



Deploy CPS VMs

- [Deploy the VMs, on page 1](#)
- [Update Default Credentials, on page 4](#)
- [Initialize SVN Synchronization, on page 5](#)
- [External Port Matrix, on page 6](#)
- [Memory Reservation on VMs, on page 6](#)
- [Session Manager Configuration for Data Replication, on page 6](#)
- [Validate VM Deployment, on page 16](#)

Deploy the VMs

If there are large number of VMs in your CPS deployment it is recommended to perform a Manual Deployment for one VM (for test purposes). After the success of the first VM, then all VMs can be deployed using Automatic Deployment process.



Note During the VM deployment, do not perform any vCenter operations on the blades and VMs installed on them.

Build VM Images

Before deploying the VMs, build the VM images by executing the following command from the Cluster Manager VM:

```
/var/qps/install/current/scripts/build_all.sh
```

Sample Output

```
Building /etc/broadhop...
Copying to /var/qps/images/etc.tar.gz...
...
Copying wispr.war to /var/qps/images/wispr.war
Output images to /var/qps/images/
[root@hostname]#
```

Manual Deployment

This section describes the steps to deploy each VM in the CPS deployment individually. To deploy all of the VMs in parallel using a single command refer to [Automatic Deployment of All CPS VMs in Parallel, on page 2](#). To deploy a selective list of VMs in parallel using a single command refer to [Automatic Deployment of Selective CPS VMs in Parallel, on page 3](#).



Note Before proceeding, refer to [License Generation and Installation](#) to confirm you have installed the license correctly.

For each host that is defined in the Hosts tab of the CPS Deployment Template spreadsheet execute the following:



Note The following command uses the short alias name (qns01 qns02 etc.) as defined in the Hosts tab of the CPS Deployment Template. It will not work if you enter the full hostname.

```
/var/qps/install/current/scripts/deployer/deploy.sh $host
```

where, *\$host* is the short alias name and not the full host name.

For example,

```
./deploy.sh qns01 <==== passed
```

```
./deploy.sh NDC2BSND2QNS01 <==== failed
```



Important Newly deployed VM/VMs need to be shutdown cleanly and started with your preferred method to reserve memory:

1. To shutdown individual VM:

```
cd /var/qps/install/current/scripts/deployer
./deploy.sh <vm alias> --shutdownvm
```

2. Start the VM:

```
./deploy.sh <vm alias> --poweronvm
```

Automatic Deployment of All CPS VMs in Parallel

This section describes the steps to deploy all VMs in parallel in the CPS deployment.



Note Before proceeding, refer to *License Generation and Installation* to confirm you have installed the license correctly.

Execute the following command:

```
python /var/qps/install/current/scripts/deployer/support/deploy_all.py
```

The order in which VMs are deployed is managed internally.



Note The amount of time needed to complete the entire deployment process depends on the number of VMs being deployed as well as the hardware on which it is being deployed.

The following is a sample list of VM hosts deployed. The list varies according to the type of CPS deployment as well as the information you entered in the CPS Deployment Template.

- pcrfclient01
- pcrfclient02
- sessionmgr01
- sessionmgr02
- lb01
- lb02
- qns01
- qns02
- qns03
- qns04



Note To install the VMs using shared or single storage, you must use `/var/qps/install/current/scripts/deployer/deploy.sh $host` command.

For more information, refer to [Manual Deployment, on page 2](#).

Automatic Deployment of Selective CPS VMs in Parallel

This section describes the steps to deploy a selective list of VMs in parallel in the CPS deployment.



Note Before proceeding, refer to *License Generation and Installation* to confirm you have installed the license correctly.

Execute the following command:

```
python /var/qps/install/current/scripts/deployer/support/deploy_all.py --vms <filename-of-vms>
```

where, `<filename-of-vms>` is the name of the file containing the list of VMs such as:

```
pcrfclient01  
lb01
```

qns01



Note The amount of time needed to complete the entire deployment process depends on the number of VMs being deployed as well as the hardware on which it is being deployed.



Important After deployment of load balancer VM, verify monit service status by executing the following command on deployed Load Balancer (lb) VM:

```
/bin/systemctl status monit.service
```

If monit service on load balancer VM is not running, then execute the following command on that VM to start it:

```
/bin/systemctl start monit.service
```



Important Newly deployed VM/VMs need to be shutdown cleanly and started with your preferred method to reserve memory:

Shut down and start Selective CPS VMs in Parallel.

1. Use your preferred editor and create `/tmp/vm-list` file and add VMs which you want to shut down and start.
2. To shutdown VMs from the given list.

```
cd /var/qps/install/current/scripts/deployer/support
python deploy_all.py --vms /tmp/vm-list --poweroffvm
```



Note Make sure that all the VMs in the list are powered OFF by using the above command.

3. To start all the VMs in list.

```
python deploy_all.py --vms /tmp/vm-list --poweronvm
```

Update Default Credentials

The passwords for the users in an HA or GR deployment are not set by default. Before you can access the deployed VMs or CPS web interfaces, you must set these passwords.

Step 1 Log into the Cluster Manager VM as the `root` user. The default credentials are `root/Cps!^246`.

Step 2 Execute the `change_passwd.sh` script to set the password.

Note `change_passwd.sh` script can also be used to change the root user password on all VMs including Cluster Manager VM.

```
/var/qps/bin/support/change_passwd.sh
```

Note The `change_passwd.sh` script changes the password on all the VMs temporarily. You also need to generate an encrypted password. The encrypted password must be added in the `Configuration.csv` spreadsheet. To make the new password persistent, execute `import_deploy.sh`. If the encrypted password is not added in the spreadsheet and `import_deploy.sh` is not executed, then after running `reinit.sh` script, the `qns-svn` user takes the existing default password from `Configuration.csv` spreadsheet.

Step 3 When prompted, enter `qns`.

```
Enter username whose password needs to be changed: qns
```

Step 4 If password not set, then below user message will be displayed:

Currently password is not set, please change the password

If password exists, then user will be prompted for the current password

Enter current password:

Note You can create passphrase or password with the following limitations, when you create or change passwords:

- You can provide one uppercase letter, one lowercase letter, one digit and one special character
- You can provide minimum 8 characters length password and up to maximum password length 256.
- Default password expiry is set to 6 months.

Step 5 When prompted, enter and reconfirm the desired password for the `qns` user.

```
Enter new password:
```

```
Re-enter new password:
```

```
Changing password on $host...
```

```
Connection to $host closed.
```

```
Password for qns changed successfully on $host
```

Note If script prompts for `[installer] Login password for 'root':`, enter default password (**Cps!^246**).

Step 6 Repeat [Step 2, on page 4](#) to [Step 5, on page 5](#) to set or change the passwords for `root` and `qns-svn` users.

For more information about this and other CPS administrative commands, refer to the *CPS Operations Guide*.

Initialize SVN Synchronization

After the VMs are deployed, execute the following script from the `perfclient01` VM:

```
/var/qps/bin/support/start_svn_sync.sh
```

This command synchronizes the master/slave Policy Builder subversion repositories.

External Port Matrix

The following table lists the services and ports that CPS makes available to external users and applications. It is recommended that connectivity to these ports be granted from the appropriate networks that require access to the below services.

Table 1: External Port Matrix

Service	Common Port (For HA Environment)	Deprecated Port (For HA Environment)
Control Center	443	443
Policy Builder	443	7443
Grafana	443	9443
Unified API	443	8443
Custom Reference Data REST API	443	8443
HAProxy Status	5540	5540

For a full list of ports used for various services in CPS, refer to the *CPS Architecture Guide*, which is available by request from your Cisco Representative.

Memory Reservation on VMs

To avoid performance impact you must reserve all allocated memory to each CPS virtual machine. For more information, refer to [Reserving Memory on the Virtual Machines \(VMs\)](#).

Session Manager Configuration for Data Replication

Before you perform service configuration, configure the session managers in the cluster. The database must be up and running for the CPS software.



Note Perform the steps mentioned in the following sections from the Cluster Manager.

Guidelines for Choosing MongoDB Ports

The standard definition for supported replica-set is defined in the `mongoConfig.cfg` file.

Use the `/etc/broadhop/ha_mongoconfig_template` file to create the `/etc/broadhop/mongoConfig.cfg` and modify it to your requirements.



Note If you are using VIP for arbiter, it is always recommended to keep VIP and all mongod processes on pcrfclient02 (by default).

Consider the following guidelines for choosing MongoDB ports for replica-sets:

- Port must not be in use by any other application. To check whether the port is in use, login to VM on which replica-set is to be created and execute the following command:

```
netstat -lnp | grep <port_no>
```

If no process is using same port, then port can be chosen for replica-set for binding.

- Port number used should be greater than 1024 and not in ephemeral port range i.e, not in between following range :

```
net.ipv4.ip_local_port_range = 32768 to 61000
```

- While configuring MongoDB ports in a geographically redundant environment, there should be a difference of 100 ports between two respective sites. For example, consider there are two sites: Site1 and Site2. For Site1, if the port number used is 27717, then you can configure 27817 as the port number for Site2. This is helpful to identify a MongoDB member's site. By looking at first three digits, you can decide where the MongoDB member belongs to. However, this is just a guideline. You must avoid having MongoDB ports of two different sites to close to each other (for example, 27717 on Site-1 and 27718 on Site2).

Reason: The `build_set.sh` script fails when you create shards on the site (for example, Site1). This is because the script calculates the highest port number in the `mongoConfig` on the site where you are creating shards. This creates a clash between the replica-sets on both sites because the port number which it allocates might overlap with the port number of `mongoConfig` on other site (for example, Site2). This is the reason why there should be some gap in the port numbers allocated between both the sites.

Supported Databases

The replica-set script is used to create replica-sets for the following databases. For more information about the script, see [Script Usage, on page 9](#).

- session
- spr
- balance
- report
- audit
- admin

Prerequisites

- It is recommended to use the specific option for creating a single replica-set rather than `--all` option as it is easy to recreate it again if it fails to create.

- If recreating a replica-set on a production system, make sure to back up the database (Refer *CPS Backup and Restore Guide*).
- Auto Intelligent DB Operations (AIDO) server is running on Cluster Manager or third-party site Arbiter.
 - It is not active on third-party site Arbiter node, i.e., using `monit summary` you can see `aido_server` is running but in `/var/log/aido_server.log` you can see the following message:


```
AIDO server is not needed on arbiter/site
```
 - It pushes latest or updated `mongoConfig.cfg` file to all database members every 60 seconds interval.
 - It checks if any database member is UP and ready to join a replica-set. If Yes, then checks whether replica-set exist or not. If replica-set exists, then join as a member in the existing replica-set. If replica-set does not exist, then create new replica sets
 - Monit process name is `aido_server`.
 - AIDO server status can be checked by using `/etc/init.d/aido_server status` and `systemctl status aido_server`
 - Log rotate file is available at: `/etc/logrotate.d/aido_server`, size limit is 10 M and 5 rotation
- AIDO client is running on sessionmgr, prfclient and third-party site Arbiter.
 - `mongoConfig.cfg` file is received from AIDO servers (in GR, multiple AIDO servers are available).


```
mongoConfig.cfg file is available at: /var/aido
```

File name format is:
`/var/aido/mongoConfig.cfg.<<cluman-host-name>>-<<--cluman-eth0-IP-->>`

AIDO server pushes `mongoConfig.cfg` file to all database members i.e., AIDO clients.
 - AIDO client status can be checked by using `/etc/init.d/aido_client status` and `systemctl status aido_client`
 - Log rotate file is available at: `/etc/logrotate.d/aido_client`, size limit is 10 M and 5 rotation



Note You have to refer to `/etc/broadhop/ha_mongoconfig_template` file and use this file to create `/etc/broadhop/mongoConfig.cfg` file based on your requirements.

All the replica set members and required information like Host Name and port number arbiter host name and port number should be defined in `/etc/broadhop/mongoConfig.cfg` file.



Note Make sure all the replica set ports defined in the `mongoConfig.cfg` file are outside the range 32768 to 61000. For more information about the port range, refer to http://www.ncftp.com/ncftpd/doc/misc/ephemeral_ports.html.

The following example shows replica-set set04:

Table 2: Replica-set Example

[SPR-SET1]	[Beginning Set Name-Set No]
SETNAME=rep_set04	Set name i.e. rep_set04
ARBITER1=pcrfclient0127720	Arbiter VM host with port number
ARBITER_DATA_PATH=/var/data/sessions.4	Arbiter data directory
MEMBER1=sessionmgr0127720	Primary Site Member1
MEMBER2=sessionmgr0227720	Primary Site Member2
DATA_PATH=/var/data/sessions.4	Data Directory Path for members
[SPR-SET1-END]	[Closing Set Name-Set No]

Run the `/var/qps/install/current/scripts/build/build_etc.sh` script from the Cluster Manager to finalize `mongoConfig.cfg` file after AIDO automatically takes care of updating it.

`build_set.sh` script copies `/etc/broadhop/mongoConfig.cfg` file to `/var/www/html/images/mongoConfig.cfg` file.

Script Usage

`build_set.sh` script is used to verify replica-set creation.

Option to view help: `/var/qps/bin/support/mongo/build_set.sh --help`

```
build_set.sh --help
```

```
-----
Replica-set Configuration
-----
```

```
Usage: build_set.sh <--option1> <--option2> [--setname SETNAME] [--help]
option1: Database name
option2: Build operations (create, add or remove members)
option3: Use --setname SETNAME to build or alter a specific replica-set
         replica-set setnames are defined in the /etc/broadhop/mongoConfig.cfg file
```

```
The script applies to Database: session, spr, balance, report, portal, admin, audit and
bindings db replica-sets
```

```
Config Server: session_configs, spr_configs and bindings_configs db
```

```
replica-sets
```

```
--all           : Alias for all databases in the configuration
--create        : Create a replica-set if force option is given, else it just
validate
--create-asc    : Create a replica-set with set priority in the ascending format
if
                 force option is given, else it just validate
--create-des    : Create a replica-set with set priority in the descending format
if
                 force option is given, else it just validate
--add-members   : Add members to a replica-set if force option is given, else it
```

```

just validate
replica-set using the
--remove-members          : Remove specific members from a replica-set
                           For example, a non-active member
--remove-failed-members   : Remove failed/not reachable members from a replica-set
                           On occasion, replica-set members are not reachable due to network
                           issues
--remove-replica-set      : Remove a replica-set
--create-scripts          : Create init.d script for the replica-set members if force option
                           is given
--setname                  : The name of a replica-set as configured in
                           /etc/broadhop/mongoConfig.cfg
--force                    : This option can be used with create & add-members

```

Examples:

General operation

```

build_set.sh --all --create
build_set.sh --session --create
build_set.sh --session --create-asc
build_set.sh --session --create-des
build_set.sh --session --add-members
build_set.sh --session --remove-members
build_set.sh --session --remove-failed-members
build_set.sh --session --remove-replica-set
build_set.sh --session --create-scripts
build_set.sh --help

```

To perform build operations on a sepecific replica-set:

```

build_set.sh --spr --create --setname set04
build_set.sh --spr --create-asc --setname set04
build_set.sh --spr --create-des --setname set04
build_set.sh --spr --add-members --setname set04
build_set.sh --spr --remove-failed-members --setname set04
build_set.sh --spr --remove-replica-set --setname set04
build_set.sh --spr --create-scripts --setname set04

```

If you want to use build_set.sh to create replica-set then use option --force.



Note When you execute `execute build_set.sh <dbname> --remove_replica_set <setname>`, it creates a `/var/tmp/stopped-XXXX` (XXXX is the port number of the replica-set member) file on the respective sessionmgr and arbiter VM. If you want to recreate the same replica-set again on the same port then you have to manually remove the `/var/tmp/stopped-XXXX` file from respective sessionmgr or arbiter VM. AIDO monitors the `/var/tmp/stopped-XXXX` file on the VM and don't do any action on the replica-set member if file is present.

Guidelines for Adding Replica-sets

You must create the database replica-set members on the same VM and the same port on both sites.

For example: For session manager database, among four replica-set members (except arbiter), if `sessionmgr01:27717` and `sessionmgr02:27717` are two members of replica-set from SITE1, then choose `sessionmgr01:27717` and `sessionmgr02:27717` of SITE2 as other two replica-set members as shown in following example:

```
[SESSION-SET]
  SETNAME=set01
  OPLOG_SIZE=5120
  ARBITER1=SITE-ARB-sessionmgr05:27717
  ARBITER_DATA_PATH=/var/data/sessions.1/set1
  PRIMARY-MEMBERS
  MEMBER1=SITE1-sessionmgr01:27717
  MEMBER2=SITE1-sessionmgr02:27717
  SECONDARY-MEMBERS
  MEMBER1=SITE2-sessionmgr01:27717
  MEMBER2=SITE2-sessionmgr02:27717
  DATA_PATH=/var/data/sessions.1/set1
[SESSION-SET-END]
```

Defining a Replica-set

- Step 1** Update the `mongoConfig.cfg` file with the new replica-set.
- Step 2** Execute the following command from the Cluster Manager to finalize `mongoConfig.cfg` file after AIDO automatically takes care of updating it:
- ```
/var/qps/install/current/scripts/build/build_etc.sh
```
- Step 3** To verify replica-set has been created, run the `build_set.sh` command for the different replica-sets. The following table describes the commands for each type of replica set:

**Table 3: Replica-set Commands**

| Replica-set         | Command                                                                                                                                                                          |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Session Replica-set | <code>/var/qps/bin/support/mongo/build_set.sh --session</code>                                                                                                                   |
| SPR Replica-set     | SPR (USum) supports MongoDB hashed sharding.<br><code>/var/qps/bin/support/mongo/build_set.sh --spr</code>                                                                       |
| Balance Replica-set | <code>/var/qps/bin/support/mongo/build_set.sh --balance</code>                                                                                                                   |
| Report Replica-set  | <code>/var/qps/bin/support/mongo/build_set.sh --report</code>                                                                                                                    |
| Audit Replica-set   | <code>/var/qps/bin/support/mongo/build_set.sh --audit</code>                                                                                                                     |
| Admin Replica-set   | The ADMIN database holds information related to licensing, diameter end-points and sharding for runtime.<br>use.<br><code>/var/qps/bin/support/mongo/build_set.sh --admin</code> |

| Replica-set       | Command                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| WT_CACHESIZEGB    | <p>This parameter configures <b>wiredtiger</b> cache in GB on Session Manager VMs. The configured <b>WT_CACHESIZEGB</b> reflects in mongo processes as <b>--wiredTigerCacheSizeGB</b> parameter. This is an optional parameter.</p> <p>Default value: 2 GB</p> <p><b>Note</b> With <b>WiredTiger</b>, MongoDB utilizes both the WiredTiger internal cache and the filesystem cache. The default WiredTiger internal cache size is the larger of either 50% of (RAM - 1 GB), or 256 MB.</p> <p>For example, on a system with a total of 4 GB of RAM the WiredTiger cache can use 1.5GB of RAM (<math>0.5 * (4 \text{ GB} - 1 \text{ GB}) = 1.5 \text{ GB}</math>). Conversely, a system with a total of 1.25 GB of RAM can allocate 256 MB to the WiredTiger cache because that is more than half of the total RAM minus one gigabyte (<math>0.5 * (1.25 \text{ GB} - 1 \text{ GB}) = 128 \text{ MB} &lt; 256 \text{ MB}</math>).</p> |
| WT_CACHEARBSIZEGB | <p>This parameter configures <b>wiredtiger</b> cache in GB on arbiter VMs. The configured <b>WT_CACHEARBSIZEGB</b> will be reflected in mongo processes <b>--wiredTigerCacheSizeGB</b> parameter. This is an optional parameter.</p> <p>Default value: 1 GB</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |

Instead of the specific command described in table, you can also use the following command:

```
diagnostics.sh --get_replica_status
```

**Note** The installation logs are generated in the appropriate directories (`/var/log/broadhop/scripts/`) for debugging or troubleshooting purposes.

## Example of Replica set Creation

Here are some examples for replica-sets:

**Step 1** Log in to Cluster Manager.

**Step 2** Refer to `/etc/broadhop/ha_mongoconfig_template` file and use this file to create `/etc/broadhop/mongoConfig.cfg` file based on your requirements.

```
vi /etc/broadhop/mongoConfig.cfg

[SESSION-SET1]
SETNAME=set01
```

```

OPLOG_SIZE=1024
WT_CACHESIZEGB=2
WT_CACHEARBSIZEGB=1
ARBITER=pcrfclient01:27717
ARBITER_DATA_PATH=/var/data/sessions.1
MEMBER1=sessionmgr01:27717
MEMBER2=sessionmgr02:27717
DATA_PATH=/var/data/sessions.1
[SESSION-SET1-END]

[BALANCE-SET1]
SETNAME=set02
OPLOG_SIZE=1024
WT_CACHESIZEGB=2
WT_CACHEARBSIZEGB=1
ARBITER=pcrfclient01:27718
ARBITER_DATA_PATH=/var/data/sessions.2
MEMBER1=sessionmgr01:27718
MEMBER2=sessionmgr02:27718
DATA_PATH=/var/data/sessions.2
[BALANCE-SET1-END]

[REPORTING-SET1]
SETNAME=set03
OPLOG_SIZE=1024
WT_CACHESIZEGB=2
WT_CACHEARBSIZEGB=1
ARBITER=pcrfclient01:27719
ARBITER_DATA_PATH=/var/data/sessions.3
MEMBER1=sessionmgr01:27719
MEMBER2=sessionmgr02:27719
DATA_PATH=/var/data/sessions.3
[REPORTING-SET1-END]

[SPR-SET1]
SETNAME=set04
OPLOG_SIZE=1024
WT_CACHESIZEGB=2
WT_CACHEARBSIZEGB=1
ARBITER=pcrfclient01:27720
ARBITER_DATA_PATH=/var/data/sessions.4
MEMBER1=sessionmgr01:27720
MEMBER2=sessionmgr02:27720
DATA_PATH=/var/data/sessions.4
[SPR-SET1-END]

```

**Step 3** After defining the admin database details, rebuild etc.tar.gz.

```
/var/qps/install/current/scripts/build/build_etc.sh
```

### What to do next

After replica sets are created, you need to configure the priorities for the replica set members using `set_priority.sh` command. For more information on `set_priority.sh`, refer to *CPS Operations Guide*.

## Guidelines to Configure More than Seven Replica-set Members

If it is required to configure more than seven members (including arbiters), then data members must be defined as non-voting-members in `/etc/broadhop/mongoConfig.cfg` file.

Non-voting members allow you to add additional data members for read distribution beyond the maximum seven voting members.

To configure a member as non-voting, votes and priority value must be configured to 0.

This configuration is done by `build_set.sh` and `set_priority.sh` scripts. So, it is expected to have priority as 0 for non-voting-member.

For more information, see <https://docs.mongodb.com/manual/tutorial/configure-a-non-voting-replica-set-member/> (select appropriate mongo version).

### Configure Non-Voting Members

If there are total eight data members and one arbiter (i.e. total nine members), six must be defined as `MEMBER $n$`  and all other remaining data members must be defined as `NON-VOTING-MEMBER $n$`  in `/etc/broadhop/mongoConfig.cfg` file.

where,  $n$  in `MEMBER $n$`  and `NON-VOTING-MEMBER $n$`  represents number 1, 2, 3 and so on.

```
[SPR-SET1]
SETNAME=set04
OPLOG_SIZE=3072
ARBITER=site3-arbiter:27720
ARBITER_DATA_PATH=/var/data/sessions.4
PRIMARY-MEMBERS
MEMBER1=site1-sessionmgr01:27720
MEMBER2=site1-sessionmgr02:27720
MEMBER3=site1-sessionmgr03:27720
NON-VOTING-MEMBER4=site1-sessionmgr04:27720
SECONDARY-MEMBERS
MEMBER1=site2-sessionmgr01:27720
MEMBER2=site2-sessionmgr02:27720
MEMBER3=site2-sessionmgr03:27720
NON-VOTING-MEMBER4=site2-sessionmgr04:27720
DATA_PATH=/var/data/sessions.4
[SPR-SET1-END]
```




---

**Note** You can have only maximum seven voting members including arbiter which can be defined as `MEMBER $n$`  and `ARBITER $n$`  and all other member must be defined as `NON-VOTING-MEMBER $n$` .

---

## Session Cache Scaling

The session cache can be scaled by adding an additional sessionmgr VM (additional session replica-set). You must create separate administration database and the hostname and port should be defined in Policy Builder (cluster) as defined in the following sections:

- [Service Restart, on page 14](#)
- [Create Session Shards, on page 15](#)

### Service Restart

After mongo configuration is done successfully (The `build_set.sh` script gives the status of the mongo configuration after the configuration has been finished) from Cluster Manager, run `/var/qps/bin/control/restartall.sh` script.



**Caution** Executing `restartall.sh` will cause messages to be dropped.

After we modify `mongoconfig.cfg` file, we can run the `synconfig.sh` script to rebuild `etc.tar.gz` image and trigger each VM to pull and extract it.

```
/var/qps/bin/update/synconfig.sh
```

## Create Session Shards

**Step 1** From `pcrfclient01` or `pcrfclient02` VM, execute the following command:

```
session_cache_ops.sh --add-shard
```

The following screen prompts are displayed:

```
Session Sharding

Select type of session shard Default []
Hot Standby []
Sessionmgr pairs :
Session shards per pair :
```

**Step 2** Select either **Default** or **Hot Standby** by placing the cursor in the appropriate field and pressing `y`.

**Step 3** In Sessionmgr pairs, enter the name of the sessionmgr VM pairs separated by a colon (`:`) with port number.

Example: `sessionmgr01:sessionmgr02:27717`

If sharding is needed for multiple sessionmgr VMs, enter the sessionmgr VM name with port separated by a colon (`:`), with each pair separated by a colon (`:`).

Example: `sessionmgr01:sessionmgr02:27717,sessionmgr03:sessionmgr04:27717`

**Step 4** In Session shards per pair, enter the number of shards be added.

Example: `Session shards per pair: 4`

**Step 5** Login to ADMIN DB primary mongo sessionmgr VM using port number 27721 and execute the following commands to verify the shards:

```
mongo sessionmgr01:27721
set05:PRIMARY> use sharding
switched to db sharding
set05:PRIMARY> db.shards.find()
```

Example:

```
mongo sessionmgr01:27721
MongoDB shell version: 2.6.3
connecting to: sessionmgr01:27721/test
set05:PRIMARY> use sharding
switched to db sharding
set05:PRIMARY> db.shards.find()
{ "_id" : 1, "seed_1" : "sessionmgr01", "seed_2" : "sessionmgr02", "port" : 27717, "db" :
"session_cache", "online" : true, "count" : NumberLong(0), "lockTime" :
ISODate("2015-12-16T09:35:15.348Z"), "isLocked" : false, "lockedBy" : null }
{ "_id" : 2, "seed_1" : "sessionmgr01", "seed_2" : "sessionmgr02", "port" : 27717, "db" :
"session_cache_2", "online" : true, "count" : NumberLong(0), "backup_db" : false, "lockTime" :
ISODate("2015-12-16T09:35:06.457Z"), "isLocked" : false, "lockedBy" : null }
{ "_id" : 3, "seed_1" : "sessionmgr01", "seed_2" : "sessionmgr02", "port" : 27717, "db" :
```

```
"session_cache_3", "online" : true, "count" : NumberLong(0), "backup_db" : false, "lockTime" :
ISODate("2015-12-16T09:34:51.457Z"), "isLocked" : false, "lockedBy" : null }
{ "_id" : 4, "seed_1" : "sessionmgr01", "seed_2" : "sessionmgr02", "port" : 27717, "db" :
"session_cache_4", "online" : true, "count" : NumberLong(0), "backup_db" : false, "lockTime" :
ISODate("2015-12-16T09:35:21.457Z"), "isLocked" : false, "lockedBy" : null }
set05:PRIMARY>
```

## Verify CPS Sanity

From Cluster Manager, run `/var/qps/bin/diag/diagnostics.sh` script.



**Note** Currently, running `diagnostics.sh --ha_proxy` with `qns-admin` or `qns-su` user is not supported, It's only supported with `root` user.

## Validate VM Deployment

### Virtual Interface Validation

To verify that the `lbvip01` and `lbvip02` are successfully configured in `lb01` and `lb02`, perform the following steps:

- Step 1** SSH to `lb01`. The default credentials are `qns/cisco123`.
- Step 2** Check whether the virtual interface of the Policy Director (LB) is UP. Use `ifconfig` command to show the virtual interfaces are UP. If extra diameter interface were configured, verify the corresponding VIPs are up for the diameter interfaces.

## Basic Networking

From Cluster Manager, verify that you are able to ping all the hosts in the `/etc/hosts` file.

## Diagnostics and Status Check

The following commands can be used to verify whether the installation was successful or not:

- `diagnostics.sh`
- `about.sh`
- `list_installed_features.sh`
- `statusall.sh`





**Note** For more information on other CPS administrative commands, refer to *CPS Operations Guide*.

## diagnostics.sh

This command runs a set of diagnostics and displays the current state of the system. If any components are not running red failure messages will be displayed.

```
/var/qps/install/current/scripts/upgrade/reinit.sh
```

This command will prompt for reboot choice. Please select **Y** for the same and proceed.

### Syntax

```
/var/qps/bin/diag/diagnostics.sh -h
Usage: /var/qps/bin/diag/diagnostics.sh [options]
This script runs checks (i.e. diagnostics) against the various access, monitoring, and
configuration points of a running CPS system.
In HA/GR environments, the script always does a ping check for all VMs prior to any other
checks and adds any that fail the ping test to the IGNORED_HOSTS variable. This helps reduce
the possibility for script function errors.
NOTE: See /var/qps/bin/diag/diagnostics.ini to disable certain checks for the HA/GR env
persistently. The use of a flag will override the diagnostics.ini value.
```

Examples:

```
/var/qps/bin/diag/diagnostics.sh -q
/var/qps/bin/diag/diagnostics.sh --basic_ports --clock_skew -v
--ignored_hosts='portal01,portal02'
```

Options:

```
--basic_ports : Run basic port checks

For HA/GR: 80, 11211, 7070, 8080, 8081, 8090, 8182, 9091, 9092, and Mongo DB ports
based on /etc/broadhop/mongoConfig.cfg
--clock_skew : Check clock skew between lb01 and all vms (Multi-Node Environment only)
--diskspace : Check diskspace
--get_replica_status : Get the status of the replica-sets present in environment.
(Multi-Node Environment only)
--get_shard_health : Get the status of the sharded database information present in
environment. (Multi-Node Environment only)
--get_sharded_replica_status : Get the status of the shards present in environment.
(Multi-Node Environment only)
--ha_proxy : Connect to HAProxy to check operation and performance statistics, and ports
(Multi-Node Environment only)
http://lbvip01:5540/haproxy?stats
http://lbvip01:5540/haproxy-diam?stats
--help -h : Help - displays this help

--ignored_hosts : Ignore the comma separated list of hosts. For example
--ignored_hosts='portal01,portal02'
Default is 'portal01,portal02,portallb01,portallb02' (Multi-Node Environment only)
--ping_check : Check ping status for all VM
--qns_diagnostics : Retrieve diagnostics from CPS java processes
--qns_login : Check qns user passwordless login
--quiet -q : Quiet output - display only failed diagnostics

--redis : Run redis specific checks
--svn : Check svn sync status between pcrfclient01 & pcrfclient02 (Multi-Node Environment
only)
--tacacs : Check Tacacs server reachability
--swapspace : Check swap space
```

```

 --verbose -v : Verbose output - display *all* diagnostics (by default, some are grouped
for readability)
 --virtual_ips : Ensure Virtual IP Addresses are operational (Multi-Node Environment
only)
 --vm_allocation : Ensure VM Memory and CPUs have been allocated according to
recommendations

```

### Executable on VMs

- Cluster Manager and OAM (PCRFCLIENT) nodes

### Example

```

[root@pcrfclient01 ~]# diagnostics.sh
QNS Diagnostics
Checking basic ports (80, 7070, 27017, 27717-27720, 27749, 8080, 9091)...[PASS]
Checking qns passwordless logins on all boxes...[PASS]
Validating hostnames...[PASS]
Checking disk space for all VMs...[PASS]
Checking swap space for all VMs...[PASS]
Checking for clock skew...[PASS]
Retrieving QNS diagnostics from qns01:9045...[PASS]
Retrieving QNS diagnostics from qns02:9045...[PASS]
Checking HAProxy status...[PASS]
Checking VM CPU and memory allocation for all VMs...[PASS]
Checking Virtual IPs are up...[PASS]
[root@pcrfclient01 ~]#

```

## about.sh

This command displays:

- Core version
- Patch installed
- ISO version
- Feature version
- URLs to the various interfaces
- APIs for the deployment

This command can be executed from Cluster Manager or OAM (PCRFCLIENT).

### Syntax

```
/var/qps/bin/diag/about.sh [-h]
```

### Executable on VMs

- Cluster Manager
- OAM (PCRFCLIENT)

## list\_installed\_features.sh

This command displays the features and versions of the features that are installed on each VM in the environment.

### Syntax

```
/var/qps/bin/diag/list_installed_features.sh
```

### Executable on VMs

- All

## statusall.sh

This command displays whether the monit service and CPS services are stopped or running on all VMs. This script can be executed from Cluster Manager or OAM (PCRFCLIENT).

### Syntax

```
/var/qps/bin/control/statusall.sh
```

### Executable on VMs

- Cluster Manager
- pcrfclient01/02



---

**Note** Refer to *CPS Operations Guide* for more details about the output of this command.

---

## Web Application Validation

To verify that the CPS web interfaces are running navigate to the following URLs where *<lbvip01>* is the virtual IP address you defined for the lb01 VM.



---

**Note** Run the `about.sh` command from the Cluster Manager to display the actual addresses as configured in your deployment.

---

- **Policy Builder:** `https://<lbvip01>:7443/pb`  
Default credentials: `qns-svn/cisco123`
- **Control Center:** `https://<lbvip01>:443`  
Default credentials: `qns/cisco123`
- **Grafana:** `https://<lbvip01>:9443/grafana`  
Default credentials: —



---

**Note** You must create at least one Grafana user to access the web interface. Refer to the *Prometheus and Grafana* chapter of the *CPS Operations Guide* for steps to configure User Authentication for Grafana.

---

- **Unified API:** `http://<lbvip01>:8443/ua/soap`
- **CRD REST API:** `http://<lbvip01>:8443/custrefdata`

For more information related to CPS interfaces, refer to *CPS Operations Guide*.

## Supported Browsers

CPS supports the most recent versions of the following browsers:

- Firefox
- Chrome
- Safari
- Microsoft IE version 9 and above