



## **CPS UDC MoP for Migration, Release 13.0.0**

**First Published:** 2017-06-23

**Last Modified:** 2017-07-03

### **Americas Headquarters**

Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA 95134-1706  
USA  
<http://www.cisco.com>  
Tel: 408 526-4000  
800 553-NETS (6387)  
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

© 2017 Cisco Systems, Inc. All rights reserved.

# CONTENTS

---

<b>METHOD OF PROCEDURE FOR UDC MIGRATION .....</b>	<b>7</b>
<i>Pre-requisites</i> .....	7
<i>Configure Installer VM</i> .....	7
<i>Initiate UDC VM</i> .....	14
<i>Configure LB VMs</i> .....	15
<i>Configure Admin DB</i> .....	16
Configure Policy Builder .....	16
Configure Installer VM .....	17
<i>Verify UDC VM</i> .....	18
<i>Verify Installer VM</i> .....	19
<i>Verify PCRFCClient VMs</i> .....	19
<i>Verify LB VMs</i> .....	20
<b>MODIFICATIONS IN PLATFORM FILES .....</b>	<b>21</b>
<i>jconfig.py</i> .....	21
File Name .....	21
Diff Output .....	21
<i>reinit.sh</i> .....	22
File Name .....	22
Diff Output .....	22
<i>pp_status.sh</i> .....	23
File Name .....	23
Diff Output .....	23
<i>copytoall.sh</i> .....	23
File Name .....	23
Diff Output .....	23
<i>hosts.sh</i> .....	24
File Name .....	24
Diff Output .....	24
<i>diagnostics.sh</i> .....	24
File Name .....	24
Diff Output .....	24
<i>runonall.sh</i> .....	25
File Name .....	25
Diff Output .....	25
<i>udc.pp</i> .....	25
File Name .....	25
Diff Output .....	25
<i>udc.yaml</i> .....	26
File Name .....	26
File Content .....	26
<i>servers</i> .....	27
File Name .....	27
Diff Output .....	27
<i>servers.all</i> .....	27
File Name .....	27

Diff Output.....	27
<i>build_servers_all.sh</i> .....	28
File Name .....	28
Diff Output.....	28
<i>/images/image-map</i> .....	28
File Name .....	28
Diff Output.....	28
<i>/env_config/image-map</i> .....	29
File Name .....	29
Diff Output.....	29





# Method of Procedure for UDC Migration

This chapter includes the following topics:

- Pre-requisites
- Configure Installer VM
- Initiate UDC VM
- Configure LB VMs
- Configure Admin DB
- Verify UDC VM
- Verify Installer VM
- Verify PCRFCClient VMs
- Verify LB VMs

## Pre-requisites

The following tasks should be performed before migrating UDC VMs:

1. Stop the monit service on those QNS VMs identified to be replaced with UDC VMs by running the following command:

```
service monit stop
```

**NOTE:** Replace the last set of QNS VMs and leave qns01 while migrating to UDC VMs.

2. Stop the qns service on the identified QNS VMs mentioned above by running the following command:

```
service qns stop
```

3. Turn off the identified qns<yy> VMs in vCenter.

## Configure Installer VM

Perform the following steps to configure installer VM:

- Step 1** Edit `/var/qps/config/deploy/csv/Hosts.csv` and modify the QNS VM (to be converted to UDC VM) details as follows:

VM Details	Current Details of QNS VM	Modified UDC VM Details
Role	qps	udc
Alias	qns0x	udc0x
Guest Name	<prefix>-qps0x	<prefix>-udc0x

**NOTE:** It is recommended to convert only the last set of QNS VMs. For example, if there are 100 QNS VMs, convert qns100, qns99, and so on.

**Step 2** Edit `/var/qps/config/deploy/csv/Definitions.csv`. Modify the identified QNS VM definition row to be migrated, by changing `qns<yy>` row with `Diskmode` as `sparse` and `Alias` as `udc<yy>`, where the number `<yy>` remains the same as QNS VMs replaced. For the next rows to be migrated the `Diskmode` column can be left clear.

Diskmode	Datastores	Alias
thin	datastore3	lb01
monolithicSparse	datastore4	lb02
monolithicFlat	datastore5	pcrfclient01
twoGbMaxExtentSparse	datastore6	pcrfclient02
woGbMaxExtentFlat		portal01
seSparse		portal02
eagerZeroedThick		sm01
thick		sm02
sparse		qns01
		qns02
sparse		udc03
		udc04

**Step 3** Add a new row for UDC VM in `/var/qps/config/deploy/csv/VMSpecification.csv` with similar configuration as “qps”. The UDC VM configuration should have the role “udc” as shown below:

Role	Host Name Prefix	Memory	vCPU	Diskmode
lb01		16384	16	thin
lb02		16384	16	thin
sm		16384	8	thin
qps		16384	8	thin
udc		16384	8	thin
portal		2048	8	thin
pcrfclient01		16384	8	thin
pcrfclient02		16384	8	thin

**Step 4** Append `/var/qps/install/current/scripts/build/build_fact.sh` configuration as follows:

```
echo "udc_instances=1" >> $FACT_FILE
```

**Step 5** Update `/var/qps/install/current/scripts/deployer/support/jconfig.py` to include UDC cluster info similar to QNS.

**Step 6** Modify `/var/qps/install/current/scripts/upgrade/reinit.sh` to include UDC regexp and other loops.



- Step 7** Add UDC regexp pattern to `/var/qps/install/current/scripts/upgrade/pp_status.sh`
- Step 8** Add UDC regexp to `/var/qps/install/current/scripts/bin/control/copytoall.sh`
- Step 9** Add UDC regexp pattern to `/var/qps/install/current/scripts/bin/support/hosts.sh` and copy the file to all other VMs by running the following command:
- ```
SSHUSER_PREFERROOT=true copytoall.sh /var/qps/bin/support/hosts.sh
```
- Step 10** Add UDC regexp pattern to `/var/qps/install/current/scripts/bin/diag/diagnostics.sh` and copy the file to all other VMs by running the following command:
- ```
SSHUSER_PREFERROOT=true copytoall.sh /var/qps/bin/diag/diagnostics.sh
```
- Step 11** Add UDC regexp to `/var/qps/install/current/scripts/bin/control/runonall.sh` and copy the file to all other VMs by running the following command:
- ```
SSHUSER_PREFERROOT=true copytoall.sh /var/qps/bin/control/runonall.sh
```
- Step 12** Copy `/etc/puppet/modules/qps/manifests/roles/qns.pp` as follows:
- ```
/etc/puppet/modules/qps/manifests/roles/udc.pp
```
- Change all instances of “qns” to “udc”.
- Step 13** Copy `/etc/puppet/modules/qps/nodes/qns.yaml` as follows:
- ```
/etc/puppet/modules/qps/nodes/udc.yaml
```
- Update `/etc/puppet/modules/qps/nodes/udc.yaml` by modifying `qps::roles::qns:` as follows:
- ```
qps::roles::udc:
```
- Step 14** Copy `/var/qps/install/current/puppet/modules/qps/templates/etc/snmp/qns.snmpd.conf` to `/var/qps/install/current/puppet/modules/qps/templates/etc/snmp/udc.snmpd.conf` in the same directory
- Step 15** Copy `/var/qps/install/current/puppet/modules/qps/templates/etc/qns.sysctl.conf` to `/var/qps/install/current/puppet/modules/qps/templates/etc/udc.sysctl.conf` in the same directory.
- Step 16** Copy `/var/qps/current_config/etc/broadhop/pcrf` recursively (-r option) as `/var/qps/current_config/etc/broadhop/udc`
- Step 17** In the `/etc/broadhop/udc/qns.conf` file, remove the following entries:
- ```
--DjmsFlowControlHost=xxxx  
  
--DjmsFlowControlPort=xxxx
```
- Step 18** Copy `/var/qps/current_config/etc/broadhop/diameter_endpoint` recursively (-r option) as `/var/qps/current_config/etc/broadhop/udc_diameter_endpoint`

**Step 19** Modify `/var/qps/current_config/etc/broadhop/servers` as described below:

- a. Replace the lines corresponding to QNS VMs that are migrated to UDC VMs with type as `udc`. For example, `udc03=udc`.
- b. Modify the end point information to include the `udc_diameter_endpoint` for the load balancer VMs.

**Step 20** Modify `/var/qps/current_config/etc/broadhop/servers.all` as described below:

- a. Replace the lines corresponding to QNS VMs that are migrated to UDC VMs with type as `udc`. For example, `<cluster>-udc03=udc` and so on.
- b. Modify the end point information to include the `udc_diameter_endpoint` for the load balancer VMs. For example, `<cluster>-lb<xx>=udc_diameter_endpoint` and so on.

**Step 21** Modify `/var/qps/install/current/scripts/build/build_servers_all.sh` to add UDC regexp.

**Step 22** Update feature files as follows:

`/etc/broadhop/pb/features` will have 3 new features

```
com.broadhop.client.feature.udcclient
com.broadhop.client.feature.udsninterface
com.broadhop.client.feature.udcfe
```

`/etc/broadhop/pcrf/features` will have 1 new feature

```
com.broadhop.udcclient.service.feature
```

`/etc/broadhop/udc/features` will have 2 new features

```
com.broadhop.udsninterface.service.feature
com.broadhop.udcfe.service.feature
```

and 1 existing feature

```
com.cisco.api.service.feature
```

Copy the feature files to all VMs using the following commands:

```
SSHUSER_PREFERROOT=true /var/qps/bin/control/copytoall.sh /etc/broadhop/pb/features
```

```
SSHUSER_PREFERROOT=true /var/qps/bin/control/copytoall.sh /etc/broadhop/pcrf/features
```

**Step 23** Modify `/var/qps/images/image-map` and `/var/qps/env_config/image-map` to have an entry `udc=udc` and change one of the `lb=diameter_endpoint` as follows:

```
lb=udc_diameter_endpoint
```

**Step 24** In Installer VM, modify `/etc/broadhop/mongoConfig.cfg` to add a new replica set for a new Admin DB:

```
[ADMIN-SET<X>]
SETNAME=set<xx>
ARBITER=pcrfclient01:<port number>
ARBITER_DATA_PATH=/var/data/sessions.<x>
MEMBER1=sessionmgr01:<port number>
MEMBER2=sessionmgr02:<port number>
DATA_PATH=/var/data/sessions.<x>
[ADMIN-SET<X>-END]
```

**NOTE:**

1. Ensure that the port number for this Admin replica set is not currently used.
2. The “<x>” in [ADMIN-SET<X>] should be a number.

Copy the configuration to all VMs using the following command:

```
SSHUSER_PREFERROOT=true /var/qps/bin/control/copytoall.sh /etc/broadhop/mongoConfig.cfg
```

**Step 25** Run `/var/qps/bin/support/mongo/build_set.sh --admin --create --setname set<xx>` where `set<xx>` is the Set name configured in `mongoConfig.cfg` to create the Admin DB.

For the prompts, enter `non-sharded(2)` as the option and then `yes` to continue as described below:

```
Starting Replica-Set Creation
```

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

Type 2 and Enter

```
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)?
```

Type `y` or `yes` and Enter

Verify that the port is up and in listening mode in both session managers (`netstat -tupln | grep LISTEN | grep xxxxx` where `xxxxx` is the port number of the new admin DB configured in `mongoConfig.cfg`).

**Step 26** If needed, create another session DB for UDC. As an example, a new session DB can be added by adding the following in `mongoConfig.cfg`:

```
[SESSION-SET<X>]
SETNAME=set<0X>
```

```
OPLOG_SIZE=<YYYY>
ARBITER=pcrfclient01:<zzzz>
ARBITER_DATA_PATH=/var/data/sessions.v
MEMBER1=sessionmgr01:<zzzz>
MEMBER2=sessionmgr02:<zzzz>
DATA_PATH=/var/data/sessions.v
[SESSION-SET<X>-END]
```

The session DB can be built using the following command:

```
/var/qps/bin/support/mongo/build_set.sh --session --create --setname set0x
```

For more information about creating new replica sets, see *CPS Installation Guide - VMware*.

For two clusters DB setup, ensure that hosts collection from existing admin DB are copied to the new admin DB. For more information about two clusters DB deployment, see *CPS Installation Guide - VMware*.

**Step 27** Run the following command to configure priorities for the replica-sets:

```
set_priority.sh
```

**Step 28** Verify and add appropriate license to the new Admin DB created:

```
Run mongodump --host sessionmgr01 --port yyyy --db sharding --collection licensedfeats
```

Where yyyy is the port number of the existing Admin DB.

```
Run mongorestore --host sessionmgr01 --port xxxxx --db sharding --collection
licensedfeats /root/dump/sharding/licensedfeats.bson
```

Where xxxxx is the port number of the new Admin DB configured in mongoConfig.cfg.

**Step 29** Update

/var/qps/install/current/puppet/modules/qps/templates/collectd\_worker/collectd.d/jmxplugin.conf for UDC stats as described below:

a. Append the following configuration under <Plugin "GenericJMX">:

```
<MBean "qns-udc-events">
  ObjectName
  "com.broadhop.udcclient.policy.event.UdcClientStatisticsManager:name=*,type=service"
  InstancePrefix "udc."
  InstanceFrom "name"
  <Value>
```

```
    Type "qns_stat"

    Attribute "Success"

    Attribute "Error"

    Attribute "TotalTimeMs"

    Attribute "AverageTime"

</Value>

</MBean>
```

- b. Modify 'node1' connection information to include "qns-udc-events" as follows:

```
<Connection>

    InstancePrefix "node1."

    ServiceURL "service:jmx:rmi:///jndi/rmi://localhost:9045/jmxrmi"

    Collect "garbage_collector"

    Collect "java-memory"

    Collect "thread"

    Collect "classes"

    Collect "qns-counters"

    Collect "qns-spr"

    Collect "qns-actions"

    Collect "qns-messages"

    Collect "qns-andsf-events"

    Collect "qns-entitlement-events"

    Collect "qns-ldap-bind"

    Collect "qns-ldap-search"

    Collect "qns-ldap-connection"

    Collect "qns-ldap-pool"

    Collect "mog-api-stats"

    Collect "scef-api-stats"
```

```
Collect "dra-db-counters"

Collect "qns-udc-events"

</Connection>
```

## Initiate UDC VM

Perform the following steps on the Installer VM to initiate UDC VM:

**Step 1** Run `/var/qps/install/current/scripts/import/import_deploy.sh`

**Step 2** Run `/var/qps/install/current/scripts/deployer/support/jvalidate.py`

This script validates the parameters against the ESX servers to ensure ESX server can support the configuration and deploy the VMs.

**Step 3** Run `/var/qps/install/current/scripts/build_all.sh`

**Step 4** Modify `/var/qps/images/image-map` to have an entry `udc=udc` and change one of the `lb=diameter_endpoint` to be `lb=udc_diameter_endpoint` or append as needed.

**Step 5** Modify `/var/qps/current_config/etc/broadhop/servers.all` as described below:

- a. Replace the lines corresponding to QNS VMs that are migrated to UDC VMs with type as `udc`. For example, `<cluster>-udc03=udc` and so on.
- b. Modify the end point information to include the `udc_diameter_endpoint` for the load balancer VMs. For example, `<cluster>-lb<xx>=udc_diameter_endpoint` and so on.

**Step 6** Run `/var/qps/install/current/scripts/deployer/deploy.sh udc0x`

**NOTE:** Other VMs will not be reachable for few minutes during this procedure.

**Step 7** After UDC VM is initiated, copy `/etc/snmp/scripts` recursively (-r) from any of the QNS VMs to UDC VM.

**Step 8** Log in to Admin DB from the installer VM using `mongo sessionmgr:xxxxx`, where `xxxxx` is the port number of the new (UDC) Admin DB.

By default, UDC Admin DB takes `sessionmgr01` and `sessionmgr02` with port 27717 as UDC session DB. To create session manager DBs, refer to *CPS Installation Guide - VMware*.

Update the fields in bold with UDC session manager information as described below:

```
set06:PRIMARY> db.shards.findOne()

{

    "_id" : 1,

    "seed_1" : "sessionmgr01",
```

```

    "seed_2" : "sessionmgr02",
    "port" : 27727,
    "db" : "session_cache",
    "online" : true,
    "count" : NumberLong(5),
    "lockTime" : ISODate("2017-06-29T20:47:43.690Z"),
    "isLocked" : false,
    "lockedBy" : null
  }

```

In the following example, the command below sessionmgr03 and sessionmgr04 becomes the UDC session DBs.

```

db.shards.update(
  { _id: 1 },
  { $set:
    {
      "seed_1" : "sessionmgr03",
      "seed_2" : "sessionmgr04",
      port : NumberInt(27727)
    }
  }
)

```

## Configure LB VMs

**NOTE:** This task needs to be carried out after LB reinit/relaunch at any point of time.

Perform the following steps to configure LB VMs:

**Step 1** In `/etc/haproxy/haproxy.cfg`, remove any reference to older QNS VMs which has been replaced with UDC VMs.

**NOTE:** Do not modify `/etc/haproxy/haproxy.cfg` if there are no references to older QNS VMs.

## ■ Configure Admin DB

- Step 2** Copy `/etc/broadhop/diameter_endpoint` recursively (`-r` option) as `/etc/broadhop/udc_diameter_endpoint`.
- Step 3** For each additional UDC diameter endpoint that needs to be created, copy pre-existing `qns` folder under `/opt/broadhop/` recursively (`-r` option) as new folder called `qns-<xx>`.
- Step 4** For each additional UDC diameter endpoints created, register the service with `monit` by modifying and reloading the `monit` configuration as follows:
- Copy `/etc/monit.d/monitor-qns-<xx>` as `/etc/monit.d/monitor-qns-<yy>`, where `yy` is the new QNS service number. Update the file content to modify all `<xx>` occurrences inside the file as `<yy>`.
  - Copy `/etc/monit.d/qns-<xx>` as `/etc/monit.d/qns-<yy>`, where `yy` is the new QNS service number. Update the file content to modify all `<xx>` occurrences inside the file as `<yy>`.
  - Reload the `monit` config using the following command:

```
monit reload
```

- Step 5** Append `/etc/broadhop/servers` with `udc_diameter_endpoint` entries as follows:

```
<prefix>-lb<yy>=udc_diameter_endpoint
```

- Step 6** Verify the services are up and running using the following command:

```
service qns status
```

If the new service status has stopped, start the service using the following command:

```
service qns start
```

## Configure Admin DB

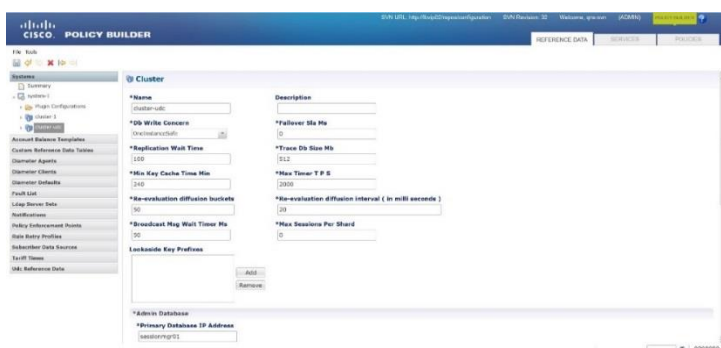
Configuring Admin DB includes the following tasks:

- Configure Policy Builder
- Configure Installer VM

### Configure Policy Builder

- Step 1** In Policy Builder, navigate to **System** under **Reference Data**.

- Step 2** Click on **Create Child: Cluster**.





A new cluster screen is loaded.

**Step 3** Add the name as follows:

```
cluster-udc
```

**Step 4** Scroll down and add the database IP and port address that are configured in `mongoConfig.cfg` for the Admin Database.

**Step 5** Publish the configuration.

## Configure Installer VM

**Step 1** Modify `/etc/broadhop/udc/qns.conf` to add the following two lines at the top inside the QNS\_OPTS string:

```
-Dcom.broadhop.run.systemId=system-1
-Dcom.broadhop.run.clusterId=cluster-udc
```

**Step 2** Run `SSHUSER_PREFERROOT=true /var/qps/bin/control/copytoall.sh /etc/broadhop/udc/qns.conf`

**Step 3** Modify `/etc/broadhop/udc_diameter_endpoint/qns.conf` to add the following two lines at the top inside the QNS\_OPTS string:

```
-Dcom.broadhop.run.systemId=system-1
-Dcom.broadhop.run.clusterId=cluster-udc
```

**Step 4** Run `SSHUSER_PREFERROOT=true /var/qps/bin/control/copytoall.sh /etc/broadhop/udc_diameter_endpoint/qns.conf`

**Step 5** Modify `/etc/broadhop/pcrf/qns.conf` to add the following two lines at the top inside the QNS\_OPTS string:

```
-Dcom.broadhop.run.systemId=system-1
-Dcom.broadhop.run.clusterId=cluster-1
```

**Step 6** Run `SSHUSER_PREFERROOT=true /var/qps/bin/control/copytoall.sh /etc/broadhop/pcrf/qns.conf`

**Step 7** Modify `/etc/broadhop/diameter_endpoint/qns.conf` to add the following two lines at the top inside the QNS\_OPTS string:

```
-Dcom.broadhop.run.systemId=system-1
-Dcom.broadhop.run.clusterId=cluster-1
```

**Step 8** Run

```
SSHUSER_PREFERROOT=true /var/qps/bin/control/copytoall.sh /etc/broadhop/diameter_endpoint/qns.conf
```

**Step 9** Modify `/etc/broadhop/controlcenter/qns.conf` to add the following two lines at the top inside the QNS\_OPTS string:

## ■ Verify UDC VM

```
-Dcom.broadhop.run.systemId=system-1
```

```
-Dcom.broadhop.run.clusterId=cluster-1
```

### Step 10 Run

```
SSHUSER_PREFERROOT=true /var/qps/bin/control/copytoall.sh /etc/broadhop/controlcenter/qns.conf
```

### Step 11 Modify /etc/broadhop/iomanager/qns.conf to add the following two lines at the top inside the QNS\_OPTS string:

```
-Dcom.broadhop.run.systemId=system-1
```

```
-Dcom.broadhop.run.clusterId=cluster-1
```

### Step 12 Run

```
SSHUSER_PREFERROOT=true /var/qps/bin/control/copytoall.sh /etc/broadhop/iomanager/qns.conf
```

### Step 13 Modify /etc/broadhop/qns.conf by deleting the following two lines at the top:

```
-Dcom.broadhop.run.systemId=system-1
```

```
-Dcom.broadhop.run.clusterId=cluster-1
```

### Step 14 Run

```
SSHUSER_PREFERROOT=true /var/qps/bin/control/copytoall.sh /etc/broadhop/qns.conf
```

### Step 15 Reinit the VMs from installer by executing the following script:

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

### Step 16 Verify if /etc/broadhop/iomanager/qns.conf changes in *Step 11* are reflected in both LB VMs. If changes are not reflected after running `reinit.sh`, repeat *Step 11*.

### Step 17 Restart the services on all required VMs by executing the following script:

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

### Step 18 Run `diagnostics.sh` and verify that the new Admin DB set is available under MongoDB Replica-Sets Status Information.

## Verify UDC VM

1. Verify if all peripheral script output includes details on UDC VMs and no UDC related errors are seen by running the following commands:

- a. `hosts.sh`

- b. `diagnostics.sh`

- c. `about.sh`
  - d. `statusall.sh`
  - e. `list_installed_features.sh`
2. From UDC VM, run `ssh` for all other VMs from UDC nodes to verify login is successful without entering the password.

## Verify Installer VM

1. Verify if all peripheral script output includes details on UDC VMs and no UDC related errors are seen by running the following commands:
  - a. `hosts.sh`
  - b. `diagnostics.sh`
  - c. `about.sh`
  - d. `statusall.sh`
  - e. `list_installed_features.sh`
  - f. `runonone.sh udc03 /etc/init.d/qns status`
2. From Installer VM, run `ssh udc<xx>` for all UDC nodes to verify login is successful without entering the password.

## Verify PCRFCClient VMs

1. Verify if all peripheral script output includes details on UDC VMs and no UDC related errors are seen by running the following commands:
  - a. `hosts.sh`
  - b. `diagnostics.sh`
  - c. `about.sh`
  - d. `statusall.sh`
  - e. `list_installed_features.sh`
2. From PCRFCClient VM, run `ssh udc<xx>` for all UDC nodes to verify login is successful without entering the password.

## Verify LB VMs

1. Verify if all peripheral script output includes details on UDC VMs and no UDC related errors are seen by running the following commands:
  - a. `hosts.sh`
  - b. `about.sh`
  - c. `statusall.sh`
  - d. `list_installed_features.sh`
2. From LB VM, run `ssh udc<xx>` for all UDC nodes to verify login is successful without entering the password.

# Modifications in Platform Files

The platform files have some changes and modifications in them. This section describes the diff output of each of the files and the changes in the files.

The following table describes various changes in each of the platform scripts/files:

Diff Output Indicator	Type of Change in File
< hello	Deprecated lines to be deleted.
> goodbye	Lines to be added.
39,40c39,40	<p>The letter in the middle explains how the file was changed. Given below are the different letters used:</p> <ul style="list-style-type: none"> <li>• C – Changed</li> <li>• A – added</li> <li>• D – deleted</li> </ul> <p>The numbers before the letter denotes the line numbers in the original file. The numbers after the letter denotes the line numbers in the updated file.</p>
<b>Test</b>	Lines in bold indicate that they need to be added or modified.

**NOTE:** These changes are applicable to 13.0.0 platform scripts/files.

## jconfig.py

### File Name

/var/qps/install/current/scripts/deployer/support/jconfig.py

### Diff Output

```

127a128
>     cluster.UDC = 0

146a148,149
>
>         elif vmType == "udc":
>
>             cluster.UDC += 1

694a698,699
>
>         line = "udc_instances=%s\n"%clusterInfo.UDC

```

```
> f.write(line)
```

## reinit.sh

### File Name

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

### Diff Output

```
39,40c39,40
```

```
< fetch_list_of_host_required_reboot
"/var/qps/install/current/scripts/bin/support/hosts-all.sh '^(sessionmgr[0-9]+|portal[0-9]+|pcrfclient[0-9]+|qns[0-9]+|lb[0-9]+)$' | tr '\n' ' '`
```

```
< for host in `/var/qps/install/current/scripts/bin/support/hosts-all.sh
'^(sessionmgr[0-9]+|portal[0-9]+|pcrfclient[0-9]+|qns[0-9]+|lb[0-9]+)$'`
```

```
---
```

```
> fetch_list_of_host_required_reboot
"/var/qps/install/current/scripts/bin/support/hosts-all.sh '^(sessionmgr[0-9]+|portal[0-9]+|pcrfclient[0-9]+|udc[0-9]+|qns[0-9]+|lb[0-9]+)$' | tr '\n' ' '`
```

```
> for host in `/var/qps/install/current/scripts/bin/support/hosts-all.sh
'^(sessionmgr[0-9]+|portal[0-9]+|pcrfclient[0-9]+|udc[0-9]+|qns[0-9]+|lb[0-9]+)$'`
```

```
99a100,113
```

```
>
```

```
> UDC_VMS=`/bin/echo $REBOOT_NEED_HOST_LIST | tr ' ' '\n' | grep '^udc[0-9]'`
```

```
> i=0
```

```
> for udc in $UDC_VMS
```

```
> do
```

```
> rem=$(( $i % 2 ))
```

```
> if [[ $rem -eq 0 ]]; then
```

```
> REBOOT_SET1="$REBOOT_SET1 $udc"
```

```
> else
```

```
> REBOOT_SET2="$REBOOT_SET2 $udc"
```

```
> fi
```

```

>         i=`expr $i + 1`
>     done
>

```

## pp\_status.sh

### File Name

/var/qps/install/current/scripts/upgrade/pp\_status.sh

### Diff Output

```

41c41
<         hosts=`/var/qps/install/current/scripts/bin/support/hosts-all.sh
'^ (sessionmgr[0-9]+|portal[0-9]+|pcrfclient[0-9]+|qns[0-9]+|lb[0-9]+)'`
---
>         hosts=`/var/qps/install/current/scripts/bin/support/hosts-all.sh
'^ (sessionmgr[0-9]+|portal[0-9]+|pcrfclient[0-9]+|udc[0-9]+|qns[0-9]+|lb[0-9]+)'`

```

## copytoall.sh

### File Name

/var/qps/install/current/scripts/bin/control/copytoall.sh

### Diff Output

```

48c48
< for HOST in $($BIN_SUPPORT/hosts-all.sh '^pcrfclient[0-9][0-9]$|^lb[0-9][0-9]$|^lbs[0-9][0-9]$|^qns[0-9][0-9]$|^qns[0-9][0-9]$|^qns[0-9][0-9]$|^sessionmgr[0-9][0-9]$|^portal[0-9][0-9]$|^portallb[0-9][0-9]$|^*arbiter)' 'diam|esx'); do
---
> for HOST in $($BIN_SUPPORT/hosts-all.sh '^pcrfclient[0-9][0-9]$|^lb[0-9][0-9]$|^lbs[0-9][0-9]$|^udc[0-9][0-9]$|^qns[0-9][0-9]$|^qns[0-9][0-9]$|^sessionmgr[0-9][0-9]$|^portal[0-9][0-9]$|^portallb[0-9][0-9]$|^*arbiter)' 'diam|esx'); do

```

## hosts.sh

### File Name

/var/qps/install/current/scripts/bin/support/hosts.sh

### Diff Output

```
21c21
< "$BIN_HOME/support/hosts-all.sh" '^(\pcrfclient[0-9]+|qns[0-9]+|lb[0-9]+)$' 'diam|esx'
---
> "$BIN_HOME/support/hosts-all.sh" '^(\pcrfclient[0-9]+|udc[0-9]+|qns[0-9]+|lb[0-9]+)$'
'diam|esx'
```

## diagnostics.sh

### File Name

/var/qps/install/current/scripts/bin/diag/diagnostics.sh

### Diff Output

```
145c145
<     if [[ $rolename =~ ^(sessionmgr|portal|lb|qns)[0-9]+$ ]]
---
>     if [[ $rolename =~ ^(sessionmgr|portal|lb|udc|qns)[0-9]+$ ]]

151c151
<     QNS_VMS=$( $BIN_SUPPORT/hosts.sh | grep qns)
---
>     QNS_VMS=$( $BIN_SUPPORT/hosts.sh | egrep 'qns|udc')

338,339c338,343
<         check_port $QNS 8090 "Control Center"
<         #check_port $QNS 9443 "Grafana"
---
>         ## not udc
```



```

>         if [[ ! $QNS =~ ^udc ]]
>
>         then
>
>             check_port $QNS 8090 "Control Center"
>
>             #check_port $QNS 9443 "Grafana"
>
>         fi

```

## runonall.sh

### File Name

/var/qps/install/current/scripts/bin/control/runonall.sh

### Diff Output

```

16c16
< for HOST in $($BIN_SUPPORT/hosts-all.sh '^(sessionmgr[0-9]+|portal[0-9]+|pcrfclient[0-9]+|qns[0-9]+|lb[0-9]+)$'); do
---
> for HOST in $($BIN_SUPPORT/hosts-all.sh '^(sessionmgr[0-9]+|portal[0-9]+|pcrfclient[0-9]+|udc[0-9]+|qns[0-9]+|lb[0-9]+)$'); do

```

## udc.pp

### File Name

/etc/puppet/modules/qps/manifests/roles/udc.pp

This is a new file based on /etc/puppet/modules/qps/manifests/roles/qns.pp in which all instances of the string “qns” is replaced by “udc”.

### Diff Output

```

1c1
< # == Class: qps::role:qns
---
> # == Class: qps::role:udc
11c11
< class qps::roles::qns inherits qps::roles::qps_base {

```

```
---  
  
> class qps::roles::udc inherits qps::roles::qps_base {  
  
25c25  
  
<   vmType => "qns",  
  
---  
  
>   vmType => "udc",  
  
28c28  
  
<   vmType => "qns",  
  
---  
  
>   vmType => "udc",  
  
33c33  
  
<   Class['qps::collectd_worker'] -> Class['qps']  
  
---  
  
>   Class['qps::collectd_worker'] -> Class['udc']  
  
}
```

## udc.yaml

### File Name

/etc/puppet/modules/qps/nodes/udc.yaml

This is a new file.

### File Content

```
parameters:  
  
  qnsUserName: param  
  
classes:  
  
  qps::roles::udc:
```

## servers

### File Name

```
/var/qps/current_config/etc/broadhop/servers
```

### Diff Output

```
5a6
> lb01=udc_diameter_endpoint
9a11
> lb02=udc_diameter_endpoint
12,13c14,15
< qns03=pcrf
< qns04=pcrf
---
> udc03=udc
> udc04=udc
```

## servers.all

### File Name

```
/var/qps/current_config/etc/broadhop/servers.all
```

### Diff Output

```
4c4
< <cluster-name>-lb01=diameter_endpoint
---
> <cluster-name>-lb01=udc_diameter_endpoint
8c8
< <cluster-name>-lb02=diameter_endpoint
```

## ■ build\_servers\_all.sh

```
---  
  
> <cluster-name>-lb02=udc_diameter_endpoint  
  
11,12c11,12  
  
< <cluster-name>-qns03=pcrf  
< <cluster-name>-qns04=pcrf  
  
---  
  
> <cluster-name>-udc03=udc  
  
> <cluster-name>-udc04=udc
```

## build\_servers\_all.sh

### File Name

/var/qps/install/current/scripts/build/build\_servers\_all.sh

### Diff Output

```
21c21  
  
< CPS_HOSTS=$( $BIN_SUPPORT/hosts-all.sh ' (^pcrfclient[0-9][0-9]$|^lb[0-9][0-9]$|^lbs[0-9][0-9]$|^qns[0-9][0-9]$|^qns[0-9][0-9]$|^qns[0-9][0-9]$)' )  
  
---  
  
> CPS_HOSTS=$( $BIN_SUPPORT/hosts-all.sh ' (^pcrfclient[0-9][0-9]$|^lb[0-9][0-9]$|^lbs[0-9][0-9]$|^udc[0-9][0-9]$|^qns[0-9][0-9]$|^qns[0-9][0-9]$)' )
```

## /images/image-map

### File Name

/var/qps/images/image-map

### Diff Output

```
4a5  
  
> lb=udc_diameter_endpoint  
  
5a7  
  
> udc=udc
```

## /env\_config/image-map

### File Name

/var/qps/env\_config/image-map

### Diff Output

5a6

> **lb=udc\_diameter\_endpoint**

6a7

> **udc=udc**