

Content Classification Manager Administration Guide, StarOS Release 21.23

First Published: 2021-03-31

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/c/en/us/about/legal/trademarks.html. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2021 Cisco Systems, Inc. All rights reserved.



CONTENTS

PREFACE

About this Guide vii

Conventions Used vii

Supported Documents and Resources ix

Contacting Customer Support x

CHAPTER 1

Installation and Administration 1

Introduction 1

Modules 1

Installation 2

OpenStack 2

VMware 5

Initial Configuration 9

Updater Configuration 10

StarOS Configuration 11

Troubleshooting 11

Updater Issues 11

Secure FTP Issues 12

Startup Issues 13

CHAPTER 2

CLI Commands 15

CLI Command Overview 17

CLI Command Modes 17

OPERATIONAL Mode 17

CONFIG Mode 18

apply patches 20

database cluster 20

database cluster db-name config-server name 21
database cluster db-name config-server-seed name 22
database cluster db-name router name 23
database cluster db-name shard name 24
database cluster db-name shard shard-name shard-server name 25
database cluster db-name shard shard-name shard-server-seed name 27
db connect admin 28
debug packet-capture gather 28
debug packet-capture purge 29
debug packet-capture start 30
debug tech 30
docker connect 31
docker restart 31
license feature 32
logger set 33
logger clear 34
monitor log application 34
monitor log container 35
network dns server 36
network dns host 36
network virtual-service 37
network virtual-service name host 40
ntp server 40
scheduling external-service 41
scheduling vm-target 43
show alert status 44
show database status 45
show docker engine 47
show docker service 48
show history 49
show license details 50
show log application 50
show log engine 51
show logger level 51

show patches 52
show scheduling effective-scheduler 52
show scheduling status 52
show scheduling vm-target 53
show system diagnostics 54
show system history 55
show system secrets open 56
show system secrets paths 56
show system software available-versions 57
show system software docker-repository 58
show system software version 58
show system software iso stage file 58
show system software iso details 59
show system status debug 60
show system status downgrade 61
show system status running 61
show system status upgrade 61
statistics bulk file 62
statistics bulk interval 63
statistics icmp-ping 64
statistics detail 65
statistics icmp-ping 66
statistics summary 67
system abort-downgrade 68
system abort-upgrade 69
system downgrade 69
system disable-debug 71
system disable-external-services 71
system enable-debug 72
system enable-external-services 72
system secrets add-secret 73
system secrets remove-secret 74
system secrets set-passcode 74
system secrets unseal 75

system software iso stage clean 75
system software iso stage pull 76
system software iso activate 77
system software iso delete 78
system software iso load 79
system start 80
system stop 80
system upgrade 81
updater 81



About this Guide

Note The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

This preface describes the *Content Classification Manager Administration Guide* and its document conventions.

This document provides overview, installation, configuration and troubleshooting information for the Content Classification Manager. This standalone VM provides updates for URL classification and reputation scores for content filtering capabilities to StarOS systems.

- Conventions Used, on page vii
- Supported Documents and Resources, on page ix
- Contacting Customer Support, on page x

Conventions Used

The following tables describe the conventions used throughout this documentation.

Notice Type	Description
Information Note	Provides information about important features or instructions.
Caution	Alerts you of potential damage to a program, device, or system.
Warning	Alerts you of potential personal injury or fatality. May also alert you of potential electrical hazards.

Typeface Conventions	Description
Text represented as a `screen display`	This typeface represents displays that appear on your terminal screen, for example: `Login:`
Text represented as **commands**	This typeface represents commands that you enter, for example: **show ip access-list** This document always gives the full form of a command in lowercase letters. Commands are not case sensitive.
Text represented as a **command** *variable*	This typeface represents a variable that is part of a command, for example: **show card** *slot_number* *slot_number* is a variable representing the desired chassis slot number.
Text represented as menu or sub-menu names	This typeface represents menus and sub-menus that you access within a software application, for example: Click the **File** menu, then click **New**

Command Syntax Conventions	Description
{ keyword or *variable* **}**	Required keyword options and variables are those components that are required to be entered as part of the command syntax. Required keyword options and variables are surrounded by grouped braces { }. For example: **sctp-max-data-chunks { limit** *max_chunks* **| mtu-limit }** If a keyword or variable is not enclosed in braces or brackets, it is mandatory. For example: **snmp trap link-status**
[keyword or *variable* **]**	Optional keywords or variables, or those that a user may or may not choose to use, are surrounded by brackets.

Command Syntax Conventions	Description
|	Some commands support multiple options. These are documented within braces or brackets by separating each option with a vertical bar. These options can be used in conjunction with required or optional keywords or variables. For example: **action activate-flow-detection { intitiation | termination }** or **ip address [count** *number_of_packets* **| size** *number_of_bytes* **]**

Supported Documents and Resources

Related Common Documentation

The following common documents are available:

- *AAA Interface Administration and Reference*
- *Command Line Interface Reference*
- *GTPP Interface Administration and Reference*
- *Installation Guide* (platform dependant)
- *Release Change Reference*
- *SNMP MIB Reference*
- *Statistics and Counters Reference*
- *System Administration Guide* (platform dependant)
- *Thresholding Configuration Guide*

Related Product Documentation

The most up-to-date information for this product is available in the product Release Notes provided with each product release.

The following product documents are also available and work in conjunction with Content Classification Manager:

- *ECS Administration Guide*
- *GGSN Administration Guide*
- *HA Administration Guide*
- *PDSN Administration Guide*
- *P-GW Administration Guide*
- *SaMOG Administration Guide*

Obtaining Documentation

The most current Cisco documentation is available on the following website:

`http://www.cisco.com/cisco/web/psa/default.html`

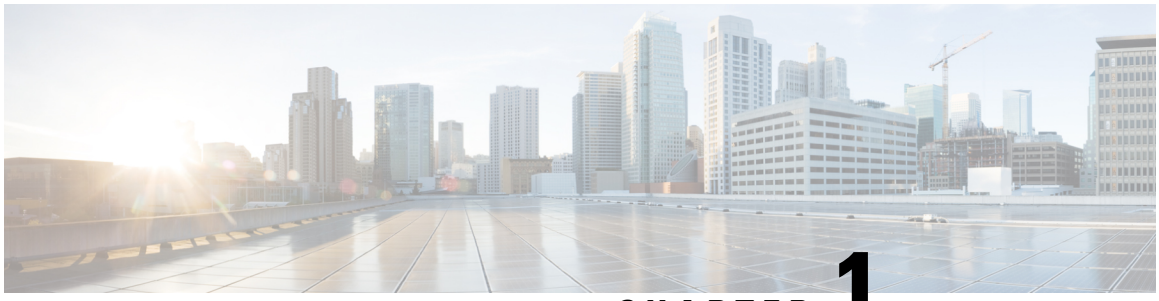
Use the following path selections to access the Content Classification Manager documentation:

Products > Wireless > Mobile Internet > Platforms > ASR 5000 Series > ASR 5000 Series

Contacting Customer Support

Use the information in this section to contact customer support.

Refer to the support area of http://www.cisco.com for up-to-date product documentation or to submit a service request. A valid username and password are required to access this site. Please contact your Cisco sales or service representative for additional information.



CHAPTER 1

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
```

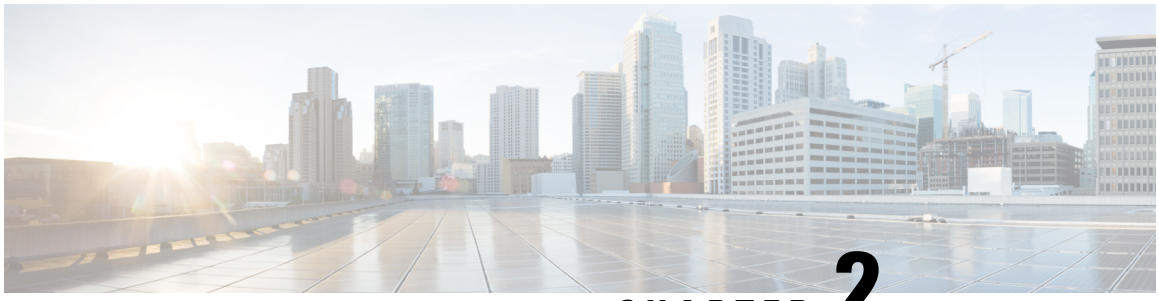
From the CLI:

```
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
```



CHAPTER 2

CLI Commands

- CLI Command Overview, on page 17
- CLI Command Modes, on page 17
- apply patches, on page 20
- database cluster, on page 20
- database cluster *db-name* config-server *name* , on page 21
- database cluster *db-name* config-server-seed *name*, on page 22
- database cluster *db-name* router *name* , on page 23
- database cluster *db-name* shard *name*, on page 24
- database cluster *db-name* shard *shard-name* shard-server *name*, on page 25
- database cluster *db-name* shard *shard-name* shard-server-seed *name*, on page 27
- db connect admin, on page 28
- debug packet-capture gather, on page 28
- debug packet-capture purge, on page 29
- debug packet-capture start, on page 30
- debug tech, on page 30
- docker connect, on page 31
- docker restart, on page 31
- license feature, on page 32
- logger set, on page 33
- logger clear, on page 34
- monitor log application, on page 34
- monitor log container, on page 35
- network dns server, on page 36
- network dns host, on page 36
- network virtual-service, on page 37
- network virtual-service name host, on page 40
- ntp server, on page 40
- scheduling external-service, on page 41
- scheduling vm-target, on page 43
- show alert status, on page 44
- show database status, on page 45
- show docker engine, on page 47
- show docker service, on page 48

- show history, on page 49
- show license details, on page 50
- show log application, on page 50
- show log engine, on page 51
- show logger level, on page 51
- show patches, on page 52
- show scheduling effective-scheduler, on page 52
- show scheduling status, on page 52
- show scheduling vm-target, on page 53
- show system diagnostics, on page 54
- show system history , on page 55
- show system secrets open , on page 56
- show system secrets paths , on page 56
- show system software available-versions , on page 57
- show system software docker-repository , on page 58
- show system software version , on page 58
- show system software iso stage file, on page 58
- show system software iso details, on page 59
- show system status debug, on page 60
- show system status downgrade , on page 61
- show system status running , on page 61
- show system status upgrade , on page 61
- statistics bulk file, on page 62
- statistics bulk interval, on page 63
- statistics icmp-ping, on page 64
- statistics detail, on page 65
- statistics icmp-ping, on page 66
- statistics summary, on page 67
- system abort-downgrade, on page 68
- system abort-upgrade , on page 69
- system downgrade, on page 69
- system disable-debug, on page 71
- system disable-external-services, on page 71
- system enable-debug, on page 72
- system enable-external-services, on page 72
- system secrets add-secret , on page 73
- system secrets remove-secret , on page 74
- system secrets set-passcode , on page 74
- system secrets unseal , on page 75
- system software iso stage clean, on page 75
- system software iso stage pull, on page 76
- system software iso activate, on page 77
- system software iso delete, on page 78
- system software iso load, on page 79
- system start , on page 80
- system stop , on page 80

• system upgrade , on page 81
• updater, on page 81

CLI Command Overview

The command-line interface (CLI) is one of the available user interfaces to configure and monitor the launched application. This user interface provides direct access to execute commands via remote access methods over SSH.

Note The commands described in this chapter are part of the Cisco Policy Suite microservices platform. Not all of these commands apply for Content Classification Manager (CCM) deployments. In general, only the commands described in the Installation and Configuration sections of this document should be needed for a CCM deployment.

CLI Command Modes

The CLI provides two separate command modes – OPERATIONAL and CONFIG.

Each command mode has a separate set of commands available for configuration and monitoring of the application. Entering a "?" at the command problem will indicate the list of available commands for execution within a given mode.

When you start a session, the default mode is OPERATIONAL mode. From this mode, you can access monitoring "show" commands, debugging commands and system maintenance commands. You can enter CONFIG mode to change configuration by issuing the "config" command at the OPERATIONAL prompt.

OPERATIONAL Mode

Logging into the master VM on port 2024 via SSH will allow you to access OPERATIONAL mode. The login into the system will require the use of a username and password. You may attempt to enter a correct password up to three times before the connection attempt is refused.

The commands available at the OPERTIONAL level are separate from the ones available at the CONFIG level. In general, the OPERATIONAL commands encompass monitoring, debugging, and maintenance activity a user will perform.

To list the available OPERATIONAL commands, use the following command:

Table 1: List Commands of OPERATIONAL Mode

Command	Purpose
scheduler# ?	Lists the user OPERATIONAL commands

Example:

```
scheduler# ?
Possible completions:
  aaa                    AAA management
```

```
  apply
  autowizard              Automatically query for mandatory elements
  cd                      Change working directory
  clear                   Clear parameter
  commit                  Confirm a pending commit
  compare                 Compare running configuration to another configuration or a file
  complete-on-space       Enable/disable completion on space
  config                  Manipulate software configuration information
  db                      DB connection and monitoring
  debug                   Debug commands
  describe                Display transparent command information
  devtools                Enable/disable development tools
  display-level           Configure show command display level
  docker                  Docker Management
  exit                    Exit the management session
  file                    Perform file operations
  help                    Provide help information
  history                 Configure history size
  id                      Show user id information
  idle-timeout            Configure idle timeout
  ignore-leading-space    Ignore leading whitespace (true/false)
  job                     Job operations
  logger                  Log level management
  logout                  Logout a user
  monitor                 Application monitoring
  no                      Negate a command or set its defaults
  output-file             Copy output to file or terminal
  paginate                Paginate output from CLI commands
  prompt1                 Set operational mode prompt
  prompt2                 Set configure mode prompt
  pwd                     Display current mode path
  quit                    Exit the management session
  screen-length           Configure screen length
  screen-width            Configure screen width
  script                  Script actions
  send                    Send message to terminal of one or all users
  show                    Show information about the system
  show-defaults           Show default values when showing the configuration
  source                  File to source
  system                  System management
  terminal                Set terminal type
  timestamp               Enable/disable the display of timestamp
  who                     Display currently logged on users
  write                   Write configuration
scheduler#
```

The list of commands will vary based on the version of software installed.

CONFIG Mode

Within OPERATIONAL mode, you can enter CONFIG mode by issuing the “config” command. In general, the CONFIG commands modify the system configuration.

To enter CONFIG mode, use the following command:

Table 2: Enter CONFIG mode

Command	Purpose
scheduler# config	Enter CONFIG mode of the CLI

In CONFIG mode, the prompt changes to include a “(config)” at the end of the prompt.

Example:

```
scheduler# config
Entering configuration mode terminal
scheduler(config)#
```

To list the available CONFIG commands, use the following command:

Table 3: List commands in CONFIG mode

Command	Purpose
scheduler(config)# ?	List the user CONFIG commands

Example:

```
scheduler(config)# ?
Possible completions:
  aaa             AAA management
  alert           Alert status
  alias           Create command alias.
  binding         Binding DB connections
  control-plane   Cross data center control plane
  docker          Docker Management
  license         CPS License Management
  nacm            Access control
  ntp             NTP configuration
  scheduling      Service scheduling
  session         Global default CLI session parameters
  statistics      Application statistics
  system          System configuration
  user            User specific command aliases and default CLI session parameters
  webui           Web UI specific configuration
  ---
  abort           Abort configuration session
  annotate        Add a comment to a statement
  clear           Remove all configuration changes
  commit          Commit current set of changes
  compare         Compare configuration
  copy            Copy a list entry
  describe        Display transparent command information
  do              Run an operational-mode command
  end             Terminate configuration session
  exit            Exit from current mode
  help            Provide help information
  insert          Insert a parameter
  load            Load configuration from an ASCII file
  move            Move a parameter
  no              Negate a command or set its defaults
  pwd             Display current mode path
  rename          Rename an identifier
  resolved        Conflicts have been resolved
  revert          Copy configuration from running
  rollback        Roll back database to last committed version
  save            Save configuration to an ASCII file
  service         Modify use of network based services
  show            Show a parameter
  tag             Manipulate statement tags
  top             Exit to top level and optionally run command
  validate        Validate current configuration
```

apply patches

Applies patches that are staged in the `/data/orchestrator/patches/` directory of the master VM.

This command should only be used by the Cisco TAC and Engineering team to address specific problems and debug the application.

Syntax

```
apply patches
```

Command Parameters

Table 4: Parameter Description

Command Parameter	Description
Service Name or Prefix	The exact name of the service to apply the patch or the prefix of the services to apply.

Command Mode

OPERATIONAL

VNFs

All

Command Usage

This command should only be used at the recommendation of Cisco TAC and Engineering teams.

database cluster

Create a MongoDB database sharded cluster.

Syntax

```
database cluster name sharded-cluster-master {true|false}
no database cluster name
```

Command Parameters

Table 5: Parameter Description

Command Parameter	Description
Name	A short name describing the DB cluster. Each application will use a set of pre-defined names and this name should match one of the application names. For example, DRA uses the name “binding” for storing binding and session records.
sharded-cluster-master	This parameter indicates if the current VNF will execute provisioning operations on the given cluster. If multiple VNF (s) have the same database cluster configuration only one of them should have the “sharded-cluster-master” set to true.

Command Mode

CONFIG

VNFs

All

Command Usage

Use the database cluster command and sub-commands to instruct the application to provision a database cluster for use in application database operations.

Examples

The following is an example of creating a “binding” sharded cluster that is being managed by the current VNF.

```
scheduler(config)# database cluster binding sharded-cluster-master true
```

database cluster *db-name* config-server *name*

Add a MongoDB configuration server process to the named database cluster.

Note This command is applicable only for MongoDB based sharding.

Syntax

```
database cluster db-name config-server name address address
no database cluster db-name config-server name
```

Command Parameters

Table 6: Parameter Description

Command Parameter	Description
DB Name	A short name describing the DB cluster. Each application will use a set of pre-defined names and this name should match one of the application names. For example, DRA uses the name “binding” for storing binding and session records
Name	A short description of the config server name.
address	The IPv4 or IPv6 address of the config server. This parameter does not accept FQDN address format.

Command Mode

CONFIG

VNFs

All

Command Usage

Use the database cluster config-server to add a config-server to the system.

Examples

The following is an example of adding a new config server to the “binding” cluster.

```
scheduler(config)# database cluster binding config-server cfg-1 address 10.10.10.10
```

database cluster *db-name* config-server-seed *name*

Set the initial seed configuration server for boot-strapping the MongoDB replica set initialization process.

Note This command is applicable only for MongoDB based sharding.

Syntax

```
database cluster db-name config-server-seed name
```

Command Parameters

Table 7: Parameter Description

Command Parameter	Description
DB Name	A short name describing the DB cluster. Each application will use a set of pre-defined names and this name should match one of the application names. For example, DRA uses the name “binding” for storing binding and session records
Name	A reference to the configuration server name that will act as the seed for bootstrapping the initial replica set.

Command Mode

CONFIG

VNFs

All

Command Usage

Use the database cluster config-server-seed command to set the initial seed configuration server for boot-strapping the MongoDB replica set initialization process. This is required if a config server is set.

Examples

The following is an example of setting cfg-1 as the initial seed for a new config server to the “binding” cluster.

```
scheduler(config)# database cluster binding config-server-seed cfg-1
```

database cluster *db-name* router *name*

Add a new MongoDB router to the named DB cluster.

Note This command is applicable only for MongoDB based sharding.

Syntax

```
database cluster db-name router name
```

Command Parameters

Table 8: Parameter Description

Command Parameter	Description
DB Name	A short name describing the DB cluster. Each application will use a set of pre-defined names and this name should match one of the application names. For example, DRA uses the name “binding” for storing binding and session records
Name	A short description of the router name.
address	The IPv4 or IPv6 address of the config server. This parameter does not accept FQDN address format
port	The port to bind the router. Generally 27017

Command Mode

CONFIG

VNFs

All

Command Usage

Use the database cluster router command to add a router to named database cluster. Full initialization of database cluster requires at least one router to be defined and often for HA purposes multiple routers are required.

Examples

The following is an example of adding a router to the “binding” cluster.

```
scheduler(config)# database cluster binding router router-1 address 10.10.10.10 port 27017
```

database cluster *db-name* shard *name*

Add a new MongoDB shard to the named database cluster.

Syntax

```
database cluster db-name shard name
no database cluster db-name shard name
```

Command Parameters

Table 9: Parameter Description

Command Parameter	Description
DB Name	A short name describing the DB cluster. Each application will use a set of pre-defined names and this name should match one of the application names. For example, DRA uses the name “binding” for storing binding and session records
Name	A short description of the shard name.

Command Mode

CONFIG

VNFs

All

Command Usage

Use the database cluster shard command to add a new shard to the named database cluster. Full initialization of database cluster requires at least the definition of one shard and often for scaling purposes multiple shards are required.

Examples

The following is an example of adding a shard to the “binding” cluster.

```
database cluster binding shard shard-1
```

database cluster *db-name* shard *shard-name* shard-server *name*

Add a new MongoDB shard to the named DB cluster.

Syntax

```
database cluster db-name shard shard-name shard-server name address address port port
[arbiter {true|false}] [memory_allocation_percent percent] [priority priority] [voter
{true|false}] [storage-engine {IN_MEMORY|MMAPv1|WT}]
```

```
no database cluster db-name shard shard-name server name
```

Note When creating replica set, ensure that all ports are the same, i.e, the replica set should have same port for ARBITER, PRIMARY, and SECONDARY.

Command Parameters

Table 10: Parameter Description

Command Parameter	Description
DB Name	A short name describing the DB cluster. Each application will use a set of pre-defined names and this name should match one of the application names. For example, DRA uses the name “binding” for storing binding and session records
Shard Name	A short description of the shard name.
Name	A short description of the server name.
address	The IPv4 or IPv6 address of the router server. This parameter does not accept FQDN address format.
port	The port to bind the router. Generally -27017
arbiter	Indicates if this node is only an arbiter node.
memory_allocation_percent	Percent (expresses as a positive integer) of the amount of memory to allocate to the DB process for the in-memory storage option.
priority	Relative priority of the node in the shard
voter	Whether this node is a voter.
storage-engine	The storage engine to provision for the process. Valid values are: • IN_MEMORY - pure in memory storage • MMAPv1 – Memory mapped files • WT –wired tirger

Command Mode

CONFIG

VNFs

All

Command Usage

Use the database cluster shard server command to add a new server to named database cluster. Full initialization of database cluster requires at least the definition of one shard server and for HA at least 3 nodes are required.

Examples

The following is an example of adding a new shard to the “binding” cluster.

```
scheduler(config)# database cluster binding shard shard-1 shard-server server-1 storage-engine
 WT address 10.10.10.10 port 27017
```

Note Ports to be used for all database operations must be in the range of 27017 to 27047. Ports outside the defined range are not supported since the application must limit the port mappings. The selected range is sufficient for 30 Mongo processes on a given node.

database cluster *db-name* shard *shard-name* shard-server-seed *name*

Set the initial seed shard server for boot-strapping the MongoDB replica set initialization process.

Syntax

```
database cluster db-name shard shard-name shard-server-seed name
```

Command Parameters

Table 11: Parameter Description

Command Parameter	Description
DB Name	A short name describing the DB cluster. Each application will use a set of pre-defined names and this name should match one of the application names. For example, DRA uses the name “binding” for storing binding and session records
Shard Name	A short description of the shard name.
Name	A reference to the shard server name that will act as the seed for bootstrapping the initial replica set.

Command Mode

CONFIG

VNFs

All

Command Usage

Use the database cluster shard-server-seed command to set the initial seed shard server for boot-strapping the MongoDB replica set initialization process. This is required if a shard is defined.

Note To create or add a member to an existing replica set, you must also run the Mongo console-based commands as shown: `mongo> rs.add("name")`

To remove a replica set or a shard in a sharded cluster case, remove the member from the Mongo console as shown: `mongo> rs.remove("name")`

You must also navigate to the container and the VM on which the member resides and clear the data manually. The data path is the same as the one that is used when the replica-set member is created. Typically, the path is `//mmapv1-tmpfs-2xxxx` where `2xxxx` is the port where the replica set member is started.

Examples

The following is an example of setting server-1 as the initial seed for a new shard called "shard-1" to the "binding" cluster.

```
scheduler(config)# database cluster binding shard shard-1 shard-server-seed server-1
```

db connect admin

Connects to an underlying admin database.

Syntax

No additional arguments.

Command Mode

OPERATIONAL

VNFs

All

Command Usage

Use the db connect admin command to connect to the underlying admin database. Once within this database, the user will have read / write access to the admin database via a mongodb CLI. The capabilities of the mongodb CLI are not described in this document.

debug packet-capture gather

Gathers all running packet captures.

Syntax

`debug packet-capture gather directory` *`directory`*

Command Parameters

Table 12: Parameter Description

Command Parameter	Description
directory	The directory to store the resultant pcap files. This directory is available for downloading via the web file download interface at http://<masterip>:8888/<directory>.

Command Mode

OPERATIONAL

VNFs

All

Command Usage

Use the `debug packet-capture gather` to gather all completed or currently running pcaps. This command is sent to all machines with active tcpdump commands and stops the given commands. After all commands are stopped, the command will gather the resultant pcap files and make them available at http://<masterip>:8888/<directory>.

debug packet-capture purge

Purges all existing pcap files.

Syntax

```
debug packet-capture purge
```

Command Mode

OPERATIONAL

VNFs

All

Command Usage

Use the `debug packet-capture purge` after all relevant packet captures have been downloaded from the application. The system does not automatically purge packet captures. You need to manage the amount of space used by the packet captures using this command.

debug packet-capture start

Starts a packet capture on a given IP address and port.

Syntax

```
debug packet-capture start ip-address ip-address port port timer-seconds timer-seconds
```

Command Parameters

Table 13: Parameter Description

Command Parameter	Description
ip-address	The IP address to start the packet capture. This address can either be IPv4 or IPv6..
port	The port to start the packet capture.
timer-seconds	Duration to run the packet capture - measured in seconds

Command Mode

OPERATIONAL

VNFs

All

Command Usage

Use the debug packet-capture start command to start a tcp-dump on the given IP address and port within the CPS cluster. The packet capture will run for the given timer period and then shutdown automatically. The packet captures can be gathered using the debug packet-capture gather command.

debug tech

Gather logs and debug information to support troubleshooting.

Syntax

```
debug tech
```

Command Parameters

None

Command Mode

OPERATIONAL – Not available via NETCONF/RESTCONF

VNFs

All

Command Usage

Use this command to gather logs and debug information to support troubleshooting.

Examples

```
scheduler# debug tech
```

docker connect

Connects to a docker service and launches a bash shell running on the system.

Syntax

```
docker connect container-id
```

Command Parameters

Table 14: Parameter Description

Command Parameter	Description
container-id	The docker container to open a bash shell. Use the **show docker service** command to find the list of valid container-ids.

Command Mode

OPERATIONAL

VNFs

All

Command Usage

Use the `docker connect` to open a bash shell within a container. This command is primarily used for advanced debugging of the system. Once within a container, you can execute Linux commands and interact with the running container processes.

docker restart

Restarts a docker service that is currently running.

Syntax

```
docker restart container-id container-id
```

Command Parameters

Table 15: Parameter Description

Command Parameter	Description
container-id	The docker container to restart. Use the **show docker service** command to find the list of valid container-ids.

Command Mode

OPERATIONAL

VNFs

All

Command Usage

Use the `docker restart` to restart a running docker service. This command is primarily useful to restore a non-responsive service at the request of Cisco TAC or Cisco Engineering.

license feature

Registers a system license.

Syntax

```
license feature id encrypted-license encrypted-license
no license feature id
```

Command Parameters

Table 16: Parameter Description

Command Parameter	Description
id	ID of the license as provided by Cisco.
encrypted-license	The encrypted license as provided by Cisco.

Command Mode

CONFIG

VNFs

All

Command Usage

Use the `license feature` to add and remove licenses from the running system.

logger set

Sets the various log levels for application logging.

Syntax

`logger set` *`logger-name`* `{ trace | debug | info | warn | error | off }`

Command Parameters

Table 17: Parameter Description

Command Parameter	Description
logger-name	Name of the logger to enable at the given log level.
trace	Enables trace logging and higher.
debug	Enables debug logging and higher.
info	Enables info logging and higher.
warn	Enables warn logging and higher.
error	Enables error logging.
off	Turns off all logging for the logger.

Command Mode

OPERATIONAL

VNFs

All

Command Usage

Use the `logger set` to enable various levels of application logging. The logger names are provided by Cisco per application and are not defined here.

Examples

The following is an example:

```
logger set com.broadhop debug
```

logger clear

Clears a log level defined using the `logger set` command.

Syntax

```
logger clear logger-name
```

Command Parameters

Table 18: Parameter Description

Command Parameter	Description
logger-name	Name of the logger to enable at the given log level.

Command Mode

OPERATIONAL

VNFs

All

Command Usage

Use the `logger clear` to reset the logging level for an application logger to the default level. The current set of logger levels can be found using the `show logger level` command.

monitor log application

Tails the cluster wide application log.

Syntax

```
monitor log application
```

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use the `monitor log application` to tail the `consolidated-qns.log` running on the cc-monitor docker services. If the cc-monitor docker services are not running, this command will fail.

Examples

The following is an example:

```
scheduler# monitor log application
binding-s3.weave.local  2017-03-06 00:07:07,256 [LicenseManagerProxy] INFO
consolidated.sessions - TPS_COUNT:                              SESSION_COUNT:
             LICENSE_COUNT: 100000000
binding-s4.weave.local  2017-03-06 00:07:15,577 [LicenseManagerProxy] INFO
consolidated.sessions - TPS_COUNT:                              SESSION_COUNT:
             LICENSE_COUNT: 100000000
diameter-endpoint-s1.weave.local  2017-03-06 00:07:21,041 [LicenseManagerProxy] INFO
consolidated.sessions - TPS_COUNT:                              SESSION_COUNT:
```

monitor log container

Tails a specific docker container using the `monitor log container` command.

Syntax

`monitor log container` *container-id*

Command Parameters

Table 19: Parameter Description

Command Parameter	Description
container-id	The container's log file to monitor. Use the `show docker service` command to list the valid container-ids.

Command Mode

OPERATIONAL

VNFs

All

Command Usage

Use the `monitor log container` command to tail the docker container log. This will provide the output for all non-application messages for the given container.

Examples

The following is an example:

```
scheduler# monitor log container svn
<<< Started new transaction, based on original revision 94
     * editing path : __tmp_run_stage ... done.

------- Committed revision 94 >>>
```

```
<<< Started new transaction, based on original revision 95
     * editing path : __tmp_run_backup ... done.
```

network dns server

Adds a network DNS server for the cluster to use.

Syntax

```
network dns server address
```

```
no network dns server address
```

Command Parameters

Table 20: Parameter Description

Command Parameter	Description
address	The IP address of the DNS server that the cluster can use. **Note** This address must be available to all servers within the cluster and is generally on an OAM network or the internal network.

Command Mode

CONFIG

VNFs

All

Command Usage

The network DNS server command triggers the addition of a DNS server to the DNS resolution that the application utilizes. These servers are added in the order they appear in the configuration to the DNS resolution.

Examples

The following example adds a DNS server:

```
scheduler(config)# network dns server 10.10.10.10
```

network dns host

Adds a network host to IP address mapping for the cluster to use.

Syntax

```
network dns host host domain address address
no network dns host host domain
```

Command Parameters

Table 21: Parameter Description

Command Parameter	Description
host	The host name of the host mapping to store.
domain	The domain name of the host mapping to store. Use local for hosts that do not have a domain name.
address	The IP address of the host / domain name mapping.

Command Mode

CONFIG

VNFs

All

Command Usage

The network DNS host command triggers the addition of a host / domain mapping to a specific IP address. This is useful when the upstream DNS services do not have a host / domain name mapping or upstream DNS server is not available to the cluster.

Examples

The following example adds a DNS server:

```
scheduler(config)# network dns host test local address 10.10.10.10
```

network virtual-service

Used to configure virtual floating IP address on various interfaces.

Note Virtual Network Service Name must contain minimum 1 character and a maximum length of 8 characters.

Syntax

```
network virtual-service name of floating ip floating-ip floating ip address mask net mask
digits broadcast
broadcast address virtual-router-id virtual router id tracking-service prefix of service
to monitor for IP address
host ip address of host to put the floating ip priority priority of host
```

```
exit
host ip address of host to put the floating ip priority priority of host
commit
end
```

Command Parameters

Table 22: Parameter Description

Command Parameter	Description
name of floating ip	Name of the floating IP address to be configured
floating ip address	The floating IP address to manage with the virtual service.
net mask digits	The network mask (digits) for the floating IP address. Default: 24
broadcast address	The broadcast address of the floating IP.
virtual router id	virtual-router-id is the identity for a virtual router for hosts that are managed for VIP. Value range is from 0 to 255. For more details, refer to VRRP (Virtual Router Redundancy Protocol) RFC 3768 and keepalive documentation.
prefix of service to monitor for IP address	This parameter is a string used to define the service to be monitored.
ip address of host to put the floating ip	IP address of the host where floating IP is hosted.
priority of host	Priority of the host on which the service must run. Priority range is from 1 to 255. Higher the value, higher is the priority.

Command Mode

CONFIG

Command Usage

Use the `network virtual-service` command to configure virtual floating IP address on various interfaces that is managed using keepalive and the VRRP protocol. This command should be used in conjunction with the network virtual-service host command to assign floating IPs to given hosts.

Note To use within OpenStack, you must enable Protocol 112 on the security group – this is the VRRP protocol used by Keepalive.

Examples

The following example creates a floating IP on two hosts:

Note Enter the command manually.

```
scheduler(config)# network virtual-service testvip floating-ip 10.81.70.44 mask 24 broadcast
 10.81.70.255 virtual-router-id 71 tracking-service
diameter-end-point host 10.81.70.47 priority 55
exit
host 10.81.70.46 priority 56
commit
end
```

You can check the status of configuration on the scheduler by running the following command:

```
show running-config network
```

Sample Output:

```
network virtual-service testvip
 virtual-router-id 71
 floating-ip       10.81.70.44
 mask              24
 broadcast         10.81.70.255
 host 10.81.70.46
  priority 100
 !
 host 10.81.70.47
  priority 101
 !
!
```

Requirement

As a part of OpenStack configuration to have allowed-address-pairs configured on the VMs that are going to host the VIP.

Here is an example for ESC:

Under **vm_group** > **interfaces** > **interface**, you need to add the following configuration:

```
<allowed_address_pairs>
  <address>
    <ip_address>10.81.70.44</ip_address>
    <netmask>255.255.255.0</netmask>
  </address>
</allowed_address_pairs>
```

Note The above mentioned configuration needs to be done on all the interfaces of all the VMs where we are looking for virtual IP.

network virtual-service name host

Adds a new virtual-service floating IP address to the system.

Syntax

```
network virtual-service name host address priority priority
no network virtual-service name host address
```

Command Parameters

Table 23: Parameter Description

Command Parameter	Description
name	The logical name of the virtual service floating IP. Virtual Network Service Name must contain a minimum of 1 character and a maximum length of 8 characters.
address	The IP of the host that should manage this floating IP.
priority	The priority of the host relative other hosts within the group. Default: 100

Command Mode

CONFIG

VNFs

All

Command Usage

Use this command to add new hosts to a virtual service. The hosts added will be start a Keepalive process to manage the floating IP via the VRRP process.

Examples

The following example adds a floating IP on a host:

```
scheduler(config)# network virtual-service test host 10.84.100.136 priority 100
```

ntp server

Creates an NTP server for the system to synchronize system clocks.

Syntax

```
ntp server name address address
```

Command Parameters

Table 24: Parameter Description

Command Parameter	Description
name	Name of the server.
address	IP address or FQDN of the NTP server.

Command Mode

CONFIG

VNFs

All

Command Usage

Use the `ntp server` command to synchronize the clocks of each virtual machine within the cluster. When this command is used, each node will run an NTP service. The NTP service is either a client or relay as described below:

- A relay node is a node that can reach at least one of the NTP servers defined in the configuration. The relay nodes are configured to point to the ntp servers defined in the server.
- A client node is an internal node that cannot reach an NTP server. The client nodes are configured to point to the relay nodes.

Examples

The following is an example:

```
scheduler(config)# ntp server server1 address 10.10.10.10
```

scheduling external-service

Creates a docker service that is external to the installed application.

Syntax

```
scheduling external-service name image image cap-add cap-add environment environment
host-network { true | false } port-mapping port-mapping run-level run-level scalable { true
 | false } scheduling-slot scheduling-slot volume volume
```

Command Parameters

Table 25: Parameter Description

Command Parameter	Description
name	Name of the service
image	Fully qualified image name.
scalable (optional)	Scale multiple instances across hosts. Default is false.
run-level (optional)	Relative run level between external services. Default is 0.
host-network (optional)	Bind to the host network. Default is to the overlay network.
volume (optional)	Volume mounts in the format is as follows: <host path>:<docker path>. Additional mounts are separated by ",".
port-mapping (optional)	Port mapping of the format is as follows: <external>:<internal>. Additional mounts are separated by ",".
cap-add (optional)	Linux capabilities to add to the container. Additional mounts are separated by ",".
scheduling-slot (optional)	Scheduling slot to start the container (for all containers). Use the **show running-config docker engine** command to view list of scheduling slots.
environment (optional)	Environment variables to export into the container in the format given below: <KEY>=<VALUE> Additional mounts are separated by ",".

Command Mode

CONFIG

VNFs

All

Command Usage

The `scheduling external-service` instructs the scheduling application to run the defined docker image on the given scheduling slots based on the configuration defined. Once scheduled the external-service appears in the `show scheduling status` and the `show docker service` commands.

scheduling vm-target

Calculates a vm-target for an external scaling system.

Syntax

`scheduling vm-target` *`name`* `group-size` *`group-size`* `k` *`k`* `max` *`max`* `min` *`min`* `override` *`override`* `query` *`query`* `scale-up-threshold` *`scale-up-threshold`*

`no scheduling vm-target name`

Command Parameters

Table 26: Parameter Description

Command Parameter	Description
name	Name or identifier for the vm-target rule.
group-size (optional)	Size of the scaling group. Default is one
k (optional)	K value in an n + k redundancy model. Default is one.
max (optional)	Maximum value to calculate for the vm-target.
min (optional)	Minimum value to calculate for the vm-target.
override (optional)	Override value for the vm-target. This overrides anything the equation would calculate.
query	Query to calculate a raw scaling value.
scale-up-threshold	Divisor when calculating the scaling number. The query's raw value is divided by the scale-up-threshold to get a the value of n in an n+k redundancy model.

Command Mode

CONFIG

VNFs

All

Command Usage

The `scheduling vm-target` instructs the system to calculate VM scaling targets which can be used by the system to add and remove scaling VMs as required. The following algorithm is used to calculate the VM target for a given “name”:

vm-target(name) = roundup ((query value) / (scale-up-threshold))*group-size+K

show alert status

Displays the status of all alerts in the system. It displays either all alert statuses or alerts for a specific named alert.

Syntax

`show alert status` *`rule-name`*

Command Parameters

Table 27: Parameter Description

Command Parameter	Description
rule-name (optional)	Displays alert statuses for a given rule-name.

Command Mode

OPERATIONAL

VNFs

All

Examples

The following is an example:

```
scheduler# show scheduling status
                           OUT
                RUN        OF
MODULE INSTANCE LEVEL STATE DATE
-------------------------------------------------------
consul 1        50   RUNNING false
admin-db 1      75 RUNNING false
memcached-vip 1 100 RUNNING false
prometheus 1    100 RUNNING false
prometheus 2    100 RUNNING false
prometheus 3    100 RUNNING false
```

Table 28: Parameter Description

Parameter	Description
Name	Rule-name of the alert.

Parameter	Description
Event Host	Host where the alert was generated.
Status	Status of the alert. Valid values are: • firing • resolved
Message	Current alert message.
Update Time	Timestamp of the first alert message that transitioned to the given status.

show database status

Displays the currently configured database clusters members.

Syntax

```
show database status
```

Command Parameters

Table 29: Parameter Description

Command Parameter	Description
Address	The address of the database process.
Port	The port the database service is running.
Name	Name of the database process.

Command Parameter	Description
Status	The current status of the mongo process. Valid states are: • CONNECTED – The mongo router is connected to the config servers • NOT_CONNECTED – The mongo router is not connected to the config servers • NO_CONNECTION – The process is not up or is not monitored • STARTUP – The DB node is in the STARTUP mode • PRIMARY – The DB node is the current PRIMARY • SECONDARY – The DB node is a SECONDARY node • RECOVERING – The DB node is currently RECOVERING from a restart or other failure • STARTUP2 – The DB node is in STARTUP2 mode • UNKNOWN – The DB node is in an UNKNOWN state • ARBITER – The DB node is currently an active ARBITER • NOT_INITIALIZED – The DB node is not initialized and pending initialization
Type	The type of the mongo process. Valid values are: • replica_set – a member of the replica set • config_server – a member of the config server replica set • mongos – a mongo router process
Cluster Name	The name of the cluster that owns the process.
Shard	The name of the associated shard.
Replica Set	The name of the replica set associated to the process.

Command Mode

OPERATIONAL

VNFs

All

Examples

The following is an example:

```
scheduler# show database status
                                                                CLUSTER
ADDRESS       PORT   NAME    STATUS     TYPE          NAME  SHARD   REPLICA SET
-------------------------------------------------------------------------------------
192.168.65.2  27018  shardA  PRIMARY    replica_set   test  shardA  rs-shardA
192.168.65.2  27019  -       PRIMARY    config_server test  cfg     test-configsrv
192.168.65.2  27017  -       CONNECTED  mongos        test  router-1 test-configsrv
```

show docker engine

Displays the status of the clusters docker engines.

Syntax

```
show docker engine
```

Command Mode

OPERATIONAL

VNFs

All

Examples

The following is an example:

```
scheduler# show docker engine
                                    MISSED
ID                        STATUS    PINGS
-------------------------------------------
binding-73d3dc            CONNECTED 0
binding-8a8d17            CONNECTED 0
binding-c74547            CONNECTED 0
binding-dabba5            CONNECTED 0
control-0                 CONNECTED 0
control-1                 CONNECTED 0
control-2                 CONNECTED 0
diameter-endpoint-0       CONNECTED 0
diameter-endpoint-1       CONNECTED 0
diameter-endpoint-2       CONNECTED 0
diameter-endpoint-3       CONNECTED 0
master-0                  CONNECTED 0
session-shard-1-e079cf    CONNECTED 0
session-shard-2-80941f    CONNECTED 0
```

Table 30: Parameter Description

Parameter	Description
ID	The identifier within the cluster of the docker engine. Generally, this maps to the hostname where the engine resides.
Status	Indicates if the scheduling application is connected to the docker engine running on a host.
Missed Pings	The number of consecutive missed pings for a given host.

show docker service

Displays the currently running docker services.

Syntax

```
show docker service
```

Command Mode

OPERATIONAL

VNFs

All

Examples

The following is an example:

```
scheduler# show docker service
MODULE   INSTANCE  NAME           VERSION              ENGINE         CONTAINER ID
STATE   MESSAGE PENALTY BOX
-------------------------------------------------------------------------------------------------------
admin-db    1     mongo-admin-a      3.4.0.0          control-0       mongo-admin-a
HEALTHY  false     -
admin-db    1     mongo-admin-arb    3.4.0.0          master-0        mongo-admin-arb
HEALTHY  false     -
admin-db    1     mongo-admin-b      3.4.0.0          control-1       mongo-admin-b
HEALTHY  false     -
admin-db    1     mongo-admin-setup  12.9.9-2017      master-0        mongo-admin-setup
HEALTHY  false     -
                                -03-03.123.797af71
binding     1     binding            12.9.9-dra.2017  binding-73d3dc  binding-s1
HEALTHY  false     -
                                -03-03.115.0f485ef
binding     1     session-router     3.4.0.0          binding-73d3dc  session-router-s1
HEALTHY  false     -
binding     2     binding            12.9.9-dra.2017  binding-8a8d17  binding-s2
HEALTHY  false
                                -03-03.115.0f485ef
```

Table 31: Parameter Description

Parameter	Description
Module	Scheduling module that is executing the docker service.
Instance	For scalable modules, the instance number that the service relates.
Name	Logical name of the service.
Version	Version of the image executing.
Engine	Engine identifier that is executing the docker service.
Container ID	Container id of the docker service.
State	Current state of the docker service.
Penalty Box	Indicates if the service is waiting to be rescheduled if an error occurred.
Message	Message related to the penalty box designation.

show history

Displays the history of commands executed on the system.

Syntax

```
show history
```

Command Mode

OPERATIONAL

VNFs

All

Examples

The following is an example:

```
scheduler# show history
03-04 16:56:03 -- show docker service | include diameter
03-04 16:56:22 -- show docker service | include diameter | include diameter-endpoint-0
03-04 16:57:31 -- docker connect docker-host-info-s8
03-04 16:59:19 -- docker connect socket-forwarder-s1
03-04 17:01:02 -- ifconfig
03-04 17:01:22 -- docker connect socket-forwarder-s1
03-04 17:01:54 -- docker connect diameter-endpoint-s2
03-04 17:03:32 -- docker connect diameter-endpoint-s2
03-04 17:05:25 -- docker connect diameter-endpoint-s1
```

show license details

Displays the current license details installed on the system.

Syntax

```
show license details
```

Command Mode

OPERATIONAL

VNFs

All

Examples

The following is an example:

```
scheduler# show license details
ID       DEFAULT  COUNT      EXPIRATION
-----------------------------------------------------------
SP_CORE  true     100000000  2017-06-02T02:04:07+00:00
```

Table 32: Parameter Description

Parameter	Description
ID	ID of the license entry.
Default	Indicates if this is the default 90 day license installed on system install.
Count	Count for the given license.
Expiration	Expiration timestamp for the license.

show log application

Displays the application log in a viewer that enables you to scroll and search.

Syntax

```
show log application
```

Command Mode

OPERATIONAL

VNFs

DRA

show log engine

Displays the engine log in a viewer that enables you to scroll and search.

Syntax

```
show log engine
```

Command Mode

OPERATIONAL

VNFs

DRA

show logger level

Displays the current logger levels in the system that overrides the default logging.

Syntax

```
show logger level
```

Command Mode

OPERATIONAL

VNFs

All

Examples

The following is an example:

```
scheduler# show logger level
Logger    Current Level
--------  ---------------
dra       warn
```

Table 33: Parameter Description

Parameter	Description
Logger	The logger that is overridden.
Current Level	The current level of logging.

show patches

Lists the patches that are in `/data/orchestrator/patches` directory.

Syntax

```
show patches
```

Command Mode

OPERATIONAL

VNFs

All

Command Usage

The `show patches` indicates the patch that is loaded in the given patch directory and not a patch that is applied to the system .

show scheduling effective-scheduler

Displays the effective scheduler running in the system.

Valid results are HA and AIO.

Syntax

```
show scheduling effective-scheduler
```

Command Mode

OPERATIONAL

VNFs

All

Examples

The following is an example:

```
scheduler# show scheduling effective-scheduler
scheduling effective-scheduler HA
```

show scheduling status

Displays the currently loaded modules.

Syntax

```
show scheduling status
```

Command Mode

OPERATIONAL

VNFs

All

Examples

The following is an example:

```
scheduler# show scheduling status
                                                  OUT
                                  RUN             OF
MODULE                INSTANCE    LEVEL   STATE   DATE
-------------------------------------------------------
consul                1           50      RUNNING false
admin-db              1           75      RUNNING false
memcached-vip         1           100     RUNNING false
prometheus            1           100     RUNNING false
prometheus            2           100     RUNNING false
prometheus            3           100     RUNNING false
```

Table 34: Parameter Description

Parameter	Description
Module	Module name that is running.
Instance	The instance number scheduled for scalable modules.
Run Level	The relative run level of the module compared to other modules. In an upgrade, the system reschedules from highest run level to lowest run level and in a downgrade the system schedules from low to high.
State	The current state of the module. Valid states are: • RUNNING • SCHEDULING • STOPPING
Out of Date	Indicates whether the software is out of date with the running system.

show scheduling vm-target

Displays the results of the scheduling vm-target calculation.

Syntax

```
show scheduling vm-target
```

Command Mode

OPERATIONAL

VNFs

All

Parameter Description

Parameter	Description
group	The vm-target group name that the count applies.
Count	The calculated count of VMs for scaling.

show system diagnostics

Shows the current diagnostics.

Syntax

There are no arguments for this command.

Command Mode

OPERATIONAL

VNFs

All

Command Parameters

Table 35: Parameter Description

Command Parameter	Description
Node ID	ID of the node where the diagnostics was run.
Check	The ID of the check that was run.
IDX	For Checks that return multiple results the corresponding index number
Status	Indicates if the check is passing or not.
Message	The corresponding message for the diagnostic.

Examples

```
scheduler# show system diagnostics | tab
NODE        CHECK ID                  IDX  STATUS   MESSAGE

-------------------------------------------------------------------------------------------------------------------------------

binding-s1  serfHealth                1    passing  Agent alive and reachable

binding-s1  service:cisco-policy-api  1    passing  TCP connect localhost:8080: Success

binding-s1  service:cisco-policy-app  1    passing  CLEARED: Session creation is allowed

binding-s1  service:cisco-policy-app  2    passing  CLEARED: -Dcom.broadhop.developer.mode
 is disabled
```

show system history

Shows the history of system events.

Syntax

There are no arguments for this command.

Command Mode

OPERATIONAL

VNFs

All

Command Parameters

Table 36: Parameter Description

Command Parameter	Description
IDX	The index of the event in the system history log.
Event Time	Timestamp of the event in the system history log.
Module	The internal module that generated the history log entry.
Message	The message associated with the log entry.

Examples

```
scheduler# show system history
IDX  EVENT TIME                    MODULE         MESSAGE

-------------------------------------------------------------------------------------------------------------------------------
```

```
1    2017-02-04T02:04:02.469+00:00  system         System started
2   2017-02-04T02:04:29.021+00:00  docker-engine  Adding docker engine session-shard-2-80941f
3    2017-02-04T02:04:29.096+00:00  docker-engine  Adding docker engine diameter-endpoint-3
4    2017-02-04T02:04:29.187+00:00  docker-engine  Adding docker engine diameter-endpoint-2
5    2017-02-04T02:04:29.303+00:00  docker-engine  Adding docker engine binding-c74547
6    2017-02-04T02:04:29.375+00:00  docker-engine  Adding docker engine control-2
7   2017-02-04T02:04:29.503+00:00  docker-engine  Adding docker engine session-shard-1-e079cf
8    2017-02-04T02:04:29.583+00:00  docker-engine  Adding docker engine control-1
9    2017-02-04T02:04:29.671+00:00  docker-engine  Adding docker engine control-0
10   2017-02-04T02:04:29.751+00:00  docker-engine  Adding docker engine binding-dabba5
11   2017-02-04T02:04:29.843+00:00  docker-engine  Adding docker engine binding-73d3dc
12   2017-02-04T02:04:29.981+00:00  docker-engine  Adding docker engine binding-8a8d17
```

show system secrets open

Shows if the system secrets are unsealed.

This command returns true if the secrets are unsealed and false if they are still sealed. To open the system secrets, see system secrets unseal , on page 75.

Syntax

There are no arguments for this command.

Command Mode

OPERATIONAL

VNFs

All

Examples

```
scheduler# show system secrets open
system secrets open true
```

show system secrets paths

Shows the current set secrets.

This command does not show the value of the secrets only the path and if the value is readable by the system.

Syntax

There are no arguments for this command.

Command Mode

OPERATIONAL

VNFs

All

Command Parameters

Table 37: Parameter Description

Command Parameter	Description
Path	The identifying path of the secret.
Status	Indicates if the path can be read by the system.

Examples

```
scheduler# show system secrets paths
PATH  STATUS
--------------
test  valid
```

show system software available-versions

Shows the list of available software versions to upgrade or downgrade a system.

Syntax

There are no arguments for this command.

Command Mode

OPERATIONAL

VNFs

All

Examples

```
scheduler# show system software available-versions
VERSION
-----------------------------------
12.9.9-dra.2017-03-03.115.0f485ef
```

show system software docker-repository

Shows the currently configured docker-repository.

Syntax

There are no arguments for this command.

Command Mode

OPERATIONAL

VNFs

All

Examples

```
scheduler# show system software docker-repository
system software docker-repository registry:5000
```

show system software version

Shows the currently installed software version.

Syntax

There are no arguments for this command.

Command Mode

OPERATIONAL

VNFs

All

Examples

```
scheduler# show system software version
system software version 12.9.9-dra.2017-03-03.115.0f485ef
```

show system software iso stage file

Displays the currently staged files in the /data/isos/staged-isos folder.

Syntax

```
show system software iso stage file
```

Command Parameters

None

Command Mode

OPERATIONAL

VNFs

All

Examples

The following example also shows a sample output:

```
scheduler# show system software iso stage file
NAME                  CREATED                    SIZE MB  MD5 SUM

-------------------------------------------------------------------------------------------
cisco-policy-dra.iso  2017-05-17T12:35:58+00:00  1100.04  c636794475b76e84041901b0ca3dcac4
```

Where:

- Name: The filename of the iso.
- Created: The date the file was created on the file system.
- Size MB: The size of the file in megabytes.
- MD5 Sum: The MD5 sum of the file.

show system software iso details

Displays the currently active ISOs that are loaded on the system.

Syntax

```
show system software iso details
```

Command Parameters

None

Command Mode

OPERATIONAL

VNFs

All

Examples

The following example also shows a sample output:

```
CATEGORY  NAME     VERSION  QUALIFIER    CREATED  ACTIVE  MB
-------------------------------------------------------------------------
product cisco-policy-dra 12.9.9   dra.2017-05-   2017-05  true   1102.9
                                    17.441.69    -17T13:
                                    68d89        4:15.708
                                                 +00:00
```

Where:

- Category: The type of ISO. Either product or extras. Extras can be used to load external docker images for use by external services.
- Name: The product name of the ISO
- Version: The version of the ISO
- Qualifier: The qualifier of the ISO
- Created Date: The creation date of the ISO on the file system
- Active: Indicates if the registry is currently pointing to the ISO to download images.
- Size: The size of the ISO on the file system.

show system status debug

Shows if the system is currently configured with debug tools.

Syntax

```
show system status debug
```

Command Parameters

None

Command Mode

OPERATIONAL

VNFs

All

Examples

The following example also shows a sample output:

```
scheduler# show system status debug
system status debug false
```

Where:

- Debug: Indicates if the system is configured to deploy containers with debug tools

show system status downgrade

Shows if the system is currently downgrading the installed software.

Syntax

There are no arguments for this command.

Command Mode

OPERATIONAL

VNFs

All

Examples

```
scheduler# show system status downgrade
system status downgrade false
```

show system status running

Shows if the system is currently running.

Syntax

There are no arguments for this command.

Command Mode

OPERATIONAL

VNFs

All

Examples

```
scheduler# show system status running
system status running true
```

show system status upgrade

Shows if the system is currently upgrading an installed software.

Syntax

There are no arguments for this command.

Command Mode

OPERATIONAL

VNFs

All

Examples

```
scheduler# show system status upgrade
system status upgrade false
```

statistics bulk file

Defines a new bulk statistics file that the system generates on a regular basis.

Syntax

```
statistics bulk file name header
 header query query format
format no bulk file name
```

Command Parameters

Table 38: Parameter Description

Command Parameter	Description
name	The base name of the bulk statistics file to create. The final file name generated has the following format:<name>-<timestamp in seconds>.csv
header	The exact text of the header to put at the start of all new files.
query	The Prometheus query to execute to build the bulk statistics. The query format is described in the Prometheus documentation:https://prometheus.io/docs/querying/basics/

Command Parameter	Description
format	The format of the output line. Each time series returned from the query that is executed will pass through the formatting string. Substitution variables appear as ${variable}. The following pre-defined variables exist in addition to the ones returned from Prometheus: • current-value – last value returned • max-value – max value over last 5 minutes • avg-value – average value over last 5 minutes • min-value – minimum value over last 5 minutes • timestamp – timestamp of when the sample was taken in the following format: yyyy-MM-dd'T'HH:mm:ss'Z'

Command Mode

CONFIG

VNFs

All

Command Usage

Use the bulk file command to define a bulk statistics file that supplements the default bulk statistics files created by the system. The format and queries are user defined.

Examples

The following example creates a bulk file on peer message rates:

```
statistics bulk file peer_tps
 query  "peer_message_total{remote_peer!=\"\"}"
 format ${app_id},${direction},${instance},${local_peer},
${remote_peer},${type},${current-value}
!
```

statistics bulk interval

Modifies the timer that the system uses to generate the bulk statistics that are defined via the bulk file command.

Syntax

```
statistics bulk interval interval no bulk interval
```

Command Parameters

Table 39: Parameter Description

Command Parameter	Description
interval	Timer length (in seconds) used to trigger a new bulk statistics file.

Command Mode

CONFIG

VNFs

All

Command Usage

Use the bulk interval command to control the timer length in triggering a new bulk statistics file.

Notes:

1. The generation of bulk statistics runs +/- 10 seconds of the interval.
2. The generation of bulk statistics is not synchronized to the minute.
3. The default interval, if not defined, is 300 seconds.

Examples

The following example creates a bulk file every 10 minutes:

```
scheduler(config)# bulk interval 600
```

statistics icmp-ping

Creates a probe that tests whether a host is up using ICMP ping.

Syntax

`statistics icmp-ping` *`address`* `no statistics icmp-ping` *`address`*

Command Parameters

Table 40: Parameter Description

Command Parameter	Description
address	The address to ping via ICMP. The resultant statistics are stored in the following metric: • probe_success • probe_duration_seconds • probe_ip_protocol

Command Mode

CONFIG

VNFs

All

Command Usage

Use the statistic icmp-ping command to instruct the monitoring system to ping the given address using the ICMP protocol. The IP address must be reachable via the master, control-a, and control-b hosts.

Examples

The following example creates an ICMP ping test:

```
scheduler(config)# statistics icmp-ping 10.10.10.10
```

statistics detail

Adds a statistics detail for the system to capture.

Syntax

```
statistics detail query category name query query format format scale scale
```

Command Parameters

Table 41: Parameter Description

Command Parameter	Description
category	Category of the statistic.
name	Name of the statistic.

Command Parameter	Description
query	Prometheus query to execute in order to retrieve the statistics.
format (optional)	Formatting rule for the statistic. The labels from the Prometheus query are substituted using the ${label} format.
scale (optional)	Scaling factor to take the raw value and scale to by the scale factor. A negative value divides by the scale factor and a positive value multiples by the scale factor.

Command Mode

CONFIG

VNFs

All

Command Usage

The statistics detail command triggers the application to monitor a given statistic and record it in memory and for reporting using the show statistics detail command. The values are refreshed every 10 seconds.

Examples

```
statistics detail query diameter success-message-tps
 query  "sum(rate(diameter_endpoint_request_total{result_code=\"2001\"}[10s])) by
(app_id,message_type)"
 format "${app_id} ${message_type}"
!
```

statistics icmp-ping

Creates a probe that tests whether a host is up using ICMP ping.

Syntax

```
statistics icmp-ping address no statistics icmp-ping address
```

Command Parameters

Table 42: Parameter Description

Command Parameter	Description
address	The address to ping via ICMP. The resultant statistics are stored in the following metric: • probe_success • probe_duration_seconds • probe_ip_protocol

Command Mode

CONFIG

VNFs

All

Command Usage

Use the statistic icmp-ping command to instruct the monitoring system to ping the given address using the ICMP protocol. The IP address must be reachable via the master, control-a, and control-b hosts.

Examples

The following example creates an ICMP ping test:

```
scheduler(config)# statistics icmp-ping 10.10.10.10
```

statistics summary

Adds a statistics summary for the system to capture.

Syntax

```
statistics summary query category name query query scale scale
```

Command Parameters

Table 43: Parameter Description

Command Parameter	Description
category	Category of the statistic.
name	Name of the statistic.

Command Parameter	Description
query	Prometheus query to execute in order to retrieve the statistics.
scale (optional)	Scaling factor to take the raw value and scale to by the scale factor. A negative value divides by the scale factor and a positive value multiples by the scale factor.

Command Mode

CONFIG

VNFs

All

Command Usage

The statistics summary command triggers the application to monitor a given statistic and record it in memory and for reporting using the show statistics summary command. The values are refreshed every 10 seconds.

The summary command does not support "group by" operations to show multiple lines from a single query.

Examples

```
statistics summary query diameter tps
 query "sum(rate(diameter_endpoint_request_total{result_code=\"2001\"}[10s]))"
!
```

system abort-downgrade

Stops a downgrade that is in progress.

Syntax

There are no arguments for this command.

Command Mode

OPERATIONAL

VNFs

All

Command Usage

The system abort-downgrade command stops the current rolling downgrade of the system. This command is only available when the system is in the process of downgrading and is not available after the downgrade is complete. Once this command is issued, system upgrade , on page 81 command should be issued to revert this software to the previous version.

system abort-upgrade

Stops an upgrade that is in progress.

Syntax

There are no arguments for this command.

Command Mode

OPERATIONAL

VNFs

All

Usage Guidelines

The system abort-upgrade command stops the current rolling upgrade of the system. This command is only available when the system is in the process of upgrading is not available after the upgrade is complete. Once the command is issued, system downgrade, on page 69 command should be issued to revert this software to the previous version.

system downgrade

Downgrades the system to a older software version.

Syntax

```
system downgrade version version [consul-downgrade [true/false] [snapshot_name]]
```

Command Mode

OPERATIONAL

VNFs

All

Command Parameters

Table 44: Parameter Description

Command Parameter	Description
version	The new software version to install into the system.
consul-downgrade	consul-downgrade is an optional parameter. For more information , see consul-downgrade, on page 70.

consul-downgrade

During upgrade, it takes a snapshot of existing consul data which are yet to be upgraded and saves as <*version-name*> (to which you are upgrading) and upgrade proceeds normally. Post upgrade all consul servers/agents will be upgraded to newer version.

For example, if you are upgrading from 19.5.0.release to 20.x.0.release, snapshot name is *20.x.0.release*.

If the value is set as true, following operations are carried out:

- Check if you have provided snapshot-name. If you have not provided the snapshot name, by default, it takes current version as snapshot. You can also provide the snapshot name. To list all the available snapshots, use `consul list-snapshots` command.
- If snapshot is present, then consul is restored with the snapshot and further downgrade proceeds normally.
- If snapshot is not present, then downgrade does not get started and an error is displayed.
- If you have provided the snapshot-name, then snapshot (if exists) in `/data/orchestrator/config/snapshot/` is verified and consul is restored with the given snapshot and downgrade continues.
- In case of no snapshot, an error is displayed.

Note Post rollback, consul data is of state before upgrade if consul downgrade is selected during ISO rollback. Hence, if changes are made to the consul config post upgrade, they are lost and you need to reapply the config changes.

Caution You cannot restore newer version snapshot on an old consul server.

Example:

1. When upgrading to any new version (for example, from v1 to v2), it takes consul snapshot as `/data/orchestrator/config/snapshot-consul/v2`.
2. During downgrade (for example, from v2 to v1), provide snapshot name in system-downgrade command.
3. When upgrading to v3 from v2 (for example, consul version for v1 is 1.0.0, consul version for v2 is 1.5.3 and consul version for v3 is 1.5.3). Upgrade from v1 to v2, snapshot is store as v2; from v2 to v3, snapshot is stored as v3.
4. If you want to downgrade directly from v3 to v1 and you do not provide the snapshot name, by default, it takes the snapshot of v3 and consul version is 1.5.3. The downgrade fails. You must provide the snapshot name in system-downgrade command as v2.

Command Usage

The system downgrade command installs new software on the system using a rolling downgrade approach to minimize service interruption. Care must be taken to ensure that the system downgrade command is used when moving from a higher software version to a lower version of the software. The rolling downgrade upgrades the software modules in startup order. After the command is issued, the CLI disconnects while the

CLI software is restarted. The CLI generally becomes available within 30 seconds. Once the CLI becomes available, the status of the upgrade can be monitored using the show scheduling status, on page 52 command.

Examples

```
system downgrade version 12.9.9-dra.2017-03-03.115.0f485ef
```

system disable-debug

Disables debug tools in deployed containers.

Syntax

```
system disable-debug
```

Command Parameters

None

Command Mode

OPERATIONAL

VNFs

All

Command Usage

Use the system disable-debug command to turn off debugging tools on newly launched containers.

Examples

The following example disables debug tools:

```
scheduler# system disable-debug
```

system disable-external-services

Disables external services that are currently running in the system.

Syntax

```
system disable-external-services
```

Command Parameters

None

Command Mode

OPERATIONAL

VNFs

All

Command Usage

Use the system disable-external-services to stop all services registered with the scheduling external-service command.

Examples

The following example disables external services:

```
scheduler# system disable-external-services
```

system enable-debug

Enables debug tools in deployed containers.

Syntax

```
system enable-debug
```

Command Parameters

None

Command Mode

OPERATIONAL

VNFs

All

Command Usage

Use the system enable-debug command to turn on debugging tools on newly launched containers.

Examples

The following example enables debug tools:

```
scheduler# system enable-debug
```

system enable-external-services

Enable external registered services.

Syntax

```
system enable-external-services
```

Command Parameters

None

Command Mode

OPERATIONAL

VNFs

All

Command Usage

Use the system enable-external-services command to enable external services that are currently registered with the scheduling external-service command.

Examples

The following example enables external services:

```
scheduler# system enable-external-services
```

system secrets add-secret

Adds a secret to the system.

Syntax

```
system add-secret path path secret secret
```

Command Mode

OPERATIONAL

VNFs

All

Command Parameters

Table 45: Parameter Description

Command Parameter	Description
Path	The identifying path of the secret to add.
Secret	The clear text value of the secret to add.

Command Usage

The system add-secret command adds a secret to the system. This command is available only if the secrets are open. See show system secrets open , on page 56.

system secrets remove-secret

Removes a secret from the system.

Syntax

```
system remove-secret path path
```

Command Mode

OPERATIONAL

VNFs

All

Command Parameters

Table 46: Parameter Description

Command Parameter	Description
Path	The identifying path of the secret to remove.

Command Usage

The system remove-secret command removes a secret from the system. This command is available only if the secrets are open. See show system secrets open , on page 56.

system secrets set-passcode

Overwrites the current passcode that is used to encrypt or decrypt the master key for the secrets.

Syntax

```
system secrets set-passcode passcode
```

Command Mode

OPERATIONAL

VNFs

All

Command Parameters

Table 47: Parameter Description

Command Parameter	Description
Passcode	The new passcode to seal the secrets.

Command Usage

The system secrets command is used to change the passcode to unlock the secrets stored within the operational database. All secrets are encrypted using a randomly generated master-key that is encrypted/decrypted by the end-user provided passcode. If the passcode is lost, then the secrets currently stored are not recoverable. This command is available only if the secrets are open. See show system secrets open , on page 56.

system secrets unseal

Unseals the secrets if a non-default passcode is used to seal the secrets.

Syntax

```
system secrets unseal passcode passcode
```

Command Mode

OPERATIONAL

VNFs

All

Command Parameters

Table 48: Parameter Description

Command Parameter	Description
Passcode	The passcode to unseal the secrets.

Command Usage

The system secrets unseal command is used to unlock any stored secrets so that they can be shared with services that require a clear text secret or password. An example of this is a database connection password.

system software iso stage clean

Remove all downloaded ISOs from the stage directory.

Syntax

```
system software iso stage clean
```

Command Parameters

None

Command Mode

OPERATIONAL

VNFs

All

Command Usage

The system software iso stage clean command removes all files that have been staged in the hosts /data/isos/staged-isos/ directory. This command should be run after an ISO file has been uploaded via the system software iso load command.

Examples

```
scheduler# system software iso stage clean
```

system software iso stage pull

Downloads a software ISO to the stage directory on the host.

Syntax

```
system software iso stage pull URL
```

Command Parameters

Table 49: Parameter Description

Command Parameter	Description
URL	The URL to download into the hosts /data/isos/staged-isos/ directory. If the URL ends with the zsync suffix, then the zsync command is invoked to retrieve the file.

Command Mode

OPERATIONAL - Not available via NETCONF/RESTCONF

VNFs

All

Command Usage

Invocation of the command downloads the given URL to the /data/isos/staged-isos/ directory. After invocation of this command, invocation of the show system software iso stage file command shows details of the downloaded file and the system software iso load command loads the file into the system.

Examples

The following example also shows a sample output:

```
scheduler# system software iso stage pull
http://171.70.34.121/microservices/latest/cisco-policy-dra.iso
--2017-05-17 15:08:39--  http://171.70.34.121/microservices
/latest/cisco-policy-dra.iso
Connecting to 171.70.34.121:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1153468416 (1.1G) [application/octet-stream]
Saving to: 'cisco-policy-dra.iso'

cisco-policy-dra.iso                              4%[=====>

                                ]  45.85M  4.07MB/s    eta 4m 27s
```

system software iso activate

Activate an existing ISO.

Syntax

```
system software iso activate category [product|extras] name name version version qualifier
 qualifier
```

Command Parameters

Table 50: Parameter Description

Command Parameter	Description
Category	The category to load the ISO. Either product or extras can be selected. The extras category represents a docker registry that contains external (non-product) docker images.
Name	The product name of the ISO to activate.
Version	The version of the ISO to activate
Qualifier	The qualifier of the ISO to activate

Command Mode

OPERATIONAL

VNFs

All

Command Usage

The system software iso activate command triggers the system to restart the local docker registry to point to the given ISO. This command should be run before upgrading or downgrading the software.

Examples

The following example loads and activates a product ISO:

```
scheduler# system software iso activate category product name cisco-policy-dra version
12.9.9 qualifier dra.2017-05-17.441.6968d89
```

system software iso delete

Deletes an existing ISO.

Syntax

```
system software iso delete category [product|extras] name name version version qualifier
qualifier
```

Command Parameters

Table 51: Parameter Description

Command Parameter	Description
Category	The category to load the ISO. Either product or extras can be selected. The extras category represents a docker registry that contains external (non-product) docker images.
Name	The product name of the ISO to delete.
Version	The version of the ISO to delete
Qualifier	The qualifier of the ISO to delete

Command Mode

OPERATIONAL

VNFs

All

Command Usage

The system software iso delete command triggers the system to remove the ISO. This command can only be run on non-active ISOs.

Examples

The following example deletes an ISO:

```
scheduler# system software iso delete category product name cisco-policy-dra version 12.9.9
 qualifier dra.2017-05-17.441.6968d89
```

system software iso load

Load a new ISO into the system.

Syntax

```
system software iso load category [product|extras] file filename activate [true|false]
```

Command Parameters

Table 52: Parameter Description

Command Parameter	Description
Category	The category to load the ISO. Either product or extras can be selected. The extras category represents a docker registry that contains external (non-product) docker images.
Filename	The filename of the ISO to load.
Activate	Indicates whether the system should switch the internal docker registry to point to the new ISO.

Command Mode

OPERATIONAL

VNFs

All

Command Usage

The system software iso load command triggers unpacking of the staged ISO into a permanent location on the host. This command is executed before a system upgrade command can be executed.

Examples

The following example loads and activates an ISO:

```
scheduler# system software iso load category product file cisco-policy-dra.iso activate
true
```

system start

Starts all the services on a system that has been currently stopped.

Syntax

There are no arguments for this command.

Command Mode

OPERATIONAL

VNFs

All

Usage Guidelines The system start command performs a controlled startup of the system by starting all the services in a rolling fashion taking into account various service dependencies.

system stop

Stops all the services on the system (excluding the CLI, NETCONF, and RESTCONF service).

Syntax

There are no arguments for this command.

Command Mode

OPERATIONAL

VNFs

All

Command Usage

The system stop commands performs a controlled shutdown of the system by stopping all the services in the reverse order of start-up.



Note For ephemeral databases (such as session), all data is lost on a system stop command.

system upgrade

Upgrades the system to a new software version.

Syntax

```
system upgrade version version
```

Command Mode

OPERATIONAL

VNFs

All

Command Parameters

Table 53: Parameter Description

Command Parameter	Description
Version	The new software version to install into the system.

Command Usage

The system upgrade command installs new software on the system using a rolling upgrade approach to minimize service interruption. Care must be taken to ensure that upgrade command is used when moving from a lower software version to a higher version of the software. The rolling upgrade upgrades the software modules in reverse start-up order. After the command is issued, the CLI disconnects while the CLI software is restarted. The CLI generally become available within 30 seconds. Once the CLI becomes available, the status of the upgrade can be monitored using the show scheduling status, on page 52 command.

Examples

```
system upgrade version 12.9.9-dra.2017-03-03.115.0f485ef
```

updater

Content Classification Manager (CCM) module configuration.

Syntax

```
[ no ] updater [ log level { debug | error | info | notice | warning } | server udi { true
 | false } | server uri uri_string ]
```

Command Parameters

Table 54: Parameter Description

Command Parameter	Description
no	Returns the specific setting to the default value.
log level	Configure the logging level for the CCM updater module. The levels increase verbosity of the logs in this order: error < warning < notice < info < debug The "no" option returns the command to the system to the default setting of "notice".
server udi { true | false }	Sets the UDI (Unique Device Identifier) based authentication when communicating with the CCM backend server. UDI based authentication is the only authentication method currently supported. Do not set this value to false. Doing so will result in failure to communicate with the CCM backend server. The “no” option returns the command to the default setting of "true".
server uri <string>	Defines the URI of the CCM backend server. The “no” form restores the default setting of "https://update-manifests.ironport.com".

Command Mode

CONFIG

VNFs

All

Command Usage

Use this command to configure the CCM updater module.

The default settings are typically sufficient for proper functioning.

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.

Examples

The following example changes the log level for this module to "debug":

```
host# updater log level debug
```