

- RedHat Openstack 16.2 Installation** 3
- 3
- Undercloud(Director)** 3
- containers-prepare-parameter.yaml 3
- undercloud.conf 4
- Overcloud** 5
- roles** 5
- network** 5
- template rendering** 6
- node** 6
- nodes.yaml 6
- template** 8
- network-environment.yaml 8
- NIC Config** 9
- controller.yaml 9
- compute.yaml 15
- compute-dvr.yaml 20

RedHat Openstack 16.2 Installation

1. undercloud(director)
2. roles
3. network
4. template rendering
5. node

Undercloud(Director)

containers-prepare-parameter.yaml

```
# Generated with the following on 2022-02-07T15:38:01.717320
#
# openstack tripleo container image prepare default --local-push-
destination --output-env-file containers-prepare-parameter.yaml
#

parameter_defaults:
  ContainerImagePrepare:
    - push_destination: true
      set:
        ceph_alertmanager_image: ose-prometheus-alertmanager
        ceph_alertmanager_namespace: registry.redhat.io/openshift4
        ceph_alertmanager_tag: 4.1
        ceph_grafana_image: rhceph-4-dashboard-rhel8
        ceph_grafana_namespace: registry.redhat.io/rhceph
        ceph_grafana_tag: 4
        ceph_image: rhceph-4-rhel8
        ceph_namespace: registry.redhat.io/rhceph
        ceph_node_exporter_image: ose-prometheus-node-exporter
        ceph_node_exporter_namespace: registry.redhat.io/openshift4
        ceph_node_exporter_tag: v4.1
        ceph_prometheus_image: ose-prometheus
        ceph_prometheus_namespace: registry.redhat.io/openshift4
        ceph_prometheus_tag: 4.1
        ceph_tag: latest
        name_prefix: openstack-
        name_suffix: ''
        namespace: registry.redhat.io/rhosp-rhel8
        neutron_driver: ovn
```

```
rhel_containers: false
tag: '16.1.7'
tag_from_label: '{version}-{release}'
ContainerImageRegistryCredentials:
  registry.redhat.io:
    userid@domain.co.kr: 'password'
ContainerImageRegistryLogin: true
```

undercloud.conf

```
[DEFAULT]
# /
clean_nodes = true
# ( false)
#cleanup = true
#
container_images_file = /home/stack/containers-prepare-parameter.yaml
#
#container_insecure_registries = registry:5000
# telemetry
enable_telemetry = false
local_interface = enp2s0
local_ip = 192.168.10.1/24
#local_mtu = 1500
local_subnet = ctlplane-subnet
# DNS . CloudDomain

#overcloud_domain_name = cloud.localdomain

subnets = ctlplane-subnet
undercloud_admin_host = 192.168.10.3
undercloud_hostname = director.osp.kepri
undercloud_public_host = 192.168.10.2
undercloud_ntp_servers = 192.168.10.250
undercloud_timezone = 'Asia/Seoul'

[ctlplane-subnet]
cidr = 192.168.10.0/24
dhcp_end = 192.168.10.249
dhcp_start = 192.168.10.230
gateway = 192.168.10.1
inspection_iprange = 192.168.10.210,192.168.10.229
masquerade = true
```

Overcloud

Undercloud/Director 가

Overcloud

roles

```
# roles
$ openstack overcloud roles list

# role roles_data.yaml
$ openstack overcloud roles generate \
  --roles-path /usr/share/openstack-tripleo-heat-templates/roles/ \
  -o /home/stack/roles_data.yaml \
  Controller Compute ComputeAlt BlockStorage ObjectStorage CephStorage
```

network

VLAN

```
- name: Storage
  vip: true
  vlan: 30
  name_lower: storage
  ip_subnet: '172.16.1.0/24'
  allocation_pools: [{'start': '172.16.1.4', 'end': '172.16.1.250'}]
  mtu: 1500
- name: StorageMgmt
  name_lower: storage_mgmt
  vip: true
  vlan: 40
  ip_subnet: '172.16.3.0/24'
  allocation_pools: [{'start': '172.16.3.4', 'end': '172.16.3.250'}]
  mtu: 1500
- name: InternalApi
  name_lower: internal_api
  vip: true
  vlan: 20
  ip_subnet: '172.16.2.0/24'
  allocation_pools: [{'start': '172.16.2.4', 'end': '172.16.2.250'}]
  mtu: 1500
- name: Tenant
  vip: false # Tenant network does not use VIPs
  name_lower: tenant
  vlan: 50
```

```
ip_subnet: '172.16.0.0/24'  
allocation_pools: [{'start': '172.16.0.4', 'end': '172.16.0.250'}]  
mtu: 1500  
- name: External  
vip: true  
name_lower: external  
ip_subnet: '192.168.0.0/24'  
allocation_pools: [{'start': '192.168.0.140', 'end': '192.168.0.149'}]  
gateway_ip: '192.168.0.1'  
mtu: 1500
```

template rendering

roles_data.yaml network_data.yaml heat template

```
#!/bin/bash  
cd /usr/share/openstack-tripleo-heat-templates  
./tools/process-templates.py -o ~/templates -n  
/home/stack/custom/network_data.yaml -r /home/stack/custom/roles_data.yaml
```

node

- boot_mode: uefi, bios
- boot_option: local, pxe
- disk_label: gpt, msdos → uefi gpt , bios msdos

: <https://docs.openstack.org/ironic/latest/install/advanced.html>

nodes.yaml

```
nodes:  
- mac:  
  - "52:54:00:00:65:49"  
  name: "control0"  
  pm_type: "pxe_ipmitool"  
  pm_user: "admin"  
  pm_password: "admin"  
  pm_addr: "192.168.0.31"  
  pm_port: "6231"  
  capabilities: "profile:control,node:control0,boot_option:local"  
- mac:  
  - "52:54:00:28:b4:7a"  
  name: "control1"  
  pm_type: "pxe_ipmitool"
```

```
pm_user: "admin"
pm_password: "admin"
pm_addr: "192.168.0.31"
pm_port: "6232"
capabilities: "profile:control,node:control1,boot_option:local"
- mac:
  - "52:54:00:a8:5d:6f"
name: "control2"
pm_type: "pxe_ipmitool"
pm_user: "admin"
pm_password: "admin"
pm_addr: "192.168.0.31"
pm_port: "6233"
capabilities: "profile:control,node:control2,boot_option:local"
- mac:
  - "52:54:00:ef:83:7f"
name: "compute0"
pm_type: "pxe_ipmitool"
pm_user: "admin"
pm_password: "admin"
pm_addr: "192.168.0.31"
pm_port: "6234"
capabilities: "profile:compute,node:compute0,boot_option:local"
- mac:
  - "52:54:00:f9:24:75"
name: "compute1"
pm_type: "pxe_ipmitool"
pm_user: "admin"
pm_password: "admin"
pm_addr: "192.168.0.31"
pm_port: "6235"
capabilities: "profile:compute,node:compute1,boot_option:local"
- mac:
  - "52:54:00:28:e9:fb"
name: "compute2"
pm_type: "pxe_ipmitool"
pm_user: "admin"
pm_password: "admin"
pm_addr: "192.168.0.31"
pm_port: "6236"
capabilities: "profile:compute,node:compute2,boot_option:local"
- mac:
  - "52:54:00:74:7c:52"
name: "compute3"
pm_type: "pxe_ipmitool"
pm_user: "admin"
pm_password: "admin"
pm_addr: "192.168.0.31"
pm_port: "6237"
capabilities: "profile:compute,node:compute3,boot_option:local"
- mac:
```

```
- "52:54:00:76:e0:a4"  
name: "compute4"  
pm_type: "pxe_ipmitool"  
pm_user: "admin"  
pm_password: "admin"  
pm_addr: "192.168.0.31"  
pm_port: "6238"  
capabilities: "profile:compute,node:compute4,boot_option:local"
```

template

network-environment.yaml

```
resource_registry:  
  OS::Triple0::Controller::Net::SoftwareConfig:  
    ../network/config/custom-nic/controller.yaml  
  OS::Triple0::Compute::Net::SoftwareConfig:  
    ../network/config/custom-nic/compute.yaml  
parameter_defaults:  
  StorageNetCidr: '172.16.1.0/24'  
  StorageAllocationPools: [{'start': '172.16.1.4', 'end': '172.16.1.250'}]  
  StorageNetworkVlanID: 30  
  StorageMgmtNetCidr: '172.16.3.0/24'  
  StorageMgmtAllocationPools: [{'start': '172.16.3.4', 'end':  
'172.16.3.250'}]  
  StorageMgmtNetworkVlanID: 40  
  InternalApiNetCidr: '172.16.2.0/24'  
  InternalApiAllocationPools: [{'start': '172.16.2.4', 'end':  
'172.16.2.250'}]  
  InternalApiNetworkVlanID: 20  
  TenantNetCidr: '172.16.0.0/24'  
  TenantAllocationPools: [{'start': '172.16.0.4', 'end': '172.16.0.250'}]  
  TenantNetworkVlanID: 50  
  TenantNetPhysnetMtu: 1500  
  ExternalNetCidr: '192.168.0.0/24'  
  ExternalAllocationPools: [{'start': '192.168.0.140', 'end':  
'192.168.0.149'}]  
  ExternalInterfaceDefaultRoute: '192.168.0.1'  
  DnsServers: ['8.8.8.8']  
  NeutronNetworkType: 'geneve,vlan,flat'  
  NeutronNetworkVLANRanges: 'datacentre:1:1000'  
  BondInterfaceOvsOptions: "bond_mode=active-backup"  
  NeutronEnableDVR: false
```

NIC Config

controller.yaml

```
heat_template_version: rocky
description: >
  Software Config to drive os-net-config to configure multiple interfaces
  for the Controller role.
parameters:
  ControlPlaneIp:
    default: ''
    description: IP address/subnet on the ctlplane network
    type: string
  ControlPlaneSubnetCidr:
    default: ''
    description: >
      The subnet CIDR of the control plane network. (The parameter is
      automatically resolved from the ctlplane subnet's cidr attribute.)
    type: string
  ControlPlaneDefaultRoute:
    default: ''
    description: The default route of the control plane network. (The
    parameter
      is automatically resolved from the ctlplane subnet's gateway_ip
    attribute.)
    type: string
  ControlPlaneStaticRoutes:
    default: []
    description: >
      Routes for the ctlplane network traffic.
      JSON route e.g. [{'destination':'10.0.0.0/16', 'nexthop':'10.0.0.1'}]
      Unless the default is changed, the parameter is automatically resolved
      from the subnet host_routes attribute.
    type: json
  ControlPlaneMtu:
    default: 1500
    description: The maximum transmission unit (MTU) size(in bytes) that is
    guaranteed to pass through the data path of the segments in the
    network.
      (The parameter is automatically resolved from the ctlplane network's
    mtu attribute.)
    type: number

  StorageIpSubnet:
    default: ''
    description: IP address/subnet on the storage network
    type: string
  StorageNetworkVlanID:
    default: 30
```

```
description: Vlan ID for the storage network traffic.
type: number
StorageMtu:
  default: 1500
  description: The maximum transmission unit (MTU) size(in bytes) that is
    guaranteed to pass through the data path of the segments in the
    Storage network.
  type: number
StorageInterfaceRoutes:
  default: []
  description: >
    Routes for the storage network traffic.
    JSON route e.g. [{'destination':'10.0.0.0/16', 'nexthop':'10.0.0.1'}]
    Unless the default is changed, the parameter is automatically resolved
    from the subnet host_routes attribute.
  type: json
StorageMgmtIpSubnet:
  default: ''
  description: IP address/subnet on the storage_mgmt network
  type: string
StorageMgmtNetworkVlanID:
  default: 40
  description: Vlan ID for the storage_mgmt network traffic.
  type: number
StorageMgmtMtu:
  default: 1500
  description: The maximum transmission unit (MTU) size(in bytes) that is
    guaranteed to pass through the data path of the segments in the
    StorageMgmt network.
  type: number
StorageMgmtInterfaceRoutes:
  default: []
  description: >
    Routes for the storage_mgmt network traffic.
    JSON route e.g. [{'destination':'10.0.0.0/16', 'nexthop':'10.0.0.1'}]
    Unless the default is changed, the parameter is automatically resolved
    from the subnet host_routes attribute.
  type: json
InternalApiIpSubnet:
  default: ''
  description: IP address/subnet on the internal_api network
  type: string
InternalApiNetworkVlanID:
  default: 20
  description: Vlan ID for the internal_api network traffic.
  type: number
InternalApiMtu:
  default: 1500
  description: The maximum transmission unit (MTU) size(in bytes) that is
    guaranteed to pass through the data path of the segments in the
    InternalApi network.
```

```
type: number
InternalApiInterfaceRoutes:
  default: []
  description: >
    Routes for the internal_api network traffic.
    JSON route e.g. [{'destination':'10.0.0.0/16', 'nexthop':'10.0.0.1'}]
    Unless the default is changed, the parameter is automatically resolved
    from the subnet host_routes attribute.
  type: json
TenantIpSubnet:
  default: ''
  description: IP address/subnet on the tenant network
  type: string
TenantNetworkVlanID:
  default: 50
  description: Vlan ID for the tenant network traffic.
  type: number
TenantMtu:
  default: 1500
  description: The maximum transmission unit (MTU) size(in bytes) that is
    guaranteed to pass through the data path of the segments in the
    Tenant network.
  type: number
TenantInterfaceRoutes:
  default: []
  description: >
    Routes for the tenant network traffic.
    JSON route e.g. [{'destination':'10.0.0.0/16', 'nexthop':'10.0.0.1'}]
    Unless the default is changed, the parameter is automatically resolved
    from the subnet host_routes attribute.
  type: json
ExternalIpSubnet:
  default: ''
  description: IP address/subnet on the external network
  type: string
ExternalNetworkVlanID:
  default: 10
  description: Vlan ID for the external network traffic.
  type: number
ExternalMtu:
  default: 1500
  description: The maximum transmission unit (MTU) size(in bytes) that is
    guaranteed to pass through the data path of the segments in the
    External network.
  type: number
ExternalInterfaceDefaultRoute:
  default: ''
  description: default route for the external network
  type: string
ExternalInterfaceRoutes:
  default: []
```

```
description: >
  Routes for the external network traffic.
  JSON route e.g. [{'destination':'10.0.0.0/16', 'nexthop':'10.0.0.1'}]
  Unless the default is changed, the parameter is automatically resolved
  from the subnet host_routes attribute.
type: json

DnsServers: # Override this via parameter_defaults
  default: []
  description: >
    DNS servers to use for the Overcloud (2 max for some implementations).
    If not set the nameservers configured in the ctlplane subnet's
    dns_nameservers attribute will be used.
  type: comma_delimited_list
DnsSearchDomains: # Override this via parameter_defaults
  default: []
  description: A list of DNS search domains to be added (in order) to
  resolv.conf.
  type: comma_delimited_list
#####
resources:

  MinViableMtu:
    # This resource resolves the minimum viable MTU for interfaces, bonds
    and
    # bridges that carry multiple VLANs. Each VLAN may have different MTU.
    The
    # bridge, bond or interface must have an MTU to allow the VLAN with the
    # largest MTU.
    type: OS::Heat::Value
    properties:
      type: number
      value:
        yaql:
          expression: $.data.max()
          data:
            - {get_param: ControlPlaneMtu}
            - {get_param: StorageMtu}
            - {get_param: InternalApiMtu}
            - {get_param: TenantMtu}

  OsNetConfigImpl:
    type: OS::Heat::SoftwareConfig
    properties:
      group: script
      config:
        str_replace:
          template:
            get_file: ../../scripts/run-os-net-config.sh
          params:
            $network_config:
```

```
network_config:
```

```
##### Main Interface [Provision]
```

```
- type: interface
  name: nic1
  mtu:
    get_param: ControlPlaneMtu
  use_dhcp: false
  dns_servers:
    get_param: DnsServers
  domain:
    get_param: DnsSearchDomains
  addresses:
  - ip_netmask:
    list_join:
      - /
      - - get_param: ControlPlaneIp
      - get_param: ControlPlaneSubnetCidr
  routes:
    list_concat_unique:
      - get_param: ControlPlaneStaticRoutes
```

```
##### Main Interface [VLAN]
```

```
- type: ovs_bridge
  name: br-local
  dns_server:
    get_param: DnsServers
  use_dhcp: false
```

```
members:
```

```
##### Sub Interface
```

```
- type: interface
  name: nic2
  mtu:
    get_attr: [MinViableMtu, value]
  # force the MAC address of the bridge to this interface
  primary: true
```

```
##### Sub Interface [Storage]
```

```
- type: vlan
  mtu:
    get_param: StorageMtu
  vlan_id:
    get_param: StorageNetworkVlanID
  addresses:
  - ip_netmask:
    get_param: StorageIpSubnet
  routes:
    list_concat_unique:
      - get_param: StorageInterfaceRoutes
```

```
##### Sub Interface [StorageMgmt]
```

```
- type: vlan
  mtu:
    get_param: StorageMgmtMtu
  vlan_id:
    get_param: StorageMgmtNetworkVlanID
  addresses:
    - ip_netmask:
        get_param: StorageMgmtIpSubnet
  routes:
    list_concat_unique:
      - get_param: StorageMgmtInterfaceRoutes
##### Sub Interface [InternalApi]
- type: vlan
  mtu:
    get_param: InternalApiMtu
  vlan_id:
    get_param: InternalApiNetworkVlanID
  addresses:
    - ip_netmask:
        get_param: InternalApiIpSubnet
  routes:
    list_concat_unique:
      - get_param: InternalApiInterfaceRoutes
##### Sub Interface [Tenant]
- type: vlan
  mtu:
    get_param: TenantMtu
  vlan_id:
    get_param: TenantNetworkVlanID
  addresses:
    - ip_netmask:
        get_param: TenantIpSubnet
  routes:
    list_concat_unique:
      - get_param: TenantInterfaceRoutes

##### Main Interface [External - NIC]
- type: ovs_bridge
  name: bridge_name
  mtu:
    get_param: ExternalMtu
  dns_servers:
    get_param: DnsServers
  use_dhcp: false
  addresses:
    - ip_netmask:
        get_param: ExternalIpSubnet
  routes:
    list_concat_unique:
      - get_param: ExternalInterfaceRoutes
      - - default: true
```

```

        next_hop:
            get_param: ExternalInterfaceDefaultRoute
    members:
    - type: interface
      name: nic3
      mtu:
        get_param: ExternalMtu
      use_dhcp: false
      primary: true

#####
outputs:
  OS::stack_id:
    description: The OsNetConfigImpl resource.
    value:
      get_resource: OsNetConfigImpl(undercloud)

```

compute.yaml

```

heat_template_version: rocky
description: >
  Software Config to drive os-net-config to configure multiple interfaces
  for the Compute role.
parameters:
  ControlPlaneIp:
    default: ''
    description: IP address/subnet on the ctlplane network
    type: string
  ControlPlaneSubnetCidr:
    default: ''
    description: >
      The subnet CIDR of the control plane network. (The parameter is
      automatically resolved from the ctlplane subnet's cidr attribute.)
    type: string
  ControlPlaneDefaultRoute:
    default: ''
    description: The default route of the control plane network. (The
    parameter
      is automatically resolved from the ctlplane subnet's gateway_ip
    attribute.)
    type: string
  ControlPlaneStaticRoutes:
    default: []
    description: >
      Routes for the ctlplane network traffic.
      JSON route e.g. [{'destination':'10.0.0.0/16', 'nexthop':'10.0.0.1'}]
      Unless the default is changed, the parameter is automatically resolved
      from the subnet host_routes attribute.
    type: json

```

```
ControlPlaneMtu:
  default: 1500
  description: The maximum transmission unit (MTU) size(in bytes) that is
    guaranteed to pass through the data path of the segments in the
network.
  (The parameter is automatically resolved from the ctlplane network's
mtu attribute.)
  type: number

StorageIpSubnet:
  default: ''
  description: IP address/subnet on the storage network
  type: string
StorageNetworkVlanID:
  default: 30
  description: Vlan ID for the storage network traffic.
  type: number
StorageMtu:
  default: 1500
  description: The maximum transmission unit (MTU) size(in bytes) that is
    guaranteed to pass through the data path of the segments in the
    Storage network.
  type: number
StorageInterfaceRoutes:
  default: []
  description: >
    Routes for the storage network traffic.
    JSON route e.g. [{'destination':'10.0.0.0/16', 'nexthop':'10.0.0.1'}]
    Unless the default is changed, the parameter is automatically resolved
    from the subnet host_routes attribute.
  type: json
InternalApiIpSubnet:
  default: ''
  description: IP address/subnet on the internal_api network
  type: string
InternalApiNetworkVlanID:
  default: 20
  description: Vlan ID for the internal_api network traffic.
  type: number
InternalApiMtu:
  default: 1500
  description: The maximum transmission unit (MTU) size(in bytes) that is
    guaranteed to pass through the data path of the segments in the
    InternalApi network.
  type: number
InternalApiInterfaceRoutes:
  default: []
  description: >
    Routes for the internal_api network traffic.
    JSON route e.g. [{'destination':'10.0.0.0/16', 'nexthop':'10.0.0.1'}]
    Unless the default is changed, the parameter is automatically resolved
```

```

    from the subnet host_routes attribute.
    type: json
  TenantIpSubnet:
    default: ''
    description: IP address/subnet on the tenant network
    type: string
  TenantNetworkVlanID:
    default: 50
    description: Vlan ID for the tenant network traffic.
    type: number
  TenantMtu:
    default: 1500
    description: The maximum transmission unit (MTU) size(in bytes) that is
      guaranteed to pass through the data path of the segments in the
      Tenant network.
    type: number
  TenantInterfaceRoutes:
    default: []
    description: >
      Routes for the tenant network traffic.
      JSON route e.g. [{'destination':'10.0.0.0/16', 'nexthop':'10.0.0.1'}]
      Unless the default is changed, the parameter is automatically resolved
      from the subnet host_routes attribute.
    type: json

  ExternalMtu:
    default: 1500
    description: The maximum transmission unit (MTU) size(in bytes) that is
      guaranteed to pass through the data path of the segments in the
      External network.
    type: number

  DnsServers: # Override this via parameter_defaults
    default: []
    description: >
      DNS servers to use for the Overcloud (2 max for some implementations).
      If not set the nameservers configured in the ctlplane subnet's
      dns_nameservers attribute will be used.
    type: comma_delimited_list
  DnsSearchDomains: # Override this via parameter_defaults
    default: []
    description: A list of DNS search domains to be added (in order) to
      resolv.conf.
    type: comma_delimited_list

#####
resources:

  MinViableMtu:
    # This resource resolves the minimum viable MTU for interfaces, bonds
    and

```

```
# bridges that carry multiple VLANs. Each VLAN may have different MTU.
The
# bridge, bond or interface must have an MTU to allow the VLAN with the
# largest MTU.
type: OS::Heat::Value
properties:
  type: number
  value:
    yaql:
      expression: $.data.max()
      data:
        - {get_param: ControlPlaneMtu}
        - {get_param: StorageMtu}
        - {get_param: InternalApiMtu}
        - {get_param: TenantMtu}

OsNetConfigImpl:
  type: OS::Heat::SoftwareConfig
  properties:
    group: script
    config:
      str_replace:
        template:
          get_file: ../../scripts/run-os-net-config.sh
      params:
        $network_config:
          network_config:

##### Main Interface [Provision]
- type: interface
  name: nic1
  mtu:
    get_param: ControlPlaneMtu
  use_dhcp: false
  dns_servers:
    get_param: DnsServers
  domain:
    get_param: DnsSearchDomains
  addresses:
    - ip_netmask:
      list_join:
        - /
        - - get_param: ControlPlaneIp
        - get_param: ControlPlaneSubnetCidr
  routes:
    list_concat_unique:
      - get_param: ControlPlaneStaticRoutes
      - - default: true
        next_hop:
          get_param: ControlPlaneDefaultRoute
```

```
##### Main Interface [VLAN]
- type: ovs_bridge
  name: br-local
  dns_server:
    get_param: DnsServers
  use_dhcp: false

  members:
##### Sub Interface
- type: interface
  name: nic2
  mtu:
    get_attr: [MinViableMtu, value]
  # force the MAC address of the bridge to this interface
  primary: true
##### Sub Interface [Storage]
- type: vlan
  mtu:
    get_param: StorageMtu
  vlan_id:
    get_param: StorageNetworkVlanID
  addresses:
    - ip_netmask:
        get_param: StorageIpSubnet
  routes:
    list_concat_unique:
      - get_param: StorageInterfaceRoutes
##### Sub Interface [InternalApi]
- type: vlan
  mtu:
    get_param: InternalApiMtu
  vlan_id:
    get_param: InternalApiNetworkVlanID
  addresses:
    - ip_netmask:
        get_param: InternalApiIpSubnet
  routes:
    list_concat_unique:
      - get_param: InternalApiInterfaceRoutes
##### Sub Interface [Tenant]
- type: vlan
  mtu:
    get_param: TenantMtu
  vlan_id:
    get_param: TenantNetworkVlanID
  addresses:
    - ip_netmask:
        get_param: TenantIpSubnet
  routes:
    list_concat_unique:
      - get_param: TenantInterfaceRoutes
```

```
#####  
outputs:  
  OS::stack_id:  
    description: The OsNetConfigImpl resource.  
    value:  
      get_resource: OsNetConfigImpl
```

compute-dvr.yaml

```
# FIXME: This legacy template should be converted to a composable role  
heat_template_version: rocky  
description: >  
  Software Config to drive os-net-config to configure multiple interfaces  
  for the  
  compute role with external bridge for DVR.  
parameters:  
  ControlPlaneIp:  
    default: ''  
    description: IP address/subnet on the ctlplane network  
    type: string  
  ControlPlaneSubnetCidr:  
    default: ''  
    description: >  
      The subnet CIDR of the control plane network. (The parameter is  
      automatically resolved from the ctlplane subnet's cidr attribute.)  
    type: string  
  ControlPlaneDefaultRoute:  
    default: ''  
    description: The default route of the control plane network. (The  
parameter  
    is automatically resolved from the ctlplane subnet's gateway_ip  
attribute.)  
    type: string  
  ControlPlaneStaticRoutes:  
    default: []  
    description: >  
      Routes for the ctlplane network traffic.  
      JSON route e.g. [{'destination':'10.0.0.0/16', 'nexthop':'10.0.0.1'}]  
      Unless the default is changed, the parameter is automatically resolved  
      from the subnet host_routes attribute.  
    type: json  
  ControlPlaneMtu:  
    default: 1500  
    description: The maximum transmission unit (MTU) size(in bytes) that is  
      guaranteed to pass through the data path of the segments in the  
network.  
      (The parameter is automatically resolved from the ctlplane network's  
mtu attribute.)  
    type: number
```

```
StorageIpSubnet:
  default: ''
  description: IP address/subnet on the storage network
  type: string
StorageNetworkVlanID:
  default: 30
  description: Vlan ID for the storage network traffic.
  type: number
StorageMtu:
  default: 1500
  description: The maximum transmission unit (MTU) size(in bytes) that is
    guaranteed to pass through the data path of the segments in the
    Storage network.
  type: number
StorageInterfaceRoutes:
  default: []
  description: >
    Routes for the storage network traffic.
    JSON route e.g. [{'destination':'10.0.0.0/16', 'nexthop':'10.0.0.1'}]
    Unless the default is changed, the parameter is automatically resolved
    from the subnet host_routes attribute.
  type: json
StorageMgmtIpSubnet:
  default: ''
  description: IP address/subnet on the storage_mgmt network
  type: string
StorageMgmtNetworkVlanID:
  default: 40
  description: Vlan ID for the storage_mgmt network traffic.
  type: number
StorageMgmtMtu:
  default: 1500
  description: The maximum transmission unit (MTU) size(in bytes) that is
    guaranteed to pass through the data path of the segments in the
    StorageMgmt network.
  type: number
StorageMgmtInterfaceRoutes:
  default: []
  description: >
    Routes for the storage_mgmt network traffic.
    JSON route e.g. [{'destination':'10.0.0.0/16', 'nexthop':'10.0.0.1'}]
    Unless the default is changed, the parameter is automatically resolved
    from the subnet host_routes attribute.
  type: json
InternalApiIpSubnet:
  default: ''
  description: IP address/subnet on the internal_api network
  type: string
InternalApiNetworkVlanID:
  default: 20
  description: Vlan ID for the internal_api network traffic.
```

```
type: number
InternalApiMtu:
  default: 1500
  description: The maximum transmission unit (MTU) size(in bytes) that is
    guaranteed to pass through the data path of the segments in the
    InternalApi network.
  type: number
InternalApiInterfaceRoutes:
  default: []
  description: >
    Routes for the internal_api network traffic.
    JSON route e.g. [{'destination':'10.0.0.0/16', 'nexthop':'10.0.0.1'}]
    Unless the default is changed, the parameter is automatically resolved
    from the subnet host_routes attribute.
  type: json
TenantIpSubnet:
  default: ''
  description: IP address/subnet on the tenant network
  type: string
TenantNetworkVlanID:
  default: 50
  description: Vlan ID for the tenant network traffic.
  type: number
TenantMtu:
  default: 1500
  description: The maximum transmission unit (MTU) size(in bytes) that is
    guaranteed to pass through the data path of the segments in the
    Tenant network.
  type: number
TenantInterfaceRoutes:
  default: []
  description: >
    Routes for the tenant network traffic.
    JSON route e.g. [{'destination':'10.0.0.0/16', 'nexthop':'10.0.0.1'}]
    Unless the default is changed, the parameter is automatically resolved
    from the subnet host_routes attribute.
  type: json
ExternalIpSubnet:
  default: ''
  description: IP address/subnet on the external network
  type: string
ExternalNetworkVlanID:
  default: 1
  description: Vlan ID for the external network traffic.
  type: number
ExternalMtu:
  default: 1500
  description: The maximum transmission unit (MTU) size(in bytes) that is
    guaranteed to pass through the data path of the segments in the
    External network.
  type: number
```

```

ExternalInterfaceRoutes:
  default: []
  description: >
    Routes for the external network traffic.
    JSON route e.g. [{'destination':'10.0.0.0/16', 'nexthop':'10.0.0.1'}]
    Unless the default is changed, the parameter is automatically resolved
    from the subnet host_routes attribute.
  type: json
# Uncomment when including environments/network-management.yaml and
setting
# default route on the Management interface. Also comment out the default
# route on the Control Plane and add the Management network to the roles
# default_route_networks in roles data.
# ManagementInterfaceDefaultRoute:
#   default: ''
#   description: default route for the management network
#   type: string
DnsServers: # Override this via parameter_defaults
  default: []
  description: >
    DNS servers to use for the Overcloud (2 max for some implementations).
    If not set the nameservers configured in the ctlplane subnet's
    dns_nameservers attribute will be used.
  type: comma_delimited_list
DnsSearchDomains: # Override this via parameter_defaults
  default: []
  description: A list of DNS search domains to be added (in order) to
  resolv.conf.
  type: comma_delimited_list

#####
resources:

  MinViableMtu:
    # This resource resolves the minimum viable MTU for interfaces, bonds
and
    # bridges that carry multiple VLANs. Each VLAN may have different MTU.
The
    # bridge, bond or interface must have an MTU to allow the VLAN with the
    # largest MTU.
    type: OS::Heat::Value
    properties:
      type: number
      value:
        yaql:
          expression: $.data.max()
          data:
            - {get_param: ControlPlaneMtu}
            - {get_param: StorageMtu}
            - {get_param: InternalApiMtu}
            - {get_param: TenantMtu}

```

```
OsNetConfigImpl:  
  type: OS::Heat::SoftwareConfig  
  properties:  
    group: script  
    config:  
      str_replace:  
        template:  
          get_file: ../../scripts/run-os-net-config.sh  
      params:  
        $network_config:  
          network_config:
```

```
##### Main Interface [Provision]  
  - type: interface  
    name: nic1  
    mtu:  
      get_param: ControlPlaneMtu  
    use_dhcp: false  
    dns_servers:  
      get_param: DnsServers  
    domain:  
      get_param: DnsSearchDomains  
    addresses:  
      - ip_netmask:  
          list_join:  
            - /  
            - - get_param: ControlPlaneIp  
              - get_param: ControlPlaneSubnetCidr  
    routes:  
      list_concat_unique:  
        - get_param: ControlPlaneStaticRoutes  
        - - default: true  
          next_hop:  
            get_param: ControlPlaneDefaultRoute
```

```
##### Main Interface [VLAN]  
  - type: ovs_bridge  
    name: br-local  
    dns_server:  
      get_param: DnsServers  
    use_dhcp: false
```

```
members:
```

```
##### Sub Interface  
  - type: interface  
    name: nic2  
    mtu:  
      get_attr: [MinViableMtu, value]  
    # force the MAC address of the bridge to this interface  
    primary: true
```

```
##### Sub Interface [Storage]
```

```

- type: vlan
  mtu:
    get_param: StorageMtu
  vlan_id:
    get_param: StorageNetworkVlanID
  addresses:
  - ip_netmask:
    get_param: StorageIpSubnet
  routes:
    list_concat_unique:
      - get_param: StorageInterfaceRoutes
##### Sub Interface [InternalApi]
- type: vlan
  mtu:
    get_param: InternalApiMtu
  vlan_id:
    get_param: InternalApiNetworkVlanID
  addresses:
  - ip_netmask:
    get_param: InternalApiIpSubnet
  routes:
    list_concat_unique:
      - get_param: InternalApiInterfaceRoutes
##### Sub Interface [Tenant]
- type: vlan
  mtu:
    get_param: TenantMtu
  vlan_id:
    get_param: TenantNetworkVlanID
  addresses:
  - ip_netmask:
    get_param: TenantIpSubnet
  routes:
    list_concat_unique:
      - get_param: TenantInterfaceRoutes

#####

# External bridge for DVR (no IP address required)
- type: ovs_bridge
  name: bridge_name
  mtu:
    get_param: ExternalMtu
  dns_servers:
    get_param: DnsServers
  use_dhcp: false
  members:
  - type: interface
    name: nic3
    mtu:
      get_param: ExternalMtu

```

```
        primary: true
    # Uncomment when including environments/network-management.yaml
    # If setting default route on the Management interface, comment
    # out the default route on the Control Plane.
    #- type: interface
    #   name: nic7
    #   mtu:
    #     get_param: ManagementMtu
    #   use_dhcp: false
    #   addresses:
    #     - ip_netmask:
    #         get_param: ManagementIpSubnet
    #   routes:
    #     list_concat_unique:
    #       - get_param: ManagementInterfaceRoutes
    #       - - default: true
    #         next_hop:
    #           get_param: ManagementInterfaceDefaultRoute
outputs:
  OS::stack_id:
    description: The OsNetConfigImpl resource.
    value:
      get_resource: OsNetConfigImpl
```

From:
<https://atl.kr/dokuwiki/> - **AllThatLinux!**

Permanent link:
https://atl.kr/dokuwiki/doku.php/redhat_openstack_16.2_installation

Last update: **2022/02/18 06:26**

