



OpenStack Sample Files - GR

The information in the following sections is for your reference only. You need to modify them according to your requirements.



Note

Use `NM_CONTROLLED=no` parameter at the interface config file in `cloud-config` file.

For Cluman/Arbiter VM, include `ifup/ifdown` commands under `runcmd` section of cloud config file sequentially for all the interfaces to `/etc/rc.d/rc.local` to persist across boot.

- [Sample Heat Environment File, on page 1](#)
- [Sample Heat Template File, on page 3](#)
- [Sample YAML Configuration File - site1, on page 28](#)
- [Sample YAML Configuration File - site2, on page 36](#)
- [Sample Mongo Configuration File - site1, on page 43](#)
- [Sample Mongo Configuration File - site2, on page 45](#)
- [Sample Mongo GR Configuration File, on page 47](#)
- [Sample GR Cluster Configuration File - site1, on page 49](#)
- [Sample GR Cluster Configuration File - site2, on page 50](#)
- [Sample Set Priority File - site1, on page 50](#)
- [Sample Set Priority File - site2, on page 50](#)
- [Sample Shard Configuration File - site1, on page 51](#)
- [Sample Shard Configuration File - site2, on page 51](#)
- [Sample Ring Configuration File, on page 51](#)
- [Sample Geo Site Lookup Configuration File - site1, on page 51](#)
- [Sample Geo Site Lookup Configuration File - site2, on page 52](#)
- [Sample Geo-tagging Configuration File - site1, on page 52](#)
- [Sample Geo-tagging Configuration File - site2, on page 52](#)
- [Sample Monitor Database Configuration File - site1, on page 52](#)
- [Sample Monitor Database Configuration File - site2, on page 53](#)

Sample Heat Environment File

```
# This is an example environment file from os24
```

```
parameters:
  cps_iso_image_name: CPS_XXX.iso <----- where, XXX is iso build name.
  base_vm_image_name: base_vm
  cps_az_1: az-1
  cps_az_2: az-2

  internal_net_name: internal
  internal_net_cidr: 192.169.21.0/24

  management_net_name: management
  management_net_cidr: 192.169.23.0/24
  management_net_gateway: 192.169.23.1

  gx_net_name: gx
  gx_net_cidr: 192.169.22.0/24

  external_net_name: external
  external_net_cidr: 192.169.24.0/24
  external_net_gateway: 192.169.24.1

  cluman_flavor_name: cluman
  cluman_internal_ip: 192.169.21.10
  cluman_management_ip: 192.169.23.10
  cluman_external_ip: 192.169.24.10

  lb_internal_vip: 192.169.21.21
  lb_management_vip: 192.169.23.21
  lb_gx_vip: 192.169.22.21
  lb_external_vip: 192.169.24.21
  lb01_flavor_name: lb01
  lb01_internal_ip: 192.169.21.11
  lb01_management_ip: 192.169.23.11
  lb01_gx_ip: 192.169.22.11
  lb01_external_ip: 192.169.24.11
  lb02_flavor_name: lb02
  lb02_internal_ip: 192.169.21.12
  lb02_management_ip: 192.169.23.12
  lb02_gx_ip: 192.169.22.12
  lb02_external_ip: 192.169.24.12

  pcrfclient01_flavor_name: pcrfclient01
  pcrfclient01_internal_ip: 192.169.21.19
  pcrfclient01_management_ip: 192.169.23.19
  pcrfclient01_external_ip: 192.169.24.19
  pcrfclient02_flavor_name: pcrfclient02
  pcrfclient02_internal_ip: 192.169.21.20
  pcrfclient02_management_ip: 192.169.23.20
  pcrfclient02_external_ip: 192.169.24.20

  qns01_internal_ip: 192.169.21.15
  qns01_management_ip: 192.169.23.15
  qns01_external_ip: 192.169.24.15

  qns02_internal_ip: 192.169.21.16
  qns02_management_ip: 192.169.23.16
  qns02_external_ip: 192.169.24.16

  qns03_internal_ip: 192.169.21.17
  qns03_management_ip: 192.169.23.17
  qns03_external_ip: 192.169.24.17

  qns04_internal_ip: 192.169.21.18
  qns04_management_ip: 192.169.23.18
  qns04_external_ip: 192.169.24.18
```

```

sessionmgr01_internal_ip: 192.169.21.13
sessionmgr01_management_ip: 192.169.23.13
sessionmgr01_external_ip: 192.169.24.13

sessionmgr02_internal_ip: 192.169.21.14
sessionmgr02_management_ip: 192.169.23.14
sessionmgr02_external_ip: 192.169.24.14

sessionmgr03_internal_ip: 192.169.21.22
sessionmgr03_management_ip: 192.169.23.22
sessionmgr03_external_ip: 192.169.24.22

sessionmgr04_internal_ip: 192.169.21.23
sessionmgr04_management_ip: 192.169.23.23
sessionmgr04_external_ip: 192.169.24.23

svn01_volume_id: "19d61e3e-a948-46e1-aa38-d953ab98e9a3"
svn02_volume_id: "3d07bf7f-7a23-43e2-8b93-d705f3bd0619"
mongo01_volume_id: "23e10db6-0f51-463d-97b9-5b8329f30ec4"
mongo02_volume_id: "57adb91c-be6e-449e-9f31-8061df726e45"
mongo03_volume_id: "0e2ebce2-9996-4a6f-96ad-c22f3f873570"
mongo04_volume_id: "552c311a-1082-4898-bc18-2d959fbefc39"
cps_iso_volume_id: "023528a2-ac87-4f7c-b868-5ba0346c2673"

```

Sample Heat Template File



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.

```

description: A minimal CPS deployment for big bang deployment

parameters:
#####
# Global Paramaters
#####
  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|mog|pats|arbiter|andsf|escef)
    description: cps installation type (mobile|mog|arbiter|andsf|escef)
    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

#=====
# Network Paramaters
#=====
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

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: ""

gx_net_name:
    type: string
    label: gx network name
    description: name of the gx network
gx_net_cidr:
    type: string
    label: cps gx cidr
    description: cidr of gx subnet
gx_net_gateway:
    type: string
    label: gx network gateway
    description: gateway on gx network
    default: ""

external_net_name:
    type: string
    label: external network name
    description: name of the external network
external_net_cidr:
    type: string
    label: cps external cidr
    description: cidr of external subnet
external_net_gateway:
    type: string
    label: external network gateway
    description: gateway on external network
    default: ""

cps_secgroup_name:
    type: string
    label: cps secgroup name
    description: name of cps security group
    default: cps_secgroup

#=====

```

```
# Volume Paramaters
#=====
mongo01_volume_id:
  type: string
  label: mongo01 volume id
  description: uuid of the mongo01 volume

mongo02_volume_id:
  type: string
  label: mongo02 volume id
  description: uuid of the mongo02 volume

mongo03_volume_id:
  type: string
  label: mongo03 volume id
  description: uuid of the mongo03 volume

mongo04_volume_id:
  type: string
  label: mongo04 volume id
  description: uuid of the mongo04 volume

svn01_volume_id:
  type: string
  label: svn01 volume id
  description: uuid of the svn01 volume

svn02_volume_id:
  type: string
  label: svn02 volume id
  description: uuid of the svn02 volume

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
cluman_external_ip:
  type: string
  label: external ip of cluster manager
  description: external ip of cluster manager

lb_internal_vip:
  type: string
  label: internal vip of load balancer
  description: internal vip of load balancer
lb_management_vip:
  type: string
```

```

    label: management vip of load balancer
    description: management vip of load balancer
lb_gx_vip:
  type: string
  label: gx ip of load balancer
  description: gx vip of load balancer
lb_external_vip:
  type: string
  label: external ip of load balancer
  description: external vip of load balancer
lb01_flavor_name:
  type: string
  label: lb01 flavor name
  description: flavor lb01 vms will use
  default: lb01
lb01_internal_ip:
  type: string
  label: internal ip of load balancer
  description: internal ip of load balancer
lb01_management_ip:
  type: string
  label: management ip of load balancer
  description: management ip of load balancer
lb01_gx_ip:
  type: string
  label: gx ip of load balancer
  description: gx ip of load balancer
lb01_external_ip:
  type: string
  label: external ip of load balancer
  description: external ip of load balancer
lb02_flavor_name:
  type: string
  label: lb02 flavor name
  description: flavor lb02 vms will use
  default: lb02
lb02_internal_ip:
  type: string
  label: internal ip of load balancer
  description: internal ip of load balancer
lb02_management_ip:
  type: string
  label: management ip of load balancer
  description: management ip of load balancer
lb02_gx_ip:
  type: string
  label: gx ip of load balancer
  description: gx ip of load balancer
lb02_external_ip:
  type: string
  label: external ip of load balancer lb02
  description: external ip of load balancer lb02

pcrfclient01_flavor_name:
  type: string
  label: pcrfclient01 flavor name
  description: flavor pcrfclient01 vm will use
  default: pcrfclient01
pcrfclient01_internal_ip:
  type: string
  label: internal ip of pcrfclient01
  description: internal ip of pcrfclient01
pcrfclient01_management_ip:
  type: string

```

```
    label: management ip of pcrfclient01
    description: management ip of pcrfclient01
pcrfclient01_external_ip:
  type: string
  label: external ip of pcrfclient01
  description: external ip of pcrfclient01

pcrfclient02_flavor_name:
  type: string
  label: pcrfclient02 flavor name
  description: flavor pcrfclient02 vm will use
  default: pcrfclient02
pcrfclient02_internal_ip:
  type: string
  label: internal ip of pcrfclient02
  description: internal ip of pcrfclient02
pcrfclient02_management_ip:
  type: string
  label: management ip of pcrfclient02
  description: management ip of pcrfclient02
pcrfclient02_external_ip:
  type: string
  label: external ip of pcrfclient02
  description: external ip of pcrfclient02

qns_flavor_name:
  type: string
  label: qns flavor name
  description: flavor qns vms will use
  default: qns
qns01_internal_ip:
  type: string
  label: internal ip of qns01
  description: internal ip of qns01
qns01_management_ip:
  type: string
  label: management ip of qns01
  description: management ip of qns01
qns01_external_ip:
  type: string
  label: external ip of qns01
  description: external ip of qns01

qns02_internal_ip:
  type: string
  label: internal ip of qns02
  description: internal ip of qns02
qns02_management_ip:
  type: string
  label: management ip of qns02
  description: management ip of qns02
qns02_external_ip:
  type: string
  label: external ip of qns02
  description: external ip of qns02

qns03_internal_ip:
  type: string
  label: internal ip of qns03
  description: internal ip of qns03
qns03_management_ip:
  type: string
  label: management ip of qns03
```

```
description: management ip of qns03
qns03_external_ip:
  type: string
  label: external ip of qns03
  description: external ip of qns03

qns04_internal_ip:
  type: string
  label: internal ip of qns04
  description: internal ip of qns04
qns04_management_ip:
  type: string
  label: management ip of qns04
  description: management ip of qns04
qns04_external_ip:
  type: string
  label: external ip of qns04
  description: external ip of qns04

sessionmgr_flavor_name:
  type: string
  label: sessionmgr flavor name
  description: flavor sessionmgr vms will use
  default: sm
sessionmgr01_internal_ip:
  type: string
  label: internal ip of sessionmgr01
  description: internal ip of sessionmgr01
sessionmgr01_management_ip:
  type: string
  label: management ip of sessionmgr01
  description: management ip of sessionmgr01
sessionmgr01_external_ip:
  type: string
  label: external ip of sessionmgr01
  description: external ip of sessionmgr01

sessionmgr02_internal_ip:
  type: string
  label: internal ip of sessionmgr02
  description: internal ip of sessionmgr02
sessionmgr02_management_ip:
  type: string
  label: management ip of sessionmgr02
  description: management ip of sessionmgr02
sessionmgr02_external_ip:
  type: string
  label: external ip of sessionmgr02
  description: external ip of sessionmgr02

sessionmgr03_internal_ip:
  type: string
  label: internal ip of sessionmgr03
  description: external ip of sessionmgr03
sessionmgr03_management_ip:
  type: string
  label: management ip of sessionmgr03
  description: management ip of sessionmgr03
sessionmgr03_external_ip:
  type: string
  label: external ip of sessionmgr03
  description: external ip of sessionmgr03

sessionmgr04_internal_ip:
```



```

    type: string
    label: internal ip of sessionmgr04
    description: internal ip of sessionmgr04
  sessionmgr04_management_ip:
    type: string
    label: management ip of sessionmgr04
    description: management ip of sessionmgr04
  sessionmgr04_external_ip:
    type: string
    label: external ip of sessionmgr04
    description: external ip of sessionmgr04

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 }
        - port: { get_resource: cluman_external_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_external_port:
    type: OS::Neutron::Port
    properties:
      network: { get_param: external_net_name }
      fixed_ips: [{ ip_address: { get_param: cluman_external_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 }
      $gateway: { get_param: management_net_gateway }
- path: /etc/sysconfig/network-scripts/ifcfg-eth2
  permissions: "0644"
  content:
    str_replace:
      template: |
        DEVICE=eth2
        BOOTPROTO=none
        NM_CONTROLLED=no
        IPADDR=$ip
        GATEWAY=$gateway
      params:
      $ip: { get_param: cluman_external_ip }
      $gateway: { get_param: external_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
          l
          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 }
- str_replace:
  template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth2
  params:
  $cidr: { get_param: external_net_cidr }
- ifdown eth0 && ifup eth0

```

```

- ifdown eth1 && ifup eth1
- ifdown eth2 && ifup eth2
- echo HOSTNAME=cluman >> /etc/sysconfig/network
- echo cluman > /etc/hostname
- hostname cluman
- /root/.autoinstall.sh

lb01:
  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: lb01_flavor_name }
    networks:
      - port: { get_resource: lb01_internal_port }
      - port: { get_resource: lb01_management_port }
      - port: { get_resource: lb01_gx_port }
      - port: { get_resource: lb01_external_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_gx_port:
  type: OS::Neutron::Port
  properties:
    network: { get_param: gx_net_name }
    fixed_ips: [{ ip_address: { get_param: lb01_gx_ip }}]
    allowed_address_pairs:
      - ip_address: { get_param: lb_gx_vip }
lb01_external_port:
  type: OS::Neutron::Port
  properties:
    network: { get_param: external_net_name }
    fixed_ips: [{ ip_address: { get_param: lb01_external_ip }}]
    allowed_address_pairs:
      - ip_address: { get_param: lb_external_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 }
        $gateway: { get_param: management_net_gateway }
- path: /etc/sysconfig/network-scripts/ifcfg-eth2
  content:
    str_replace:
      template: |
        DEVICE=eth2
        BOOTPROTO=none
        NM_CONTROLLED=no
        IPADDR=$ip
        GATEWAY=$gateway
    params:
        $ip: { get_param: lb01_gx_ip }
        $gateway: { get_param: gx_net_gateway }
- path: /etc/sysconfig/network-scripts/ifcfg-eth3
  content:
    str_replace:
      template: |
        DEVICE=eth3
        BOOTPROTO=none
        NM_CONTROLLED=no
        IPADDR=$ip
        GATEWAY=$gateway
    params:
        $ip: { get_param: lb01_external_ip }
        $gateway: { get_param: external_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 }
- str_replace:
  template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth2
  params:
    $cidr: { get_param: gx_net_cidr }
- str_replace:
  template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth3
  params:
    $cidr: { get_param: external_net_cidr }
- ifdown eth0 && ifup eth0
- ifdown eth1 && ifup eth1
- ifdown eth2 && ifup eth2
- ifdown eth3 && ifup eth3

```

```

- echo HOSTNAME=lb01 >> /etc/sysconfig/network
- echo lb01 > /etc/hostname
- 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 }
      - port: { get_resource: lb02_gx_port }
      - port: { get_resource: lb02_external_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_gx_port:
  type: OS::Neutron::Port
  properties:
    network: { get_param: gx_net_name }
    fixed_ips: [{ ip_address: { get_param: lb02_gx_ip }}]
    allowed_address_pairs:
      - ip_address: { get_param: lb_gx_vip }
lb02_external_port:
  type: OS::Neutron::Port
  properties:
    network: { get_param: external_net_name }
    fixed_ips: [{ ip_address: { get_param: lb02_external_ip }}]
    allowed_address_pairs:
      - ip_address: { get_param: lb_external_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 }
        $gateway: { get_param: management_net_gateway }
- path: /etc/sysconfig/network-scripts/ifcfg-eth2
  content:
    str_replace:
      template: |
        DEVICE=eth2
        BOOTPROTO=none
        NM_CONTROLLED=no
        IPADDR=$ip
        GATEWAY=$gateway
      params:
        $ip: { get_param: lb02_gx_ip }
        $gateway: { get_param: gx_net_gateway }
- path: /etc/sysconfig/network-scripts/ifcfg-eth3
  content:
    str_replace:
      template: |
        DEVICE=eth3
        BOOTPROTO=none
        NM_CONTROLLED=no
        IPADDR=$ip
        GATEWAY=$gateway
      params:
        $ip: { get_param: lb02_external_ip }
        $gateway: { get_param: external_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 }
- str_replace:
  template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth2
  params:
    $cidr: { get_param: gx_net_cidr }
- str_replace:
  template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth3
  params:
    $cidr: { get_param: external_net_cidr }
- ifdown eth0 && ifup eth0
- ifdown eth1 && ifup eth1
- ifdown eth2 && ifup eth2
- ifdown eth3 && ifup eth3
- echo HOSTNAME=lb02 >> /etc/sysconfig/network
- echo lb02 > /etc/hostname
- hostname lb02

```

```

pcrfclient01:
  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: pcrfclient01_flavor_name }
    networks:
      - port: { get_resource: pcrfclient01_internal_port }
      - port: { get_resource: pcrfclient01_management_port }
      - port: { get_resource: pcrfclient01_external_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_external_port:
  type: OS::Neutron::Port
  properties:
    network: { get_param: external_net_name }
    fixed_ips: [{ ip_address: { get_param: pcrfclient01_external_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 }
        - path: /etc/sysconfig/network-scripts/ifcfg-eth2
          content:

```

```

    str_replace:
      template: |
        DEVICE=eth2
        BOOTPROTO=none
        NM_CONTROLLED=no
        IPADDR=$ip
        GATEWAY=$gateway
      params:
        $ip: { get_param: pcrfclient01_external_ip }
        $gateway: { get_param: external_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 }
    - str_replace:
      template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth2
      params:
        $cidr: { get_param: external_net_cidr }
    - ifdown eth0 && ifup eth0
    - ifdown eth1 && ifup eth1
    - ifdown eth2 && ifup eth2
    - echo HOSTNAME=pcrfclient01 >> /etc/sysconfig/network
    - echo pcrfclient01 > /etc/hostname
    - hostname pcrfclient01

pcrfclient02:
  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: pcrfclient02_flavor_name }
    networks:
      - port: { get_resource: pcrfclient02_internal_port }
      - port: { get_resource: pcrfclient02_management_port }
      - port: { get_resource: pcrfclient02_external_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_external_port:
  type: OS::Neutron::Port
  properties:
    network: { get_param: external_net_name }

```



```

    fixed_ips: [{ ip_address: { get_param: pcrfclient02_external_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 }
                $gateway: { get_param: management_net_gateway }
        - path: /etc/sysconfig/network-scripts/ifcfg-eth2
          content:
            str_replace:
              template: |
                DEVICE=eth2
                BOOTPROTO=none
                NM_CONTROLLED=no
                IPADDR=$ip
                GATEWAY=$gateway
              params:
                $ip: { get_param: pcrfclient02_external_ip }
                $gateway: { get_param: external_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 }
    - str_replace:
      template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth2
      params:
        $cidr: { get_param: external_net_cidr }
    - ifdown eth0 && ifup eth0
    - ifdown eth1 && ifup eth1
    - ifdown eth2 && ifup eth2
    - echo HOSTNAME=pcrfclient02 >> /etc/sysconfig/network

```

```

- echo pcrfclien02 > /etc/hostname
- hostname pcrfclient02

qns01:
  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: qns_flavor_name }
    networks:
      - port: { get_resource: qns01_internal_port }
      - port: { get_resource: qns01_external_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_external_port:
  type: OS::Neutron::Port
  properties:
    network: { get_param: external_net_name }
    fixed_ips: [{ ip_address: { get_param: qns01_external_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:
                $ip: { get_param: qns01_internal_ip }
        - path: /etc/sysconfig/network-scripts/ifcfg-eth1
          content:
            str_replace:
              template: |
                DEVICE=eth1
                BOOTPROTO=none
                NM_CONTROLLED=no
                IPADDR=$ip
              params:
                $ip: { get_param: qns01_external_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 }
      - str_replace:
          template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth1

```

```

        params:
          $cidr: { get_param: external_net_cidr }
        - ifdown eth0 && ifup eth0
        - ifdown eth1 && ifup eth1
        - echo HOSTNAME=qns01 >> /etc/sysconfig/network
        - echo qns01 > /etc/hostname
        - hostname qns01

qns02:
  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: qns_flavor_name }
    networks:
      - port: { get_resource: qns02_internal_port }
      - port: { get_resource: qns02_external_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_external_port:
  type: OS::Neutron::Port
  properties:
    network: { get_param: external_net_name }
    fixed_ips: [{ ip_address: { get_param: qns02_external_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:
                $ip: { get_param: qns02_internal_ip }
        - path: /etc/sysconfig/network-scripts/ifcfg-eth1
          content:
            str_replace:
              template: |
                DEVICE=eth1
                BOOTPROTO=none
                NM_CONTROLLED=no
                IPADDR=$ip
              params:
                $ip: { get_param: qns02_external_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 }
-   str_replace:
        template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth1
        params:
          $cidr: { get_param: external_net_cidr }
-   ifdown eth0 && ifup eth0
-   ifdown eth1 && ifup eth1
-   echo HOSTNAME=qns02 >> /etc/sysconfig/network
-   echo qns02 > /etc/hostname
-   hostname qns02

qns03:
  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: qns03_internal_port }
      - port: { get_resource: qns03_external_port }
    user_data_format: RAW
    user_data: { get_resource: qns03_config }
qns03_internal_port:
  type: OS::Neutron::Port
  properties:
    network: { get_param: internal_net_name }
    fixed_ips: [{ ip_address: { get_param: qns03_internal_ip }}]
qns03_external_port:
  type: OS::Neutron::Port
  properties:
    network: { get_param: external_net_name }
    fixed_ips: [{ ip_address: { get_param: qns03_external_ip }}]
qns03_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=qns03\n"
        - path: /etc/sysconfig/network-scripts/ifcfg-eth0
          content:
            str_replace:
              template: |
                DEVICE=eth0
                BOOTPROTO=none
                NM_CONTROLLED=no
                IPADDR=$ip
              params:
                $ip: { get_param: qns03_internal_ip }
        - path: /etc/sysconfig/network-scripts/ifcfg-eth1
          content:
            str_replace:
              template: |
                DEVICE=eth1
                BOOTPROTO=none
                NM_CONTROLLED=no
                IPADDR=$ip
              params:
                $ip: { get_param: qns03_external_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 }
- str_replace:
  template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth1
  params:
    $cidr: { get_param: external_net_cidr }
- ifdown eth0 && ifup eth0
- ifdown eth1 && ifup eth1
- echo HOSTNAME=qns03 >> /etc/sysconfig/network
- echo qns03 > /etc/hostname
- hostname qns03

qns04:
  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: qns04_internal_port }
      - port: { get_resource: qns04_external_port }
    user_data_format: RAW
    user_data: { get_resource: qns04_config }
qns04_internal_port:
  type: OS::Neutron::Port
  properties:
    network: { get_param: internal_net_name }
    fixed_ips: [{ ip_address: { get_param: qns04_internal_ip }}]
qns04_external_port:
  type: OS::Neutron::Port
  properties:
    network: { get_param: external_net_name }
    fixed_ips: [{ ip_address: { get_param: qns04_external_ip }}]
qns04_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=qns04\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: qns04_internal_ip }
        - path: /etc/sysconfig/network-scripts/ifcfg-eth1
          content:
            str_replace:
              template: |
                DEVICE=eth1
                BOOTPROTO=none

```

```

        NM_CONTROLLED=no
        IPADDR=$ip
        params:
          $ip: { get_param: qns04_external_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 }
      - str_replace:
        template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth1
        params:
          $cidr: { get_param: external_net_cidr }
      - ifdown eth0 && ifup eth0
      - ifdown eth1 && ifup eth1
      - echo HOSTNAME=qns04 >> /etc/sysconfig/network
      - echo qns04 > /etc/hostname
      - hostname qns04

sessionmgr01:
  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: sessionmgr_flavor_name }
    networks:
      - port: { get_resource: sessionmgr01_internal_port }
      - port: { get_resource: sessionmgr01_management_port }
      - port: { get_resource: sessionmgr01_external_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_external_port:
    type: OS::Neutron::Port
    properties:
      network: { get_param: external_net_name }
      fixed_ips: [{ ip_address: { get_param: sessionmgr01_external_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 }
        $gateway: { get_param: management_net_gateway }
- path: /etc/sysconfig/network-scripts/ifcfg-eth2
  content:
    str_replace:
      template: |
        DEVICE=eth2
        BOOTPROTO=none
        NM_CONTROLLED=no
        IPADDR=$ip
        GATEWAY=$gateway
      params:
        $ip: { get_param: sessionmgr01_external_ip }
        $gateway: { get_param: external_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 }
- str_replace:
  template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth2
  params:
    $cidr: { get_param: external_net_cidr }
- ifdown eth0 && ifup eth0
- ifdown eth1 && ifup eth1
- ifdown eth2 && ifup eth2
- echo HOSTNAME=sessionmgr01-site2 >> /etc/sysconfig/network
- echo sessionmgr01-site2 > /etc/hostname
- hostname sessionmgr01-site2

sessionmgr02:
  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: sessionmgr_flavor_name }
    networks:

```

```

    - port: { get_resource: sessionmgr02_internal_port }
    - port: { get_resource: sessionmgr02_management_port }
    - port: { get_resource: sessionmgr02_external_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_external_port:
    type: OS::Neutron::Port
    properties:
      network: { get_param: external_net_name }
      fixed_ips: [{ ip_address: { get_param: sessionmgr02_external_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 }
                $gateway: { get_param: management_net_gateway }
          - path: /etc/sysconfig/network-scripts/ifcfg-eth2
            content:
              str_replace:
                template: |
                  DEVICE=eth2
                  BOOTPROTO=none
                  NM_CONTROLLED=no
                  IPADDR=$ip
                  GATEWAY=$gateway
              params:
                $ip: { get_param: sessionmgr02_external_ip }

```



```

        $gateway: { get_param: external_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 }
  - str_replace:
    template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth2
    params:
      $cidr: { get_param: external_net_cidr }
  - ifdown eth0 && ifup eth0
  - ifdown eth1 && ifup eth1
  - ifdown eth2 && ifup eth2
  - echo HOSTNAME=sessionmgr02-site2 >> /etc/sysconfig/network
  - echo sessionmgr02-site2 > /etc/hostname
  - hostname sessionmgr02-site2

sessionmgr03:
  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: sessionmgr_flavor_name }
    networks:
      - port: { get_resource: sessionmgr03_internal_port }
      - port: { get_resource: sessionmgr03_management_port }
      - port: { get_resource: sessionmgr03_external_port }
    block_device_mapping:
      - device_name: vdb
        volume_id: { get_param: mongo03_volume_id }
    user_data_format: RAW
    user_data: { get_resource: sessionmgr03_config }
sessionmgr03_internal_port:
  type: OS::Neutron::Port
  properties:
    network: { get_param: internal_net_name }
    fixed_ips: [{ ip_address: { get_param: sessionmgr03_internal_ip }}]
sessionmgr03_management_port:
  type: OS::Neutron::Port
  properties:
    network: { get_param: management_net_name }
    fixed_ips: [{ ip_address: { get_param: sessionmgr03_management_ip }}]
sessionmgr03_external_port:
  type: OS::Neutron::Port
  properties:
    network: { get_param: external_net_name }
    fixed_ips: [{ ip_address: { get_param: sessionmgr03_external_ip }}]
sessionmgr03_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=sessionmgr03\n"

```

```

- path: /etc/sysconfig/network-scripts/ifcfg-eth0
  content:
    str_replace:
      template: |
        DEVICE=eth0
        BOOTPROTO=none
        NM_CONTROLLED=no
        IPADDR=$ip
    params:
      $ip: { get_param: sessionmgr03_internal_ip }
- path: /etc/sysconfig/network-scripts/ifcfg-eth1
  content:
    str_replace:
      template: |
        DEVICE=eth1
        BOOTPROTO=none
        NM_CONTROLLED=no
        IPADDR=$ip
        GATEWAY=$gateway
    params:
      $ip: { get_param: sessionmgr03_management_ip }
      $gateway: { get_param: management_net_gateway }
- path: /etc/sysconfig/network-scripts/ifcfg-eth2
  content:
    str_replace:
      template: |
        DEVICE=eth2
        BOOTPROTO=none
        NM_CONTROLLED=no
        IPADDR=$ip
        GATEWAY=$gateway
    params:
      $ip: { get_param: sessionmgr03_external_ip }
      $gateway: { get_param: external_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 }
- str_replace:
  template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth2
  params:
    $cidr: { get_param: external_net_cidr }
- ifdown eth0 && ifup eth0
- ifdown eth1 && ifup eth1
- ifdown eth2 && ifup eth2
- echo HOSTNAME=sessionmgr03-site2 >> /etc/sysconfig/network
- echo sessionmgr03-site2 > /etc/hostname
- hostname sessionmgr03-site2

sessionmgr04:
  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: sessionmgr_flavor_name }
networks:
  - port: { get_resource: sessionmgr04_internal_port }
  - port: { get_resource: sessionmgr04_management_port }
  - port: { get_resource: sessionmgr04_external_port }
block_device_mapping:
  - device_name: vdb
    volume_id: { get_param: mongo04_volume_id }
user_data_format: RAW
user_data: { get_resource: sessionmgr04_config }
sessionmgr04_internal_port:
  type: OS::Neutron::Port
  properties:
    network: { get_param: internal_net_name }
    fixed_ips: [{ ip_address: { get_param: sessionmgr04_internal_ip }}]
sessionmgr04_management_port:
  type: OS::Neutron::Port
  properties:
    network: { get_param: management_net_name }
    fixed_ips: [{ ip_address: { get_param: sessionmgr04_management_ip }}]
sessionmgr04_external_port:
  type: OS::Neutron::Port
  properties:
    network: { get_param: external_net_name }
    fixed_ips: [{ ip_address: { get_param: sessionmgr04_external_ip }}]
sessionmgr04_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=sessionmgr04\n"
        - path: /etc/sysconfig/network-scripts/ifcfg-eth0
          content:
            str_replace:
              template: |
                DEVICE=eth0
                BOOTPROTO=none
                NM_CONTROLLED=no
                IPADDR=$ip
              params:
                $ip: { get_param: sessionmgr04_internal_ip }
        - path: /etc/sysconfig/network-scripts/ifcfg-eth1
          content:
            str_replace:
              template: |
                DEVICE=eth1
                BOOTPROTO=none
                NM_CONTROLLED=no
                IPADDR=$ip
                GATEWAY=$gateway
              params:
                $ip: { get_param: sessionmgr04_management_ip }
                $gateway: { get_param: management_net_gateway }
        - path: /etc/sysconfig/network-scripts/ifcfg-eth2
          content:
            str_replace:
              template: |
                DEVICE=eth2
                BOOTPROTO=none
                NM_CONTROLLED=no
                IPADDR=$ip
                GATEWAY=$gateway

```

```

    params:
      $ip: { get_param: sessionmgr04_external_ip }
      $gateway: { get_param: external_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 }
    - str_replace:
      template: ipcalc -m $cidr >> /etc/sysconfig/network-scripts/ifcfg-eth2
      params:
        $cidr: { get_param: external_net_cidr }
    - ifdown eth0 && ifup eth0
    - ifdown eth1 && ifup eth1
    - ifdown eth2 && ifup eth2
    - echo HOSTNAME=sessionmgr04-site2 >> /etc/sysconfig/network
    - echo sessionmgr04-site2 > /etc/hostname
    - hostname sessionmgr04-site2

```

Sample YAML Configuration File - site1



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.

```

---
#
# 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
# Order of interfaces should be same as your cloud-config.
# For example, Internal > eth0; Management > eth1; Gx > eth2; External > eth3
- name: "lb01"
  alias: "lb01"
  interfaces:
    - network: "Internal"
      ipAddress: "192.169.21.11"
    - network: "Management"
      ipAddress: "192.169.23.11"
    - network: "Gx"
      ipAddress: "192.169.22.11"
    - network: "External"
      ipAddress: "192.169.24.11"
- name: "lb02"
  alias: "lb02"
  interfaces:
    - network: "Internal"
      ipAddress: "192.169.21.12"
    - network: "Management"
      ipAddress: "192.169.23.12"
    - network: "Gx"
      ipAddress: "192.169.22.12"
    - network: "External"
      ipAddress: "192.169.24.12"
- name: "sessionmgr01-site1"
  alias: "sessionmgr01"
  interfaces:
    - network: "Internal"
      ipAddress: "192.169.21.13"
    - network: "Management"
      ipAddress: "192.169.23.13"
    - network: "External"
      ipAddress: "192.169.24.13"
- name: "sessionmgr02-site1"
  alias: "sessionmgr02"
  interfaces:
    - network: "Internal"
      ipAddress: "192.169.21.14"
    - network: "Management"
      ipAddress: "192.169.23.14"
    - network: "External"
      ipAddress: "192.169.24.14"
- name: "sessionmgr03-site1"
  alias: "sessionmgr03"
  interfaces:
    - network: "Internal"
      ipAddress: "192.169.21.22"
    - network: "Management"
      ipAddress: "192.169.23.22"
    - network: "External"
      ipAddress: "192.169.24.22"
- name: "sessionmgr04-site1"
  alias: "sessionmgr04"
  interfaces:
    - network: "Internal"
      ipAddress: "192.169.21.23"
    - network: "Management"
      ipAddress: "192.169.23.23"
    - network: "External"
      ipAddress: "192.169.24.23"
```

```

- name: "qns01"
  alias: "qns01"
  interfaces:
    - network: "Internal"
      ipAddress: "192.169.21.15"
    - network: "External"
      ipAddress: "192.169.24.15"
- name: "qns02"
  alias: "qns02"
  interfaces:
    - network: "Internal"
      ipAddress: "192.169.21.16"
    - network: "External"
      ipAddress: "192.169.24.16"
- name: "qns03"
  alias: "qns03"
  interfaces:
    - network: "Internal"
      ipAddress: "192.169.21.17"
    - network: "External"
      ipAddress: "192.169.24.17"
- name: "qns04"
  alias: "qns04"
  interfaces:
    - network: "Internal"
      ipAddress: "192.169.21.18"
    - network: "External"
      ipAddress: "192.169.24.18"
- name: "pcrfclient01"
  alias: "pcrfclient01"
  interfaces:
    - network: "Internal"
      ipAddress: "192.169.21.19"
    - network: "Management"
      ipAddress: "192.169.23.19"
    - network: "External"
      ipAddress: "192.169.24.19"
- name: "pcrfclient02"
  alias: "pcrfclient02"
  interfaces:
    - network: "Internal"
      ipAddress: "192.169.21.20"
    - network: "Management"
      ipAddress: "192.169.23.20"
    - network: "External"
      ipAddress: "192.169.24.20"

# 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 on 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: "192.169.21.21"
- name: "Management"
  vipAlias: "lbvip01"
  vip: "192.169.23.21"

```

```
- name: "Gx"
  vipAlias: "gxvip"
  vip: "192.169.22.21"
- name: "External"
  vipAlias: "exvip"
  vip: "192.169.24.21"

# 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: "192.169.23.21"
  alias: "lbvip01"
- name: "lbvip02"
  ipAddress: "192.169.21.21"
  alias: "lbvip02"
- name: "diam-int1-vip"
  ipAddress: "192.169.22.21"
  alias: "gxvip"
- name: "arbitervip"
  ipAddress: "192.169.21.40"
  alias: "arbitervip"
- name: "cluman-site2"
  alias: "cluman-site2"
  ipAddress: "192.169.24.50"
- name: "sessionmgr01-site2"
  alias: "psessionmgr01"
  ipAddress: "192.169.24.60"
- name: "sessionmgr02-site2"
  alias: "psessionmgr02"
  ipAddress: "192.169.24.61"
- name: "sessionmgr03-site2"
  alias: "psessionmgr03"
  ipAddress: "192.169.24.66"
- name: "sessionmgr04-site2"
  alias: "psessionmgr04"
  ipAddress: "192.169.24.67"
- name: "arbiter"
  alias: "arbiter-site3"
  ipAddress: "192.169.24.90"

# 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: "CPE1704TKS"

# 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"

# 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: "qns"
    password:
"$6$HtEnOu7S$8kkHDFJtAZtJXnhRPrPFI8KAlHFch41OJ405OnCCqO0CFuRmexvCRTkCIC3QW5hkd6P/S13OD8qFHn1aYHxcel"

    groups:
      - pwauth

      - name: "qns-svn"
        password:
"$6$HtEnOu7S$8kkHDFJtAZtJXnhRPrPFI8KAlHFch41OJ405OnCCqO0CFuRmexvCRTkCIC3QW5hkd6P/S13OD8qFHn1aYHxcel"

      - name: "qns-ro"
        password:
"$6$HtEnOu7S$8kkHDFJtAZtJXnhRPrPFI8KAlHFch41OJ405OnCCqO0CFuRmexvCRTkCIC3QW5hkd6P/S13OD8qFHn1aYHxcel"

# Hypervisor Users
hvUsers:
  - name: "root"
    password: "CpS!^246"

# Other Users for the CPS
# e.g. Control Center Users
additionalUsers:
  - name: "admin"
    password: "qns123"
    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:
"25D220C6817CD63603D72ED51C811F9B7CB093A53B5CB6FB04FF6C5C6A21ED1962F0491D4EED4441D826F1BC110B05EE35B78CF43B8B8B7A8127B4545538E365"

- feature: "RADIUS_AUTH"
  license:
"118D767CE11EC2B1E3AAA846A916FA57CB093A53B5CB6FB04FF6C5C6A21ED1962F0491D4EED4441D826F1BC110B05EE35B78CF43B8B8B7A8127B4545538E365"

# 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
#   arbiters: The Arbiters hostnames and ports
#   arbiterDataPath: The data directory on the arbiter VM
#   primaryMembers: List of primaryMembers for the replica set. Each list element
#                   will be a session manager hostname:port
#   dataPath: The data directory path on the session manager VMs
- title: SESSION-SET1
  setName: set01
  oplogSize: 1024
  arbiters:
    - "arbiter-site3:27717"
  arbiterDataPath: "/var/data/sessions.1"
  siteId: "SITE1"
  members:
    - "sessionmgr02-site1:27717"
    - "sessionmgr01-site1:27717"
  dataPath: "/var/data/sessions.1/set01"
  primaryMembersTag: "SITE1"
  secondaryMembersTag: "SITE2"
  shardCount: "4"
  hotStandBy: "false"
  seeds: "sessionmgr01:sessionmgr02:27717"
- title: SESSION-SET2
  setName: set07
  oplogSize: 1024
  arbiters:
    - "arbiter-site3:27722"
  arbiterDataPath: "/var/data/sessions.7"
  siteId: "SITE1"
  members:
    - "sessionmgr03-site1:27722"
    - "sessionmgr04-site1:27722"
  dataPath: "/var/data/sessions.7"
  primaryMembersTag: "SITE1"
  secondaryMembersTag: "SITE2"
  shardCount: "4"
  hotStandBy: "true"
  seeds: "sessionmgr03:sessionmgr04:27722"
- title: BALANCE-SET1
  setName: set02
  oplogSize: 1024
  arbiters:
    - "arbiter-site3:27718"
  arbiterDataPath: "/var/data/sessions.2"
  siteId: "SITE1"
  members:

```

```

    - "sessionmgr01-sitel:27718"
    - "sessionmgr02-sitel:27718"
  dataPath: "/var/data/sessions.2"
  primaryMembersTag: "SITE1"
  secondaryMembersTag: "SITE2"
- title: REPORTING-SET1
  setName: set03
  oplogSize: 1024
  arbiters:
    - "arbiter-site3:27719"
  arbiterDataPath: "/var/data/sessions.3"
  siteId: "SITE1"
  members:
    - "sessionmgr03-sitel:27719"
    - "sessionmgr04-sitel:27719"
  dataPath: "/var/data/sessions.3"
- title: SPR-SET1
  setName: set04
  oplogSize: 1024
  arbiters:
    - "arbiter-site3:27720"
  arbiterDataPath: "/var/data/sessions.4"
  siteId: "SITE1"
  members:
    - "sessionmgr01-sitel:27720"
    - "sessionmgr02-sitel:27720"
  dataPath: "/var/data/sessions.4"
  primaryMembersTag: "SITE1"
  secondaryMembersTag: "SITE2"
- title: AUDIT-SET1
  setName: set05
  oplogSize: 1024
  arbiters:
    - "arbiter-site3:27017"
  arbiterDataPath: "/var/data/sessions.5"
  siteId: "SITE1"
  members:
    - "sessionmgr03-sitel:27017"
    - "sessionmgr04-sitel:27017"
  dataPath: "/var/data/sessions.5"
- title: ADMIN-SET1
  setName: set06
  oplogSize: 1024
  arbiters:
    - "arbiter-site3:27721"
  arbiterDataPath: "/var/data/sessions.6"
  siteId: "SITE1"
  members:
    - "sessionmgr01-sitel:27721"
    - "sessionmgr02-sitel:27721"
  dataPath: /var/data/sessions.6
applicationConfig:
  policyServerConfig:
    geoSiteName: "SITE1"
    clusterId: "Cluster-SITE1"
    siteId: "SITE1"
    remoteSiteId: "SITE2"
    heartBeatMonitorThreadSleepMS: "500"
    mongodbupdaterConnectTimeoutMS: "1000"
    mongodbupdaterSocketTimeoutMS: "1000"
    dbConnectTimeout: "1200"
    threadMaxWaitTime: "1200"
    dbSocketTimeout: "600"
    remoteLockingOff: ""

```

```

    apirouterContextPath: ""
    uaContextPath: ""
    balanceDbs: ""
    clusterPeers: ""
    isGeoHaEnabled: "true"
    geoHaSessionLookupType: "realm"
    enableReloadDict: "true"
    sprLocalGeoSiteTag: "SITE1"
    balanceLocalGeoSiteTag: "SITE1"
    sessionLocalGeoSiteTag: "SITE1"
    deploymentType: "GR"

```

Sample YAML Configuration File - site2



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.

```

---
#
# 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
  # Order of interfaces should be same as your cloud-config.
  # For example, Internal > eth0; Management > eth1; Gx > eth2; External > eth3
  - name: "lb01"
    alias: "lb01"
    interfaces:
      - network: "Internal"
        ipAddress: "192.169.21.52"
      - network: "Management"
        ipAddress: "192.169.23.52"
      - network: "Gx"
        ipAddress: "192.169.22.52"
      - network: "External"

```

```
        ipAddress: "192.169.24.52"
- name: "lb02"
  alias: "lb02"
  interfaces:
    - network: "Internal"
      ipAddress: "192.169.21.53"
    - network: "Management"
      ipAddress: "192.169.23.53"
    - network: "Gx"
      ipAddress: "192.169.22.53"
    - network: "External"
      ipAddress: "192.169.24.53"
- name: "sessionmgr01-site2"
  alias: "sessionmgr01"
  interfaces:
    - network: "Internal"
      ipAddress: "192.169.21.60"
    - network: "Management"
      ipAddress: "192.169.23.60"
    - network: "External"
      ipAddress: "192.169.24.60"
- name: "sessionmgr02-site2"
  alias: "sessionmgr02"
  interfaces:
    - network: "Internal"
      ipAddress: "192.169.21.61"
    - network: "Management"
      ipAddress: "192.169.23.61"
    - network: "External"
      ipAddress: "192.169.24.61"
- name: "sessionmgr03-site2"
  alias: "sessionmgr03"
  interfaces:
    - network: "Internal"
      ipAddress: "192.169.21.66"
    - network: "Management"
      ipAddress: "192.169.23.66"
    - network: "External"
      ipAddress: "192.169.24.66"
- name: "sessionmgr04-site2"
  alias: "sessionmgr04"
  interfaces:
    - network: "Internal"
      ipAddress: "192.169.21.67"
    - network: "Management"
      ipAddress: "192.169.23.67"
    - network: "External"
      ipAddress: "192.169.24.67"
- name: "qns01"
  alias: "qns01"
  interfaces:
    - network: "Internal"
      ipAddress: "192.169.21.56"
    - network: "External"
      ipAddress: "192.169.24.56"
- name: "qns02"
  alias: "qns02"
  interfaces:
    - network: "Internal"
      ipAddress: "192.169.21.57"
    - network: "External"
      ipAddress: "192.169.24.57"
- name: "qns03"
  alias: "qns03"
```

```

    interfaces:
      - network: "Internal"
        ipAddress: "192.169.21.58"
      - network: "External"
        ipAddress: "192.169.24.58"
  - name: "qns04"
    alias: "qns04"
    interfaces:
      - network: "Internal"
        ipAddress: "192.169.21.59"
      - network: "External"
        ipAddress: "192.169.24.59"
  - name: "pcrfclient01"
    alias: "pcrfclient01"
    interfaces:
      - network: "Internal"
        ipAddress: "192.169.21.54"
      - network: "Management"
        ipAddress: "192.169.23.54"
      - network: "External"
        ipAddress: "192.169.24.54"
  - name: "pcrfclient02"
    alias: "pcrfclient02"
    interfaces:
      - network: "Internal"
        ipAddress: "192.169.21.55"
      - network: "Management"
        ipAddress: "192.169.23.55"
      - network: "External"
        ipAddress: "192.169.24.55"

# 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 on 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: "192.169.21.51"
  - name: "Management"
    vipAlias: "lbvip01"
    vip: "192.169.23.51"
  - name: "Gx"
    vipAlias: "gxvip"
    vip: "192.169.22.51"
  - name: "External"
    vipAlias: "exvip"
    vip: "192.169.24.51"

# 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: "192.169.23.51"
    alias: "lbvip01"
- name: "lbvip02"
  ipAddress: "192.169.21.51"
  alias: "lbvip02"
- name: "diam-int1-vip"
  ipAddress: "192.169.22.51"
  alias: "gxvip"
- name: "arbitervip"
  ipAddress: "192.169.21.70"
  alias: "arbitervip"
- name: "cluman-site2"
  alias: "cluman-site2"
  ipAddress: "192.169.24.50"
- name: "sessionmgr01-site1"
  alias: "pessionmgr01"
  ipAddress: "192.169.24.13"
- name: "sessionmgr02-site1"
  alias: "pessionmgr02"
  ipAddress: "192.169.24.14"
- name: "sessionmgr03-site1"
  alias: "pessionmgr03"
  ipAddress: "192.169.24.22"
- name: "sessionmgr04-site1"
  alias: "pessionmgr04"
  ipAddress: "192.169.24.23"
- name: "arbiter"
  alias: "arbiter-site3"
  ipAddress: "192.169.24.90"
# 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: "CPE1704TKS"

  # 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"

# 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: "qns"
    password:
"$6$HtEnOu7S$8kkHDFJtAZtJXnhRPrPFI8KALHFch410J405OnCCqO0CFuRmexvCRTkCIC3QW5hkd6P/S13OD8qFHn1aYHxcel"

    groups:
      - pwauth

  - name: "qns-svn"
    password:
"$6$HtEnOu7S$8kkHDFJtAZtJXnhRPrPFI8KALHFch410J405OnCCqO0CFuRmexvCRTkCIC3QW5hkd6P/S13OD8qFHn1aYHxcel"

  - name: "qns-ro"
    password:
"$6$HtEnOu7S$8kkHDFJtAZtJXnhRPrPFI8KALHFch410J405OnCCqO0CFuRmexvCRTkCIC3QW5hkd6P/S13OD8qFHn1aYHxcel"

# Hypervisor Users
hvUsers:
  - name: "root"
    password: "CpS!^246"

# Other Users for the CPS
# e.g. Control Center Users
additionalUsers:
  - name: "admin"
    password: "qns123"
    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:
"$25D220C6817CD63603D72ED51C811F9B7CB093A53B5CB6FB04FF6C5C6A21ED1962F0491D4EED4441D826F1BC110B05EE35B78CF43B888B7A8127B4545538E365"

  - feature: "RADIUS_AUTH"
    license:
"$118D767CE11FC2CBE3AAA846A916FA57CB093A53B5CB6FB04FF6C5C6A21ED1962F0491D4EED4441D826F1BC110B05EE35B78CF43B888B7A8127B4545538E365"

# 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
#   arbiters: The Arbiter hostnames and ports
#   arbiterDataPath: The data directory on the arbiter VM
#   members: List of members for the replica set. Each list element
#             will be a session manager hostname:port
#   dataPath: The data directory path on the session manager VMs
- title: SESSION-SET63
  setName: set63
  oplogSize: 1024
  arbiters:
    - "arbiter-site3:27763"
  arbiterDataPath: "/var/data/sessions.1/set63"
  siteId: "SITE2"
  members:
    - "sessionmgr01-site2:27763"
    - "sessionmgr02-site2:27763"
  dataPath: /var/data/sessions.63
  primaryMembersTag: "SITE2"
  secondaryMembersTag: "SITE1"
  shardCount: "4"
  seeds: "sessionmgr01:sessionmgr02:27763"
- title: SESSION-SET68
  setName: set68
  oplogSize: 1024
  arbiters:
    - "arbiter-site3:27768"
  arbiterDataPath: "/var/data/sessions.68"
  siteId: "SITE2"
  members:
    - "sessionmgr03-site2:27768"
    - "sessionmgr04-site2:27768"
  dataPath: "/var/data/sessions.68:"
  primaryMembersTag: "SITE2"
  secondaryMembersTag: "SITE1"
  shardCount: "4"
  hotStandBy: "true"
  seeds: "sessionmgr03:sessionmgr04:27768"
- title: BALANCE-SET64
  setName: set64
  oplogSize: 1024
  arbiters:
    - "arbiter-site3:27764"
  arbiterDataPath: "/var/data/sessions.64"
  siteId: "SITE2"
  members:
    - "sessionmgr01-site2:27764"
    - "sessionmgr02-site2:27764"
  dataPath: "/var/data/sessions.64"
  primaryMembersTag: "SITE2"
  secondaryMembersTag: "SITE1"
- title: REPORTING-SET66
  setName: set66
  oplogSize: 1024
  arbiters:
    - "arbiter-site3:27766"
  arbiterDataPath: "/var/data/sessions.66"
  siteId: "SITE2"
  members:
    - "sessionmgr03-site2:27719"
    - "sessionmgr04-site2:27719"
  dataPath: "/var/data/sessions.66"
- title: SPR-SET67
  setName: set67
  oplogSize: 1024

```

```

    arbiters:
      - "arbiter-site3:27767"
    arbiterDataPath: "/var/data/sessions.67"
    siteId: "SITE2"
    members:
      - "sessionmgr01-site2:27767"
      - "sessionmgr02-site2:27767"
    dataPath: "/var/data/sessions.67"
    primaryMembersTag: "SITE2"
    secondaryMembersTag: "SITE1"
  - title: AUDIT-SET65
    setName: set65
    oplogSize: 1024
    arbiters:
      - "arbiter-site3:27765"
    arbiterDataPath: "/var/data/sessions.65"
    siteId: "SITE2"
    members:
      - "sessionmgr03-site2:27017"
      - "sessionmgr04-site2:27017"
    dataPath: "/var/data/sessions.65"
  - title: ADMIN-SET2
    setName: set69
    oplogSize: 1024
    arbiters:
      - "arbiter-site3:27769"
    arbiterDataPath: "/var/data/sessions.69"
    siteId: "SITE2"
    members:
      - "sessionmgr01-site2:27769"
      - "sessionmgr02-site2:27769"
    dataPath: "/var/data/sessions.69"

applicationConfig:
  policyServerConfig:
    geoSiteName: "SITE2"
    clusterId: "Cluster-SITE2"
    siteId: "SITE2"
    remoteSiteId: "SITE1"
    heartBeatMonitorThreadSleepMS: "500"
    mongodbupdaterConnectTimeoutMS: "1000"
    mongodbupdaterSocketTimeoutMS: "1000"
    dbConnectTimeout: "1200"
    threadMaxWaitTime: "1200"
    dbSocketTimeout: "600"
    remoteLockingOff: ""
    apirouterContextPath: ""
    uaContextPath: ""
    balanceDbs: ""
    clusterPeers: ""
    isGeoHaEnabled: "true"
    geoHaSessionLookupType: "realm"
    enableReloadDict: "true"
    sprLocalGeoSiteTag: "SITE2"
    balanceLocalGeoSiteTag: "SITE2"
    sessionLocalGeoSiteTag: "SITE2"
    deploymentType: "GR"

```

Sample Mongo Configuration File - site1

```

---
- title: "SESSION-SET1"

```

```

setName: "set01"
oplogSize: "1024"
arbiters:
  - "arbiter-site3:27717"
arbiterDataPath: "/var/data/sessions.1"
primaryMembers:
  - "sessionmgr02-site1:27717"
  - "sessionmgr01-site1:27717"
secondaryMembers:
  - "sessionmgr02-site2:27717"
  - "sessionmgr01-site2:27717"
dataPath: "/var/data/sessions.1/set01"
hotStandBy: "false"
shardCount: "4"
seeds: "sessionmgr01:sessionmgr02:27717"
primaryMembersTag: "SITE1"
secondaryMembersTag: "SITE2"
siteId: "SITE1"
- title: "SESSION-SET2"
  setName: "set07"
  oplogSize: "1024"
  arbiters:
    - "arbiter-site3:27722"
  arbiterDataPath: "/var/data/sessions.7"
  members:
    - "sessionmgr03-site1:27722"
    - "sessionmgr04-site1:27722"
  dataPath: "/var/data/sessions.7"
  hotStandBy: "true"
  shardCount: "4"
  seeds: "sessionmgr03:sessionmgr04:27722"
  primaryMembersTag: "SITE1"
  secondaryMembersTag: "SITE2"
  siteId: "SITE1"
- title: "BALANCE-SET1"
  setName: "set02"
  oplogSize: "1024"
  arbiters:
    - "arbiter-site3:27718"
  arbiterDataPath: "/var/data/sessions.2"
  primaryMembers:
    - "sessionmgr01-site1:27718"
    - "sessionmgr02-site1:27718"
  secondaryMembers:
    - "sessionmgr01-site2:27718"
    - "sessionmgr02-site2:27718"
  dataPath: "/var/data/sessions.2"
  primaryMembersTag: "SITE1"
  secondaryMembersTag: "SITE2"
  siteId: "SITE1"
- title: "REPORTING-SET1"
  setName: "set03"
  oplogSize: "1024"
  arbiters:
    - "arbiter-site3:27719"
  arbiterDataPath: "/var/data/sessions.3"
  members:
    - "sessionmgr03-site1:27719"
    - "sessionmgr04-site1:27719"
  dataPath: "/var/data/sessions.3"
  siteId: "SITE1"
- title: "SPR-SET1"
  setName: "set04"
  oplogSize: "1024"

```

```

    arbiters:
      - "arbiter-site3:27720"
    arbiterDataPath: "/var/data/sessions.4"
    primaryMembers:
      - "sessionmgr01-site1:27720"
      - "sessionmgr02-site1:27720"
    secondaryMembers:
      - "sessionmgr01-site2:27720"
      - "sessionmgr02-site2:27720"
    dataPath: "/var/data/sessions.4"
    primaryMembersTag: "SITE1"
    secondaryMembersTag: "SITE2"
    siteId: "SITE1"
- title: "AUDIT-SET1"
  setName: "set05"
  oplogSize: "1024"
  arbiters:
    - "arbiter-site3:27017"
  arbiterDataPath: "/var/data/sessions.5"
  members:
    - "sessionmgr03-site1:27017"
    - "sessionmgr04-site1:27017"
  dataPath: "/var/data/sessions.5"
  siteId: "SITE1"
- title: "ADMIN-SET1"
  setName: "set06"
  oplogSize: "1024"
  arbiters:
    - "arbiter-site3:27721"
  arbiterDataPath: "/var/data/sessions.6"
  primaryMembers:
    - "sessionmgr01-site1:27721"
    - "sessionmgr02-site1:27721"
  secondaryMembers:
    - "sessionmgr01-site2:27721"
    - "sessionmgr02-site2:27721"
  dataPath: "/var/data/sessions.6"
  siteId: "SITE1"

```

Sample Mongo Configuration File - site2

```

- title: "SESSION-SET63"
  setName: "set63"
  oplogSize: "1024"
  arbiters:
    - "arbiter:27763"
  arbiterDataPath: "/var/data/sessions.63"
  primaryMembers:
    - "sessionmgr01-site2:27763"
    - "sessionmgr02-site2:27763"
  secondaryMembers:
    - "sessionmgr01-site1:27763"
    - "sessionmgr02-site1:27763"
  dataPath: "/var/data/sessions.1/set63"
  secondaryMembersTag: "SITE1"
  primaryMembersTag: "SITE2"
  siteId: "SITE2"
  shardCount: "4"
  seeds: "sessionmgr01:sessionmgr02:27763"
- title: "SESSION-SET68"
  setName: "set68"
  oplogSize: "1024"

```

```

    arbiters:
      - "arbiter:27768"
    arbiterDataPath: "/var/data/sessions.68"
    primaryMembers:
      - "sessionmgr03-site2:27768"
      - "sessionmgr04-site2:27768"
    secondaryMembers:
      - "sessionmgr03-site1:27768"
      - "sessionmgr04-site1:27768"
    dataPath: "/var/data/sessions.68"
    primaryMembersTag: "SITE2"
    secondaryMembersTag: "SITE1"
    hotStandBy: "true"
    shardCount: "4"
    seeds: "sessionmgr03:sessionmgr04:27768"
    siteId: "SITE2"
- title: "BALANCE-SET64"
  setName: "set64"
  oplogSize: "1024"
  arbiters:
    - "arbiter:27764"
  arbiterDataPath: "/var/data/sessions.64"
  primaryMembers:
    - "sessionmgr03-site2:27764"
    - "sessionmgr04-site2:27764"
  secondaryMembers:
    - "sessionmgr03-site1:27764"
    - "sessionmgr04-site1:27764"
  dataPath: "/var/data/sessions.64"
  primaryMembersTag: "SITE2"
  secondaryMembersTag: "SITE1"
  siteId: "SITE2"
- title: "REPORTING-SET66"
  setName: "set66"
  oplogSize: "1024"
  arbiters:
    - "arbiter:27766"
  arbiterDataPath: "/var/data/sessions.66"
  members:
    - "sessionmgr03-site2:27766"
    - "sessionmgr04-site2:27766"
  dataPath: "/var/data/sessions.66"
  siteId: "SITE2"
- title: "SPR-SET67"
  setName: "set67"
  oplogSize: "1024"
  arbiters:
    - "arbiter:27767"
  arbiterDataPath: "/var/data/sessions.67"
  primaryMembers:
    - "sessionmgr01-site2:27767"
    - "sessionmgr02-site2:27767"
  secondaryMembers:
    - "sessionmgr01-site1:27767"
    - "sessionmgr02-site1:27767"
  dataPath: "/var/data/sessions.67"
  primaryMembersTag: "SITE2"
  secondaryMembersTag: "SITE1"
  siteId: "SITE2"
- title: "AUDIT-SET65"
  setName: "set65"
  oplogSize: "1024"
  arbiters:
    - "arbiter:37017"

```

```

    arbiterDataPath: "/var/data/sessions.65"
    members:
    - "sessionmgr03-site2:37017"
    - "sessionmgr04-site2:37017"
    dataPath: "/var/data/sessions.65"
    siteId: "SITE2"
- title: "ADMIN-SET2"
  setName: "set69"
  oplogSize: "1024"
  arbiters:
    - "arbiter:27769"
  arbiterDataPath: "/var/data/sessions.69"
  primaryMembers:
    - "sessionmgr01-site2:27769"
    - "sessionmgr02-site2:27769"
  secondaryMembers:
    - "sessionmgr01-site1:27769"
    - "sessionmgr02-site1:27769"
  dataPath: "/var/data/sessions.69"
  siteId: "SITE2"

```

Sample Mongo GR Configuration File

```

---
- title: "SESSION-SET1"
  setName: "set01"
  oplogSize: "1024"
  arbiters:
    - "arbiter-site3:27717"
  arbiterDataPath: "/var/data/sessions.1"
  primaryMembers:
    - "sessionmgr02-site1:27717"
    - "sessionmgr01-site1:27717"
  secondaryMembers:
    - "sessionmgr02-site2:27717"
    - "sessionmgr01-site2:27717"
  dataPath: "/var/data/sessions.1/set01"
  hotStandBy: "false"
  shardCount: "4"
  seeds: "sessionmgr01:sessionmgr02:27717"
  primaryMembersTag: "SITE1"
  secondaryMembersTag: "SITE2"
  siteId: "SITE1"
- title: "SESSION-SET2"
  setName: "set07"
  oplogSize: "1024"
  arbiters:
    - "arbiter-site3:27722"
  arbiterDataPath: "/var/data/sessions.7"
  members:
    - "sessionmgr03-site1:27722"
    - "sessionmgr04-site1:27722"
  dataPath: "/var/data/sessions.7"
  hotStandBy: "true"
  shardCount: "4"
  seeds: "sessionmgr03:sessionmgr04:27722"
  primaryMembersTag: "SITE1"
  secondaryMembersTag: "SITE2"
  siteId: "SITE1"
- title: "BALANCE-SET1"
  setName: "set02"
  oplogSize: "1024"

```

```

    arbiters:
    - "arbiter-site3:27718"
    arbiterDataPath: "/var/data/sessions.2"
    primaryMembers:
    - "sessionmgr01-site1:27718"
    - "sessionmgr02-site1:27718"
    secondaryMembers:
    - "sessionmgr01-site2:27718"
    - "sessionmgr02-site2:27718"
    dataPath: "/var/data/sessions.2"
    primaryMembersTag: "SITE1"
    secondaryMembersTag: "SITE2"
    siteId: "SITE1"
- title: "REPORTING-SET1"
  setName: "set03"
  oplogSize: "1024"
  arbiters:
  - "arbiter-site3:27719"
  arbiterDataPath: "/var/data/sessions.3"
  members:
  - "sessionmgr03-site1:27719"
  - "sessionmgr04-site1:27719"
  dataPath: "/var/data/sessions.3"
  siteId: "SITE1"
- title: "SPR-SET1"
  setName: "set04"
  oplogSize: "1024"
  arbiters:
  - "arbiter-site3:27720"
  arbiterDataPath: "/var/data/sessions.4"
  primaryMembers:
  - "sessionmgr01-site1:27720"
  - "sessionmgr02-site1:27720"
  secondaryMembers:
  - "sessionmgr01-site2:27720"
  - "sessionmgr02-site2:27720"
  dataPath: "/var/data/sessions.4"
  primaryMembersTag: "SITE1"
  secondaryMembersTag: "SITE2"
  siteId: "SITE1"
- title: "AUDIT-SET1"
  setName: "set05"
  oplogSize: "1024"
  arbiters:
  - "arbiter-site3:27017"
  arbiterDataPath: "/var/data/sessions.5"
  members:
  - "sessionmgr03-site1:27017"
  - "sessionmgr04-site1:27017"
  dataPath: "/var/data/sessions.5"
  siteId: "SITE1"
- title: "ADMIN-SET1"
  setName: "set06"
  oplogSize: "1024"
  arbiters:
  - "arbiter-site3:27721"
  arbiterDataPath: "/var/data/sessions.6"
  primaryMembers:
  - "sessionmgr01-site1:27721"
  - "sessionmgr02-site1:27721"
  secondaryMembers:
  - "sessionmgr01-site2:27721"
  - "sessionmgr02-site2:27721"
  dataPath: "/var/data/sessions.6"

```



```
siteId: "SITE1"
- title: "SESSION-SET63"
  setName: "set63"
  oplogSize: "1024"
  arbiters:
  - "arbiter-site3:27763"
  arbiterDataPath: "/var/data/sessions.63"
  primaryMembers:
  - "sessionmgr01-site2:27763"
  - "sessionmgr02-site2:27763"
  secondaryMembers:
  - "sessionmgr01-site1:27763"
  - "sessionmgr02-site1:27763"
  dataPath: "/var/data/sessions.1/set63"
  shardCount: "4"
  seeds: "sessionmgr01:sessionmgr02:27763"
  primaryMembersTag: "SITE2"
  secondaryMembersTag: "SITE1"
  siteId: "SITE2"
- title: "SESSION-SET68"
  setName: "set68"
  oplogSize: "1024"
  arbiters:
  - "arbiter-site3:27768"
  arbiterDataPath: "/var/data/sessions.68"
  members:
  - "sessionmgr03-site2:27768"
  - "sessionmgr04-site2:27768"
  dataPath: "/var/data/sessions.68"
  hotStandBy: "true"
  shardCount: "4"
  seeds: "sessionmgr01:sessionmgr02:27768"
  primaryMembersTag: "SITE2"
  secondaryMembersTag: "SITE1"
  siteId: "SITE2"
- title: "REPORTING-SET66"
  setName: "set66"
  oplogSize: "1024"
  arbiters:
  - "arbiter-site3:27766"
  arbiterDataPath: "/var/data/sessions.66"
  members:
  - "sessionmgr03-site2:27719"
  - "sessionmgr04-site2:27719"
  dataPath: "/var/data/sessions.66"
  siteId: "SITE2"
- title: "AUDIT-SET65"
  setName: "set65"
  oplogSize: "1024"
  arbiters:
  - "arbiter-site3:27765"
  arbiterDataPath: "/var/data/sessions.65"
  members:
  - "sessionmgr03-site2:27017"
  - "sessionmgr04-site2:27017"
  dataPath: "/var/data/sessions.65"
  siteId: "SITE2"
```

Sample GR Cluster Configuration File - site1

IPv4 Example:

```
grConfig:
  clusterInfo:
    remotePcrfclient01IP: "192.169.21.54"
    remotePcrfclient02IP: "192.169.21.55"
```

IPv6 Example:

```
grConfig:
  clusterInfo:
    remotePcrfclient01IP: "[fd00:854::231]"
    remotePcrfclient02IP: "[fd00:854::232]"
```

Sample GR Cluster Configuration File - site2

IPv4 Example:

```
grConfig:
  clusterInfo:
    remotePcrfclient01IP: "192.169.21.19"
    remotePcrfclient02IP: "192.169.21.20"
```

IPv6 Example:

```
grConfig:
  clusterInfo:
    remotePcrfclient01IP: "[fd00:854::221]"
    remotePcrfclient02IP: "[fd00:854::222]"
```

Sample Set Priority File - site1

```
- op: "set-priority"
  siteId: "SITE1"
  title: "SESSION"
- op: "set-priority"
  siteId: "SITE1"
  title: "SPR"
- op: "set-priority"
  siteId: "SITE1"
  title: "BALANCE"
- op: "set-priority"
  siteId: "SITE1"
  title: "ADMIN"
```

Sample Set Priority File - site2

```
- op: "set-priority"
  siteId: "SITE2"
  title: "SESSION"
```

Sample Shard Configuration File - site1

```
'- op: "modify-shards"
  setName: "set01"
  hotStandBy: "false"
  shardCount: "4"
  seeds: "sessionmgr01:sessionmgr02:27717"
- op: "modify-shards"
  setName: "set07"
  hotStandBy: "true"
  shardCount: "4"
  seeds: "sessionmgr03:sessionmgr04:27722"
```

Sample Shard Configuration File - site2

```
- op: "modify-shards"
  setName: "set63"
  hotStandBy: "false"
  shardCount: "4"
  seeds: "sessionmgr01:sessionmgr02:27763"
- op: "modify-shards"
  setName: "set68"
  hotStandBy: "true"
  shardCount: "4"
  seeds: "sessionmgr03:sessionmgr04:27768"
```

Sample Ring Configuration File

```
- op: "modify-rings"
  setName: "set01"
```

Sample Geo Site Lookup Configuration File - site1

```
grConfig:
  geoLookupConfig:
    - siteId: "SITE1"
      lookupKey:
        - "sitel-gx-client.com"
```



Note The pattern matching is supported for site lookup mapping. In case the incoming host/realm does not match any of the values configured under LookupValues, request is dropped with the following exception in log:

```
GeoHASiteMappingNotFound - No realm/host to site mapping matched for:
<incoming value>
```

Sample Geo Site Lookup Configuration File - site2

```
grConfig:
  geoLookupConfig:
    - siteId: "SITE2"
      lookupKey:
        - "site2-gx-client.com"
```



Note The pattern matching is supported for site lookup mapping. In case the incoming host/realm does not match any of the values configured under LookupValues, request is dropped with the following exception in log:

GeoHASiteMappingNotFound - No realm/host to site mapping matched for:
<incoming value>

Sample Geo-tagging Configuration File - site1

```
- op: "modify-geotag"
  title: "session"
  setName: "set01"
  primaryMembersTag: "SITE1"
  secondaryMembersTag: "SITE2"
- op: "modify-geotag"
  title: "balance"
  setName: "set02"
  primaryMembersTag: "SITE1"
  secondaryMembersTag: "SITE2"
- op: "modify-geotag"
  title: "spr"
  setName: "set04"
  primaryMembersTag: "SITE1"
  secondaryMembersTag: "SITE2"
```

Sample Geo-tagging Configuration File - site2

```
- op: "modify-geotag"
  title: "session"
  setName: "set63"
  primaryMembersTag: "SITE2"
  secondaryMembersTag: "SITE1"
```

Sample Monitor Database Configuration File - site1

```
dbMonitorForLb:
  setName:
    - SPR-SET1
    - SESSION-SET1
    - BALANCE-SET1
    - ADMIN-SET1
```

```
dbMonitorForQns:  
  stopUapi: "false"  
  percentageSessDBFailure: 50  
  setName:  
    - SPR-SET1  
    - SESSION-SET1  
    - BALANCE-SET1  
    - ADMIN-SET1
```

Sample Monitor Database Configuration File - site2

```
dbMonitorForLb:  
  setName:  
    - SESSION-SET63  
dbMonitorForQns:  
  stopUapi: "false"  
  percentageSessDBFailure: 50  
  setName:  
    - SESSION-SET63
```

