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
```

abort

Used to terminate a configuration session and discard all uncommitted changes without system confirmations. You can use the abort command in any configuration mode.

Syntax

```
abort
```

Command Mode

CONFIG

VNFs

All

Command Usage

Use the abort command to terminate a configuration session and return to the operational mode from any configuration mode. This command causes all uncommitted configuration changes to be discarded. You are not prompted to commit the changes.

Examples

The following is an example:

```
aaa authentication users user test1 password test123 gid 100 homedir / ssh_keydir / uid
9340
admin@orchestrator[an-master](config-user-test1)# exit
admin@orchestrator[an-master](config)# abort
admin@orchestrator[an-master]#
```

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
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.

Command Parameter	Description
snmp-facility (optional)	The target snmp-facility to use when generating SNMP trap: • application • hardware • networking • os • proc • virtualization Default is application.
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"
!
```

Note Alert message supports threshold label and the format for threshold label support in alert message is `{{$threshold}}`.

alert clear-stale-alerts

The **show alert status** output displays Stale alerts. Stale alerts get created when all prometheus instances are brought down and back up. When prometheus instances come up, they recognise only new alerts and will not recognise existing alerts in firing state. Hence the Update time for the alerts in firing state will not be updated and will end up as Stale Alerts.

Examples

Following example shows alert status with Stale alerts. The update time for these alerts does not have latest timestamp and is different in the other alerts.

```
alert status
NAME EVENT HOST STATUS MESSAGE
CREATE TIME RESOLVE TIME UPDATE TIME
------------------------------------------------------------------------------------------------------------------------------------------------------
IP_NOT_REACHABLE 192.168.10.10 resolved VM/VIP IP 192.168.10.10 is reachable.
2021-05-26T16:12:23.545+00:00 2021-05-26T16:12:28.543+00:00 2021-05-26T16:13:38.55+00:00
IP_NOT_REACHABLE 192.168.10.13 firing VM/VIP IP 192.168.10.13 is not reachable.
2021-05-27T10:32:08.542+00:00 - 2021-05-27T10:32:13.555+00:00
IP_NOT_REACHABLE 192.168.10.14 firing VM/VIP IP 192.168.10.14 is not reachable.
2021-05-27T10:32:08.546+00:00 - 2021-05-27T10:32:13.556+00:00
IP_NOT_REACHABLE 192.168.10.17 firing VM/VIP IP 192.168.10.17 is not reachable.
2021-05-27T10:32:13.556+00:00 - 2021-05-27T10:32:13.556+00:00
IP_NOT_REACHABLE 192.168.10.23 resolved VM/VIP IP 192.168.10.23 is reachable.
2021-05-26T16:12:23.55+00:00 2021-05-26T16:12:28.539+00:00 2021-05-26T16:13:38.537+00:00
IP_NOT_REACHABLE 192.168.10.25 resolved VM/VIP IP 192.168.10.25 is reachable.
2021-05-26T16:12:23.551+00:00 2021-05-26T16:12:28.54+00:00 2021-05-26T16:13:38.547+00:00
```

Syntax

```
alert clear-stale-alerts
```

Command Parameters

None

Command Mode

CONFIG

VNFs

All

Command Usage

Use the following **alert-clear-stale-alerts** CLI command to clear stale alerts.

```
alert ?
Possible completions:
clear-all Clear all alerts
clear-stale-alerts Clear all stale alerts
rule
snmp-v2-destination
snmp-v3-destination
alert clear-stale-alerts
Stale Alerts removed
```

Examples

Following example shows alert status output after the removal of Stale alerts.

```
 show alert status
Name EVENT HOST STATUS MESSAGE
CREATE TIME RESOLVE TIME UPDATE TIME
-------------------------------------------------------------------------------------------------------------------------------------------------------
IP_NOT_REACHABLE 192.168.10.10 resolved VM/VIP IP 192.168.10.10 is reachable.
2021-05-26T16:12:23.545+00:00 2021-05-26T16:12:28.543+00:00 2021-05-26T16:13:38.55+00:00
IP_NOT_REACHABLE 192.168.10.23 resolved VM/VIP IP 192.168.10.23 is reachable.
2021-05-26T16:12:23.55+00:00 2021-05-26T16:12:28.539+00:00 2021-05-26T16:13:38.537+00:00
IP_NOT_REACHABLE 192.168.10.25 resolved VM/VIP IP 192.168.10.25 is reachable.
2021-05-26T16:12:23.551+00:00 2021-05-26T16:12:28.54+00:00 2021-05-26T16:13:38.547+00:00
```

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.

Command Parameter	Description
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.

Command Parameter	Description
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
```

api-user add group-details gid auth-type write-enable

Used to create or remove API dedicated users using group details.

Syntax

```
api-user add group-details gid <GID> auth-type local/external write-enable true/false
```

```
api-user remove group-details gid <GID> auth-type local/external write-enable true/false
```

Command Parameters

Table 7: Parameter Description

Command Parameter	Description
GID	Group ID of the user.

Command Mode

OPERATIONAL

VNFs

All

Command Usage

Use this command to create or remove API dedicated users using group details.

Examples

Example 1: The following is an example to create an user:

```
api-user add group-details gid 100 auth-type external write-enable false
```

Exampe 2: The following is an example to remove the user:

```
api-user remove group-details gid 100 auth-type external write-enable false
```

api-user add user-details name auth-type write-enable

Used to create or remove API dedicated user.

Syntax

```
api-user add user-details name <USER_NAME> auth-type local/external write-enable true/false
api-user remove user-details name <USER_NAME> auth-type local/external write-enable true/false
```

Command Parameters

Table 8: Parameter Description

Command Parameter	Description
USER_NAME	Name of the user.

Command Mode

OPERATIONAL

VNFs

All

Command Usage

Use this command to create or remove an API dedicated user.

Note Set the `write-enable` to `true` to give the read-write API access and set as `false` to give the read API access.

Examples

Example 1: The following is an example to create an user:

```
api-user add user-details name test auth-type local write-enable true
```

Exampe 2: The following is an example to remove the user:

```
api-user remove user-details name test auth-type local write-enable true
```

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 9: 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 cluster-binding-dbs imsiapn-msisdnapn

Used to configure same connection pool on IMSIAPN-MSISDNAPN database transactions.

Note This command is applicable only for application client based sharding.

Syntax

```
binding cluster-binding-dbs imsiapn-msisdnapn
```

```
no binding cluster-binding-dbs
```

Command Mode

CONFIG

VNFs

DRA

Command Usage

Use this CLI to indicate to the application that IMSI APN bindings DB and MSISDN APN Bindings DB will use the same connection pool for DB transactions.

IMSI-APN connection settings for both read and write will apply to this combined pool.

In this mode change in MSISDN APN connection settings for read or write connection pools will have no effect.

Note This is not recommended for small deployments. It is required for the deployments for which the database spans across 48 shards or more.

Examples

The following is an example:

```
admin@orchestrator(config)# binding cluster-binding-dbs imsiapn-msisdnapn
```

binding db-connection

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

Note This command is applicable only for MongoDB based sharding.

Syntax

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

Command Parameters

Table 10: Parameter Description

Command Parameter	Description
ipv4	Connection definition for the IPv4 binding database.
ipv6	Connection definition for the IPv6 binding database.
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 write mongo connection settings. The connections are used for database create/update and delete of session and bindings.

Note This command is applicable for MongoDB based and application client based sharding.

Syntax

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

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

Note For Policy DRA, supported values are drasession/imsiapn/ipv4/ipv6/msisdnapn.

For recommended values, refer to *Database Connection Settings* section in the *CPS vDRA Advanced Tuning Guide*.

Command Parameters

Table 11: 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.

Command Parameter	Description
acceptable-latency-difference-for-read	The maximum difference in ping-time latency between the fastest ping time and the slowest of the chosen servers. Default: 5
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 write 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:
  acceptable-latency-difference  connect-timeout  connections-per-host  max-wait-time
socket-timeout  <cr>

admin@orchestrator(config-db-connection-settings- drasession)# acceptable-latency-difference
 ?
Possible completions:
  <int>[5]
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]
```

binding db-max-record-limit

Used to configure maximum record limit on session and bindings.

Syntax

```
binding db-max-record-limit { all | drasession | imsiapn | ipv4 | ipv6 | msisdnapn | range
 | slf } <limit>
```

```
no binding db-max-record-limit drasession <limit>
```

Command Parameters

Table 12: Parameter Description

Command Parameter	Description
all	Maximum record limit on drasession, ipv6, ipv4, imsiapn and msisdnapn.
drasession	Maximum record limit on DRA session.
imsiapn	Maximum record limit on IMSI-APN.
ipv4	Maximum record limit on IPv4.
ipv6	Maximum record limit on IPv6.
msisdnapn	Maximum record limit on MSISDN-APN.
range	Not Applicable
slf	Not Applicable
limit	Maximum number of records to be stored in database. Default: Value of limit depends on deployment and number of shards. Hence, no default value for limit.

Command Mode

CONFIG

VNFs

DRA

Command Usage

Use the `db-max-record-limit` command to configure maximum record limit on session and bindings.

Examples

The following is an example:

```
admin@orchestrator[master-0m](config)# binding db-max-record-limit
Possible completions:
  all  drasession  imsiapn  ipv4  ipv6  msisdnapn  range  slf

admin@orchestrator[master-0m](config)# binding db-max-record-limit all  1000
admin@orchestrator[master-0m](config)# binding db-max-record-limit drasession 1000
admin@orchestrator[master-0m](config)# binding db-max-record-limit imsiapn 1000
admin@orchestrator[master-0m](config)# binding db-max-record-limit ipv4 1000
admin@orchestrator[master-0m](config)# binding db-max-record-limit ipv6 1000
admin@orchestrator[master-0m](config)# binding db-max-record-limit msisdnapn 1000
```

binding db-read-connection-settings

Used to configure the read mongo connection parameters.

Note This command is applicable only for application client based sharding.

Read connections are used for:

- Rx-AAR based binding look up
- Rest API binding query
- Reset of next evaluation time for both sessions and bindings
- Health checks

Syntax

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

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

Note For Policy DRA, supported values are drasession/imsiapn/ipv4/ipv6/msisdnapn.

For recommended values, refer to *Database Connection Settings* section in the *CPS vDRA Advanced Tuning Guide*.

Command Parameters

Table 13: 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.
acceptable-latency-difference-for-read	The maximum difference in ping-time latency between the fastest ping time and the slowest of the chosen servers. Default: 5
connect-timeout-for-read	Connection timeout in milliseconds for read connection. It is used only when establishing a new connection. Default: 500
connections-per-host-for-read	Maximum number of connections allowed per host for this MongoClient instance of read connection. 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: 5
max-wait-time-for-read	Maximum wait time in milliseconds that a thread may wait for a connection to become available. Default: 500
socket-timeout-for-read	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-read-connection-setting` commands to configure the read mongo connection parameters. Applicable only for connection with client-sharded database cluster.

Examples

The following is an example for setting the connection-per-host for read connections with session-db to 5:

```
admin@orchestrator[master-0](config)# binding db-read-connection-settings drasession
connections-per-host-for-read 5
```

binding imsi-msisdn enable aggregate query

Use the CLI command to set the flag (true or false) to enable or disable the aggregate status query for IMSI and MSISDN binding tables.

Syntax

```
binding imsi-msisdn enable aggregate query True|False
```

Command Parameters

Table 14: Parameter Description

Command Parameter	Description
True | false	Set the flag (true or false) to enable or disable the DB aggregate status query for IMSI or MSISDN binding table.

Command Mode

CONFIG

VNFs

DRA

Command Usage

The CLI command set the flag value as true or false to enable or disable the aggregate status query for IMSI and MSISDN binding tables.

Examples

Following is an example to set the flag value:

```
admin@orchestrator[vpas-A1-master-0](config)# binding imsi-msisdn enable-aggregate-query
Possible completions:
  true  false
```

binding shard-metadata-db-connection

Used to configure binding shard metadata database connections from DRA to a DRA shard metadata binding database.

Note This command is applicable only for application client based sharding.

Syntax

```
binding shard-metadata-db-connection { all | drasession | imsiapn | ipv4 | ipv6 | loadmetrics
 | msisdnapn | range } <ip-address> <port>
```

```
no binding shard-metadata-db-connection { drasession | imsiapn | ipv6 | loadmetrics |
msisdnapn } <ip-address> <port>
```

Command Parameters

Table 15: Parameter Description

Command Parameter	Description
all	Connection definition for Session, IPv4, IPv6, IMSI-APN, MSISDN-APN shard metadata binding database.
drasession	Connection definition for the Session binding shard metadata database.
imsiapn	Connection definition for the IMSI-APN shard metadata binding database.
ipv4	Connection definition for the IPv4 binding shard metadata database.
ipv6	Connection definition for the IPv6 binding shard metadata database.
loadmetrics	Connection definition for the IMSI-APN or MSISDN-APN shard metadata binding database.
msisdnapn	Connection definition for the MSISDN-APN shard metadata binding database.
range	Not Applicable
ip-address	Address of the binding DRA database. This is either an IP address or an FQDN.
port	Port number of the binding DRA database.

Command Mode

CONFIG

VNFs

DRA

Command Usage

Use the `binding shard-metadata-db-connection` command to instruct the application on how to connect to the remote shard metadata 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 shard-metadata-db-connection drasession 172.16.82.195 27017
!
binding shard-metadata-db-connection drasession 172.16.82.196 27017
!
binding shard-metadata-db-connection ipv6 172.16.82.195 27017
!
binding shard-metadata-db-connection ipv6 172.16.82.196 27017
!
binding shard-metadata-db-connection ipv4 172.16.82.195 27017
!
binding shard-metadata-db-connection ipv4 172.16.82.196 27017
!
binding shard-metadata-db-connection imsiapn 172.16.82.195 27017
!
binding shard-metadata-db-connection imsiapn 172.16.82.196 27017
!
binding shard-metadata-db-connection msisdnapn 172.16.82.195 27017
!
binding shard-metadata-db-connection msisdnapn 172.16.82.196 27017
!
binding shard-metadata-db-connection loadmetrics 172.16.82.195 27017
!
binding shard-metadata-db-connection loadmetrics 172.16.82.196 27017
!
```

binding throttle-db-operation

Used to configure CPU usage threshold value for read and write database operations.

Note This command is applicable only for application client based sharding.

Important The following commands need to be configured to monitor CPU usage of all the database VMs:

`binding shard-metadata-db-connection loadmetrics <ip-address> <port>`

For more information on `binding shard-metadata-db-connection`, refer to binding shard-metadata-db-connection, on page 27.

Syntax

`binding throttle-db-operation { range | read | write } max-cpu-usage <cpu_value>`

`no binding throttle-db-operation { range | read | write } max-cpu-usage`

Command Parameters

Table 16: Parameter Description

Command Parameter	Description
range	Not applicable.
read	CPU threshold for read database operations.
write	CPU threshold for write database operations.
cpu_value	CPU threshold value.

Command Mode

CONFIG

VNFs

DRA

Command Usage

Use the `binding throttle-db-operation` command to configure the read and write CPU threshold value to throttle the read and write database operations.

Examples

The following configuration defines CPU threshold value for read and write database operations:

```
binding throttle-db-operation read max-cpu-usage 70
!
binding throttle-db-operation write max-cpu-usage 70
!
```

clear

Used to clear uncommitted changes.

Syntax

```
clear
```

Command Mode

CONFIG

VNFs

All

Command Usage

Use the clear command to discard all the uncommited changes.

Examples

The following is an example:

```
clear
All configuration changes will be lost. Proceed? [yes, NO]
```

compare

Used to compare the similar configurations.

Syntax

```
compare cfg <configuration path> to <configuration path>
```

Command Mode

CONFIG

VNFs

All

Command Usage

- To compare the similar configurations in configuration mode.
- Need to represent exact ideal configuration paths.

Examples

The following is an example:

```
compare cfg aaa authentication users user admin to aaa authentication users user oper
- password $1$ftGF2fQE$4P145tnwbouLSr8pbm4EW1;
+ password $1$sFadxrqz$Tp88/Go3jTNUuloSdPB9K.;
- ssh_keydir /var/confd/homes/oper/.ssh;
+ ssh_keydir /var/confd/homes/admin/.ssh;
- homedir /var/confd/homes/oper;
+ homedir /var/confd/homes/admin;
```

consul

Used to list, save, delete, and restore the consul snapshot from the `/data/orchestrator/config/snapshot/` directory.

Syntax

```
consul [list-snapshots | save-snapshot snapshot-name nameofsnapshot | restore-snapshot
snapshot-name nameofsnapshot | delete-snapshot nameofsnapshot]
```

Command Parameters

Table 17: Parameter Description

Command Parameter	Description
list-snapshots	Lists all the snapshots present in `/data/orchestrator/config/snapshot/` directory.
save-snapshot	Saves the snapshot.
snapshot-name *nameofsnapshot*	Snapshot name.
restore-snapshot	Restore the snapshot.
delete-snapshot	Delete the snapshot.

Command Mode

OPERATIONAL

VNFs

All

Command Usage

Use the `consul` command to list, save, delete, and restore the consul snapshot in the `/data/orchestrator/config/snapshot/` directory.

Examples

The following are the examples:

```
admin@orchestrator[ss-master-binding-0]# consul list-snapshots
Snapshot Name
******************************************************

19.5.5-20200105_131756.6477
19.5.8-20200214_025459.6674

******************************************************
admin@orchestrator[ss-master-binding-0]# consul save-snapshot snapshot-name snap1
result Snapshot is created in /data/orchestrator/config/snapshot-consul/snap1
admin@orchestrator[ss-master-binding-0]# consul list-snapshots
Snapshot Name
******************************************************

19.5.5-20200105_131756.6477
19.5.8-20200214_025459.6674
```

```
snap1

*******************************************************
admin@orchestrator[ss-master-binding-0]# consul save-snapshot snapshot-name snap2
result Snapshot is created in /data/orchestrator/config/snapshot-consul/snap2
admin@orchestrator[ss-master-binding-0]# consul list-snapshots
Snapshot Name
*******************************************************

19.5.5-20200105_131756.6477
19.5.8-20200214_025459.6674
snap1
snap2

*******************************************************
admin@orchestrator[ss-master-binding-0]# consul delete-snapshot snap2
Snapshot is deleted
admin@orchestrator[ss-master-binding-0]# consul list-snapshots
Snapshot Name
*******************************************************

19.5.5-20200105_131756.6477
19.5.8-20200214_025459.6674
snap1

*******************************************************
admin@orchestrator[ss-master-binding-0]# consul restore-snapshot ?
Possible completions:
  snapshot-name
admin@orchestrator[ss-master-binding-0]# consul restore-snapshot snapshot-name snap1
result Snapshot restore success.
admin@orchestrator[ss-master-binding-0]# consul list-snapshots
Snapshot Name
*******************************************************

19.5.5-20200105_131756.6477
19.5.8-20200214_025459.6674
snap1

*******************************************************
admin@orchestrator[ss-master-binding-0]# consul delete-snapshot snap1
Snapshot is deleted
admin@orchestrator[ss-master-binding-0]#
```

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 18: Parameter Description

Command Parameter	Description
Name	A short name describing the connection.

Command Parameter	Description
address	An IP address or FQDN of the connection. IPv6 address must be enclosed in square brackets.
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
```

control-plane ipc-endpoint update-interval

Used to configure IPC endpoint update interval.

Syntax

```
control-plane ipc-endpoint update-interval <time-in-milliseconds>
no control-plane ipc-endpoint update-interval
```

Command Parameters

Table 19: Parameter Description

Command Parameter	Description
time-in-milliseconds	IPC endpoint update interval in milliseconds. Default: 100 milliseconds

Command Mode

CONFIG

VNFs

DRA

Command Usage

This command is used to configure the frequency for updating the IPC endpoints.

Examples

The following configuration adds an 200 milliseconds interval for updating the IPC endpoints.

```
scheduler(config)# control-plane ipc-endpoint update-interval 200
```

Note For more information on the values to be configured, refer to *Control Plane Tuning Configuration* section in the *CPS vDRA Advanced Tuning Guide*.

control-plane remote-peer-policy global accept

Used to configure the control plane remote peer policy.

Syntax

```
control-plane remote-peer-policy global accept all
```

```
control-plane remote-peer-policy global accept diameter-applications [Gx | Gy | Rx | Sd |
Sy]
```

Command Parameters

Table 20: Parameter Description

Command Parameter	Description
[Gx | Gy | Rx | Sd | Sy]	Application type.

By default, DRA accepts all the applications from all the sites.

Command Mode

CONFIG

VNFs

DRA

Command Usage

This command is used to configure the control plane remote peer policy for the DRA system to accept peer connection information from other DRA systems. Policy can be configured to accept peer connection

information for all Diameter application types or only specific Diameter application types. DRA system can route messages only to remote peers accepted by policy.

Examples

Example 1:

```
control-plane remote-peer-policy global accept diameter-applications Rx
```

Example 2:

```
control-plane remote-peer-policy global accept diameter-applications [ Gx Rx Gy ]
```

Example 3:

```
control-plane remote-peer-policy global accept all
```

Example 4:

```
no control-plane remote-peer-policy global accept diameter-applications [ Gx Rx Gy ]
```

control-plane remote-peer-policy global filter-fqdn-with-string

Used to filter the RxB control plane messages, which is configured as a list of substring. It can hold upto five types of strings in the list.

Syntax

```
control-plane remote-peer-policy global filter-fqdn-with-string
```

```
control-plane remote-peer-policy global filter-fqdn-with-string [ str1 str2 str3 str4 str5
 ]
```

Command Parameters

Table 21: Parameter Description

Command Parameter	Description
[str1 str2 str3 str4 str5]	Types of strings.

By default, DRA accepts all the applications from all the sites.

Command Mode

CONFIG

VNFs

DRA

Command Usage

This command is used to configure the control plane remote peer policy for the RxB peer FQDNs as a list of substrings. The command usage does not allow more than five types of strings. Whenever the configuration is done, it replaces the existing configuration to limit the content of the list.

Note Configure the part of FQDN as a substring.

Examples

Following is an example to configure the control plane remote peer policy for the RxB peer FQDN: site-b-server-calipers21-rxb.af.b-rx-b1-1 as a list of substrings:

```
control-plane remote-peer-policy global filter-fqdn-with-string [ rxb ]
```

control-plane remote-peer-policy mated-system id

Used to configure the mated system ID.

Syntax

```
control-plane remote-peer-policy mated-system id <system-id>
```

Command Parameters

Table 22: Parameter Description

Command Parameter	Description
system-id	System ID of the mated system.

Command Mode

CONFIG

VNFs

DRA

Command Usage

This command is used to configure the system ID of the mated DRA system. DRA system accepts peer information for all Diameter application types from the mated system.

Example

```
control-plane remote-peer-policy mated-system id system-02
```

control-plane timers peer-status-update-interval

Used to modify the value of peer status update interval and peer expiration duration.

Syntax

```
control-plane timers peer-status-update-interval <time-in-ms> peer-expiration-duration
<duration-in-ms>
```

Command Parameters

Table 23: Parameter Description

Command Parameter	Description
time-in-ms	Peer status update interval time in ms. Default: 2000 milliseconds
duration-in-ms	Peer expiration duration in ms. Default: 10000 milliseconds

Command Mode

CONFIG

VNFs

DRA

Command Usage

This command allows tuning the frequency at which director nodes send periodic status updates for peers connected to the nodes. The command also allows tuning the expiration time for peers maintained in topology when consecutive periodic status updates are not received for the peers.

Peer expiration duration should be equal to three times of peer status update interval.

For example, if peer-status-update-interval = 4000 ms then, peer-expiration-duration = 12000

To reflect the peer expiration duration change, application should be restarted in both director and worker nodes.

Example

```
control-plane timers peer-status-update-interval 4000 peer-expiration-duration 12000
```

control-plane timers dynamic-peer-ratelimit dbcpu-status-update-interval

Used to configure the control plane messages for DB VM CPU from Worker to Director.

Syntax

```
control-plane timers dynamic-peer-ratelimit dbcpu-status-update-interval <time-in-ms>
```

Command Parameters

Table 24: Parameter Description

Command Parameter	Description
time-in-ms	The DB VM CPU status update interval time in milliseconds. Default: 10000 milliseconds

Command Mode

CONFIG

VNFs

DRA

Command Usage

This command allows tuning the frequency at which worker nodes send periodic status updates for DRA DB VM CPU messages to the Director. As DB VM CPU is updated every 10 seconds and duplicate DB CPU VM status gets published, it is recommended to configure minimum update interval as 10 seconds or multiple of 10 seconds.

Example

```
control-plane timers dynamic-peer-ratelimit dbcpu-status-update-interval 20000
```

crd access-restriction externaluser-tablegroup-mapping add-gid

Used to map the external users to the CRD table group.

Syntax

```
crd access-restriction externaluser-tablegroup-mapping add-gid "table-group-name" "group-id"
```

Command Parameters

Table 25: Parameter Description

Command Parameter	Description
group-id	Group mapping value.

Command Mode

CONFIG

VNFs

All

Command Usage

Use this CLI command to map the external users to the CRD table group.

Examples

Example 1: The following is an example to assign a group ID to the CRD table group:

```
crd access-restriction externaluser-tablegroup-mapping add-gid “apm_tool” “100”
```

Example 2: The following is an example to remove a group ID from the CRD table group:

```
crd access-restriction externaluser-tablegroup-mapping remove-gid “apm_tool” “100”
```

Example 3: The following is an example to display the assigned CRD table group for a Group ID:

```
crd access-restriction externaluser-tablegroup-mapping show-tablegroup “100”
```

Example 4: The following is an example to display the assigned Group IDs for the CRD table group:

```
crd access-restriction externaluser-tablegroup-mapping show-gid “apm_tool”
```

crd access-restriction localuser-tablegroup-mapping add-user

Used to map the local users to the CRD table group.

Syntax

```
crd access-restriction localuser-tablegroup-mapping add-user "table-group-name" "table-name"
```

Command Parameters

Table 26: Parameter Description

Command Parameter	Description
user-name	Name of the user to be assigned.

Command Mode

CONFIG

VNFs

All

Command Usage

Use this CLI command to map the local users to the CRD table group.

Examples

Example 1: The following is an example to assign a user to the CRD table group:

```
crd access-restriction localuser-tablegroup-mapping add-user “apm_tool” “oper”
```

Example 2: The following is an example to remove a user from the CRD table group:

```
crd access-restriction localuser-tablegroup-mapping remove-user "apm-tool" “oper”
```

Example 3: The following is an example to display the assigned CRD table group for a user:

```
crd access-restriction localuser-tablegroup-mapping show-tablegroup “oper”
```

Example 4: The following is an example to display the assigned users for CRD table group:

```
crd access-restriction localuser-tablegroup-mapping show-users “apm_tool”
```

crd access-restriction table-group add-table

Used to create, remove and display the CRD table group.

Syntax

```
crd access-restriction table-group add-table "table-group-name" "table-name"
```

Note As the running-config backup does not save the backup of the *crd* configuration, take the backup using the following command:

show crd access-restriction table-groups | save /data/config/backup.txt command.

Command Parameters

Table 27: Parameter Description

Command Parameter	Description
table-group-name	Name of the CRD table group.
table-name	Name of the CRD table.

Command Mode

OPERATIONAL

VNFs

All

Command Usage

Use this CLI command to create, remove and display the CRD table group or CRD table.

Examples

Example 1: The following is an example to create a CRD table group and add a table to the group:

```
crd access-restriction table-group add-table “amp_tool” “apn_mapping”
```

Example 2: The following is an example to remove a CRD table from CRD table group:

```
crd access-restriction table-group remove-table “amp_tool” “apn_mapping”
```

Example 3: The following is an example to display the added CRD table list from CRD table group:

```
crd access-restriction table-group show-tables "amp_tool"
```

Example 4: The following is an example to display the list of CRD table groups:

```
show crd-access-restriction table-groups
```

Example 5: The following is an example to display the list of all CRD tables:

```
show crd tables
```

database check-ipv6-zone-config

Used to verify the discrepancy of ipv6 zones ranges configured on Mongo Database and Confd.

Syntax

```
database check-ipv6-zone-config Cluster Name
```

Command Parameters

Table 28: Parameter Description

Command Parameter	Description
Cluster Name	Name of the cluster to which shard belongs.

Command Mode

Operational

VNFs

Binding

Command Usage

Use this **database check-ipv6-zone-config** command to find out the discrepancy of ipv6 zone ranges configured on Mongo Database and confd.

Examples

Example 1: If no discrepancy is found for the cluster:

```
admin@orchestrator[an-dbmaster]# database check-ipv6-zone-config binding
Shard_DB Primary IP is for binding is 192.168.11.41 and its port number 27030
Enter Mongo Admin password:
Starting  ipv6shardb zones Validation..
#### Verifying ipv6 zone discrepancy from CLI Running Config to Mongo database. ####
#### Verifying ipv6 zone discrepancy from Mongo Database to CLI Running Config. ####
IPV6 zone Config Shardb Validation Completed..
```

Example 2: If discrepancy is found for the cluster:

```
admin@orchestrator[an-dbmaster]# database check-ipv6-zone-config binding
Shard_DB Primary IP is for binding is 192.168.11.41 and its port number 27030
Enter Mongo Admin password:
```

```
Starting  ipv6shardb zones Validation..
#### Verifying ipv6 zone discrepancy from CLI Running Config to Mongo database. ####
**************************************************
FAILURE: IPV6 Zone ASGHAR and its Range ASGHAR_TEST for Start Value
2606:ae00:bd80:0000:0000:0000:0000:0010 and End Value 2606:ae00:bd80:0000:0000:0000:0000:0020
 is InValid. Please check..
**************************************************
**************************************************
FAILURE: IPV6 Zone MEGAZONE2 and its Range MEGARANGE3 for Start Value 2606:ae00:90a0:0000
and End Value 2606:ae00:90af:ffff is InValid. Please check..
**************************************************
#### Verifying ipv6 zone discrepancy from Mongo Database to CLI Running Config. ####
**************************************************
FAILURE: IPV6 Zone ASGHAR and its Range ASGHAR_TEST for Start Value
2606:ae00:bd80:0000:0000:0000:0000:0030 and End Value 2606:ae00:bd80:0000:0000:0000:0000:0040
 is InValid. Please check..
**************************************************
**************************************************
FAILURE: IPV6 Zone AA and its Range ASGHAR_AA for Start Value
2606:ae00:bd80:0000:0000:0000:0000:1111 and End Value 2606:ae00:bd80:0000:0000:0000:0000:1111
 is InValid. Please check..
**************************************************
IPV6 zone Config Shardb Validation Completed..
```

Example 3: If zones were not configured for cluster but user tries to validate the zones:

```
admin@orchestrator[an-dbmaster]# database check-ipv6-zone-config imsi
FAILURE: imsi is not Configured..
```

database clearzoneinfo dump clustername

Used to print all IPV6 zone range configuration entries present in Mongo shard db for the cluster.

Syntax

```
database clearzoneinfo dump clustername CLUSTER-NAME
```

Command Parameters

Table 29: Parameter Description

Command Parameter	Description
CLUSTER-NAME	Name of the cluster to which shard belongs.

Command Mode

OPERATIONAL

VNFs

Binding

Command Usage

This command prints the IPV6 range configuration entries in the Mongo sharddb for the cluster.

database clearzoneinfo delete-all clustername

Used to delete and re-add all the IPV6 range configuration entries in the Mongo sharddb for the cluster.

Prerequisties

Edit the NACM rule (in configuration mode) to permit deletion of the database ipv6-zones in the DB VNF.

```
nacm rule-list any-group rule data-base access-operations delete action permit
```

Syntax

```
database clearzoneinfo delete-all clustername CLUSTER-NAME
```

Command Parameters

Table 30: Parameter Description

Command Parameter	Description
CLUSTER-NAME	Name of the cluster to which shard belongs.

Command Mode

OPERATIONAL

VNFs

Binding

Command Usage

This command deletes and re-adds all the IPV6 range configuration entries in the Mongo sharddb for the cluster.

database clearzoneinfo delete clustername zonename rangename

Used to delete a single IPV6 zone range configuration entry in the Mongo sharddb for the cluster.

Prerequisties

Edit the NACM rule (in configuration mode) to permit deletion of the database ipv6-zones in the DB VNF.

```
nacm rule-list any-group rule data-base access-operations delete action permit
```

Syntax

```
database clearzoneinfo delete clustername CLUSTER-NAME zonename ZONE-NAME rangename RANGE-NAME
```

Command Parameters

Table 31: Parameter Description

Command Parameter	Description
ZONE-NAME	A short name describing the Zone . An 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.

Command Mode

OPERATIONAL

VNFs

Binding

Command Usage

This command deletes a single IPV6 zone range configuration entry in the Mongo sharddb for the cluster.

database clearzoneinfo update clustername zonename rangename start end

Used to update a single IPV6 zone range config entry in the Mongo sharddb for the cluster with the new starting and ending IPV6 addresses.

Syntax

```
database clearzoneinfo update clustername zonename rangename start START end END
```

Command Parameters

Table 32: Parameter Description

Command Parameter	Description
START	Starting address of IPV6 address range.
END	Ending address of IPV6 address range.

Command Mode

OPERATIONAL

VNFs

Binding

Command Usage

This command updates a single IPV6 zone range config entry in the Mongo sharddb for the cluster with the new starting and ending ipv6 addresses.

database clearzoneinfo delete-zone clustername binding zonename

Used to delete all the range entries for a single IPV6 zone configuration in Mongo sharddb for the cluster.

Syntax

```
database clearzoneinfo delete-zone clustername binding zonename ZONE-NAME
```

Command Parameters

Table 33: Parameter Description

Command Parameter	Description
ZONE-NAME	A short name describing the Zone . An unique name to identify the zone that the shard configuration uses to map to zone.

Command Mode

OPERATIONAL

VNFs

Binding

Command Usage

This command deletes all the range entries for a single IPV6 zone configuration in Mongo sharddb for the cluster.

Examples

The following is an example to delete all range entries in ZONE1 of IPV6 zone configuration.

```
admin@orchestrator[an-dbmaster]# show running-config database cluster binding ipv6-zones-range
 | tab
ZONE   RANGE
NAME   NAME    START                END
-----------------------------------------------------------
ZONE1  RANGE1  2606:ae00:c780:1     2606:ae00:c789:ffff
       RANGE2  2606:ae00:c790:0000  2606:ae00:c79f:ffff
```

```
admin@orchestrator[an-dbmaster]# database clearzoneinfo delete-zone clustername binding
zonename ZONE1
 2 documents deleted.
{'_id': 1, 'version': 137299, 'state': 'READY', 'zoneSharding': 'enabled', 'modifiedAt':
datetime.datetime(2022, 11, 23, 10, 27, 34, 498000)}
admin@orchestrator[an-dbmaster]#
```

database cluster

Create a MongoDB database sharded cluster.

Syntax

```
database cluster name sharded-cluster-master {true|false}
```

```
no database cluster name
```

Command Parameters

Table 34: Parameter Description

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

Command Mode

CONFIG

VNFs

All

Command Usage

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

Examples

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

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

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

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

Note This command is applicable only for MongoDB based sharding.

Syntax

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

Command Parameters

Table 35: Parameter Description

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

Command Mode

CONFIG

VNFs

All

Command Usage

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

Examples

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

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

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

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

Note This command is applicable only for MongoDB based sharding.

Syntax

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

Command Parameters

Table 36: 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* multi-db-collections *noOfShardsPerDB*

Used to add a MongoDB sharding configuration server process to the named database cluster.

Note This command is applicable only for application client based sharding.

Syntax

```
database cluster db-name mutli-db-collections noOfShardsPerDB
no database cluster db-name multi-db-collections
```

Command Parameters

Table 37: Parameter Description

Command Parameter	Description
DB Name	A short name describing the database 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.
noOfShardsPerDB	Number of shards created per database.

Command Mode

CONFIG

VNFs

All

Command Usage

Use the database cluster multi-db-cluster to create those number of shards per database.

Examples

The following is an example of enabling multi-db-collections to the “binding” cluster.

```
admin@orchestrator[master-hostname](config)# database cluster binding multi-db-collections
 2
```

database cluster *db-name* router *name*

Add a new MongoDB router to the named DB cluster.

Note This command is applicable only for MongoDB based sharding.

Syntax

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

Command Parameters

Table 38: Parameter Description

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

Command Mode

CONFIG

VNFs

All

Command Usage

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

Examples

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

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

database cluster *db-name* shard *name*

Add a new MongoDB shard to the named database cluster.

Syntax

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

Command Parameters

Table 39: 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 40: Parameter Description

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

Command Mode

CONFIG

VNFs

All

Command Usage

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

Examples

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

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

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

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

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

Syntax

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

Command Parameters

Table 41: 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* sharding-db *name*

Adds a MongoDB sharding configuration server process to the named database cluster.

Note This command is applicable only for application client based sharding.

Syntax

```
database cluster db-name sharding-db name address address
no database cluster db-name sharding-db name
```

Command Parameters

Table 42: Parameter Description

Command Parameter	Description
DB Name	A short name describing the database 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.
Name	A short description of the sharding database name.
address	The IPv4 or IPv6 address of the configuration server. This parameter does not accept FQDN address format.

Command Mode

CONFIG

VNFs

All

Command Usage

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

Examples

The following is an example of adding new sharding database to “binding” cluster.

```
admin@orchestrator[master-hostname](config)# database cluster binding sharding-db shdb-1
address 10.10.10.10
```

database cluster *db-name* sharding-db-seed *name*

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

Note This command is applicable only for application client based sharding.

Syntax

```
database cluster db-name sharding-db-seed name
```

Command Parameters

Table 43: Parameter Description

Command Parameter	Description
DB Name	A short name describing the database 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.
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 sharding-db-seed command to set the initial seed configuration server for boot-strapping the MongoDB replica set initialization process. This is required if a sharding database is set.

Examples

The following is an example of setting shdb-1 as the initial seed for a new sharding database to the "binding" cluster.

```
admin@orchestrator[master-hostname](config)# database cluster binding sharding-db-seed
shdb-1
```

database cluster *db-name* ipv6-zone-sharding

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 44: 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

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 45: 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 Prefix address for the particular range that can be from same physical location as PGW.
Pool Ending Address	The ending IPv6 Prefix 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
```

The following is the output for database cluster ipv6-zones-range command.

```
                    RANGE
NAME       ZONE NAME  NAME    START                END
-------------------------------------------------------------------------
binding    pune       range1  2003:3051:0000:0001  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 46: 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
```

database ipv6-zones clustername *CLUSTER-NAME* zonename *ZONE-NAME* rangename *RANGE-NAME* start *pool-start-IPv6-addr* end *pool-end-IPv6-addr*

Create zones for IPv6 shards based on IPv6 pools with ipv6 address validation. This allows the primary member of the replica set for an IPv6 address reside at the same physical location as the P-GW assigning addresses from the IPv6 pool. This results in local write (and read) for the IPv6 binding database.

This command supersedes the existing

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

CLI.

The **database ipv6-zones clustername** CLI adds IPv6 validation whereas the **database cluster db-name ipv6-zones-range** CLI does not have IPv6 validation.

Note You can create multiple ranges for each zone. Configure the IPv6 ranges only in the short format.

Syntax

database ipv6-zones clustername *CLUSTER-NAME* **zonename** *ZONE-NAME*
rangename *RANGE-NAME* **start** *pool-start-IPv6-addr* **end** *pool-end-IPv6-addr*

Command Parameters

Table 47: Parameter Description

Command Parameter	Description
Cluster Name	A short name that describes the DB cluster. Each application uses a set of predefined names and this name must match one of the application names. For example, DRA uses the name “binding” for storing binding and session records.
Zone Name	A short name that describes Zone name. This unique name identifies the zone that the shard configuration uses to map to zone.
Range name	A short name that describes the range within the zone. The starting IPv6 Prefix address for the particular range that can be from the same physical location as P-GW.

Command Parameter	Description
Pool Starting Address	The starting IPv6 Prefix address for the particular range that can be from the same physical location as P-GW.
Pool Ending Address	The ending IPv6 Prefix address for the particular range that can be from the same physical location as P-GW.

Command Mode

CONFIG

VNFs

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:

```
admin@orchestrator[site3-master-db-0](config)#
database ipv6-zones clustername imsi zonename
IMSIZONE2 rangename IMSIRANGE2 start 2606:ae00:bd80:0000:0000:0000:0000:0000
end 2606:ae00:bd80:0000:0000:0000:0000:0010
result SUCCESS
```

database delete all-bindings-sessions

Deletes the data belonging to given range and zone for all the bindings and sessions databases.

Syntax

```
database delete all-bindings-sessions <bindings-cluster-name> <sessions-cluster-name>
<zone-name> <start-address> <end-address>
```

Command Parameters

Table 48: Parameter Description

Command Parameter	Description
bindings-cluster-name	Specifies the bindings cluster name on which deletion jobs has to be performed.
sessions-cluster-name	Specifies the sessions cluster name on which deletion job has to be performed.

Command Parameter	Description
zone-name	Specifies the zone from which bindings have to be deleted. **Note** • If zone name is default, bindings records (all types and sessions) for the specified range data are deleted from all shards in database clusters. • If zone name is not default, bindings records (all types include sessions) for the specified range data are deleted from shards assigned to the specified zone.
start-address	Start address of IPv6 address range.
end-address	End address of IPv6 address range.

Command Mode

OPERATIONAL

VNFs

Binding

Command Usage

Use the `database delete all-bindings-sessions zone` command to delete IPv6 bindings and all the associated bindings for the specified address range from the specified zone.

Examples

The following example deletes IPv6 bindings and all the associated bindings from a specific zone:

```
database delete all-bindings-sessions imsi-msisdn session-ipv6-AB pune 7507:9903:1808:8000
 7507:9903:1808:8fff
```

The following example deletes IPv6 bindings and all the associated bindings from the default zone:

```
database delete all-bindings-sessions imsi-msisdn session-ipv6-AB default 7507:9903:1808:8000
 7507:9903:1808:8fff
```

database delete ipv6bindings

Deletes IPv6 bindings for the specified address range from the specified zone.

Syntax

```
database delete ipv6bindings <sessions-cluster-name> <zone-name> <start-address> <end-address>
```

Command Parameters

Table 49: Parameter Description

Command Parameter	Description
sessions-cluster-name	Specifies the sessions cluster name on which deletion job has to be performed.
zone-name	Specifies the zone from which bindings have to be deleted. **Note** • If zone name is default, IPv6 bindings for the range are deleted from all shards of database cluster. • If zone name is not default, IPv6 bindings for the range are deleted only from shards assigned to the specified zone.
start-address	Start address of IPv6 address range.
end-address	End address of IPv6 address range.

Command Mode

OPERATIONAL

VNFs

Binding

Command Usage

Use the `database delete ipv6bindings` command to delete IPv6 bindings for the specified address range from the specified zone.

Examples

The following example deletes IPv6 bindings from a specific zone:

```
database delete ipv6bindings session-ipv6-AB pune 7507:9903:1808:8000 7507:9903:1808:8fff
```

The following example deletes IPv6 bindings from the default zone:

```
database delete ipv6bindings session-ipv6-AB default 7507:9903:1808:8000 7507:9903:1808:8fff
```

database dwccheck

Used to check and set the defaultWriteConcern (w:1) on the databases.

Syntax

```
database dwccheck
```

Command Mode

OPERATIONAL

VNFs

Binding

Command Usage

Use the `database dwccheck` command to check and set the defaultWriteConcern on the databases.

Examples

The following is an example to check and set the defaultWriteConcern on the databases.

```
admin@orchestrator[site2-binding-master-1]# database dwccheck
Press "set" to check and set dwc on primary members or
Press "check" to only check dwc on all members.
(set/check) << check
DWC check for shard databases for all members is in Progress......
DWC check for shard databases for all members is Completed
DWC check for shardingDb databases for all members is in Progress......
DWC check for shardingDb databases for all members is Completed
DWC check for orchestrator and mongo-admin-db databases for all members is in Progress......
DWC check for orchestrator and mongo-admin-db databases for all members is Completed
admin@orchestrator[site2-binding-master-1]# database dwccheck
Press "set" to check and set dwc on primary members or
Press "check" to only check dwc on all members.
(set/check) << set
This is going to check and set required DWC on orchestrator,mongo-admin-db,shards and
shardingDb databases.
Press yes to continue
(yes/no) << yes
Please do not kill the terminal untill DWC check completes.Kindly check
logs(Path:/var/log/broadhop/dwc.log) for more info
DWC check for shard databases is in Progress......
DWC check for shard databases is Completed
DWC check for shardingDb databases is in Progress......
DWC check for shardingDb databases is Completed
DWC check for orchestrator and mongo-admin-db databases is in Progress......
DWC check for orchestrator and mongo-admin-db databases is Completed
admin@orchestrator[site2-binding-master-1]#
```

database fcvcheck

Used to check and set fcv on databases.

Syntax

```
database fcvcheck
```

Command Parameters

Table 50: Parameter Description

Command Parameter	Description
check	Check fcv on all members (primary/secondary).
set	Check and set fcv on primary member.

Command Mode

OPERATIONAL

VNFs

All

Command Usage

Use the database fcvcheck command to check and set fcv on databases.

Examples

Binding VNF (set): The following is an example to set fcv on primary member.

```
database fcvcheck

Press "set" to check and set fcv on primary members or
Press "check" to only check fcv on all members.
(set/check) << set
This is going to check and set required FCV on orchestrator,mongo-admin-db,shards and
shardingDb databases.
Press yes to continue
(yes/no) << yes
Please do not kill the terminal untill FCV check completes.Kindly check
logs(Path:/var/log/broadhop/fcv.log) for more info
FCV check for shard databases is in Progress......
FCV check for shard databases is Completed
FCV check for shardingDb databases is in Progress......
FCV check for shardingDb databases is Completed
FCV check for orchestrator and mongo-admin-db databases is in Progress......
FCV check for orchestrator and mongo-admin-db databases is Completed
```

Binding VNF (check): The following is an example to check fcv on all members.

```
database fcvcheck

Press "set" to check and set fcv on primary members or
Press "check" to only check fcv on all members.
(set/check) << check
FCV check for shard databases for all members is in Progress......
FCV check for shard databases for all members is Completed
FCV check for shardingDb databases for all members is in Progress......
FCV check for shardingDb databases for all members is Completed
FCV check for orchestrator and mongo-admin-db databases for all members is in Progress......
FCV check for orchestrator and mongo-admin-db databases for all members is Completed
```

DRA VNF (set): The following is an example to set fcv on primary member.

```
database fcvcheck
```

```
Press "set" to check and set fcv on primary members or
Press "check" to only check fcv on all members.
(set/check) << set
This is going to check and set required FCV on orchestrator and mongo-admin-db databases.
Press yes to continue
(yes/no) << yes
Please do not kill the terminal untill FCV check completes.Kindly check
logs(Path:/var/log/broadhop/fcv.log) for more info
FCV check for orchestrator and mongo-admin-db databases is in Progress......
FCV check for orchestrator and mongo-admin-db databases is Completed
```

DRA VNF (check): The following is an example to check fcv on all members.

```
database fcvcheck
```

```
Press "set" to check and set fcv on primary members or
Press "check" to only check fcv on all members.
(set/check) << check
FCV check for orchestrator and mongo-admin-db databases for all members is in Progress......
FCV check for orchestrator and mongo-admin-db databases for all members is Completed
```

database query

Fetches records for database in a specified cluster.

Note This feature has not been validated for all customer deployment scenarios. Please contact your Sales Account team for support.

Syntax

```
database query clustername cluster-name dbname db-name query query-to-run
method method max max-value start start-value
```

Command Parameters

Table 51: Parameter Description

Command Parameter	Description
cluster-name	Cluster name to obtain records from.
db-name	DB to obtain records from.
query-to-run	Input query inside double quotes.
Method	Method to perform on database.
max-value	Number of records to be limited. Default: 10.
start-value	Range of records to start from. Default: 0.

Command Mode

OPERATIONAL

VNFs

Binding

Command Usage

Use the database query command to fetch records for db in a specified cluster.

Examples

Example 1:

The following is an example of fetching 30 records starting from 10 index from an IPv6 database in a binding cluster for the query "{'uuid': {'$regex': 'vpas-system-1200'}}".

Default <= 5 records are displayed, all 30 records are saved in /data/config/Query.log.
admin@orchestrator[Binding-master]# database query clustername binding dbname IPV6 query "{'uuid': {'$regex': 'vpas-system-1200'}}" method find max 30 start 10

```
Press yes to continue
(yes/no) << yes
{'_id': '2507:9903:1808:0583',
 'fqdn': 'site-d-client-calipers21-gx.pcef.gx',
 'sessionid': 'kmanchan21411kunalmanchanda2',
 'srk': 'server.sitedstandalone',
 'staleBindingExpiryTime': datetime.datetime(2021, 5, 10, 13, 27, 24, 762000),
 'systemId': 'vpas-system-1',
 'ts': 1620048444762,
 'uuid': 'vpas-system-12002439730'}
{'_id': '2507:9903:1808:061a',
 'fqdn': 'site-d-client-calipers21-gx.pcef.gx',
 'sessionid': 'kmanchan21562kunalmanchanda2',
 'srk': 'server.sitedstandalone',
 'staleBindingExpiryTime': datetime.datetime(2021, 5, 10, 13, 27, 24, 763000),
 'systemId': 'vpas-system-1',
 'ts': 1620048444763,
 'uuid': 'vpas-system-12002439881'}
{'_id': '2507:9903:1808:065d',
 'fqdn': 'site-d-client-calipers21-gx.pcef.gx',
 'sessionid': 'kmanchan21629kunalmanchanda2',
 'srk': 'server.sitedstandalone',
 'staleBindingExpiryTime': datetime.datetime(2021, 5, 10, 13, 27, 24, 763000),
 'systemId': 'vpas-system-1',
 'ts': 1620048444763,
 'uuid': 'vpas-system-12002439948'}
{'_id': '2507:9903:1808:0694',
 'fqdn': 'site-d-client-calipers21-gx.pcef.gx',
 'sessionid': 'kmanchan21684kunalmanchanda2',
 'srk': 'server.sitedstandalone',
 'staleBindingExpiryTime': datetime.datetime(2021, 5, 10, 13, 27, 24, 764000),
 'systemId': 'vpas-system-1',
 'ts': 1620048444764,
 'uuid': 'vpas-system-12002440003'}
{'_id': '2507:9903:1808:06b7',
 'fqdn': 'site-d-client-calipers21-gx.pcef.gx',
 'sessionid': 'kmanchan21719kunalmanchanda2',
 'srk': 'server.sitedstandalone',
 'staleBindingExpiryTime': datetime.datetime(2021, 5, 10, 13, 27, 24, 766000),
```

```
 'systemId': 'vpas-system-1',
 'ts': 1620048444766,
 'uuid': 'vpas-system-12002440038'}
Default <= 5 records are displayed, Kindly check records in /data/config/Query.log based
on max value provided.
```

Example 2:

The following is an example for fetching maximum record value starting from 10 index from an IPv4 database in a binding cluster.

Default <= 5 records are displayed, all the records are saved in /data/config/Query.log.

admin@orchestrator[site2-binding-master-1]# database query clustername binding dbname IPV4 query "" method find max 5 start 10

```
Press yes to continue
(yes/no) << yes
{'_id': '2001:0000:0000:01d9',
 'fqdn': 'site-b-client-calipers21-gx.pcef.gx1',
 'sessionid': 'ClpGx0:192.168.31.213:4007:1608137498:0000000474',
 'srk': 'server.sitebstandalone',
 'staleBindingExpiryTime': datetime.datetime(2020, 12, 26, 17, 18, 14, 518000),
 'ts': 1608138406690,
 'uuid': 'site1-system552221945'}
...
Default <= 5 records are displayed, Kindly check records in /data/config/Query.log based
on max value provided.
admin@orchestrator[site2-binding-master-1]#
```

Query Example:

format: Query should be in a standard mongo db query format

query "{'key1': 'value1'}"

query "{'key1': 'value1', 'key2': 'value2'}"

Query Restrictions

Queries are allowed only for exact key matches that are specified in the Parameters Choices table. Any new key is allowed in query-based on requirement as a part of a patch. This ensures unnecessary processing on the shards if invalid/service impacting fields are mentioned in queries.

Adding New Query

For new query key support, update the `queries.jar` as part of a patch.

- **method**: [count, find].
- **method restrictions**: Other methods [update, insert, delete] are not allowed.
- **max**: [5,25000], default =5
- **start**: default = 0

database repair

Used to recover single/multiple/all shards and sharding database.

Attention

In HA deployment, CLI needs to be run on single site.

Logs (`/var/log/broadhop/shardrecovery.log`) should be checked after executing CLI.

The database repair script runs using the **db_user** instead of **cps user**, which is created as part of the installation or upgrade process.

Prerequisites

Ensure to execute these steps manually on each virtual machine to set up both password-based and *pem* file-based authentication for **db_user**:

1. Configure a password in all DB VM's.

   ```
   sudo passwd db_user
   ```

2. Generate public and private key using **ssh-keygen -t rsa** command. For example, use the sample configuration:

   ```
   cps@csdl2-bnd-master-0-1:~$ ssh-keygen -t rsa
   Generating public/private rsa key pair.
   Enter file in which to save the key (/home/cps/.ssh/id_rsa): db_user.pem
   Enter passphrase (empty for no passphrase):
   Enter same passphrase again:
   Your identification has been saved in db_user.pem
   Your public key has been saved in db_user.pem.pub
   The key fingerprint is:
   SHA256:DxcvHlz5TXzfUvAhICNjUP3Gh/C9UeW498aINfcbzcU cps@csdl2-bnd-master-0-1
   The key's randomart image is:
   +---[RSA 3072]----+
   |      .o=.o .....o|
   |       . ooo  ..B.|
   |           * = o B|
   |          . X = *+|
   |         S * o O E|
   |          = o + O=|
   |           o . ..B|
   |                .o|
   |                . |
   +----[SHA256]-----+
   cps@csdl2-bnd-master-0-1:~$
   ```

3. Copy **db_user.pem.pub** to all VM's in DB VNF:

   ```
   scp -i cps.pem db_user.pem.pub cps@<ip>:/home/cps
   ```

4. Enable the *pem* file based authentication. Follow these steps in each VM of DB VNF:

   ```
   sudo cp db_user.pem.pub /home/db_user
   sudo su
   chmod 777 /home/db_user/*;exit
   su db_user
   mkdir -p ~/.ssh
   ```

```
chmod 700 ~/.ssh
cat ~/db_user.pem.pub >> ~/.ssh/authorized_keys
```

5. Copy the private key to orchestrator under /data/keystore:

```
docker cp db_user.pem orchestrator:/data/keystore
```

Syntax

```
database repair <clustername> <shardname>
database repair <clustername> <shardname1> <shardname2> <shardname3>
database repair <clustername> All
database repair <clustername> sharddb
```

Command Parameters

Table 52: Parameter Description

Command Parameter	Description
clustername	Name of the cluster to which shard belongs.
shardname	Name of the shard to be recovered.
All	All shards in the cluster.
sharddb	Sharding database recovery.

Command Mode

OPERATIONAL

VNFs

Binding

Command Usage

Use the database repair commands to recover single/multiple/all shards and sharding database.

Examples

The following is an example to recover shard1 in binding cluster.

```
database repair binding shard1
```

The following is an example to recover shard1, shard2, shard3, and shard4 in binding cluster

```
database repair binding shard1 shard2 shard3 shard4
```

The following is an example to recover all shards in the binding cluster.

```
database repair All.
```

The following is an example to recover sharding database in the binding cluster

```
databse repair shard-db
```

database wiredTiger-Concurrent-Transactions get-transaction

Used to get the wire tiger concurrent read and write static transactions.

Syntax

```
database wiredTiger-Concurrent-Transactions get-transaction
```

Command Mode

Operational

VNFs

Binding

Command Usage

Use the command to get the wire tiger concurrent read and write static transactions.

Examples

Following is an example to get the wire tiger concurrent read and write static transactions.

```
admin@orchestrator[vpas-B1-bind-master-1]# database wiredTiger-Concurrent-Transactions
get-transaction
CLUSTER WRITE READ
imsi-msisdn : 128 8
session-ipv6 : 128 128
```

database wiredTiger-Concurrent-Transactions set-transaction dynamic

If you configure from the dynamic mode, the values reflect in mongo without a restart action. When the mongo process the restart action, the dynamic values are removed. While the mongo process start, it takes the static configuration.

Syntax

```
database wiredTiger-Concurrent-Transactions set-transaction dynamic cluster Cluster Name
Write Value  Read Value
```

Command Parameters

Table 53: Parameter Description

Command Parameter	Description
Cluster Name	Name of the DB cluster.

Command Parameter	Description
Write Value	Value for write transactions.
Read Value	Value for read transactions.

Command Mode

Operational

VNFs

Binding

Command Usage

Use the command to set the wire tiger transactions for read and write values to the mongo clusters dynamically.

Examples

Following is an example to set the wire tiger transactions for read and write values to the mongo clusters.

```
admin@orchestrator[vpas-B1-bind-master-1]# database wiredTiger-concurrent-transactions
set-transaction dynamic cluster imsi-msisdn write 9 read 9
Started checking cluster name
Starting script to set wiredTiger Concurrent Transactions to configure cluster imsi-msisdn
 writ :9 read: 9
Setting on Mongo Host and Port : 2606:ae00:3001:8311:172:16:244:8a:27020 imsi-msisdn
Setting on Mongo Host and Port : 2606:ae00:3001:8311:172:16:244:8c:27020 imsi-msisdn
Setting on Mongo Host and Port : 2606:ae00:3001:8311:172:16:244:2a:27020 imsi-msisdn
Setting on Mongo Host and Port : 2606:ae00:3001:8311:172:16:244:7c:27020 imsi-msisdn
Setting on Mongo Host and Port : 2606:ae00:3001:8311:172:16:244:7d:27020 imsi-msisdn
Setting on Mongo Host and Port : 2606:ae00:3001:8311:172:16:244:9c:27020 imsi-msisdn
Setting on Mongo Host and Port : 2606:ae00:3001:8311:172:16:244:9d:27020 imsi-msisdn
Setting on Mongo Host and Port : 2606:ae00:3001:8311:172:16:244:8b:27021 imsi-msisdn
Setting on Mongo Host and Port : 2606:ae00:3001:8311:172:16:244:2a:27021 imsi-msisdn
Setting on Mongo Host and Port : 2606:ae00:3001:8311:172:16:244:2b:27021 imsi-msisdn
Setting on Mongo Host and Port : 2606:ae00:3001:8311:172:16:244:7d:27021 imsi-msisdn
Setting on Mongo Host and Port : 2606:ae00:3001:8311:172:16:244:7c:27021 imsi-msisdn
Setting on Mongo Host and Port : 2606:ae00:3001:8311:172:16:244:9d:27021 imsi-msisdn
Setting on Mongo Host and Port : 2606:ae00:3001:8311:172:16:244:9c:27021 imsi-msisdn
Setting on Mongo Host and Port : 2606:ae00:3001:8311:172:16:244:8b:27022 imsi-msisdn
Setting on Mongo Host and Port : 2606:ae00:3001:8311:172:16:244:8c:27022 imsi-msisdn
Setting on Mongo Host and Port : 2606:ae00:3001:8311:172:16:244:2b:27022 imsi-msisdn
```

database wiredTiger-Concurrent-Transactions set-transaction static

If the user configure from static, the values are stored in consul. When the mongo process the restart action, the values reflect in the mongo.

Syntax

```
database wiredTiger-Concurrent-Transactions set-transaction static cluster Cluster Name
Write Value  Read Value
```

Command Mode

Operational

VNFs

Binding

Command Usage

Use the command to set the wire tiger transactions for read and write values to the mongo clusters.

Examples

Following is an example to set the wire tiger transactions for read and write values to the mongo clusters.

```
admin@orchestrator[vpas-B1-bind-master-1]# database wiredTiger-Concurrent-Transactions
set-transaction static cluster session-ipv6 write 200 read 8

Started checking cluster name

wiredTiger-Concurrent-Transactions configured to mongo. Changes will be applied after mongo
 restart.
```

db-authentication set-password database redis password

To set the Redis authentication password.

Syntax

```
db-authentication set-password database redis password <clear text password>
```

Command Parameters

Table 54: Parameter Description

Command Parameter	Description
<clear text password>	A clear text password used for Redis authentication. The password is stored in consul datastore in encrypted format. The Redis password is stored in consul datastore in encrypted format and synchronized to `draTopology.ini` which is used by dra-endpoint application. The service reads the password from consul datastore and password is updated in the console data store with encrypted password. Data store and `draTopology.ini` format: `redis/config/password:<encrypted password>`

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use the database authentication command to set the Redis password which is used to access Redis data store.

Examples

The following is an example to set the Redis authentication password:

```
admin@orchestrator[master-0m]# db-authentication set-password database redis password
Value for 'password' (<string>): ***********
result SUCCESS
```

db-authentication show-password database redis

To display the encrypted redis password.

Syntax

```
db-authentication show-password database redis
```

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use the database authentication command to display the Redis password.

Examples

The following is an example to display the Redis authentication password:

```
admin@orchestrator[master-0m]# db-authentication show-password database redis
result
result PASSWORD : 72261348A44594381D2E84ADDD1E6D9A
```

db-authentication remove-password database redis

To remove Redis authentication password.

Syntax

```
db-authentication remove-password database redis current-password password
```

Command Parameters

Table 55: Parameter Description

Command Parameter	Description
password	Clear text password to be removed on redis need to be provided.

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use the `db-authentication` command to remove Redis authentication password.

Examples

The following is an example to remove Redis authentication password:

```
admin@orchestrator[master-0m]# db-authentication remove-password database redis
Value for 'current-password' (<string>): *******
result SUCCESS
```

db-authentication show-password database mongo

To display the encrypted MongoDB password.

Syntax

```
db-authentication show-password database mongo
```

Command Mode

OPERATIONAL

VNFs

All

Command Usage

Use the database authentication command to display the MongoDB password.

Examples

The following is an example:

```
scheduler# db-authentication show-password database mongo
result
adminuser: 3300901EA069E81CE29D4F77DE3C85FA
```

db-authentication set-password database mongo password

Used to create users (adminuser and backupuser) with credentials in the MongoDB.

Syntax

```
db-authentication set-password database mongo password <password>
```

Command Parameters

Table 56: Parameter Description

Command Parameter	Description
password	Clear text password to be set on Mongo DB need to be provided.

Command Mode

OPERATIONAL

VNFs

DRA and Binding

Command Usage

This command is used to create users (adminuser and backupuser) with credentials in the MongoDB.

Examples

The following is an example to create users with credentials:

```
admin@orchestrator[binding-master]# db-authentication set-password database mongo password
Value for 'password' (<string>): ******
result SUCCESS
```

db-authentication remove-password database mongo

Used to remove the users (admin user and backup user) and password from all the databases.

Syntax

```
db-authentication remove-password database mongo current-password <password>
```

Command Parameters

Table 57: Parameter Description

Command Parameter	Description
password	Clear text password to be removed on MongoDB need to be provided.

Command Mode

OPERATIONAL

VNFs

DRA and Binding

Command Usage

Use to remove users and password from the mongo databases. Before using this command the database should be in transition authentication state and after this command rolling restart is mandatory.

Examples

The following is an example to remove-password in mongo database:

```
admin@orchestrator[binding-master]# db-authentication remove-password database mongo
Value for 'current-password' (<string>): *******
result SUCCESS
```

db-authentication change-password database mongo

Used to change the admin user password in all the databases.

Syntax

```
db-authentication change-password database mongo current-password <current pasword>
new-password <New password> user adminuser
```

Command Parameters

Table 58: Parameter Description

Command Parameter	Description
Current Password	Current password set in MongoDB.
New Password	New password to be set in MongoDB.

Command Mode

OPERATIONAL

VNFs

DRA and Binding

Command Usage

This command change password of adminuser in all the MongoDB.

Examples

The following is an example to change-password in MongoDB:

```
admin@orchestrator[binding-master]# db-authentication change-password database mongo user
adminuser
Value for 'current-password' (<string>): ******
Value for 'new-password' (<string>): ********
result SUCCESS
```

db-authentication sync-password database mongo

Used to synchronize the backup user password same as admin user password..

Syntax

```
db-authentication sync-password database mongo
```

Command Mode

OPERATIONAL

VNFs

DRA and Binding

Command Usage

This command is used to sync password in all the MongoDB.

Examples

The following is an example to synchronize the passwords:

```
admin@orchestrator[binding-master]# db-authentication sync-password database mongo
result
SUCCESS : Mongo password sync successful
```

db-authentication enable-transition-auth database mongo

Used to configure the transition authentication parameter. Rolling restart should be executed after this command.

Syntax

```
db-authentication enable-transition-auth database mongo
```

Command Mode

OPERATIONAL

VNFs

Binding

Command Usage

Use this command to configure the transition authentication parameter.

Examples

The following is an example to configure the transition authentication parameter:

```
admin@orchestrator[binding-master]# db-authentication enable-transition-auth database mongo
```

db-authentication disable-transition-auth database mongo

Used to remove the transition authentication parameter. Rolling restart should be done after this command.

Syntax

```
db-authentication disable-transition-auth database mongo
```

Command Mode

OPERATIONAL

VNFs

Binding

Command Usage

Use this command to remove the transition authentication parameter.

Examples

The following is an example to disable transition authorization in MongoDB:

```
admin@orchestrator[binding-master]# db-authentication disable-transition-auth database mongo
```

db-authentication rolling-restart database mongo

Used to restart all the database instances where primary members is followed by secondary members.

Syntax

```
db-authentication rolling-restart database mongo
```

Command Mode

OPERATIONAL

VNFs

Binding

Command Usage

Use this command to restart all the database instances where primary members will be followed by secondary members.

Examples

The following is an example to restart all the database instances:

```
admin@orchestrator[binding-master]# db-authentication rolling-restart database mongo
```

db-authentication rolling-restart-parallel database mongo

Used to restart multiple database instances in parallel without affecting the availability of DB cluster.

Syntax

```
db-authentication rolling-restart-parallel database mongo
```

Command Mode

OPERATIONAL

VNFs

Binding

Command Usage

Use `db-authentication rolling-restart-parallel database mongo` command to restart multiple database instances in parallel without affecting the availability of DB cluster.

Note `db-authentication rolling-restart-parallel database mongo` command is dependent on `show database parallel-upgrade-plan`.

If `show database parallel-upgrade-plan` does not provide any output, then do not use `db-authentication rolling-restart-parallel database mongo`. Instead use `db-authentication rolling-restart database mongo` command.

Example: If `show database parallel-upgrade-plan` displays the following output:

```
admin@orchestrator[an-dbmaster]# show database parallel-upgrade-plan
BATCH  MODULE          HOST            ADDRESS
-----------------------------------------------------
1      mongo-node-101  an-dbmaster     192.168.11.40
1      mongo-node-102  an-dbcontrol-0  192.168.11.41
1      mongo-node-103  an-dbcontrol-1  192.168.11.42
2      mongo-node-104  an-pers-db-0    192.168.11.43
3      mongo-node-105  an-pers-db-1    192.168.11.44
```

then, `db-authentication rolling-restart-parallel database mongo` combines 101, 102 and 103 in batch 1 and restarts all of them at the same time.

After batch 1, node 104 from batch 2 is restarted followed by node 105 (from batch 3). So, all the nodes from same batch are restarted in parallel. However nodes from different batch are restarted in sequential manner.

A batch_interval parameter can be added as follows:

```
admin@orchestrator[an-master]# db-authentication rolling-restart-parallel batch_interval 8

database mongo
```

where, batch_interval is an integer and accepts range between 8 to 60. By default, the value is 10. It represents the delay duration in seconds between processing of 2 subsequent batches. After executing this command, batch-wise status can be tracked using `db-authentication rolling-restart-parallel-status database mongo` command.

Examples

The following is an example to trigger a parallel restart for mongo-nodes.

```
db-authentication rolling-restart-parallel database mongo
```

db-authentication rolling-restart-parallel-status database mongo

Used to track the status of rolling-restart-parallel command.

Syntax

```
db-authentication rolling-restart-parallel-status database mongo
```

Command Mode

OPERATIONAL

VNFs

Binding

Command Usage

Use `db-authentication rolling-restart-parallel-status database mongo` command to track the status of rolling-restart-parallel command.

Examples

The following example shows which batch is completed out of total batches.

```
admin@orchestrator[an-dbmaster]# db-authentication rolling-restart-parallel-status database
 mongo
result Still in progress...Batch 1 out of total 3 is completed at 2019-12-10T23:16:25.799
admin@orchestrator[an-dbmaster]# db-authentication rolling-restart-parallel-status database
 mongo
result Still in progress...Batch 2 out of total 3 is completed at 2019-12-10T23:16:37.656
admin@orchestrator[an-dbmaster]# db-authentication rolling-restart-parallel-status database
 mongo
result Parallel rolling restart completed!! Batch 3 out of total 3 got completed at
2019-12-10T23:16:49.844
admin@orchestrator[an-dbmaster]# db-authentication rolling-restart-parallel-status database
 mongo
result
Parallel Rolling Restart: Not Scheduled/Completed/Just triggered
admin@orchestrator[an-dbmaster]#
```

db-authentication rolling-restart-status database mongo

Used to display the status of rolling restart as in-progress or completed.

Syntax

```
db-authentication rolling-restart-status database mongo
```

Command Mode

OPERATIONAL

VNFs

Binding

Command Usage

Use this command to display the status of rolling restart as in-progress or completed.

Examples

The following is an example to display the status of rolling restart:

```
admin@orchestrator[binding-master]# db-authentication rolling-restart-status database mongo
result
Rolling Restart: In Progress ...
```

db-encryption enable

To enable TLS in the DB and DRA VNF, set the TLS mode to allowTLS in the Consul server. Then, restart the MongoDB servers using the following command:

db-authentication rolling-restart-parallel database mongo

Prerequisites:

- All sites must be upgraded to the latest version (25.2) before this feature can be enabled.
- Enable TLS on the DB VNF as the initial step, then proceed to the DRA VNF.
- Enabling, disabling, or mode changes requires a database restart that must be performed using the **rolling-restart-parallel** command.
- Similarly, restarting the Binding app on the DRA VNF is required when enabling or disabling TLS.
- Before enabling TLS on the DRA VNF, ensure that TLS is enabled for all database members.
- Before enabling TLS on the DRA VNF, ensure that TLS is enabled for all database members. Before enabling TLS on the application side, ensure that TLS is enabled on all MongoDB replica set members across all sites. Because the MongoDB replica set is shared across all sites, the DRA connects to all MongoDB members from every site. If TLS is not enabled on any site's MongoDB, the DRA will fail to connect to those MongoDB instances using TLS.
- The **db-encryption enable** CLI command functions only after the import CLI command has been executed.
- Certificate creation supports passwords that contain more than five characters.
- When certificates expire or new certificates are imported on top of existing ones, restart the containers connected to MongoDB to update the cached certificates. Restarting the database and mongo-monitor-s/mongo-status containers using rolling-restart-parallel is required for new certificates to take effect. Similarly, you must restart the binding app on the DRA side when new certificates are installed.

Syntax

db-encryption enable *{ false | true }*

Command Mode

OPERATIONAL

VNFs

All

Examples

The following is an example to enable the TLS mode:

```
db-encryption enable true
```

db-encryption import

Imports the necessary certificates in DB and DRA VNF into the required containers such as mongo-s1, mongo-status, mongo-monitor and orchestrator.

Note Copy required certificates to `/data/pemKey` path inside orchestrator container.

Syntax

`db-encryption import` *`ca_pem_key_path ca_pem_path server_key_path server_crt_path`*

Command Mode

OPERATIONAL

VNFs

All

Examples

The following is an example to import the neecessary certificates into the container:

```
db-encryption import /data/pemKey/ca_key.pem /data/pemKey/ca.pem /data/pemKey/server.key
/data/pemKey/server.crt
```

db-encryption mode set

Changes the TLS mode from allowTLS to preferTLS or requireTLS, by storing the new mode in the Consul server. No MongoDB server restart is required after changing the mode.

Note Recommended mode is allowTLS. This mode supports both TLS and TCP connections and provides flexibility for mixed client environments.

Testing mode is requireTLS. Use this mode only for specific testing scenarios that require strict TLS enforcement. This mode blocks all TCP connections and permits only TLS-encrypted connections to MongoDB server

Important considerations:

- requireTLS impact: This mode completely restricts TCP connectivity.
- Production safety: The allowTLS mode ensures backward compatibility and prevents connection failures during gradual TLS adoption.

Syntax

db-encryption mode set *{ allowTLS | preferTLS | requireTLS }*

Command Mode

OPERATIONAL

VNFs

All

Usage Guidelines

This command sets the TLS mode for all MongoDB members specified in the configuration. You must run this command at each site. According to MongoDB documentation, the user must transition through the modes in sequential order: allowTLS, then preferTLS, then requireTLS, without skipping or changing the order. Verify each mode transition before starting the next phase:

- Execute the **db-encryption mode set preferTLS** CLI command on all four Database sites one at a time. Then,verify configuration using **db-encryption mode get** command to confirm that preferTLS mode is active across all sites.
- Execute the **db-encryption mode set requireTLS CLI** command on all four Database, one at a time. Then, verify configuration using **db-encryption mode get** command to confirm that requireTLS mode is active and enforced across all sites.
- Perform a rolling-restart-parallel on all four Database sites, one at a time.
- TLS cannot be disabled directly from requireTLS mode. MongoDB requires sequential step-downs in this order: requireTLS, then preferTLS, then allowTLS, and finally disabled.
  - RequireTLS to PreferTLS : o transition from requireTLS to preferTLS, execute **db-encryption enable false** to iniitiate the switch:
    - Check the status using **show database db-encryption-status**,
    - Prform a rolling restart of MongoDB processes, and
    - Verify preferTLS mode using **db-encryption mode get**command.,
    - Perform these steps for each database sites individually.
  - PreferTLS to AllowTLS : To transition from preferTLS to allowTLS, execute **db-encryption enable false**'to initiate the switch:
    - Perform a rolling restart of MongoDB processes,
    - Verify allowTLS mode using **db-encryption mode get** command.
    - Complete these actions for each database site individually.
  - Complete the TLS disablement. To complete TLS disablement, first ensure that:
    - TLS is disabled on DRA before making changes to the Database,
    - Execute **db-encryption enable false** to transition from allowTLS to disabled mode,
    - Perform a rolling restart of MongoDB processes, and

- Verify disabled mode using **db-encryption mode get** command. Perform these actions for each database sites individually.

Examples

The following is an example to change the TLS mode to preferTLS:

```
db-encryption mode set preferTLS
```

db-encryption mode get

Retrieves the current TLS mode in DB VNF for all MongoDB instances and compares it with the configured mode in consul.

Syntax

db-encryption mode get

Command Mode

OPERATIONAL

VNFs

All

Examples

The following is an example to get the current TLS mode:

```
db-encryption mode get
```

db-encryption sync certs

Ensures all nodes in the cluster have the same TLS certificates by copying them to any missing nodes.

Syntax

db-encryption sync certs

Command Mode

OPERATIONAL

VNFs

All

Examples

The following is an example synchronize certificates across the cluster:

```
db-encryption sync certs
```

db-encryption sync mode

Applies the configured TLS mode from consul to all MongoDB instances in the cluster, ensuring consistency.

Syntax

db-encryption sync mode

Command Mode

OPERATIONAL

VNFs

All

Examples

The following is an example to synchronize TLS mode across the cluster:

```
db-encryption sync mode
```

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 59: Parameter Description

Command Parameter	Description
ipv4	Connect to the IPv4 binding database.
ipv6	Connect to the IPv6 binding database.
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 collect-db-logs-advanced collect

Used to collect mongod logs from specified VMs based on the start and end timestamps.

You can also add the maximum storage capacity of logs to be allowed as input. Once the maximum capacity is reached, the log collection stops.

Note The log collection is limited to 15 days. If you need logs beyond 15, you must login to VM directly to pull the logs.

Syntax

```
debug collect-db-logs-advanced collect <max-allowed-log-size-in-gb> <start-time> <end-time>
 [VM-names]
```

Command Parameters

Table 60: Parameter Description

Command Parameter	Description
max-allowed-log-size-in-gb	Maximum size of the logs to be collected.
start-time	Specify the start time to start collecting the logs.
end-time	Specify the end time to end collecting the logs.
VM-names (Optional)	Docker engine VM names to be mentioned with space in between. If the VM names are not specified, the logs are collected for all the VMs in VNF.

Command Mode

Operational

VNFs

DRA Binding

Command Usage

Use this command to collect mongod logs from specified binding VNF VMs based on the start and end timestamps.

Output files from this command can be accessed using the following link:

https://<*MasterVM*>/orchestrator/downloads/debug/consolidated/consolidated-db-logs/

Examples

The following is an example:

```
debug collect-db-logs-advanced collect 4 2020-07-14T23:30:09 2020-07-15T04:15:20 VM-1 VM-2
```

Output file: `consolidated-db-logs_<StartDate>_<EndDate>.tar.gz`

debug collect-db-logs-advanced scan

Used to create a single consolidated log file of all MongoDB logs collected from different VMs based on start and end timestamps.

Before executing `debug collect-db-logs-advanced scan` command, you need to execute `collect` command which pulls all the logs from different VMs into `tar.gz`.

Note This command allows you to input timestamps in maximum of 6 hours time interval.

Currently, this command expects `tar.gz` file to be present in the respective storage location and creates `consolidated-log-output` in same place.

Syntax

```
debug collect-db-logs-advanced scan <start-time> <end-time> [VM-names]
```

Command Parameters

Table 61: Parameter Description

Command Parameter	Description
start-time	Specify the start time to start scanning the logs.
end-time	Specify the end time to end scanning the logs.
VM-names (Optional)	Docker engine VM names to be mentioned with space in between. If the VM names are not specified, the logs are collected for all the VMs in VNF.

Command Mode

Operational

VNFs

DRA Binding

Command Usage

Use this command to scan the MongoDB logs collected from different binding VMs based on start and end timestamps.

Output files from this command can be accessed using the following link:

https://<*MasterVM*>/orchestrator/downloads/debug/consolidated/consolidated-db-logs/

Examples

The following is an example:

```
debug collect-db-logs-advanced scan 2020-07-14T23:30:09 2020-07-15T04:15:20 VM-1 VM-2
```

Output file: `consolidated-logs-output`

debug log collect

Used to gather various logs to support troubleshooting.

Syntax

```
debug log collect [ docker { all all | vmname name  } ]
[ journalctl { all all | vmname name  } ]
[ pb { all all | vmname name  } ]
[ qns { all all | vmname name  } ]
[ tech  ]
[ top { all all | vmname name  } ]
[ mongo { all all | vmname name  } ]
```

Command Parameters

Table 62: Parameter Description

Command Parameter	Description
VM-names (Optional)	Docker engine, journalctl, qns, mongo DB VM names to be mentioned with space in between. If the VM names are not specified, the logs are collected for all the VMs in VNF.

Command Mode

Operational

VNFs

DRA

Command Usage

Use this command to gather various logs to support troubleshooting.

debug log collect heapdump containername <name>

Considers diameter-endpoint container name as input and collects heapdump outputs.

Examples

The following is an example:

```
debug log collect heapdump containername diameter-endpoint-s104
Collection of heapdump completed
```

debug log collect threaddump containername <name>

Considers diameter-endpoint container name as input and collects a set of threadump outputs.

Examples

The following is an example:

```
debug log collect threaddump containername diameter-endpoint-s104
Collection of threaddump completed
```

debug packet-capture gather

Gathers all running packet captures.

Syntax

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

Command Parameters

Table 63: 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 64: 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 65: 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 exec

Used to support executing specific command on specific or all the containers.

Syntax

`docker exec <container-name> <command>`

Command Parameters

Table 66: Parameter Description

Command Parameter	Description
container-name	Specifies the container-name (prefix or full-name). Enter the complete name for running the command in all the containers.
command	The command that needs to be executed on the containers.

Command Mode

Operational

VNFs

All

Command Usage

Use `docker exec <container-name> <command>` to take container-name and command as an argument. Container-name can be prefix or full name. If the command is having space then it should be provided between double quotes.

Examples

The following example shows sample commands:

Example 1: Stop the db-monitor process in mongo-monitor containers.

```
docker exec mongo-mon "supervisorctl stop db-monitor"
```

Example 2: Get the supervisorctl status from all the containers.

```
docker exec all "supervisorctl status"
```

docker repair

Used to remove mongo-s running containers from VMs.

Note This command must be executed in Maintenance Window (MW).

Syntax

```
docker repair <prefix> <VM-1 VM-2 ... VM-n>
```

Command Parameters

Table 67: Parameter Description

Command Parameter	Description
prefix	Container name to be removed. **Note** Currently, only mongo-s prefix is supported.
--no-prompt	Used to force repair.
VMs	Specify engine node name. You can get the VM names using `show docker engine` command.
all <module-name>	Used to remove all the containers which contain module-name as mongo-node and prefix as mongo-s. It won't remove mongo-monitor containers.

Command Mode

Operational

VNFs

All

Command Usage

Use this command to remove the mongo-s containers from VMs to clear the high usage of tmpfs file system memory. In case if any mongo-s container fails to come up or mongod inside it doesn't come up with healthy state then the entire repair operation is aborted.

Examples

The following example shows sample commands:

Example 1: Remove mongo-s container from a single VM with user prompt.

```
docker repair mongo-s control-binding-0
Are you sure to repair this mongo-s102 (y/n)? y
mongo-s102
Checking health status for mongo-s102.
Healthy Check Status for mongo-s102 = true
```

Example 2: Remove mongo-s container from multiple VMs with user prompt.

```
docker repair mongo-s control-binding-0 control-binding-1
Are you sure to repair this mongo-s102 (y/n)? y
mongo-s102
Checking health status for mongo-s102.
Healthy Check Status for mongo-s102 = true
Are you sure to repair this mongo-s103 (y/n)? y
mongo-s103
Checking health status for mongo-s103.
Healthy Check Status for mongo-s103 = true
```

Example3: Remove mongo-s container from multiple VMs without user prompt.

```
docker repair mongo-s --no-prompt control-binding-0 control-binding-1
mongo-s102
Checking health status for mongo-s102.
Healthy Check Status for mongo-s102 = true
mongo-s103
Checking health status for mongo-s103.
Healthy Check Status for mongo-s103 = true
```

Example 4: Remove all the mongo-s containers from the module-name with prefix mongo-s with user prompt.

```
docker repair mongo-s all mongo-node
Are you sure to repair this mongo-s101 (y/n)? y
mongo-s101
Checking health status for mongo-s101.
Healthy Check Status for mongo-s101 = true
Are you sure to repair this mongo-s102 (y/n)? y
mongo-s102
Checking health status for mongo-s102.
Healthy Check Status for mongo-s102 = true
Are you sure to repair this mongo-s103 (y/n)? y
mongo-s103
Checking health status for mongo-s103.
Healthy Check Status for mongo-s103 = true
Are you sure to repair this mongo-s104 (y/n)? y
mongo-s104
Checking health status for mongo-s104.
Healthy Check Status for mongo-s104 = true
Are you sure to repair this mongo-s105 (y/n)? y
```

```
mongo-s105
Checking health status for mongo-s105.
Healthy Check Status for mongo-s105 = true
Are you sure to repair this mongo-s106 (y/n)? y
mongo-s106
Checking health status for mongo-s106.
Healthy Check Status for mongo-s106 = true
Are you sure to repair this mongo-s107 (y/n)? y
mongo-s107
Checking health status for mongo-s107.
Healthy Check Status for mongo-s107 = true
```

Example 5: Remove all the mongo-s containers from the module-name with prefix mongo-s without user prompt.

```
docker repair mongo-s --no-prompt  all mongo-node
mongo-s101
Checking health status for mongo-s101.
Healthy Check Status for mongo-s101 = true
mongo-s102
Checking health status for mongo-s102.
Healthy Check Status for mongo-s102 = true
mongo-s103
Checking health status for mongo-s103.
Healthy Check Status for mongo-s103 = true
mongo-s104
Checking health status for mongo-s104.
Healthy Check Status for mongo-s104 = true
mongo-s105
Checking health status for mongo-s105.
Healthy Check Status for mongo-s105 = true
mongo-s106
Checking health status for mongo-s106.
Healthy Check Status for mongo-s106 = true
mongo-s107
Checking health status for mongo-s107.
Healthy Check Status for mongo-s107 = true
```

docker restart

Restarts a docker service that is currently running.

Syntax

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

Command Parameters

Table 68: 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.

docker start

Starts Diameter application container.

Syntax

`docker start <container-id>`

Command Parameters

Table 69: Parameter Description

Command Parameter	Description
container-id	Diameter application container name

Command Mode

OPERATIONAL

VNFs

DRA

docker stop

Stops Diameter application container.

Syntax

`docker stop <container-id>`

Command Parameters

Table 70: Parameter Description

Command Parameter	Description
container-id	Diameter application container name

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

This command ensures the following tasks are completed before the container is stopped:

- the required DPR messages are sent out to all connected peers
- VIP moves to another director

dra-distributor balance connection

Used to audit peer connections with the provided service name.

Syntax

```
dra-distributor balance connection <cluster-name> <service-name> audit
```

Command Parameters

Table 71: Parameter Description

Command Parameter	Description
cluster-name	Cluster name of the distributor service.
service-name	Service name of the floating IP address.
audit	Displays the number of connections per director based on the service name. Used to verify the total active peer connections on each diameter-endpoint containers and determines whether the connections are balanced or unbalanced between the containers.

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

This command is used to audit the peer connections.

Syntax

```
dra-distributor balance connection <cluster-name> <service-name>
```

Command Parameters

Table 72: Parameter Description

Command Parameter	Description
cluster-name	Cluster name of the distributor service.
service-name	Service name of the floating IP address.

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

This command checks the balancing and determines if connections need to be balanced. If the connections are unbalanced, it allows user to balance the connections.

Example

```
admin@orchestrator[vpas-A-dra-master-0]# dra-distributor balance connection client Gx-PCRFA
 audit
================================================================================
Total Directors                   8
Total Weight                      8
Total Active Connections          184
Connection Per Weight             23.0
================================================================================

Real-Server                                              Weight  Active  Expected
                                                                 Conn    Conn
172.16.XX.YY:3868 (diameter-endpoint-s104)               1       23      23.0
172.16.XX.YY:3868 (diameter-endpoint-s105)               1       23      23.0
172.16.XX.YY:3868 (diameter-endpoint-s106)               1       23      23.0
172.16.XX.YY:3868 (diameter-endpoint-s107)               1       23      23.0
172.16.XX.YY:3868 (diameter-endpoint-s108)               1       23      23.0
172.16.XX.YY:3868 (diameter-endpoint-s109)               1       23      23.0
172.16.XX.YY:3868 (diameter-endpoint-s110)               1       23      23.0
172.16.XX.YY:3868 (diameter-endpoint-s111)               1       23      23.0
================================================================================

Connections are properly distributed
```

dra-distributor balance traffic

Used to audit per director's TPS with the provided service name.

Syntax

```
dra-distributor balance traffic <cluster-name> <service-name> <threshold> <margin> audit
```

Command Parameters

Table 73: Parameter Description

Command Parameter	Description
cluster-name	Cluster name of the distributor service.
service-name	Service name of the floating IP address.
threshold	Threshold value. Balance traffic when any director traffic exceeds this threshold. **Note** Currently, threshold attribute is not considered for this command. It's for experimental purpose only.
margin	Traffic margin value. Balance traffic when any director traffic is not in the range from (Threshold-Margin) to (Threshold+Margin). **Note** Currently, margin attribute is not considered for this command. It's for experimental purpose only.
audit	Displays the per director TPS based on service name. Audits whether the existing traffic is distributed or not-distributed equally among directors.

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

This command is used to view per director's traffic to VIPs.

Example

```
admin@orchestrator[vpas-A-dra-master-0]# dra-distributor balance traffic client Sy-OCSA 100
 122 audit
Peer disconnect is sensitive operation, so please re-authentication
Enter The Admin Role User Name [default:admin]:
Enter Password:
=========================================================
```

```
Real-Server Active Traffic
Conn
diameter-endpoint-s104(172.16.XX.YY) 1 1224 *
diameter-endpoint-s105(172.16.XX.YY) 1 1211 *
diameter-endpoint-s106(172.16.XX.YY) 1 1196 *
diameter-endpoint-s107(172.16.XX.YY) 1 1193 *
================================================================================
Total Directors 4
Total Traffic 4824
Traffic Per Director 1206
================================================================================
Traffic of all directors between 1084 and 1328
Traffic are properly distributed
```

dra migration

enable-migration

Enable migration handling for moving from mongo-sharded database to application-sharded database.

Syntax

```
dra migration enable-migration true
```

```
no dra migration enable-migration
```

Command Mode

CONFIG

VNFs

DRA VNF

Command Usage

Enable handling of database migration. If binding record is not found in primary database (default, application-sharded database cluster) then the binding lookup is done in secondary database (default, mongo-sharded database cluster).

Examples

The following is an example:

```
admin@orchestrator[master-0](config)# dra migration enable-migration true
```

enable-mongo-sharded-db-as-primary-db

Mongo-sharded database is considered as primary database during migration handling.

Syntax

```
dra migration enable-mongo-sharded-db-as-primary-db [true|false]
no dra migration enable-mongo-sharded-db-as-primary-db
```

Command Mode

CONFIG

VNFs

DRA VNF

Command Usage

Make mongo-sharded database as the primary database for binding lookup (lookup bindings in mongo-sharded database first and if the binding record is not found then the binding is lookup in application-sharded database).

Note By default, application-sharded database is considered as primary database.

Examples

The following is an example:

```
admin@orchestrator[master-0](config)# dra migration enable-mongo-sharded-db-as-primary-db
true
```

enable-skipping-probe-message-binding-lookup

Skip binding lookup in secondary database for probe/dummy AAR messages.

Syntax

```
dra migration enable-skipping-probe-message-binding-lookup [true|false]
no dra migration enable-skipping-probe-message-binding-lookup
```

Command Mode

CONFIG

VNFs

DRA VNF

Command Usage

Enable skipping binding lookup in secondary database for probe/dummy Rx AAR messages (sent by PCRF as part of binding database health check).

Examples

The following is an example:

```
admin@orchestrator[master-0](config)# dra migration
enable-skipping-probe-message-binding-lookup true
```

dra policy-builder-must-plugins plugins-name

To configure a bare minimum essential plugin that is considered as mandatory to run the traffic properly. These plugins will be checked during the import process for the validation.

Syntax

dra policy-builder-must-plugins plugins-name DRAConfiguration threadingConfiguration customerReferenceDataConfiguration asyncThreadingConfiguration

Command Parameters

Table 74: Parameter Description

Command Parameter	Description
xxxx	

Command Mode

CONFIG

VNFs

DRA

Command Usage

Use this CLI command to configure a bare minimum essential plugin that is considered as mandatory to run the traffic properly.

If these plugins are not configured, then default four plugins such as **DRAConfiguration**, **threadingConfiguration**, **customerReferenceDataConfiguration**, and **asyncThreadingConfiguration** are considered.

Once these plugins configuration is done, system checks the presence of these configured plugins in zip or in System json. If any of the required plugin is not present, then API throws error and does not move forward.

Examples

The following is an example:

dra set-ratelimit topology-api 150.

dra jvm zulu enable

Used to switch the JVM memory management from Zing to Zulu.

Prerequisite

You must place the *cps.pem* file on the orchestrator container under `/data/keystore`

Note The **dra jvm zulu enable** command must be executed in the Maintenance Window (MW).

Syntax

```
dra jvm zulu enable jvm-memory  Memory Size in GB
        dra jvm status
```

Command Parameters

Table 75: Parameter Description

Command Parameter	Description
Zulu	The name of the service that manages the application.
JVM memory	Heap Size for the Zulu memory allocation.
Status	Display the status on the enabled JVM application.

Command Mode

OPERATIONAL

VNFs

DRA VNF

Command Usage

Use this **dra jvm** command to switch between zulu and zing.

Examples

Login to the Orchestrator container CLI and run the below cmd's to enable zulu.

Note To enable Zulu, the recommended heap memory is 16GB and Heap Memory can be increase/decrease based on the requirement.

```
admin@orchestrator[dra1-sys04-master-0]# dra jvm zulu enable jvm-memory 16
Are you sure to enable Zulu JVM (y/n)?
Enabling Zulu service on binding-s108.
Success! Data written to: zing/enabled
Success! Data written to: jvm/memory
Enabling Zulu service on binding-s109.
Success! Data written to: zing/enabled
Success! Data written to: jvm/memory
Enabling Zulu service on diameter-endpoint-s104.
Success! Data written to: zing/enabled
Success! Data written to: jvm/memory
Enabling Zulu service on diameter-endpoint-s105.
Success! Data written to: zing/enabled
```

```
Success! Data written to: jvm/memory
Enabling of Zulu JVM will restart the App Service,
Are you sure to proceed with App Service Restart (y/n)?
Executing Zulu Tasks
Stopping app service on binding-s108.
app: stopped
Stopping app service on binding-s109.
app: stopped
Stopping and disabling Zing-memory Service on Worker 192.169.67.174 VM.
Created symlink /etc/systemd/system/zing-memory.service -> /dev/null.
Stopping and disabling Zing-memory Service on Worker 192.169.67.175 VM.
Created symlink /etc/systemd/system/zing-memory.service -> /dev/null.
Starting app service on binding-s108.
app: started
Starting app service on binding-s109.
app: started
Stopping app service on diameter-endpoint-s104.
app: stopped
Stopping app service on diameter-endpoint-s105.
app: stopped
Stopping and disabling Zing-memory Service on Director 192.169.67.176 VM.
mknod: /dev/zing_mm0: File exists
Stopping and disabling Zing-memory Service on Director 192.169.67.177 VM.
Created symlink /etc/systemd/system/zing-memory.service -> /dev/null.
Starting app service on diameter-endpoint-s104.
app: started
Starting app service on diameter-endpoint-s105.
app: started
```

Verifying Current Running JVM Management Service

```
[dra1-sys04-master-0]# dra jvm status
Zulu is enabled..
To Verify Manually login into any worker or director VM check the app process.

cps@dra1-sys04-director-1:~$ ps -aef | grep zulu
root      7643  7639 44 08:13 ?        00:02:45 /usr/lib/jvm/zulu-8-amd64/bin/java -Xms16g
 -Xmx16g -XX:MaxTenuringThreshold=4 -XX:+PrintGCDetails
-XX:+PrintGCTimeStamps -XX:+PrintGCDateStamps -XX:+PrintGCApplicationStoppedTime
-XX:+UnlockDiagnosticVMOptions -XX:+UnsyncloadClass -XX:+TieredCompilation
-XX:+DisableExplicitGC -server -XX:+UseG1GC -XX:+UnlockExperimentalVMOptions
-XX:G1NewSizePercent=5 -XX:G1MaxNewSizePercent=75 -XX:ConcGCThreads=8
-XX:ParallelGCThreads=32 -XX:MaxGCPauseMillis=10000
-XX:ParGCCardsPerStrideChunk=4096 -XX:+ParallelRefProcEnabled
-XX:+AlwaysPreTouch -javaagent:/var/broadhop/jmx/jmxagent.jar
-DnoMemcacheVip=true -DmemcacheIp=localhost:11211 -Dqns.app.type=diameter_endpoint
-Dosgi.classloader.lock=classname -Dqns.config.dir=/etc/broadhop/
-Dqns.instancenum=1 -Dcom.broadhop.run.instanceId=diameter-endpoint-s104.weave.local-1
-Dlogback.configurationFile=/etc/broadhop/logback.xml -Djmx.port=9045
-Dorg.osgi.service.http.port=8080 -Dsnmp.port=1161 -DenableConsulServiceDiscovery=true
 -Dcom.broadhop.run.systemId=test-system -Dcom.broadhop.run.clusterId=test-cluster
-Dcom.broadhop.config.url=http://svn/repos/run/
-Dcom.broadhop.repository.credentials=qns-svn/cisco123@svn
 -Dsession.db.init.1=mongo-session-a-s1 -Dsession.db.init.2=mongo-session-b-s1
-Dsession.db.init.port=27017 -DdisableJms=true -DembeddedBrokerEnabled=false
-Dtarget.compressed.size=1024 -Denable.compression=true
-Denable.dictionary.compression=true -DuseZlibCompression=true
-DenableBestCompression=false -Dcom.broadhop.locking.autodiscovery=true
-DlookasideThreshold=3 -DcompressDebits -Dnetworkguard.tcp.local.ipv6=false
-DrefreshOnChange=true
-DenableRuntimePolling=true -DdefaultNasIp=127.0.0.1
 -Dua.version.2.0.compatible=true -DsessionPadding=1200
-DnodeHeartBeatInterval=9000 -Dcom.mongodb.updaterIntervalMS=400
-Dcom.mongodb.updaterConnectTimeoutMS=600 -Dcom.mongodb.updaterSocketTimeoutMS=600
-DdbSocketTimeout=1000 -DdbSocketTimeout.balance=1000 -DdbConnectTimeout=1200
```

```
-DdbConnectTimeout.balance=1200 -Dmongo.client.thread.maxWaitTime=1200
-Dmongo.client.thread.maxWaitTime.balance=1200 -Dstatistics.step.interval=1
-Dmongo.connections.per.host=5 -Dmongo.connections.per.host.balance=10
-Dmongo.threads.allowed.to.wait.for.connection=10
-Dmongo.threads.allowed.to.wait.for.connection.balance=10 -DmaxLockAttempts=3
-DretryMs=3 -DmessageSlaMs=1500 -DshardPingLoopLength=3 -DshardPingCycle=200
-DshardPingerTimeoutMs=75 -Ddiameter.default.timeout.ms=2000
-DmemcacheClientTimeout=200 -Dlocking.disable=true -Dcontrolcenter.disableAndsf=true
-Ddra.nat.bind.if=eth0 -Dcom.broadhop.q.if=ethwe -DclusterLBIF=eth1
-Djava.rmi.server.disableHttp=true -Dsun.rmi.transport.tcp.handshakeTimeout=90000
-DrqDisableMsgTTL=true -Denable_tcp_nodelay=true -DenableQueueSystem=false
-Ddra.app.mediation.enable=false -Dallow.crd.export.file.with.system.name.prefix=true
-Dcrd.mongo.cache.refresh.interval.margin=15000 -Dcrd.next.reload.delay.time=120
-DudpPrefix=qns -DudpEndPort=5000 -DqueueHeartbeatIntervalMs=25
-Dzmq.send.hwm=1000 -Dzmq.recv.hwm=1000 -Djdiameter.dra=true
-DdraDiameterEndpoint=true -DdraGeoRelay=true -Dosgi.framework.activeThreadType=normal
-jar /var/qps/images/run/plugins/org.eclipse.equinox.launcher_1.1.0.v20100507.jar
-console 9091 -clean -os linux -ws gtk -arch x86_64
cps      25269 15445  0 08:20 pts/0    00:00:00 grep --color=auto zulu
```

dra jvm zing enable

Used to switch the JVM memory management from Zing to Zulu.

Prerequisite

You must place the *cps.pem* file on the orchestrator container under `/data/keystore`

Note The **dra jvm zing enable** command must be executed in the Maintenance Window (MW).

Syntax

```
dra jvm zing enable
       dra jvm status
```

Command Parameters

Table 76: Parameter Description

Command Parameter	Description
Zing	The name of the service that manages the application.
Status	Display the status on the enabled JVM application.

Command Mode

OPERATIONAL

VNFs

DRA VNF

Command Usage

Use this **dra jvm** command to switch between zing.

Examples

Note To enable Zing passing heap memory is not required because this heap memory is handled by its own zing service.

```
# dra jvm zing enable
Are you sure to enable Zing JVM (y/n)?
Enabling Zing service on binding-s108.
Success! Data written to: zing/enabled
Enabling Zing service on binding-s109.
Success! Data written to: zing/enabled
Enabling Zing service on diameter-endpoint-s104.
Success! Data written to: zing/enabled
Enabling Zing service on diameter-endpoint-s105.
Success! Data written to: zing/enabled
Enabling of Zing JVM will restart the App Service,
Are you sure to proceed with App Service Restart (y/n)?
Executing Zing Tasks
Stopping app service on binding-s108.
app: stopped
Stopping app service on binding-s109.
app: stopped
Starting and Enabling Zing-memory Service on Worker 192.169.67.174 VM.
Removed /etc/systemd/system/zing-memory.service.
Starting and Enabling Zing-memory Service on Worker 192.169.67.175 VM.
Removed /etc/systemd/system/zing-memory.service.
Starting app service on binding-s108.
app: started
Starting app service on binding-s109.
app: started
Stopping app service on diameter-endpoint-s104.
app: stopped
Stopping app service on diameter-endpoint-s105.
app: stopped
Starting and Enabling Zing-memory Service on Director 192.169.67.176 VM.
Removed /etc/systemd/system/zing-memory.service.
Starting and Enabling Zing-memory Service on Director 192.169.67.177 VM.
Removed /etc/systemd/system/zing-memory.service.
Starting app service on diameter-endpoint-s104.
app: started
Starting app service on diameter-endpoint-s105.
app: started
```

Verifying Current Running JVM Management Service

To Verify Manually login into any worker or director VM check the app process.

```
cps@dra1-sys04-director-1:~$ ps -aef | grep zing
root     11218 11217 19 07:36 ?        00:06:31 /opt/zing/zing-jdk8/bin/java -server
-XX:+UseC2 -Xmx35248m -XX:ARTAPort=20000 -XX:+UseTickProfiler -XX:+PrintGCDateStamps
-XX:+DisableExplicitGC -XX:CIMaxCompilerThreads=2 -XX:-DisplayVMOutput
-XX:+LogVMOutput -javaagent:/var/broadhop/jmx/jmxagent.jar -DnoMemcacheVip=true
-DmemcacheIp=localhost:11211 -Dqns.app.type=diameter_endpoint
-Dosgi.classloader.lock=classname -Dqns.config.dir=/etc/broadhop/ -Dqns.instancenum=1
-Dcom.broadhop.run.instanceId=diameter-endpoint-s104.weave.local-1
-Dlogback.configurationFile=/etc/broadhop/logback.xml -Djmx.port=9045
```

```
-Dorg.osgi.service.http.port=8080 -Dsnmp.port=1161 -DenableConsulServiceDiscovery=true
-Dcom.broadhop.run.systemId=test-system -Dcom.broadhop.run.clusterId=test-cluster
-Dcom.broadhop.config.url=http://svn/repos/run/
-Dcom.broadhop.repository.credentials=qns-svn/cisco123@svn
-Dsession.db.init.1=mongo-session-a-s1 -Dsession.db.init.2=mongo-session-b-s1
-Dsession.db.init.port=27017 -DdisableJms=true -DembeddedBrokerEnabled=false
-Dtarget.compressed.size=1024 -Denable.compression=true
-Denable.dictionary.compression=true -DuseZlibCompression=true
-DenableBestCompression=false -Dcom.broadhop.locking.autodiscovery=true
-DlookasideThreshold=3 -DcompressDebits -Dnetworkguard.tcp.local.ipv6=false
-DrefreshOnChange=true -DenableRuntimePolling=true -DdefaultNasIp=127.0.0.1
-Dua.version.2.0.compatible=true -DsessionPadding=1200 -DnodeHeartBeatInterval=9000
-Dcom.mongodb.updaterIntervalMS=400 -Dcom.mongodb.updaterConnectTimeoutMS=600
-Dcom.mongodb.updaterSocketTimeoutMS=600 -DdbSocketTimeout=1000
-DdbSocketTimeout.balance=1000 -DdbConnectTimeout=1200 -DdbConnectTimeout.balance=1200
-Dmongo.client.thread.maxWaitTime=1200 -Dmongo.client.thread.maxWaitTime.balance=1200
-Dstatistics.step.interval=1 -Dmongo.connections.per.host=5
-Dmongo.connections.per.host.balance=10 -Dmongo.threads.allowed.to.wait.for.connection=10
-Dmongo.threads.allowed.to.wait.for.connection.balance=10 -DmaxLockAttempts=3
-DretryMs=3 -DmessageSlaMs=1500 -DshardPingLoopLength=3 -DshardPingCycle=200
-DshardPingerTimeoutMs=75 -Ddiameter.default.timeout.ms=2000
-DmemcacheClientTimeout=200 -Dlocking.disable=true -Dcontrolcenter.disableAndsf=true
 -Ddra.nat.bind.if=eth0 -Dcom.broadhop.q.if=ethwe -DclusterLBIF=eth1
-Djava.rmi.server.disableHttp=true -Dsun.rmi.transport.tcp.handshakeTimeout=90000
-DrqDisableMsgTTL=true -Denable_tcp_nodelay=true -DenableQueueSystem=false
-Ddra.app.mediation.enable=false -Dallow.crd.export.file.with.system.name.prefix=true
-Dcrd.mongo.cache.refresh.interval.margin=15000 -Dcrd.next.reload.delay.time=120
-DudpPrefix=qns -DudpEndPort=5000 -DqueueHeartbeatIntervalMs=25
-Dzmq.send.hwm=1000 -Dzmq.recv.hwm=1000 -Djdiameter.dra=true
-DdraDiameterEndpoint=true -DdraGeoRelay=true -Dosgi.framework.activeThreadType=normal
-jar /var/qps/images/run/plugins/org.eclipse.equinox.launcher_1.1.0.v20100507.jar
-console 9091 -clean -os linux -ws gtk -arch x86_64
cps      16965 15445  0 08:09 pts/0    00:00:00 grep --color=auto zing
```

dra subscriber-trace db-connection

To configure mongo db uri.

Syntax

```
dra subscriber-trace db-connection <ip> <port>
```

Command Mode

OPERATIONAL

VNFs

All

Command Usage

Use this CLI command to set new mongo db uri. By default, DRA uses mongo-admin-a:27017, mongo-admin-b:27017,mongo-admin-c:27017 to store all pcap, version, and trace keys.

Examples

The following is an example:

```
show running-config dra subscriber-trace
dra subscriber-trace db-connection 182.22.31.60.27017
!
admin@orchestrator[master-00]
```

dra subscriber-trace db-pcap-collection-max-size

To change pcap collection size in Megabytes.

Syntax

```
dra subscriber-trace db-pcap-collectection-max-size <size in MB>
```

Command Mode

OPERATIONAL

VNFs

All

Command Usage

Use this command to change the size of pcap_files collection. By default, the collection “pcap_files” is created with size 1024 MB. Since the collection “pcap_files” is created as capped collection, DRA automatically deletes oldest pcap entries from the collection and stores new pcap entries.

Examples

The following is an example:

```
show running-config dra subcriber-trace
dra subscriber-trace db-pcap-collection-max-size 1024
!
admin@orchestrator[master-00]
```

dra subscriber-monitor-activity db-activity-collection-max-size

To change activity collection size in megabytes.

Syntax

```
dra subscriber-monitor-activity db-activity-collection-max-size <size in MB>
```

Command Mode

OPERATIONAL

VNFs

All

Command Usage

Use this command to change the activity collection size in megabytes. By default, the collection to store subscriber activities is created with size 1024 MB.

Examples

The following is an example:

```
show running-config dra subcriber-monitor-activity
dra subscriber-monitor-activity db-activity-collection-max-size 1024
!
admin@orchestrator[master-00]
```

dra subscriber-monitor-activity db-connection

To change mongo db uri:

Note The monitor subscriber-activity CLI is used only to view live logs and is not used to store/stop monitor logging activity.

Syntax

```
dra subscriber-monitor-activity db-connection <ip> <port>
```

Command Mode

OPERATIONAL

VNFs

All

Command Usage

Use this CLI command to change mongo db uri. By default, DRA stores monitor activity keys and activity logs in mongo-admin-a:27017, mongo-admin-b:27017, mongo-admin-c:27017.

Examples

The following is an example:

```
show running-config subscriber-monitor-activity
dra subscriber-monitor-activity db-connection 182.22.31.60.27017
!
admin@orchestrator[master-00]
```

dra set-ratelimit binding-api

To configure common rate limit for all binding APIs.

Syntax

dra set-ratelimit binding-api *rate limit value*

Command Parameters

Table 77: Parameter Description

Command Parameter	Description
rate limit value	Specifies the value at which all binding API queries will be rate limited. **Note** Additional 25 per cent is added to the configured rate limit to make sure that DRA reaches the configured rate limit.

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use this CLI command to set the rate limit for all binding API queries, such as imsi imsi-apn, msisdn,msisdn-apn, or Ipv6.

Examples

The following is an example:

dra set-ratelimit binding-api 100.

dra set-ratelimit binding-api-imsi

To configure rate limit for imsi binding API queries.

Syntax

dra set-ratelimit binding-api-imsi *rate limit value*

Command Parameters

Table 78: Parameter Description

Command Parameter	Description
rate limit value	Specifies the value at which imsi binding API queries will be rate limited. **Note** Additional 25 per cent is added to the configured rate limit to make sure that DRA reaches the configured rate limit.

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use this CLI command to set rate limit only for imsi binding API queries.

Examples

The following is an example:

dra set-ratelimit binding-api-imsi 200.

dra set-ratelimit binding-api-imsi-apn

To configure binding API with IMSI APN rate limit.

Syntax

dra set-ratelimit binding-api-imsi-apn *rate limit value*

Command Parameters

Table 79: Parameter Description

Command Parameter	Description
rate limit value	Specifies the value at which imsi-apn binding API queries will be rate limited. **Note** Additional 25 per cent is added to the configured rate limit to make sure that DRA reaches the configured rate limit.

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use this CLI command to set rate limit only for imsi-apn binding API queries.

Examples

The following is an example:

dra set-ratelimit binding-api-imsi-apn 200.

dra set-ratelimit topology-api

To configure rate limit for topology related API queries.

Syntax

dra set-ratelimit topology-api *rate limit value*

Command Parameters

Table 80: Parameter Description

Command Parameter	Description
rate limit value	Specifies the value at which topology related API queries will be rate limited. **Note** Additional 25 per cent is added to the configured rate limit to make sure that DRA reaches the configured rate limit.

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use this CLI command to set rate limit for topology related API queries.

Examples

The following is an example:

dra set-ratelimit topology-api 150.

dra set-ratelimit binding-api-ipv6

To configure rate limit for ipv6 binding API queries.

Syntax

dra set-ratelimit binding-api-ipv6 *rate limit value*

Command Parameters

Table 81: Parameter Description

Command Parameter	Description
rate limit value	Specifies the value at which ipv6 binding API queries will be rate limited. **Note** Additional 25 per cent is added to the configured rate limit to make sure that DRA reaches the configured rate limit.

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use this CLI command to set rate limit only for ipv6 binding API queries.

Examples

The following is an example:

dra set-ratelimit binding-api-ipv6 100.

dra set-ratelimit oam-api

To configure rate limit for OAM-related API queries.

Syntax

dra set-ratelimit oam-api *rate limit value*

Command Parameters

Table 82: Parameter Description

Command Parameter	Description
rate limit value	Specifies the value at which OAM (svn/grafana/prometheus/pb/central/custrefdata) related API queries will be rate limited. **Note** Additional 25 per cent is added to the configured rate limit to make sure that DRA reaches the configured rate limit.

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use this CLI command to set rate limit for OAM (svn/grafana/prometheus/pb/centralcustrefdata) related API queries.

Examples

The following is an example:

dra set-ratelimit oam-api 50

dra set-ratelimit slf-api

To configure rate limit for SLF-related API queries.

Syntax

dra set-ratelimit slf-api *rate limit value*

Command Parameters

Table 83: Parameter Description

Command Parameter	Description
rate limit value	Specifies the value at which slf related API queries will be rate limited. **Note** Additional 25 per cent is added to the configured rate limit to make sure that DRA reaches the configured rate limit.

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use this CLI command to set rate limit for SLF-related API queries..

Examples

The following is an example:

dra set-ratelimit slf-api 300

dra set-ratelimit session-api

To configure rate limit for session related API queries.

Syntax

dra set-ratelimit session-api *rate limit value*

Command Parameters

Table 84: Parameter Description

Command Parameter	Description
rate limit value	Specifies the value at which session related queries will be rate limited. **Note** Additional 25 per cent is added to the configured rate limit to make sure that DRA reaches the configured rate limit.

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use this CLI command to set rate limit for session related API queries.

Examples

The following is an example:

dra set-ratelimit session-api 50

dra set-ratelimit binding-api-msisdn

To configure rate limit only for msisdn binding API queries.

Syntax

dra set-ratelimit binding-api-msisdn *rate limit value*

Command Parameters

Table 85: Parameter Description

Command Parameter	Description
rate limit value	Specifies the value at which msisdn binding API queries will be rate limited. **Note** Additional 25 per cent is added to the configured rate limit to make sure that DRA reaches the configured rate limit.

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use this CLI command to set rate limit only for msisdn binding API queries.

Examples

The following is an example:

dra set-ratelimit binding-api-msisdn 200

dra set-ratelimit binding-api-msisdn-apn

To configure rate limit only for msisdn-apn binding API queries.

Syntax

dra set-ratelimit binding-api-msisdn-apn *rate limit value*

Command Parameters

Table 86: Parameter Description

Command Parameter	Description
rate limit value	Specifies the value at which msisdn-apn binding queries will be rate limited.

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use this CLI command to set rate limit only for msisdn-apn binding API queries.

Examples

The following is an example:

dra set-ratelimit binding-api-msisdn-apn 200.

dra remove-ratelimit binding-api-imsi

Removes rate limit for all binding API queries.

Syntax

```
dra remove-ratelimit binding-api-imsi
```

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use this command to remove rate limit only for imsi binding API queries.

dra remove-ratelimit binding-api-imsi-apn

Removes rate limit only for imsi-apn binding API queries.

Syntax

dra remove-ratelimit binding-api-imsi-apn

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use this CLI command to remove rate limit only for imsi-apn binding API queries.

dra remove-ratelimit binding-api-ipv6

Removes rate limit for ipv6 binding API queries.

Syntax

```
dra remove-ratelimit binding-api-ipv6
```

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use this command to remove rate limit only for ipv6 binding API queries.

dra remove-ratelimit binding-api-msisdn-apn

Removes rate limit for msisdn-apn binding API queries.

Syntax

```
dra remove-ratelimit binding-api-msisdn-apn
```

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use this command to remove rate limit only for msisdn-apn binding API queries.

dra remove-ratelimit binding-api-msisdn

Removes rate limit for msisdn binding API queries.

Syntax

```
dra remove-ratelimit binding-api-msisdn
```

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use this command to remove rate limit only for msisdn binding API queries.

dra remove-ratelimit binding-api

To remove rate limit for all binding API queries.

Syntax

```
dra remove-ratelimit binding-api
```

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use this command to remove rate limit for all binding API queries (imsi/imsi-apn/msisdn/msisdn-apn/Ipv6), which was configured using option *binding-api*.

Examples

The following is an example to remove rate limit binding-api:

```
dra remove-ratelimit binding-api
```

dra remove-ratelimit oam-api

Removes rate limit for rate limit for OAM related API queries.

Syntax

```
dra remove-ratelimit oam-api
```

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use this command to remove rate limit for OAM (svn/grafana/prometheus/pb/centralcustrefdata) related API queries.

dra remove-ratelimit session-api

Removes rate limit for session related API queries.

Syntax

```
dra remove-ratelimit session-api
```

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use this command to remove rate limit for session related API queries.

dra remove-ratelimit slf-api

Removes rate limit for slf related API queries.

Syntax

```
dra remove-ratelimit slf-api
```

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use this command to remove rate limit for slf related API queries.

dra show-ratelimit topology-api

Used to display the configured rate limit for topology related API queries.

Syntax

```
dra show-ratelimit topology-api
```

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use this command to view the configured rate limit for topology related API queries.

Examples

The following is an example:

```
dra show-ratelimit topology-api
dra/ratelimit/topology-api:150
```

dra show-ratelimit binding-api-imsi-apn

Used to display the configured rate limit for imsi-apn binding API queries.

Syntax

```
dra show-ratelimit binding-api-imsi-apn
```

Command Mode

CONFIG

VNFs

DRA

Command Usage

Use this command to view the configured rate limit for imsi-apn binding API queries.

Examples

The following is an example:

```
dra show-ratelimit binding-api-imsi-apn
dra/ratelimit/binding-api-imsi-apn:200
```

dra show-ratelimit binding-api-imsi

Used to display the configured rate limit for imsi or imsi-apn binding API queries.

Syntax

```
dra show-ratelimit binding-api-imsi
```

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use this command to view the configured rate limit for imsi or imsi-apn binding API queries.

Examples

The following is an example:

```
dra show-ratelimit binding-api-imsi
dra/ratelimit/binding-api-imsi:200
dra/ratelimit/binding-api-imsi-apn:200
```

dra show-ratelimit binding-api-msisdn-apn

Used to display the configured rate limit for msisdn-apn binding API queries.

Syntax

```
dra show-ratelimit binding-api-msisdn-apn
```

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use this command to view the configured rate limit for msisdn-apn binding API queries.

Examples

The following is an example:

```
dra show-ratelimit binding-api-msisdn-apn
dra/ratelimit/binding-api-msisdn-apn:200
```

dra show-ratelimit binding-api-ipv6

Used to display the configured rate limit for ipv6 binding API queries.

Syntax

```
dra show-ratelimit binding-api-ipv6
```

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use this command to view the configured rate limit for ipv6 binding API queries.

Examples

The following is an example:

```
dra show-ratelimit binding-api-ipv6
dra/ratelimit/binding-api-ipv6:100
```

dra show-ratelimit binding-api-msisdn

Used to display the configured rate limit for msisdn or msisdn-apn binding API queries.

Syntax

```
dra show-ratelimit binding-api-msisdn
```

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use this command to view the configured rate limit for msisdn or msisdn-apn binding API queries.

Examples

The following is an example:

```
dra show-ratelimit binding-api-msisdn
dra/ratelimit/binding-api-msisdn:200
dra/ratelimit/binding-api-msisdn-apn:200
```

dra show-ratelimit binding-api

Used to display the configured rate limit of all binding API queries.

Syntax

```
dra show-ratelimit binding-api
```

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use this command to view the configured rate limit of all binding API queries.

Examples

The following is an example:

```
dra show-ratelimit binding-api
dra/ratelimit/binding-api:100
dra/ratelimit/binding-api-imsi:200
dra/ratelimit/binding-api-imsi-apn:200
dra/ratelimit/binding-api-ipv6:100
dra/ratelimit/binding-api-msisdn:200
dra/ratelimit/binding-api-msisdn-apn:200
```

dra show-ratelimit oam-api

Used to display the configured rate limit for OAM-related API queries.

Syntax

```
dra show-ratelimit oam-api
```

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use this command to view the configured rate limit for OAM (svn/grafana/prometheus/pb/centralcustrefdata) related API queries.

Examples

The following is an example:

```
dra show-ratelimit oam-api
dra/ratelimit/oam-api:50
```

dra show-ratelimit session-api

Used to display the configured rate limit for session related API queries.

Syntax

```
dra show-ratelimit session-api
```

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use this command to view the configured rate limit for session related API queries.

Examples

The following is an example:

```
dra show-ratelimit session-api
dra/ratelimit/session-api:50
```

dra show-ratelimit slf-api

Used to display the configured rate limit for SLF related API queries.

Syntax

```
dra show-ratelimit slf-api
```

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use this command to view the configured rate limit for SLF related API queries.

Examples

The following is an example:

```
dra show-ratelimit slf-api
dra/ratelimit/slf-api:300
```

dra show-ratelimit

Used to display the configured rate limit for all ingress APIs.

Syntax

```
dra show-ratelimit
```

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use this command to view the configured rate limit for all ingress APIs.

Examples

The following is an example:

```
dra show-ratelimit
dra/ratelimit/binding-api:100
dra/ratelimit/binding-api-imsi:200
dra/ratelimit/binding-api-imsi-apn:200
dra/ratelimit/binding-api-ipv6:100
dra/ratelimit/binding-api-msisdn:200
dra/ratelimit/binding-api-msisdn-apn:200
dra/ratelimit/oam-api:50
dra/ratelimit/session-api:50
dra/ratelimit/slf-api:300
dra/ratelimit/topology-api:150
```

dra-tls cert import

Used to import the TLS Certificates from Orchestrator to diameter-endpoint containers.

You can provide certificate and private files as an input for this CLI command and also provide the keystore password to encrypt the *cert* file. Back-end script converts the files into java-based file format called JKS with encryption, then it copies to the diameter-endpoint containers. This cmd supports only *.pem* file format certificates

Pre-requisites

- Ensure to place the *cps.pem* file on orchestrator container under `/data/keystore`
- Ensure to place the TLS Certificate files on Master VM under `/data/orchestrator/pemKey`

The new keystore file gets placed in a folder in the diameter container `/etc/tls/certs/`.

Syntax

`dra-tls cert import < Certificate file > < Private key file >`

Note This command must be executed in Maintenance Window (MW).

Command Parameters

Table 87: Parameter Description

Command Parameter	Description
Certificate file	The name of the tls certificate file.
Private Key file	The name of the *tls* private key file.

Command Mode

OPERATIONAL

VNFs

DRA VNF

Command Usage

This command allows you to load the *tls* certificates file into diameter-endpoint containers with encrypted format.

Example 1

1. Copy your tls certificate and private file into /data/orchestrator/pemKey location under master VM.

root@dra1-sys04-master-0:/data/orchestrator/pemKey# ls private-key.pem tls-cert.pem

2. Login into Orchestrator container CLI and run the below cmd's to import the certs.

admin@orchestrator[dra1-sys04-master-0]# dra-tls cert import tls-cert.pem private.pem

Please enter the Keystore Password for this private.pem cert:********

Importing keystore /data/pemKey/certificate-tls.p12 to /data/pemKey/diameter-endpoint-tls.jks...

Example 2

1. Copy your tls certificate and private file into /data/orchestrator/pemKey location under master VM.

root@dra1-sys04-master-0:/data/orchestrator/pemKey# ls private-key.pem tls-cert.pem

2. Login into Orchestrator container CLI and run the below cmd's to import the certs.

admin@orchestrator[pn-master-0]# dra-tls cert import CA-cert.pem CA-key.pem

Please enter the Keystore Password for this private.pem cert:********

Importing keystore /data/pemKey/certificate-tls.p12 to /data/pemKey/diameter-endpoint-tls.jks...

Warning:

The JKS keystore uses a proprietary format. We recommend migrating to PKCS12 which is an industry standard format using "keytool -importkeystore -srckeystore /data/pemKey/diameter-endpoint-tls.jks -destkeystore /data/pemKey/diameter-endpoint-tls.jks -deststoretype pkcs12".

Importing Certificate to 192.169.67.176.

Import Successfully Completed for 192.169.67.176

Importing Certificate to 192.169.67.177.

Import Successfully Completed for 192.169.67.177

dra ipc-send-thread

Used to configure the IPC send thread parameters.

Syntax

dra ipc-send-thread limit *thread-limit* **lock-sla-timeout** *time-in-ms* **message-throttle-duration** *duration-in- ms* **timeout-sample-to-throttle** *max-samples*

Command Parameters

Table 88: Parameter Description

Command Parameter	Description
thread-limit	The maximum number of IPC send threads that waits to acquire the lock per peer connection.
time-in-ms	The maximum time the IPC send thread that waits to acquire the lock in Milli-Seconds. .
duration-in-ms	The maximum duration the IPC send threads throttle the messages in MilliSeconds.
max-samples	The maximum number of SLA timeout samples to throttle the peer.

Command Mode

CONFIG

VNFs

DRA

Command Usage

Use this command to tune the IPC send thread parameters to handle the slow network peers.

Examples

The following is an example of tuning IPC send thread parameters.

```
(config)# dra ipc-send-thread limit 50 lock-sla-timeout 250 message-throttle-duration 30000
 timeout-sample-to-throttle 150
```

Note For more information, see the *IPC Queue Send Thread Tuning Configuration* section in the *CPS vDRA Advance Tuning Guide*.

dra ipc-send-thread priority

Used to configure the IPC send thread priority parameters.

Syntax

```
dra ipc-send-thread priority limit thread-limit priority lock-sla-timeout time-in-ms priority
 message-throttle-duration duration-in- ms priority timeout-sample-to-throttle max-samples
```

Command Parameters

Table 89: Parameter Description

Command Parameter	Description
thread-limit	The maximum number of IPC priority send threads that waits to acquire the lock per peer connection.
time-in-ms	The maximum time the IPC priority send thread that waits to acquire the lock in Milli-Seconds. .
duration-in-ms	The maximum duration the IPC priority send threads throttle the messages in MilliSeconds.
max-samples	The maximum number of priority SLA timeout samples to throttle the peer.

Command Mode

CONFIG

VNFs

DRA

Command Usage

Use this command to tune the IPC send thread priority parameters to handle the slow network peers.

Examples

The following is an example of tuning IPC send thread priority parameters with default values.

```
(config)# dra ipc-send-thread priority limit 5 priority lock-sla-timeout 200 priority
message-throttle-duration 30000 priority timeout-sample-to-throttle 150
```

end

Used to terminate a configuration session.

Syntax

```
end
```

Command Mode

CONFIG

VNFs

All

Command Usage

Use the end command to exit any configuration mode and return directly to operational mode. If you enter this command without committing the changes to the target configuration, you are prompted to do so:

```
Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]:
```

- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns to the operational mode.
- If errors are found in the running configuration, the configuration session does not end. To view the errors, enter the `show configuration (config)` command with the failed keyword.
- Entering **no** exits the configuration session and returns to the operational mode without committing the configuration changes.
- Entering **cancel** leaves the CLI prompt in the current configuration session without exiting or committing the configuration changes.

Examples

The following is an example:

```
network dns host reladsdsdydra1.client.3gppnetwork.org local address X:X::X:X
admin@orchestrator[scheduler](config-host-reladsdsdydra1.client.3gppnetwork.org/local)# end
Uncommitted changes found, commit them? [yes/no/CANCEL]
```

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 90: Parameter Description

Command Parameter	Description
gid:int	GID mapping value.

Command Parameter	Description
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
```

external-aaa pam username-mapping

Configures the user name mapping for various roles.

Syntax

```
external-aaa pam username-mapping <USER-NAME> <ROLE>
```

Command Parameters

Table 91: Parameter Description

Command Parameter	Description
USER-NAME	Name of the user.
ROLE	Role for which the mapping is required.

Command Mode

OPERATIONAL

VNFs

All

Command Usage

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

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

Example

```
external-aaa pam username-mapping anil policy-admin
```

license feature

Registers a system license.

Syntax

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

Command Parameters

Table 92: 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.

load

Used to load configuration from file or terminal.

Syntax

```
load { merge | replace | override } { <file> | terminal }
```

Command Parameters

Table 93: Parameter Description

Command Parameter	Description
merge	Merge content of file/terminal with current configuration..
replace	Replace the content of file/terminal for the corresponding parts of the current configuration. In case of replace, the parts that are common in the file/terminal are replaced and rest of the configuration is not modified.
override	In case of override, the entire configuration is deleted (with the exception of hidden data) before loading the new configuration from the file/terminal.

Command Mode

CONFIG

VNFs

All

Command Usage

Use the load command to merge/replace/override from file or terminal.

Examples

The configuration file can contain replace: and delete: directives. The following is an example:

```
system {
parent-mo {
child-mo 1 {
attr 10;
}
child-mo 2 {
attr 5;
}
}
}
```

If you want to delete child-mo 2, you can create a configuration file containing either:

- **replace:**

```
system {
replace:
parent-mo {
child-mo 1 {
attr 2;
}
}
}
```

- **delete:**

```
system {
parent-mo {
delete:
child-mo 2 {
attr 5;
}
}
}
```

logger set

Sets the various log levels for application logging.

Syntax

```
logger set logger-name { trace | debug | info | warn | error | off } logger-level
container-name,container-name2,containers-name3
```

Command Parameters

Table 94: 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.
logger-level	Specifies application logs are enabled and logging for all the containers.

Command Parameter	Description
container-name	Enable, disable, or view the application logs for a specific container. For example, you can enable application logging for up to three containers by specifying the logger level and the names of three different containers.

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 95: 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.

log collect config

Configures the destination server details for log collection.

Syntax

Log collect config ip *ip* **port** *port* **user** *user*

Command Mode

OPERATIONAL

VNFs

All

Command Usage

Use the **log collect config** command to configure the destination server details. You can specify a password in the interactive mode.

If user is "cps", PEM file will be used.

If user is gtac or any other user, a correct password must be specified in the interactive mode.

The password is stored in an encrypted format and displayed in an encrypted format only.

Examples

The following sample output is an example for log collect feature configs.

```
log collect show
Log collect configurations    Current Value
----------------------------  --------------------------------
ip                            173.39.57.214
port                          22
user                          msivapra
password                      42A61FBF99537F7C972E7ADDC2BA453F
```

log collect all

Collects all required logs.

Syntax

log collect all
[duration *timeperiod in hours* **]**

Command Mode

DEBUG

VNFs

All

Command Usage

Use the **log collect all** to collect all required logs.

Internally output files are stored in the below location inside orchestrator container.

`/var/broadhop/fileserver/logs/`

The log files are copied to the DIM server and removed from the Master VM after copying is complete.

If the SCP to External server fails, the files remains in the `/var/broadhop/fileserver/logs/` in the orchestrator. You can copy or delete these log files manually.

If such files are not manually copied, they can be copied in the next run of the command and gets removed from the orchestrator.

log-forward fluentbit local-forward

Used to configure the OAM IP address (master, control-a or control-b) to forward all logs from other non-proxy VMs to the OAM VM/proxy VM before it is forwarded to external server.

Syntax

`log-forward fluentbit local-forward OAM-ip <OAM-ip> OAM-Port <OAM-port>`

Command Parameters

Table 96: Parameter Description

Command Parameter	Description
OAM-ip	The VIP to be set to enable local forwarding.
OAM-port	The port to be set for local forwarding.

Command Mode

OPERATIONAL

VNFs

DRA and Binding

Command Usage

Use the `log-forward fluentbit local-forward` command to forward all logs locally before forwarding to external server.

Examples

The following is an example:

```
admin@orchestrator[site](config)# log-forward fluentbit local-forward
Value for 'OAM-ip' (<IP address>): x.x.x.x
Value for 'OAM-port' (<int>): 9200
status {
 data Starting Local Forwarding logs to : host x.x.x.x, port 9200
}
```

log-forward fluentbit external-forward

Used to forward logs that are collected by fluentbit to any external server.

Note An External server can have fluentbit or fluentd or any log forwarder installed.

Syntax

```
log-forward fluentbit external-forward
```

Command Parameters

Table 97: Parameter Description

Command Parameter	Description
external-ip	The IP of the external server hosting fluent-bit instance where logs are forwarded without authentication
external-port	Port of the external server hosting fluent-bit instance where logs will be forwarded without authentication.

Command Mode

OPERATIONAL

VNFs

DRA and Binding

Command Usage

This command enables fluentbit services to forward all the logs that are collected at one of the OAM server to any external server (having fluentbit or fluentd or any log forwarder installed).

The following logs are forwarded:

- Journalctl
- Mongod logs (from db VMs)
- Consolidated-qns.log
- Consolidated-pb.log
- Audit logs

Examples

The following is an example to enable the forwarding:

```
admin@orchestrator[site]#log-forward fluentbit external-forward external-ip
Value for 'external-ip' (<IP address>): X.X.X.X
Value for 'external-port' (<int>): XXXX
status {
    data Starting forwarding logs to external server: host X.X.X.X, port XXXX
}
```

Note Use this command only after the local forwarding is enabled. If local forwarding is modified to any different OAM server, you must configure this external forwarding again.

log-forward fluentbit elasticsearch

Used to configure the Elasticsearch details to forward all DRA logs. Make sure to use this command after **log-forward fluentbit local-forward OAM-ip***OAM-ip* **OAM-port** *OAM-port*

Syntax

```
log-forward fluentbit elasticsearch elastic-ip <elastic-ip> elastic-port
<elastic-port> elastic-user <elastic-user> elastic-password <elastic-password>
```

Command Parameters

Table 98: Parameter Description

Command Parameter	Description
elasticsearch IP	IP to be set to enable remote forwarding for elasticsearch server.
elasticsearch port	Port to be set to enable remote forwarding.
elastic user	User to be specified to enable remote forwarding.
elastic password	Password to be set to authenticate remote forwarding.

Command Mode

OPERATIONAL

VNFs

DRA and Binding

Command Usage

Use the `log-forward fluentbit elasticsearch` command to configure the Elasticsearch details to forward all DRA logs.

Examples

The following is an example:

```
admin@orchestrator[site](config)# log-forward fluentbit elasticsearch
Value for 'elastic-ip' (<IP address>): x.x.x.x
Value for 'elastic-port' (<int>): 9400
Value for 'elastic-user' (<string>): elastic
Value for 'elastic-password' (<string>): ********
status {
 data Starting forwarding logs to elasticsearch: host x.x.x.x, port 9400 for elastic user
}
```

log-forward fluentbit enable

Used to enable the fluentbit services before configuring the local forwarding and remote forwarding of logs to external server. Also it enables/resumes the forwarding if already disabled/paused.

Syntax

```
log-forward fluentbit enable
```

Command Mode

OPERATIONAL

VNFs

DRA and Binding

Command Usage

This command enables the fluentbit services before configuring the local forwarding and remote forwarding of logs. Also it enables/resumes the forwarding if already disabled/paused.

Examples

The following is an example to enable the forwarding:

```
admin@orchestrator[site2]# log-forward fluentbit enable
Enabling fluentbit success: Setting enable flag on X.X.X.X
Enabling fluentbit success: Setting enable flag on X.X.X.X
Enabling fluentbit success: Setting enable flag on X.X.X.X
```

log-forward fluentbit disable

Used to disable the fluentbit services at DRA (all VMs) for forwarding logs to external server.

Syntax

```
log-forward fluentbit disable
```

Command Mode

OPERATIONAL

VNFs

DRA and Binding

Command Usage

This command disables the fluentbit services at DRA (all VMs) during local forwarding and remote forwarding of logs.

Examples

The following is an example to disable the forwarding:

```
admin@orchestrator[site2]# log-forward fluentbit disable
Disabling fluentbit success: Setting disable flag on X.X.X.X
Disabling fluentbit success: Setting disable flag on X.X.X.X
Disabling fluentbit success: Setting disable flag on X.X.X.X
```

log-forward fluentbit filter

Used to set the filters to suppress any logs that should not be forwarded to elasticsearch.

Syntax

```
log-forward fluentbit filter key <Key-name> pattern <pattern-to-be-suppressed>
```

Command Parameters

Table 99: Parameter Description

Command Parameter	Description
key	Key name to be set for journalctl keys or log keys.
pattern	Pattern to be set for regex patterns of logs to be suppressed. **Note** Filter-regex: Regex expression to be parsed to apply filter configuration at OAM VM.

Command Mode

OPERATIONAL

VNFs

DRA and Binding

Command Usage

Use the `log-forward fluentbit filter` command to set the filters to suppress any logs that should not be forwarded to elasticsearch.

Examples

The following is an example:

```
admin@orchestrator[site](config)# log-forward fluentbit filter
Value for 'key' (<string>): CONTAINER_NAME
Value for 'pattern' (<string>): orchestrator
status {
 data Filters Applied for Forwarding logs : Key CONTAINER_NAME, Pattern orchestrator
}
```

log-forward fluentbit filter-clear

Used to clear all filters.

Syntax

```
log-forward fluentbit filter-clear
```

Command Mode

OPERATIONAL

VNFs

DRA and Binding

Command Usage

Use the `log-forward fluentbit filter-clear` command to clear all filters.

Examples

The following is an example:

```
admin@orchestrator[site2-dra-master0](config)# log-forward fluentbit filter-clear
status {
 data Filters are cleared successfully
}
```

log-forward fluentbit tune

Used to configure tuning parameters.

Syntax

```
log-forward fluentbit tune flush_interval <flush_interval>
max_chunks_up <max_chunks_up> max_memory_backlog <max_memory_backlog>
```

Command Parameters

Table 100: Parameter Description

Command Parameter	Description
flush interval	Interval to be set for flushing logs to elastisearch.
max chunks up	The maximum chunks at the OAM, which should be active for getting flushed at one time.
max memory backlog	The maximum backlog size OAM server to keep the logs.

Command Mode

OPERATIONAL

VNFs

DRA and Binding

Command Usage

Use the `log-forward fluentbit tune` command to tune paramters for flushing.

Examples

The following is an example:

```
admin@orchestrator[site](config)# log-forward fluentbit tune flush_interval 900 max_chunks_up
 3500 max_memory_backlog 2048M
status {
    data Tunings modified for remote forwarding
}
```

monitor binding-db-vms clustername

Used to monitor only particular types of DBs based on cluster names in DB VNF.

Syntax

```
monitor binding-db-vms clustername  cluster-name
```

Command Parameters

Table 101: Parameter Description

Command Parameter	Description
cluster-name	The name of the DB cluster.

Command Mode

CONFIG

VNFs

Binding

Command Usage

Use the `monitor binding-db-vms clustername cluster-name` command to monitor only particular types of DBs based on cluster names in DB VNF.

Examples

The following configuration monitors particular types of DBs based on the cluster name.

`monitor binding-db-vms clustername` *`binding`*

The following CLI command fetches CPU monitoring values in DRA VNF.

`binding shard-metadata-db-connection loadmetrics ipaddress port`

mfa-cli enable user-id

Enables MFA support for orchestrator in vDRA.

Syntax

`mfa-cli enable user-id` *`USER_ID`*
`no-cli mfa enable user-id` *`USER_ID`*

Command Mode

CONFIGURATION

VNFs

All

Examples

Example 1:The following is an example to enable the MFA user:

```
admin@orchestrator[TEST-Binding-master](config)# mfa-cli enable user-id 16181
admin@orchestrator[TEST-Binding-master](config)# commit
Commit complete.
```

Here is the show command output to display the enabled MFA user:

```
admin@orchestrator[TEST-Binding-master]# show running-config mfa-cli enable
mfa-cli enable user-id 16181
```

Example 3:The following is an example to disable the MFA user:

```
admin@orchestrator[TEST-Binding-master](config)# no mfa-cli enable user-id 16181
admin@orchestrator[TEST-Binding-master](config)# commit
Commit complete.
```

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 102: 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.

monitor subscriber-activity

To view live monitor subscriber activity logs in the vDRA

Note The monitor subscriber-activity CLI is used only to view live logs and is not used to store/stop monitor logging activity.

Syntax

```
monitor subscriber-activity imsi <IMSI value> user <admin>
monitor subscriber-activity msisdn <MSISDN value> user <admin>
monitor subscriber-activity ipv6 <IPv6 value> user <admin>
```

Command Mode

OPERATIONAL

VNFs

All

Command Usage

Use this CLI command to view only monitor subscriber activity logs. Specify the Subscriber identity (IMSI/MSISDN/IPV6), DRA central username, and password to fetch live monitor logs from “monitor_activity_db” in admin-db for the subscriber.

Examples

The following is an example:

```
admin@orchestrator[master-00]# monitor subscriber-activity imsi 450005978851103 user admin
Enter host password for user 'admin':
```

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 103: 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.
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 consul-cleanup true

Use this command to cleanup the consul database and restart the cosul servers during network migration, if the migration is stuck with the consul modules.

Syntax

network consul-cleanup true

Command Mode

OPERATIONAL

VNFs

All

Command Usage

Use the `network consul-cleanup true` to cleanup the consul database and estart the cosul servers during network migration, if the migration is stuck with the consul modules in vDRA.

Note After the **consul-cleanup**, restore the *consul kv* data. On the master node, navigate to `/data/orchestrator/overlay-scripts/` and execute `./network-migration -q <consul_kv_data.json >`.

network detach-overlay true

Use this command to detach the Overlay network after migrating to Weave network.

Syntax

network detach-overlay true

Command Mode

OPERATIONAL

VNFs

All

Command Usage

Use the `network detach-overlay true` to detach the Overlay network after migrating to Weave network.

network detach-weave true

Use this command to detach the Weave network after migrating to Overlay network.

Syntax

network detach-weave true

Command Mode

OPERATIONAL

VNFs

All

Command Usage

Use the `network detach-weave true` to detach the Weave network after migrating to Overlay network.

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 104: 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 105: 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.

Command Parameter	Description
address	The IP address of the host / domain name mapping. **Note** Local address must not be used from the pool 172.17.0.0/16. This IP address set is dedicated to docker containers.

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 migrate-to-overlay true

Use this command to enable Docker Overlay network driver in vDRA.

Syntax

network migrate-to-overlay true

Command Mode

OPERATIONAL

VNFs

All

Command Usage

Use the `network-migrate-to-overlay true` to enable Docker Overlay network driver in vDRA.

network migrate-to-weave true

Use this command to enable Weave network in vDRA.

Syntax

network migrate-to-weave true

Command Mode

OPERATIONAL

VNFs

All

Command Usage

Use the `network migrate-to-weave true` to enable Weave network in vDRA.

network migration-status

Use this command to verify the current migration status of network driver.

Syntax

network migration-status

Command Mode

OPERATIONAL

VNFs

All

Command Usage

Use the `network migration-status` to verify the current migration status of network driver.

network refresh-overlay-config

Use this command to refresh the latest Overlay script configuration file update.

.

Syntax

network refresh-overlay-config true

Command Mode

OPERATIONAL

VNFs

All

Command Usage

Use the `network refresh-overlay-config` to refresh the latest Overlay script configuration file update.

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 106: 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.

Command Parameter	Description
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 107: 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 108: 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
```

prometheus delete-snapshot

Deletes the Prometheus snapshots based on data source type.

Syntax

`prometheus delete-snapshot <vm-name> <data-source>`

Command Parameters

Table 109: Parameter Description

Command Parameter	Description
vm-name	Takes the value as master, control-a or control-b.
data-source	Takes the value as planning, trending or hi-res.

Command Mode

OPERATIONAL

VNFs

DRA and Binding

Command Usage

The **prometheus delete-snapshot <vm-name> <data-source>** command delete the Prometheus snapshots.

Examples

The following is an example to delete the Prometheus snapshots:

```
prometheus delete-snapshot master hi-res
```

prometheus list-snapshot

Lists the Prometheus snapshots based on data source type.

Syntax

prometheus list-snapshot <vm-name> <data-source>

Command Mode

OPERATIONAL

VNFs

DRA and Binding

Command Usage

The **prometheus list-snapshot <vm-name> <data-source>** command list the Prometheus snapshots.

Examples

The following is an example to list the Prometheus snapshots:

```
prometheus list-snapshot control-a trending
```

prometheus restore-snapshot

Restores the Prometheus snapshots based on data source type.

Syntax

prometheus restore-snapshot <vm-name> <data-source>

Command Mode

OPERATIONAL

VNFs

DRA and Binding

Command Usage

The **prometheus restore-snapshot <vm-name> <data-source>** command restores the Prometheus snapshots.

Examples

The following is an example to restore the Prometheus snapshots:

```
prometheus restore-snapshot control-b hi-res
```

prometheus save-snapshot

Saves the Prometheus snapshots based on data source type.

Syntax

prometheus save-snapshot <vm-name> <data-source>

Command Mode

OPERATIONAL

VNFs

DRA and Binding

Command Usage

The **prometheus save-snapshot <vm-name> <data-source>** command save the Prometheus snapshots.

Examples

The following is an example to save the Prometheus snapshots:

```
prometheus save-snapshot master planning
```

prometheus retention-period planning config

Modifies the default retention time (8760h) to user specific retention time for planning datastore to retain Prometheus data.

Syntax

prometheus retention-period planning config *retention_time retention time in hrs*

Command Parameters

Table 110: Parameter Description

Command Parameter	Description
Retention time	Set new retention time for planning datastore of prometheus.

Command Mode

OPERATIONAL

VNFs

DRA and Binding

Command Usage

The **prometheus retention-period planning config** command configures the retention time for the planning datastore to retain Prometheus data.

Examples

The following is an example to modify the retention time of 3600h:

```
prometheus retention-period planning config retention_time 3600h
```

prometheus retention-period planning show

Displays the configured retention time for the planning datastore to retain Prometheus data. If there is an empty value, configure the default value of 8760h.

Syntax

prometheus retention-period planning show

Command Mode

OPERATIONAL

VNFs

DRA and Binding

Command Usage

The **prometheus retention-period planning show** command displays the configured retention time for the planning datastore to retain Prometheus data.

Examples

The following is an example to display the newly configured retention time for planning datastore:

```
prometheus retention-period planning show
Prometheus-Retention-Period    Current Value
-----------------------------  ---------------
retention_time                 3600h
```

prometheus retention-period planning clear

Clears the configured retention time for the planning datastore to retain Prometheus data and reverts the retention time to default retention time (8760h).

Syntax

prometheus retention-period planning clear

Command Mode

OPERATIONAL

VNFs

DRA and Binding

Command Usage

The **prometheus retention-period planning clear** clears the configured retention time for the planning datastore to retain Prometheus data and reverts the retention time to default time (8760h).

Examples

The following is an example to clear the configured retention time for planning datastore:

```
prometheus retention-period planning clear
```

prometheus scrape-interval hi-res config scrape_interval

Modifies the default scrape interval timing to user specific scrape interval to retain the data for longer time.

Syntax

prometheus scrape-interval hi-res config scrape_interval *interval time in seconds*

Command Parameters

Table 111: Parameter Description

Command Parameter	Description
Interval time	Set new interval time for hi-res container of prometheus.

Command Mode

OPERATIONAL

VNFs

DRA and Binding

Command Usage

The **prometheus scrape-interval hi-res config scrape_interval** command configures the interval time for the hi-res container to retain Prometheus data.

Examples

The following is an example to modify the interval time of 30s:

```
prometheus scrape-interval hi-res config scrape_interval 30s
```

prometheus scrape-interval hi-res clear

Clears the configured scrape interval to retain Prometheus data and reverts the interval time to default scrape interval.

Syntax

prometheus scrape-interval hi-res clear

Command Mode

OPERATIONAL

VNFs

DRA and Binding

Command Usage

The **prometheus scrape-interval hi-res clear** clears the configured scrape interval time to retain Prometheus data and reverts the scrape interval to default time.

Examples

The following is an example to clear the interval time in hi-res container:

```
prometheus scrape-interval hi-res clear
```

prometheus scrape-interval hi-res show

Displays the configured scrape period to retain Prometheus data.

Syntax

prometheus scrape-interval hi-res show

Command Mode

OPERATIONAL

VNFs

DRA and Binding

Command Usage

The **prometheus scrape-interval hi-res show**command displays the configured scrape interval time to retain Prometheus data for longer time.

Examples

The following is an example to display the newly configured scrape interval:

```
prometheus scrape-interval hi-res show
```

prometheus scrape-interval planning config scrape_interval

Modifies the default scrape period to user specific scrape period for planning datastore to retain Prometheus data.

Syntax

prometheus scrape-interval planning config scrape_interval *interval time in seconds*

Command Parameters

Table 112: Parameter Description

Command Parameter	Description
Interval time	Set new interval time for the planning container of prometheus.

Command Mode

OPERATIONAL

VNFs

DRA and Binding

Command Usage

The **prometheus scrape-interval planning config scrape_interval** command configures the interval time for the planning container to retain Prometheus data.

Examples

The following is an example to modify the interval time of 125s:

```
prometheus scrape-interval planning config scrape_interval 125s
```

prometheus scrape-interval planning clear

Clears the configured scrape interval to retain Prometheus data and reverts the interval time to default scrape interval.

Syntax

prometheus scrape-interval planning clear

Command Mode

OPERATIONAL

VNFs

DRA and Binding

Command Usage

The **prometheus scrape-interval planning clear** clears the configured scrape interval time to retain Prometheus data and reverts the scrape interval to default time.

Examples

The following is an example to clear the interval time in hi-res container:

```
prometheus scrape-interval planning clear
```

prometheus scrape-interval planning show

Displays the configured scrape period to retain Prometheus data.

Syntax

prometheus scrape-interval planning show

Command Mode

OPERATIONAL

VNFs

DRA and Binding

Command Usage

The **prometheus scrape-interval planning show** command displays the configured scrape interval time to retain Prometheus data for longer time.

Examples

The following is an example to display the newly configured scrape interval:

```
prometheus scrape-interval planning show
```

prometheus scrape-interval trending config scrape_interval

Modifies the default scrape period to user specific scrape period for trending to retain Prometheus data.

Syntax

prometheus scrape-interval trending config scrape_interval *interval_time interval time in seconds*

Command Parameters

Table 113: Parameter Description

Command Parameter	Description
Interval time	Set new interval time for the trending container of prometheus.

Command Mode

OPERATIONAL

VNFs

DRA and Binding

Command Usage

The **prometheus scrape-interval trending config scrape_interval** command configures the interval time for the trending container to retain Prometheus data.

Examples

The following is an example to modify the interval time of 60s:

```
prometheus scrape-interval trending config scrape_interval 60s
```

prometheus scrape-interval trending clear

Clears the configured scrape interval to retain Prometheus data and reverts the interval time to default scrape interval.

Syntax

prometheus scrape-interval trending clear

Command Mode

OPERATIONAL

VNFs

DRA and Binding

Command Usage

The **prometheus scrape-interval trending clear**clears the configured scrape interval time to retain Prometheus data and reverts the scrape interval to default time

Examples

The following is an example to clear the interval time in trending container:

```
prometheus scrape-interval trending clear
```

prometheus scrape-interval trending show

Displays the configured scrape period to retain Prometheus data.

Syntax

prometheus scrape-interval trending show

Command Mode

OPERATIONAL

VNFs

DRA and Binding

Command Usage

The **prometheus scrape-interval trending show** command displays the configured scrape interval time to retain Prometheus data for longer time.

Examples

The following is an example to display the newly configured scrape interval:

```
prometheus scrape-interval trending show
```

revert

Used to copy running configuration into current configuration.

Syntax

```
revert
```

Command Mode

CONFIG

VNFs

All

Command Usage

Use the revert command to copy running configuration into the current configuration.

Examples

The following is an example:

```
admin@orchestrator[an-master](config)#revert
```

rollback configuration

Used to rollback the running configuration to a previous configuration.

Syntax

```
rollback configuration <commit-id>
```

Command Mode

CONFIG

VNFs

All

Command Usage

- Each time the commit command is entered, a commit ID is assigned to the new configuration. You can revert the system to the configuration of a previous commit ID with the rollback configuration command.

- The system stores a limited number of old configurations. The number of old configurations to store is configured in the `confd.conf` file. If more configurations are stored than the configured number, then the oldest configuration is removed before creating a new one.
- The most recently committed configuration (the running configuration) is number 0, the next most recent 1, and so on.
- The files are called rollback0 - rollbackX, where X is the maximum number of saved committed configurations.
- Use `show configuration commit list` to display a list of the commit IDs available for rollback operations.

```
show configuration commit list
2018-10-15 09:58:21
SNo. ID       User       Client      Time Stamp          Label       Comment
~~~~ ~~       ~~~~       ~~~~~~      ~~~~~~~~~~          ~~~~~       ~~~~~~~
0    10012    admin      cli         2018-10-15 09:57:59
```

Examples

The following is an example:

```
rollback configuration 0
```

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 114: 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.

Command Parameter	Description
host-network (optional)	Bind to the host network. Default is to the overlay network.
volume (optional)	Volume mounts in the format is as follows: <host path>:<docker path>. Additional mounts are separated by ",".
port-mapping (optional)	Port mapping of the format is as follows: <external>:<internal>. Additional mounts are separated by ",".
cap-add (optional)	Linux capabilities to add to the container. Additional mounts are separated by ",".
scheduling-slot (optional)	Scheduling slot to start the container (for all containers). Use the **show running-config docker engine** command to view list of scheduling slots.
environment (optional)	Environment variables to export into the container in the format given below: <KEY>=<VALUE> Additional mounts are separated by ",".

Command Mode

CONFIG

VNFs

All

Command Usage

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

scheduling vm-target

Calculates a vm-target for an external scaling system.

Syntax

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

```
no scheduling vm-target name
```

Command Parameters

Table 115: Parameter Description

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

Command Mode

CONFIG

VNFs

All

Command Usage

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

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

show alert status

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

Syntax

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

Command Parameters

Table 116: 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 117: 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 configuration

Used to display information about the current configuration session changes.

Syntax

```
show configuration
```

Command Mode

CONFIG

VNFs

All

Command Usage

- To display the configuration changes compared to the running configuration if any.
- Possible to display the configuration changes based on configuration component.

Examples

The following is an example:

```
admin@orchestrator[an-master](config)# aaa authentication users user test1 password ****
gid 100 homedir / ssh_keydir / uid 9340
admin@orchestrator[an-master](config-user-test1)#
admin@orchestrator[an-master](config)# show configuration
aaa authentication users user test1
 uid        9340
 gid        100
 password   $1$AWYdJW5S$g2wXilsJSumbCXPYgGzQW0
 ssh_keydir /
 homedir    /
!
```

show configuration commit

Used to display the changes made to the running configuration by previous configuration commits, a configuration commit, or for a range of configuration commits.

Use the `show configuration commit changes` command to display the information about the current configuration session changes.

Syntax

```
show configuration commit changes
```

```
show configuration commit list
```

Command Mode

CONFIG

VNFs

All

Command Usage

- Each time a configuration is committed with the commit command, the configuration commit operation is assigned a commit ID. The `show configuration commit changes` command displays the configuration changes made since the specified commit.
- To display a list of the available commit IDs, enter the `show configuration commit list` command.

Examples

The following is an example:

```
show configuration commit changes
!
! Created by: admin
! Date: 2018-10-15 09:57:59
! Client: cli
!
aaa authentication users user anil
 uid        9340
 gid        100
 password   $1$7aB1WW0D$3ln7YEGkLeTjWHoK2cVOE/
 ssh_keydir /
 homedir    /
!
```

```
show configuration commit list
2018-10-15 11:20:39
SNo. ID       User       Client      Time Stamp          Label       Comment
~~~~ ~~       ~~~~       ~~~~~~      ~~~~~~~~~~          ~~~~~       ~~~~~~~
0    10012    admin      cli         2018-10-15 09:57:59
```

show configuration rollback

Used to display changes that are made by the rollback configuration command. To display the list of rollback commit IDs, use the `show configuration rollback changes` command.

Syntax

```
show configuration rollback changes
```

Command Mode

ALL

VNFs

All

Command Usage

Use `show configuration rollback changes` command to display changes that are made by the rollback configuration command.

Note The most recent commits are retained by the system. As new commit IDs are added, the oldest commit IDs are discarded and are no longer available for rollback operations.

Examples

The following is an example:

```
show configuration rollback changes 0
no aaa authentication users user test1
```

show control-plane remote-peer-policy

Used to display the configured control plane remote peer policy.

Syntax

```
show control-plane remote-peer-policy
```

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

This command is used to display the current configured control plane remote peer policy in DRA.

Example

```
admin@orchestrator[vpas-A-dra-master-0]# show control-plane remote-peer-policy
Mated System: system-02
Accept remote peers for diameter applications : All
All Systems:
Accept remote peers for diameter applications :  Rx
```

show database

`show database status` displays the currently configured database clusters members.

`show database parallel-upgrade-plan` is used to print the parallel upgrade plan appropriate for database shard layout across nodes. If parallel upgrade option is selected, all the nodes in a batch are upgraded in parallel.

`show database parallel-upgrade-plan-details` is used to print the parallel upgrade plan with details of shards and servers selected in each batch. Orchestrator ensures that the members of the same shard are scheduled

in different batches to minimize the impact on the shards during parallel upgrade. You can use this command to review the plan and assess the impact of performing a parallel upgrade of DB cluster.

Syntax

```
show database status
show database parallel-upgrade-plan
show database parallel-upgrade-plan-details
```

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
```

Table 118: Output Description

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

Command Parameter	Description
Status	The current status of the mongo process. Valid states are: • CONNECTED – The mongo router is connected to the config servers • NOT_CONNECTED – The mongo router is not connected to the config servers • NO_CONNECTION – The process is not up or is not monitored • STARTUP – The DB node is in the STARTUP mode • PRIMARY – The DB node is the current PRIMARY • SECONDARY – The DB node is a SECONDARY node • RECOVERING – The DB node is currently RECOVERING from a restart or other failure • STARTUP2 – The DB node is in STARTUP2 mode • UNKNOWN – The DB node is in an UNKNOWN state • ARBITER – The DB node is currently an active ARBITER • NOT_INITIALIZED – The DB node is not initialized and pending initialization • Unable to get status - Displays this response when the consul agent service is not running or in a bad state on any of the mongo-status containers.
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.

To print the parallel upgrade plan appropriate for database shard layout across nodes.

```
admin@orchestrator[master-6]# show database parallel-upgrade-plan
BATCH  MODULE          HOST              ADDRESS
-------------------------------------------------------------------------------
1      mongo-node-101  master-6          172.20.27.36
1      mongo-node-102  control-7         172.20.27.40
1      mongo-node-103  control-8         172.20.27.39
1      mongo-node-104  persistence-db-3  2003:3030:27c1:913:250:56ff:fea6:53
2      mongo-node-105  persistence-db-4  2003:3030:27c1:913:250:56ff:fea6:54
```

To print the parallel upgrade plan with details of shards and servers selected in each batch.

```
admin@orchestrator[master-6]# show database parallel-upgrade-plan-details
BATCH  MODULE          HOST              ADDRESS                                 PORT   CLUSTER
 NAME     SHARD    SERVER           STATUS
-------------------------------------------------------------------------------------------------
1      mongo-node-101  master-6          172.20.27.36                            27019
imsi-msisdn      shdb-3   imsi-msisdn      SECONDARY
1      mongo-node-102  control-7         172.20.27.40                            27019
session-ipv6-AB  shdb-3   session-ipv6-AB  SECONDARY
1      mongo-node-103  control-8         172.20.27.39                            27019
imsi-msisdn      shdb-2   imsi-msisdn      SECONDARY
1      mongo-node-104  persistence-db-3  2003:3030:27c1:913:250:56ff:fea6:53  27017
imsi-msisdn      shard-1  server-c         SECONDARY
1      mongo-node-104  persistence-db-3  2003:3030:27c1:913:250:56ff:fea6:53  27018
imsi-msisdn      shard-2  server-c         SECONDARY
1      mongo-node-104  persistence-db-3  2003:3030:27c1:913:250:56ff:fea6:53  27021
imsi-msisdn      shard-3  server-c         PRIMARY
1      mongo-node-104  persistence-db-3  2003:3030:27c1:913:250:56ff:fea6:53  27020
imsi-msisdn      shard-4  server-c         SECONDARY
1      mongo-node-104  persistence-db-3  2003:3030:27c1:913:250:56ff:fea6:53  27022
session-ipv6-AB  shard-5  server-c         SECONDARY
1      mongo-node-104  persistence-db-3  2003:3030:27c1:913:250:56ff:fea6:53  27023
session-ipv6-AB  shard-6  server-c         SECONDARY
1      mongo-node-104  persistence-db-3  2003:3030:27c1:913:250:56ff:fea6:53  27024
session-ipv6-AB  shard-7  server-c         PRIMARY
1      mongo-node-104  persistence-db-3  2003:3030:27c1:913:250:56ff:fea6:53  27025
session-ipv6-AB  shard-8  server-c         SECONDARY
2      mongo-node-105  persistence-db-4  2003:3030:27c1:913:250:56ff:fea6:54  27017
imsi-msisdn      shard-1  server-d         SECONDARY
2      mongo-node-105  persistence-db-4  2003:3030:27c1:913:250:56ff:fea6:54  27018
imsi-msisdn      shard-2  server-d         SECONDARY
2      mongo-node-105  persistence-db-4  2003:3030:27c1:913:250:56ff:fea6:54  27021
imsi-msisdn      shard-3  server-d         SECONDARY
2      mongo-node-105  persistence-db-4  2003:3030:27c1:913:250:56ff:fea6:54  27020
imsi-msisdn      shard-4  server-d         PRIMARY
2      mongo-node-105  persistence-db-4  2003:3030:27c1:913:250:56ff:fea6:54  27022
session-ipv6-AB  shard-5  server-d         SECONDARY
2      mongo-node-105  persistence-db-4  2003:3030:27c1:913:250:56ff:fea6:54  27023
session-ipv6-AB  shard-6  server-d         SECONDARY
2      mongo-node-105  persistence-db-4  2003:3030:27c1:913:250:56ff:fea6:54  27024
session-ipv6-AB  shard-7  server-d         SECONDARY
2      mongo-node-105  persistence-db-4  2003:3030:27c1:913:250:56ff:fea6:54  27025
session-ipv6-AB  shard-8  server-d         PRIMARY
```

show database cluster

Verifies difference between a CLI and Database (DB) configuration.

Syntax

show database cluster *db name* **status**

Command Mode

CONFIG

VNFs

All

Examples

The following is an example:

show database db-encryption-status

Displays the consul server TLS mode status.

Syntax

show database db-encryption-status

Command Mode

OPERATIONAL

VNFs

All

Examples

The following is an example:

```
show database db-encryption-status
```

show database details

The command displays the actual concurrent transactions values from mongo process.

Syntax

```
show database details
```

Command Mode

Operational

VNFs

Binding

Command Usage

This command displays the concurrent transactions values from mongo process.

Examples

Following is an example:

```
admin@orchestrator[vpas-A1-bind-master-1]# show database details
ADDRESS PORT CLUSTER NAME STATUS WRITE READ OPLOG-SIZE CACHE-SIZE
2606:ae00:3001:8311:172:16:244:7c 27020 imsi-msisdn PRIMARY 8 8 3215 6.4M
2606:ae00:3001:8311:172:16:244:7d 27020 imsi-msisdn SECONDARY 8 8 3215 8.5M
2606:ae00:3001:8311:172:16:244:9c 27020 imsi-msisdn SECONDARY 128 128 3215 65M
2606:ae00:3001:8311:172:16:244:9d 27020 imsi-msisdn SECONDARY 128 128 3215 103M
2606:ae00:3001:8311:172:16:244:7d 27021 imsi-msisdn PRIMARY 8 8 3215 6.4M
2606:ae00:3001:8311:172:16:244:7c 27021 imsi-msisdn SECONDARY 8 8 3215 7.5M
2606:ae00:3001:8311:172:16:244:9d 27021 imsi-msisdn SECONDARY 128 128 3215 27M
2606:ae00:3001:8311:172:16:244:9c 27021 imsi-msisdn SECONDARY 128 128 3215 100M
2606:ae00:3001:8311:172:16:244:7e 27022 imsi-msisdn SECONDARY 8 8 3215 13M
2606:ae00:3001:8311:172:16:244:7f 27022 imsi-msisdn PRIMARY 8 8 3215 12M
2606:ae00:3001:8311:172:16:244:9e 27022 imsi-msisdn SECONDARY 128 128 3215 83M
2606:ae00:3001:8311:172:16:244:9f 27022 imsi-msisdn SECONDARY 128 128 3215 152M
```

show running-config database cluster ipv6-zones-range

Displays the ipv6 zone ranges configured in vDRA.

Syntax

show running-config database cluster ipv6-zones-range

Command Mode

OPERATIONAL

VNFs

All

Examples

The following is an example:

```
admin@orchestrator[site3-master-db-0]# show running-config database cluster ipv6-zones-range
 | tab
NAME     NAME         NAME     START                END
-------------------------------------------------------------------------
imsi     IMSIZONE     IRAN421  2606:ae00:c780:0010  2606:ae00:c780:0050
session  SESSIONZONE1    IRAN423  2606:ae00:c780:0055  2606:ae00:c780:0060
         SESSIONZONE  IRAN424  2606:ae00:c780:0200  2606:ae00:c780:0300

admin@orchestrator[site3-master-db-0]#
```

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 119: 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 120: 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.

Parameter	Description
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 dra-distributor

Displays the output of ipvsadm (Virtual Server administration) from all distributor VMs.

Syntax

```
show dra-distributor [ daemon | list | rate | stats ]
```

Command Parameters

Table 121: Parameter Description

Command Parameter	Description
daemon	Displays the sync daemon status and multicast interface.
list	Lists the Distributor Service table.
rate	Displays rate information for connection, bytes, and packets per second of Distributor services.
stats	Displays statistic information of Distributor Services.

Command Mode

OPERATIONAL

VNFs

All

Examples

The following are examples:

```
show dra-distributor list
================================================================================
    dra-distributor stats for  vpas-A-dra-distributor-client-a
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  172.16.241.10:3868 wlc
```

```
  -> 172.16.241.3:3868            Route   1      6          0
  -> 172.16.241.4:3868            Route   1      7          0
  -> 172.16.241.5:3868            Route   1      6          0
  -> 172.16.241.6:3868            Route   1      6          0
TCP  172.16.241.74:3868 wlc
  -> 172.16.241.67:3868           Route   1      1          0
  -> 172.16.241.68:3868           Route   1      1          0
  -> 172.16.241.69:3868           Route   1      1          0
  -> 172.16.241.70:3868           Route   1      1          0
TCP  [2606:ae00:3001:8311:172:16:241:109]:3868 wlc
  -> [2606:ae00:3001:8311:172:16:241:102]:3868 Route   1      5          0
  -> [2606:ae00:3001:8311:172:16:241:103]:3868 Route   1      5          0
  -> [2606:ae00:3001:8311:172:16:241:104]:3868 Route   1      5          0
  -> [2606:ae00:3001:8311:172:16:241:105]:3868 Route   1      9          0
================================================================================
    dra-distributor stats for  vpas-A-dra-distributor-client-b
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  172.16.241.10:3868 wlc
  -> 172.16.241.3:3868            Route   1      6          0
  -> 172.16.241.4:3868            Route   1      7          0
  -> 172.16.241.5:3868            Route   1      6          0
  -> 172.16.241.6:3868            Route   1      6          0
TCP  172.16.241.74:3868 wlc
  -> 172.16.241.67:3868           Route   1      1          0
  -> 172.16.241.68:3868           Route   1      1          0
  -> 172.16.241.69:3868           Route   1      1          0
  -> 172.16.241.70:3868           Route   1      1          0
TCP  [2606:ae00:3001:8311:172:16:241:109]:3868 wlc
  -> [2606:ae00:3001:8311:172:16:241:102]:3868 Route   1      5          0
  -> [2606:ae00:3001:8311:172:16:241:103]:3868 Route   1      5          0
  -> [2606:ae00:3001:8311:172:16:241:104]:3868 Route   1      5          0
  -> [2606:ae00:3001:8311:172:16:241:105]:3868 Route   1      9          0
================================================================================
    dra-distributor stats for  vpas-A-dra-distributor-server-a
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  172.16.242.10:3868 wlc
  -> 172.16.242.3:3868            Route   1      3          0
  -> 172.16.242.4:3868            Route   1      3          0
  -> 172.16.242.5:3868            Route   1      3          0
  -> 172.16.242.6:3868            Route   1      3          0
TCP  172.16.242.138:3868 wlc
  -> 172.16.242.131:3868          Route   1      5          0
  -> 172.16.242.132:3868          Route   1      4          0
  -> 172.16.242.133:3868          Route   1      4          0
  -> 172.16.242.134:3868          Route   1      4          0
================================================================================
    dra-distributor stats for  vpas-A-dra-distributor-server-b
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  172.16.242.10:3868 wlc
  -> 172.16.242.3:3868            Route   1      3          0
  -> 172.16.242.4:3868            Route   1      3          0
  -> 172.16.242.5:3868            Route   1      3          0
  -> 172.16.242.6:3868            Route   1      3          0
TCP  172.16.242.138:3868 wlc
  -> 172.16.242.131:3868          Route   1      5          0
  -> 172.16.242.132:3868          Route   1      4          0
  -> 172.16.242.133:3868          Route   1      4          0
  -> 172.16.242.134:3868          Route   1      4          0

show dra-distributor daemon
================================================================================
    dra-distributor stats for  vpas-A-dra-distributor-client-a
```

```
backup sync daemon (mcast=ens160, syncid=201)
================================================================================================
    dra-distributor stats for  vpas-A-dra-distributor-client-b
backup sync daemon (mcast=ens160, syncid=201)
================================================================================================
    dra-distributor stats for  vpas-A-dra-distributor-server-a
backup sync daemon (mcast=ens160, syncid=202)
================================================================================================
    dra-distributor stats for  vpas-A-dra-distributor-server-b
backup sync daemon (mcast=ens160, syncid=202)
```

```
show dra-distributor rate
================================================================================================
    dra-distributor stats for  vpas-A-dra-distributor-client-a
Prot LocalAddress:Port                 CPS    InPPS   OutPPS    InBPS   OutBPS
  -> RemoteAddress:Port
TCP  172.16.241.10:3868                  0    35080        0 17784626        0
  -> 172.16.241.3:3868                   0     7753        0  3856300        0
  -> 172.16.241.4:3868                   0     9521        0  4867718        0
  -> 172.16.241.5:3868                   0     7249        0  3607296        0
  -> 172.16.241.6:3868                   0    10557        0  5453342        0
TCP  172.16.241.74:3868                  0     2896        0  1269265        0
  -> 172.16.241.67:3868                  0      740        0   317735        0
  -> 172.16.241.68:3868                  0      824        0   321847        0
  -> 172.16.241.69:3868                  0      550        0   309638        0
  -> 172.16.241.70:3868                  0      782        0   320007        0
TCP  [2606:ae00:3001:8311:172:16:241:109]:3868        0    18551        0 17887169
0
  -> [2606:ae00:3001:8311:172:16:241:102]:3868        0     3714        0  3581895
0
  -> [2606:ae00:3001:8311:172:16:241:103]:3868        0     4037        0  3878454
0
  -> [2606:ae00:3001:8311:172:16:241:104]:3868        0     4012        0  3877344
0
  -> [2606:ae00:3001:8311:172:16:241:105]:3868        0     6789        0  6549476
0
================================================================================================
    dra-distributor stats for  vpas-A-dra-distributor-client-b
Prot LocalAddress:Port                 CPS    InPPS   OutPPS    InBPS   OutBPS
  -> RemoteAddress:Port
TCP  172.16.241.10:3868                  0        0        0        0        0
  -> 172.16.241.3:3868                   0        0        0        0        0
  -> 172.16.241.4:3868                   0        0        0        0        0
  -> 172.16.241.5:3868                   0        0        0        0        0
  -> 172.16.241.6:3868                   0        0        0        0        0
TCP  172.16.241.74:3868                  0        0        0        0        0
  -> 172.16.241.67:3868                  0        0        0        0        0
  -> 172.16.241.68:3868                  0        0        0        0        0
  -> 172.16.241.69:3868                  0        0        0        0        0
  -> 172.16.241.70:3868                  0        0        0        0        0
TCP  [2606:ae00:3001:8311:172:16:241:109]:3868        0        0        0        0
0
  -> [2606:ae00:3001:8311:172:16:241:102]:3868        0        0        0        0
0
  -> [2606:ae00:3001:8311:172:16:241:103]:3868        0        0        0        0
0
  -> [2606:ae00:3001:8311:172:16:241:104]:3868        0        0        0        0
0
  -> [2606:ae00:3001:8311:172:16:241:105]:3868        0        0        0        0
0
================================================================================================
    dra-distributor stats for  vpas-A-dra-distributor-server-a
Prot LocalAddress:Port                 CPS    InPPS   OutPPS    InBPS   OutBPS
  -> RemoteAddress:Port
TCP  172.16.242.10:3868                  0    29969        0 19567201        0
```

```
  -> 172.16.242.3:3868                    0     7363        0  4884850        0
  -> 172.16.242.4:3868                    0     7435        0  4885241        0
  -> 172.16.242.5:3868                    0     7636        0  4911014        0
  -> 172.16.242.6:3868                    0     7534        0  4886099        0
TCP  172.16.242.138:3868                  0    24373        0  8103149        0
  -> 172.16.242.131:3868                  0     5940        0  1677292        0
  -> 172.16.242.132:3868                  0     8316        0  3543717        0
  -> 172.16.242.133:3868                  0     4823        0  1429692        0
  -> 172.16.242.134:3868                  0     5293        0  1452448        0
================================================================================
    dra-distributor stats for  vpas-A-dra-distributor-server-b
Prot LocalAddress:Port                CPS    InPPS   OutPPS    InBPS   OutBPS
  -> RemoteAddress:Port
TCP  172.16.242.10:3868                   0        0        0        0        0
  -> 172.16.242.3:3868                    0        0        0        0        0
  -> 172.16.242.4:3868                    0        0        0        0        0
  -> 172.16.242.5:3868                    0        0        0        0        0
  -> 172.16.242.6:3868                    0        0        0        0        0
TCP  172.16.242.138:3868                  0        0        0        0        0
  -> 172.16.242.131:3868                  0        0        0        0        0
  -> 172.16.242.132:3868                  0        0        0        0        0
  -> 172.16.242.133:3868                  0        0        0        0        0
  -> 172.16.242.134:3868                  0        0        0        0        0

show dra-distributor stats
================================================================================
    dra-distributor stats for  vpas-A-dra-distributor-client-a
Prot LocalAddress:Port              Conns   InPkts  OutPkts  InBytes OutBytes
  -> RemoteAddress:Port
TCP  172.16.241.10:3868                 5  130888K        0   67428M        0
  -> 172.16.241.3:3868                  1 28763786        0   14532M        0
  -> 172.16.241.4:3868                  2 34872671        0   17887M        0
  -> 172.16.241.5:3868                  1 26758954        0   13554M        0
  -> 172.16.241.6:3868                  1 37715757        0   19818M        0
TCP  172.16.241.74:3868                 1  9892533        0    4791M        0
  -> 172.16.241.67:3868                 0  2535586        0    1206M        0
  -> 172.16.241.68:3868                 0  2627786        0    1208M        0
  -> 172.16.241.69:3868                 1  1940733        0    1058M        0
  -> 172.16.241.70:3868                 0  2578653        0    1208M        0
TCP  [2606:ae00:3001:8311:172:16:241:109]:3868        5 70270305        0   68098M
0
  -> [2606:ae00:3001:8311:172:16:241:102]:3868        0 14039247        0   13718M
0
  -> [2606:ae00:3001:8311:172:16:241:103]:3868        0 15233935        0   14707M
0
  -> [2606:ae00:3001:8311:172:16:241:104]:3868        0 15271681        0   14903M
0
  -> [2606:ae00:3001:8311:172:16:241:105]:3868        5 24425635        0   23490M
0
================================================================================
    dra-distributor stats for  vpas-A-dra-distributor-client-b
Prot LocalAddress:Port              Conns   InPkts  OutPkts  InBytes OutBytes
  -> RemoteAddress:Port
TCP  172.16.241.10:3868                25    3046M        0    1577G        0
  -> 172.16.241.3:3868                  5  575759K        0     295G        0
  -> 172.16.241.4:3868                  5  568825K        0     288G        0
  -> 172.16.241.5:3868                  5  564563K        0     282G        0
  -> 172.16.241.6:3868                  5  534960K        0     273G        0
TCP  172.16.241.74:3868                 4  172396K        0   83111M        0
  -> 172.16.241.67:3868                 1 41803986        0   20709M        0
  -> 172.16.241.68:3868                 1 45090996        0   20883M        0
  -> 172.16.241.69:3868                 0       11        0    10472        0
  -> 172.16.241.70:3868                 1 41847316        0   20711M        0
TCP  [2606:ae00:3001:8311:172:16:241:109]:3868       24    1635M        0    1581G
0
```

```
  -> [2606:ae00:3001:8311:172:16:241:102]:3868        5  306946K        0     298G
0
  -> [2606:ae00:3001:8311:172:16:241:103]:3868        5  357153K        0     342G
0
  -> [2606:ae00:3001:8311:172:16:241:104]:3868        5  339724K        0     332G
0
  -> [2606:ae00:3001:8311:172:16:241:105]:3868        4  300748K        0     290G
0
================================================================================
    dra-distributor stats for  vpas-A-dra-distributor-server-a
Prot LocalAddress:Port               Conns   InPkts  OutPkts  InBytes OutBytes
  -> RemoteAddress:Port
TCP  172.16.242.10:3868                 38  401155K        0    3050G        0
  -> 172.16.242.3:3868                   7  782570K        0     508G        0
  -> 172.16.242.4:3868                   7  789557K        0     513G        0
  -> 172.16.242.5:3868                   7  979702K        0     623G        0
  -> 172.16.242.6:3868                   8    1166M        0     759G        0
TCP  172.16.242.138:3868                55    3724M        0    1230G        0
  -> 172.16.242.131:3868                11  579165K        0     159G        0
  -> 172.16.242.132:3868                12    1079M        0     418G        0
  -> 172.16.242.133:3868                11  843389K        0     306G        0
  -> 172.16.242.134:3868                10  557000K        0     156G        0
================================================================================
    dra-distributor stats for  vpas-A-dra-distributor-server-b
Prot LocalAddress:Port               Conns   InPkts  OutPkts  InBytes OutBytes
  -> RemoteAddress:Port
TCP  172.16.242.10:3868                  0       0        0        0        0
  -> 172.16.242.3:3868                   0     785        0  1149048        0
  -> 172.16.242.4:3868                   0     817        0    43388        0
  -> 172.16.242.5:3868                   0    1178        0  2029844        0
  -> 172.16.242.6:3868                   0    2069        0  2386744        0
TCP  172.16.242.138:3868                 0   10926        0  3854392        0
  -> 172.16.242.131:3868                 0    2648        0   994176        0
  -> 172.16.242.132:3868                 0    1358        0   537100        0
  -> 172.16.242.133:3868                 0    2271        0   995296        0
  -> 172.16.242.134:3868                 0    1440        0   618544        0
```

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 122: 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
```

Note As the running-config backup does not save the backup of the logger-set configuration, take the backup using the following command:

show logger level | save /data/config/backup.txt.

Command Mode

OPERATIONAL

VNFs

All

Examples

The following is an example:

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

Table 123: Parameter Description

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

show ntp-server-status

Display the reachability status of the configured NTP servers.

Syntax

```
show ntp-server-status
```

Command Mode

Operational

VNFs

DRA and Binding

Command Usage

The command displays the reachability status of the configured NTP servers.

Examples

Following is an example that displays the reachability status of the configured NTP servers.

```
admin@orchestrator[site1-dra-master0]# show ntp-server-status
ADDRESS                  STATUS
---------------------------------------
10.197.98.241         REACHABLE
10.197.98.248         REACHABLE
10.197.98.249        NOT-REACHABLE
10.197.98.250        NOT REACHABLE
```

show orchestrator-database-status

Displays the MongoDB members database status running on orchestrator, orchestrator-backup-a, and orchestrator-backup-b containers.

Syntax

```
show orchestrator-database-status
```

Command Parameters

None

Command Mode

Operational

VNFs

All

Examples

The following example also shows a sample output:

```
admin@orchestrator[an-dbmaster]# show orchestrator-database-status
ADDRESS                  PORT        STATUS
-------------------------------------------------
orchestrator             27017     PRIMARY
orchestrator-backup-a    27017     SECONDARY
orchestrator-backup-b    27017     SECONDARY
```

Note In case any member is down or not able to retrieve its status, it is shown as NO_CONNECTION. For all other members, respective mongo status is displayed.

For example, if orchestrator-backup-a mongo member is down.

```
admin@orchestrator[an-dbmaster]# show orchestrator-database-status
ADDRESS                  PORT        STATUS
-------------------------------------------------
orchestrator             27017     PRIMARY
orchestrator-backup-a    27017     NO_CONNECTION
orchestrator-backup-b    27017     SECONDARY
```

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 write 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 running-config binding db-read-connection-settings

Displays the binding DB read connection settings.

Syntax

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

Command Mode

OPERATIONAL

VNFs

All

Examples

The following is an example:

```
scheduler# show running-config binding db-connection-settings | tab
                               MAX
```

```
            CONNECT  SOCKET   WAIT                ACCEPTABLE
            TIMEOUT  TIMEOUT  TIME   CONNECTIONS  LATENCY
BINDING     FOR      FOR      FOR    PER HOST     DIFFERENCE
TYPE        READ     READ     READ   FOR READ     FOR READ
-------------------------------------------------------------
ipv6        500      1000     500    10           5
imsiapn     500      1000     500    10           5
msisdnapn   500      1000     500    10           5
drasession  500      1000     500    10           5
```

show running-config binding shard-metadata-db-connection

Displays the binding shard metadata database connection.

Syntax

```
show running-config binding shard-metadata-db-connection
```

Command Mode

OPERATIONAL

VNFs

All

Examples

The following is an example:

```
scheduler# show running-config binding shard-metadata-db-connection | tab
SHARD
METADATA
BINDING
TYPE        HOST           PORT
---------------------------------
ipv6        193.1.163.114  27019
ipv6        193.1.163.115  27019
ipv6        193.1.163.164  27019
ipv4        193.1.163.114  27019
ipv4        193.1.163.115  27019
ipv4        193.1.163.164  27019
imsiapn     193.1.163.116  27019
imsiapn     193.1.163.25   27019
imsiapn     193.1.163.63   27019
imsiapn     193.1.163.65   27019
imsiapn     93.1.163.165   27019
msisdnapn   193.1.163.116  27019
msisdnapn   193.1.163.25   27019
msisdnapn   193.1.163.63   27019
msisdnapn   193.1.163.65   27019
msisdnapn   93.1.163.165   27019
drasession  193.1.163.114  27019
drasession  193.1.163.115  27019
drasession  193.1.163.164  27019
```

show running-config mfa-cli enable

Displays the configured MFA for orchestrator in vDRA.

Syntax

show running-config mfa-cli enable

Command Mode

OPERATIONAL

VNFs

All

Examples

The following is an example:

```
admin@orchestrator[TEST-Binding-master]# show running-config mfa-cli enable
mfa-cli enable user-id 16181
```

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 124: 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 snmp-server-status

Display the reachability status of the configured SNMP servers.

Syntax

```
show snmp-server-status
```

Command Mode

Operational

VNFs

DRA and Binding

Command Usage

The command displays the reachability status of the configured SNMP servers.

Examples

Following is an example that displays the reachability status of the configured SNMP servers.

```
admin@orchestrator[site1-dra-master0]# show snmp-server-status
ADDRESS                  STATUS
---------------------------------------
```

```
10.197.98.243         REACHABLE
10.197.98.241       NOT-REACHABLE
```

show system diagnostics

Shows the current diagnostics.

Syntax

There are no arguments for this command.

Command Mode

OPERATIONAL

VNFs

All

Command Parameters

Table 125: 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 126: 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 224.

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 127: 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

Shows 100% if the minimum set of containers are running for the system to operate.

A system status of 100% does not guarantee the following:

- The system is fully configured through the CLI or Policy Builder.
- All redundant VMs are UP. For example, a worker VM, and a distributor VM.
- Distributor VMs are UP.

To verify a system is healthy and all desired VMs are active, execute the following commands:

- `show docker engine`
- `show system status`
- `show system diagnostics`
- `show docker service`
- `show alerts`

Syntax

```
show system status
```

Command Parameters

None

Command Mode

OPERATIONAL

VNFs

All

Examples

The following example also shows a sample output:

```
scheduler# show system status
```

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 128: 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 129: 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 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 130: 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 131: 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 132: 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.
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]))"
!
```

Storage Health Check Service Commands

show storage-health-check service

Displays the health check settings. The following are default values:

Interval = 3 seconds

Failover Hold Time = 30 seconds

Syntax

```
admin@orchestrator[vPAS-A-master]# show storage-health-check service
Parameter           Value
------------------  -------
```

```
enable              true
failover-hold-time  10
interval            2
```

storage-health-check service <enable | disable | restart>

enable – Enables storage health check on diameter nodes

disable – Disables storage health check on diameter nodes

restart – Restarts storage health check on diameter nodes. Restart needs to be performed if health check settings are modified after enabling the service.

Configuring Storage Health Check Settings

The following commands can be used to configure storage health check settings.

```
storage-health-check set interval <value in seconds>
```

```
storage-health-check set failover-hold-time <value in seconds>
```

```
storage-health-check clear interval
```

Reset to default.

```
storage-health-check clear failover-hold-time
```

Reset to default.

Applying Configuration Changes

If the interval or failover-hold-time is updated after enabling health check service, then the changes does not automatically take effect. The service needs to be restarted for the changes to take effect by using the following command:

```
storage-health-check service restart
```

If the configuration is updated prior to enabling the service, enabling the service applies the latest settings.

```
storage-health-check service enable
```

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 230 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 217 command should be issued to revert this software to the previous version.

system-config get-cpu-priority

Used to display information about the CPU priority assigned to DRA application process and healthchecks.

Syntax

```
system-config get-cpu-priority
```

Command Mode

CONFIG

VNFs

All

Command Usage

Each time the command is entered the CPU priority assigned to DRA application process and healthcheck is returned.

Examples

The following is an example:

```
admin@orchestrator[sk-master0](config)# system-config get-cpu-priority
cpu_priority -20
admin@orchestrator[sk-master0](config)#
```

system-config set-cpu-priority

Specifiy the CPU priority value for the DRA application process and healtcheck.

Syntax

system-config set-cpu-priority *value*

Command Mode

CONFIG

VNFs

All

Command Usage

Each time the command is entered the value provided is assigned as CPU priority for the DRA application process and healtcheck.

Examples

The following is an example:

```
admin@orchestrator[sk-master0](config)# system-config set-cpu-priority cpu_priority ?
Possible completions:
<int, -20 .. 19>
admin@orchestrator[sk-master0](config)# system-config set-cpu-priority cpu_priority -10
admin@orchestrator[sk-master0](config)# system-config get-cpu-priority
cpu_priority -10
admin@orchestrator[sk-master0](config)#
```

system downgrade

Downgrades the system to a older software version.

Syntax

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

Command Mode

OPERATIONAL

VNFs

All

Command Parameters

Table 133: Parameter Description

Command Parameter	Description
version	The new software version to install into the system.
consul-downgrade	consul-downgrade is an optional parameter. For more information , see consul-downgrade, on page 218.

consul-downgrade

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

For example, if you are upgrading from 20.2.0.release to 21.x.0.release, snapshot name is *21.x.0.release*.

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

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

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

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

Example:

1. When upgrading to any new version (for example, from v1 to v2), it takes consul snapshot as `/data/orchestrator/config/snapshot-consul/v2`.
2. During downgrade (for example, from v2 to v1), provide snapshot name in system-downgrade command.

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

Command Usage

The system downgrade command installs new software on the system using a rolling downgrade approach to minimize service interruption. Care must be taken to ensure that the system downgrade command is used when moving from a higher software version to a lower version of the software. The rolling downgrade upgrades the software modules in startup order. After the command is issued, the CLI disconnects while the CLI software is restarted. The CLI generally becomes available within 30 seconds. Once the CLI becomes available, the status of the upgrade can be monitored using the show scheduling status, on page 199 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
```

show fluent-bit configurations

Displays the status of fluent-bit configurations.

Syntax

```
show fluent-bit configurations
```

Note As the running-config backup does not save the backup of the log-forward configuration, take the backup using the following command:

```
show fluent-bit configurations | save /data/config/backup.txt.
```

Command Mode

OPERATIONAL

VNFs

All

Examples

The following is an example:

```
show fluent-bit configurations
Fluent-Bit configurations    Current Value
---------------------------  --------------------------------
OAM-ip                        x.x.x.x
OAM-port                     9200
backlog-mem-limit            2048M
elasticsearch-ip             x.x.x.x
elasticsearch-password       3300901EA069E81CE29D4F77DE3C85FA
elasticsearch-port           9400
elasticsearch-user           elastic
flush-interval               900
max-chunks-up                3500
```

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 134: 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 203.

system secrets remove-secret

Removes a secret from the system.

Syntax

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

Command Mode

OPERATIONAL

VNFs

All

Command Parameters

Table 135: 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 203.

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 136: 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 203.

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 137: 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 138: 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 139: 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 140: 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 141: Parameter Description

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

Command Mode

OPERATIONAL

VNFs

All

Command Usage

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

Examples

The following example loads and activates an ISO:

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

system start

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

Syntax

There are no arguments for this command.

Command Mode

OPERATIONAL

VNFs

All

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

system stop

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

Syntax

There are no arguments for this command.

Command Mode

OPERATIONAL

VNFs

All

Command Usage

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

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

system upgrade

Upgrades the system to a new software version.

The option `database-upgrade-parallel` enables parallel upgrade of database nodes. This option is provided to reduce the upgrade time without impacting the availability of DB cluster.

Syntax

```
system upgrade version <version>
system upgrade version <version> database-upgrade-parallel <true/false>
```

Command Mode

OPERATIONAL

VNFs

All

Command Parameters

Table 142: 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` command.

Examples

To trigger an upgrade as usual. Mongo nodes goes sequential upgrade.

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

To trigger a parallel upgrade for mongo-nodes.

```
system upgrade version 19.5.0 database-upgrade-parallel true
```

vip-failover

Used to move the VIP between directors/distributors.

Syntax

```
vip-failover <vip-name> <current-vip-host-vm> <vip-ip> <vip-tracking-service> [ timeout ]
```

Command Parameters

Table 143: Parameter Description

Command Parameter	Description
vip-name	The VIP name.
current-vip-host-vm	The hostname where the VIP is present.
vip-ip	The floating IP of the VIP.
vip-tracking-service	The tracking service of the VIP in the format "Service-ip:Service-port".
timeout	The optional timeout value in seconds.

Command Mode

OPERATIONAL

VNFs

DRA

Command Usage

Use this command to move the VIP between director or distributor.

Examples

The following example moves the VIP between director or distributor.

```
network vip-failover testvip an-dra-director-0 10.225.115.253 10.225.115.253-3868
VIP failover completed successfully
```