



CPS Commands

- [about.sh](#), page 2
- [adduser.sh](#), page 2
- [auditrpms.sh](#), page 3
- [build_all.sh](#), page 3
- [build_etc.sh](#), page 5
- [build_set.sh](#), page 5
- [capture_env.sh](#), page 6
- [change_passwd.sh](#), page 6
- [cleanup_license.sh](#), page 7
- [component_alarm_reports.py](#), page 7
- [copytoall.sh](#), page 8
- [diagnostics.sh](#), page 9
- [dump_utility.py](#), page 11
- [list_installed_features.sh](#), page 14
- [reinit.sh](#), page 16
- [restartall.sh](#), page 17
- [restartqns.sh](#), page 17
- [runonall.sh](#), page 18
- [service](#), page 18
- [session_cache_ops.sh](#), page 18
- [set_priority.sh](#), page 22
- [startall.sh](#), page 23
- [startqns.sh](#), page 24
- [statusall.sh](#), page 24

- [stopall.sh](#), page 26
- [stopqns.sh](#), page 26
- [summaryall.sh](#), page 27
- [sync_times.sh](#), page 30
- [syncconfig.sh](#), page 30
- [terminatesessions](#), page 31
- [top_qps.sh](#), page 33
- [vm-init.sh](#), page 35

about.sh

This command displays core, patch, and feature software version information and URLs to the various interfaces and APIs for the deployment.

Syntax

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

Executable on VMs

- Cluster Manager
- pcrfclient01/02

adduser.sh

This utility adds a new user to the specified nodes that are part of the CPS deployment. These accounts will be provisioned without shell access and, as such, they're only useful for authenticating against the various web-based GUIs used to administrate CPS.

The hosts that get provisioned with these new accounts can be selected using the 'node-regex' option. The default regular expression used by the script is:

```
node-regex ::= ^(pcrfclient|qns|lb[0-9]+|sessionmgr)
```

Syntax

```
/var/qps/bin/support/adduser.sh [-h] [node-regex]
```

When prompted for the user's group, set 'qns-svn' for read-write permissions or 'qns-ro' for read-only permissions.

To add a user with 'read/write' access to Control Center, their group should be 'qns'.

- To check if a user already exists, login in as root and enter 'su <username>'.
- To check a user's 'groups', enter 'groups <username>'.

Executable on VMs

All

Example

```
[root@host /]# /var/qps/bin/support/adduser.sh
Enter username: username
Enter group for the user: groupname
Enter password: password
Re-enter password: password
The above example adds username to all the VMs in the cluster.
```

auditrpms.sh

This script runs in background on all VMs except Cluster Manager. This script/daemon should be always running and is monitored via monit. No intervention from end user is required. Corresponding logs are generated at individual nodes in `/var/log/broadhop/audit/audit_rpms.log`.

**Note**

All successful attempts i.e. installation or removal are tracked in this file. In case package is upgraded there would be two entries seen in log file, one for removal of old package and one for installation of new package.

Executable on VMs

On all VMs except Cluster Manager

Example

```
[root@lb01 ~]# monsum | grep auditrpms
Process 'auditrpms.sh' Running
```

build_all.sh

This command is executed from Cluster Manager to rebuild CPS package.

Syntax

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

Executable on VMs

Cluster Manager

Example

```
[root@host /]# /var/qps/install/current/scripts/build_all.sh
Building /etc/broadhop...
Copying to /var/qps/images/etc.tar.gz...
Creating MD5 Checksum...
```

```

Copying /etc/puppet to /var/qps/images/puppet.tar.gz...
Creating MD5 Checksum...
Copying Policy Builder configuration (/var/qps/current_config/pb_config) to
/var/qps/images/svn.tar.gz...
Creating MD5 Checksum...
Updating tar from: /var/qps/env_config/ to /var/www/html/images/
Creating MD5 Checksum...
Building /var/qps/bin...
Copying /var/qps/bin to /var/qps/images/scripts_bin.tar.gz...
Creating MD5 Checksum...
Building images...
Building image: /var/qps/images/controlcenter.tar.gz
Installing from:
file:///var/qps/.tmp/release
Installing features:
com.broadhop.controlcenter.feature.feature.group
com.broadhop.faultmanagement.service.feature.feature.group
com.broadhop.infrastructure.feature.feature.group
com.broadhop.server.runtime.product
com.broadhop.snmp.feature.feature.group
Creating MD5 Checksum... /var/qps/images/controlcenter.tar.gz.md5chksum
Building image: /var/qps/images/diameter_endpoint.tar.gz
Installing from:
file:///var/qps/.tmp/release
Installing features:
com.broadhop.diameter2.service.feature.feature.group
com.broadhop.server.runtime.product
com.broadhop.snmp.feature.feature.group
Creating MD5 Checksum... /var/qps/images/diameter_endpoint.tar.gz.md5chksum
Building image: /var/qps/images/iomanager01.tar.gz
Installing from:
file:///var/qps/.tmp/release
Installing features:
com.broadhop.iomanager.feature.feature.group
com.broadhop.notifications.service.feature.feature.group
com.broadhop.server.runtime.product
com.broadhop.snmp.feature.feature.group
Creating MD5 Checksum... /var/qps/images/iomanager01.tar.gz.md5chksum
Building image: /var/qps/images/iomanager02.tar.gz
Installing from:
file:///var/qps/.tmp/release
Installing features:
com.broadhop.iomanager.feature.feature.group
com.broadhop.notifications.service.feature.feature.group
com.broadhop.server.runtime.product
com.broadhop.snmp.feature.feature.group
Creating MD5 Checksum... /var/qps/images/iomanager02.tar.gz.md5chksum
Building image: /var/qps/images/pb.tar.gz
Installing from:
file:///var/qps/.tmp/release
Installing features:
com.broadhop.client.feature.audit.feature.group
com.broadhop.client.feature.balance.feature.group
com.broadhop.client.feature.custrefdata.feature.group
com.broadhop.client.feature.diameter2.feature.group
com.broadhop.client.feature.notifications.feature.group
com.broadhop.client.feature.spr.feature.group
com.broadhop.client.feature.unifiedapi.feature.group
com.broadhop.client.feature.vouchers.feature.group
com.broadhop.client.feature.ws.feature.group
com.broadhop.client.product
Creating MD5 Checksum... /var/qps/images/pb.tar.gz.md5chksum
Building image: /var/qps/images/pcrf.tar.gz
Installing from:
file:///var/qps/.tmp/release
Installing features:
com.broadhop.audit.service.feature.feature.group
com.broadhop.balance.service.feature.feature.group
com.broadhop.balance.spr.feature.feature.group
com.broadhop.custrefdata.service.feature.feature.group
com.broadhop.diameter2.local.feature.feature.group
com.broadhop.externaldatacache.memcache.feature.feature.group
com.broadhop.notifications.local.feature.feature.group

```

```

com.broadhop.policy.feature.feature.group
com.broadhop.server.runtime.product
com.broadhop.snmp.feature.feature.group
com.broadhop.spr.dao.mongo.feature.feature.group
com.broadhop.spr.feature.feature.group
com.broadhop.ui.controlcenter.feature.feature.group
com.broadhop.unifiedapi.interface.feature.feature.group
com.broadhop.unifiedapi.ws.service.feature.feature.group
com.broadhop.vouchers.service.feature.feature.group
com.broadhop.ws.service.feature.feature.group
Creating MD5 Checksum... /var/qps/images/pcrf.tar.gz.md5chksum
Copying portal default database to /var/qps/images/portal_dump.tar.gz
Creating MD5 Checksum for portal dump...
Copying portal to /var/qps/images/portal.tar.gz
Creating MD5 Checksum for portal.tar.gz...
Copying wispr.war to /var/qps/images/wispr.war
Output images to /var/qps/images/

```

build_etc.sh

This command is executed from Cluster Manager to rebuild etc.tar.gz in /etc/broadhop/ directory.

Syntax

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

Executable on VMs

Cluster Manager

Example

```

[root@host /]# /var/qps/install/current/scripts/build/build_etc.sh
Building /etc/broadhop...
Copying to /var/qps/images/etc.tar.gz...
Creating MD5 Checksum...

```

build_set.sh

This command is used to rebuild replica sets. This command is normally only run the first time the environment starts, but can be used if CPS databases must be rebuilt.

Syntax

```
/var/qps/bin/support/mongo/build_set.sh [--help]
```

Executable on VMs

All

Example

To create replica-sets for SPR:

```

[root@host /]# /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

```

capture_env.sh

This command collects most of the debug logs to debug an issue.

Syntax

```
/var/qps/bin/support/env/capture_env.sh
```

Executable on VMs

pcrfclient01/02

Output

This command provides the following information to collect logs:

- -h|--help: Show usage
- -q|--qns: For capturing qns logs (default is to skip qns logs)
- -t|--trap: For capturing trap logs (default is to skip trap logs)
- -m|--mongo: For capturing mongo logs (default is to skip mongo logs)
- -v|--var-log: For capturing /var/log/messages (default is to skip the log)
- -a|--age: Should be followed by maximum age of log based on last modification time (defaults to 1 day)
- -n|--host: Should be followed by common separated list of hostnames for capturing logs (defaults to all hosts)

Example

```
[root@host /]# /var/qps/bin/support/env/capture_env.sh
Creating archive of QPS environment information...
-----
Capturing /etc/broadhop...
Capturing logs...
Capturing Policy Builder data...
Capturing installed software versions...
```

change_passwd.sh

Change the Control Center user's (Linux user) password on Cluster Manager VM or OAM (pcrfclient) VM.

Syntax

```
/var/qps/bin/support/change_passwd.sh [-h]
```

Executable on VMs

All

Example

```

Enter username whose password needs to be changed:
Enter new password:
Re-enter new password:

Done.
Disconnecting from pcrfclient01... done.

```

cleanup_license.sh

Cleans up the records related to license in the licensedfeats collection in the sharding database. This command must be run as root user when license file is updated on the OAM (pcrfclient) machine.

Syntax

```
/var/qps/bin/support/mongo/cleanup_license.sh [-h]
```

Executable on VMs

- Cluster Manager
- pcrfclient01/02

component_alarm_reports.py

This command is used to store or retrieve the open/active component alarms in CPS.

- For clear alarms, it removes the alarms matching the clear alarm.
- For active alarms, it clears old alarms if any and adds the latest alarm.

Syntax

```

component_alarm_reports.py -h
usage: component_alarm_reports.py [-h] --action {update,report}
                                   [--eventhost EVENTHOST] [--date DATE]
                                   [--name NAME] [--facility FACILITY]
                                   [--severity SEVERITY] [--info INFO]

CPS Update/Report Component Alarm(s) to/from Mongo DB
optional arguments:
  -h, --help                show this help message and exit
  --action {update,report}, -a {update,report}
                             Action value update : Update an alarm. report : Report
                             active alarms
  --eventhost EVENTHOST, -e EVENTHOST
                             Event Host Name
  --date DATE, -d DATE      Date of event
  --name NAME, -n NAME      Name of alarm
  --facility FACILITY, -f FACILITY
                             Facility of alarm
  --severity SEVERITY, -s SEVERITY
                             Severity of alarm
  --info INFO, -i INFO      Info of alarm

```



Attention The `--action update` parameter is for Cisco Internal Use Only.

Path:

On Cluster Manager: `/var/qps/install/current/scripts/modules/component_alarm_reports.py`

On perfclicent and policy director VMs:

`/var/qps/bin/install/current/scripts/modules/component_alarm_reports.py`

Executable on VMs

Cluster Manager, Policy Director and OAM (perfclicent) nodes

Examples

To retrieve the active alarms:

```
component_alarm_reports.py -a report
event_host=1b02 name=ProcessDown severity=critical facility=operatingsystem
date=2017-22-11,10:13:49,310329511,+00:00 info=corosync process is down
```

copytoall.sh

Prior to 7.0.5 release, in order to propagate the changes done in Cluster Manager, user used to execute `reinit.sh` which in turn triggers each CPS VM to download and install the updated VM images from the Cluster Manager and it time consuming process.

In CPS 7.0.5 and higher releases, if minor changes are made to any file in Cluster Manager, instead of executing `reinit.sh` script, use this command to synchronize the modified files from Cluster Manager to all other VMs.

Syntax

```
copytoall.sh
```

Executable on VMs

Cluster Manager



Note In case executing `copytoall.sh` command from `qns-admin`, prefix `sudo` before the command.

Example

- 1 If the user updated `/etc/broadhop/logback.xml` file in Cluster Manager.
- 2 Build etc directory on each cluster by executing `build_all.sh` from Cluster Manager to rebuild CPS package script.


```
/var/qps/install/current/scripts/build_all.sh
```
- 3 Execute the following command to copy the file:


```
SSHUSER_PREFERROOT=true copytoall.sh /etc/broadhop/logback.xml /etc/broadhop/logback.xml
```


diagnostics.sh

Runs a set of diagnostics and displays the current state of the system. If any components are not running, red failure messages are displayed.



Note

RADIUS-based policy control is no longer supported in CPS 14.0.0 and later releases as 3GPP Gx Diameter interface has become the industry-standard policy control interface.

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_active_alarms : Get the active alarms in the CPS
--get_replica_status : Get the status of the replica-sets present in environment.
(Multi-Node Environment only)
--get_sharding_status : Get the status of the sharding information 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_peer_status: Get the diameter peers present in the environment.
--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
--radius : Run radius specific checks
--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 ~]#
```

List of Active Alarms

To get the list of active alarms, execute the `diagnostics.sh --get_active_alarms` command. Here is a sample output:

```
#diagnostics.sh --get_active_alarms

CPS Diagnostics HA Multi-Node Environment
-----
Active Application Alarm Status
-----
id=1000 sub_id=3001 event_host=lb02 status=down date=2017-11-22,
10:47:34,051+0000 msg="3001:Host: site-host-gx Realm: site-gx-client.com is down"
id=1000 sub_id=3001 event_host=lb02 status=down date=2017-11-22,
10:47:34,048+0000 msg="3001:Host: site-host-sd Realm: site-sd-client.com is down"
id=1000 sub_id=3001 event_host=lb01 status=down date=2017-11-22,
10:45:17,927+0000 msg="3001:Host: site-server Realm: site-server.com is down"
id=1000 sub_id=3001 event_host=lb02 status=down date=2017-11-22,
10:47:34,091+0000 msg="3001:Host: site-host-rx Realm: site-rx-client.com is down"
id=1000 sub_id=3002 event_host=lb02 status=down date=2017-11-22,
10:47:34,111+0000 msg="3002:Realm: site-server.com:applicationId: 7:all peers are down"
Active Component Alarm Status
-----
event_host=lb02 name=ProcessDown severity=critical facility=operatingsystem
date=2017-22-11,10:13:49,310329511,+00:00 info=corosync process is down
```



Attention

- Due to the limitation of architecture of the CPS SNMP implementation, if the SNMP daemon or policy server (QNS) process on pcrfclient VM restarts, there can be gap between active alarms displayed by the `diagnostics.sh` and active alarms in NMS.
- The date printed for application alarm status is when the alarm was seen at pcrfclient VM. The time for the alarm at NMS is the time before the alarm is received from Policy Director (LB) VM. So there can be a difference in the dates for the same alarm reported in `diagnostics.sh` and in NMS.

Sample Output of --get_sharding_status

```
-
```

```

|-----|
|                                     MONGODB SHARDING STATUS INFORMATION
|      Date : 2017-12-20 19:02:38 |
|-----|

Shard Id   Mongo DB                               State   Backup DB   Removed   Session
Count

1          sessionmgr01:27717/session_cache         online  false       false     0
2          sessionmgr01:27717/session_cache_2     online  false       false     0
4          sessionmgr01:27717/session_cache_4     online  false       false     0

Rebalance Status: Rebalanced

|-----|

Shard Id   Mongo DB                               State   Backup DB   Removed   Session
Count

1          sessionmgr01:37717/session_cache         online  false       false     0

Rebalance Status: Rebalanced

```

dump_utility.py

This collection utility is used to collect standard information from the CPS system in case of issues (system, application, database). This utility collects such information from VM, depending on type of information and VMs selected in the input.

This utility can be executed from anywhere from the terminal. Logs are printed on terminal and written to a log file: `/var/tmp/dumputility-<date_time_when_executed>.log`.



Important

Warning messages related to the files that does not exist in the system will not be displayed on the terminal but will be logged only to the log file (`/var/tmp/dumputility-<date_time_when_executed>.log`).



Caution

Running the dump utility can be CPU intensive.



Important

The dump utility should be run from the Cluster Manager wherever possible.

The following types of information can be collected:

- **Common Information:** This information is common for all type of issues. Information is collected from `perfclient01` VM. If `perfclient01` is down, information is collected from `perfclient02` VM. If both VMs are down, information is collected from Cluster Manager VM. The following information can be fetched:
 - `about.sh` output
 - `diagnostics.sh` output
 - `list_installed_features.sh` output

- Facter output
 - Consolidated logs
 - Bulkstats files
 - SVN dump from PB config
- **System Information:** This information is useful in troubleshooting system related issues. The following information can be fetched:
- `sysctl -a` output
 - Information about processes running
 - Firewall configuration
 - Netstat statistics
 - Complete lsof output
 - Total number of open files
 - `ifconfig` output
 - Routing table information
 - Disk usage
 - Monit status
 - Monit summary
 - System logs
 - Sar logs
 - Dmesg logs
 - Secure logs
 - Yum logs
 - Whisper logs
 - Puppet logs
- **Application Information:** This information is useful in troubleshooting application-related issues. The following information can be fetched:
- Contents of `/etc/broadhop` directory
 - `/var/log/broadhop` logs
 - monit status
 - monit summary
- **Database Information:** This information is useful in troubleshooting database related issues. The following information can be fetched:
- MongoDB logs

- Mongostat output
 - `rs.status()` output
 - `rs.conf()` output
 - `/var/qps/bin/support/mongo/session_cache_ops.sh -count` output
 - `top_qps.sh` output for 10 seconds
 - `mongotop` output for 3 seconds
- **OAM (PCRFLIENT) Specific Data:** The following information can be fetched from OAM (pcrflient) VMs:
 - carbon logs
 - httpd logs
 - `pcs resource show` output
- **Policy Director (Ib) Specific Data:** The following information can be fetched from policy director (load balancer) VMs:
 - SNMP trap logs
 - HAproxy logs
 - `pcs resource show` output
- **Policy Server (QNS) Specific Data:** The following information can be fetched from policy server (QNS) VMs:
 - Thread level CPU/memory usage of java process
 - `jstack` output of java process
 - Policy Server (QNS) logs
 - Policy Server service logs

Syntax

`dump_utility.py`

The following options are supported:

- `-v, --vm-type`: Specifies type of VM or single VM name from which information has to be fetched. Multiple VMs are separated by colon. For example, `--vm-type qns:sessionmgr01`.
- `-i, --info-type`: Specifies type of information to be collected. Possible values are `application`, `db`, `system`, `vm_specific`. Multiple values are separated by colon. For example, `--info-type application:system`.
- `-o, --output-file-name`: Name of the tar file to store fetched information.
- `-h, --help`: Displays help.

Executable on VMs

- Cluster Manager
- pcrfclient01/02

Example

- To fetch system information from Policy Director (lb) VMs:

```
dump_utility.py --info-type system --vm-type lb
```

- To fetch application and VM specific information from qns01:

```
dump_utility.py --info-type application:vm_specific --vm-type qns01
```

OR

```
dump_utility.py --info-type application:vm_specific --vm-type sav-qns01
```

where, *sav-qns01* is hostname of qns01 VM.

- To fetch database specific information from all replica sets:

```
dump_utility.py --info-type db --vm-type pcrfclient:sessionmgr
```

Sample output:

```
dump_utility.py --info-type application --vm-type sav-qns01
Logs are also getting stored in /var/tmp/dumputility-07-06-2016-04-07-47.log
*****
Collecting information, please wait...
*****
Fetching common information like about.sh/list_installed_features/diagnostics etc from
pcrfclient01
This step takes time, please wait...
Fetching command outputs from pcrfclient01
Fetching files hosts file from pcrfclient01
Fetching files consolidated logs from pcrfclient01
Fetching files Bulkstats file from pcrfclient01
Fetching command outputs from qns01
Fetching files Broadhop dir from qns01
Fetching files Broadhop logs from qns01
*****
Information is collected at : /var/tmp/07-06-2016-04-07-47.tar.gz
*****
Disconnecting from pcrfclient01... done.
```

**Important**

For non-root users, certain CPS scripts (`about.sh`, `diagnostics.sh` and so on) expects `sudo` password. For such scripts, output is displayed on terminal and is saved in the file. Also for some data which can only be accessed by root user, permission denied related warning is displayed.

list_installed_features.sh

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

Example

```
[root@host /]# /var/qps/bin/diag/list_installed_features.sh
Features installed on lb01:9045
com.broadhop.infrastructure.feature=7.0.2.r072627
com.broadhop.iomanager.feature=7.0.2.r072627
com.broadhop.server.runtime.product=7.0.2.r072627
com.broadhop.snmp.feature=7.0.2.r072627
Features installed on lb02:9045
com.broadhop.infrastructure.feature=7.0.2.r072627
com.broadhop.iomanager.feature=7.0.2.r072627
com.broadhop.server.runtime.product=7.0.2.r072627
com.broadhop.snmp.feature=7.0.2.r072627
Features installed on qns01:9045
com.broadhop.balance.service.feature=3.4.2.r071203
com.broadhop.balance.spr.feature=3.4.2.r071203
com.broadhop.custrefdata.service.feature=2.4.2.r072158
com.broadhop.diameter2.local.feature=3.4.2.r072694
com.broadhop.externaldatacache.memcache.feature=7.0.2.r072627
com.broadhop.infrastructure.feature=7.0.2.r072627
com.broadhop.policy.feature=7.0.2.r072627
com.broadhop.server.runtime.product=7.0.2.r072627
com.broadhop.snmp.feature=7.0.2.r072627
com.broadhop.spr.dao.mongo.feature=2.3.2.r071887
com.broadhop.spr.feature=2.3.2.r071887
com.broadhop.ui.controlcenter.feature=3.4.2.r070445
com.broadhop.unifiedapi.interface.feature=2.3.2.r072695
com.broadhop.unifiedapi.ws.service.feature=2.3.2.r072695
com.broadhop.vouchers.service.feature=3.4.2.r071203
com.broadhop.ws.service.feature=1.5.2.r071537
Features installed on qns02:9045
com.broadhop.balance.service.feature=3.4.2.r071203
com.broadhop.balance.spr.feature=3.4.2.r071203
com.broadhop.custrefdata.service.feature=2.4.2.r072158
com.broadhop.diameter2.local.feature=3.4.2.r072694
com.broadhop.externaldatacache.memcache.feature=7.0.2.r072627
com.broadhop.infrastructure.feature=7.0.2.r072627
com.broadhop.policy.feature=7.0.2.r072627
com.broadhop.server.runtime.product=7.0.2.r072627
com.broadhop.snmp.feature=7.0.2.r072627
com.broadhop.spr.dao.mongo.feature=2.3.2.r071887
com.broadhop.spr.feature=2.3.2.r071887
com.broadhop.ui.controlcenter.feature=3.4.2.r070445
com.broadhop.unifiedapi.interface.feature=2.3.2.r072695
com.broadhop.unifiedapi.ws.service.feature=2.3.2.r072695
com.broadhop.vouchers.service.feature=3.4.2.r071203
com.broadhop.ws.service.feature=1.5.2.r071537
Features installed on qns03:9045
com.broadhop.balance.service.feature=3.4.2.r071203
com.broadhop.balance.spr.feature=3.4.2.r071203
com.broadhop.custrefdata.service.feature=2.4.2.r072158
com.broadhop.diameter2.local.feature=3.4.2.r072694
com.broadhop.externaldatacache.memcache.feature=7.0.2.r072627
com.broadhop.infrastructure.feature=7.0.2.r072627
com.broadhop.policy.feature=7.0.2.r072627
com.broadhop.server.runtime.product=7.0.2.r072627
com.broadhop.snmp.feature=7.0.2.r072627
com.broadhop.spr.dao.mongo.feature=2.3.2.r071887
com.broadhop.spr.feature=2.3.2.r071887
com.broadhop.ui.controlcenter.feature=3.4.2.r070445
com.broadhop.unifiedapi.interface.feature=2.3.2.r072695
com.broadhop.unifiedapi.ws.service.feature=2.3.2.r072695
com.broadhop.vouchers.service.feature=3.4.2.r071203
com.broadhop.ws.service.feature=1.5.2.r071537
Features installed on qns04:9045
com.broadhop.balance.service.feature=3.4.2.r071203
com.broadhop.balance.spr.feature=3.4.2.r071203
com.broadhop.custrefdata.service.feature=2.4.2.r072158
```

```

com.broadhop.diameter2.local.feature=3.4.2.r072694
com.broadhop.externaldatacache.memcache.feature=7.0.2.r072627
com.broadhop.infrastructure.feature=7.0.2.r072627
com.broadhop.policy.feature=7.0.2.r072627
com.broadhop.server.runtime.product=7.0.2.r072627
com.broadhop.snmp.feature=7.0.2.r072627
com.broadhop.spr.dao.mongo.feature=2.3.2.r071887
com.broadhop.spr.feature=2.3.2.r071887
com.broadhop.ui.controlcenter.feature=3.4.2.r070445
com.broadhop.unifiedapi.interface.feature=2.3.2.r072695
com.broadhop.unifiedapi.ws.service.feature=2.3.2.r072695
com.broadhop.vouchers.service.feature=3.4.2.r071203
com.broadhop.ws.service.feature=1.5.2.r071537
Features installed on pcrfclient01:9045
com.broadhop.controlcenter.feature=7.0.2.r072627
com.broadhop.faultmanagement.service.feature=1.0.2.r071534
com.broadhop.infrastructure.feature=7.0.2.r072627
com.broadhop.server.runtime.product=7.0.2.r072627
com.broadhop.snmp.feature=7.0.2.r072627
Features installed on pcrfclient02:9045
com.broadhop.controlcenter.feature=7.0.2.r072627
com.broadhop.faultmanagement.service.feature=1.0.2.r071534
com.broadhop.infrastructure.feature=7.0.2.r072627
com.broadhop.server.runtime.product=7.0.2.r072627
com.broadhop.snmp.feature=7.0.2.r072627
Features installed on all (combined)
com.broadhop.balance.service.feature=3.4.2.r071203
com.broadhop.balance.spr.feature=3.4.2.r071203
com.broadhop.controlcenter.feature=7.0.2.r072627
com.broadhop.custrefdata.service.feature=2.4.2.r072158
com.broadhop.diameter2.local.feature=3.4.2.r072694
com.broadhop.externaldatacache.memcache.feature=7.0.2.r072627
com.broadhop.faultmanagement.service.feature=1.0.2.r071534
com.broadhop.infrastructure.feature=7.0.2.r072627
com.broadhop.iomanager.feature=7.0.2.r072627
com.broadhop.policy.feature=7.0.2.r072627
com.broadhop.server.runtime.product=7.0.2.r072627
com.broadhop.snmp.feature=7.0.2.r072627
com.broadhop.spr.dao.mongo.feature=2.3.2.r071887
com.broadhop.spr.feature=2.3.2.r071887
com.broadhop.ui.controlcenter.feature=3.4.2.r070445
com.broadhop.unifiedapi.interface.feature=2.3.2.r072695
com.broadhop.unifiedapi.ws.service.feature=2.3.2.r072695
com.broadhop.vouchers.service.feature=3.4.2.r071203
com.broadhop.ws.service.feature=1.5.2.r071537

```

reinit.sh

This command is executed from Cluster Manager. It SSHs to all the CPS VMs and triggers the `/etc/init.d/vm-init.sh` script on each VM to download all the Puppet scripts, CPS softwares, `/etc/hosts` files and updates the VM with the new software from Cluster Manager to the VM.

Refer to [vm-init.sh](#), on page 35, to trigger this process for a single VM as opposed to all VMs.

Syntax

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

Executable on VMs

Cluster Manager

Example

```
[root@host ~]# /var/qps/install/current/scripts/upgrade/reinit.sh
Running pupdate on lab
Updating /etc/hosts file from installer VM...
Updating /etc/facter/facts.d/bxbl-lb01...
Updating /etc/puppet from installer VM...
```

restartall.sh

This command is executed from Cluster Manager. It stops and restarts all of the Policy Server (QNS) services on all VMs in the CPS cluster. This command is also executed when new software is installed on VMs.

Refer to [restartqns.sh](#), on page 17 to restart Policy Server (QNS) services on a specific VM as opposed to all VMs.

Syntax

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

Executable on VMs

Cluster Manager

**Note**

When executing restartall.sh command from qns-admin, prefix sudo before the command.

Example

```
/var/qps/bin/control/restartall.sh
Currently active LB: lb01
```

This process will restart all QPS software on the nodes in this order:

```
lb02 pcrfclient02 qns01 qns02 pcrfclient01 lb01
```

restartqns.sh

This command stops and restarts all Policy Server (QNS) services on the target VM.

Syntax

```
/var/qps/bin/control/restartqns.sh hostname
```

Executable on VMs

Cluster Manager

**Note**

When executing restartqns.sh command from qns-admin, prefix sudo before the command.

Example

```
/var/qps/bin/control/restartqns.sh qns01
/var/qps/bin/control/restartqns.sh pcrfclient01
```

runonall.sh

Executes a command, as provided as an argument, on all of the VMs listed in the servers file. These commands must be run as the CPS user on the remote VMs, or they will fail to execute properly.

Syntax

```
/var/qps/bin/control/runonall.sh <executable command>
```

Executable on VMs

All

**Note**

In case executing runonall.sh command from qns-admin, prefix sudo before the command.

Example

```
/var/qps/bin/control/runonall.sh ntpdate -u
```

service

This command is used to control individual services on each VM.

Syntax

```
service < option > | --status-all | [ service_name [ command | --full-restart ] ]
```

**Caution**

Do not use this command for any services managed by the monit service. Use the monit summary command to view the list of services managed by monit. The list of services managed by monit is different on each CPS VM.

session_cache_ops.sh

This command provides information about, and performs operations on the session database.

Syntax

```
/var/qps/bin/support/mongo/session_cache_ops.sh <Argument1> <Argument2>
<Argument1>: --count or --remove
--count
--remove
```

```

--statistics-count
--add-shard
--add-ringset
--db-shrink
<Argument2>: site1 or site2 or site3 ... siten
This argument for GR only, in GR setup user need to pass the site number (site1 or site2
...) as second argument

```

Options

--count

This option prints the count of sessions present in all available session_cache* databases.

The session count is the number of allocated entries in the database for unique subscriber sessions on the network. Each allocated entry may have related nested sub-sessions with other session types such as Sy/Rx.

- A session count is incremented when a Gx CCR-I arrives and an entry (Mongo data structure called a document) is allocated.
- If there are other types of sessions related to that unique subscriber during the life of the Gx session (or Sy/Rx) these are nested within the "document".
- If these "other" types of sessions are terminated, they are removed from the document and those counter types are decremented immediately.
- When the Gx CCR-T arrives it is decremented immediately from the Gx Type count.
- Up to 30 seconds later the unique session entry/document is removed, and the Session Total count is decremented.

The other session types should not be used in validating the number of total sessions as this varies greatly between call models and time. These are simply specific totals drawn from each entry.

It is typical for the session_total_count will be slightly more than the Gx_TYPE count due to the 30 second delay. The reason for this delay is that the entry (document) needs to wait for any other related (nested) sessions to close.

On occasion there may be small variance of data between what the pcrfclient01 and pcrfclient02 report, although they are querying the same database. These are, however, comparable.

The counters processing order is that the total session count is performed first and the detailed session (types) are done second. Slight discrepancies/variance in the numbers may occur.

Example

```

# session_cache_ops.sh --count
Session cache operation script
Tue Dec 22 02:26:49 MST 2015
-----
Session Replica-set SESSION-SET1
-----
Session Database          : Session Count
-----
session_cache             : 14
session_cache_2           : 15
session_cache_3           : 12
session_cache_4           : 10
-----
No of Sessions in SET1   : 51
-----
Total Number of Sessions : 51

```

--remove

This option removes sessions from all available session_cache* databases.

**Warning**

You will be prompted to confirm this action after running this command. If you proceed, this will remove existing sessions in the replica-set.

Example

```
# session_cache_ops.sh --remove
Session cache operation script
Tue Dec 22 02:29:42 MST 2015
-----
Session Replica-set SESSION-SET1
-----
WARNING: Continuing will remove existing sessions in
        replica-set : SESSION-SET1
CAUTION: This result into loss of session data
Are you sure you want to continue (y/yes or n/no)? : y
Removing sessions from session_cache db
connecting to: sessionmgr04:27717/session_cache
WriteResult({ "nRemoved" : 1 })
Remove sessions operation completed on session_cache db.
Removing sessions from session_cache_2 db
connecting to: sessionmgr04:27717/session_cache_2
WriteResult({ "nRemoved" : 0 })
Remove sessions operation completed on session_cache_2 db.
Removing sessions from session_cache_3 db
connecting to: sessionmgr04:27717/session_cache_3
WriteResult({ "nRemoved" : 0 })
Remove sessions operation completed on session_cache_3 db.
Removing sessions from session_cache_4 db
connecting to: sessionmgr04:27717/session_cache_4
WriteResult({ "nRemoved" : 0 })
Remove sessions operation completed on session_cache_4 db.
-----
```

--statistics-count

This option prints statistics count of the sessions (types if the session Gx, Rx, and so on) in all available session_cache* databases.

Example

```
# session_cache_ops.sh --statistics-count
Session cache operation script
Tue Dec 22 02:28:38 MST 2015
-----
Sessions statistic counter on General
-----
  Session Type          : Session Count
-----
ADMIN-SET1
  EDR                   : 5
  GX_SCE                 : 10
-----
```

--add-shard

Adds session shards to the session database, either normal shards or hot standby shards.

Example

```
# session_cache_ops.sh --add-shard
Session cache operation script
Tue Dec 22 02:22:24 MST 2015
      Session Sharding
-----
```

```
Select type of session shard Default [*]
                               Hot Standby [ ]

Sessionmgr pairs : sessionmgr01:sessionmgr02:27717

Session shards per pair : 4

Creating Session sharding [ Done ]

-----
Note :
- Press 'y' to select the shard type
- If sharding needed for multiple sessionmgr vms with port
  please provide sessionmgr vm with port separated by ':',
  and pair separated by ',',
(Ex: sessionmgr01:sessionmgr02:27717,sessionmgr03:sessionmgr04:27717)
```

--add-ringset

This option adds a new set to the ring.

Example

```
# session_cache_ops.sh --add-ringset
Session cache operation script
Wed Jun  8 18:23:15 EDT 2016
Session cache operation script: addRingSet
The progress of this script can be monitored in the following log:
/var/log/broadhop/scripts/session_cache_ops_08062016_182315.log
Note :
Please provide sessionmgr vm separated by ':' and pair separated by ','

(Ex HA: sessionmgr01-lab:sessionmgr02-lab)
(Ex GR: sessionmgr01-sitel:sessionmgr02-sitel,sessionmgr01-site2:sessionmgr02-site2)
Enter cache servers: sessionmgr01,sessionmgr02
Verifying Qnses processes is running
Adding set sessionmgr01,sessionmgr02 to ring
Executing OSGI Command> setSkRingSet 1 4 sessionmgr01:11211,
Executing OSGI Command> setSkRingSet 1 4 sessionmgr02:11211,
Executing OSGI Command> rebuildSkRing 1
Ringset added successfully
```

--db-shrink

This option is used after clean of all sessions from CPS mongo database. It performs a synchronization operation by removing session cache database files and copying data files from primary member. This reduces the database size and compact database files and/or reclaim disk space. Currently, this operation does not support specific to replica-set.

**Note**

This option must be performed in maintenance window (if required in production) and when there is no session data.

Example

```
# session_cache_ops.sh --db-shrink
Session cache operation script
Fri May 13 06:17:42 EDT 2016
-----
Session DB Shrink Replica-set
-----
CAUTION: This option must performed in maintenance window and no session data
Are you sure you want to continue (y/yes or n/no)? : yes
Verify log /var/log/broadhop/scripts/session_cache_ops_13052016_061742.log
```

```

DB Shrink operation completed successfully for set - SESSION-SET1
  DB File count before Shrink: 36
  DB File count after Shrink: 16
  DB Size before Shrink: 4.2G
  DB Size after Shrink: 256M

DB Shrink operation completed successfully for set - SESSION-SET2
  DB File count before Shrink: 28
  DB File count after Shrink: 8
  DB Size before Shrink: 4.0G
  DB Size after Shrink: 128M

```

Executable on VMs

perfclient01/02

set_priority.sh

This command sets the priorities of replica-sets, and replica-set members for High Availability (HA) or Geo-Redundant (GR) CPS deployments.

By default, priority of mongo databases, replica-sets, and members are set in order (with higher priority) as defined in the Mongo Config (mongoConfig.cfg).

Use the `diagnostics.sh --get_replica_status` command to view the status and current priorities of all databases replica-sets.

Syntax

`/var/qps/bin/support/mongo/set_priority.sh`

The following options are supported:

- **Mandatory Options:**

```
--db <db_name>
      [all|session|spr|admin|balance|report|portal|audit|bindings]
```

The `set_priority --db all` command would set the priority of all replica-sets listed in `mongoConfig.cfg` in descending order. The member that is listed first in the configuration would be assigned the highest priority.

The `set_priority --db session` command would set the priority of all replica-sets of db type SESSION. By default, priorities are set in descending order.

- **General Options:**

```
--h [ --help ]           show syntax and usage information for this script
--version                show version information of this script
--asc                    Set priority in ascending order (default is descending)
--dsc                    Set priority in descending order
--priority <0|1000>     Set specific priority
--force [false|true]    forces the new priority to be applied (default is false).
```



Note The `--priority <0|1000>` option is not currently supported. Do not use.

**Caution**

Do not use the `--force` option unless instructed by a Cisco representative. By default, the `set_priority.sh` script will only attempt to set the priorities when all members of a replica set are in a healthy state. The `--force` option can be used when the members are NOT in a healthy state.

• Specific Replica-set Options:

`--replSet <setname>` specifies the replica-set name

This option enables you to specify priority for a particular replica-set. You must provide the `<setname>`.

• Geo-Redundancy Options:

`--sitename [site1|site2]` specifies the GR site to which the operation applies

This option enables you to specify a GR site. The `mongoConfig.cfg` must have relevant start and end tags (like `#SITE1_START` and `#SITE1_END`).

Executable on VMs

Cluster Manager

Examples

High Availability Options:

```
set_priority.sh --db all
set_priority.sh --db session
set_priority.sh --db session --asc
set_priority.sh --db session --replSet set01
```

Geo-Redundancy Options:

```
set_priority.sh --db session --replSet set01 --sitename <site1|site2>
set_priority.sh --db session --replSet set01 --sitename <site1|site2>
set_priority.sh --db session --replSet set01 --sitename <site1|site2> --force true
```

startall.sh

This command is executed from Cluster Manager. It starts all Policy Server (QNS) services on all VMs in the CPS cluster. This command is also executed when a new software is installed on VMs.

Refer to [startqns.sh](#), on page 24 to start services on a specific VM as opposed to all VMs.

Syntax

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

**Note**

When executing `startall.sh` command from `qns-admin`, prefix `sudo` before the command.

Executable on VMs

Cluster Manager

Example

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

startqns.sh

This command is executed from Cluster Manager. It starts all Policy Server (QNS) services on the specified VM.

Syntax

```
/var/qps/bin/control/startqns.sh hostname
```

**Note**

When executing startqns.sh command from qns-admin, prefix sudo before the command.

Executable on VMs

Cluster Manager

Example

```
/var/qps/bin/control/startqns.sh qns01
/var/qps/bin/control/startqns.sh pcrfclient01
```

statusall.sh

This command displays whether the services managed by monit 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
```

**Note**

When executing statusall.sh command from qns-admin, prefix sudo before the command.

Executable on VMs

- Cluster Manager
- pcrfclient01/02

Output

For each process or program, the command displays:

- **Status**
 - Running – the process/Program is healthy and running

- Does not exist – the process id specified in the /var/run/processname-pid does not exist. This is a cause for concern if recurring.
- Waiting – This is normal for a program /process monitored by monit
- Status ok – This is normal for a program monitored by monit

• Monitoring Status

- Monitored – The process/program is being monitored
- Not Monitored – The process/program is not under the control of monit
- Waiting – A transient state which reports as waiting depending upon when the statusall.sh command is run which internally uses monit status command.



Note For more details, see: <https://bitbucket.org/tildeslash/monit/issue/114/>.

• Uptime

The number of days, hours, and minutes the process or program has been running.

Example

```
[root@host ~]# /var/qps/bin/control/statusall.sh
Executing 'sudo /usr/bin/monit status' on all QNS Servers
The Monit daemon 5.5 uptime: 2h 12m
Process 'snmptrapd'
  status          Running
  monitoring status Monitored
  uptime          15h 33m
Process 'snmpd'
  status          Running
  monitoring status Monitored
  uptime          2h 12m
Process 'sessionmgr-27017'
  status          Running
  monitoring status Monitored
  uptime          15h 33m
Process 'qns-2'
  status          Running
  monitoring status Monitored
  uptime          15h 33m
Process 'qns-1'
  status          Running
  monitoring status Monitored
  uptime          15h 33m
Process 'memcached'
  status          Running
  monitoring status Monitored
  uptime          15h 33m
Process 'logstash'
  status          Running
  monitoring status Monitored
  uptime          15h 33m
Process 'elasticsearch'
  status          Running
  monitoring status Monitored
  uptime          15h 33m
Process 'collectd'
  status          Running
  monitoring status Monitored
  uptime          15h 33m
```

```

Process 'carbon-cache'
  status          Running
  monitoring status Monitored
  uptime          15h 33m
Process 'carbon-aggregator'
  status          Running
  monitoring status Monitored
  uptime          15h 33m
System 'lab'
  status          Running
  monitoring status Monitored
Connection to 127.0.0.1 closed.

```

stopall.sh

This command is executed from Cluster Manager. It stops the Policy Server (QNS) services on each VMs in the CPS cluster.

Refer to [stopqns.sh](#), on page 26 to stop Policy Server (QNS) services on a specific VM as opposed to all VMs.

Syntax

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



Note

When executing stopall.sh command from qns-admin, prefix sudo before the command.

Executable on VMs

Cluster Manager

Example

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

stopqns.sh

This command is executed from Cluster Manager. It stops all Policy Server (QNS) services on the specified VM.

Syntax

```
/var/qps/bin/control/stopqns.sh hostname
```



Note

When executing stopqns.sh command from qns-admin, prefix sudo before the command.

Executable on VMs

Cluster Manager

Example

```
/var/qps/bin/control/stopqns.sh qns01
```

summaryall.sh

This command provides a brief status of the services managed by monit on all VMs in the CPS cluster.

Syntax

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

**Note**

When executing summaryall.sh command from qns-admin, prefix sudo before the command.

Executable on VMs

Cluster Manager

Example

```
[root@host /]# /var/qps/bin/control/summaryall.sh
The Monit daemon 5.17.1 uptime: 3d 19h 21m
```

```
Process 'whisper'           Running
Process 'snmptrapd'        Running
Process 'snmpd'            Running
Program 'vip_trap'         Status ok
Process 'redis'            Running
Process 'qns-4'            Running
Process 'qns-3'            Running
Process 'qns-2'            Running
Process 'qns-1'            Running
File 'monitor-qns-4'       Accessible
File 'monitor-qns-3'       Accessible
File 'monitor-qns-2'       Accessible
File 'monitor-qns-1'       Accessible
Process 'memcached'        Running
Process 'haproxy-diameter' Running
Process 'haproxy'          Running
Process 'cutter'           Running
Process 'corosync'         Running
Program 'cpu_load_monitor' Status ok
Program 'cpu_load_trap'    Status ok
Program 'gen_low_mem_trap' Status ok
Process 'collectd'         Running
Process 'auditrpms.sh'     Running
System 'C-pd01'            Running
The Monit daemon 5.17.1 uptime: 13h 37m
```

```
Process 'whisper'           Running
Process 'snmptrapd'        Running
Process 'snmpd'            Running
Program 'vip_trap'         Status ok
Process 'redis'            Running
Process 'qns-4'            Running
Process 'qns-3'            Running
Process 'qns-2'            Running
Process 'qns-1'            Running
File 'monitor-qns-4'       Accessible
File 'monitor-qns-3'       Accessible
File 'monitor-qns-2'       Accessible
File 'monitor-qns-1'       Accessible
```

```

Process 'memcached'           Running
Process 'haproxy-diameter'    Running
Process 'haproxy'            Running
Process 'cutter'              Running
Process 'corosync'           Running
Program 'cpu_load_monitor'    Status ok
Program 'cpu_load_trap'       Status ok
Program 'gen_low_mem_trap'    Status ok
Process 'collectd'           Running
Process 'auditrpm.sh'        Running
System 'C-pd02'               Running
The Monit daemon 5.17.1 uptime: 13h 37m

```

```

Process 'whisper'            Running
Process 'snmpd'              Running
Process 'qns-1'              Running
File 'monitor-qns-1'        Accessible
Program 'cpu_load_monitor'    Status ok
Program 'cpu_load_trap'       Status ok
Program 'gen_low_mem_trap'    Status ok
Process 'collectd'           Running
Process 'auditrpm.sh'        Running
System 'C-qns01'             Running
The Monit daemon 5.17.1 uptime: 13h 37m

```

```

Process 'whisper'            Running
Process 'snmpd'              Running
Process 'qns-1'              Running
File 'monitor-qns-1'        Accessible
Program 'cpu_load_monitor'    Status ok
Program 'cpu_load_trap'       Status ok
Program 'gen_low_mem_trap'    Status ok
Process 'collectd'           Running
Process 'auditrpm.sh'        Running
System 'C-qns02'             Running
The Monit daemon 5.17.1 uptime: 13h 37m

```

```

Process 'whisper'            Running
Process 'snmpd'              Running
Process 'qns-1'              Running
File 'monitor-qns-1'        Accessible
Program 'cpu_load_monitor'    Status ok
Program 'cpu_load_trap'       Status ok
Program 'gen_low_mem_trap'    Status ok
Process 'collectd'           Running
Process 'auditrpm.sh'        Running
System 'C-qns03'             Running
The Monit daemon 5.17.1 uptime: 13h 36m

```

```

Process 'whisper'            Running
Process 'snmpd'              Running
Process 'qns-1'              Running
File 'monitor-qns-1'        Accessible
Program 'cpu_load_monitor'    Status ok
Program 'cpu_load_trap'       Status ok
Program 'gen_low_mem_trap'    Status ok
Process 'collectd'           Running
Process 'auditrpm.sh'        Running
System 'C-qns04'             Running
The Monit daemon 5.17.1 uptime: 13h 36m

```

```

Process 'whisper'            Running
Process 'snmpd'              Running
Process 'memcached'          Running
Program 'cpu_load_monitor'    Status ok
Program 'cpu_load_trap'       Status ok
Program 'gen_low_mem_trap'    Status ok
Process 'collectd'           Running
Process 'auditrpm.sh'        Running
System 'C-sm01'               Running
The Monit daemon 5.17.1 uptime: 13h 36m

```

```

Process 'whisper'            Running

```

```

Process 'snmpd' Running
Process 'memcached' Running
Program 'cpu_load_monitor' Status ok
Program 'cpu_load_trap' Status ok
Program 'gen_low_mem_trap' Status ok
Process 'collectd' Running
Process 'auditrpms.sh' Running
System 'C-sm02' Running
The Monit daemon 5.17.1 uptime: 13h 35m

Process 'whisper' Running
Process 'snmpd' Running
Program 'kpi_trap' Status failed
Program 'db_trap' Status ok
Program 'failover_trap' Status ok
Program 'qps_process_trap' Status ok
Program 'admin_login_trap' Status ok
Program 'vm_trap' Status ok
Program 'gr_site_status_trap' Status ok
Program 'qps_message_trap' Status ok
Program 'ldap_message_trap' Status ok
Process 'qns-2' Running
Process 'qns-1' Running
Program 'monitor_replica' Status ok
File 'monitor-qns-2' Accessible
File 'monitor-qns-1' Accessible
Process 'logstash' Running
Process 'grafana-server' Running
Program 'mon_db_for_lb_failover' Status ok
Process 'elasticsearch' Running
Process 'corosync' Running
Program 'cpu_load_monitor' Status ok
Program 'cpu_load_trap' Status ok
Program 'gen_low_mem_trap' Status ok
Process 'collectd' Running
Process 'carbon-cache' Running
Process 'carbon-aggregator' Running
Process 'auditrpms.sh' Running
System 'C-cc01' Running
The Monit daemon 5.17.1 uptime: 13h 35m

Process 'whisper' Running
Process 'snmpd' Running
Program 'kpi_trap' Status failed
Program 'db_trap' Status ok
Program 'failover_trap' Status ok
Program 'qps_process_trap' Status ok
Program 'admin_login_trap' Status ok
Program 'vm_trap' Status ok
Program 'gr_site_status_trap' Status ok
Program 'qps_message_trap' Status ok
Program 'ldap_message_trap' Status ok
Process 'qns-2' Running
Process 'qns-1' Running
Program 'monitor_replica' Status ok
File 'monitor-qns-2' Accessible
File 'monitor-qns-1' Accessible
Process 'logstash' Running
Process 'grafana-server' Running
Program 'mon_db_for_lb_failover' Status ok
Process 'elasticsearch' Running
Process 'corosync' Running
Program 'cpu_load_monitor' Status ok
Program 'cpu_load_trap' Status ok
Program 'gen_low_mem_trap' Status ok
Process 'collectd' Running
Process 'carbon-cache' Running
Process 'carbon-aggregator' Running
Process 'auditrpms.sh' Running
System 'C-cc02' Running

qns-1 (pid 23717) is running...
qns-2 (pid 27878) is running...

```

```

qns-3 (pid 30976) is running...
qns-4 (pid 3502) is running...
qns-1 (pid 23787) is running...
qns-2 (pid 24337) is running...
qns-3 (pid 24852) is running...
qns-4 (pid 25356) is running...
qns-1 (pid 29570) is running...
qns-1 (pid 6270) is running...
qns-1 (pid 2909) is running...
qns-1 (pid 3453) is running...
qns-1 (pid 30040) is running...
qns-2 (pid 32207) is running...
qns-1 (pid 9939) is running...
qns-2 (pid 8682) is running...

```

sync_times.sh

This command synchronizes the time between all CPS VMs.

Syntax

For High Availability deployments:

```
/var/qps/bin/support/sync_times.sh ha
```

For Geographic Redundancy deployments:

```
/var/qps/bin/support/sync_times.sh gr
```

Executable on VMs

Cluster Manager

To check the current clock skew of the system, execute the following command:

```
diagnostics.sh --clock_skew -v
```

The output numbers are in seconds. Refer to the following sample output:

```

CPS Diagnostics Multi-Node Environment
-----
Checking for clock skew...
Clock skew not detected between qns01 and lb01. Skew: 1...[PASS]
Clock skew not detected between qns02 and lb01. Skew: 0...[PASS]
Clock skew not detected between lb01 and lb01. Skew: 0...[PASS]
Clock skew not detected between lb02 and lb01. Skew: 0...[PASS]
Clock skew not detected between sessionmgr01 and lb01. Skew: 0...[PASS]
Clock skew not detected between sessionmgr02 and lb01. Skew: 0...[PASS]
Clock skew not detected between pcrfclient01 and lb01. Skew: 0...[PASS]
Clock skew not detected between pcrfclient02 and lb01. Skew: 0...[PASS]

```

syncconfig.sh

This command is executed to synchronize the changes to the VM nodes. The files in the

`/var/qps/current_config/etc/broadhop` are zipped to a file and stored in `/var/www/html`. The Puppet scripts in VM downloads the file to the VM and applies the changes to the VM.

Syntax

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

```
/var/qps/install/currentfolder/scripts/bin/update/syncconfig.sh
```

where, `currentfolder` is version of the current installation.

For example, for CPS 7.0.5, it is 7.0.5.

```
/var/qps/install/7.0.5/scripts/bin/update/synconfig.sh
```

Executable on VMs

All

Example

```
[root@host ~]# /var/qps/bin/update/synconfig.sh
Building /etc/broadhop...
Copying to /var/qps/images/etc.tar.gz...
Creating MD5 Checksum...
```

terminatesessions

This utility submits bulk session terminate requests.



Note

For fresh installations of CPS 10.1.0, this feature is enabled by default. However, for upgrades from systems prior to CPS 10.1.0, this feature needs to be enabled as follows:

In the `/etc/broadhop/pcrf/features` file, add `com.broadhop.policy.command.feature`. For more information, refer to "Customize Features" in the "Deployment" section in *CPS Installation Guide for VMware*



Important

To eliminate the impact of TPS and session count in the system, add the following entry in the `/etc/broadhop/qns.conf` file on the Cluster Manager VM:

```
-Ddistribution.blocked.duration=1800000
```

The entry value is in milliseconds, which converts to 30 minutes. The recommended value is multiples of 30 minutes.

After configuring the above values, run the following commands:

```
copytoall.sh /etc/broadhop/qns.conf
stopall.sh
startall.sh
```

Syntax

```
/var/qps/bin/support/command --username <USERNAME> --password <PASSWORD> terminatesessions
--criteria <criteria> [--disable_signaling <y/n - default n>] [--rate <throttling rate -
default 100>]
```

Where,

- `--username` and `--password` are the user's Control Center credentials.
- `--criteria`: Identifies the session. Following are some examples:
 - ALL

- APN eq SOS
- APN except SOS
- IMSIRANGE A-B



Remember

For the termination of sessions without any criteria (ALL) and termination of sessions with IMSI range as criteria (IM SIRANGE A-B), CPS must be configured to create sessions with **tags** field having **ImsiKey:imsi:<imsivalue>** as element. If this element is not configured, the command does not terminate sessions for ALL and IMSI range as criteria.

- `--disable_signaling`: Disables signaling on external interface.
- `--rate`: Defines the throttling rate.

`/var/qps/bin/support/command terminatesessions -h` shows help related to the command option.

Executable on VMs

pcrfclient01/02

Example

```
/var/qps/bin/support/command -u testuser -p cisco123 terminatesessions -c "ALL" -d y
Do you want to proceed with delete command? [y]|n: y
deleteBulkSession testuser "ALL" false 100
User is : testuser
Criterion is : ALL
Command Criteria type : ALL
Command Criteria value : null
Signalling is set to : false
Rate-Limiter value is set to : 100
CommandId submitted successfully : 1471941788159
```

show

This utility shows the status of the submitted command(s).

Syntax

```
/var/qps/bin/support/command --username <USERNAME> --password <PASSWORD> show [--all <All>]
[--id <ID>]
```

Where,

- `--username` and `--password` are the user's Control Center credentials.
- `--all`: Shows the status of all the command requests submitted.
- `--id`: Shows the status of the submitted command.

Executable on VMs

pcrfclient01/02

Example

```

/var/qps/bin/support/command -u testuser -p cisco123 show
getCommands
BulkTerminateCommand(1471492548449)- state: COMPLETED submitted: Thu Aug 18 09:25:48 IST
2016 status:
[Eligible for Deletion = 1, Submitted For Deletion = 1, Not Submitted Due To Later Creation
= 0]
BulkTerminateCommand(1471492739896)- state: COMPLETED submitted: Thu Aug 18 09:28:59 IST
2016 status:
[Eligible for Deletion = 1, Submitted For Deletion = 1, Not Submitted Due To Later Creation
= 0]
BulkTerminateCommand(1471493146320)- state: COMPLETED submitted: Thu Aug 18 09:35:46 IST
2016 status:
[Eligible for Deletion = 1, Submitted For Deletion = 1, Not Submitted Due To Later Creation
= 0]
BulkTerminateCommand(1471494348267)- state: COMPLETED submitted: Thu Aug 18 09:55:48 IST
2016 status:
[Eligible for Deletion = 1, Submitted For Deletion = 1, Not Submitted Due To Later Creation
= 0]
BulkTerminateCommand(1471494588431)- state: COMPLETED submitted: Thu Aug 18 09:59:48 IST
2016 status:
[Eligible for Deletion = 1, Submitted For Deletion = 1, Not Submitted Due To Later Creation
= 0]

/var/qps/bin/support/command -u testuser -p cisco123 show --id 1471494588431
getCommand 1471494588431
BulkTerminateCommand(1471494588431)- state: COMPLETED submitted: Thu Aug 18 09:59:48 IST
2016 status:
[Eligible for Deletion = 1, Submitted For Deletion = 1, Not Submitted Due To Later Creation
= 0]

```

cancel

This utility cancels the further execution of the submitted command.

Syntax

```
/var/qps/bin/support/command --username <USERNAME> --password <PASSWORD> cancel --id <ID>
```

Where,

- --username and --password are the user's Control Center credentials.
- --id: ID of the submitted command.

Executable on VMs

perfcient01/02

Example

```

/var/qps/bin/support/command -u testuser -p cisco123 cancel --id 1471941788159
Do you want to proceed with cancel command? [y]|n: y
cancelCommand 1471941788159
Command Already completed: 1471941788159

```

top_qps.sh

This command displays performance statistics of CPS VMs.

Syntax

```
/var/qps/bin/control/top_qps.sh <time>
```

where <time> is the number of seconds for which the statistics are to be captured.



Note

When executing top_qps.sh command from qns-admin, prefix sudo before the command.

Executable on VMs

perfclient01/02

Output

- Average time in ms.
- Number of message transactions processed during n seconds, where n is an integer value in seconds.
- Transactions per second (TPS) is messages/n.
- Error shows any error occurred during execution on the Policy Server (QNS) VM. It could be database error, authentication failure and so on. Details of the error can be seen in the consolidated engine or in the consolidated Policy Server (QNS) log.
- Times used is how much total time it took to process the message.

Example

Figure 1: Example for top_qps.sh

```
-----
Host Detail:
qns03,qns01,qns07,qns06,qns08,qns05,qns04
qns02
Measurement timer: 5   QNS Count: 8
-----
Average Success TPS Error Time Used Messages
31.3868 31706 6341.2000 0 995.1514 diameter_Gx_CCR-U
32.4724 4490 898.0000 0 145.8012 diameter_Rx_AAR
31.1475 4630 926.0000 0 144.2129 diameter_Gx_CCR-I
29.8733 4697 939.4000 0 140.3147 diameter_Syp_AAA
31.7974 4837 807.4000 36 128.3662 diameter_Rx_STR
30.5017 3722 744.4000 0 113.5272 diameter_Gx_CCR-T
30.1642 415 83.0000 0 12.5181 diameter_Syp_STA
33.1618 221 44.2000 0 7.3288 diameter_Gx_RAA
27.6945 120 24.0000 0 3.3233 class com.broadhop.cache.TimerExpired
6.0920 1 0.2000 0 0.0061 diameter_Rx_ASA
-----
Average Success TPS Error Time Used Actions
13.6113 82620 16524.0000 0 1124.5648 com.broadhop.locking.impl.LockSessionAction
4.4117 54757 10951.4000 0 241.5705 com.broadhop.cache.impl.actions.GetSessionAction
2.2590 50002 10000.4000 36 112.9528 com.broadhop.session.UpdateEntry
0.9880 54760 10952.0000 0 54.1017 com.broadhop.spr.impl.actions.GetSubscriberActionImpl
0.8557 36553 7310.6000 0 31.2784 com.broadhop.balance.service.actions.OCSLoadBalanceState
2.4404 4691 938.2000 0 11.4408 com.broadhop.session.CreateEntry
0.0674 15322 3064.4000 0 1.8321 com.broadhop.balance.service.actions.OCSGetReservationStatusRequest
0.0174 54734 10946.8000 0 0.9521 com.broadhop.policyintel.impl.actions.StartPolicyReporting
0.0138 54758 10951.6000 0 0.7581 com.broadhop.policy.impl.service.AddSubscriberService
0.0183 32091 6418.2000 0 0.5866 send.diameter_Gx_CCA-U
0.0284 13374 2674.8000 0 0.3795 diameter.create.remote.session.Syp
0.0283 8649 1729.8000 0 0.2447 send.diameter_Gx_RAR
0.0316 4691 938.2000 0 0.1484 send.diameter_Gx_CCA-I
0.0026 54747 10949.4000 0 0.1407 com.broadhop.policyintel.impl.actions.StopPolicyReporting
0.0279 4691 938.2000 0 0.1309 send.diameter_Syp_AAR
0.0007 163853 32770.6000 0 0.1194 com.broadhop.policy.impl.actions.LogMessage
0.0231 3998 799.6000 0 0.0922 send.diameter_Syp_STR
0.0197 4540 908.0000 0 0.0896 send.diameter_Rx_AAA
0.0183 4091 818.2000 0 0.0749 send.diameter_Rx_STA
0.0166 3776 755.2000 0 0.0626 send.diameter_Gx_CCA-T
1.0123 15 3.0000 0 0.0152 com.broadhop.session.DeleteEntry
0.0191 102 20.4000 0 0.0020 send.diameter_Rx_ASR
-----
Fri May 1 01:48:21 IST 2015
*** End-of-Collection ***
```

275526

Diameter Synchronization Message Behavior

Some Diameter messages (like UDR) are synchronous Diameter calls, which means that the Policy Server (QNS) will be waiting for a response after sending the Diameter request.

Response of these Diameter message is not captured in `top_qps` as those message are not processed in policy engine separately.

Average time 3.3676 shown below is round trip time (from UDR sent to UDA received)

Sample Top_Qns

```
-----
```

Average	Success	TPS	Error	Time Used	Messages
9.7211	2910	727.5000	0	28.2883	diameter_Gx_CCR-I

```
-----
```

Average	Success	TPS	Error	Time Used	Actions
3.6854	2924	731.0000	0	10.7761	
3.3676	2922	730.5000	0	9.8400	send.sync.diameter_Sh_UDR
0.7908	2919	729.7500	0	2.3083	com.broadhop.session.CreateEntry
0.1981	2924	731.0000	0	0.5793	
0.0480	2919	729.7500	0	0.1400	send.diameter_Gx_CCA-I
0.0126	2924	731.0000	0	0.0370	diameter.create.remote.session.Sh

Average time is not applicable for these response messages. However, number of response messages (UDA) received can be seen from Grafana.

vm-init.sh

This command is executed from the VM nodes from `/etc/init.d`, (starts up automatically if VM reboots too). It downloads all the Puppet script, CPS software, `/etc/hosts` files and updates the VM with the new software.

This command only updates the software and does not restart the CPS software. The new software will be run only after process restart (for example, by executing `/var/qps/bin/control/restartall.sh` script from Cluster Manager).

Syntax

```
/etc/init.d/vm-init.sh
```

Executable on VMs

Any CPS VM

Example

```
/etc/init.d/vm-init.sh
```

