Red Hat OpenStack 8 + Juniper Contrail 3.0.2 Implementation with Complete vRouter Lifecycle

November 14th, 2016 - Rev 2.0 - fork & rewrite to add vRouter lifecycle fixes.
February 28th, 2017 - Rev 2.1 - update from comments and additional testing.

Disclaimer

This document is not an update from the similar document named "Red Hat OpenStack 8 + Juniper Contrail 3.0.2. This is a complete re-write including new functionality.

Introduction

Ongoing work between Red Hat and Juniper has shown strong interest from customers in a joint architecture including RHT OSP and JNPR Contrail. This document will review the deployment process for this architecture, specific design decisions, and discuss any concerns to be aware of and recommendations when implementing and running.

For the OSP8 deployment many steps are manual as a full integration of Contrail into OSP director has not been completed. OSP is installed and then Contrail is installed after. We have included functionality to automatically install vRouter modules into nova. Future versions of OSP should allow for more deep integration, removing many of the manual steps.

Deployment Process

Undercloud

Install the undercloud following the official documentation:
https://access.redhat.com/documentation/en/red-hat-openstack-platform/version-8/director-installation-and-usage/#chap-Installing_the_Undercloud

Overcloud

Deployment Steps:
Openstack controller includes contrail-neutron-plugin and Openstack compute includes contrail-vrouter. Contrail-controller uses rabbitmq (amqp) and keystone services running in Openstack controller. Contrail-neutron-plugin and contrail-vrouter depends on contrail-api which runs on Contrail controller. Because of these dependencies, deployment steps are split into four steps.

0. Prepare Overcloud configuration.
1. Deploy Overcloud vanilla controllers.
2. Deploy Contrail Backend and connect to Overcloud Controllers.
3. Update Overcloud Controllers with Neutron Contrail plugin.
4. Deploy Compute Nodes.

Step 0 - Prepare Overcloud configuration

- Create contrail local repo

Contrail packages are provided as an rpm contrail-install-packages-<release>-<version>~<sku>.el7.noarch.rpm. Contrail packages can be extracted from the rpm and a local repo can be created in any local servers in the lab. This local repo can be used to install contrail packages in the openstack-controller and openstack-compute nodes.

Extract Contrail packages from contrail-install-packages-<release>-<version>~<sku>.el7.noarch.rpm

```
[root@hypervisor] # yum -y install createrepo

[root@hypervisor] # mkdir /root/contrail-repo
[root@hypervisor] # cd /root/contrail-repo
[root@hypervisor] # cp
contrail-install-packages-<release>-<version>~<sku>.el7.noarch.rpm
/root/contrail-repo/
# rpm2cpio contrail-install-packages-3.0.2.0-51~liberty.el7.noarch.rpm |
cpio -idmv
./opt/contrail/contrail_packages/contrail_rpms.tgz
./opt/contrail/contrail_packages/setup.sh.new
./opt/contrail/puppet/contrail-puppet-manifest.tgz
278772 blocks
#
```

All contrail packages are available in below tgz

```
 [root@hypervisor] # ls
/root/contrail-repo/opt/contrail/contrail_packages/contrail_rpms.tgz
```

Extract contrail_rpms.tgz and create a local repo

```
[root@hypervisor] # mkdir /root/contrail-repo/3.0.2.0-51/
[root@hypervisor] # tar -xzf
/root/contrail-repo/opt/contrail/contrail_packages/contrail_rpms.tgz -C
/root/contrail-repo/3.0.2.0-51/
[root@hypervisor] # createrepo /root/contrail-repo/3.0.2.0-51/
```

```
Below packages are Contrail dependencies missing in upstream RHEL repos and
included in missing-depends.tgz available at
https://app.box.com/s/xgcwul1attt90zkee1iw7qjw0krme025. Download and untar
the archive to retrieve the packages and setup a local repo as instructed
below
```

```
net-snmp-python-5.7.2-24.el7_2.1.x86_64.rpm
python-meld3-0.6.10-1.el7cp.x86_64.rpm
supervisor-3.0-2.el7cp.noarch.rpm
protobuf-2.5.0-8.0contrail0.el7.x86_64.rpm
yum-plugin-priorities-1.1.31-34.0contrail0.el7.noarch.rpm

[root@hypervisor] # mkdir -p /root/contrail-repo/
[root@hypervisor] # tar -xzf missing-depends.tgz -C /root/contrail-repo/
[root@hypervisor] # createrepo /root/contrail-repo/missing-depends/
```

Host the repo over http. Below commands starts an python http server at port 8080

```
[root@hypervisor] # cd /root/contrail-repo/
[root@hypervisor] # python -m SimpleHTTPServer 8080 &
```

Contrail and dependencies repo can be accessed by adding below repo file to /etc/yum.repos.d/ directory to any node

```
# vi /root/contrail-repo/contrail.repo
[contrail-3.0.2.0-51]
name=contrail-3.0.2.0-51
baseurl=http://<IP Address of the repo>:8080/3.0.2.0-51
enabled=1
priority=1
gpgcheck=0

[missing-depends]
name=missing-depends
baseurl=http://<IP Address of the repo>:8080/missing-depends
enabled=1
priority=1
gpgcheck=0
```

These two repo files needs to be added to the overcloud image during overcloud Image customization to add contrail packages.

- Prepare the overcloud images

Add the contrail rpms, updated openstack-puppet-modules, and removing openvswitch.

a. Controllers:

i. Create new overcloud image copy for controller:

```
[root@hypervisor] # mkdir -p /tmp/overcloud-prep
[stack@undercloud] # scp
/home/stack/images/overcloud-full.qcow2
root@<hypervisor>:/tmp/overcloud-prep/overcloud-full-controller
-with-contrail.qcow2
```

ii. Add neutron-plugin packages:
Copy contrail local repo definition to the image

```
[root@hypervisor] # cd /tmp/overcloud-prep
[root@hypervisor] # virt-customize -a
overcloud-full-controller-with-contrail.qcow2 --upload
/root/contrail-repo/contrail.repo:/etc/yum.repos.d/
```

Clean up yum db

```
[root@hypervisor] # virt-customize -a
overcloud-full-controller-with-contrail.qcow2 --run-command
'yum clean all'
```

Install priorities package so the repo priority configuration in the contrail.repo will be effective.

```
[root@hypervisor] # virt-customize -a
overcloud-full-controller-with-contrail.qcow2  --install
yum-plugin-priorities
```

Attach RHEL subscription and enable required repos

```
[root@hypervisor] # virt-customize -a
overcloud-full-controller-with-contrail.qcow2 --run-command
'subscription-manager register --username <username> --password
<password> --force; subscription-manager attach --pool
<pool-id>; subscription-manager repos
--enable=rhel-7-server-rpms --enable=rhel-7-server-extras-rpms
--enable=rhel-7-server-openstack-8-rpms; subscription-manager
repos --disable=*; subscription-manager repos
--enable=rhel-7-server-rpms --enable=rhel-7-server-extras-rpms
--enable=rhel-7-server-openstack-8-rpms'
```

Install contrail-neutron plugin and its related packages in the controller vm

```
[root@hypervisor] # virt-customize -a
overcloud-full-controller-with-contrail.qcow2 --install
python-contrail,neutron-plugin-contrail,python-bottle,consisten
t_hash --selinux-relabel
```

Deattach subscription and remove contrail local repo definition

```
[root@hypervisor] # virt-customize -a
overcloud-full-controller-with-contrail.qcow2 --run-command
'subscription-manager remove --all'
```

```
[root@hypervisor] # virt-customize -a
overcloud-full-controller-with-contrail.qcow2 --run-command 'rm
-f /etc/yum.repos.d/contrail.repo'
```

Clean up yum cache. This command may fail as all repo definitions are removed

```
[root@hypervisor] # virt-customize -a
overcloud-full-controller-with-contrail.qcow2 --run-command
'yum clean all'
```

Now that the openstack-controller is installed with contrail-plugin and it needs to be added to the glance in the undercloud.

Scp image back from hypervisor to undercloud node.

```
scp
/tmp/overcloud-prep/overcloud-full-controller-with-contrail.qcow
2 stack@<undercloud>:/home/stack/images/
```

iii. Load image into Glance:

```
##get existing kernel_id & ramdisk_id from overcloud-full
[stack@undercloud]# kernel_id=$(glance image-show overcloud-full
| grep kernel_id | cut -d '|' -f3)
[undercloud]#ramdisk_id=$(glance image-show overcloud-full | grep
ramdisk_id | cut -d '|' -f3)

##Load new image file
[stack@undercloud]# glance image-create --name
overcloud-full-controller-with-contrail --disk-format qcow2
--container-format bare --property kernel_id="$kernel_id"
--property ramdisk_id="$ramdisk_id" --is-public true <
/home/stack/images/overcloud-full-controller-with-contrail.qcow2

##Check image was properly loaded
[stack@undercloud]# glance image-show
overcloud-full-controller-with-contrail
+-----------------------+---------------------------------------+
| Property              | Value                                 |
+-----------------------+---------------------------------------+
| Property 'kernel_id'  | 999542b8-83c4-49f8-b65e-e5337e6d1db8  |
| Property 'ramdisk_id'| b49d8f46-d702-4f36-adae-e919b445a834  |
| checksum              | 6c59a4b03799d326c46489d6ac88093a      |
| container_format      | bare                                  |
| created_at            | 2016-11-08T23:02:48.000000            |
| deleted               | False                                 |
| disk_format           | qcow2                                 |
| id                    | 11d91df5-ebf8-4f05-b8ed-97c3cd2eaf67  |
| is_public             | True                                   |
| min_disk              | 0                                      |
| min_ram               | 0                                      |
| name                  |overcloud-full-controller-with-contrail|
| owner                 | c0181bc21ede44bdad81c0e5515a165a       |
| protected             | False                                  |
| size                  | 1120075776                             |
| status                | active                                 |
| updated_at            | 2016-11-08T23:02:52.000000             |
+-----------------------+---------------------------------------+
```

b. Compute:

i. Create new overcloud image copies for compute

```
[root@hypervisor] # mkdir -p /tmp/overcloud-compute-prep
[stack@undercloud] # scp
/home/stack/images/overcloud-full.qcow2
root@<hypervisor>:/tmp/overcloud-compute-prep/overcloud-full-co
mpute-with-contrail.qcow2
```

ii. Add vrouter packages

Copy contrail local repo definition to the image

```
[root@hypervisor] # cd /tmp/overcloud-compute-prep
[root@hypervisor] # virt-customize -a
overcloud-full-compute-with-contrail.qcow2 --upload
/root/contrail-repo/contrail.repo:/etc/yum.repos.d/
```

Clean up yum db

```
[root@hypervisor] # virt-customize -a
overcloud-full-compute-with-contrail.qcow2 --run-command 'yum
clean all'
```

Attach RHEL subscription and enable required repos. Update username, password, pool-id in below commands and then execute from hypervisor

```
[root@hypervisor] # virt-customize -a
overcloud-full-compute-with-contrail.qcow2 --run-command
'subscription-manager register --username <username> --password
<password> --force; subscription-manager attach --pool
<pool-id>; subscription-manager repos
--enable=rhel-7-server-rpms --enable=rhel-7-server-extras-rpms
--enable=rhel-7-server-openstack-8-rpms; subscription-manager
repos --disable=*; subscription-manager repos
--enable=rhel-7-server-rpms --enable=rhel-7-server-extras-rpms
--enable=rhel-7-server-openstack-8-rpms'
```

Install priorities package so the repo priority configuration in the contrail.repo will be effective.

```
[root@hypervisor] # virt-customize -a
overcloud-full-compute-with-contrail.qcow2  --install
yum-plugin-priorities
```

Install contrail vrouter package in the compute image

```
[root@hypervisor] #  virt-customize -a
overcloud-full-compute-with-contrail.qcow2 --install
contrail-openstack-vrouter --selinux-relabel

Following packages will be installed as part of above
customization to the openstack compute image
    ●   consistent_hash-1.0-0contrail0.el7.noarch.rpm
    ●   contrail-lib-3.0.2.0-51.el7.x86_64.rpm
    ●   contrail-nodemgr-3.0.2.0-51.el7.x86_64.rpm
    ●   contrail-openstack-vrouter-3.0.2.0-51.el7.noarch.rpm
    ●   contrail-setup-3.0.2.0-51.el7.noarch.rpm
    ●   contrail-utils-3.0.2.0-51.el7.x86_64.rpm
    ●   contrail-vrouter-3.0.2.0-51.el7.x86_64.rpm
    ●   contrail-vrouter-agent-3.0.2.0-51.el7.x86_64.rpm
    ●   contrail-vrouter-common-3.0.2.0-51.el7.noarch.rpm
    ●   contrail-vrouter-init-3.0.2.0-51.el7.x86_64.rpm
    ●   contrail-vrouter-source-3.0.2.0-51.el7.x86_64.rpm
    ●   contrail-vrouter-utils-3.0.2.0-51.el7.x86_64.rpm
    ●   python-bitarray-0.8.0-0contrail.el7.x86_64.rpm
    ●   python-bottle-0.11.6-0contrail.el7.noarch.rpm
    ●   python-contrail-3.0.2.0-51.el7.x86_64.rpm
    ●   python-contrail-vrouter-api-3.0.2.0-51.el7.x86_64.rpm
    ●   python-meld3-0.6.10-1.el7cp.x86_64.rpm
    ●   python-opencontrail-vrouter-netns-3.0.2.0-51.el7.x86_64.
        rpm
    ●   python-pycassa-1.10.0-0contrail.el7.noarch.rpm
    ●   python-thrift-0.9.1-0contrail.el7.x86_64.rpm
    ●   supervisor-3.0-2.el7cp.noarch.rpm
    ●   tunctl-1.5-3.el6.x86_64.rpm
    ●   xmltodict-0.7.0-0contrail.el7.noarch.rpm
```

iii. Remove openvswitch packages

```
[root@hypervisor] # virt-customize -a
overcloud-full-compute-with-contrail.qcow2 --uninstall
openvswitch --selinux-relabel
```

iv. Deattach subscription and remove contrail local repo definition

```
[root@hypervisor] # virt-customize -a
overcloud-full-compute-with-contrail.qcow2 --run-command
'subscription-manager remove --all'
```

```
[root@hypervisor] # virt-customize -a
overcloud-full-compute-with-contrail.qcow2 --run-command 'rm -f
/etc/yum.repos.d/contrail.repo'
```

v. Clean up yum cache. This command may fail as all repo definitions are removed

```
[root@hypervisor] # virt-customize -a
overcloud-full-compute-with-contrail.qcow2 --run-command 'yum
clean all'
```

vi. Scp image back from hypervisor to undercloud node.

```
scp
/tmp/overcloud-compute-prep/overcloud-full-compute-with-contrail
.qcow2 stack@<undercloud>:/home/stack/images/
```

Now the contrail vrouter packages are added to compute image and it needs to be added to glance in undercloud

vii. Load image into glance

```
##get existing kernel_id & ramdisk_id from overcloud-full
[undercloud]# kernel_id=$(glance image-show overcloud-full |
grep kernel_id | cut -d '|' -f3)
[undercloud]# ramdisk_id=$(glance image-show overcloud-full |
grep ramdisk_id | cut -d '|' -f3)

##Load new image file
[undercloud]# glance image-create --name
overcloud-full-compute-with-contrail --disk-format qcow2
--container-format bare --property kernel_id="$kernel_id"
--property ramdisk_id="$ramdisk_id" --is-public true <
overcloud-full-compute-with-contrail.qcow2

##Check image was properly loaded
[undercloud]# glance image-show
overcloud-full-compute-with-contrail
+----------------------+--------------------------------------+
| Property             | Value                                |
+----------------------+--------------------------------------+
| Property 'kernel_id' | 999542b8-83c4-49f8-b65e-e5337e6d1db8 |
| Property 'ramdisk_id'| b49d8f46-d702-4f36-adae-e919b445a834 |
| checksum             | 19ae486554ec67765021d7902c3bf014     |
| container_format     | bare                                 |
| created_at           | 2016-11-09T22:08:17.000000           |
| deleted              | False                                |
| disk_format          | qcow2                                |
| id                   | fbccdab8-6b4e-4a30-8215-d85ab2de6b3e |
| is_public              | True                                 |
| min_disk               | 0                                    |
| min_ram                | 0                                    |
| name                   | overcloud-full-compute-with-contrail |
| owner                  | c0181bc21ede44bdad81c0e5515a165a     |
| protected              | False                                |
| size                   | 1360330752                           |
| status                 | active                               |
| updated_at             | 2016-11-09T22:08:24.000000           |
+------------------------+--------------------------------------+
```

- Create deploy.sh scripts:
  a. Vanilla Controllers for Step 1 (/home/stack/deploy.sh):

```
#!/bin/bash
set -x
if [ $PWD != /home/stack ] ; then echo "USAGE: $0 this script needs to
be executed in /home/stack"; exit 1 ; fi

# deploy.sh <control_scale compute_scale ceph_scale>
control_scale=3
compute_scale=0
ceph_scale=0
if [ $# -eq 3 ]; then
control_scale="$1"
compute_scale="$2"
ceph_scale="$3"
fi
echo "control_scale=$control_scale, compute_scale=$compute_scale,
ceph_scale=$ceph_scale"


DIR="$( cd "$( dirname "${BASH_SOURCE[0]}" )" && pwd )"
template_base_dir="$DIR/templates"

ntpserver=<local ntp server address>

openstack overcloud deploy --templates \
-e
/usr/share/openstack-tripleo-heat-templates/environments/network-isola
tion.yaml \
-e ${template_base_dir}/network-environment.yaml \
--control-flavor control --compute-flavor compute
--ceph-storage-flavor ceph-storage \
--control-scale $control_scale --compute-scale $compute_scale
--ceph-storage-scale $ceph_scale \
--ntp-server $ntpserver \
--neutron-network-type vxlan --neutron-tunnel-types vxlan
```

b. Contrail-enabled OSP Controllers for Step 3 (/home/stack/deploy-with-contrail.sh):

```
#!/bin/bash
set -x
if [ $PWD != /home/stack ] ; then echo "USAGE: $0 this script needs to
be executed in /home/stack"; exit 1 ; fi

# deploy.sh <control_scale compute_scale ceph_scale>
control_scale=3
compute_scale=0
ceph_scale=0
if [ $# -eq 3 ]; then
control_scale="$1"
compute_scale="$2"
ceph_scale="$3"
fi
echo "control_scale=$control_scale, compute_scale=$compute_scale,
ceph_scale=$ceph_scale"


DIR="$( cd "$( dirname "${BASH_SOURCE[0]}" )" && pwd )"
template_base_dir="$DIR/templates"

ntpserver=<local ntp server>

openstack overcloud deploy --templates \
-e
/usr/share/openstack-tripleo-heat-templates/environments/network-isola
tion.yaml \
-e ${template_base_dir}/network-environment.yaml \
-e ${template_base_dir}/neutron-opencontrail.yaml \
--control-flavor control --compute-flavor compute
--ceph-storage-flavor ceph-storage \
--control-scale $control_scale --compute-scale $compute_scale
--ceph-storage-scale $ceph_scale \
--ntp-server $ntpserver \
--neutron-network-type vxlan --neutron-tunnel-types vxlan
```

- Prepare deployment yaml files:
  a. Copy necessary templates from /usr/share/openstack-tripleo-heat-templates/
  b. /home/stack/templates/network-environment.yaml:

```
resource_registry:
  OS::TripleO::Compute::Net::SoftwareConfig: nic-configs/compute.yaml
  OS::TripleO::Controller::Net::SoftwareConfig:
nic-configs/controller.yaml
  OS::TripleO::Compute::Ports::ExternalPort:
/usr/share/openstack-tripleo-heat-templates/network/ports/external.yaml
```

```
parameter_defaults:
  ExternalInterfaceDefaultRoute: <gateway-ip-address-of-external-iface
of Undercloud in x.x.x.x format>

  ExternalNetCidr: <network cidr of external iface as x.x.x.x/x>

  # Gateway router for the provisioning network (or Undercloud IP)
  ControlPlaneDefaultRoute: <IP address of ctlplane interface of
Undercloud Node in x.x.x.x format>
  # The IP address of the EC2 metadata server. Generally the IP of the
Undercloud
  EC2MetadataIp: <IP address of ctlplane interface of Undercloud Node
in x.x.x.x format>
  # Set to "br-ex" if using floating IPs on native VLAN on bridge br-ex
  NeutronExternalNetworkBridge: "''"

   DnsServers: ["<IP address of ctlplane interface of Undercloud Node
in x.x.x.x format>"]

  controllerImage: 'overcloud-full-controller-with-contrail'

  NovaImage: 'overcloud-full-compute-with-contrail'

  # Start and end of IP range for external interfaces in the overcloud
  ExternalAllocationPools: [{'start': 'x.x.x.x', 'end': 'x.x.x.x'}]
```

c. /home/stack/templates/neutron-opencontrail.yaml:

```
resource_registry:
  OS::TripleO::ControllerExtraConfigPre:
/usr/share/openstack-tripleo-heat-templates/puppet/extraconfig/pre_dep
loy/controller/neutron-opencontrail.yaml
  OS::TripleO::ComputeExtraConfigPre:
/usr/share/openstack-tripleo-heat-templates/puppet/extraconfig/pre_dep
loy/controller/neutron-opencontrail.yaml
parameter_defaults:
  NeutronCorePlugin:
neutron_plugin_contrail.plugins.opencontrail.contrail_plugin.NeutronPl
uginContrailCoreV2
  NeutronServicePlugins:
neutron_plugin_contrail.plugins.opencontrail.loadbalancer.v2.plugin.Lo
adBalancerPluginV2
  NeutronEnableDHCPAgent: false
  NeutronEnableL3Agent: false
  NeutronEnableMetadataAgent: false
  NeutronEnableOVSAgent: false
  NeutronEnableTunnelling: false

  ContrailApiServerIp: ['<Contrail VIP IP ADDRESS in x.x.x.x format>']

ContrailExtensions:'ipam:neutron_plugin_contrail.plugins.opencontrail.
contrail_plugin_ipam.NeutronPluginContrailIpam,policy:neutron_
```

```
plugin_contrail.plugins.opencontrail.contrail_plugin_policy.NeutronPlu
ginContrailPolicy,route-table:neutron_plugin_contrail.plugins.ope
ncontrail.contrail_plugin_vpc.NeutronPluginContrailVpc,contrail:None'
  ContrailApiServerPort: 8082
  ContrailMultiTenancy: true

parameters:
  controllerExtraConfig:
    neutron::config::plugin_opencontrail_config:
      keystone_authtoken/auth_host:
        value: '%{hiera(''keystone_public_api_vip'')}'
      keystone_authtoken/auth_port:
        value: 5000
      keystone_authtoken/auth_protocol:
        value: 'http'

  NovaComputeExtraConfig:
    contrail::vrouter::install: 'contrail-vrouter'
    #contrail::vrouter::config::vhost_ip: '<Interface assigned for
Vrouter>'
    contrail::vrouter::config::discovery_ip: '<Contrail VIP IP ADDRESS
in x.x.x.x format>'
    contrail::vrouter::config::device: '<Interface assigned for
vRouter>'
    contrail::vrouter::config::compute_device: '<Interface assigned
for vRouter>'
    contrail::vrouter::config::mask: 24
    contrail::vrouter::config::netmask: '<Network mask of Interface
assigned for vRouter in x.x.x.x format>'
    contrail::vrouter::config::gateway: '<Gateway address of Interface
assigned for vRouter in x.x.x.x format'
    contrail::vrouter::config::vrouter_nodemgr_config:
      DISCOVERY/server:
        value: '<CNTRL DISC VIP>'
      DISCOVERY/port:
        value: '5998'
    contrail::vrouter::config::kmod_path: 'vrouter'
    contrail::vrouter::config::vgw_interface: '__VGW_INTF_LIST__'
    contrail::vrouter::config::vgw_public_subnet: '__VGW_SUBNET_IP__'
    # start /etc/contrail/contrail-vrouter-agent.conf
    contrail::vrouter::config::vrouter_agent_config:
      NETWORKS/control_network_ip:
        value: '%{::ipaddress_VROUTER_INTERFACE_NAME}'
      DISCOVERY/server:
        value: '<Contrail VIP IP ADDRESS in x.x.x.x format>'
      VIRTUAL-HOST-INTERFACE/name:
        value: 'vhost0'
      VIRTUAL-HOST-INTERFACE/physical_interface:
        value: '<Interface assigned for vRouter>'
      HYPERVISOR/type:
        value: 'kvm'
      FLOWS/thread_count:
        value: 2
      METADATA/metadata_proxy_secret:
        value: '<metadata_proxy_shared_secret
in neutron section of /etc/nova/nova.conf in openstack controller>'
      VIRTUAL-HOST-INTERFACE/gateway:
        value: '<Gateway IP of Interface assigned for vRouter in
x.x.x.x format>'
    #end /etc/contrail/contrail-vrouter-agent.conf
    #start /etc/contrail/vnc_api_lib.ini
    contrail::vnc_api::vnc_api_config:
      auth/AUTHN_TYPE:
        value: 'keystone'
      auth/AUTHN_PROTOCOL:
        value: 'http'
      auth/AUTHN_SERVER:
        value:  '<keystone admin vip in x.x.x.x format>'
      auth/AUTHN_PORT:
        value: '35357'
      auth/AUTHN_URL:
        value: '/v2.0/tokens'
      global/WEB_SERVER:
        value: '<Contrail VIP IP ADDRESS in x.x.x.x format>'
    #end /etc/contrail/vnc_api_lib.ini
  contrail::vrouter::provision_vrouter::keystone_admin_password:
'<OVERCLOUD_ADMIN_PASSWORD from tripleo-overcloud-passwords file in
undercloud>'
    contrail::vrouter::provision_vrouter::api_address: <Contrail VIP
IP ADDRESS in x.x.x.x format>
```

- Network Templates:
    a. /home/stack/templates/nic-configs/controller.yaml:

```
heat_template_version: 2015-04-30

description: >
  Software Config to drive os-net-config to configure VLANs for the
  controller role.

parameters:
  ControlPlaneIp:
    default: ''
    description: IP address/subnet on the ctlplane network
    type: string
  ExternalIpSubnet:
    default: ''
    description: IP address/subnet on the external network
    type: string
  InternalApiIpSubnet:
    default: ''
    description: IP address/subnet on the internal API network
    type: string
  StorageIpSubnet:
    default: ''
    description: IP address/subnet on the storage network
    type: string
  StorageMgmtIpSubnet:
    default: ''
    description: IP address/subnet on the storage mgmt network
    type: string
  TenantIpSubnet:
    default: ''
    description: IP address/subnet on the tenant network
    type: string
  ManagementIpSubnet: # Only populated when including
environments/network-management.yaml
    default: ''
    description: IP address/subnet on the management network
    type: string
  ExternalNetworkVlanID:
    default: 10
    description: Vlan ID for the external network traffic.
    type: number
  InternalApiNetworkVlanID:
    default: 20
    description: Vlan ID for the internal_api network traffic.
    type: number
  StorageNetworkVlanID:
    default: 30
    description: Vlan ID for the storage network traffic.
    type: number
  StorageMgmtNetworkVlanID:
    default: 40
    description: Vlan ID for the storage mgmt network traffic.
    type: number
  TenantNetworkVlanID:
    default: 50
    description: Vlan ID for the tenant network traffic.
    type: number
  ManagementNetworkVlanID:
    default: 60
    description: Vlan ID for the management network traffic.
    type: number
  ExternalInterfaceDefaultRoute:
    default: '10.0.0.1'
    description: default route for the external network
    type: string
  ControlPlaneSubnetCidr: # Override this via parameter_defaults
    default: '24'
    description: The subnet CIDR of the control plane network.
    type: string
  DnsServers: # Override this via parameter_defaults
    default: []
    description: A list of DNS servers (2 max for some
implementations) that will be added to resolv.conf.
```

```
    type: comma_delimited_list
  EC2MetadataIp: # Override this via parameter_defaults
    description: The IP address of the EC2 metadata server.
    type: string
resources:
  OsNetConfigImpl:
    type: OS::Heat::StructuredConfig
    properties:
      group: os-apply-config
      config:
        os_net_config:
          network_config:
            -
              type: interface # physical eth0, provioning network
              name: nic2
              use_dhcp: false
              addresses:
                -
                  ip_netmask: {get_param: ExternalIpSubnet}
              routes:
                -
                  default: true
                  next_hop: {get_param: ExternalInterfaceDefaultRoute}
            -
              type: ovs_bridge
              name: {get_input: bridge_name}
              use_dhcp: false
              dns_servers: {get_param: DnsServers}
              addresses:
                -
                  ip_netmask:
                    list_join:
                      - '/'
                      - - {get_param: ControlPlaneIp}
                        - {get_param: ControlPlaneSubnetCidr}
              routes:
                -
                  ip_netmask: 169.254.169.254/32
                  next_hop: {get_param: EC2MetadataIp}
              members:
                -
                  type: interface
                  name: nic1
                  # force the MAC address of the bridge to this
interface
                  primary: true
                -
                  type: vlan
                  vlan_id: {get_param: InternalApiNetworkVlanID}
                  addresses:
                    -
                      ip_netmask: {get_param: InternalApiIpSubnet}
                -
```

```
                  type: vlan
                  vlan_id: {get_param: StorageNetworkVlanID}
                  addresses:
                    -
                      ip_netmask: {get_param: StorageIpSubnet}
                -
                  type: vlan
                  vlan_id: {get_param: StorageMgmtNetworkVlanID}
                  addresses:
                    -
                      ip_netmask: {get_param: StorageMgmtIpSubnet}
                -
                  type: vlan
                  vlan_id: {get_param: TenantNetworkVlanID}
                  addresses:
                    -
                      ip_netmask: {get_param: TenantIpSubnet}

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

b. /home/stack/templates/nic-configs/compute.yaml:

```
heat_template_version: 2015-04-30

description: >
  Software Config to drive os-net-config to configure VLANs for the
  compute role.

parameters:
  ControlPlaneIp:
    default: ''
    description: IP address/subnet on the ctlplane network
    type: string
  ExternalIpSubnet:
    default: ''
    description: IP address/subnet on the external network
    type: string
  InternalApiIpSubnet:
    default: ''
    description: IP address/subnet on the internal API network
    type: string
  StorageIpSubnet:
    default: ''
    description: IP address/subnet on the storage network
    type: string
  StorageMgmtIpSubnet:
    default: ''
    description: IP address/subnet on the storage mgmt network
    type: string
  TenantIpSubnet:
    default: ''
    description: IP address/subnet on the tenant network
    type: string
  ManagementIpSubnet: # Only populated when including
environments/network-management.yaml
    default: ''
    description: IP address/subnet on the management network
    type: string
  InternalApiNetworkVlanID:
    default: 20
    description: Vlan ID for the internal_api network traffic.
    type: number
  StorageNetworkVlanID:
    default: 30
    description: Vlan ID for the storage network traffic.
    type: number
  TenantNetworkVlanID:
    default: 50
    description: Vlan ID for the tenant network traffic.
    type: number
  ManagementNetworkVlanID:
    default: 60
    description: Vlan ID for the management network traffic.
    type: number
  ControlPlaneSubnetCidr: # Override this via parameter_defaults
    default: '24'
    description: The subnet CIDR of the control plane network.
    type: string
  ControlPlaneDefaultRoute: # Override this via parameter_defaults
    description: The default route of the control plane network.
    type: string
  DnsServers: # Override this via parameter_defaults
    default: []
    description: A list of DNS servers (2 max for some
implementations) that will be added to resolv.conf.
    type: comma_delimited_list
  EC2MetadataIp: # Override this via parameter_defaults
    description: The IP address of the EC2 metadata server.
    type: string
  ExternalNetworkVlanID:
    default: 10
    description: Vlan ID for the external network traffic.
    type: number
  ExternalInterfaceDefaultRoute:
    default: '10.0.0.1'
    description: default route for the external network
    type: string

resources:
  OsNetConfigImpl:
    type: OS::Heat::StructuredConfig
    properties:
```

```
group: os-apply-config
config:
  os_net_config:
    network_config:
      -
        type: interface # mgmt interface eth1
        name: nic2
        use_dhcp: false
        dns_servers: {get_param: DnsServers}
        addresses:
          -
            ip_netmask: {get_param: ExternalIpSubnet}
        routes:
          -
            default: true
            next_hop: {get_param: ExternalInterfaceDefaultRoute}
      -
        type: interface
        name: nic1
        use_dhcp: false
        addresses:
          -
            ip_netmask:
              list_join:
                - '/'
                - - {get_param: ControlPlaneIp}
                  - {get_param: ControlPlaneSubnetCidr}
        routes:
          -
            ip_netmask: 169.254.169.254/32
            next_hop: {get_param: EC2MetadataIp}
      -
        type: vlan
        vlan_id: {get_param: InternalApiNetworkVlanID}
        device: nic1
        addresses:
          -
            ip_netmask: {get_param: InternalApiIpSubnet}
      -
        type: vlan
        vlan_id: {get_param: StorageNetworkVlanID}
        device: nic1
        addresses:
          -
            ip_netmask: {get_param: StorageIpSubnet}
      -
        type: vlan
        vlan_id: {get_param: TenantNetworkVlanID}
        device: nic1
        addresses:
          -
            ip_netmask: {get_param: TenantIpSubnet}
```

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

- Introspect overcloud nodes with ironic
  a. Create instackenv.json for your overcloud nodes.
     In this case we used kvm virtual machines but physical nodes would have a different configuration. Please see installation documentation for examples.

```
{
  "nodes": [
    {
      "pm_user": "root",
      "arch": "x86_64",
      "name": "overcloud-node1",
      "pm_addr": "uchost",
      "pm_password": "-----BEGIN RSA PRIVATE KEY-----
-----END RSA PRIVATE KEY-----",
      "pm_type": "pxe_ssh",
      "mac": [
        "52:54:00:b1:ed:28"
      ],
      "cpu": "4",
      "memory": "8192",
      "disk": "60"
    },
    {
      "pm_user": "root",
      "arch": "x86_64",
      "name": "overcloud-node2",
      "pm_addr": "uchost",
      "pm_password": "-----BEGIN RSA PRIVATE KEY-----
-----END RSA PRIVATE KEY-----",
      "pm_type": "pxe_ssh",
      "mac": [
        "52:54:00:5e:b9:ba"
      ],
      "cpu": "4",
      "memory": "8192",
      "disk": "60"
    },
    {
      "pm_user": "root",
      "arch": "x86_64",
      "name": "overcloud-node3",
      "pm_addr": "uchost",
      "pm_password": "-----BEGIN RSA PRIVATE KEY-----
-----END RSA PRIVATE KEY-----",
      "pm_type": "pxe_ssh",
      "mac": [
        "52:54:00:95:8a:9d"
      ],
      "cpu": "4",
      "memory": "8192",
      "disk": "60"
    },
    {
      "pm_user": "root",
      "arch": "x86_64",
      "name": "overcloud-node4",
      "pm_addr": "uchost",
      "pm_password": "-----BEGIN RSA PRIVATE KEY-----
-----END RSA PRIVATE KEY-----",
      "pm_type": "pxe_ssh",
      "mac": [
        "52:54:00:95:aa:70"
      ],
      "cpu": "4",
      "memory": "8192",
      "disk": "60"
    },
    {
      "pm_user": "root",
      "arch": "x86_64",
      "name": "overcloud-node5",
      "pm_addr": "uchost",
      "pm_password": "-----BEGIN RSA PRIVATE KEY-----
-----END RSA PRIVATE KEY-----",
      "pm_type": "pxe_ssh",
      "mac": [
        "52:54:00:cf:8f:d6"
      ],
      "cpu": "4",
      "memory": "8192",
      "disk": "60"
    }
  ],
  "arch": "x86_64",
  "host-ip": "uchost",
  "power_manager":
"nova.virt.baremetal.virtual_power_driver.VirtualPowerManager",
  "ssh-key": "-----BEGIN RSA PRIVATE KEY-----
-----END RSA PRIVATE KEY-----",
  "ssh-user": "root"
}
```

b. Kick off introspection:

```
[stack@undercloud]# openstack baremetal import --json
/home/stack/instackenv.json
[stack@undercloud]# openstack baremetal configure boot
[stack@undercloud]# openstack baremetal introspection bulk start
```

c. Tag introspected nodes with their profile & boot setting:

```
[stack@undercloud]# ironic node-update <node1> add
properties/capabilities='profile:control,boot_option:local'
[stack@undercloud]# ironic node-update <node2> add
properties/capabilities='profile:control,boot_option:local'
[stack@undercloud]# ironic node-update <node3> add
properties/capabilities='profile:control,boot_option:local'
[stack@undercloud]# ironic node-update <node4> add
properties/capabilities='profile:compute,boot_option:local'
[stack@undercloud]# ironic node-update <node5> add
properties/capabilities='profile:compute,boot_option:local'
...
```

Note: Before we run the deploy script, There might be a mismatch in key on undercloud and overcloud-controllers. It is better to re-genrate default key. We might not be able to login to overcloud-controller vm from undercloud using heat-admin.

Please refer: https://access.redhat.com/solutions/2701051

Step 1 - Deploy Vanilla Overcloud Controllers

```
[stack@undercloud]# cd /home/stack
[stack@undercloud]#./deploy.sh
+ '[' /home/stack '!=' /home/stack ']'
+ control_scale=3
+ compute_scale=0
+ ceph_scale=0
+ '[' 0 -eq 3 ']'
+ echo 'control_scale=3, compute_scale=0, ceph_scale=0'
control_scale=3, compute_scale=0, ceph_scale=0
+++ dirname ./deploy.sh
++ cd .
++ pwd
+ DIR=/home/stack
+ template_base_dir=/home/stack/templates
+ ntpserver=10.84.5.100
+ openstack overcloud deploy --templates -e
/usr/share/openstack-tripleo-heat-templates/environments/network-isolation.yaml -e
/home/stack/templates/network-environment.yaml -e /home/stack/templates/nen
There are 7 ironic nodes with no profile that will not be used:
cd49c266-df4c-45ed-9489-23ee20b09a32, 7f6545d2-666e-41aa-a41e-c86cd92d88d2,
bba9cc07-d160-4351-bf3d-b558e90432a2, 04d39cae-1fc8-4a8b-83c0b
Configuration has 1 warnings, fix them before proceeding.
Deploying templates in the directory /usr/share/openstack-tripleo-heat-templates
...
...
[overcloud]: UPDATE_COMPLETE  Stack UPDATE completed successfully
Stack overcloud UPDATE_COMPLETE
Overcloud Endpoint: http://10.84.22.150:5000/v2.0
Overcloud Deployed
```

Step 2 - Configure and deploy Contrail nodes

Contrail packages are not pre installed and fabric-utils from contrail provides necessary scripts to install and provision contrail components. Below preparation would configure basic networks in the contrail nodes

Create Static eth0/eth1/eth2 interfaces in each of the contrail node.

```
cp /etc/sysconfig/network-scripts/ifcfg-eth{0,1} && sed -i
s/DEVICE=.*/DEVICE=eth1/g /etc/sysconfig/network-scripts/ifcfg-eth1'
/etc/sysconfig/network-scripts/ifcfg-eth{0,2} && sed -i s/DEVICE=.*/DEVICE=eth2/g
/etc/sysconfig/network-scripts/ifcfg-eth2'

cat << EOF > /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE="eth0"
IPADDR=<IP-ADDRESS-FROM-EXTERNAL-SUBNET>
NETMASK=<NET-MASK>
ONBOOT="yes"
TYPE="Ethernet"
PEERDNS="yes"
IPV6INIT="no"
BOOTPROTO=none
EOF

cat << EOF > /etc/sysconfig/network-scripts/ifcfg-eth1
DEVICE="eth1"
IPADDR=<IP-ADDRESS-FROM-CTLPLANE-SUBNET>
NETMASK=<NET-MASK>
ONBOOT="yes"
TYPE="Ethernet"
PEERDNS="yes"
IPV6INIT="no"
BOOTPROTO=none
EOF

cat << EOF > /etc/sysconfig/network-scripts/ifcfg-<vlan name of internal_api
network ex: vlan20>
DEVICE=<vlan name of internal_api network>
ONBOOT=yes
HOTPLUG=no
NM_CONTROLLED=no
PEERDNS=no
IPADDR=<IP-ADDRESS-FROM-INTERNAL-API-SUBNET>
```

```
NETMASK=<NETMASK>
BOOTPROTO=none
VLAN=yes
PHYSDEV=<ctrlplane interface of contrail-nodes ex: eth1>
EOF
```

Contrail Nodes Preparation

Preparation

Attach Subscription and Enable repos

Attach your redhat subscription to the hypervisor. Use list command to the list of available of pools and attach the appropriate pool ID.

```
subscription-manager register --username <username> --password <password>
--force
sudo subscription-manager list --available --all
subscription-manager attach --pool <pool-id>
```

Enable required repos in the hypervisor.

```
subscription-manager repos --enable=rhel-7-server-rpms
--enable=rhel-7-server-extras-rpms  --enable=rhel-7-server-openstack-8-rpms;

subscription-manager repos --disable=*;

subscription-manager repos --enable=rhel-7-server-rpms
--enable=rhel-7-server-extras-rpms   --enable=rhel-7-server-openstack-8-rpms
```

Install support tools.

```
yum install sos
```

Contrail dependencies shipped by Juniper

Below packages are missing in the RHEL upstream repos and will be supplied by contrail as missing-depends.tgz available at https://app.box.com/s/xgcwul1attt90zkee1iw7qjw0krme025

1. supervisor-3.0-2.el7cp.noarch.rpm
2. python-meld3-0.6.10-1.el7cp.x86_64.rpm
3. protobuf-2.5.0-8.0contrail0.el7.x86_64.rpm

4. net-snmp-python-5.7.2-24.el7_2.1.x86_64.rpm
5. yum-plugin-priorities-1.1.31-34.0contrail0.el7.noarch.rpm

Contrail-Package

Setup Procedure

1. Copy contrail-install-packages-\<release\>-\<version\>~\<sku\>.el7.noarch.rpm to the host build/ install server. Usually contrail's first server can be used as the install server

2. Install contrail-install-packages. This package contains all contrail build packages

```
yum localinstall --disablerepo=*
/path/to/contrail-install-packages-\<release\>-\<version\>~\<sku\>.el7.noarc
h.rpm
```

3. During Step #0, a repo in a local server is created to host RHEL missing packages. Please include below repo definition in the contrail nodes so they get automatically installed when required

```
# vi /etc/yum.repos.d/missing-depends.repo
[missing-depends]
name=missing-depends
baseurl=http://<IP Address of the repo>:8080/missing-depends
enabled=1
priority=1
gpgcheck=0
```

4. Cleanup yum db

```
yum clean all
```

5. Make sure the newly added contrail-rhel-missing repo is reachable. Below command should list contrail-rhel-missing repo name and its package count

```
yum repolist
```

6. Execute setup.sh. During this script, contrail's fabric-utils package which provides fabric commands and contrail-setup package which provides provisioning scripts are installed in the node. Along with this, a local repo with all contrail packages are created at /opt/contrail/contrail_install_repo director and a repo file would have been added at /etc/yum.repos.d director

```
cd /opt/contrail/contrail_packages/ && ./setup.sh
```

7. Contrail-api is tested and work with python-2.7.5-39.el7_2.x86_64. However RHEL upstream may have very latest version which may not work with contrail-api. Downgrade python to recommended version - python-2.7.5-39.el7_2.x86_64
   Note: This limitation is removed in recent builds as contrail includes python-gevent-1.1rc5 package in its distribution

```
Check the install python version in cfgm node:
rpm -qa  | grep 2.7.5 | grep python
Expected Version:
python-libs-2.7.5-39.el7_2.x86_64
python-devel-2.7.5-39.el7_2.x86_64
python-2.7.5-39.el7_2.x86_64

Newer version which might not work:
python-libs-2.7.5-48.el7.x86_64
python-2.7.5-48.el7.x86_64
python-devel-2.7.5-48.el7.x86_64

yum downgrade python-libs-2.7.5-39.el7_2.x86_64 python-2.7.5-39.el7_2.x86_64
python-devel-2.7.5-39.el7_2.x86_64
```

8. Check JAVA JRE available on the RHEL repos. Contrail is tested and works with below listed version and may not be compatible with the latest version. Please install these two packages explicitly in the Database nodes

```
java-1.7.0-openjdk-headless-1.7.0.91-2.6.2.3.el7.x86_64
java-1.7.0-openjdk-1.7.0.91-2.6.2.3.el7.x86_64
```

   Use below command to verify the list of available version of java packages

```
yum list --show-duplicate java-1.7.0-openjdk
yum list --show-duplicate java-1.7.0-openjdk-headless
```

Prepare testbed.py (touch points with Openstack)

During setup.sh execution at #6, contrail-fabric-utils package will be installed in the build server. This package creates /opt/contrail/utils directory and installs all contrail fabric utils scripts in it. Contrail fabric script uses /opt/contrail/utils/fabfile/testbeds/testbed.py file as a config file. Below is the example format of testbed.py and it needs be placed in the build server.

Note:

1. New Variable: “manage_neutron_server” in env.keystone section, which can be configured as “no” or “yes”. When configured as “no” will avoid installing neutron-server on contrail-controller nodes. This variable is effective only after contrail patches.

2. New Variable: “amqp_password” in env.cfgm section, which takes amqp_password if provisioned in the openstack-controllers for RabbitMQ. This variable is effective only after contrail patches
3. Refer variables OVERCLOUD_ADMIN_TOKEN, OVERCLOUD_NOVA_PASSWORD, OVERCLOUD_NEUTRON_PASSWORD, OVERCLOUD_ADMIN_PASSWORD, OVERCLOUD_RABBITMQ_PASSWORD from /home/stack/tripleo-overcloud-passwords in undercloud node
4. In testbed.py, Specify only contrail nodes in the variable “all” in the env.roledefs section.

```
from fabric.api import env

# Contrail Config Nodes
host1 = 'root@<IP Address of Contrail-controller1 in x.x.x.x format>'
host2 = 'root@<IP Address of Contrail-controller2 in x.x.x.x format>'
host3 = 'root@<IP Address of Contrail-controller3 in x.x.x.x format>'

# Contrail Control Nodes
host4 = 'root@<IP Address of Contrail-control1 in x.x.x.x format>'
host5 = 'root@<IP Address of Contrail-control2 in x.x.x.x format>'
host6 = 'root@<IP Address of Contrail-control3 in x.x.x.x format>'

# Compute Nodes
#host7 = 'root@10.1.1.254'
#host8 = 'root@10.1.1.253'
#host9 = 'root@10.1.1.252'

# Openstack Nodes
host10 = 'heat-admin@<IP Address of Openstack-Controller1 in x.x.x.x format>'
host11 = 'heat-admin@<IP Address of Openstack-Controller2 in x.x.x.x format>'
host12 = 'heat-admin@<IP Address of Openstack-Controller3 in x.x.x.x format>'

# Contrail WebUI
host13 = 'root@<IP Address of Contrail-Webui1 in x.x.x.x format>'

#External routers if any
#for eg.
#ext_routers = [('mx1', '10.204.216.253')]
ext_routers = []

#Autonomous system number
router_asn = 64512

#Host from which the fab commands are triggered to install and provision
host_build = host1

#Role definition of the hosts.
```

```
env.roledefs = {
        'all': [host1, host2, host3, host4, host5, host6, host13],
        'openstack': [host10, host11, host12],
        'cfgm': [host1, host2, host3],
        'control': [host4, host5, host6],
        'compute': [],
        'collector': [host1, host2, host3],
        'webui': [host13],
        'database': [host1, host2, host3],
        'build': [host_build],
}

#Hostnames
env.hostnames = {
        host1: 'contrail-config-0',
        host2: 'contrail-config-1',
        host3: 'contrail-config-2',
        host4: 'contrail-ctrl-0',
        host5: 'contrail-ctrl-1',
        host6: 'contrail-ctrl-2',
#       host7: 'cmpt-0-POP-8',
#       host8: 'cmpt-1-POP-8',
#       host9: 'cmpt-2-POP-8',
        host10: 'ctrl-0-fo-8',
        host11: 'ctrl-1-fo-8',
        host12: 'ctrl-2-fo-8',
        host13: 'contrail-ui',
}

env.passwords = {
        host1: '<password of contrail-controller1>',
        host2: '<password of contrail-controller2>',
        host3: '<password of contrail-controller3>',
        host4: '<password of contrail-control1>',
        host5: '<password of contrail-control2>',
        host6: '<password of contrail-control3>',
#       host7: 'contrail123',
#       host8: 'contrail123',
#       host9: 'contrail123',
        host10: 'SSH-KEY-SHARED',
        host11: 'SSH-KEY-SHARED',
        host12: 'SSH-KEY-SHARED',
        host13: '<password of contrail-webui1>',
        host_build: '<password of contrail-controller1>',
}
```

```
#Openstack admin password. Retrieve OVERCLOUD_ADMIN_PASSWORD from
/home/stack/tripleo-overcloud-passwords in undercloud node
env.openstack_admin_password = '<Openstack Admin Password>'

# Passwords of each host
# for passwordless login's no need to set env.passwords,
# instead populate env.key_filename in testbed.py with public key.
#env.key_filename = '/root/.ssh/id_rsa.pub'

#For reimage purpose
env.ostypes = {
      host1: 'redhat',
      host2: 'redhat',
      host3: 'redhat',
      host4: 'redhat',
      host5: 'redhat',
      host6: 'redhat',
#     host7: 'redhat',
#     host8: 'redhat',
#     host9: 'redhat',
      host10: 'redhat',
      host11: 'redhat',
      host12: 'redhat',
      host13: 'redhat',
}

minimum_diskGB = 256

#OPTIONAL BONDING CONFIGURATION
#==============================

#OPTIONAL SEPARATION OF MANAGEMENT AND CONTROL + DATA and OPTIONAL VLAN INFORMATION
#==================================================================================
control_data = {
      host1  : { 'ip': '10.4.66.15/24', 'gw' : '10.4.66.1', 'device':'vlan666' },
      host2  : { 'ip': '10.4.66.16/24', 'gw' : '10.4.66.1', 'device':'vlan666' },
      host3  : { 'ip': '10.4.66.17/24', 'gw' : '10.4.66.1', 'device':'vlan666' },
      host4  : { 'ip': '10.1.66.26/24', 'gw' : '10.1.66.1', 'device':'vlan666' },
      host5  : { 'ip': '10.1.66.27/24', 'gw' : '10.1.66.1', 'device':'vlan666' },
      host6  : { 'ip': '10.1.66.28/24', 'gw' : '10.1.66.1', 'device':'vlan666' },
      host13 : { 'ip': '10.4.66.18/24', 'gw' : '10.4.66.1', 'device':'vlan666' },
}

#To disable installing contrail interface rename package
env.interface_rename = False
```

```
#In environments where keystone is deployed outside of Contrail provisioning
#scripts , you can use the below options
#
# Note :
# "insecure" is applicable only when protocol is https
# The entries in env.keystone overrides the below options which used
# to be supported earlier :
#   service_token
#   keystone_ip
#   keystone_admin_user
#   keystone_admin_password
#   region_name
#
env.keystone = {
       'keystone_ip'      : '10.4.10.10',            # Keystone external VIP
       'auth_protocol'   : 'http',                  #Default is http
       'auth_port'       : '35357',                 #Default is 35357
       'admin_token'     : '3ne72ZEEDtuMdZGJhBHpcGF9t',  #OVERCLOUD_ADMIN_TOKEN
       'admin_user'      : 'admin',                 #Default is admin
       'admin_password'  : 'w96psg3z69t3d97azDta3KNmc',   #OVERCLOUD_ADMIN_PASSWORD

       'nova_password'   : 'NADBnaReZ7VmdYvDtjgaNxbMw', #OVERCLOUD_NOVA_PASSWORD
       'neutron_password': 'zdFvmTDc9Q3Becs64TvugEFaq', #OVERCLOUD_NEUTRON_PASSWORD
       'service_tenant'  : 'service',               # Service tenant name of
services like nova
       'admin_tenant'     : 'admin',                # Admin tenant name of
keystone admin user
       'region_name'      : 'RegionOne',            #Default is RegionOne
       'insecure'         : 'True',                 #Default = False
       'manage_neutron'  : 'no',                   #Default = 'yes' , Does
configure neutron user/role in keystone required.
       'manage_neutron_server'  : 'no',          # Avoid installing neutron-server
in contrail controller nodes
}

env.ha = {
    'contrail_internal_vip'   : '10.4.66.100',      #Internal Virtual IP of the
contrail HA Nodes.
    'contrail_external_vip'   : '10.4.10.100',      #External Virtual IP of the
contrail HA Nodes.
}


env.openstack = {
       'service_token' : '3ne72ZEEDtuMdZGJhBHpcGF9t', # OVERCLOUD_ADMIN_TOKEN
       'amqp_hosts' : '172.16.2.7',  # IP of AMQP Server in first openstack node
```

```
      'manage_amqp' : 'no',                  # Manage seperate AMQP for openstack
services in openstack nodes.
     'osapi_compute_workers' : 40,         # Default 40, For low memory system
reduce the osapi compute workers thread.
      'conductor_workers' : 40,              # Default 40, For low memory system
reduce the conductor workers thread.
}

#Config node related config knobs
#amqp_hosts : List of customer deployed AMQP servers to be used by config services.
#amqp_port : Port of the customer deployed AMQP servers.
env.cfgm = {
    'amqp_hosts' : ['172.16.2.7', '172.16.2.6', '172.16.2.8'],
    'amqp_port' : '5672',
    'amqp_password' : 'B6gW6t4BMWhWBNdX7UaFtMaAG' # OVERCLOUD_RABBITMQ_PASSWORD
}

# By default fab scripts will retrieve metadata secret from openstack node.
# To override, Specify Metadata proxy secret from Openstack node
#neutron_metadata_proxy_shared_secret = <secret>

#To enable multi-tenancy feature
multi_tenancy = True

#To enable lbaas
env.lbaas = True
```

After testbed.py is created, fabric utils commands can be used to execute given command in any nodes or depending upon the roles
Verify login credentials provided in testbed.py

```
fab all_command:"uname -r"
```

9. Set SELinux in permissive on all nodes

```
fab -R all -- "sed -i 's/SELINUX=enforcing/SELINUX=permissive/'
/etc/selinux/config"
fab -R all -- "setenforce 0"
```

Disable iptables

```
fab all_command:"iptables --flush"
fab all_command:"sudo service iptables stop; echo pass"
fab all_command:"sudo service ip6tables stop; echo pass"
fab all_command:"sudo systemctl stop firewalld; echo pass"
fab all_command:"sudo systemctl status firewalld; echo pass"
fab all_command:"sudo chkconfig firewalld off; echo pass"
fab all_command:"sudo /usr/libexec/iptables/iptables.init stop; echo pass"
fab all_command:"sudo /usr/libexec/iptables/ip6tables.init stop; echo pass"
fab all_command:"sudo service iptables save; echo pass"
fab all_command:"sudo service ip6tables save; echo pass"
```

After deployment it is possible to re-enable IPTables services in each node based in order to enable only the required flows between zones:

Port(s)	Protocol	Initiator	Target	Service
8443,8444	TCP	Control Node	Config Node	IFMAP
5998	TCP	Control Node	Config Node	Contrail Discovery
5998	TCP	Compute Node	Config Node	Contrail Discovery
8082	TCP	WebUI Node	Config Node	Contrail API
8081	TCP	WebUI Node	Config Node	Contrail Analytics API

Control nodes introspect local service is listening on 8083 TCP port.
vRouter agent introspect local service is listening on 8085 TCP port.

Install packages on Contrail nodes

Install contrail-install-packages-\<release\>-\<version\>~\<sku\>.el7.noarch.rpm located in the host_build in all the contrail nodes. Using install_pkg_all as "all" roles in testbed.py has only contrail nodes.

```
fab
install_pkg_all:/path/to/contrail-install-packages-\<release\>-\<version\>~\<sku\>.
el7.noarch.rpm
```

Contrail Patches:

Most of the manual changes done during contrail installation and provisioning are now available as patches at below location. Please apply the patches as instructed in README. These patches provide more options to fabric scripts.
https://app.box.com/s/xgcwul1attt90zkee1iw7qjw0krme025

Install Contrail packages in the contrail nodes

Note: install_vrouter argument is available only with new contrail patches

```
fab install_without_openstack:install_vrouter=no,reboot=no
```

Contrail Controller Provisioning

Note: setup_vrouter argument is available only with new contrail patches

```
fab setup_without_openstack:setup_vrouter=no,reboot=False
```

Note:
If you notice below error during contrail-api provisioning, please downgrade python-2.7.5 to python-2.7.5-39, which is a known working version explained Step 7 in Contrail Package section

```
message = "'module' object has no attribute 'sslwrap'" in
/var/log/contrail/contrail-api-0-stdout.log during “verify_cfgm” fab subtask
```

Check your EXTERNAL instance interface

FAB script misconfigured ***keepalived.conf*** external VIP instance on each Contrail Config Node. In order to correctly configure it, check and fix following properties of **EXTERNAL_INSTANCE** with right information:

```
vrrp_instance [EXTERNAL_INSTANCE_NAME] {
       interface [EXTERNAL__NW_INTERFACE]
[...]
        virtual_ipaddress {
                [EXTERNAL_VIP]/24 dev [EXTERNAL_NW_INTERFACE]
        }
[...]
        track_interface {
             [EXTERNAL_NW_INTERFACE]
        }
```

Enable keepalived at start on all contrail config

```
fab -R cfgm -- "systemctl enable keepalived"
fab -R all -- "systemctl enable ntpd && systemctl start ntpd"
```

Fix WebUI Configuration

Since there’s no internal_vip property defined in testbed.py, WebUI config.global.js file has to be fixed with the correct Openstack VIP.
From $host_build/opt/contrail/utils run below commands:

```
fab -H root@[WEBUI_NODE_1],root@[WEBUI_NODE_2], root@[...] -- "sed
's/config.computeManager.ip.*/config.computeManager.ip =
\x27[OPENSTACK_VIP]\x27;/g' /etc/contrail/config.global.js >
/etc/contrail/config.global.new | mv /etc/contrail/config.global.new
/etc/contrail/config.global.js"

fab -H root@[WEBUI_NODE_1],root@[WEBUI_NODE_2], root@[...] -- "sed
's/config.storageManager.ip.*/config.storageManager.ip =
\x27[OPENSTACK_VIP]\x27;/g' /etc/contrail/config.global.js >
/etc/contrail/config.global.new | mv /etc/contrail/config.global.new
/etc/contrail/config.global.js"

fab -H root@[WEBUI_NODE_1],root@[WEBUI_NODE_2], root@[...] -- "sed
's/config.imageManager.ip.*/config.imageManager.ip = \x27[OPENSTACK_VIP]\x27;/g'
/etc/contrail/config.global.js > /etc/contrail/config.global.new | mv
/etc/contrail/config.global.new /etc/contrail/config.global.js"

fab restart_webui
```

Once installation is complete and above ports are opened, SELinux can be in enforced on all nodes.

TIP: RH openstack-selinux package auto-configure all SELinux required rules for RabbitMQ.

Step 3 - Update Overcloud Controllers with Neutron Contrail plugin

After installing Contrail backend we need to setup OpenStack to use it.

A. Deploy update to overcloud controllers to use the neutron-contrail plugin:

```
[undercloud]#./deploy-with-contrail.sh
+ '[' /home/stack '!=' /home/stack ']'
+ control_scale=3
+ compute_scale=0
+ ceph_scale=0
+ '[' 0 -eq 3 ']'
+ echo 'control_scale=3, compute_scale=0, ceph_scale=0'
control_scale=3, compute_scale=0, ceph_scale=0
+++ dirname ./deploy-with-contrail.sh
++ cd .
++ pwd
+ DIR=/home/stack
+ template_base_dir=/home/stack/templates
+ ntpserver=10.84.5.100
+ openstack overcloud deploy --templates -e
/usr/share/openstack-tripleo-heat-templates/environments/network-isolation.y
aml -e /home/stack/templates/network-environment.yaml -e
/home/stack/templates/nen
There are 7 ironic nodes with no profile that will not be used:
cd49c266-df4c-45ed-9489-23ee20b09a32, 7f6545d2-666e-41aa-a41e-c86cd92d88d2,
bba9cc07-d160-4351-bf3d-b558e90432a2, 04d39cae-1fc8-4a8b-83c0b
Configuration has 1 warnings, fix them before proceeding.
Deploying templates in the directory
/usr/share/openstack-tripleo-heat-templates
...
...
[overcloud]: UPDATE_COMPLETE  Stack UPDATE completed successfully
Stack overcloud UPDATE_COMPLETE
Overcloud Endpoint: http://10.84.22.150:5000/v2.0
Overcloud Deployed
```

B. Check access to Contrail from Overcloud:

```
[undercloud]#source overcloudrc ; neutron net-list
+--------------------------------------+-------------------------+---------+
| id                                   | name                    | subnets |
+--------------------------------------+-------------------------+---------+
| 680354dd-30de-4feb-8952-c768f4f462c6 | ip-fabric               |         |
| d21d8a7f-4de0-4324-bb99-495283c0d578 | default-virtual-network |         |
| 0ca7f03e-4287-44f4-b555-4a36a2b35f50 | __link_local__          |         |
+--------------------------------------+-------------------------+---------+
```

You should get at least these 3 networks : default-virtual-network, ip-fabric, __link_local__

Step 4 - Deploy Compute Nodes

Now we can deploy compute nodes into the environment:

A. Edit deploy-with-contrail.sh to add compute count:

```
[undercloud]#vi deploy-with-contrail.sh
…
control_scale=3
compute_scale=1
ceph_scale=0
...
```

B. Deploy:

```
[undercloud]#./deploy-with-contrail.sh
+ '[' /home/stack '!=' /home/stack ']'
+ control_scale=3
+ compute_scale=1
+ ceph_scale=0
+ '[' 0 -eq 3 ']'
+ echo 'control_scale=3, compute_scale=1, ceph_scale=0'
control_scale=3, compute_scale=1, ceph_scale=0
+++ dirname ./deploy-with-contrail.sh
++ cd .
++ pwd
+ DIR=/home/stack
+ template_base_dir=/home/stack/templates
+ ntpserver=10.84.5.100
+ openstack overcloud deploy --templates -e
/usr/share/openstack-tripleo-heat-templates/environments/network-isolation.y
aml -e /home/stack/templates/network-environment.yaml -e
/home/stack/templates/nen
There are 7 ironic nodes with no profile that will not be used:
cd49c266-df4c-45ed-9489-23ee20b09a32, 7f6545d2-666e-41aa-a41e-c86cd92d88d2,
bba9cc07-d160-4351-bf3d-b558e90432a2, 04d39cae-1fc8-4a8b-83c0b
Configuration has 1 warnings, fix them before proceeding.
Deploying templates in the directory
/usr/share/openstack-tripleo-heat-templates
...
...
[overcloud]: UPDATE_COMPLETE  Stack UPDATE completed successfully
Stack overcloud UPDATE_COMPLETE
Overcloud Endpoint: http://10.84.22.150:5000/v2.0
Overcloud Deployed
```

C. Disable selinux on compute nodes:

```
[compute]#setenforce 0
#vi /etc/sysconfig/selinux
...
SELINUX=permissive
```

```
...
[compute]#getenforce
Disabled
```

D. Confirm Contrail status on computes:

```
[compute]# contrail-status
== Contrail vRouter ==
supervisor-vrouter:           active
contrail-vrouter-agent        active
contrail-vrouter-nodemgr      active
```

Comments & Recommendations

- Improvements were made to openstack-puppet-modules to handle lifecycle management of vRouter and neutron-plugin-contrail and can be found here: https://github.com/redhat-cip/puppet-contrail/pull/20/
  These are being QA'd and going through the process to get included in a future opm update.
- Current Contrail requires selinux to be disabled, this should no longer be required with Contrail 3.0.3.
- Please note that fencing is required for production environments. Please configure fencing as described in the OSPd documentation here: https://access.redhat.com/documentation/en/red-hat-openstack-platform/8/single/director-installation-and-usage/#sect-Fencing_the_Controller_Nodes

Updating from non-lifecycle document version

An update from the previous version of this doc would roughly include the following steps. **Please thoroughly test this in your environment as not all setups will be the same and variations may apply**. To clarify, this is updating your entire stack to newer rpm versions that were tested with the fixes. Please read the Red Hat OpenStack upgrade doc for more details: https://access.redhat.com/documentation/en/red-hat-openstack-platform/8/paged/upgrading-red-hat-openstack-platform/chapter-2-director-based-environments-performing-updates-to-minor-versions

- **This process can disrupt network and other services during the update**.
- Make sure you use the new configuration yaml files.
- Make sure to make any required new customizations to your overcloud images in glance. This can include adding/removing contrail rpms, etc.
- Remember to go back and disable selinux on your overcloud nodes.

Bugs & Known Issues

RH bugs:

1. https://bugzilla.redhat.com/show_bug.cgi?id=1418941

Contrail Bugs:

1. https://bugs.launchpad.net/juniperopenstack/+bug/1663359
2. Restarting a cassandra node corrupts cassandra and the process doesn't start. Workaround is to not restart the node.

https://bugs.launchpad.net/juniperopenstack/+bug/1669945

3.