

Installation and Administration

- Introduction, on page 1
- Modules, on page 1
- Installation, on page 2
- Initial Configuration, on page 9
- Updater Configuration, on page 10
- StarOS Configuration, on page 11
- Troubleshooting, on page 11

Introduction

The Content Classification Manager is a customized server running various modules as containers on top of the microservices platform. This server is deployed as a virtual machine (VM) using either OpenStack or VMware ESXi.

Content Classification Manager 21.6 has been qualified on Red Hat OpenStack Release 8 (OpenStack Liberty) and VMware ESXi 6.5.

This release has been qualified for use with StarOS Release 21.6.

Modules

The Content Classification Manager includes the following modules:

- Updater – This container manages external connections with the Talos Security Intelligence (TSI) database update server. It pulls updates from update server and places them in a directory with secure FTP access.
- Secure FTP – This container provides secure FTP functionality for StarOS systems to retrieve TSI database files.
- File Cleanup – This container cleans up old TSI DB files and reclaims disk space. A cron job removes files (oldest first) when the amount of free space falls below 20%, and continues until the free space is no longer below 20%.
- Orchestrator – This container launches and manages the lifecycle of the other containers. It also exports a ConfD CLI interface for configuring various application parameters.
- Consul – This container is an application that provides key-value storage and retrieval for the system.

Installation

Content Classification Manager installation files are provided in two parts:

- a VM image in a qcow2 or vmdk format for OpenStack or VMware deployments respectively
- a product ISO file.

OpenStack

For OpenStack deployments, perform the following steps to install the Content Classification Manager. It is assumed that an OpenStack installation exists with the physical networking configuration for external network connectivity.

Step 1 Ensure there is IPv4 connectivity to the internet and to the StarOS gateways from the Openstack installation. It is highly recommended to separate the traffic between the VM and the backend internet server from the traffic between the VM and the StarOS systems using separate provider networks.

If the StarOS systems are configured to pull updates from the Content Classification Manager at the same time each day, the Content Classification Manager should be configured with a high bandwidth connection (for example a 10 Gbps port) between the VM and the StarOS systems.

Step 2 Create the Project and User under which the Content Classification Manager VM will be launched.

Step 3 Define a flavor, such as TSI, with the following minimum specifications:

- 8 vCPUs
- 32 GB RAM
- 64 GB root disk

Step 4 Download the base qcow2 image and product ISO, and then upload them to Glance. For example:

```
openstack image create --file CC_Manager_21.6.0.Base.release.qcow2 --public
--container-format bare --disk-format qcow2 Base_21.6.0
```

The image name in this example can be anything as long as the same name is specified in the VM launch (nova boot) command.

```
openstack image create --file CC_Manager_21.6.0.release.iso --container-format
bare --disk-format iso tsi.iso
```

The image name in this example can be anything as long as the same name is specified when creating a volume as shown in the subsequent steps.

Step 5 Create a volume containing the product ISO:

```
openstack volume create --image tsi.iso --size 3 tsi-iso
```

The volume size must be large enough to fit the ISO. 3 GB (--size 3) is sufficient since the ISO is approximately 1 GB. The volume name (tsi-iso in the example) can be anything as long as the right volume ID is specified in the VM launch command.

Step 6 Create the cloud init file. Refer to the following example:

```
#cloud-config
debug: True
output: {all: '| tee -a /var/log/cloud-init-output.log'}

users:
  - name: cps
    sudo: ['ALL=(ALL) NOPASSWD:ALL']
    groups: docker
    ssh-authorized-keys:
      - ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDzjJjndIvUiBta4VSIbd2gJmlMWcQ8wtej
gAbiXtoFZdtMdo9G0ZDEOtxHNNDPwWujMiYAkZhZWX/zON9raavU8lgD9+YcRopWUtujIC71YjtoxIjW
IBBbrtqtPlUXMUXQsi91RQbUtslENP+tSatS3awoQupyBMMSutyBady/7Wq0UTwFsnYs5Jfs8jIQuMfV
Q9uJ4mNn7wJ0N+Iaf27rE0t3oiY5DRN6j07WhauM6lCnZ1JDlzqmTnTHQkgJ3uKmQa5x73tJ1OW89Whf
+R+dfslVn/yUwK/vf4extHTn32Dtsxkjz7kQeEDgCe/y7owimaEFcCIfEWEaj/50jegN cps@root-pu
blic-key
chpasswd:
  list: |
    cps:cisco123
  expire: False
write_files:
  - path: /home/cps/.bash_aliases
    encoding: text/plain
    content: |
      alias cli="ssh -p 2024 admin@localhost"
    owner: cps:cps
    permissions: '0644'
  - path: /root/swarm.json
    content: |
      {
        "role": "master",
        "identifier": "master-0",
        "master": "172.16.2.99",
        "network": "172.16.2.0/24",
        "registry": "172.16.2.99:5000",
        "reinitialize_data": "1",
        "zing": "1",
        "tenant": "tsi",
        "weavePw": "cisco123!",
        "scheduler": "aio",
        "deployment_name": "docker-tsi",
        "system_id": "",
        "cluster_id": "",
        "init": "cisco-mitg-tsi/init"
      }
    owner: root:root
    permissions: '0644'
  - path: /etc/update-motd.d/20-cps-text
    content: |
      #!/bin/sh
      product=`jq ".product" /mnt/install/swarm.json  | tr -d '"'`
      identifier=`jq ".identifier" /mnt/install/swarm.json  | tr -d '"'`

      printf "\n"
      printf " * CPS Microservices - ${product}\n"
      printf " * CPS Docker Engine - ${identifier}\n"
      printf "\n"
    owner: root:root
    permissions: '0755'
```

The “chpasswd:” directive specifies that a user named “cps” be created with password “cisco123”. Replace with a secure password. The user is set up with sudo access; this is helpful when collecting various troubleshooting information. The SSH key facilitates password-less login; this will need to be generated anew.

The IP 172.16.2.x IP addresses specified under swarm.json refer to the internal network, so edit as appropriate:

- “master:” value should be the IP address assigned to the VM on the internal network
- “network:” value should be the CIDR of the internal network
- “registry:” value should be the <server IP>:5000

bash_aliases file is optional. It provides a convenience alias to access the CLI once logged into the VM.

Replace weavePw value in swarm.json with a secure password.

Other values should be left as is in the above example.

Step 7 Configure tenant networks:

- An internal network is required. Address on this network is used for internal communication, for e.g. the docker registry is setup on this network when booting up to load container images.
- Two tenant networks are recommended: one for communication with StarOS gateways; one for communication with the Talos Security Intelligence backend server over the internet. One of those can be the internal network itself. For example communication with the internet can be set up by specifying a default gateway on the internal network and associating a floating IP for external communication.

Step 8 Configure a security group, for example ‘tsi’, with the following TCP ports:

Port Number	Purpose	Comments
22	Secure shell access to VM	Administrative purpose only. Restrict CIDR to management network(s).
2024	Secure shell access to CLI interface	For device configuration only. Restrict CIDR to management network(s). Note that the CLI can also be accessed by SSHing to the VM first and then SSHing to localhost port 2024.
2222	Secure FTP downloads to StarOS systems	Restrict CIDR to StarOS management network(s).
5341	Used by StarOS systems to communicate with TSI updater module	Restrict CIDR to StarOS management network(s).

Step 9 Launch the Content Classification Manager VM. For example using the nova boot command:

```
nova boot --config-drive true --user-data=node-master-0.cfg \
  --flavor=tsi \
  --image=Base_21.6.0 \
  --nic net-id="34669234-1f04-44d9-b7ab-695dddcba5fe,v4-fixed-ip=172.16.2.99" \
  --nic net-id="91835357-5c1d-4b1f-ad7e-30cf48a46a30,v4-fixed-ip=172.18.52.37" \
  --block-device id= 8dfcccfd-668c-41aa-8941-ced3a3bded34,source=volume,dest=volume,device=/dev/vdb
 \
  --security-groups tsi \
  docker-tsi-master-0
```

Argument	Description	Comments
--config-drive	Use config drive functionality to intitialize VM	
user-data=<file name>	For config drive use	<file name> should contain the cloud config parameters discussed earlier.
--image=<image name>	Base VM image name	
--nic	Virtual NICs	2 NICs are created; One for an internal network and one for communication with StarOS systems. Internet connectivity is accomplished by associating a floating IP to the instance. The IP addresses are passed in explicitily as opposed to using DHCP. Internal IP address must match value in cloud config file swarm.json.
--block-device	Volume containing CCM product ISO	Some OpenStack releases may not accept or honor the device path /dev/vdb. If so, the device=/dev/xxx argument can be left out.
--security-groups=<name>	Name or the security group created earlier	
--flavor=<name>	Name of the flavor created earlier	
docker-tsi-master-0	Instance name; use any appropriate name	

Once the instance is launched, associate floating IPs if used. This can be done via the Horizon GUI or using neutron and nova CLI commands. Console should show the VM booting up and provide a bash login prompt.

VMware

For VMware deployments, perform the following steps to install the Content Classification Manager. It is assumed that a VMware installation exists with the physical networking configuration for external network connectivity

Step 1 Ensure there is IPv4 connectivity to the internet and to the StarOS gateways from the VMware installation. It is highly recommended to separate the traffic between the VM and the backend internet server from the traffic between the VM and the StarOS systems using separate provider networks.

If the StarOS systems are configured to pull updates from the Content Classification Manager at the same time each day, the Content Classification Manager should be configured with a high bandwidth connection (for example a 10 Gbps port) between the VM and the StarOS systems.

Step 2 Using the vSphere client, create a virtual machine with the following minimum specifications:

- 8 vCPUs
- 32 GB RAM
- 64-bit Ubuntu as Guest OS
- 64 GB or larger disk with the base VM image. The base VM image is available for download as a vmdk file that represents a vmdk file size of approximately 100 GB (stream optimized).

To clone with vmkfstools command line utility:

a) Download the vmdk base image.
b) Use vmkfstools to clone this image to a disk with thin allocation (recommended):

```
vmkfstools --diskformat thin -i \
CC_Manager_21.6.0_Base.release.vmdk \
CC_Manager_21.60_Base.vmdk
```

The -i argument is the base VM vmdk and the last argument is the name of the disk.

When setting up the VM, attach the disk created above to the VM.

Step 3 Attach the Content Classification Manager product ISO as a CD/DVD drive.

Step 4 Set up user data ISO for cloud init. The VM can be initialized with openstack config drive version 2 formatted data. Cloud init expects a specific directory layout. Refer to the following recommended cloud init configuration:

Directory Layout:

```
Top level directory --> config-drive/
                                └── openstack/
                                    ├── content/
                                    |   └── 0000
                                    └── latest/
                                        ├── meta_data.json
                                        └── user_data
```

A forward slash (‘/’) at the end of the line indicates a directory.

Description of the files and example content:

meta_data.json:

```
{
    "files": [
    {
        "content_path": "/content/0000",
        "path": "/etc/network/interfaces"
        }
    ],
    "hostname": "cps-tsi-updater",
    "launch_index": 0,
    "name": "cps-master",
    "meta": {
            "dsmode": "local"
    },
    "uuid": "cps-master"
}
```

hostname, name and uuid can be modified as required.

user_data:

```
#cloud-config
debug: True
output: {all: '| tee -a /var/log/cloud-init-output.log'}

users:
  - name: cps
    sudo: ['ALL=(ALL) NOPASSWD:ALL']
    groups: docker
    ssh-authorized-keys:
      - ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDzjJjndIvUiBta4VSIbd2gJmlMWcQ8wtejgAbiXtoFZdtMdo9G0ZDEOtxHNNDPwWujMiYA
kZhZWX/zON9raavU8lgD9+YcRopWUtujIC71YjtoxIjWIBBbrtqtPlUXMUXQsi91RQbUtslENP+tSatS3awoQupyBMMSutyBady/7Wq0UTwFsnYs5Jfs
8jIQuMfVQ9uJ4mNn7wJ0N+Iaf27rE0t3oiY5DRN6j07WhauM6lCnZ1JDlzqmTnTHQkgJ3uKmQa5x73tJ1OW89Whf+R+dfslVn/yUwK/vf4extHTn32Dt
sxkjz7kQeEDgCe/y7owimaEFcCIfEWEaj/50jegN cps@root-public-key
chpasswd:
  list: |
    cps: cisco123
  expire: False
write_files:
  - path: /home/cps/.bash_aliases
    encoding: text/plain
    content: |
      alias cli="ssh -p 2024 admin@localhost"
    owner: cps:cps
    permissions: '0644'
  - path: /root/swarm.json
    content: |
      {
        "role": "master",
        "identifier": "master-0",
        "master": "172.16.2.99",
        "network": "172.16.2.0/24",
        "registry": "172.16.2.99:5000",
        "reinitialize_data": "1",
        "zing": "1",
        "tenant": "tsi",
        "weavePw": "cisco123",
        "scheduler": "aio",
        "deployment_name": "cps-tsi",
        "init": "cisco-mitg-tsi/init"
      }
    owner: root:root
    permissions: '0644'
  - path: /etc/update-motd.d/20-cps-text
    content: |
      #!/bin/sh
      product=`jq ".product" /root/swarm.json  | tr -d '"'`
      identifier=`jq ".identifier" /root/swarm.json  | tr -d '"'`

      printf "\n"
      printf " * CPS Microservices - ${product}\n"
      printf " * CPS Docker Engine - ${identifier}\n"
      printf "\n"
    owner: root:root
    permissions: '0755'
```

The “chpasswd:” directive specifies that a user named “cps” be created with password “cisco123”. Replace with a secure password. The user is set up with sudo access; this is helpful when collecting various troubleshooting information. The SSH key facilitates password-less login; this will need to be generated anew.

The IP 172.16.2.x IP addresses specified under swarm.json refer to the internal network, so edit as appropriate:

- “master:” value should be the IP address assigned to the VM on the internal network

- “network:” value should be the CIDR of the internal network
- “registry:” value should be the <server IP>:5000

bash_aliases file is optional. It provides a convenience alias to access the CLI once logged into the VM.

Replace weavePw value in swarm.json with a secure password

All other values should be left as is.

0000:

```
auto lo
iface lo inet loopback

auto ens160
iface ens160 inet static
address 172.18.45.99
netmask 255.255.255.0
gateway 172.18.45.1

auto ens192
iface ens192 inet static
address 172.16.2.99
netmask 255.255.255.0

auto ens224
iface ens224 inet static
address 172.18.53.142
netmask 255.255.255.0
```

As meta_data.json indicates, contents of content/0000 populate /etc/network/interfaces file on the VM. The example configures 3 networks, an internal network (ens192), one for communication with StarOS gateways (ens224), and one for internet access (ens160).

Step 5 Once the above configuration files and corresponding directory hierarchy is setup, generate an ISO using the following command:

`mkisofs -o user-data.iso -R -V config-2 config-drive`

This command must be run from the parent directory of config-drive directory. The -V argument value has to be config-2 per config drive naming convention. The ISO (-o argument value) can be named differently if so desired.

Step 6 Specify network interfaces for the VM. It is recommended to use at least 3 separate network interfaces:

- An internal network solely for intra-VM communication. This is used for e.g. a docker registry when initializing the system.
- A network for communication with StarOS instances.
- A network for internet connectivity to the Content Classification Manager VM.

Step 7 Open the following ports. The VM expects to receive traffic to the following TCP ports, so any firewall along the path needs to permit traffic to them:

Port Number	Purpose	Comments
22	Secure shell access to VM	Administrative purpose only.

Port Number	Purpose	Comments
2024	Secure shell access to CLI interface	For device configuration only. Note that the CLI can also be accessed by SSHing to the VM first and then SSHing to localhost port 2024.
2222	Secure FTP downloads to StarOS systems	For StarOS system use only.
5341	Used by StarOS systems to communicate with CCM updater module	For StarOS system use only.

Step 8 Save the settings and power on the virtual machine. The console should display the VM booting and then provide a login prompt.

Initial Configuration

When the VM is first started, it takes approximately 10-15 minutes for all the containers to launch. The command prompt may not be accessible at this time. There are approximately 35 container instances which run on a normal system.

The system provides the following predefined accounts:

- **admin**: Administrative privileges. Default password: **admin**
- **oper**: Operator-level privileges allow application-specific configuration. Default password: **oper**

An operator cannot configure new users or modify existing users, so initial configuration must be done using **admin** account.

Step 1 To access the command line interface (CLI), SSH to port 2024 as user **admin**.

Step 2 Enter configuration mode:

```
config
```

Step 3 Configure the Network Time Protocol server:

```
ntp server <name> address <NTP server IP address>
```

Step 4 Configure the Domain Name Service server:

```
network dns server <DNS server IP address>
```

Step 5 Configure the Secure FTP Group:

```
nacm groups group sftp user <user name>
```

StarOS systems download DB files using Secure FTP. This command configures the name of the group of users allowed to use Secure FTP. The name of this group must be **sftp**. This command requires administrative privilege.

Step 6 Configure the Secure FTP user:

aaa authentication users user <user name>

This command configures the name of the SFTP user. This user name must match what is configured in the secure FTP group configuration. This name must also match what is is configured in StarOS systems which connect to this Content Classification Manager. This name is case-sensitive and must be unique. This command requires administrative privilege.

Secure FTP also requires the following other parameters:

- group id
- user id
- home directory
- SSH directory
- password

These additional parameters are not used in the context of the Content Classification Manager, so configuring the following example values are sufficient. The following example assumes a user name of “beaker”.

```
uid 1717
gid 1717
password $1$FmXy2j6R$tuyTMWNcIx2.Iib86qBq90
ssh_keydir /home/beaker/.ssh
homedir /home/beaker
!
```

Step 7 Configure new passwords for the **admin** and **oper** accounts as follows:

aaa authentication users user admin change-password
aaa authentication users user oper change-password

Change the default passwords to strong, secure passwords. The system prompt for the new password. The user input is not reflected on the screen for security reasons.

Step 8 Commit these changes by exiting the config mode as shown in the following example:

```
admin@hostname(config)# end
Uncomnmitted changes found, commit them? [yes/no/CANCEL] yes
Commit complete
admin@hostname#
```

Updater Configuration

The default settings for Updater are typically sufficient for proper functioning. See the **updater** command in the CLI Commands chapter for more information about the Updater settings.

Any configuration changes to the updater parameter values triggers a restart of the updater (beaker) process. During this restart, existing connections with StarOS systems will be dropped temporarily. These connections automatically reconnect after the restart has finished. This restart typically takes a few seconds. Cisco recommends making updates during a maintenance window or outside the daily update time configured on the StarOS systems.

By default, the updater configuration can be modified by any user. Cisco recommends restricting access by configuring an access control rule (nacm rule-list). At a minimum, the sftp group must be prevented from having configuration change privileges.

The following example denies access to the root path and everything below to users belonging to the group "sftp":

```
nacm rule-list sftp
 group [ sftp ]
 rule sftp
  path /
  action deny
 !
!
```

StarOS Configuration

See the *CF Administration Guide* for instructions to configure a StarOS system to connect to the Content Classification Manager.



Note In this release, the Content Classification Manager module port is fixed at 5341 and the SFTP port is fixed at 2222. While these ports are configurable in the StarOS commands, you must specify port 5341 and 2222 respectively.

The following example show the StarOS commands to configure a connection with the Content Classification Manager. Replace the IP address in the example (1.1.1.1) with the IP address for your Content Classification Manager.

```
configure
  security
    server talos-intelligence my-server
      ip 1.1.1.1 port 5341
      sftp port 2222 username my-username password my-password
    exit
  category server my-server
  end
```

Troubleshooting

Updater Issues

For issues related to the Updater (beaker) process, evaluate or collect the following command outputs.

From the Content Classification Manager VM (you may need to be root user or have sudo access to collect these outputs):

```
docker ps
docker logs orchestrator
docker logs beaker
docker inspect beaker
```

```
docker inspect cleanup
docker logs cleanup
journalctl --no-page –u docker
ss –tnpl
docker exec –it cleanup cat /var/log/cleanup
docker exec –it cleanup ps –ww –ef
```

From the CLI:

```
show running-config
show system
show docker
```

All log files collected in /data/tsi/logs directory.

From the beaker container:

```
supervisorctrl status
ps –ww -ef
cat /etc/beaker/updater.cfg
```

You can log on to the container by running **docker exec –it beaker bash** from the VM. Once you have collected the necessary logs, exit the container using the **exit** command.

-

Secure FTP Issues

In case of issues relating to SFTP, evaluate or collect the following command outputs.

From the Content Classification Manager VM (you may need to be root user or have sudo access to collect these outputs):

```
docker ps
docker logs orchestrator
docker logs sftp
docker inspect sftp
docker inspect cleanup
docker logs cleanup
journalctl --no-page –u docker
ss –tnpl
docker exec –it cleanup cat /var/log/cleanup
docker exec –it cleanup ps –ww –ef
```

From the CLI:

```
show running-config
show system
show docker
```

All log files collected in /data/tsi/logs directory.

From the sftp container:

```
ps –ww -ef
cat /var/log/supervisor/supervisord.log
cat /etc/sshd_config
cat /etc/ssh_config
```

You can log on to the container by running docker exec –it sftp bash from the VM. Once you have collected the necessary logs, exit the container using the exit command.

Startup Issues

In case of issues with containers starting for the first time, evaluate or collect the following logs and command outputs.

Output of the following logs from the VM:

```
/var/log/cloud-init.log
/var/log/cloud-init-output.log
/var/log/syslog
```

Output of the following commands on the VM:

```
systemctl status docker
journalctl --no-page -u docker
docker ps
```

From the CLI:

```
show running-config
show system
show docker
```

All log files collected in /data/tsi/logs directory.

From the VM:

```
docker logs beaker
docker logs sftp
docker logs cleanup
docker logs orchestrator
```