Preinstallation Requirements

This chapter provides information about the hardware and software that you require to install the CSM server.

Hardware Requirements

The minimum hardware requirements to install CSM server 4.0 are:

- 2 CPUs
- 8-GB RAM
- 30-GB HDD

Note

- For large networks, we recommend that you increase the number of CPUs to run more network installation operations at the same time.
- You can adjust the hard disk space to store images and packages and logs from the operations.

Software Requirements

The software requirements to install CSM server 4.0 are:

- systemd Linux distribution with Docker
- Docker Proxy Configuration (Optional)
- Firewalld (Optional)

systemd

To install the CSM server, you must use systemd. It is a suite that provides the building blocks to create various Linux operating systems. For more details about systemd, refer to Wikipedia.

Ensure that you meet the following requirements before you proceed with installation of CSM server 4.0:

- You need root privileges to install the CSM server because the configuration of CSM server is stored in the /etc/csm.json file. The installation process creates the systemd service for its automatic start. To get root privileges, run the installation script as a root user or as a user with the sudo program access.
- Ensure that you install Docker on the host operating system. For more information, see https://docs.docker.com/install/. Cisco recommends using Ubuntu, CentOS, or Red Hat Enterprise Linux as the host operating system running CSM server 4.0. CSM works with both Docker Community Edition (CE) and Docker Enterprise Edition (EE).

Docker

The CSM server works with both Docker Community Edition (CE) and Docker Enterprise Edition (EE). For more information, refer to official Docker documentation, https://docs.docker.com/install/overview/.

Use Docker 19.03 or later versions to install the CSM server. You can use the following command to check the version of the Docker:

```
$ docker version
Client: Docker Engine - Community
Version:           19.03.9
API version:       1.40
Go version:        go1.13.10
Git commit:        9d988398e7
Built:             Fri May 15 00:25:34 2020
OS/Arch:           linux/amd64
Experimental:      false

Server: Docker Engine - Community
Engine:
  Version:          19.03.9
  API version:      1.40 (minimum version 1.12)
  Go version:       go1.13.10
  Git commit:       9d988398e7
  Built:            Fri May 15 00:24:07 2020
  OS/Arch:          linux/amd64
  Experimental:     false
containerd:
  Version:          1.2.13
  GitCommit:        7ad184331fa3e55e52b890ea95e65ba581ae3429
runc:
  Version:          1.0.0-rc10
  GitCommit:        dc9208a3303feef5b3839f4323d9beb36df0a9dd
docker-init:
  Version:          0.18.0
  GitCommit:        fec3683
```

Docker Proxy Configuration (Optional)

If you install the CSM server behind an HTTPS proxy, for example, in corporate settings, you must configure the Docker systemd service file as follows:

1. Create a systemd drop-in directory for the docker service:

   ```
   $ sudo mkdir -p /etc/systemd/system/docker.service.d
   ```

2. Create a file titled */etc/systemd/system/docker.service.d/https-proxy.conf* that adds the HTTPS_PROXY environment variable. This file allows the Docker daemon to pull the containers from the repository by using the HTTPS Proxy:

```
[Service]
Environment="HTTPS_PROXY=http://proxy.example.com:443/"
```

Note It is common oversight that the HTTPS_PROXY environment variable uses capital letters and the proxy URL starts with http:// and not https://.

3. Reload the configuration changes:

```
$ sudo systemctl daemon-reload
```

4. Restart the Docker:

```
$ sudo systemctl restart docker
```

5. Verify that you have loaded the configuration:

```
$ systemctl show --property=Environment docker
Environment=HTTPS_PROXY=http://proxy.example.com:443/
```

Verify the Docker configuration

To check if you have properly installed the Docker and to ensure that it is up and running, use the following command:

```
$ systemctl is-active docker
active
```

To verify whether you have properly configured the Docker demon, and whether the Docker is able to pull the images from the repository and is able execute the test container; use the following command:

```
$ docker run --rm hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
d1725b59e92d: Pull complete
Digest: sha256:0add3ace90ecb4adbf7777e9aacf18357296e799f81cabc9fde470971e499788
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/
```

```
For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

Firewalld (Optional)

CSM server can run together with Firewalld. Firewalld is provided in the following Linux distributions as the default firewall management tool:

- RHEL 7 and later versions
- CentOS 7 and later versions
- Fedora 18 and later versions
- SUSE 15 and later versions
- OpenSUSE 15 and later versions

Before you run CSM with firewalld, do the following:

1. Run the IP address command and then move the eth0 interface, which is our external interface for CSM, to the “external” zone.

```
$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen
1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
 qlen 1000
    link/ether 08:00:27:f5:d8:3b brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic eth0
       valid_lft 84864sec preferred_lft 84864sec
    inet6 fe80::a00:27ff:fef5:d83b/64 scope link
       valid_lft forever preferred_lft forever
$ sudo firewall-cmd --permanent --zone=external --change-interface=eth0
```

Note By default, the eth0 interface is in a public zone. Moving it to an external zone enables masquerading for external connections to the CSM docker containers.

2. Allow incoming traffic on port 5000 per TCP because port 5000 is the default port of the web interface of the CSM server.

Note On some systems, you must move the “br-csm” interface to the “trusted” zone. The br-csm interface is the internal bridge interface that is created by CSM and is used for communication between CSM containers. This interface may not exist before the CSM installation. However, ensure that you run the following command before the CSM installation process:

```
$ sudo firewall-cmd --permanent --zone=trusted --change-interface=br-csm
```

3. Reload the firewall daemon with new configuration.

```
$ sudo firewall-cmd –reload
```

Note If you have installed the Docker before installing firewalld, restart the docker daemon after making firewalld changes.

Note If you are using any other firewall application apart from firewalld, configure it as required and open port 5000 per TCP for any incoming traffic.