



Deploy CPS VMs

- [Deploy the VMs, on page 1](#)
- [Update Default Credentials, on page 4](#)
- [Initialize SVN Synchronization, on page 4](#)
- [External Port Matrix, on page 5](#)
- [Memory Reservation on VMs, on page 5](#)
- [Configure Session Manager for Database Replication, on page 5](#)
- [Validate VM Deployment, on page 13](#)

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
```

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
```

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
```

Step 3 When prompted, enter `qns`.

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

Step 4 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 5 Repeat [Step 2, on page 4](#) to [Step 4, on page 4](#) 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 `pcrfclient01` VM:

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

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



Note You do not need to perform this step for AIO deployments.

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)	Port (for All-in-One Environment)
Control Center	443	443	8090
Policy Builder	443	7443	7070
Grafana	443	9443	80
Unified API	443	8443	8080
Custom Reference Data REST API	443	8443	8080
HAProxy Status	5540	5540	Not Applicable

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 \(VMs\)](#).

Configure Session Manager for Database Replication

Before service configuration can be done for the CPS system, the Session Managers in the cluster should be configured. CPS software needs the database to be available before functioning.



Note The steps mentioned in the following sections must be performed in the Cluster Manager.

Configuration

The standard definition for supported replica-set is defined in mongo configuration file. This configuration file is self-explanatory which contain replica set set-name hostname port number data file path etc.

Location: `/etc/broadhop/mongoConfig.cfg`



Important

While choosing mongo ports for replica-sets, consider the following:

- Port is not in use by any other application. To check it, 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 mongo ports in a GR 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 mongo member's site. By looking at first three digits, one can decide where the mongo member belongs to. However, this is just a guideline. You should avoid having mongo ports of two different sites to close to each other (for exampl, 27717 on Site-1 and 27718 on Site2).

Reason: The reason is that 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 clash between the replica-sets on both sites. Since 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 Database

Currently, replica-set script supports creation of replica-sets for following databases:

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

Prerequisite

- 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*).



Note 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 on 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/bin/support/mongo/build_set.sh` script from the Cluster Manager.

Script Usage

Script Usage: `/var/qps/bin/support/mongo/build_set.sh --help`

Create Specific Replica-set

Session Cache Replica-set

The following convention must be used while creating cross site replica-set for the session database:

You must create the session database replica-set members on same VM and same port on both sites. For example, 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]
```

Create replica-sets for session:



Note Sharding for the Session Cache is done through a separate process ([Create Session Shards, on page 12](#)) and must not be done using the `build_set.sh` script.

```
/var/qps/bin/support/mongo/build_set.sh --session --create
Starting Replica-Set Creation
Please select your choice: replica sets sharded (1) or non-sharded (2):
```

2

SPR Replica-set

Create replica-sets for SPR:



Note SPR (USum) supports mongo hashed sharding.

```
/var/qps/bin/support/mongo/build_set.sh --spr --create
Starting Replica-Set Creation
Please select your choice: replica sets sharded (1) or non-sharded (2):
```

2



Note The installation log should be generated in the appropriate directory (`/var/log/broadhop/scripts/`) for debugging or troubleshooting purpose.

Balance Replica-set

Create replica-sets for Balance:

```
/var/qps/bin/support/mongo/build_set.sh --balance --create

Starting Replica-Set Creation
Please select your choice replica sets sharded (1) or non-sharded (2)
```


2



Note The installation log should be generated in the appropriate directory (`/var/log/broadhop/scripts/`) for debugging or troubleshooting purpose.

Report Replica-set

Create replica-sets for Reporting:

```
/var/qps/bin/support/mongo/build_set.sh --report --create
```

```
Starting Replica-Set Creation
```

```
Please select your choice: replica sets sharded (1) or non-sharded (2):
```

2



Note The installation log should be generated in the appropriate directory (`/var/log/broadhop/scripts/`) for debugging or troubleshooting purpose.

Audit Replica-set

Create replica-sets for Audit

```
/var/qps/bin/support/mongo/build_set.sh --audit --create
```

```
Starting Replica-Set Creation
```

```
Please select your choice replica sets sharded (1) or non-sharded (2)
```

2



Note The installation log should be generated in the appropriate directory (`/var/log/broadhop/scripts/`) for debugging or troubleshooting purpose.

Create ADMIN Database

The ADMIN database holds information related to licensing, diameter end-points and sharding for runtime use.

To create the ADMIN database:

Step 1

Create replica-set for admin db by running the following command and follow the prompts. Refer to the sample output below.

```
/var/qps/bin/support/mongo/build_set.sh --admin --create
```

```
Please select your choice: replica sets sharded (1) or non-sharded (2):
```

2

```
Valid Option Selected: 2
```

```
You have selected replica sets to be non-sharded
```

```
Starting Replica-Set Creation
```

```
To know the process of this script please tailf
/var/log/broadhop/scripts/build_set_27102014_013907.log on another terminal
WARNING: Continuing will drop mongo database and delete everything under /data/sessions on all
Hosts
CAUTION: This result into loss of data
Are you sure you want to continue (y/yes or n/no)?
y
Please wait replica-set creation is in progress...
Replica-set Creation completed successfully.
```

Step 2

In case the ADMIN sets are not created and you get the ADMIN Replica-set are not available in /etc/broadhop/mongoConfig.cfg file error, perform the following steps:

- a) Define the ADMIN DB details with appropriate syntax in /etc/broadhop/mongoConfig.cfg file on Cluster Manager VM.

Example:

```
[ADMIN-SET1]
SETNAME=set05
ARBITER1=pcrfclient01:27721
ARBITER_DATA_PATH=/var/data/sessions.5
MEMBER1=sessionmgr01:27721
MEMBER2=sessionmgr02:27721
DATA_PATH=/var/data/sessions.5
[ADMIN-SET1-END]
```

- b) After defining the admin database details, rebuild etc.tar.gz.

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

- c) Update the Policy Server (QNS) VMs with the new software using reinit.sh script.

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

- d) Create replica-set for admin db by running the following command and follow the prompts. Refer to the sample output below.

```
/var/qps/bin/support/mongo/build_set.sh --admin --create

Please select your choice: replica sets sharded (1) or non-sharded (2):
2
Valid Option Selected: 2
You have selected replica sets to be non-sharded
Starting Replica-Set Creation
To know the process of this script please tailf
/var/log/broadhop/scripts/build_set_27102014_013907.log on another terminal
WARNING: Continuing will drop mongo database and delete everything under /data/sessions on all
Hosts
CAUTION: This result into loss of data
Are you sure you want to continue (y/yes or n/no)?
y
Please wait replica-set creation is in progress...
Replica-set Creation completed successfully.
```

- e) Update **Primary Ip address** and **Secondary Ip address** in Policy Builder under Cluster configuration.

After updating the Policy Builder, the admin db is automatically created on port 27721.

Replica-set Example

Here are some examples for replica-sets:

Step 1 Login to Cluster Manager.

Step 2 Edit the `mongoConfig.cfg` file:

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

[SESSION-SET1]
SETNAME=set01
ARBITER1=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
ARBITER1=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
ARBITER1=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
ARBITER1=pcrfclient01:27720
ARBITER_DATA_PATH=/var/data/sessions.4
MEMBER1=sessionmgr01:27720
MEMBER2=sessionmgr02:27720
DATA_PATH=/var/data/sessions.4
[SPR-SET1-END]
```

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*.

Add Member to a Replica-Set

Step 1 If there is a requirement to add additional member in replica-set, then it should be defined in `/etc/broadhop/mongoConfig.cfg` (on Cluster Manager) file.

Step 2 The `build_set.sh` script identifies the new member and adds them into the replica set.

```
/var/qps/bin/support/mongo/build_set.sh --session --add-members
```

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 12](#)
- [Create Session Shards, on page 12](#)

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.

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.

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 AIO: 80, 11211, 27017, 27749, 7070, 8080, 8090, 8182, 9091, 9092
    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
--hostnames : Check hostnames are valid (no underscores, resolvable, in
/etc/broadhop/servers) (AIO only)
--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
--swapsspace : 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 patch and feature version information and URLs to the various interfaces and 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

