- **CpuUsageLimit** : CPU usage limit in percentage for sensor. Valid range is in between 0-100 [Optional]

- **ConcurrentScan** : Number of docker/registry asset scans to run in parallel [Optional]

- **Proxy** : IPv4/IPv6 address or FQDN of the proxy server [Optional]

- **ProxyCertFile** : Proxy certificate file path [Optional]

  ProxyCertFile is applicable only if Proxy has valid certificate file. If this option is not provided then Sensor would try to connect to the server with given https Proxy settings only.

  If only ProxyCertFile is provided without Proxy then Sensor would simply ignore the ProxyCertFile and it would try to connect to the server without any https proxy settings.

- **--silent or -s** : Run installsensor.sh in non-interactive mode [Optional]

- **--disable-auto-update** : Do not let sensor update itself automatically [Optional]

- **--cicd-deployed-sensor or -c** : Run Sensor in CI/CD environment

- **--registry-sensor or -r** : Run sensor to list and scan registry assets

- **--enable-console-logs** : Print logs on console. These logs can be retrieved using the docker logs command.

- DockerHost : IPv4 address or FQDN:Port#. The address on which the docker daemon is configured to listen. [optional]

- DockerSocketDirectory : Docker socket directory path. [optional]

For more information on installing the registry sensor, refer to the Qualys Container Sensor User Guide.

For information on deploying the sensor in CI/CD environments refer to:

- Qualys Vulnerability Analysis Plugin for Jenkins

- Qualys Vulnerability Analysis Plugin for Bamboo

See About Container Security Documentation.

Proxy Support

The install script asks for proxy configuration. You need to provide the IP Address/FQDN and port number along with the proxy certificate file path. For example,

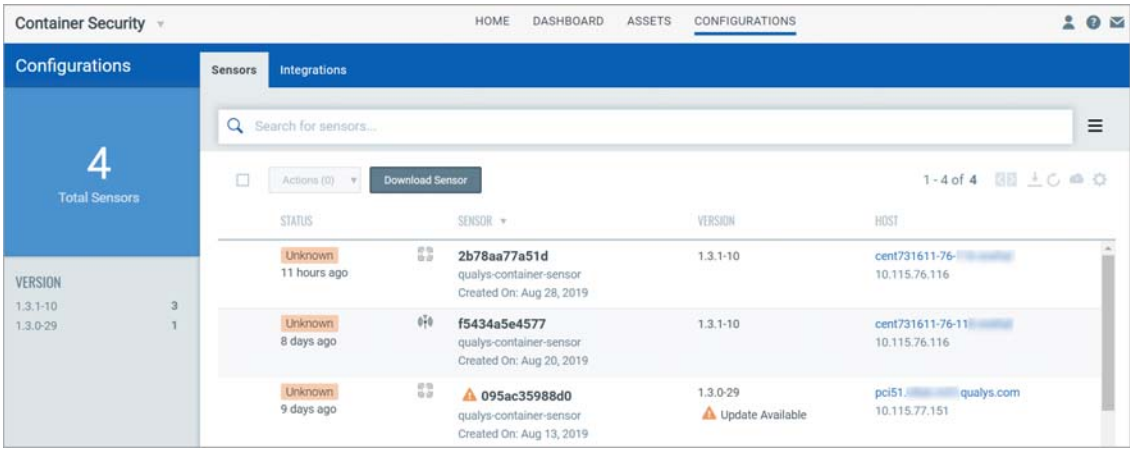
```
Do you want connection via Proxy [y/N]: y
Enter Https Proxy settings [<IP Address>:<Port #>]: 10.xxx.xx.xx:3xxx
Enter Https Proxy certificate file path: /etc/qualys/cloud-
agent/cert/ca-bundle.crt
```

Your proxy server must provide access to the Qualys Cloud Platform (or the Qualys Private Cloud Platform) over HTTPS port 443. Go to Help > About to see the URL your hosts need to access.

Sensor network configuration

The sensor is pre-configured with the Qualys URL and the subscription details it needs to communicate to. In order for the sensor to communicate to Qualys, the network configuration and firewall needs to provide accessibility to Qualys domain over port 443.

After successful installation of the Sensor, the sensor is listed under Configurations > Sensors where you can see its version, status, etc. and access details.



Additionally, you can Download the sensor from the link under Configurations > Sensors.

Installing the sensor on a MAC

You can install the Qualys Container Sensor on a MAC.

Here are the steps:

Download the QualysContainerSensor.tar.xz file using the “Download and Install Qualys Container Sensor” link on the Get Started page or from the Configurations > Sensors tab on Qualys Cloud Platform.

Copy the file to the target MAC host.

Once you copy the file on the target host, run the following commands in sequence:

This command extracts the tar file.

```
sudo tar -xvf QualysContainerSensor.tar.xz
```

This command creates the directory where the sensor data like configuration, manifest, logs, and setup is stored.

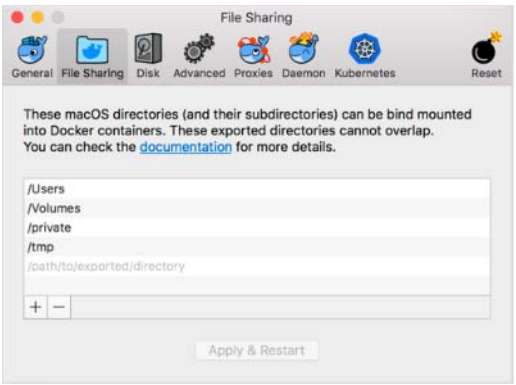
```
sudo mkdir -p /tmp/qualys/sensor/data
```

This command provides the required permissions to the directory to run the installer script.

```
sudo chmod -R 777 /tmp/qualys/sensor/data
```

If you want to specify a custom location for storage, ensure that the Docker's File Sharing is enabled for the same. On your MAC host, go to Docker > Preferences > File Sharing, add the custom path e.g. /usr/local/qualys/sensor/data, then click Apply & Restart.

Enabling file sharing is required only if the custom location is NOT from /Users, /Volumes, /private or /tmp.



To avoid this step, we recommend using Storage=/tmp/qualys/sensor/data and HostIdSearchDir=/private/etc/qualys during sensor install.

That way you can leverage the existing shared location with docker, without the need of additional configuration to launch the CS Sensor.

If you are using a custom location, provide permissions to the directory to run the installer script.

For example,

```
sudo chmod -R 777 /usr/local/qualys/sensor/data
```

The following commands install the sensor. Notice that the command includes the Activation ID and your Customer ID, both generated based on your subscription. The Storage parameter specifies where to install the sensor. Ensure that the HostIdSearchDir exists, otherwise the installer script will throw an error.

Use the following command to install a General Sensor:

```
./installsensor.sh ActivationId=d5814d5f-5fd2-44ec-8969-e03cc58a4ef5
CustomerId=6f35826e-4430-d75e-8356-c444a0abbb31
HostIdSearchDir=/private/etc/qualys Storage=/tmp/qualys/sensor/data -s
```

Use the following command to install a Registry Sensor:

```
./installsensor.sh ActivationId=d5814d5f-5fd2-44ec-8969-e03cc58a4ef5
CustomerId=6f35826e-4430-d75e-8356-c444a0abbb31
HostIdSearchDir=/private/etc/qualys Storage=/tmp/qualys/sensor/data -s -
-registry-sensor
```

Use the following command to install a CI/CD Sensor:

```
./installsensor.sh ActivationId=d5814d5f-5fd2-44ec-8969-e03cc58a4ef5
CustomerId=6f35826e-4430-d75e-8356-c444a0abbb31
HostIdSearchDir=/private/etc/qualys Storage=/tmp/qualys/sensor/data -s -
-cicd-deployed-sensor
```

Installing the sensor on CoreOS

You can install the Qualys Container Sensor on CoreOS.

Here are the steps:

Download the QualysContainerSensor.tar.xz file using the "Download and Install Qualys Container Sensor" link on the Get Started page or from the Configurations > Sensors tab on Qualys Cloud Platform.

Copy the file to the target host.

Once you copy the file on the target host, run the following commands in sequence:

This command extracts the tar file.

```
sudo tar -xvf QualysContainerSensor.tar.xz
```

This command creates the directory where the sensor data like configuration, manifest, logs, and setup is stored.

```
sudo mkdir -p /var/opt/qualys/sensor/data
```

Note: You need to set the directory path /var/opt/qualys/sensor/data to Storage which is writable on CoreOS.

This command provides the required permissions to the directory to run the installer script.

```
sudo chmod -R 777 /var/opt/qualys/sensor/data
```

The following commands install the sensor. Notice that the command includes the Activation ID and your Customer ID, both generated based on your subscription. The Storage parameter specifies where to install the sensor.

Use the following command to install a General Sensor:

```
Sudo ./installsensor.sh ActivationId=d5814d5f-5fd2-44ec-8969-
e03cc58a4ef5 CustomerId=6f35826e-4430-d75e-8356-c444a0abbb31
Storage=/var/opt/qualys/sensor/data/ -s
```

Use the following command to install a Registry Sensor:

```
Sudo ./installsensor.sh ActivationId=d5814d5f-5fd2-44ec-8969-
e03cc58a4ef5 CustomerId=6f35826e-4430-d75e-8356-c444a0abbb31
Storage=/var/opt/qualys/sensor/data/ -s --registry-sensor
```

Use the following command to install a CI/CD Sensor:

```
Sudo ./installsensor.sh ActivationId=d5814d5f-5fd2-44ec-8969-
e03cc58a4ef5 CustomerId=6f35826e-4430-d75e-8356-c444a0abbb31
Storage=/var/opt/qualys/sensor/data/ -s --cicd-deployed-sensor
```

Deploying sensor in Kubernetes

Integrate the Container Sensor into the DaemonSet like other application containers and set the replication factor to 1 to ensure there is always a sensor deployed on the Docker Host. This information is applicable for Amazon Elastic Container Service for Kubernetes (Amazon EKS), Google Kubernetes Engine (GKE), and Azure Kubernetes Service (AKS).

Perform the following steps for creating a DaemonSet for the Qualys sensor to be deployed in Kubernetes.

Note: Ensure that the Container Sensor has read and write access to the persistent storage and the docker daemon socket.

Download the **QualysContainerSensor.tar.xz** file from Qualys Cloud Portal on a Linux computer.

Untar the sensor package:

```
sudo tar -xvf QualysContainerSensor.tar.xz
```

Use the following commands to push the qualys sensor image to a repository common to all nodes in the Kubernetes cluster:

```
sudo docker load -i qualys-sensor.tar
```

```
sudo docker tag <IMAGE NAME/ID> <URL to push image to the repository>
```

For example,

```
sudo docker tag c3fa63a818df mycloudregistry.com/container-
sensor:qualys-sensor-xxx
```

```
sudo docker push <URL to push image to the repository>
```

For example,

```
sudo docker push mycloudregistry.com/container-sensor:qualys-sensor-xxx
```

Note: Do not use the examples as it is. You need to replace the registry/image path with your own.

Modify the **cssensor-ds.yml** file (extracted from QualysContainerSensor.tar.xz) to provide values for the following parameters. In order for the yml file to work properly, ensure that you do not remove/comment the respective sections mentioned below.

Ensure that all Kubernetes nodes have the latest Qualys sensor image from the URL provided.

```
containers:
   - name: qualys-container-sensor
     image: mycloudregistry.com/container-sensor:qualys-sensor-xxx
     args: ["--k8s-mode"]
```

If you want to deploy the sensor for CI/CD environment provide the **args** value as:

```
args: ["--k8s-mode","--cicd-deployed-sensor"]
```

If you want to deploy a Registry Sensor provide the **args** value as:

```
args: ["--k8s-mode","--registry-sensor"]
```

If you want print logs on the console, provide `"--enable-console-logs"` as an additional value in **args**.

To restrict the cpu usage to a certain value, change the following: (Optional)

Under **resources** specify the following:

```
limits:
  cpu: "0.2" # Default CPU usage limit(20% of overall CPU
               available) on each node for sensor.
```

For example,

For limiting the cpu usage to 5%, set resources:limits:cpu: "0.05", this limits overall cpu usage to 5% of a CPU on a node.

If there are multiple processors on a node, set the resources:limits:cpu value accordingly.

For example,

You have 5 CPUs on system and you want to set 5% of overall capacity of system, set the CPU usage limit to 5 x 0.05 = “0.25”.

To disable any CPU usage limit, set resources:limits:cpu value to 0.

Under **env** specify the following:

Activation ID (Required)

```
- name: ACTIVATIONID
  value: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
```

Customer ID (Required)

```
- name: CUSTOMERID
  value: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
```

Specify proxy information, or remove if not required:

```
- name: qualys_https_proxy
  value: proxy.localnet.com:3128
```

Under **volumes** specify the proxy cert path, or remove if not required:

```
- name: proxy-cert-path
  hostPath:
    path: /root/cert/proxy-certificate.crt
    type: File
```

Activation ID and Customer ID are required. Use the Activation ID and Customer ID from your subscription.

If you are using a proxy, ensure that all Kubernetes nodes have a valid certificate file for the sensor to communicate with the Container Management Server.

If you are not using a proxy and you have removed the above mentioned parts, you can remove the following part from **volumeMounts** as well:

```
- mountPath: /etc/qualys/qpa/cert/custom-ca.crt
  name: proxy-cert-path
```

Once you have modified the **cssensor-ds.yml** file, run the following command on Kubernetes master to create a DaemonSet:

```
kubectl create -f cssensor-ds.yml
```

If you need to uninstall Qualys Container Sensor, run the following command on Kubernetes master:

```
kubectl delete -f cssensor-ds.yml
```

Updating the sensor deployed in Kubernetes

You can update the Container Sensor DaemonSet to the latest version in Kubernetes. This information is applicable for Amazon Elastic Container Service for Kubernetes (Amazon EKS), Google Kubernetes Engine (GKE), and Azure Kubernetes Service (AKS).

Ensure that the Container Sensor has read and write access to the persistent storage and the docker daemon socket.

Perform the following steps on Kubernetes master for updating the Container Sensor.

Note: Ensure that the Container Sensor DaemonSet is running in the Kubernetes environment.

Download the **QualysContainerSensor.tar.xz** file from Qualys Cloud Portal on Kubernetes master.

Untar the sensor package:

```
sudo tar -xvf QualysContainerSensor.tar.xz
```

Copy the Sensor version from the **version-info** file (extracted from QualysContainerSensor.tar.xz)

Modify the **cssensor-ds.yml** file (extracted from QualysContainerSensor.tar.xz) to provide values for the following parameters. In order for the yml file to work properly, ensure that you do not remove/comment the respective sections mentioned below.

Ensure that all Kubernetes nodes have the latest Qualys sensor image from the URL provided.

```
containers:
   - name: qualys-container-sensor
     image: mycloudregistry.com/qualys/sensor:1.2.3-63
     args: ["--k8s-mode"]
```

The image value must be in the format:

```
registryurl/qualys/sensor:<version-info>
```

If you want to deploy the sensor for CI/CD environment provide the **args** value as:

```
args: ["--k8s-mode","--cicd-deployed-sensor"]
```

If you want to deploy a Registry Sensor provide the **args** value as:

```
args: ["--k8s-mode","--registry-sensor"]
```

If you want print logs on the console, provide "`--enable-console-logs`" as an additional value in **args**.

To restrict the cpu usage to a certain value, change the following: (Optional)

Under **`resources`** specify the following:

```
limits:
  cpu: "0.2" # Default CPU usage limit(20% of overall CPU
               available) on each node for sensor.
```

For example,

For limiting the cpu usage to 5%, set resources:limits:cpu: "0.05", this limits overall cpu usage to 5% of a CPU on a node.

If there are multiple processors on a node, set the resources:limits:cpu value accordingly.

For example,

You have 5 CPUs on system and you want to set 5% of overall capacity of system, set the CPU usage limit to 5 x 0.05 = “0.25”.

To disable any CPU usage limit, set resources:limits:cpu value to 0.

Under **env** specify the following:

Activation ID (Required: Use the same Activation ID provided in the existing Container Sensor DaemonSet that you are upgrading)

```
- name: ACTIVATIONID
  value: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
```

Customer ID (Required: Use the same Customer ID provided in the existing Container Sensor DaemonSet that you are upgrading)

```
- name: CUSTOMERID
  value: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
```

Specify proxy information, or remove if not required:

```
- name: qualys_https_proxy
  value: proxy.localnet.com:3128
```

Under **volumes** specify the proxy cert path, or remove if not required:

```
- name: proxy-cert-path
  hostPath:
    path: /root/cert/proxy-certificate.crt
    type: File
```

Activation ID and Customer ID are required. Use the Activation ID and Customer ID from your subscription.

If you are using a proxy, ensure that all Kubernetes nodes have a valid certificate file for the sensor to communicate with the Container Management Server.

If you are not using a proxy and you have removed the above mentioned parts, you can remove the following part from **volumeMounts** as well:

```
- mountPath: /etc/qualys/qpa/cert/custom-ca.crt
  name: proxy-cert-path
```

Once you have modified **cssensor-ds.yml**, save the file, and then perform docker login to the registry on Kubernetes master before running the update script (k8s-rolling-update.sh).

For example,

```
docker login mycloudregistry.com
```

The registry should be accessible from all Kubernetes nodes and the Kubernetes master from where the update is being performed.

To update the Container Sensor DaemonSet to the latest version, run the following command on Kubernetes master:

```
./k8s-rolling-update.sh Registry_Url=mycloudregistry.com
```

Note: k8s-rolling-update.sh will do docker load, docker tag and docker push to the registry.

Deploying sensor in Docker Swarm

Integrate the Container Sensor into the DaemonSet like other application containers and set the replication factor to 1 to ensure there is always a sensor deployed on the Docker Host.

Perform the following steps for creating a DaemonSet for the Qualys sensor to be deployed in Docker Swarm.

Download the **QualysContainerSensor.tar.xz** file from Qualys Cloud Portal on a Linux computer.

Untar the sensor package:

```
sudo tar -xvf QualysContainerSensor.tar.xz
```

Use the following commands to push the qualys sensor image to a repository common to all nodes in the Docker Swarm cluster:

```
sudo docker load -i qualys-sensor.tar
```

```
sudo docker tag <IMAGE NAME/ID> <URL to push image to the repository>
```

For example,

```
sudo docker tag c3fa63a818df myregistry.com/qualys_sensor:xxx
```

```
sudo docker push <URL to push image to the repository>
```

For example,

```
sudo docker push myregistry.com/qualys_sensor:xxx
```

Note: Do not use the examples as it is. You need to replace the registry/image path with your own.

Modify the **cssensor-swarm-ds.yml** file (extracted from QualysContainerSensor.tar.xz) to provide values for the following parameters. In order for the yml file to work properly, ensure that you do not remove/comment the respective sections mentioned below.

Ensure that all masters and worker nodes have the latest Qualys sensor image from the URL provided.

```
qualys-container-sensor:
  image: myregistry.com/qualys_sensor:xxx
  deploy:
    mode: global  # Deploy 1 container on each node == DaemonSet
  command: ["--swrm-mode"]
```

If you want to deploy the sensor for CI/CD environment provide the **command** value as:

```
command: ["--swrm-mode","--cicd-deployed-sensor"]
```

If you want to deploy a Registry Sensor provide the **command** value as:

```
command: ["--swrm-mode","--registry-sensor"]
```

If you want print logs on the console, provide "`--enable-console-logs`" as an additional value in **command**.

To restrict the cpu usage to a certain value, change the following: (Optional)

Under **deploy** specify the following:

```
mode: global  # Deploy 1 container on each node == DaemonSet
resources:
  limits:
    cpus: '0.20' # Default CPU usage limit(20% of overall CPU
                   available) on each node for sensor.
```

For example,

For limiting the cpu usage to 5%, set deploy:resources:limits:cpus: '0.05', this limits overall cpu usage to 5% of a CPU on a node.

If there are multiple processors on a node, set the deploy:resources:limits:cpus value accordingly.

For example,

You have 5 CPUs on system and want to set 5% of overall capacity of system, set the CPU usage limit to 5 x 0.05 = '0.25'.

To disable any CPU usage limit, set deploy:resources:limits:cpus value to 0.

Under **environment** specify the following:

```
environment:
      ACTIVATIONID: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
      CUSTOMERID: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
      qualys_https_proxy: proxy.qualys.com:3128
```

Activation ID and Customer ID are required. Use the Activation ID and Customer ID from your subscription. You can remove the proxy information if not required.

Under **volumes** ensure that you provide the following information:

```
volumes:
- type: bind
  source: /var/run/
  target: /var/run/
- type: volume
  source: persistent-volume
  target: /usr/local/qualys/qpa/data/
- type: bind
  source: /etc/qualys  # Must exist !
```

```
        target: /usr/local/qualys/qpa/data/conf/agent-data
```

Keep source as "persistent-volume". This ensures that the source directory in volume mapping is set to docker swarm root directory (i.e. /data/docker/volumes).

/etc/qualys directory must exist on all masters and worker nodes for successful volume mapping.

```
volumes:
  persistent-volume:
```

Under **configs** ensure that you provide the following information:

```
configs:
  proxy-cert-path:
    file: /root/cert/proxy-certificate.crt
```

If you are using a proxy, ensure that all masters and worker nodes have a valid certificate file for the sensor to communicate with the Container Management Server.

If you are not using a proxy and you have removed **qualys_https_proxy** from **environment**, you can remove the following parts as well:

```
configs:
  - source: proxy-cert-path
    target: /etc/qualys/qpa/cert/custom-ca.crt

configs:
  proxy-cert-path:
    file: /root/cert/proxy-certificate.crt
```

Once you have modified the **cssensor-swarm-ds.yml** file, run the following command on docker swarm master/leader to create a stack:

```
docker stack deploy -c cssensor-swarm-ds.yml qualys-container-sensor
```

If you need to uninstall Qualys Container Sensor, run the following command on docker swarm master/leader:

```
docker stack rm qualys-container-sensor
```

Deploying sensor in AWS ECS Cluster

Perform the following steps to deploy Qualys Container Sensor as a daemon service in Amazon ECS cluster.

Prerequisites: AWS ECS Cluster should be up and running.

Download the **QualysContainerSensor.tar.xz** file from Qualys Cloud Portal on a Linux computer.

Untar the sensor package:

```
sudo tar -xvf QualysContainerSensor.tar.xz
```

Use the following commands to push the qualys sensor image to a repository common to all nodes in the cluster:

```
sudo docker load -i qualys-sensor.tar
```

```
sudo docker tag <IMAGE NAME/ID> <URL to push image to the repository>
```

For example,

```
sudo docker tag c3fa63a818df 20576712438.dr.ecr.us-east-
1.amazonaws.com/container-sensor:qualys-sensor-xxx
```

```
sudo docker push <URL to push image to the repository>
```

For example,

```
sudo docker push 20576712438.dr.ecr.us-east-1.amazonaws.com/container-
sensor:qualys-sensor-xxx
```

Note: Do not use the examples as it is. You need to replace the registry/image path with your own.

Modify the **cssensor-aws-ecs.json** file (extracted from QualysContainerSensor.tar.xz) to provide values for the following parameters. In order for the json file to work properly, ensure that you do not remove/comment the respective sections mentioned below.

```
 "containerDefinitions": [
    {
       "name": "qualys-container-sensor",
       "image": "20576712438.dr.ecr.us-east-1.amazonaws.com/container-
                sensor:qualys-sensor-xxx",
       "cpu": 10,
       "memory": 512,
       "essential": true,
       "command": [
           "--ecs-mode"
           ],
```

Specify appropriate values for **cpu** (no. of vcpu) and **memory** (size in MB).

If you want to deploy the sensor for CI/CD environment provide the **command** value as:

```
"command": [
     "--ecs-mode",
     "--cicd-deployed-sensor"
     ],
```

If you want to deploy a Registry Sensor provide the **command** value as:

```
"command": [
     "--ecs-mode",
     "--registry-sensor"
     ],
```

If you want print logs on the console, provide "`--enable-console-logs`" as an additional value in **command**.

Under **`environment`** specify the following:

```
"environment": [
       {
         "name": "ACTIVATIONID",
         "value": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX"
       },
       {
         "name": "CUSTOMERID",
         "value": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX"
       },
       {
         "name": "qualys_https_proxy",
         "value": "proxy.qualys.com:3128"
       }
  ]
```

Activation ID and Customer ID are required. Use the Activation ID and Customer ID from your subscription. Specify proxy information, or remove the section if not required. If you remove the proxy section, ensure that json indentation is correct.

If you are not using a proxy and you have removed **`qualys_https_proxy`** from **`environment`**, you can remove the following parts from **`mountPoints`** and **`volumes`**:

```
configs:
  - source: proxy-cert-path
    target: /etc/qualys/qpa/cert/custom-ca.crt

configs:
  proxy-cert-path:
    file: /root/cert/proxy-certificate.crt
```

If proxy section is removed from environment, then remove proxy-cert-path sections under mountPoints and volumes as well:

```
"mountPoints": [
            {
            "sourceVolume": "proxy-cert-path",
            "containerPath": "/etc/qualys/qpa/cert/custom-ca.crt"
            },
          ]
"volumes": [
        {
            "name": "proxy-cert-path",
            "host": {
                "sourcePath": "/root/cert/proxy-certificate.crt"
            }
            }
        ]
```

Under **volumes**, provide information for persistent_volume. If you specify a custom location for persistent_volume, it would get created if not already available on the Docker Host.

Once you are done with the changes, save the **cssensor-aws-ecs.json** file.

Import the json file into Amazon ECS UI to complete the senor deployment

On the Amazon ECS UI, under Task Definitions, click **Create New Task Definition**.

Select the launch type compatibility as EC2 (Fargate is not supported). Provide the Task Definition name, and then provide Task Role, Network Mode, and Task Execution Role if applicable.

Scroll to the bottom of the page and select **Configure via JSON** option. Remove any existing content and then copy-paste the entire contents of the **cssensor-aws-ecs.json** file.

Click **Create** to create the Task Definition. Once created, it should get listed under Task Definitions.

Now go to Clusters, and click the cluster name on which you want to deploy the sensor.

Under Services tab, click **Create**.

Select the launch type as EC2. Select the Task Definition you created above and its revision, and then select a cluster. Provide the Service name, Service type as “DAEMON”, and then configure Network, Load Balancing, and Auto Scaling if applicable.

Review the provided information, and then click **Create** to create the Service. Once created, it should get listed under Services.

Verify that the service status is Active. In the tasks tab, verify that tasks are running on all ECS containers.

Stopping Qualys sensor on Amazon ECS Cluster

If you want to stop the Qualys container sensor from running on all containers, simply delete the service from the Services tab. This will kill the qualys-container-sensor service, but will not remove the sensor from the AWS ECS instances.

Deploying sensor in Mesosphere DC/OS

Perform the following steps to deploy Qualys Container Sensor as an application in DC/OS Marathon.

Prerequisites: A running DC/OS cluster with the DC/OS CLI installed.

Download the **QualysContainerSensor.tar.xz** file from Qualys Cloud Portal on DC/OS master.

Untar the sensor package:

```
sudo tar -xvf QualysContainerSensor.tar.xz
```

Use the following commands to push the qualys sensor image to a repository common to all nodes in the cluster:

```
sudo docker load -i qualys-sensor.tar
```

```
sudo docker tag <IMAGE NAME/ID> <URL to push image to the repository>
```

For example,

```
sudo docker tag c3fa63a818df myregistry.com/qualys_sensor:xxx
```

```
sudo docker push <URL to push image to the repository>
```

For example,

```
sudo docker push myregistry.com/qualys_sensor:xxx
```

Note: Do not use the examples as it is. You need to replace the registry/image path with your own.

Modify the **cssensor-dcos.json** file (extracted from QualysContainerSensor.tar.xz) to provide values for the following parameters. In order for the json file to work properly, ensure that you do not remove/comment the respective sections mentioned below.

```
"id": "/qualys-container-sensor",
"args": ["--dcos-mode"],
"cpus": 1,
"mem": 128,
"disk": 0,
"instances": 1,
"acceptedResourceRoles": ["*"],
```

Specify appropriate values for **cpus** (no. of vcpu), **mem** (size in MiB) and **disk** (size in MiB).

If you want to deploy the sensor for CI/CD environment provide the **args** value as:

```
"args": ["--dcos-mode","--cicd-deployed-sensor"],
```

If you want to deploy a Registry Sensor provide the **args** value as:

```
"args": ["--dcos-mode","--registry-sensor"],
```

If you want print logs on the console, provide "`--enable-console-logs`" as an additional value in **args**.

Ensure that **instances** value is the number of nodes in the cluster. This ensures that the container Sensor runs on each cluster node.

```
"container": {
    "type": "DOCKER",
    "docker": {
      "forcePullImage": true,
      "image": "myregistry.com/qualys_sensor:xxx",
      "parameters": [],
      "privileged": false
    },
```

Under **env** specify the following:

```
"env": {
    "ACTIVATIONID": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
    "CUSTOMERID": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
    "qualys_https_proxy": "proxy.qualys.com:3128"
  },
```

Activation ID and Customer ID are required. Use the Activation ID and Customer ID from your subscription. If you are using a proxy, ensure that all nodes have a valid certificate file for the sensor to communicate with the Container Management Server.

Under **`volumes`** specify the following:

```
"volumes": [
      {
        "containerPath": "/usr/local/qualys/qpa/data",
        "hostPath": "/usr/local/qualys/sensor/data",
        "mode": "RW"
      },
      {
        "containerPath": "/var/run",
        "hostPath": "/var/run",
        "mode": "RW"
      },
      {
        "containerPath": "/usr/local/qualys/qpa/data/conf/agent-data",
        "hostPath": "/etc/qualys",
```

```
        "mode": "RW"
      },
      {
        "containerPath": "/etc/qualys/qpa/cert/custom-ca.crt",
        "hostPath": "/root/cert/proxy-certificate.crt",
        "mode": "RO"
      }
    ]
```

The directories specified for the **hostPath** are automatically created if not already available on the nodes. Ensure to provide a valid **proxy-certificate.crt** file path if you want to deploy the Sensor using a proxy.

If you are not using a proxy and you have removed **qualys_https_proxy** from **env**, you can remove the following from **volumes** as well, while ensuring that json indentation is correct:

```
      {
        "containerPath": "/etc/qualys/qpa/cert/custom-ca.crt",
        "hostPath": "/root/cert/proxy-certificate.crt",
        "mode": "RO"
      }
```

Under **portDefinitions** specify the following:

```
"portDefinitions": [
    {
      "port": 10000,
      "protocol": "tcp"
    }
  ]
```

Specify a valid port number. Replace port number 10000, if already in use.

Once you have modified the **cssensor-dcos.json** file, run the following command on DC/OS master to add the qualys-container-sensor application to Marathon:

```
dcos marathon app add cssensor-dcos.json
```

Use this command to verify that the application is added successfully:

```
dcos marathon app list
```

If you need to uninstall Qualys Container Sensor from Marathon, run the following command on DC/OS master:

```
dcos marathon app remove --force /qualys-container-sensor
```

Deploying sensor in OpenShift

Integrate the Container Sensor into the DaemonSet like other application containers to ensure that there is always a Sensor deployed on the Docker Host.

On the OpenShift master, create a qualysuser serviceaccount, and then allow the qualysuser serviceaccount access to the privileged SCC to avoid issues related to hostNetwork, hostPath volumes, and “access to file denied”.

For example,

```
oc create serviceaccount -n kube-system qualysuser
oc adm policy add-scc-to-user privileged -n kube-system -z qualysuser
```

Where, qualysuser is the user account created in OpenShift for deploying the Qualys Container Sensor.

Perform the following steps for creating a DaemonSet for the Qualys sensor to be deployed in OpenShift.

Note: Ensure that the Container Sensor has read and write access to the persistent storage and the docker daemon socket.

Download the **QualysContainerSensor.tar.xz** file from Qualys Cloud Portal on OpenShift master.

Untar the sensor package:

```
sudo tar -xvf QualysContainerSensor.tar.xz
```

Use the following commands to push the Qualys sensor image to a repository common to all nodes in the OpenShift cluster:

```
sudo docker load -i qualys-sensor.tar

sudo docker tag <IMAGE NAME/ID> <URL to push image to the repository>
```

For example,

```
sudo docker tag c3fa63a818df mycloudregistry.com/container-
sensor:qualys-sensor-xxx


sudo docker push <URL to push image to the repository>
```

For example,

```
sudo docker push mycloudregistry.com/container-sensor:qualys-sensor-xxx
```

Note: Do not use the examples as it is. You need to replace the registry/image path with your own.

Modify the **cssensor-openshift-ds.yml** file (extracted from QualysContainerSensor.tar.xz) to provide values for the following parameters. In order for the yml file to work properly, ensure that you do not remove/comment the respective sections mentioned below.

```
serviceAccountName:
        qualysuser
```

Ensure that the serviceAccountName is provided in the pod declaration.

```
containers:
   - name: qualys-container-sensor
     image: mycloudregistry.com/container-sensor:qualys-sensor-xxx
     securityContext:
       privileged: true
     args: ["--k8s-mode"]
```

If you want to deploy the sensor for CI/CD environment provide the **args** value as:

```
args: ["--k8s-mode","--cicd-deployed-sensor"]
```

If you want to deploy a Registry Sensor provide the **args** value as:

```
args: ["--k8s-mode","--registry-sensor"]
```

If you want print logs on the console, provide "`--enable-console-logs`" as an additional value in **args**.

To restrict the cpu usage to a certain value, change the following: (Optional)

Under **`resources`** specify the following:

```
limits:
  cpu: "0.2" # Default CPU usage limit(20% of overall CPU
               available) on each node for sensor.
```

For example,

For limiting the cpu usage to 5%, set resources:limits:cpu: "0.05", this limits overall cpu usage to 5% of a CPU on a node.

If there are multiple processors on a node, set the resources:limits:cpu value accordingly.

For example,

You have 5 CPUs on system and you want to set 5% of overall capacity of system, set the CPU usage limit to 5 x 0.05 = “0.25”.

To disable any CPU usage limit, set resources:limits:cpu value to 0.

Under **env** specify the following:

Activation ID (Required)

```
- name: ACTIVATIONID
  value: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
```

Customer ID (Required)

```
- name: CUSTOMERID
  value: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
```

Specify proxy information, or remove if not required:

```
- name: qualys_https_proxy
  value: proxy.localnet.com:3128
```

Under **volumes** specify the proxy cert path, or remove if not required:

```
- name: proxy-cert-path
  hostPath:
    path: /root/cert/proxy-certificate.crt
```

Activation ID and Customer ID are required. Use the Activation ID and Customer ID from your subscription.

If you are using a proxy, ensure that all OpenShift nodes have a valid certificate file for the sensor to communicate with the Container Management Server.

If you are not using a proxy and you have removed the above mentioned parts, you can remove the following part from **volumeMounts** as well:

```
- mountPath: /etc/qualys/qpa/cert/custom-ca.crt
  name: proxy-cert-path
```

Once you have modified the **cssensor-openshift-ds.yml** file, run the following command on OpenShift master to create a DaemonSet:

```
oc create -f cssensor-openshift-ds.yml
```

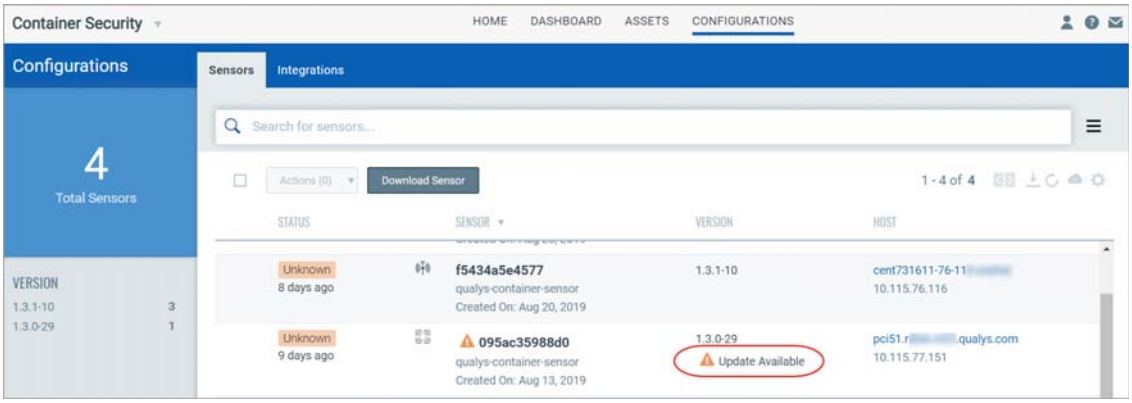
If you need to uninstall Qualys Container Sensor, run the following command on OpenShift master:

```
oc delete ds qualys-container-sensor -n kube-system
```

Administration

Sensor updates

When an update is available you'll see "Update Available" next to the sensor name.



Container sensor update is otherwise automatic, however if you are currently using the beta version of the sensor you need to update to the latest sensor version manually. Automatic update kicks off once you are on a version higher than the beta.

To manually update the sensor from beta to the latest version, download the **QualysContainerSensor.tar.xz** file from Qualys Cloud Portal and then run the following commands directly from the screen on the docker host.

Untar the sensor package:

```
sudo tar -xvf QualysContainerSensor.tar.xz
```

Launch the new sensor:

```
sudo ./installsensor.sh ActivationId=5e7e422a-a1ca-403f-9274-
506622dc5b28 CustomerId=a8cf7043-0245-6f1d-82f8-97f784652b93
Storage=/usr/local/qualys/sensor/data -s
```

Enter **Y** at the prompt asking you to upgrade 'Qualys-Container-Sensor' from version x.x.x to x.x.x.

The install script asks for proxy configuration. If you want to configure proxy, see Proxy Support.

Note: Once you have upgraded from the beta version to a higher version, future updates of Sensor are automatic.

How to uninstall the sensor

The QualysContainerSensor.tar.xz file (which you download for sensor installation from Qualys Cloud Platform) has the script **uninstallsensor.sh** for uninstalling the sensor.

To uninstall a sensor:

If the docker host is configured to communicate over docker.sock, use the following command:

```
./uninstallsensor.sh -s
```

If the docker host is configured to communicate over TCP socket then provide the address on which docker daemon is configured to listen:

```
./uninstallsensor.sh DockerHost=<<IPv4 address or FQDN>:<Port#>> -s
```

For example,

```
./uninstallsensor.sh DockerHost=10.115.27.54:3128 -s
```

Follow the on-screen prompts to uninstall the sensor. Qualys recommends not to clear the persistent storage.

Troubleshooting

Check sensor logs

The sensor log file is located at (by default):

/usr/local/qualys/sensor/data/logs/qpa.log

Diagnostic script

Qualys provides a script to collect diagnostic information about the sensor. You must run the script on the host on which you want to collect the diagnostic information from.

The diagnostic script is present in the QualysContainerSensor.tar.xz that you downloaded for installing the sensor.

The script is called Sensor_Diagnostic_Script.py. You must have Python installed on the host in order to run the script. The script collects the following information from the host and puts it in a tar file called SensorDiagnostic.tar. You can send that file to Qualys Support for further assistance.

The SensorDiagnostic.tar includes 'ScanInfo.json', 'qpa.log' of qualys-container-sensor from given persistent storage, docker logs of qualys-container-sensor, and all information described below in the 'SensorDiagnostic.log' file. If 'ScanInfo.json' and Sensor logs are not available on the Docker host then this script creates empty 'ScanInfo.json' and qpa.log files, and appends "File not found" to them.

- Operating System Information (Type of OS i.e. Linux or Mac and other details)

- Proxy Configuration (Type of proxy set e.g. system, docker, cloud-agent proxy)

- CPU Architecture (Details about model, CPUs, cores, etc)

- RAM Usage (Memory allocation and utilization on host)

- Docker Version (Docker version installed on host)

- Socket Configuration (Docker socket configuration on host e.g. TCP/unix domain)

- Number of docker images (Count of all docker images and their details)

- Number of docker containers (Count of all docker containers and their details)

- CPU and Memory usage of running containers (First result of all resource usage statistics)

Sensor crashes during upgrade

Use installsensor.sh to reinstall Qualys container sensor keeping the "Storage" value as it was for earlier Sensor. This will ensure that the new sensor will not be marked as another Sensor and will simply upgrade the existing one.

For help on install command, see Deploying Container Sensor.

Note: At any given point in time, DO NOT delete the persistent storage. Else, the sensor deployed thereafter will be marked as a new sensor.

What if sensor restarts?

The Sensor is designed to handle restart scenarios and will continue functioning normally after restart. No customer intervention is needed until the sensor crashes.

Note: The Qualys container sensor will fail to restart if it has exited due to a fatal error before the docker host/service restarts.

Duplicate Kubernetes containers

While searching for containers you may see duplicates of containers orchestrated by Kubernetes. This is because Kubernetes spins up a monitoring container for every service container it brings up. Qualys container sensor sees them as two different containers and reports and scans both of the containers.

To see results without duplicate containers add the following string to queries used for searching Kubernetes containers.

```
not label.key:POD
```

For example, use this query to find running containers in Kubernetes:

state:"RUNNING" and not label.key:POD