# CPS UDC Installation Guide, Release 18.1.0 (Restricted Release)

**First Published:** 2018-03-16

**Last Modified:** 2018-03-16

**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

# CONTENTS

# RESTRICTED RELEASE

**IMPORTANT**: This is a Short Term Support (STS) release with availability and use restrictions. Contact your Cisco Account or Support representatives for more information.

# User Data Convergence Installation

This chapter includes the following topics:

- Installing UDC in OpenStack
- Installing UDC in VMware

# Installing UDC in OpenStack

The procedure to install UDC in OpenStack is similar to PCRF Mobile installation in OpenStack.

For more information about installation, see *CPS Installation Guide for OpenStack*.

You need to modify the HEAT, ENV, and YAML configuration files to install UDC in OpenStack.

## Sample Heat Environment File

You can use the following sample file for UDC deployment in OpenStack.

```
parameters:

  base_vm_image_name: Base_18.0.0

  cps_iso_volume_id: "c62ceed2-539d-4541-9ffa-49e5ab24487d"

  cps_iso_image_name: CPS_18.0.0_ISO

  cps_az_1: az-1

  cps_az_2: az-2

  cps_az_3: az-3

  cps_az_4: az-4

  cps_az_5: az-5


  internal_net_name: Internal

  internal_net_cidr: 172.4.32.0/24

  internal_net_gateway: 172.4.32.1


  management_net_name: Management

  management_net_gateway: 4.0.0.1

  management_net_cidr: 4.0.0.0/8


  cluman_internal_ip: 172.4.32.100
```

```
cluman_management_ip: 4.4.32.100

cluman_flavor_name: cluman


lb_internal_vip: 172.4.32.102

lb_management_vip: 4.4.32.102


lb01_flavor_name: lb01

lb01_internal_ip: 172.4.32.103

lb01_management_ip: 4.4.32.103


lb02_flavor_name: lb02

lb02_internal_ip: 172.4.32.104

lb02_management_ip: 4.4.32.104


pcrfclient01_flavor_name: pcrfclient01

pcrfclient01_internal_ip: 172.4.32.105

pcrfclient01_management_ip: 4.4.32.105


pcrfclient02_flavor_name: pcrfclient02

pcrfclient02_internal_ip: 172.4.32.106

pcrfclient02_management_ip: 4.4.32.106


qns_flavor_name: qps

qns01_internal_ip: 172.4.32.107

qns02_internal_ip: 172.4.32.108


udc_flavor_name: udc
```

```
udc01_internal_ip: 172.4.32.109
udc01_management_ip: 4.4.32.109
udc02_internal_ip: 172.4.32.110
udc02_management_ip: 4.4.32.110

sessionmgr_flavor_name: sessionmgr

sessionmgr01_internal_ip: 172.4.32.111
sessionmgr01_management_ip: 4.4.32.111

sessionmgr02_internal_ip: 172.4.32.112
sessionmgr02_management_ip: 4.4.32.112

mongo01_volume_id: "cd395da3-aedb-4b1f-b552-8fe0053d655c"
mongo02_volume_id: "3fdaab95-4215-4f43-ae48-493fb1089f08"

svn01_volume_id: "bc90cafd-4339-4e1c-ae47-28da6dbafd5d"
svn02_volume_id: "28545b99-bd6b-4fc6-be8a-fb238451831e"
```

# Sample Heat Template File

You can use the following sample file for UDC deployment in OpenStack.

```
description: A minimal CPS UDC deployment

parameters:
#=========================
# Global Parameters
```

```
#=========================
  base_vm_image_name:
    type: string
    label: base vm image name
    description: name of the base vm as imported into glance
  cps_iso_image_name:
    type: string
    label: cps iso image name
    description: name of the cps iso as imported into glance
  cps_install_type:
    type: string
    label: cps installation type (mobile|wifi|mog|pats|arbiter|dra)
    description: cps installation type (mobile|wifi|mog|pats|arbiter|dra)
    default: mobile
  cps_az_1:
    type: string
    label: first availability zone
    description: az for "first half" of cluster
    default: nova
  cps_az_2:
    type: string
    label: second availability zone
    description: az for "second half" of cluster
    default: nova
  cps_az_3:
    type: string
    label: second availability zone
```

```
    description: az for "second half" of cluster
    default: nova
  cps_az_4:
    type: string
    label: second availability zone
    description: az for "second half" of cluster
    default: nova
  cps_az_5:
    type: string
    label: second availability zone
    description: az for "second half" of cluster
    default: nova
#=========================
# Network Parameters
#=========================
  internal_net_name:
    type: string
    label: internal network name
    description: name of the internal network
  internal_net_cidr:
    type: string
    label: cps internal cidr
    description: cidr of internal subnet
  internal_net_gateway:
    type: string
    label: cps internal cidr
    description: cidr of internal subnet
```

```
  management_net_name:
    type: string
    label: management network name
    description: name of the management network
  management_net_cidr:
    type: string
    label: cps management cidr
    description: cidr of management subnet
  management_net_gateway:
    type: string
    label: management network gateway
    description: gateway on management network
    default: ""


  cps_iso_volume_id:
    type: string
    label: cps iso volume id
    description: uuid of the cps iso volume


#=========================
# Instance Parameters
#=========================
  cluman_flavor_name:
    type: string
    label: cluman flavor name
    description: flavor cluman vm will use
    default: cluman
```

```
cluman_internal_ip:
  type: string
  label: internal ip of cluster manager
  description: internal ip of cluster manager
cluman_management_ip:
  type: string
  label: management ip of cluster manager
  description: management ip of cluster manager
sessionmgr01_internal_ip:
  type: string
  label: session manager ip
  description: session manager ip
sessionmgr02_internal_ip:
  type: string
  label: session manager ip
  description: session manager ip
lb_internal_vip:
  type: string
  label: session manager ip
  description: session manager ip
lb_management_vip:
  type: string
  label: session manager ip
  description: session manager ip
lb01_flavor_name:
  type: string
  label: session manager ip
```

```
    description: session manager ip
  lb01_internal_ip:
    type: string
    label: session manager ip
    description: session manager ip
  lb01_management_ip:
    type: string
    label: session manager ip
    description: session manager ip
  lb02_flavor_name:
    type: string
    label: session manager ip
    description: session manager ip
  lb02_internal_ip:
    type: string
    label: session manager ip
    description: session manager ip
  lb02_management_ip:
    type: string
    label: session manager ip
    description: session manager ip
  pcrfclient01_flavor_name:
    type: string
    label: session manager ip
    description: session manager ip
  pcrfclient01_internal_ip:
    type: string
```

```
    label: session manager ip
    description: session manager ip
  pcrfclient01_management_ip:
    type: string
    label: session manager ip
    description: session manager ip
  pcrfclient02_flavor_name:
    type: string
    label: session manager ip
    description: session manager ip
  pcrfclient02_internal_ip:
    type: string
    label: session manager ip
    description: session manager ip
  pcrfclient02_management_ip:
    type: string
    label: session manager ip
    description: session manager ip
  qns01_internal_ip:
    type: string
    label: session manager ip
    description: session manager ip
  qns_flavor_name:
    type: string
    label: session manager ip
    description: session manager ip
  qns02_internal_ip:
```

```
    type: string
    label: session manager ip
    description: session manager ip
  udc_flavor_name:
    type: string
    label: udc flavor name
    description: flavor udc vm will use
    default: udc
  udc01_management_ip:
    type: string
    label: udc01 manager ip
    description: udc02 manager ip
  udc02_management_ip:
    type: string
    label: udc02 manager ip
    description: udc02 manager ip
  udc01_internal_ip:
    type: string
    label: session manager ip
    description: session manager ip
  udc02_internal_ip:
    type: string
    label: session manager ip
    description: session manager ip
  sessionmgr_flavor_name:
    type: string
    label: session manager ip
```

```
  description: session manager ip
sessionmgr01_internal_ip:
  type: string
  label: session manager ip
  description: session manager ip
sessionmgr01_management_ip:
  type: string
  label: session manager ip
  description: session manager ip
sessionmgr02_internal_ip:
  type: string
  label: session manager ip
  description: session manager ip
sessionmgr02_management_ip:
  type: string
  label: session manager ip
  description: session manager ip
mongo01_volume_id:
  type: string
  label: session manager ip
  description: session manager ip
mongo02_volume_id:
  type: string
  label: session manager ip
  description: session manager ip
svn01_volume_id:
  type: string
```

```
    label: session manager ip
    description: session manager ip
  svn02_volume_id:
    type: string
    label: session manager ip
    description: session manager ip


resources:
#=========================
# Instances
#=========================
  cluman:
    type: OS::Nova::Server
    properties:
      availability_zone: { get_param: cps_az_1 }
      config_drive: "True"
      image: { get_param: base_vm_image_name }
      flavor: { get_param: cluman_flavor_name }
      networks:
        - port: { get_resource: cluman_internal_port }
        - port: { get_resource: cluman_management_port }
      block_device_mapping:
        - device_name: vdb
          volume_id: { get_param: cps_iso_volume_id }
      user_data_format: RAW
      user_data: { get_resource: cluman_config }
  cluman_internal_port:
```

```
    type: OS::Neutron::Port
    properties:
      network: { get_param: internal_net_name }
      fixed_ips: [{ ip_address: { get_param: cluman_internal_ip }}]
  cluman_management_port:
    type: OS::Neutron::Port
    properties:
      network: { get_param: management_net_name }
      fixed_ips: [{ ip_address: { get_param: cluman_management_ip }}]
  cluman_config:
    type: OS::Heat::CloudConfig
    properties:
      cloud_config:
        write_files:
          - path: /var/lib/cloud/instance/payload/launch-params
            permissions: "0644"
          - path: /etc/sysconfig/network-scripts/ifcfg-eth0
            permissions: "0644"
            content:
              str_replace:
                template: |
                  DEVICE=eth0
                  BOOTPROTO=none
                  NM_CONTROLLED=no
                  IPADDR=$ip
                params:
                  $ip: { get_param: cluman_internal_ip }
```

```
- path: /etc/sysconfig/network-scripts/ifcfg-eth1
  permissions: "0644"
  content:
    str_replace:
      template: |
        DEVICE=eth1
        BOOTPROTO=none
        NM_CONTROLLED=no
        IPADDR=$ip
        GATEWAY=$gateway
      params:
        $ip: { get_param: cluman_management_ip }
        $gateway: { get_param: management_net_gateway }
- path: /root/.autoinstall.sh
  permissions: "0755"
  content:
    str_replace:
      template: |
        #!/bin/bash
        if [[ -d /mnt/iso ]] && [[ -f /mnt/iso/install.sh ]]; then
        /mnt/iso/install.sh << EOF
        $install_type
        y
        1
        EOF
        fi
      params:
```

```
                $install_type: { get_param: cps_install_type }
      mounts:
        - [ /dev/vdb, /mnt/iso, iso9660, "auto,ro", 0, 0 ]
      runcmd:
        - str_replace:
            template: echo $ip installer >> /etc/hosts
            params:
              $ip: { get_param: cluman_internal_ip }
        - str_replace:
            template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth0
            params:
              $cidr: { get_param: internal_net_cidr }
        - str_replace:
            template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth1
            params:
              $cidr: { get_param: management_net_cidr }
        - ifdown eth0 && ifup eth0
        - ifdown eth1 && ifup eth1
        - echo HOSTNAME=cluman >> /etc/sysconfig/network
        - hostname cluman
        - /root/.autoinstall.sh
  lb01:
    type: OS::Nova::Server
    properties:
      availability_zone: { get_param: cps_az_3 }
      config_drive: "True"
      image: { get_param: base_vm_image_name }
```

```
      flavor: { get_param: lb01_flavor_name }
      networks:
        - port: { get_resource: lb01_internal_port }
        - port: { get_resource: lb01_management_port }
      user_data_format: RAW
      user_data: { get_resource: lb01_config }
  lb01_internal_port:
    type: OS::Neutron::Port
    properties:
      network: { get_param: internal_net_name }
      fixed_ips: [{ ip_address: { get_param: lb01_internal_ip }}]
      allowed_address_pairs:
        - ip_address: { get_param: lb_internal_vip }
  lb01_management_port:
    type: OS::Neutron::Port
    properties:
      network: { get_param: management_net_name }
      fixed_ips: [{ ip_address: { get_param: lb01_management_ip }}]
      allowed_address_pairs:
        - ip_address: { get_param: lb_management_vip }
  lb01_config:
    type: OS::Heat::CloudConfig
    properties:
      cloud_config:
        write_files:
          - path: /var/lib/cloud/instance/payload/launch-params
          - path: /etc/broadhop.profile
```

```
    content: "NODE_TYPE=lb01\n"
  - path: /etc/sysconfig/network-scripts/ifcfg-eth0
    content:
      str_replace:
        template: |
          DEVICE=eth0
          BOOTPROTO=none
          NM_CONTROLLED=no
          IPADDR=$ip
        params:
          $ip: { get_param: lb01_internal_ip }
  - path: /etc/sysconfig/network-scripts/ifcfg-eth1
    content:
      str_replace:
        template: |
          DEVICE=eth1
          BOOTPROTO=none
          NM_CONTROLLED=no
          IPADDR=$ip
          GATEWAY=$gateway
        params:
          $ip: { get_param: lb01_management_ip }
          $gateway: { get_param: management_net_gateway }
runcmd:
  - str_replace:
      template: echo $ip installer >> /etc/hosts
      params:
```

```
                $ip: { get_param: cluman_internal_ip }
          - str_replace:
              template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth0
              params:
                $cidr: { get_param: internal_net_cidr }
          - str_replace:
              template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth1
              params:
                $cidr: { get_param: management_net_cidr }
          - ifdown eth0 && ifup eth0
          - ifdown eth1 && ifup eth1
          - echo HOSTNAME=lb01 >> /etc/sysconfig/network
          - hostname lb01


  lb02:
    type: OS::Nova::Server
    properties:
      availability_zone: { get_param: cps_az_2 }
      config_drive: "True"
      image: { get_param: base_vm_image_name }
      flavor: { get_param: lb02_flavor_name }
      networks:
        - port: { get_resource: lb02_internal_port }
        - port: { get_resource: lb02_management_port }
      user_data_format: RAW
      user_data: { get_resource: lb02_config }
  lb02_internal_port:
```

```
  type: OS::Neutron::Port
  properties:
    network: { get_param: internal_net_name }
    fixed_ips: [{ ip_address: { get_param: lb02_internal_ip }}]
    allowed_address_pairs:
      - ip_address: { get_param: lb_internal_vip }
lb02_management_port:
  type: OS::Neutron::Port
  properties:
    network: { get_param: management_net_name }
    fixed_ips: [{ ip_address: { get_param: lb02_management_ip }}]
    allowed_address_pairs:
      - ip_address: { get_param: lb_management_vip }
lb02_config:
  type: OS::Heat::CloudConfig
  properties:
    cloud_config:
      write_files:
        - path: /var/lib/cloud/instance/payload/launch-params
        - path: /etc/broadhop.profile
          content: "NODE_TYPE=lb02\n"
        - path: /etc/sysconfig/network-scripts/ifcfg-eth0
          content:
            str_replace:
              template: |
                DEVICE=eth0
                BOOTPROTO=none
```

```
            NM_CONTROLLED=no
            IPADDR=$ip
          params:
            $ip: { get_param: lb02_internal_ip }
    - path: /etc/sysconfig/network-scripts/ifcfg-eth1
      content:
        str_replace:
          template: |
            DEVICE=eth1
            BOOTPROTO=none
            NM_CONTROLLED=no
            IPADDR=$ip
            GATEWAY=$gateway
          params:
            $ip: { get_param: lb02_management_ip }
            $gateway: { get_param: management_net_gateway }
  runcmd:
    - str_replace:
        template: echo $ip installer >> /etc/hosts
        params:
          $ip: { get_param: cluman_internal_ip }
    - str_replace:
        template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth0
        params:
          $cidr: { get_param: internal_net_cidr }
    - str_replace:
        template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth1
```

```
            params:
              $cidr: { get_param: management_net_cidr }
        - ifdown eth0 && ifup eth0
        - ifdown eth1 && ifup eth1
        - echo HOSTNAME=lb02 >> /etc/sysconfig/network
        - hostname lb02


pcrfclient01:
  type: OS::Nova::Server
  properties:
    availability_zone: { get_param: cps_az_5 }
    config_drive: "True"
    image: { get_param: base_vm_image_name }
    flavor: { get_param: pcrfclient01_flavor_name }
    networks:
      - port: { get_resource: pcrfclient01_internal_port }
      - port: { get_resource: pcrfclient01_management_port }
    block_device_mapping:
      - device_name: vdb
        volume_id: { get_param: svn01_volume_id }
    user_data_format: RAW
    user_data: { get_resource: pcrfclient01_config }
pcrfclient01_internal_port:
  type: OS::Neutron::Port
  properties:
    network: { get_param: internal_net_name }
    fixed_ips: [{ ip_address: { get_param: pcrfclient01_internal_ip }}]
```

```
pcrfclient01_management_port:
  type: OS::Neutron::Port
  properties:
    network: { get_param: management_net_name }
    fixed_ips: [{ ip_address: { get_param: pcrfclient01_management_ip }}]
pcrfclient01_config:
  type: OS::Heat::CloudConfig
  properties:
    cloud_config:
      write_files:
        - path: /var/lib/cloud/instance/payload/launch-params
        - path: /etc/broadhop.profile
          content: "NODE_TYPE=pcrfclient01\n"
        - path: /etc/sysconfig/network-scripts/ifcfg-eth0
          content:
            str_replace:
              template: |
                DEVICE=eth0
                BOOTPROTO=none
                NM_CONTROLLED=no
                IPADDR=$ip
              params:
                $ip: { get_param: pcrfclient01_internal_ip }
        - path: /etc/sysconfig/network-scripts/ifcfg-eth1
          content:
            str_replace:
              template: |
```

```
                DEVICE=eth1
                BOOTPROTO=none
                NM_CONTROLLED=no
                IPADDR=$ip
                GATEWAY=$gateway
              params:
                $ip: { get_param: pcrfclient01_management_ip }
                $gateway: { get_param: management_net_gateway }
        runcmd:
          - str_replace:
              template: echo $ip installer >> /etc/hosts
              params:
                $ip: { get_param: cluman_internal_ip }
          - str_replace:
              template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth0
              params:
                $cidr: { get_param: internal_net_cidr }
          - str_replace:
              template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth1
              params:
                $cidr: { get_param: management_net_cidr }
          - ifdown eth0 && ifup eth0
          - ifdown eth1 && ifup eth1
          - echo HOSTNAME=pcrfclient01 >> /etc/sysconfig/network
          - hostname pcrfclient01


    pcrfclient02:
```

```
    type: OS::Nova::Server
    properties:
      availability_zone: { get_param: cps_az_4 }
      config_drive: "True"
      image: { get_param: base_vm_image_name }
      flavor: { get_param: pcrfclient02_flavor_name }
      networks:
        - port: { get_resource: pcrfclient02_internal_port }
        - port: { get_resource: pcrfclient02_management_port }
      block_device_mapping:
        - device_name: vdb
          volume_id: { get_param: svn02_volume_id }
      user_data_format: RAW
      user_data: { get_resource: pcrfclient02_config }
  pcrfclient02_internal_port:
    type: OS::Neutron::Port
    properties:
      network: { get_param: internal_net_name }
      fixed_ips: [{ ip_address: { get_param: pcrfclient02_internal_ip }}]
  pcrfclient02_management_port:
    type: OS::Neutron::Port
    properties:
      network: { get_param: management_net_name }
      fixed_ips: [{ ip_address: { get_param: pcrfclient02_management_ip }}]
  pcrfclient02_config:
    type: OS::Heat::CloudConfig
    properties:
```

```
cloud_config:
  write_files:
    - path: /var/lib/cloud/instance/payload/launch-params
    - path: /etc/broadhop.profile
      content: "NODE_TYPE=pcrfclient02\n"
    - path: /etc/sysconfig/network-scripts/ifcfg-eth0
      content:
        str_replace:
          template: |
            DEVICE=eth0
            BOOTPROTO=none
            NM_CONTROLLED=no
            IPADDR=$ip
          params:
            $ip: { get_param: pcrfclient02_internal_ip }
    - path: /etc/sysconfig/network-scripts/ifcfg-eth1
      content:
        str_replace:
          template: |
            DEVICE=eth1
            BOOTPROTO=none
            NM_CONTROLLED=no
            IPADDR=$ip
            GATEWAY=$gateway
          params:
            $ip: { get_param: pcrfclient02_management_ip }
            $gateway: { get_param: management_net_gateway }
```

```
        runcmd:
          - str_replace:
              template: echo $ip installer >> /etc/hosts
              params:
                $ip: { get_param: cluman_internal_ip }
          - str_replace:
              template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth0
              params:
                $cidr: { get_param: internal_net_cidr }
          - str_replace:
              template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth1
              params:
                $cidr: { get_param: management_net_cidr }
          - ifdown eth0 && ifup eth0
          - ifdown eth1 && ifup eth1
          - echo HOSTNAME=pcrfclient02 >> /etc/sysconfig/network
          - hostname pcrfclient02


  qns01:
    type: OS::Nova::Server
    properties:
      availability_zone: { get_param: cps_az_3 }
      config_drive: "True"
      image: { get_param: base_vm_image_name }
      flavor: { get_param: qns_flavor_name }
      networks:
        - port: { get_resource: qns01_internal_port }
```

```
    user_data_format: RAW
    user_data: { get_resource: qns01_config }
qns01_internal_port:
  type: OS::Neutron::Port
  properties:
    network: { get_param: internal_net_name }
    fixed_ips: [{ ip_address: { get_param: qns01_internal_ip }}]
qns01_config:
  type: OS::Heat::CloudConfig
  properties:
    cloud_config:
      write_files:
        - path: /var/lib/cloud/instance/payload/launch-params
        - path: /etc/broadhop.profile
          content: "NODE_TYPE=qns01\n"
        - path: /etc/sysconfig/network-scripts/ifcfg-eth0
          content:
            str_replace:
              template: |
                DEVICE=eth0
                BOOTPROTO=none
                NM_CONTROLLED=no
                IPADDR=$ip
              params:
                $ip: { get_param: qns01_internal_ip }
      runcmd:
        - str_replace:
```

```
            template: echo $ip installer >> /etc/hosts
            params:
              $ip: { get_param: cluman_internal_ip }
        - str_replace:
            template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth0
            params:
              $cidr: { get_param: internal_net_cidr }
        - ifdown eth0 && ifup eth0
        - echo HOSTNAME=qns01 >> /etc/sysconfig/network
        - hostname qns01


qns02:
  type: OS::Nova::Server
  properties:
    availability_zone: { get_param: cps_az_2 }
    config_drive: "True"
    image: { get_param: base_vm_image_name }
    flavor: { get_param: qns_flavor_name }
    networks:
      - port: { get_resource: qns02_internal_port }
    user_data_format: RAW
    user_data: { get_resource: qns02_config }
qns02_internal_port:
  type: OS::Neutron::Port
  properties:
    network: { get_param: internal_net_name }
    fixed_ips: [{ ip_address: { get_param: qns02_internal_ip }}]
```

```
qns02_config:
  type: OS::Heat::CloudConfig
  properties:
    cloud_config:
      write_files:
        - path: /var/lib/cloud/instance/payload/launch-params
        - path: /etc/broadhop.profile
          content: "NODE_TYPE=qns02\n"
        - path: /etc/sysconfig/network-scripts/ifcfg-eth0
          content:
            str_replace:
              template: |
                DEVICE=eth0
                BOOTPROTO=none
                NM_CONTROLLED=no
                IPADDR=$ip
              params:
                $ip: { get_param: qns02_internal_ip }
      runcmd:
        - str_replace:
            template: echo $ip installer >> /etc/hosts
            params:
              $ip: { get_param: cluman_internal_ip }
        - str_replace:
            template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth0
            params:
              $cidr: { get_param: internal_net_cidr }
```

```
          - ifdown eth0 && ifup eth0
          - echo HOSTNAME=qns02 >> /etc/sysconfig/network
          - hostname qns02
    udc01:
      type: OS::Nova::Server
      properties:
        availability_zone: { get_param: cps_az_5 }
        config_drive: "True"
        image: { get_param: base_vm_image_name }
        flavor: { get_param: udc_flavor_name }
        networks:
          - port: { get_resource: udc01_internal_port }
        user_data_format: RAW
        user_data: { get_resource: udc01_config }
    udc01_internal_port:
      type: OS::Neutron::Port
      properties:
        network: { get_param: internal_net_name }
        fixed_ips: [{ ip_address: { get_param: udc01_internal_ip }}]
    udc01_management_port:
      type: OS::Neutron::Port
      properties:
        network: { get_param: management_net_name }
        fixed_ips: [{ ip_address: { get_param: udc01_management_ip }}]
    udc01_config:
      type: OS::Heat::CloudConfig
      properties:
```

```
cloud_config:
  write_files:
    - path: /var/lib/cloud/instance/payload/launch-params
    - path: /etc/broadhop.profile
      content: "NODE_TYPE=udc01\n"
    - path: /etc/sysconfig/network-scripts/ifcfg-eth0
      content:
        str_replace:
          template: |
            DEVICE=eth0
            BOOTPROTO=none
            NM_CONTROLLED=no
            IPADDR=$ip
          params:
            $ip: { get_param: udc01_internal_ip }
  runcmd:
    - str_replace:
        template: echo $ip installer >> /etc/hosts
        params:
          $ip: { get_param: cluman_internal_ip }
    - str_replace:
        template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth0
        params:
          $cidr: { get_param: internal_net_cidr }
    - ifdown eth0 && ifup eth0
    - echo HOSTNAME=udc01 >> /etc/sysconfig/network
    - hostname udc01
```

```
udc02:
  type: OS::Nova::Server
  properties:
    availability_zone: { get_param: cps_az_4 }
    config_drive: "True"
    image: { get_param: base_vm_image_name }
    flavor: { get_param: udc_flavor_name }
    networks:
      - port: { get_resource: udc02_internal_port }
    user_data_format: RAW
    user_data: { get_resource: udc02_config }
udc02_internal_port:
  type: OS::Neutron::Port
  properties:
    network: { get_param: internal_net_name }
    fixed_ips: [{ ip_address: { get_param: udc02_internal_ip }}]
udc02_management_port:
  type: OS::Neutron::Port
  properties:
    network: { get_param: management_net_name }
    fixed_ips: [{ ip_address: { get_param: udc02_management_ip }}]
udc02_config:
  type: OS::Heat::CloudConfig
  properties:
    cloud_config:
      write_files:
```

```
      - path: /var/lib/cloud/instance/payload/launch-params
      - path: /etc/broadhop.profile
        content: "NODE_TYPE=udc02\n"
      - path: /etc/sysconfig/network-scripts/ifcfg-eth0
        content:
          str_replace:
            template: |
              DEVICE=eth0
              BOOTPROTO=none
              NM_CONTROLLED=no
              IPADDR=$ip
            params:
              $ip: { get_param: udc02_internal_ip }
    runcmd:
      - str_replace:
          template: echo $ip installer >> /etc/hosts
          params:
            $ip: { get_param: cluman_internal_ip }
      - str_replace:
          template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth0
          params:
            $cidr: { get_param: internal_net_cidr }
      - ifdown eth0 && ifup eth0
      - echo HOSTNAME=udc02 >> /etc/sysconfig/network
      - hostname udc02
sessionmgr01:
  type: OS::Nova::Server
```

```
    properties:
      availability_zone: { get_param: cps_az_5 }
      config_drive: "True"
      image: { get_param: base_vm_image_name }
      flavor: { get_param: sessionmgr_flavor_name }
      networks:
        - port: { get_resource: sessionmgr01_internal_port }
        - port: { get_resource: sessionmgr01_management_port }
      block_device_mapping:
        - device_name: vdb
          volume_id: { get_param: mongo01_volume_id }
      user_data_format: RAW
      user_data: { get_resource: sessionmgr01_config }
  sessionmgr01_internal_port:
    type: OS::Neutron::Port
    properties:
      network: { get_param: internal_net_name }
      fixed_ips: [{ ip_address: { get_param: sessionmgr01_internal_ip }}]
  sessionmgr01_management_port:
    type: OS::Neutron::Port
    properties:
      network: { get_param: management_net_name }
      fixed_ips: [{ ip_address: { get_param: sessionmgr01_management_ip }}]
  sessionmgr01_config:
    type: OS::Heat::CloudConfig
    properties:
      cloud_config:
```

```
write_files:
  - path: /var/lib/cloud/instance/payload/launch-params
  - path: /etc/broadhop.profile
    content: "NODE_TYPE=sessionmgr01\n"
  - path: /etc/sysconfig/network-scripts/ifcfg-eth0
    content:
      str_replace:
        template: |
          DEVICE=eth0
          BOOTPROTO=none
          NM_CONTROLLED=no
          IPADDR=$ip
        params:
          $ip: { get_param: sessionmgr01_internal_ip }
  - path: /etc/sysconfig/network-scripts/ifcfg-eth1
    content:
      str_replace:
        template: |
          DEVICE=eth1
          BOOTPROTO=none
          NM_CONTROLLED=no
          IPADDR=$ip
          GATEWAY=$gateway
        params:
          $ip: { get_param: sessionmgr01_management_ip }
          $gateway: { get_param: management_net_gateway }
runcmd:
```

```
            - str_replace:
                template: echo $ip installer >> /etc/hosts
                params:
                  $ip: { get_param: cluman_internal_ip }
            - str_replace:
                template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth0
                params:
                  $cidr: { get_param: internal_net_cidr }
            - str_replace:
                template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth1
                params:
                  $cidr: { get_param: management_net_cidr }
            - ifdown eth0 && ifup eth0
            - ifdown eth1 && ifup eth1
            - echo HOSTNAME=sessionmgr01 >> /etc/sysconfig/network
            - hostname sessionmgr01
    sessionmgr02:
      type: OS::Nova::Server
      properties:
        availability_zone: { get_param: cps_az_4 }
        config_drive: "True"
        image: { get_param: base_vm_image_name }
        flavor: { get_param: sessionmgr_flavor_name }
        networks:
          - port: { get_resource: sessionmgr02_internal_port }
          - port: { get_resource: sessionmgr02_management_port }
        block_device_mapping:
```

```
      - device_name: vdb
        volume_id: { get_param: mongo02_volume_id }
    user_data_format: RAW
    user_data: { get_resource: sessionmgr02_config }
sessionmgr02_internal_port:
  type: OS::Neutron::Port
  properties:
    network: { get_param: internal_net_name }
    fixed_ips: [{ ip_address: { get_param: sessionmgr02_internal_ip }}]
sessionmgr02_management_port:
  type: OS::Neutron::Port
  properties:
    network: { get_param: management_net_name }
    fixed_ips: [{ ip_address: { get_param: sessionmgr02_management_ip }}]
sessionmgr02_config:
  type: OS::Heat::CloudConfig
  properties:
    cloud_config:
      write_files:
        - path: /var/lib/cloud/instance/payload/launch-params
        - path: /etc/broadhop.profile
          content: "NODE_TYPE=sessionmgr02\n"
        - path: /etc/sysconfig/network-scripts/ifcfg-eth0
          content:
            str_replace:
              template: |
                DEVICE=eth0
```

```
            BOOTPROTO=none
            NM_CONTROLLED=no
            IPADDR=$ip
          params:
            $ip: { get_param: sessionmgr02_internal_ip }
    - path: /etc/sysconfig/network-scripts/ifcfg-eth1
      content:
        str_replace:
          template: |
            DEVICE=eth1
            BOOTPROTO=none
            NM_CONTROLLED=no
            IPADDR=$ip
            GATEWAY=$gateway
          params:
            $ip: { get_param: sessionmgr02_management_ip }
            $gateway: { get_param: management_net_gateway }
  runcmd:
    - str_replace:
        template: echo $ip installer >> /etc/hosts
        params:
          $ip: { get_param: cluman_internal_ip }
    - str_replace:
        template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth0
        params:
          $cidr: { get_param: internal_net_cidr }
    - str_replace:
```

```
        template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth1
        params:
          $cidr: { get_param: management_net_cidr }
    - ifdown eth0 && ifup eth0
    - ifdown eth1 && ifup eth1
    - echo HOSTNAME=sessionmgr02 >> /etc/sysconfig/network
    - hostname sessionmgr02
```

## Adding UDC Cluster

**Step 1** Navigate to cluman.

**Step 2** Navigate to the following directory:

```
/var/qps/config/mobile/pb_config/configuration
```

**Step 3** Create the following files to configure `cluster-udc`:

**UdcClientPluginConfiguration-_2mY7MPdAEeeL0YbxIz3Brw.xmi**

```
<?xml version="1.0" encoding="UTF-8"?>
<udcclient:UdcClientPluginConfiguration
xmlns:udcclient="http://www.broadhop.com/udcclient" id="_2mY7MPdAEeeL0YbxIz3Brw"
   frontEndId="default"/>
```

**Cluster-default-_KYGtEPaPEeeL0YbxIz3Brw.xmi**

```
<?xml version="1.0" encoding="UTF-8"?>
<runtime:Cluster xmlns:runtime="http://broadhop.com/runtime" id="_KYGtEPaPEeeL0YbxIz3Brw"
    name="cluster-udc">
  <adminDb
      id="_KYGtEfaPEeeL0YbxIz3Brw"
      secondaryIpAddress="sessionmgr02"
      port="27731"/>
</runtime:Cluster>
```

**Step 4** Modify `System-default-_LyqKICHAEeGgHPPGT0vlCA.xmi` to add the following lines as part of the `runtime:System`:

```
  <pluginConfigurations

      xsi:type="udcclient:UdcClientPluginConfiguration"

      href="UdcClientPluginConfiguration-
_2mY7MPdAEeeL0YbxIz3Brw.xmi#_2mY7MPdAEeeL0YbxIz3Brw"/>

  <clusters

      href="Cluster-default-_KYGtEPaPEeeL0YbxIz3Brw.xmi#_KYGtEPaPEeeL0YbxIz3Brw"/>
```

# Sample YAML Configuration File

You can use the following sample file for UDC deployment in OpenStack.

```
---

#

#  CPS system configuration

#

#  CPS configuration is a YAML file with all the configuration required

#  to bring up a new installation of CPS.

#

#  This example file lists all possible configuration fields.

#  Fields that are not marked as required can be left out of

#  the configuration. Fields that are not provided will use

#  the default value. If not default is indicated the default

#  is an empty string.


# The version of the configuration file. The installation documentation

# for the version of the CPS you are installing will indicate which

# configuration version you must use.

# REQUIRED

configVersion: 1.0
```

```
# Configuration section for CPS hosts
# REQUIRED
hosts:
  # The host section must specify all hosts that are members of the CPS
  # deployment. Host entries consist of the following REQUIRED fields
  #  name: the string to be used as a hostname for the VM
  #  alias: the string to be used in hostname lookup for the VM
  #  interfaces: Network details consisting of the following REQUIRED fields
  #    network: The network name which must match a VLAN name (see below)
  #    ipAddress: The interface address
  - name: "lb01"
    alias: "lb01"
    interfaces:
      - network: "Internal"
        ipAddress: "172.4.32.103"
      - network: "Management"
        ipAddress: "4.4.32.103"
  - name: "lb02"
    alias: "lb02"
    interfaces:
      - network: "Internal"
        ipAddress: "172.4.32.104"
      - network: "Management"
        ipAddress: "4.4.32.104"
  - name: "sessionmgr01"
    alias: "sessionmgr01"
    interfaces:
```

```
      - network: "Internal"
        ipAddress: "172.4.32.111"
      - network: "Management"
        ipAddress: "4.4.32.111"
  - name: "sessionmgr02"
    alias: "sessionmgr02"
    interfaces:
      - network: "Internal"
        ipAddress: "172.4.32.112"
      - network: "Management"
        ipAddress: "4.4.32.112"
  - name: "qns01"
    alias: "qns01"
    interfaces:
      - network: "Internal"
        ipAddress: "172.4.32.107"
  - name: "qns02"
    alias: "qns02"
    interfaces:
      - network: "Internal"
        ipAddress: "172.4.32.108"
  - name: "udc01"
    alias: "udc01"
    interfaces:
      - network: "Internal"
        ipAddress: "172.4.32.109"
      - network: "Management"
```

```
        ipAddress: "4.4.32.109"
    - name: "udc02"
      alias: "udc02"
      interfaces:
        - network: "Internal"
          ipAddress: "172.4.32.110"
        - network: "Management"
          ipAddress: "4.4.32.110"
    - name: "pcrfclient01"
      alias: "pcrfclient01"
      interfaces:
        - network: "Internal"
          ipAddress: "172.4.32.105"
        - network: "Management"
          ipAddress: "4.4.32.105"
    - name: "pcrfclient02"
      alias: "pcrfclient02"
      interfaces:
        - network: "Internal"
          ipAddress: "172.4.32.106"
        - network: "Management"
          ipAddress: "4.4.32.106"



# Configuration section for CPS VLANs
# REQUIRED
vlans:
```

```
  # VLAN entries consist of the following REQUIRED fields
  #  name: The VLAN name. This name must be used in the "network" field
  #        host interfaces (see above)
  #  vipAlias: Hostname associated with the vip
  #  vip: Virtual IP used no this network, if any.
  #  guestNic: The name of the interface specified in the host cloud config
  #            or the Heat definition.
  #
  - name: "Internal"
    vipAlias: "lbvip02"
    vip: "172.4.32.102"
  - name: "Management"
    vipAlias: "lbvip01"
    vip: "4.4.32.102"


# Configuration section for hosts not configured in the hosts section above.
# REQUIRED
additionalHosts:
  # additionalHosts entries consist of the following REQUIRED fields
  #  name: The hostname
  #  alias: The string to be used in the etc/host file.
  #  ipAddress: The IP address to use in the etc/host file.
  #
  - name: "lbvip01"
    ipAddress: "4.4.32.102"
    alias: "lbvip01"
  - name: "lbvip02"
```

```
    ipAddress: "172.4.32.102"
    alias: "lbvip02"

# Configuration section for general configuration items.
# REQUIRED
config:
  # Do not change. See install documentation for details.
  # default: sys_user_0
  qpsUser: "sys_user_0"

  # Do not change. See install documentation for details.
  # default: disabled
  selinuxState: "disabled"

  # Do not change. See install documentation for details.
  # default: targeted
  selinuxType: "targeted"

  # See install documentation for details.
  # default: broadhop
  broadhopVar: "broadhop"

  # Set true to enable TACACS+ authentication.
  # default: FALSE
  tacacsEnabled: "FALSE"

  # The IP Address of the TACACS+ server
```

```
tacacsServer: "127.0.0.1"


# The password/secret of the TACACS+ server.
tacacsSecret: "xxxxxxxxxxxxxxxxxxxxxxxxxxxx"


# A set of SNMP Network Management Stations.
# NMS can be specified as IP addresses or IP
# addresses. Entries are space separated.
# Hostnames must also be specified in Additional
# Host configuration.
# See install documentation for details.
nmsManagers:


# Low Memory alert threshold %.
# default: 0.1 (10% free)
freeMemPer: "0.1"


# A space separated set of protocol:hostname:port
# entries. UDP is the only supported protocol.
# Example:
# upd:corporate_syslog_ip:514 udp:corporate_syslog_ip2:514
syslogManagers:


# A comma separated set of port values.
# This must match values in the syslog_managers_list.
# default: 514
syslogManagersPorts: "514"
```

```
# Port value for the rsyslog proxy server to listen
# for incoming connections
# default: 6515
logbackSyslogDaemonPort: "6515"

# IP address value used in the
# /etc/broadhop/controlcenter/logback.xml
# on the pcrfclient.
# default: lbvip02
logbackSyslogDaemonAddr: "lbvip02"



# High CPU alert threshold.
# The system will alert whenever the usage is
# higher than this value.
# default: 80
cpuUsageAlertThreshold: "80"

# Clear High CPU Trap threshold.
# The system will generate a clear trap when a
# High CPU trap has been generated and the CPU
# usage is lower than this value.
# default: 40
cpuUsageClearThreshold: "40"

# The number of 5 sec intervals to wait between
```

```
# checking the CPU usage.
# default: 12 (60 seconds)
cpuUsageTrapIntervalCycle: "12"

# The SNMP trap community string.
snmpTrapCommunity: "broadhop"

#The SNMP read community string.
snmpRoCommunity: "broadhop"

#
monQnsLb:

# Enables or disables linux firewall on all VMs (IPtables).
# default: disabled
firewallState: "disabled"
puppetTimeoutInSecs: 600000

# Users
# There are different categories of users specified for the CPS.
# All users have the following fields:
#
#  name: The user name. REQUIRED
#  password: The password for the user. REQUIRED
#            The password will need to be either in cleartext or
#            encrypted. Please refer to Install documentation for details.
#  groups: The groups for the user. Groups are specified as a list
```

```
#          of group names.


# System Users
# Note that there must be a system use named sys_user_0
sysUsers:
  - name: "xxxxxxxxxxxxxxxxxxxxxxxxxxxx"
    password: "xxxxxxxxxxxxxxxxxxxxxxxxxxxx"
    groups:
      - pwauth


  - name: "xxxxxxxxxxxxxxxxxxxxxxxxxxxx"
    password: "xxxxxxxxxxxxxxxxxxxxxxxxxxxx"
  - name: "xxxxxxxxxxxxxxxxxxxxxxxxxxxx"
    password: "xxxxxxxxxxxxxxxxxxxxxxxxxxxx"
# Hypervisor Users
hvUsers:
  - name: "xxxxxxxxxxxxxxxxxxxxxxxxxxxx"
    password: "xxxxxxxxxxxxxxxxxxxxxxxxxxxx"


# Other Users for the CPS
# e.g. Control Center Users
additionalUsers:
  - name: "xxxxxxxxxxxxxxxxxxxxxxxxxxxx"
    password: "xxxxxxxxxxxxxxxxxxxxxxxxxxxx"
    groups:
      - qns
```

```
# Configuration section for feature licenses
# REQUIRED
licenses:
  # Licenses have the following required fields:
  # feature: The name of the feature license.
  # license: The license key for the feature.
  # - feature: "feature 1  Name"
  #   license: "license 1 key string"
   - feature: "MOBILE_CORE"
     license: "xxxxxxxxxxxxxxxxxxxxxxxxxxxx"
   - feature: "RADIUS_AUTH"
     license: "xxxxxxxxxxxxxxxxxxxxxxxxxxxx"


# Configuration section for mongo replica sets.
# REQUIRED
replicaSets:
  #
  # Mongo replica sets have the following REQUIRED fields
  # <Mongo Set Identifier> : The database for which the replica
  #                          set is being created.
  #   setName: The name of the replica set
  #   oplogSize: Mongo Oplog size
  #   arbiter: The Arbiter hosthame and port
  #   arbiterDataPath: The data directory on the arbiter VM
  #   members: List of members for the replica set. Each list element
# sessionType: specify as 'udc' for a UDC session replica-set
#            will be a session manager hostname:port
```

```
#   dataPath: The data directory path on the session manager VMs
- title: SESSION-SET1
  setName: set01
  oplogSize: 5120
  arbiter: pcrfclient01:27717
  arbiterDataPath: /var/data/sessions.1
  members:
    - sessionmgr01:27717
    - sessionmgr02:27717
  dataPath: /var/data/sessions.1
  hotStandBy: true
- title: BALANCE-SET1
  setName: set02
  oplogSize: 5120
  arbiter: pcrfclient01:27718
  arbiterDataPath: /var/data/sessions.2
  members:
    - sessionmgr01:27718
    - sessionmgr02:27718
  dataPath: /var/data/sessions.2
- title: REPORTING-SET1
  setName: set03
  oplogSize: 5120
  arbiter: pcrfclient01:27719
  arbiterDataPath: /var/data/sessions.3
  members:
    - sessionmgr01:27719
```

```
      - sessionmgr02:27719
    dataPath: /var/data/sessions.3
  - title: SPR-SET1
    setName: set04
    oplogSize: 3072
    arbiter: pcrfclient01:27720
    arbiterDataPath: /var/data/sessions.4
    members:
      - sessionmgr01:27720
      - sessionmgr02:27720
    dataPath: /var/data/sessions.4
  - title: AUDIT-SET1
    setName: set05
    oplogSize: 3072
    arbiter: pcrfclient01:27017
    arbiterDataPath: /var/data/sessions.5
    members:
      - sessionmgr01:27017
      - sessionmgr02:27017
    dataPath: /var/data/sessions.5
  - title: ADMIN-SET1
    setName: set06
    oplogSize: 3072
    arbiter: pcrfclient01:27721
    arbiterDataPath: /var/data/sessions.6
    members:
      - sessionmgr01:27721
```

```
    - sessionmgr02:27721
  dataPath: /var/data/sessions.6
- title: ADMIN-SET2
  setName: set07
  oplogSize: 3072
  arbiter: pcrfclient01:27731
  arbiterDataPath: /var/data/sessions.7
  members:
    - sessionmgr01:27731
    - sessionmgr02:27731
  dataPath: /var/data/sessions.7
- title: SESSION-SET2
  setName: set08
  oplogSize: 5120
  arbiter: pcrfclient01:37717
  arbiterDataPath: /var/data/sessions.1/2
  sessionType: UDC
  members:
    - sessionmgr01:37717
    - sessionmgr02:37717
  dataPath: /var/data/sessions.1/2
  seeds: "sessionmgr01:sessionmgr02:37717"
```

# Modifying Default Configurations

The following section describes how to modify the default configurations in UDC fresh installation.

## Modifying Default Cluster Name

The default cluster name that contains the UDC Admin replica-set is `cluster-udc`. The cluster name is defined in the following files:

/etc/broadhop/udc/qns.conf

/etc/broadhop/udc_diameter_endpoint/qns.conf

Perform the following steps to change the default cluster name:

**Step 5** Log in to Policy Builder.

**Step 6** Navigate to Systems > system-1 > cluster-udc

**Step 7** Enter the new cluster name

**Step 8** Publish the new confiuguration.

**Step 9** Update the following parameters in the qns.conf files to the new name of the cluster:

-Dcom.broadhop.run.clusterId=cluster-udc

## Modifying Default Port Number

The default port number for the UDC Session replica-set is defined in the following files:

/etc/broadhop/udc/qns.conf

/etc/broadhop/udc_diameter_endpoint/qns.conf as 37717.

Perform the following steps to change the default cluster name:

**Step 1** If you are using a different port number or different seeds after the Session replica-set is created, change the following parameters in the mentioned qns.conf files to the new values:

-Dsession.db.init.1=sessionmgr01

-Dsession.db.init.2=sessionmgr02

-Dsession.db.init.port=37717

After any configuration changes in `qns.conf` files, run the following commands:

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

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

restartall.sh

# Installing UDC in VMware

The procedure to install UDC in VMware is similar to PCRF Mobile installation in VMware.

For more information about installation, see *CPS Installation Guide in VMware*.

You need to modify CSV configuration and mongoConfig.cfg files to install UDC in VMware.

## Modifying CSV Configuration Files

The following section illustrates the CSV configuration files:

**Sample hosts.csv File**

| Hypervisor Name | Guest Name | Role | Alias | Datastore | Networks --> | Internal | Management |
|---|---|---|---|---|---|---|---|
| esxi-host-1 | dc1-lb01 | lb01 | lb01 | datastore1 | | XX.XX.XX.XX | XX.XX.XX.XX |
| esxi-host-1 | dc1-lb02 | lb02 | lb02 | datastore1 | | XX.XX.XX.XX | XX.XX.XX.XX |
| esxi-host-1 | dc1-qns01 | qps | qns01 | datastore1 | | XX.XX.XX.XX | |
| esxi-host-1 | dc1-qns02 | qps | qns02 | datastore1 | | XX.XX.XX.XX | |
| esxi-host-1 | dc1-udc01 | udc | udc01 | datastore1 | | XX.XX.XX.XX | |
| esxi-host-1 | dc1-udc02 | udc | udc02 | datastore1 | | XX.XX.XX.XX | |
| esxi-host-1 | dc1-sessionmgr01 | sm | sessionmgr01 | datastore1 | | XX.XX.XX.XX | |
| esxi-host-1 | dc1-sessionmgr02 | sm | sessionmgr02 | datastore1 | | XX.XX.XX.XX | |
| esxi-host-1 | dc1-pcrfclient01 | pcrfclient01 | pcrfclient01 | datastore1 | | XX.XX.XX.XX | XX.XX.XX.XX |
| esxi-host-1 | dc1-pcrfclient02 | pcrfclient02 | pcrfclient02 | datastore1 | | XX.XX.XX.XX | XX.XX.XX.XX |

**Sample VMSpecification.csv File**

| 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 |

## Modifying mongoConfig.cfg File

You can use the following mongoConfig.cfg file for UDC deployment in VMware:

```
[SESSION-SET1]

SETNAME=set01

OPLOG_SIZE=2048
```

```
ARBITER=arbitervip:27717
ARBITER_DATA_PATH=/var/data/sessions.1
MEMBER1=sessionmgr01:27717
MEMBER2=sessionmgr02:27717
DATA_PATH=/var/data/sessions.1/1
[SESSION-SET1-END]

[SESSION-SET2]
SETNAME=set07
OPLOG_SIZE=2048
ARBITER=arbitervip:37717
ARBITER_DATA_PATH=/var/data/sessions.7
MEMBER1=sessionmgr02:37717
MEMBER2=sessionmgr01:37717
SESSIONTYPE=UDC
DATA_PATH=/var/data/sessions.1/2
[SESSION-SET2-END]

[BALANCE-SET1]
SETNAME=set02
OPLOG_SIZE=5120
ARBITER=arbitervip:27718
ARBITER_DATA_PATH=/var/data/sessions.2
MEMBER1=sessionmgr01:27718
MEMBER2=sessionmgr02:27718
DATA_PATH=/var/data/sessions.2
[BALANCE-SET1-END]
```

```
[REPORTING-SET1]
SETNAME=set03
OPLOG_SIZE=5120
ARBITER=arbitervip:27719
ARBITER_DATA_PATH=/var/data/sessions.3
MEMBER1=sessionmgr01:27719
MEMBER2=sessionmgr02:27719
DATA_PATH=/var/data/sessions.3
[REPORTING-SET1-END]

[SPR-SET1]
SETNAME=set04
OPLOG_SIZE=3072
ARBITER=arbitervip:27720
ARBITER_DATA_PATH=/var/data/sessions.4
MEMBER1=sessionmgr01:27720
MEMBER2=sessionmgr02:27720
DATA_PATH=/var/data/sessions.4
[SPR-SET1-END]

[AUDIT-SET1]
SETNAME=set05
OPLOG_SIZE=3072
ARBITER=arbitervip:27017
ARBITER_DATA_PATH=/var/data/sessions.5
MEMBER1=sessionmgr01:27017
```

```
MEMBER2=sessionmgr02:27017

DATA_PATH=/var/data/sessions.5

[AUDIT-SET1-END]


[ADMIN-SET1]

SETNAME=set06

OPLOG_SIZE=3072

ARBITER=arbitervip:27721

ARBITER_DATA_PATH=/var/data/sessions.6

MEMBER1=sessionmgr01:27721

MEMBER2=sessionmgr02:27721

DATA_PATH=/var/data/sessions.6

[ADMIN-SET1-END]


[ADMIN-SET2]

SETNAME=set08

OPLOG_SIZE=3072

ARBITER=arbitervip:27731

ARBITER_DATA_PATH=/var/data/sessions.8

MEMBER1=sessionmgr01:27731

MEMBER2=sessionmgr02:27731

DATA_PATH=/var/data/sessions.8

[ADMIN-SET2-END]
```

# Configuring Diameter Endpoints

You need to configure diameter endpoints for UDC by modifying the `/var/qps/current_config/image-map` as follows:

```
lb=iomanager
```

```
lb=diameter_endpoint

lb=diameter_endpoint

lb=diameter_endpoint

lb=diameter_endpoint

lb=diameter_endpoint

lb=diameter_endpoint

lb=udc_diameter_endpoint

lb=udc_diameter_endpoint

qns=pcrf

udc=udc

pcrfclient=controlcenter

pcrfclient=pb

aio=pcrf

aio=pb
```

# Adding UDC Cluster

**NOTE**: You need to add UDC cluster in Policy Builder after Initializing SVN Synchronization. For more information, see *CPS Installation Guide for VMware*.

**Step 1** Log in to Policy Builder.

**Step 2** Navigate to Systems > system-1

**Step 3** Create a new cluster cluster-udc

**Step 4** Navigate to Admin Database

**Step 5** Enter values in the following fields:

```
Primary Database IP Address: sessionmgr01

Secondary Database IP Address: sessionmgr02

Database Port: 27731
```

**Step 6** Navigate to Systems > system-1 > cluster-1.

**Step 7** Select UDC Admin DB.

**Step 8** Enter the same values as mentioned in *Step 5*.

**Step 9** Publish the configuration.

# Creating Session Shards in UDC

For more information, see *Create Session Shards* in *CPS Installation Guide for VMWare*.

Run the following command instead of the one mentioned in *Step 1* under *Create Session Shards*:

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

# Installing License in UDC

After configuring UDC Admin DB in Policy Builder, follow the procedure mentioned in the *License Generation and Installation* topic in *CPS Installation Guide for VMWare*.

# Modifying Default Configurations

The following section describes how to modify the default configurations in UDC fresh installation.

## Modifying Default Cluster Name

The default cluster name that contains the UDC Admin replica-set is `cluster-udc`. The cluster name is defined in the following files:

```
/etc/broadhop/udc/qns.conf
```

```
/etc/broadhop/udc_diameter_endpoint/qns.conf
```

Perform the following steps to change the default cluster name:

**Step 1** Log in to Policy Builder.

**Step 2** Navigate to Systems > system-1 > cluster-udc

**Step 3** Enter the new cluster name

**Step 4** Publish the new confiuguration.

**Step 5** Update the following parameters in the qns.conf files to the new name of the cluster:

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

## Modifying Default Port Number

The default port number for the UDC Session replica-set is defined in the following files:

```
/etc/broadhop/udc/qns.conf
```

`/etc/broadhop/udc_diameter_endpoint/qns.conf as 37717.`

Perform the following steps to change the default cluster name:

**Step 1** If you are using a different port number or different seeds after the Session replica-set is created, change the following parameters in the mentioned qns.conf files to the new values:

```
-Dsession.db.init.1=sessionmgr01

-Dsession.db.init.2=sessionmgr02

-Dsession.db.init.port=37717
```

After any configuration changes in `qns.conf` files, run the following commands:

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

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

restartall.sh
```