CLI Commands

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.

In addition to the CLI, Cisco CPS provides a NETCONF and RESTCONF interface for API access to the application.

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
```

alert rule

Creates a new alerting rule.

The alerting rule allows automatic creation of internal and SNMP traps based on system conditions. The Prometheus monitoring application must be running for alerts to trigger properly. If all Prometheus servers are down, then the system does not generate alerts.

Syntax

```
alert rule name duration duration event-host-label event-host-label
expression expression message message snmp-clear-message snmp-clear-message
snmp-facility { application | hardware | networking | os | proc | virtualization }
snmp-severity { alert | critical | debug | emergency | error | info | none | notice | warning
 }
```

Command Parameters

Table 4: Parameter Description

Command Parameter	Description
name	The name of the alert rule.
duration	The duration measured the condition must exist before triggering an alarm. The format of the duration is <value><unit>. The value is any positive integer and the unit is one of the following: • s – second • m – minute • h – hour

Command Parameter	Description
event-host-label (optional)	The label received by the alerting engine from the Prometheus monitoring application. The application generates one alert per unique value of the given label. The valid labels are determined by the query executed and can be found by executing the query without the comparison operators in the Grafana application on a sample dashboard. If not defined, then the alert is considered global.
expression	The expression that makes up the alerting rule. The expression is built using a Prometheus expressions (https://prometheus.io/docs/querying/basics/) and must conform to the rules defined in the Prometheus alerting documentation: https://prometheus.io/docs/alerting/rules/
message	A configurable message to be sent with the alert. This message supports substitution of labels as defined in the templating section of the Prometheus documentation: https://prometheus.io/docs/alerting/rules/ . The resultant alert message is sent in any associated SNMP traps when the alert is triggered.
snmp-clear-message (optional)	A configurable message that is sent as the clear message when the alert condition is no longer valid.
snmp-facility (optional)	The target snmp-facility to use when generating SNMP trap: • application • hardware • networking • os • proc • virtualization Default is application.

Command Parameter	Description
snmp-severity	The target snmp-severity to use when generating an SNMP trap: • alert • critical • debug • emergency • error • info • none • notice • warning Default is alert.

Command Mode

CONFIG

VNFs

All

Command Usage

Use the alert rule command to define monitoring rules for the system. When you create a new alert rule, the alert rule is exported to the Prometheus monitoring servers, which are monitoring the system on a 1-second interval. The Prometheus servers monitor the underlying expression defined in the alert rule and send alerts scheduling OAM node when they are triggered or when they are cleared. The OAM node tracks internally the status of all alerts and sends any SNMP traps if SNMP servers are defined.

Examples

The following example generates an alert when node_lode > 3:

```
alert rule test
 expression         "node_load5 > 3"
 event-host-label   instance
 message            "Node level exceeds 3"
 snmp-facility      application
 snmp-clear-message "Node level below 3"
!
```

alert snmp-v2-destination

Creates a new SNMPv2 destination.

Creation of a SNMPv2 destination causes the system to forward any triggered/cleared alerts to the SNMPv2 destination.

Syntax

```
alert snmp-v2-destination nms-address community community
```

Command Parameters

Table 5: Parameter Description

Command Parameter	Description
nms-address	The address to send SNMPv2 traps.
Community	The community to use for SNMPv2 traps

Command Mode

CONFIG

VNFs

All

Command Usage

Use the alert snmp-v2-destination to forward alerts from the system to an external SNMPv2 trap receiver. The traps are sent using the following algorithm:

- Sent once when the alert is cleared
- Sent once when the alert is firing
- Sent once if the OAM application is restarted and the alert is firing.

Examples

The following example sends all alerts to community “test” with address 10.10.10.10.

```
scheduler(config)# alert snmp-v2-destination 10.10.10.10 community test
```

alert snmp-v3-destination

Creates a new SNMPv3 destination.

Creation of a SNMPv3 destination causes the system to forward any triggered/cleared alerts to the SNMPv3 destination.

Syntax

```
alert snmp-v3-destination nms-address auth-password
auth-password auth-proto auth-proto engine-id
```

```
engine-id privacy-password privacy-password
user user
```

Command Parameters

Table 6: Parameter Description

Command Parameter	Description
nms-address	The address to send SNMPv3 traps.
auth-password	Authentication passphrase used for authenticated SNMPv3 messages.
auth-proto	Authentication protocol used for authenticated SNMPv3 messages. Valid values are MD5 and SHA
engine-id	Context engine id as a hexadecimal string.
privacy-password	Privacy passphrase used for encrypted SNMPv3 messages.
privacy-protocol	Privacy protocol used for encrypted SNMPv3 messages. Valid values are DES and AES.
user	Security name used for authenticated SNMPv3 messages.

Command Mode

CONFIG

VNFs

All

Command Usage

Use the alert snmp-v3-destination to forward alerts from the system to an external SNMPv2 trap receiver. The traps are sent using the following algorithm:

- Sent once when the alert is cleared
- Sent once when the alert is firing
- Sent once if the OAM application is restarted and the alert is firing.

Examples

The following example sends all alerts to community “test” with address 10.10.10.10.

```
scheduler(config)# alert snmp-v3-destination 10.10.10.10 user
test auth-proto SHA auth-password test engine-id 0x01020304 privacy-protocol
 AES privacy-password  test
```

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 7: 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.

binding db-connection

Adds additional binding db connections from the DRA to a DRA binding database.

Syntax

```
binding db-connection { ipv4 | ipv6 | imsiapn | msisdnapn| slf }
 address port
```

Command Parameters

Table 8: Parameter Description

Command Parameter	Description
ipv4	Connection definition for the IPv4 binding database.
ipv6	Connection definition for the IPv6 binding database.

Command Parameter	Description
imsiapn	Connection definition for the IMSI-APN binding database.
msisdnapn	Connection definition for the MSISDN-APN binding database.
slf	Connection definition for the SLF database.
address	Address of the binding DRA database. This is either an IP address or an FQDN.
port	Port of the binding DRA database.

Command Mode

CONFIG

VNFs

DRA

Command Usage

Use the binding db-connection command to instruct the application on how to connect to the remote binding database. In general, there should be configuration lines entered per binding database type in order to support high availability.

Examples

The following configuration defines two redundant connections per database.

```
binding db-connection ipv6 172.16.82.195 27017
!
binding db-connection ipv6 172.16.82.196 27017
!
binding db-connection ipv4 172.16.82.195 27017
!
binding db-connection ipv4 172.16.82.196 27017
!
binding db-connection imsiapn 172.16.82.195 27017
!
binding db-connection imsiapn 172.16.82.196 27017
!
binding db-connection msisdnapn 172.16.82.195 27017
!
binding db-connection msisdnapn 172.16.82.196 27017
!
binding db-connection slf 172.16.82.195 27017
!
binding db-connection slf 172.16.82.196 27017
!
```

binding db-connection-settings

Used to configure the mongo connection settings.

Syntax

```
binding db-connection-settings { drasession | imsiapn | ipv4 | ipv6 | msisdnapn | range |
slf }
 connect-timeout connections-per-host max-wait-time socket-timeout

no binding db-connection-settings <database>
```

Command Parameters

Table 9: Parameter Description

Command Parameter	Description
drasession	Connection definition for the DRA session database.
imsiapn	Connection definition for the IMSI-APN binding database.
ipv4	Connection definition for the IPv4 binding database.
ipv6	Connection definition for the IPv6 binding database.
msisdnapn	Connection definition for the MSISDN-APN binding database.
range	Port range to be used.
slf	Connection definition for the SLF database.
connect-timeout	Connection timeout in milliseconds. It is used only when establishing a new connection. Default: 500
connections-per-host	Maximum number of connections allowed per host for this MongoClient instance. Those connections are kept in a pool when idle. Once the pool is exhausted, any operation requiring a connection blocks waiting for an available connection. Default: 10
max-wait-time	Maximum wait time in milliseconds that a thread may wait for a connection to become available. Default: 500
socket-timeout	Socket timeout in milliseconds. It is used for I/O socket read and write operations. Default: 1000

Command Mode

CONFIG

VNFs

DRA

Command Usage

Use the binding db-connection-settings command to configure the mongo connection settings.

Examples

The following is an example:

```
admin@orchestrator(config)# binding db-connection-settings ?
Possible completions:
  drasession  imsiapn  ipv4  ipv6  msisdnapn  range  slf

admin@orchestrator(config)# binding db-connection-settings drasession ?
Possible completions:
  connect-timeout  connections-per-host  max-wait-time  socket-timeout  <cr>

admin@orchestrator(config-db-connection-settings- drasession)# connect-timeout ?
Possible completions:
  <int>[500]

admin@orchestrator(config-db-connection-settings- drasession)# connections-per-host ?
Possible completions:
  <int>[10]

admin@orchestrator(config-db-connection-settings- drasession)# max-wait-time ?
Possible completions:
  <int>[500]

admin@orchestrator(config-db-connection-settings- drasession )# socket-timeout ?
Possible completions:
  <int>[1000]
```

control-plane relay

Adds additional control-plane entries between two disconnected CPS vDRA sites.

Syntax

```
control-plane relay name address
address port port
```

Command Parameters

Table 10: Parameter Description

Command Parameter	Description
Name	A short name describing the connection.
address	An IP address or FQDN of the connection. IPv6 address must be enclosed in square brackets.

Command Parameter	Description
port (optional)	The destination port of the connection. Defaults to 6379 if not defined.

Command Mode

CONFIG

VNFs

DRA

Command Usage

Use the control-plane relay command to instruct the application how which links it should use to relay CPS vDRA control traffic. CPS vDRA control traffic is the traffic that describes the current endpoints within a site and the relay IPs for site to site communication. For a 2 site model there should be at least 4 entries defined in this definition (two for each site). For a 3 site model there should be at least 6 entries in this definition.

Examples

The following configuration adds a relay connection to siteA over address 10.10.10.10 port 6379.

```
scheduler(config)# control-plane relay siteA-1 address 10.10.10.10
port 6379
```

database cluster

Create a MongoDB database sharded cluster.

Syntax

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

Command Parameters

Table 11: 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.

Command Parameter	Description
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.

Syntax

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

Command Parameters

Table 12: 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.

Command Parameter	Description
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.

Syntax

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

Command Parameters

Table 13: 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.

Syntax

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

Command Parameters

Table 14: 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 DB cluster.

Syntax

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

Command Parameters

Table 15: 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 16: 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

Command Parameter	Description
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 17: 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
```

database cluster *<db name>* ipv6-zone-sharding true/false

Enable the zone-based sharding for IPv6. When zone-based sharding is enabled on IPv6 database, hash-based sharding can still be configured on other databases.

Syntax

```
database cluster <db name>
ipv6-zone-sharding true/false
```

Command Parameters

Table 18: Parameter Description

Command Parameter	Description
DB Name	A short name describing the DB cluster. Each application uses 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.
ipv6-zone-sharding	Enables (true) or disables (false) zone-based sharding for IPv6 database. Default: False

Command Mode

CONFIG

VNFs

DRA Binding

Command Usage

Use database cluster binding ipv6-zone-sharding to enable/disable zone sharding on IPv6 database.

Examples

The following is an example of enabling zone-based sharding for the IPv6 database in the cluster `binding`:

```
database cluster binding ipv6-zone-sharding true
```

database cluster *<db name>* ipv6-zones-range *<zone-name>* zone-range *<range-name>* start *<pool starting address>* end *<pool ending address>*

Create zones for IPv6 shards based on IPv6 pools, so that the primary member of the replica set for an IPv6 address resides at the same physical location as the PGW assigning addresses from the IPv6 pool. This results in local writes (and reads) for the IPv6 binding database.

Note It is possible to create multiple ranges for each zone. Configure the IPv6 ranges in short format only.

Syntax

```
database cluster <db name> ipv6-zones-range
<zone-name> zone-range <range-name>
start <pool starting address> end
<pool ending address>
```

Command Parameters

Table 19: Parameter Description

Command Parameter	Description
DB Name	A short name describing the DB cluster. Each application uses 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.
Zone name	A short name describing Zone name. Unique name to identify the zone that the shard configuration uses to map to zone.
Range name	A short name describing the range within the zone.
Pool Starting Address	The starting IPv6 address for the particular range that can be from same physical location as PGW.
Pool Ending Address	The ending IPv6 address for the particular range that can be from same physical location as PGW.

Command Mode

CONFIG

VNFs

DRA Binding

Command Usage

This command creates a zone and also creates ranges for the zone.

Examples

The following is an example of creating a IPv6 zone with name `pune` for the cluster `binding` and a range of 2003:3051:0000:0001 to 2003:3051:0000:0500 for the zone:

```
database cluster binding ipv6-zones-range pune
zone-range range1 start 2003:3051:0000:0001 end
2003:3051:0000:0500
```

database cluster *<db name>* shard *<shard name>* zone-name *<zone-name>*

Add shards to a zone.

Syntax

```
database cluster <db name> shard
<shard name> zone-name <zone-name>
```

Command Parameters

Table 20: Parameter Description

Command Parameter	Description
DB Name	A short name describing the DB cluster. Each application uses 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.
Zone name	A short name describing Zone name.
Shard name	A short description of the shard name.

Command Mode

CONFIG

VNFs

DRA Binding

Command Usage

Use the command to add the shard to a zone.

Examples

The following is an example of mapping the IPv6 zone with name `pune` with the shard `shard-1` in the cluster `binding`:

```
database cluster binding shard shard-1 zone-name pune
```

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.

db connect binding

Connects to an underlying binding database.

Syntax

```
db connect binding { ipv4 | ipv6 | imsi-apn | msisdn-apn
| slf }
```

Command Parameters

Table 21: Parameter Description

Command Parameter	Description
ipv4	Connect to the IPv4 binding database.
ipv6	Connect to the IPv6 binding database.

Command Parameter	Description
imsi-apn	Connect to the IMSI-APN binding database.
msisdn-apn	Connect to the MSISDN-APN binding database.

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

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

db connect session

Connects to an underlying admin database.

Syntax

No additional arguments.

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use the db connect session command to connect to the underlying session database. Once within this database, the user will have read / write access to the session 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 22: 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 https://<master ip>/orchestrator/downloads/debug/<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 https://<master ip>/orchestrator/downloads/debug/<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 23: 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.

The results of the command are available at https://<master ip>/orchestrator/downloads/debug/tech.

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 24: 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 25: 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.

external-aaa pam gid-mapping

Configures the gid mapping for various group roles.

Syntax

```
external-aaa pam gid-mapping <gid:int> <group name>
```

Command Parameters

Table 26: Parameter Description

Command Parameter	Description
gid:int	GID mapping value.
group name	Group name for which gid mapping is required.

Command Mode

CONFIG

VNFs

All

Command Usage

Use `external-aaa pam gid-mapping` to configure LDAP user gid mapping for various group roles such as, grafana-admin, policy-admin, policy-ro, and so on.

Based on the roles configured for the LDAP user gid, access permissions can be set accordingly.

Example

```
admin@orchestrator(config)# external-aaa pam gid-mapping 1000 policy-admin
admin@orchestrator(config-gid-mapping-1000/policy-admin)# commit
Commit complete.
```

You can display the status of configuration by running the following command:

```
admin@orchestrator# show running-config external-aaa | tab
```

Sample Output:

```
admin@orchestrator# show running-config external-aaa | tab
GID   GROUP
--------------------
1000  policy-admin
```

license feature

Registers a system license.

Syntax

```
license feature id encrypted-license encrypted-license
```

```
no license feature id
```

Command Parameters

Table 27: 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 28: 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 29: 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 30: 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.
```

monitor log engine

Tails the cluster wide engine log using the `monitor log engine` command.

Syntax

```
monitor log engine
```

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

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

nacm rule-list

Specifies access restrictions for a user group.

Verify the users in the group before applying restrictions. To specify restrictions for any group, ensure that the admin user is not part of that group. By default, admin user is configured in a each group.

Syntax

```
nacm rule-list <rule-name>
group <group-name> cmdrule <cmdrule-name>
command <command to restrict>
access-operations exec action deny
```

Command Parameters

Table 31: Parameter Description

Command Parameter	Description
rule-list	Name of rule list.
group	Name of the group or list of groups to which the rules apply.
command	Command that is restricted for the user group.

Command Parameter	Description
access-operations	Used to match the operation that ConfD tries to perform. It must be one or more of the values from the accessoperations-type: create, read, update, delete, exec
action	If all of the previous fields match, the rule as a whole matches and the value of action (permit or deny) is taken. If a match is found, a decision is made whether to permit or deny the request in its entirety. If action is permit, the request is permitted; if action is deny, the request is denied.

Command Mode

CONFIG

VNFs

All

Command Usage

To delete the admin user from the read-only group, use the following command:

```
scheduler(config)#no nacm groups group crd-read-only
user-name admin
```

For the configuration to take effect, log out of the CLI session and log in again after configuring any nacm rule-list.

Examples

Restrict crd-read-only group from config command:

```
scheduler(config)#nacm rule-list crdreadgrp group
crd-read-only cmdrule denyconfig command config access-operations
exec action deny
scheduler(config-cmdrule-denyconfig)# commit
```

Restrict crd-read-only and policy-ro group from config command:

```
scheduler(config)#nacm rule-list readonly-restrict
group [ crd-read-only policy-ro ] cmdrule cfg-restrict command
config access-operations exec action deny
scheduler(config-cmdrule-cfg-restrict)#commit
```

Restrict crd-read-only and policy-ro group from docker command:

```
scheduler(config)#nacm rule-list readonly-restrict
group [ crd-read-only policy-ro ] cmdrule docker-restrict command
docker access-operations exec action deny
scheduler(config-cmdrule-docker-restrict)# commit
```

Restrict crd-read-only and policy-ro group from system stop command:

```
scheduler(config)#nacm rule-list readonly-restrict group
[ crd-read-only policy-ro ] cmdrule sys-stop command
"system stop" access-operations exec action deny
scheduler(config-cmdrule-sys-stop)# commit
```

Restrict crd-read-only and policy-ro group from system start command:

```
scheduler(config)#nacm rule-list readonly-restrict
group [ crd-read-only policy-ro ] cmdrule sys-start command
"system start" access-operations exec action deny
scheduler(config-cmdrule-sys-start)# commit
```

Restrict load override command for all the users including admin:

```
scheduler(config)#nacm rule-list readonly-restrict
group [ * ] cmdrule load-override command "load override"
access-operations exec action deny
scheduler(config-cmdrule-load-override)# commit
```

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 32: 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 33: 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.

Syntax

```
network virtual-service name of floating ip floating-ip floating ip address mask
net mask digits broadcast broadcast address interface interface-id virtual-router-id
virtual router id tracking-service prefix of service to monitor for IP address
diameter-endpoint 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 34: Parameter Description

Command Parameter	Description
name of floating ip	Name of the floating IP address. to be configured Virtual Network Service Name must contain a minimum of 1 character and a maximum length of 8 characters.
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.
interface-id	Interface ID.
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

VNFs

All

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. VRRP is configured as protocol number and not name. Hence, while configuring from dashboard, select protocol as 'Other' and in the text box below, enter 112 as protocol.

Examples

The following example creates a floating IP on two hosts:

Note Enter the command manually.

IPv4 VIP config:

```
scheduler(config)# network virtual-service GxVip12 floating-ip 172.22.33.51 mask 24 broadcast
 172.22.33.255 interface
ens161 virtual-router-id 1 tracking-service diameter-endpoint host 172.22.33.43 priority 2
exit
host 172.22.33.44 priority 1
commit
end
```

IPv6 VIP config:

```
scheduler(config)# network virtual-service RxVip12 floating-ip 2003:2235::51 mask 64 interface

ens192 virtual-router-id 2 tracking-service diameter-endpoint host 2003:2235::44 priority
2
exit
host 2003:2235::43 priority 1
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 GxVip12
 virtual-router-id 1
 floating-ip       172.22.33.51
 mask              24
 broadcast         172.22.33.255
 host 172.22.33.43
  priority 2
 !
```

```
 host 172.22.33.44
  priority 1
 !
!
```

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 you want a 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 35: 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 36: 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 37: 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 ",".

Command Parameter	Description
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 38: 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.

Command Parameter	Description
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 39: 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 40: Parameter Description

Parameter	Description
Name	Rule-name of the alert.
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

Display the currently configured database clusters members.

Syntax

```
show database status
```

Command Parameters

Table 41: Parameter Description

Command Parameter	Description
Address	The address of the database process.

Command Parameter	Description
Port	The port the database service is running.
Name	Name of the database process.
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 42: 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 43: 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 44: 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 45: 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 running-config binding db-connection-settings

Displays the binding DB connection settings.

Syntax

```
show running-config binding db-connection-settings
```

Command Mode

OPERATIONAL

VNFs

All

Examples

The following is an example:

```
scheduler# show running-config binding db-connection-settings | tab
                           MAX
BINDING  CONNECT  SOCKET   WAIT  CONNECTIONS
TYPE     TIMEOUT  TIMEOUT  TIME  PER HOST
-------------------------------------------
drasession       500      1000     500   10
```

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 46: 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 47: 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 48: 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 76.

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 49: 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 50: 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/
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 51: 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 52: 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 53: Parameter Description

Command Parameter	Description
category	Category of the statistic.
name	Name of the statistic.
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 54: 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 55: 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 71 command should be issued to revert this software to the previous version.

system downgrade

Downgrades the system to a new software version.

Syntax

```
system downgrade version version
```

Command Mode

OPERATIONAL

VNFs

All

Command Parameters

Table 56: Parameter Description

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

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 55 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 57: 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 59.

system secrets remove-secret

Removes a secret from the system.

Syntax

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

Command Mode

OPERATIONAL

VNFs

All

Command Parameters

Table 58: 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 59.

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 59: 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 59.

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 60: 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 61: 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 62: 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 63: 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 64: 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

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 65: 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 55 command.

Examples

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