

Contrail Cloud Deployment Guide

Published
2023-05-25

RELEASE
Release 16.2

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Juniper Networks, the Juniper Networks logo, Juniper, and Junos are registered trademarks of Juniper Networks, Inc. in the United States and other countries. All other trademarks, service marks, registered marks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Contrail Cloud Deployment Guide

Release 16.2

Copyright © 2023 Juniper Networks, Inc. All rights reserved.

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at <https://support.juniper.net/support/eula/>. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

About This Guide | iv

1

Deploy Contrail Cloud

Contrail Cloud Overview | 2

Deploy Contrail Cloud | 9

Prerequisites for Contrail Cloud Deployment | 10

Deployment Sequence for Contrail Cloud Deployment | 11

Install Contrail Cloud Installer on the Jump Host | 12

Prepare the Configuration Files | 17

Add Nodes to the Openstack Ironic Inventory | 20

Deploy Control Hosts | 21

Create VMs for all Control Roles | 21

Assign Compute Nodes | 22

Assign Storage Nodes | 22

Deploy the Kubernetes Cluster | 22

Deploy the OpenStack Cluster | 22

Deploy the Contrail Cloud Control Plane | 23

Validate the OpenStack Environment | 23

Install VNF Images and Templates | 24

Add New Compute and Storage Nodes | 24

Gather Logs | 24

Appendix A: Sample Configuration Files | 25

Appendix B: Remove a Ceph Storage Node | 87

About This Guide

Use this guide to deploy Contrail Cloud. The appendices include additional sample configuration files and procedures.

1

CHAPTER

Deploy Contrail Cloud

[Contrail Cloud Overview](#) | 2

[Deploy Contrail Cloud](#) | 9

[Appendix A: Sample Configuration Files](#) | 25

[Appendix B: Remove a Ceph Storage Node](#) | 87

Contrail Cloud Overview

IN THIS SECTION

- [Contrail Cloud Node Types | 3](#)
- [Contrail Cloud Directory Structure | 4](#)
- [Contrail Cloud Configuration Files | 6](#)
- [Navigating Contrail Cloud | 7](#)

Juniper Networks® Contrail Cloud provides cloud service providers with a bundled solution to build cloud platform infrastructures. Contrail Cloud simplifies your network by integrating multiple software components into a bundle with pre-configured files that you can install on multiple devices using a single installation procedure. The bundle includes:

- Red Hat OpenStack
- Red Hat Ceph Storage
- Juniper Networks Contrail Networking
- Juniper Networks Contrail Insights (formerly known as AppFormix)
- Pre-configured YAML files to simplify the initial configuration procedure, and Ansible scripts to deploy configurations made in the YAML files.

A Contrail Cloud environment includes servers that function as non-blocking compute nodes, storage nodes, and a variety of management nodes that are used to manage components for the compute and storage nodes and these nodes are interconnected to one another and connect to the fabric to connect to the larger network.

The following sections define the key component of Contrail Cloud and how to deploy it. For a detailed understanding of Contrail Cloud beyond the scope of the basic deployment, see the [Contrail Cloud Reference Architecture Guide](#).

Contrail Cloud Node Types

Table 1 on page 3 describes the types of nodes used in Contrail Cloud. Understanding the different node types will help you configure the YAML files for deployment.

Table 1: Contrail Cloud Node Types

Node Type	Definition
jump host (also known as the Contrail Cloud host)	Physical hypervisor from which all Contrail Cloud deployment scripts are run.
undercloud	Provisions and manages all nodes in the overcloud (controller, compute, and storage nodes). The undercloud runs as a virtual machine (VM) on the jump host.
RHV Manager	Controller for the control host hypervisors that resides as a VM on the jump host. It is the core component of Red Hat's server virtualization platform and manages the control hosts and their resources.
overcloud	Includes the resources for the control plane, baremetal storage nodes, and baremetal compute nodes as deployed by the undercloud. Considered the production OpenStack Cloud.
control host	Baremetal (physical) server on which one or more control VMs run on. The servers host all OpenStack, Contrail Networking, and Contrail Insights control functions.

Table 1: Contrail Cloud Node Types *(Continued)*

Node Type	Definition
control VMs	<p>VMs that run on one or more control hosts. The control VMs include:</p> <ul style="list-style-type: none"> • OpenStack controller • Contrail Insights controller • Kubernetes (also known as K8s) <p>The K8s control hosts include:</p> <ul style="list-style-type: none"> • Contrail controller • Contrail analytics • Contrail analytics database • Contrail Web UI
compute nodes	<p>Baremetal servers on which tenant VMs run. The three types of compute nodes are:</p> <ul style="list-style-type: none"> • Kernel • DPDK (data plane development kit): • Single Root I/O Virtualization and Sharing (SR-IOV) ports
storage nodes	<p>Baremetal servers loaded with storage drives for running Ceph storage software (optional).</p>

Contrail Cloud Directory Structure

[Table 2 on page 5](#) describes the file directory structure and YAML file parameters used in Contrail Cloud.

Contrail Cloud is installed on the jump host and resides in the `/var/lib/contrail_cloud/` directory.

Table 2: Contrail Cloud Directory Structure

Directory	Description
ansible	Contains the configuration management automation for Contrail Cloud.
appformix	<p>Holds the Contrail Insights license and any user-defined Contrail Insights plugins.</p> <p>NOTE: AppFormix was renamed to Contrail Insights, however the directory name is still called appformix.</p>
auth	Stores the overcloudrc (overcloud) and stackrc (undercloud) credential files.
certs	Holds the SSL certificates for Contrail and OpenStack.
config	Contains all the configuration files for deploying Contrail Cloud. The configuration files are in the YAML file format.
facts.d	Contains information about the Contrail Cloud environment and holds the state across Ansible invocations to allow the playbooks to remember important deployment information.
introspection	Contains detailed information about the hardware that is used and the hardware configuration. You can use this information when planning your overall Contrail Cloud deployment and to verify that NICs are assigned properly, along with VLANs and bonds.
samples	Contains sample snippets of the various YAML configuration files that you can reference for deployment. Each snippet details a specific use case configuration. This directory also contains information about the various knobs and configuration options available and is separated by features and scaling use cases.
scripts	Contains various required and optional scripts used with Contrail Cloud, including pre-deployment verification scripts, deployment scripts, introspection and post-deployment verification tools.
tmp	Holds the temporary files required by Contrail Cloud.

Contrail Cloud Configuration Files

Table 3 on page 6 describes the configuration files you use in your Contrail Cloud deployment. All configuration files are in the YAML file format and use the Jinja2 template. The files reside in the `/var/lib/contrail_cloud/config` directory. For specific configuration examples, see the `/var/lib/contrail_cloud/samples` directory on the jump host.

Table 3: Contrail Cloud Configuration Files

YAML configuration files	Description
site.yml	Contains information about the specific undercloud and overcloud instance.
inventory.yml	Configuration file for baremetal (physical) server configuration, such as IPMI and UEFI hardware standards.
overcloud_nics.yml	Use to configure network options for the overcloud nodes.
control-host-nodes.yml	Use to configure the network layout for the control host server(s).
k8s-host-nodes.yml	Use to configure the network layout for the Kubernetes host nodes.
compute-nodes.yml	Use to assign naming and role mapping for all compute nodes.
storage-nodes.yml	Defines the role mapping of the storage nodes.
vault-data.yml	Encrypted file that holds all sensitive user data, such as passwords, product keys and user data.

Contrail Cloud also provides default values for all Contrail Cloud playbooks. The default values reside in the **default.yml** directory (`/var/lib/contrail_cloud/ansible/playbooks/default.yml`). If you are an advanced user, you can use this file to look for values you may want to override in your **config/site.yml** configuration file.

NOTE: Never modify the **default.yml** file directly. Always apply the customized values from your **config/site.yml** configuration file.

Navigating Contrail Cloud

Table 4 on page 7 describes how to navigate Contrail Cloud using the CLI and Web UI.

Table 4: Navigating Contrail Cloud

Service	From	Cloud Endpoint	Description
SSH	your computer	jumphost as contrail user	The default password is c0ntrail123.
SSH	jumphost as contrail user	undercloud as stack user	SSH keys provide password-less access.
SSH	jumphost as contrail user	control hosts as contrail user	SSH keys provide password-less access.
SSH	jumphost as contrail user	K8s hosts as contrail user	SSH keys provide password-less access.
SSH	jumphost as contrail user	RHV Manager as contrail user	SSH keys provide password-less access.
SSH	undercloud as stack user	overcloud hosts as heat-admin	SSH keys provide password-less access.
RHV Manager Web UI	your computer	https://<jumphost> as admin user	The default password is c0ntrail123.
kubectl	—	k8s hosts as root user	Used to manage Kubernetes and pods from k8s hosts.

Table 4: Navigating Contrail Cloud *(Continued)*

Service	From	Cloud Endpoint	Description
crictl	k8s hosts	k8s hosts as root user	Used to manage images and containers from k8s hosts. (see Debugging Kubernetes nodes with crictl)
undercloud OpenStack CLI	—	undercloud as stack user	source /home/stack/stackrc Used for RHOSP director OpenStack services.
overcloud OpenStack CLI	—	undercloud as stack user	source /home/stack/overcloudrc Used for RHOSP overcloud OpenStack services.
overcloud Contrail Web UI	your computer	https://<contrail VIP>:8443 with admin user	Web browser access to the Contrail UI. The default password is c0ntrail123.
overcloud Contrail Command Web UI	your computer	https://<contrail VIP>:9443 with admin user	Web browser access to the Contrail UI. The default password is c0ntrail123. NOTE: Not deployed by default.
overcloud Horizon Web UI	your computer	https://<overcloud VIP> with admin user	Web browser access to the Openstack Horizon UI. The default password is c0ntrail123.

Table 4: Navigating Contrail Cloud *(Continued)*

Service	From	Cloud Endpoint	Description
overcloud HAProxy stats	your computer	https://<control plane>:1993 with admin user	HAProxy traffic stats exposed on each OpenStack control node's control plane interface. The password is autogenerated. See /var/lib/config-data/puppet-generated/haproxy/etc/haproxy/haproxy.cfg on the control nodes).
Contrail Insights Web UI	your computer	https://<appformix vip>:9000 with admin user	Web browser access to the Contrail Insights UI. The default password is c0ntrail123.

RELATED DOCUMENTATION

[Contrail Cloud Documentation](#)

[Contrail Cloud Reference Architecture Guide](#)

[Product Documentation for Kubernetes](#)

[Product Documentation for Red Hat OpenStack Platform 16.1](#)

Deploy Contrail Cloud

IN THIS SECTION

- [Prerequisites for Contrail Cloud Deployment | 10](#)
- [Deployment Sequence for Contrail Cloud Deployment | 11](#)

Prerequisites for Contrail Cloud Deployment

Before you deploy Contrail Cloud, ensure that your system meets the following prerequisites:

Infrastructure Networking

- Every system must have access to the Contrail Cloud repository satellite. The satellite is used to distribute packages and control software versions.
- The Contrail Cloud jump host must have access to the Intelligent Platform Management Interface (IPMI) of every managed server.
- The jump host must be in the same broadcast domain as each managed server's management interface to allow Preboot Execution Environment (PXE) booting.

NOTE: When running multiple networks that use different switching devices per rack, PXE booting is accomplished by stretching a VLAN across the interfaces. BOOTP forwarding in the network fabric is not supported. The undercloud is the only DHCP server in this network.

- You must set the jump host hostname to a long name (FQDN).
 - You must also set proper `/etc/hosts` entry on the jump host for the given FQDN.
- The jump host FQDN should also be resolvable by DNS, returning an IP that is reachable/routable from the whole cloud environment.

Contrail Cloud Jump Host Setup

The Red Hat Openstack Platform Director (also known as the undercloud) is deployed as a virtual machine on a Linux kernel-based virtual machine (KVM) Contrail Cloud jump host. You must ensure that the KVM host OS:

- Runs Red Hat Enterprise Linux 8.2 or earlier with only base packages installed. Contrail Cloud installs RHEL 8.2 and all necessary packages as part of the installation process.
- Is not running other virtual machines.
- Has a network connection that can reach the Contrail Cloud Repository Satellite and has IPMI access to physical hardware.
- Has a network connection that can be used for provisioning other infrastructure resources.
- Has at least 500 GB space in the `/var` directory to host virtual machines, packages, and images.
- Has at least 40 GB RAM and 24 vCPUs.

- Supports users such as a root user with password-less sudo permissions.
- Provides password-less SSH access in loopback for users with sudo permissions.
- Resolves Internet and satellite sites with DNS.
- Has the time synchronized with an NTP source.

Deployment Sequence for Contrail Cloud Deployment

IN THIS SECTION

- Install Contrail Cloud Installer on the Jump Host | 12
- Prepare the Configuration Files | 17
- Add Nodes to the Openstack Ironic Inventory | 20
- Deploy Control Hosts | 21
- Create VMs for all Control Roles | 21
- Assign Compute Nodes | 22
- Assign Storage Nodes | 22
- Deploy the Kubernetes Cluster | 22
- Deploy the OpenStack Cluster | 22
- Deploy the Contrail Cloud Control Plane | 23
- Validate the OpenStack Environment | 23
- Install VNF Images and Templates | 24
- Add New Compute and Storage Nodes | 24
- Gather Logs | 24

The following deployment sequence describes how to install, configure, and deploy Contrail Cloud.

NOTE: If you encounter an error in any step in the sequence, you can undo the step with the clean up feature. You reverse the installation sequence by running each script (using the “-c”

argument) to get back to the desired state in the sequence. For example, to redeploy the Contrail Cloud and OpenStack clusters:

```
/var/lib/contrail_cloud/scripts/k8s-tf-operator-deploy.sh -c
/var/lib/contrail_cloud/scripts/openstack-deploy.sh -c
/var/lib/contrail_cloud/scripts/openstack-deploy.sh
/var/lib/contrail_cloud/scripts/k8s-tf-operator-deploy.sh
```

When you clean the **k8s-tf-operator-deploy.sh** script, you must also clean the **openstack-deploy.sh** script and then deploy both to ensure that each has a consistent state.

Install Contrail Cloud Installer on the Jump Host

The jump host is the Contrail Cloud host and is the starting point for deploying Contrail Cloud. Before you begin the installation, do the following:

1. Send a request to mailto:contrail_cloud_subscriptions@juniper.net to obtain the activation keys for Contrail Cloud. You will receive an email containing:
 - A unique satellite activation key
 - The satellite DNS name
 - The satellite organization

NOTE: Contrail Cloud Satellite is the repository that contains the bundle for Contrail Cloud.

2. Create new SSH keys. Verify that the root user has SSH keys before performing the installation.

```
yes '' | sudo ssh-keygen -t rsa -N ''
```

3. Create a passphrase-protected key.

```
sudo ssh-keygen -t rsa
```


If a passphrase is set on the SSH key, you can use the `ssh-agent` to cache the passphrase. For example, as the `contrail` user on the jump host:

```
ssh-agent bash
ssh-add <key_path>    (default: /home/contrail/.ssh/id_rsa)
```

4. Ensure that the root user can connect through SSH to the localhost without a password. To authorize access, a password might be required the first time.

```
sudo ssh-copy-id localhost
```

Install Contrail Cloud

1. Untar the **contrail_cloud_installer.sh** on the jump host.

You can download the installer at: [Juniper Networks Contrail Cloud Download Site](#).

2. Specify the activation key by setting the environment variables.

For example:

```
SATELLITE="contrail-cloud-satellite.juniper.net"
SATELLITE_KEY="ak-my-account-key"
SATELLITE_ORG="ContrailCloud"
```

3. Verify that the installer script has the required permissions to install the packages. The packages are installed in the **/var/lib/contrail_cloud** directory.

```
./contrail_cloud_installer.sh \
--satellite_host ${SATELLITE} \
--satellite_key ${SATELLITE_KEY} \
--satellite_org ${SATELLITE_ORG}
```

4. Define site-specific information in the Ansible variables:
 - a. Change the directory to **/var/lib/contrail_cloud/config**.
 - b. Copy the sample **/var/lib/contrail_cloud/samples/*.yaml** configuration files to the **/var/lib/contrail_cloud/config** directory.

NOTE: You can skip this step if you have existing configuration files in the **config** directory.

- c. Add your activation key (satellite organization and FQDN) to the **site.yml** configuration file.

```
global:
  rhel:
    # Contrail Cloud Activation Key
    # These details are provided when you request an activation key from
    # contrail cloud subscriptions <contrail_cloud_subscriptions@juniper.net>
    #
  satellite:
    #SATELLITE_KEY should be defined in vault-data.yml file
    #SATELLITE_ORG
    organization: "ContrailCloud"
    #SATELLITE_FQDN
    fqdn: contrail-cloud-satellite.juniper.net
```

- d. Customize the **site.yml** configuration file (**/var/lib/contrail_cloud/config/site.yml**) with site-specific settings for your environment. Ensure that the following fields are changed for each site:

```
global:
  # List of DNS nameservers
  dns:
    # Google Public DNS
    - "8.8.8.8"
    - "8.8.4.4"
  # List of NTP time servers
  ntp:
    # public pool.ntp.org
    - "0.pool.ntp.org"
    - "1.pool.ntp.org"
    - "2.pool.ntp.org"
    - "3.pool.ntp.org"
  # Timezone for all servers
  timezone: 'America/Los_Angeles'
  rhel:
    # Contrail Cloud Activation Key
    # These details are provided when you request an activation key from
```

```
# contrail cloud subscriptions <contrail_cloud_subscriptions@juniper.net>
#
satellite:
  #SATELLITE_KEY should be defined in vault-data.yml file
  #SATELLITE_ORG
  organization: "ContrailCloud16"
  #SATELLITE_FQDN
  fqdn: contrail-cloud-satellite.juniper.net
# DNS domain information.
# Must be unique for every deployment to avoid name conflicts.
# Need not be a registered DNS domain.
domain: "my.unique.domain"

jumphost:
  network:
    # network used for provisioning (PXE booting) servers
  provision:
    # jumphost nic to be used for provisioning (PXE booting) servers
    nic: eno1
```

NOTE: If you are deploying DPDK on an Intel X710 NIC, set the DPDK driver to `vfio-pci` in the `site.yml` configuration file as follows:

```
overcloud:
  contrail:
    vrouter:
      dpdk:
        driver: "vfio-pci"
```

For a complete matrix of supported NIC and driver mapping, see the [Contrail Networking NIC Support Matrix](#).

5. Prepare the Ansible Vault. The vault allows you to keep sensitive data such as passwords or keys in encrypted files, rather than as plaintext in playbooks or roles.

- a. Customize the `vault-data.yml` configuration file:

```
ansible-vault edit /var/lib/contrail_cloud/config/vault-data.yml
```

- b. Change the password for the vault-encrypted file. The default password is c0ntrail123.

```
ansible-vault rekey /var/lib/contrail_cloud/config/vault-data.yml
```

NOTE: Use a plain-text password to create the vault password. Using a plain-text password prevents Ansible from asking you for a password every time. When creating a the file for the contrail user, make sure it is read-only. We recommend that you delete the file after the deployment completes.

```
sudo chown contrail /var/lib/contrail_cloud/config/.vault_password
sudo chmod 0400 /var/lib/contrail_cloud/config/.vault_password
```

6. Add the satellite key to the vault by using the **ansible-vault edit config/vault-data.yml** command.

```
vault:
  global:
    rhel:
      # Contrail Cloud Activation Key
      satellite:
        #SATELLITE_KEY
        key: "ak-my-account-key"
```

7. Run the Ansible provisioning. The provisioning includes setting up the jump host, RHV Manager, and the undercloud (RHOSPd).

- a. Make sure that you can establish an SSH connection without specifying a password:

```
sudo ssh localhost true
```

- b. Install the automation scripts:

```
sudo /var/lib/contrail_cloud/scripts/install_contrail_cloud_manager.sh
```

When the provisioning is finished, a new user with the username `contrail` is created on the jump host and a new set of SSH keys is generated that gives the user access to the undercloud VM and the control hosts. The overcloud nodes, including the Contrail Insights nodes, are accessible by the `heat-admin` user and use a separate pair of keys stored on the undercloud VM, by default.

Make sure that you change the default password in your **vault-data.yml** file as discussed above.

Contrail Cloud adds entries to the `/home/contrail/.ssh/config` directory that includes the username used for each of the overcloud nodes (and the undercloud). This means you can use `ssh undercloud` or `ssh <address>` without specifying a user.

You can authorize the `contrail` user keys for a `heat-admin` by defining them in the **site.yml** configuration file:

```
global:
  service_user:
    use_ssh_key_in_overcloud: true
```

Use the `contrail` user to run all subsequent operations in Contrail Cloud from the `/var/lib/contrail_cloud/scripts` directory:

```
su - contrail
cd /var/lib/contrail_cloud/scripts
```

NOTE: The `contrail` user's SSH keys are authorized by the root. This means that the `contrail` user can SSH to root on the jump host.

Prepare the Configuration Files

[Table 5 on page 18](#) describes the configuration files that you use in Contrail Cloud. See ["Appendix A" on page 25](#) for the corresponding sample YAML files.

You can validate your configuration files at any time by running the `/var/lib/contrail_cloud/scripts/node-configuration.py` script. This script loads all the configuration files, checks the syntax, and verifies that the structures and values conform to the schema. You can use different arguments with the Python script depending on the results you are looking for.

Secured registry is used as of Contrail Cloud Release 16.3. You must provide your container image registry credentials in your `vault-data.yml` file. From the jump host, see `/var/lib/contrail_cloud/samples/unencrypted-vault-data.yml` for more information and vault data example. Always update your `vault-data.yml` file with your most recent credentials before performing any deployment activities.

NOTE: You can copy the sample files from the `/var/lib/contrail_cloud/samples/` directory on your jump host.

Table 5: Contrail Cloud Configuration Files

Configuration Settings	Filename and Location	Description
Site settings	<ul style="list-style-type: none"> • site.yml (/var/lib/contrail_cloud/config/site.yml) • sample file: <i>sample site.yml</i> 	Defines the properties for your deployment environment. The properties in this file are unique for every deployment and need to be customized.
Inventory settings	<ul style="list-style-type: none"> • inventory.yml (/var/lib/contrail_cloud/config/inventory.yml) • sample file: <i>sample inventory.yml</i> 	Defines all servers used by Contrail Cloud.

Table 5: Contrail Cloud Configuration Files *(Continued)*

Configuration Settings	Filename and Location	Description
Control hosts settings	<ul style="list-style-type: none"> control-host-nodes.yml (/var/lib/contrail_cloud/ config/ control-host-nodes.yml) sample file: <i>sample control-host-nodes.yml</i> <p>To ensure high availability of the control functions, you must define the three control hosts in your configuration and also in the inventory.yml file.</p>	<p>Defines the server and network properties for each control host. Runs virtual machines for all Contrail Cloud control functions. The VMs created on the control hosts include:</p> <ul style="list-style-type: none"> OpenStack and Ceph Controller K8s hosts nodes Contrail Insights Controller <p>To host the control VMs, the control host must meet the following minimum specifications:</p> <ul style="list-style-type: none"> 256 GB RAM Minimum 100 GB first disk for the operating system Minimum 1 TB hard disk for VM storage (multiple SSDs with RAID is recommended) Hardware RAID controller set to the right RAID level for your operating environment. The operating environment includes: operating system disk, VM storage, and VM journals.
Kubernetes host settings	<ul style="list-style-type: none"> k8s-host-nodes.yml (/var/lib/contrail_cloud /config/k8s-host-nodes.yml) sample file: <i>sample k8s-host-nodes.yml</i> 	Defines the Kubernetes VMs host nodes.

Table 5: Contrail Cloud Configuration Files *(Continued)*

Configuration Settings	Filename and Location	Description
Overcloud network settings	<ul style="list-style-type: none"> overcloud-nics.yml (/var/lib/contrail_cloud/ config/ overcloud-nics.yml) sample file: <i>sample overcloud-nics.yml</i> 	Roles that are deployed to the OpenStack and Contrail Insights VMs. Defines the network layout for each role.
Compute node settings	<ul style="list-style-type: none"> compute-nodes.yml (/var/lib/contrail_cloud/ config/ compute-nodes.yml) sample file: <i>sample compute-nodes.yml</i> 	<p>Defines the compute resources and host aggregates. You also manage host aggregates and match them with availability zones in this file.</p> <p>You must also define the compute nodes in the <code>inventory.yml</code> configuration file.</p>
Storage node settings	<ul style="list-style-type: none"> storage-nodes.yml (/var/lib/contrail_cloud/ config/ storage-nodes.yml) sample file: <i>sample storage-nodes.yml</i> 	<p>Defines the storage nodes that run Ceph storage services.</p> <p>You must define a minimum of three storage hosts to ensure high availability of the storage functions. You must also define the storage nodes in the <code>inventory.yml</code> configuration file.</p>
Vault data settings	<ul style="list-style-type: none"> vault-data.yml (/var/lib/contrail_cloud/ config/ vault-data.yml) sample file: <i>sample vault-data.yml</i> 	Encrypted file that holds all sensitive user data, such as passwords, product keys, user data and secured registry information.

Add Nodes to the Openstack Ironic Inventory

The `/var/lib/contrail_cloud/scripts/inventory-assign.sh` script adds all nodes you define in the `inventory.yml` file to the ironic inventory. The nodes added to the ironic inventory are managed by Contrail Cloud.

To add nodes to the ironic inventory:

1. Log in to the jump host with the username `contrail` and password `c0ntrail123`.
2. Run the **inventory-assign.sh** script.

```
/var/lib/contrail_cloud/scripts/inventory-assign.sh
```

3. Generate a report of the available resource properties.

These details are helpful when configuring roles, disk devices, and network interfaces. Nodes must be loaded into the Ironic inventory before running the **node-configuration.py** script. This script is also used to validate configurations against the schema, and can be used after editing any of the configuration files. The report can be generated by running:

```
/var/lib/contrail_cloud/scripts/node-configuration.py group.
```

You can generate more detailed reports for a specific resource (where `<resource>` is the inventory resource name) as follows:

```
/var/lib/contrail_cloud/scripts/node-configuration.py show -i /var/lib/contrail_cloud/
introspection/<resource>.introspection
```

Deploy Control Hosts

A control host is a hypervisor running on a server that hosts virtualized control functions as controller nodes. Controller nodes are VMs responsible for managing server functions. The **control-hosts-deploy.sh** script assigns all nodes that are defined in the `/var/lib/contrail_cloud/config/control-host-nodes.yml` file as control hosts. The hosts are imaged, booted, configured, and prepared to host the overcloud control plane VMs.

To deploy control host roles to the inventory:

1. Log in to the jump host with the username `contrail` and password `c0ntrail123`.
2. Run the **control-hosts-deploy.sh** script.

```
/var/lib/contrail_cloud/scripts/control-hosts-deploy.sh.
```

Create VMs for all Control Roles

The **control-vms-deploy.sh** script imports VM details into the ironic inventory.

To create VMs for control roles:

1. Log in to the jump host with the username `contrail` and password `c0ntrail123`.
2. Run the **control-vms-deploy.sh** script.

```
/var/lib/contrail_cloud/scripts/control-vms-deploy.sh.
```

Assign Compute Nodes

A compute node is a server that hosts virtual machines that provide services over the network. The **compute-nodes-assign.sh** script assigns the Nova compute role for all nodes that you define in the **compute-nodes.yml** configuration file (`/var/lib/contrail_cloud/config/compute-nodes.yml`).

To assign compute nodes:

1. Log in to the jump host with the username `contrail` and password `c0ntrail123`.
2. Run the **compute-nodes-assign.sh** script.

```
/var/lib/contrail_cloud/scripts/compute-nodes-assign.sh
```

Assign Storage Nodes

Storage nodes are servers whose purpose is storing data. Storage nodes run Red Hat Ceph storage software in Contrail Cloud. The **storage-nodes-assign.yml** playbook assigns the Ceph storage role for all nodes that are defined in the **storage-nodes.yml** (`/var/lib/contrail_cloud/config/sorage-nodes.yml`) file.

To assign storage nodes:

1. Log in to the jump host with the username `contrail` and password `c0ntrail123`.
2. Run the **storage-nodes-assign.sh** script.

```
/var/lib/contrail_cloud/scripts/storage-nodes-assign.sh
```

Deploy the Kubernetes Cluster

The Kubernetes (k8s) cluster is used to provide the infrastructure for the Contrail control plane, which is deployed adjacently (i.e. outside of) the overcloud OpenStack cluster. The **k8s-cluster-deploy.sh** script initiates the deployment of the Kubernetes cluster.

To deploy the Kubernetes cluster:

1. Log in to the Contrail Cloud jump host with the username `contrail` and password `c0ntrail123`.
2. Run the **k8s-cluster-deploy.sh** script.

```
/var/lib/contrail_cloud/scripts/k8s-cluster-deploy.sh.
```

Deploy the OpenStack Cluster

The **openstack-deploy.sh** script deploys the OpenStack overcloud with all control functions and all compute and storage resources that were defined in the previous playbooks.

To deploy the OpenStack cluster:

1. Log in to the Contrail Cloud jump host with the username `contrail` and password `c0ntrail123`.
2. Run the **validate-node.sh** script to verify that the environment is set correctly.

Run the script on the jump host in `/var/lib/contrail_cloud/scripts` to validate the YAML configuration files for:

- Network for OpenStack Controllers
- Networking for controller hosts and compute hosts
- Disk resource and configuration validation

3. Run the **openstack-deploy.sh** script.

`/var/lib/contrail_cloud/scripts/openstack-deploy.sh.`

Deploy the Contrail Cloud Control Plane

The **k8s-tf-operator-deploy.sh** script deploys the services that make up the Contrail control plane into Kubernetes.

NOTE: This deployment is best done in parallel with the `openstack-deploy.sh` to allow both clusters to more efficiently synchronize with each other.

To deploy the control plane:

1. Log in to the Contrail Cloud jump host with the username `contrail` and password `c0ntrail123`.
2. Run the **k8s-tf-operator-deploy.sh** script.

`/var/lib/contrail_cloud/scripts/k8s-tf-operator-deploy.sh.`

Validate the OpenStack Environment

You can validate and check that the environment By default, tests that require floating IPs (FIPs) are skipped. You can execute the `provision-sdn-gateway.sh` script before validation to provision SDN gateways and external network that can be used by Tempest. Tempest is set of integration tests to be run against a live OpenStack cluster. You can find examples of the object definitions in the **site.yml** file (`/var/lib/contrail_cloud/samples/features/provision-sdn-gateway/site.yml`).

Use the **overcloud-validation.sh** script to run Tempest test collections in newly deployed environments. The script downloads a CirrOS VM image, uploads it to the overcloud, and creates new flavors. After the script execution, results of the test can be found in the undercloud home directory where two files are created:

- **tempest-subunit-smoke.xml**
- **tempest-subunit-full.xml**

The first line of the file shows the number of failures and the total count of conducted tests.

Install VNF Images and Templates

You can use Horizon or OpenStack command-line clients to install Glance images and Heat templates for the VNF services.

Add New Compute and Storage Nodes

To add new compute and storage nodes to an existing environment:

1. Update the **inventory.yml** configuration file and run the **inventory-assign.sh** script.
2. Update the **compute-nodes.yml** configuration file with the new nodes, and run the **compute-nodes-assign.sh** script.
3. Update the **storage-nodes.yml** configuration file with the new nodes, and run the **storage-nodes-assign.sh** script.
4. Rerun the **openstack-deploy.sh** script.

Gather Logs

You can run a script that gathers important log, configuration, and status data from your deployed nodes all into one place. This script is useful if you need specific information for troubleshooting or while making support calls.

We recommend you use the script after a successful deployment to provide a baseline that can be compared against future upgrades or failures. To archive the configuration, status, and logs from the deployment:

```
/var/lib/contrail_cloud/scripts/collect_data.sh -r all
```

The usage description of the `collect_data.sh` script is as follows:

```
Usage: [-r ROLES ] [-d ] [-h]? ]
  -r ROLES  collect data from specific role or environment. Possible values for ROLES:
            jumphost, undercloud, rhvm, control_hosts, k8s_hosts, openstack_controllers,
            contrail_tsn, compute_nodes, storage_nodes, contrail-insights_controller.
            You can specify multiple roles, eg. -r \"undercloud jumphost \"
            If not set data will be collected from all hosts.
  -d        enable debugging messages
  -e        external config
  -h        print usage information
```

Appendix A: Sample Configuration Files

IN THIS SECTION

- Sample site.yml Configuration File | 25
- Sample inventory.yml Configuration File | 49
- Sample control-host-nodes.yml Configuration File | 51
- Sample overcloud-nics.yml Configuration File | 53
- Sample compute-nodes.yml Configuration File | 78
- Sample storage-nodes.yml Configuration File | 79
- Sample K8's-host-nodes.yml Configuration File | 80
- Sample vault-data.yml Configuration File | 84

NOTE: AppFormix was renamed to Contrail Insights, but is still called appformix in the YAML configuration files.

Sample site.yml Configuration File

```
# Copyright 2023 Juniper Networks, Inc. All rights reserved.
# Licensed under the Juniper Networks Script Software License (the "License").
# You may not use this script file except in compliance with the License, which is located at
# http://www.juniper.net/support/legal/scriptlicense/
# Unless required by applicable law or otherwise agreed to in writing by the parties,
# software distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
#
#
#

global:
  # List of DNS nameservers
```

```

dns:
  # Google Public DNS
  - "8.8.8.8"
  - "8.8.4.4"
# List of NTP time servers
ntp:
  # public pool.ntp.org
  - "0.pool.ntp.org"
  - "1.pool.ntp.org"
  - "2.pool.ntp.org"
  - "3.pool.ntp.org"
# Timezone for all servers
timezone: 'America/Los_Angeles'
rhel:
  # Contrail Cloud Activation Key
  # These details are provided when you request an activation key from
  # contrail cloud subscriptions <contrail_cloud_subscriptions@juniper.net>
  #
  satellite:
    #SATELLITE_KEY should be defined in vault-data.yml file
    #SATELLITE_ORG
    organization: "ContrailCloud16"
    #SATELLITE_FQDN
    fqdn: contrail-cloud.juniper.net
# DNS domain information.
# Must be unique for every deployment to avoid name conflicts.
# Need not be a registered DNS domain.
domain: "my.unique.domain"

jumphost:
  network:
    # network used for provisioning (PXE booting) servers
  provision:
    # jumphost nic to be used for provisioning (PXE booting) servers
    nic: eno1

control_hosts:
  # Contains a list of label to disk mappings for roles
  disk_mapping:
    # the control host always uses the "baremetal" role
  baremetal:
    # Mapping of labels to disk devices. The label is assigned to the disk
    # device so that the disk can be referenced by the alias in other

```

```

# configurations. for example /dev/disk/by-alias/<label>
# Each list element contains:
#   label: label to assign
#   name: disk device path (e.g. /dev/sdb)
#   OR
#   hctl: alternative notation for disk paths specifying SCSI address
#         (Host, Channel, Target and Lun) The HCTL can be found with the
#         ls SCSI (or lspci) command or it can be found in introspection data
#
- label: spinning-0
  name: /dev/sdb
- label: spinning-1
  name: /dev/sdc
- label: spinning-2
  name: /dev/sdd
- label: spinning-3
  name: /dev/sde
- label: ssd-0
  hctl: "0:2:3:0"
storage:
# Define a set of disk groups that can be referenced for VM virtual disk allocations
# These become virsh storage pools on the control host
# Each pool has:
#   mountpoint: "/absolute/path/where/lvm/will/get/mounted"
#   type: Either "dir" or "logical".
#       "dir" does not create any new volumes
#       it is useful if one large hardware raid is used as /
#       "logical" is a LVM volume placed on the list of "disk".
#   disk: List of disk devices to use for the pool
hdd_storage:
  mountpoint: "/srv/hdd_storage"
  type: logical
  disk:
    - "/dev/disk/by-alias/spinning-0"
    - "/dev/disk/by-alias/spinning-1"
    - "/dev/disk/by-alias/spinning-2"
    - "/dev/disk/by-alias/spinning-3"
ssd_storage:
  mountpoint: "/srv/ssd_storage"
  type: logical
  disk:
    - "/dev/disk/by-alias/ssd-0"
#srv:

```

```

# mountpoint: "/srv"
# type: dir

vm:
# VM for Openstack Controller role
control:
  disk:
    # Root disk
  vda:
    # Virsh storage pool (see storage above)
    pool: hdd_storage
# VMs for ContrailController role
contrail-k8s:
  disk:
    # Root disk
  vda:
    # Virsh storage pool (see storage above)
    pool: hdd_storage
# VM for ContrailTsn role
contrail-tsn:
  disk:
    # Root disk
  vda:
    # Virsh storage pool (see storage above)
    pool: hdd_storage
# VM for AppFormix controller role
appformix-controller:
  disk:
    # Root disk
  vda:
    # Virsh storage pool (see storage above)
    pool: hdd_storage

compute_hosts:
  sriov:
    #enable sriov support
    enabled: true
    #enable sriov with dpdk
    # Contrail vrouter mode:
    #   supported values are: dpdk or anything else means kernel vRouter
    mode: dpdk
    #Sriov NumVFs separated by comma
    num_vf:

```



```

- "ens2f1:7"
#NovaPCIPassthrough settings
pci_passthrough:
  - devname: "ens2f1"
    physical_network: "sriov1"
root_disk:
# Define root disk for the listed ironic profiles.
# The default of "/dev/sda" will be used if there is no
# specific profile definition
#
# In case 'name' hint should be dropped in favor of other
# hints the block scalar can be used:
#
#   ComputeKernel0Hw0: |
#     vendor: VendorName
#
# which will overwrite default values.
# To keep key-value structure and not use 'name' hint, it can be set to the
# value which always evaluate to True, for example:
#
#   ComputeKernel0Hw0:
#     name: "s!=NonExistingDevice"
#     vendor: VendorName
#
# For more details please check:
# https://docs.openstack.org/ironic/latest/install/advanced.html#specifying-the-disk-for-
# deployment-root-device-hints
#
ComputeKernel0Hw0:
  name: "/dev/sda"
ComputeKernel0Hw1:
  name: "/dev/sda"
ComputeKernel1Hw1:
  name: "/dev/sda"
ComputeKernel1Hw0:
  name: "/dev/sda"
ComputeDpdk0Hw2:
  name: "/dev/sda"
ComputeDpdk1Hw3:
  name: "/dev/sda"
ComputeSriov0Hw4:
  name: "/dev/sda"
ComputeSriov1Hw5:

```

```

    name: "/dev/sda"
resource:
  minimal_disk:
    # This value will be used as the local_gb size for the listed ironic profiles
    # If not defined for a profile then the default will be used
    ComputeKernel0Hw0: 50
    ComputeKernel0Hw1: 50
    ComputeKernel1Hw1: 50
    ComputeKernel1Hw0: 50
    ComputeDpdk0Hw2: 50
    ComputeDpdk1Hw3: 50
    ComputeSriov0Hw4: 50
    ComputeSriov1Hw5: 50

storage_hosts:
  root_disk:
    # Define root disk for the listed ironic profiles.
    # The default of "/dev/sda" will be used if there is no
    # specific profile definition
    CephStorage0Hw6:
      name: "/dev/sda"
    CephStorage1Hw7:
      name: "/dev/sda"

undercloud:
  nova:
    # Nova flavor definitions for roles
    flavor:
      CephStorage0Hw6:
        cpu: 1
        memory: 4
        disk: 40
        ephemeral: 0
      CephStorage1Hw7:
        cpu: 1
        memory: 4
        disk: 40
        ephemeral: 0
      ComputeKernel0Hw0:
        cpu: 8
        memory: 24
        disk: 40
        ephemeral: 0

```

```

ComputeKernel0Hw1:
  cpu: 8
  memory: 24
  disk: 40
  ephemeral: 0
ComputeKernel1Hw1:
  cpu: 8
  memory: 24
  disk: 40
  ephemeral: 0
ComputeKernel1Hw0:
  cpu: 8
  memory: 24
  disk: 40
  ephemeral: 0
ComputeDpdk0Hw2:
  cpu: 8
  memory: 24
  disk: 40
  ephemeral: 0
ComputeDpdk1Hw3:
  cpu: 8
  memory: 24
  disk: 40
  ephemeral: 0
ComputeSriov0Hw4:
  cpu: 8
  memory: 24
  disk: 40
  ephemeral: 0
ComputeSriov1Hw5:
  cpu: 8
  memory: 24
  disk: 40
  ephemeral: 0

k8s:
  external_vip_ip: 10.10.10.102
  internal_api_vip_ip: 172.16.0.91

overcloud:
  # Contains a list of label to disk mappings for roles.
  # When Ceph Storage is disabled, compute-related roles (Compute* and

```

```

# ComputeDpdk* roles) will use any disks labeled with
# "ephemeral-<digits>" for local Nova ephemeral storage.
disk_mapping:
  ComputeKernel:
    # Mapping of labels to disk devices. The label is assigned to the disk
    # device so that the disk can be referenced by the alias in other
    # configurations. for example /dev/disk/by-alias/<label>
    # Each list element contains:
    #   label: label to assign
    #   hctl: disk device path H:C:T:L (the path must exist). see lsscsi
    - label: ephemeral-0
      hctl: '5:0:0:0'
    - label: ephemeral-1
      hctl: '6:0:0:0'
    - label: ephemeral-2
      hctl: '7:0:0:0'
    - label: ephemeral-3
      hctl: '8:0:0:0'
  ComputeKernel0Hw0:
    # Mapping of labels to disk devices. The label is assigned to the disk
    # device so that the disk can be referenced by the alias in other
    # configurations. for example /dev/disk/by-alias/<label>
    # Each list element contains:
    #   label: label to assign
    #   hctl: disk device path H:C:T:L (the path must exist). see lsscsi
    - label: ephemeral-0
      hctl: '5:0:0:0'
    - label: ephemeral-1
      hctl: '6:0:0:0'
    - label: ephemeral-2
      hctl: '7:0:0:0'
    - label: ephemeral-3
      hctl: '8:0:0:0'
  ComputeKernel1Hw0:
    # Mapping of labels to disk devices. The label is assigned to the disk
    # device so that the disk can be referenced by the alias in other
    # configurations. for example /dev/disk/by-alias/<label>
    # Each list element contains:
    #   label: label to assign
    #   hctl: disk device path H:C:T:L (the path must exist). see lsscsi
    - label: ephemeral-0
      hctl: '5:0:0:0'
    - label: ephemeral-1

```

```

    hctl: '6:0:0:0'
  - label: ephemeral-2
    hctl: '7:0:0:0'
  - label: ephemeral-3
    hctl: '8:0:0:0'
ComputeKernel1Hw1:
  # Mapping of labels to disk devices. The label is assigned to the disk
  # device so that the disk can be referenced by the alias in other
  # configurations. for example /dev/disk/by-alias/<label>
  # Each list element contains:
  #   label: label to assign
  #   hctl: disk device path H:C:T:L (the path must exist). see lsscsi
  - label: ephemeral-0
    hctl: '5:0:0:0'
  - label: ephemeral-1
    hctl: '6:0:0:0'
  - label: ephemeral-2
    hctl: '7:0:0:0'
  - label: ephemeral-3
    hctl: '8:0:0:0'
ComputeKernel0Hw1:
  # Mapping of labels to disk devices. The label is assigned to the disk
  # device so that the disk can be referenced by the alias in other
  # configurations. for example /dev/disk/by-alias/<label>
  # Each list element contains:
  #   label: label to assign
  #   hctl: disk device path H:C:T:L (the path must exist). see lsscsi
  - label: ephemeral-0
    hctl: '5:0:0:0'
  - label: ephemeral-1
    hctl: '6:0:0:0'
  - label: ephemeral-2
    hctl: '7:0:0:0'
  - label: ephemeral-3
    hctl: '8:0:0:0'
ComputeDpdk:
  # Mapping of labels to disk devices. The label is assigned to the disk
  # device so that the disk can be referenced by the alias in other
  # configurations. for example /dev/disk/by-alias/<label>
  # Each list element contains:
  #   label: label to assign
  #   hctl: disk device path H:C:T:L (the path must exist). see lsscsi
  - label: ephemeral-0

```

```

    hctl: '5:0:0:0'
  - label: ephemeral-1
    hctl: '6:0:0:0'
  - label: ephemeral-2
    hctl: '7:0:0:0'
  - label: ephemeral-3
    hctl: '8:0:0:0'
ComputeDpdk0Hw2:
  # Mapping of labels to disk devices. The label is assigned to the disk
  # device so that the disk can be referenced by the alias in other
  # configurations. for example /dev/disk/by-alias/<label>
  # Each list element contains:
  #   label: label to assign
  #   hctl: disk device path H:C:T:L (the path must exist). see lsscsi
  - label: ephemeral-0
    hctl: '5:0:0:0'
  - label: ephemeral-1
    hctl: '6:0:0:0'
  - label: ephemeral-2
    hctl: '7:0:0:0'
  - label: ephemeral-3
    hctl: '8:0:0:0'
ComputeDpdk1Hw3:
  # Mapping of labels to disk devices. The label is assigned to the disk
  # device so that the disk can be referenced by the alias in other
  # configurations. for example /dev/disk/by-alias/<label>
  # Each list element contains:
  #   label: label to assign
  #   hctl: disk device path H:C:T:L (the path must exist). see lsscsi
  - label: ephemeral-0
    hctl: '5:0:0:0'
  - label: ephemeral-1
    hctl: '6:0:0:0'
  - label: ephemeral-2
    hctl: '7:0:0:0'
  - label: ephemeral-3
    hctl: '8:0:0:0'
ComputeSriov:
  # Mapping of labels to disk devices. The label is assigned to the disk
  # device so that the disk can be referenced by the alias in other
  # configurations. for example /dev/disk/by-alias/<label>
  # Each list element contains:
  #   label: label to assign

```

```

# hctl: disk device path H:C:T:L (the path must exist). see lsscsi
- label: ephemeral-0
  hctl: '5:0:0:0'
- label: ephemeral-1
  hctl: '6:0:0:0'
- label: ephemeral-2
  hctl: '7:0:0:0'
- label: ephemeral-3
  hctl: '8:0:0:0'
ComputeSriov0Hw4:
# Mapping of labels to disk devices. The label is assigned to the disk
# device so that the disk can be referenced by the alias in other
# configurations. for example /dev/disk/by-alias/<label>
# Each list element contains:
#   label: label to assign
#   hctl: disk device path H:C:T:L (the path must exist). see lsscsi
- label: ephemeral-0
  hctl: '5:0:0:0'
- label: ephemeral-1
  hctl: '6:0:0:0'
- label: ephemeral-2
  hctl: '7:0:0:0'
- label: ephemeral-3
  hctl: '8:0:0:0'
ComputeSriov1Hw5:
# Mapping of labels to disk devices. The label is assigned to the disk
# device so that the disk can be referenced by the alias in other
# configurations. for example /dev/disk/by-alias/<label>
# Each list element contains:
#   label: label to assign
#   hctl: disk device path H:C:T:L (the path must exist). see lsscsi
- label: ephemeral-0
  hctl: '5:0:0:0'
- label: ephemeral-1
  hctl: '6:0:0:0'
- label: ephemeral-2
  hctl: '7:0:0:0'
- label: ephemeral-3
  hctl: '8:0:0:0'
extra_config:
  ComputeDpdkParameters:
    TunedProfileName: "cpu-partitioning"
    ContrailDpdkOptions: "--vr_flow_entries=2000000 --yield_option 0"

```

```

    KernelArgs: "intel_iommu=on iommu=pt default_hugepagesz=1GB hugepagesz=1G hugepages=64
hugepagesz=2M hugepages=2048 isolcpus=2-9,22-29"
    IsolCpusList: "2-9,22-29"
    NovaVcpuPinSet: ['4-9','24-29']
    ContrailSettings:
        SERVICE_CORE_MASK: '0x1'
        DPDK_CTRL_THREAD_MASK: '0x1'
    ExtraSysctlSettings:
        vm.nr_hugepages:
            value: 64
        vm.max_map_count:
            value: 128960
    ComputeDpdk0Hw2Parameters:
        TunedProfileName: "cpu-partitioning"
        ContrailDpdkOptions: "--vr_flow_entries=2000000 --yield_option 0"
        KernelArgs: "intel_iommu=on iommu=pt default_hugepagesz=1GB hugepagesz=1G hugepages=64
hugepagesz=2M hugepages=2048 isolcpus=2-9,22-29"
        IsolCpusList: "2-9,22-29"
        NovaVcpuPinSet: ['4-9','24-29']
        ContrailSettings:
            SERVICE_CORE_MASK: '0x1'
            DPDK_CTRL_THREAD_MASK: '0x1'
        ExtraSysctlSettings:
            vm.nr_hugepages:
                value: 64
            vm.max_map_count:
                value: 128960
    ComputeDpdk1Hw3Parameters:
        TunedProfileName: "cpu-partitioning"
        ContrailDpdkOptions: "--vr_flow_entries=2000000 --yield_option 0"
        KernelArgs: "intel_iommu=on iommu=pt default_hugepagesz=1GB hugepagesz=1G hugepages=64
hugepagesz=2M hugepages=2048 isolcpus=2-9,22-29"
        IsolCpusList: "2-9,22-29"
        NovaVcpuPinSet: ['4-9','24-29']
        ContrailSettings:
            SERVICE_CORE_MASK: '0x1'
            DPDK_CTRL_THREAD_MASK: '0x1'
        ExtraSysctlSettings:
            vm.nr_hugepages:
                value: 64
            vm.max_map_count:
                value: 128960
    ComputeKernelParameters:

```



```

    ContrailVrouterHugepages1GB: 64
    ContrailVrouterHugepages2MB: 2048
    KernelArgs: "intel_iommu=on iommu=pt default_hugepagesz=1GB hugepagesz=1G hugepages=64
hugepagesz=2M hugepages=2048"
    ExtraSysctlSettings:
        vm.nr_hugepages:
            value: 64
        vm.max_map_count:
            value: 128960
    ComputeKernel0Hw0Parameters:
        ContrailVrouterHugepages1GB: 64
        ContrailVrouterHugepages2MB: 2048
        KernelArgs: "intel_iommu=on iommu=pt default_hugepagesz=1GB hugepagesz=1G hugepages=64
hugepagesz=2M hugepages=2048"
        ExtraSysctlSettings:
            vm.nr_hugepages:
                value: 64
            vm.max_map_count:
                value: 128960
    ComputeKernel0Hw1Parameters:
        ContrailVrouterHugepages1GB: 64
        ContrailVrouterHugepages2MB: 2048
        KernelArgs: "intel_iommu=on iommu=pt default_hugepagesz=1GB hugepagesz=1G hugepages=64
hugepagesz=2M hugepages=2048"
        ExtraSysctlSettings:
            vm.nr_hugepages:
                value: 64
            vm.max_map_count:
                value: 128960
    ComputeKernel1Hw0Parameters:
        ContrailVrouterHugepages1GB: 64
        ContrailVrouterHugepages2MB: 2048
        KernelArgs: "intel_iommu=on iommu=pt default_hugepagesz=1GB hugepagesz=1G hugepages=64
hugepagesz=2M hugepages=2048"
        ExtraSysctlSettings:
            vm.nr_hugepages:
                value: 64
            vm.max_map_count:
                value: 128960
    ComputeKernel1Hw1Parameters:
        ContrailVrouterHugepages1GB: 64
        ContrailVrouterHugepages2MB: 2048
        KernelArgs: "intel_iommu=on iommu=pt default_hugepagesz=1GB hugepagesz=1G hugepages=64

```

```

hugepagesz=2M hugepages=2048"
  ExtraSysctlSettings:
    vm.nr_hugepages:
      value: 64
    vm.max_map_count:
      value: 128960

network:
  # The external network is used for referencing the overcloud APIs from outside the
  infrastructure.
  external:
    # Network name used by TripleO Heat Templates
    heat_name: External
    # CIDR (IP/prefix) for the external network subnet
    # Corresponds to the ExternalIpSubnet heat property
    cidr: "10.84.36.64/28"
    # Default route for the external network
    # Corresponds to the ExternalInterfaceDefaultRoute heat property
    gateway: "10.84.36.78"
    # VLAN tag for the external network
    # Corresponds to the ExternalNetworkVlanID heat property
    vlan: 1350
    # Floating virtual IP for the Openstack APIs on the external network
    # Corresponds to the PublicVirtualFixedIPs heat property
    vip: "10.84.36.77"
    # DHCP pool for the external network
    # Be sure that the range is large enough to accomodate all nodes in the external network
    pool:
      # Range start for the DHCP pool
      start: "10.84.36.65"
      # Range end for the DHCP pool
      end: "10.84.36.75"
    # MTU for external network
    # Corresponds to the ExternalNetworkMtu heat property
    mtu: 9000
    # List of roles that can be on this network
    role:
      - Controller
      - AppformixController
  # The internal API network is used for control plane signalling and service API calls
  internal_api:
    # Network name used by TripleO Heat Templates
    heat_name: InternalApi

```

```

# VLAN tag for the internal API network
# Corresponds to the InternalApiNetworkVlanID heat property
vlan: 1200
# CIDR (IP/prefix) for the internal api supernet network subnet
# Corresponds to the InternalApiSupernet heat property
# Supernet is used in spine/leaf configuration
# Supernet accommodate all related leaf networks, e.g. internal_api0 and internal_api1
# Supernet is used to create static routes between leafs
# Supernet is defined only for main network, not per leafs
supernet: "172.16.0.0/22"
# CIDR (IP/prefix) for the internal api network subnet
# Corresponds to the InternalApiIpSubnet heat property
cidr: "172.16.0.0/24"
# Default route for the internal api network
# Corresponds to the InternalApiInterfaceDefaultRoute heat property
gateway: 172.16.0.1
# MTU for internal api network
# Corresponds to the InternalApiNetworkMtu heat property
mtu: 9000
# DHCP pool for the internal api network
# Be sure that the range is large enough to accomodate all nodes in the internal api
network
pool:
  # Range start for the DHCP pool
  start: 172.16.0.100
  # Range end for the DHCP pool
  end: 172.16.0.160
# Floating virtual IP for the Openstack APIs on the internal api network
# Corresponds to the InternalApiVirtualFixedIPs heat property
vip: 172.16.0.90
# List of roles that can be on this network
role:
  - Controller
  - ContrailTsn
  - AppformixController
  - ComputeKernel
  - ComputeDpdk
  - ComputeSriov
# Leaf 0 subnet of the internal_api network
internal_api0:
  # Network name used by TripleO Heat Templates
  heat_name: InternalApi0
  # VLAN tag for the internal API 0 network

```

```

# Corresponds to the InternalApi0NetworkVlanID heat property
vlan: 1201
# CIDR (IP/prefix) for the internal api 0 network subnet
# Corresponds to the InternalApi0IpSubnet heat property
cidr: "172.16.1.0/24"
# Default route for the internal api 0 network
# Corresponds to the InternalApi0InterfaceDefaultRoute heat property
gateway: 172.16.1.1
# MTU for internal api 0 network
# Corresponds to the InternalApi0NetworkMtu heat property
mtu: 9000
# DHCP pool for the internal api 0 network
# Be sure that the range is large enough to accomodate all nodes in the internal api
network
  pool:
    # Range start for the DHCP pool
    start: 172.16.1.100
    # Range end for the DHCP pool
    end: 172.16.1.200
  # List of roles that can be on this network
  role:
    - ComputeKernel0Hw0
    - ComputeKernel0Hw1
    - ComputeSriov0Hw4
    - ComputeDpdk0Hw2
# Leaf 1 subnet of the internal_api network
internal_api1:
  # Network name used by TripleO Heat Templates
  heat_name: InternalApi1
  # VLAN tag for the internal API 1 network
  # Corresponds to the InternalApi1NetworkVlanID heat property
  vlan: 1202
  # CIDR (IP/prefix) for the internal api 1 network subnet
  # Corresponds to the InternalApi1IpSubnet heat property
  cidr: "172.16.2.0/24"
  # Default route for the internal api 1 network
  # Corresponds to the InternalApi1InterfaceDefaultRoute heat property
  gateway: 172.16.2.1
  # MTU for internal api 1 network
  # Corresponds to the InternalApi1NetworkMtu heat property
  mtu: 9000
  # DHCP pool for the internal api 1 network
  # Be sure that the range is large enough to accomodate all nodes in the internal api

```

```

network
  pool:
    # Range start for the DHCP pool
    start: 172.16.2.100
    # Range end for the DHCP pool
    end: 172.16.2.200
  # List of roles that can be on this network
  role:
    - ComputeSriov1Hw5
    - ComputeKernel1Hw0
    - ComputeDpdk1Hw3
    - ComputeKernel1Hw1
  # The management network is defined for backwards-compatibility in RHOSP and is not
  # used by default by any roles.
  management:
    # Network name used by TripleO Heat Templates
    heat_name: Management
    # VLAN tag for the management network
    # Corresponds to the ManagementNetworkVlanID heat property
    vlan: 1300
    # CIDR (IP/prefix) for the network subnet
    # Corresponds to the ManagementIpSubnet heat property
    cidr: "192.168.1.0/24"
    # MTU for the network
    # Corresponds to the ManagementNetworkMtu heat property
    mtu: 9000
    # DHCP pool for the network
    # Be sure that the range is large enough to accomodate all nodes in the network
    pool:
      # Range start for the DHCP pool
      start: 192.168.1.100
      # Range end for the DHCP pool
      end: 192.168.1.200
  # The storage network is used for Compute storage access
  storage:
    # Network name used by TripleO Heat Templates
    heat_name: Storage
    # VLAN tag for the storage network
    # Corresponds to the StorageNetworkVlanID heat property
    vlan: 1500
    supernet: "172.16.16.0/22"
    cidr: "172.16.16.0/24"
    gateway: 172.16.16.1

```

```

mtu: 9000
pool:
  start: 172.16.16.100
  end: 172.16.16.200
# List of roles that can be on this network
role:
  - Controller
  - CephStorage
  - ComputeKernel
  - ComputeDpdk
  - ComputeSriov
  - ContrailTsn
# Leaf 0 subnet of the storage network
storage0:
  # Network name used by TripleO Heat Templates
  heat_name: Storage0
  vlan: 1501
  cidr: "172.16.17.0/24"
  gateway: 172.16.17.1
  mtu: 9000
  pool:
    start: 172.16.17.100
    end: 172.16.17.200
  # List of roles that can be on this network
  role:
    - CephStorage0Hw6
    - ComputeKernel0Hw0
    - ComputeKernel0Hw1
    - ComputeDpdk0Hw2
    - ComputeSriov0Hw4
# Leaf 1 subnet of the storage network
storage1:
  # Network name used by TripleO Heat Templates
  heat_name: Storage1
  vlan: 1502
  cidr: "172.16.18.0/24"
  gateway: 172.16.18.1
  mtu: 9000
  pool:
    start: 172.16.18.100
    end: 172.16.18.200
  # List of roles that can be on this network
  role:

```

```

- ComputeSriov1Hw5
- ComputeKernel1Hw0
- ComputeDpdk1Hw3
- ComputeKernel1Hw1
- CephStorage1Hw7
# The storage management network is used for storage operations such as replication
storage_mgmt:
  # Network name used by TripleO Heat Templates
  heat_name: StorageMgmt
  # VLAN tag for the storage management network
  # Corresponds to the StorageMgmtNetworkVlanID heat property
  vlan: 1450
  supernet: "172.16.20.0/22"
  cidr: "172.16.20.0/24"
  gateway: 172.16.20.1
  vip_enable: false
  mtu: 9000
  pool:
    start: 172.16.20.100
    end: 172.16.20.200
  # List of roles that can be on this network
  role:
    - Controller
# Leaf 0 subnet of the storage_mgmt network
storage_mgmt0:
  # Network name used by TripleO Heat Templates
  heat_name: StorageMgmt0
  vlan: 1451
  cidr: "172.16.21.0/24"
  gateway: 172.16.21.1
  mtu: 9000
  pool:
    start: 172.16.21.100
    end: 172.16.21.200
  # List of roles that can be on this network
  role:
    - CephStorage0Hw6
# Leaf 1 subnet of the storage_mgmt network
storage_mgmt1:
  # Network name used by TripleO Heat Templates
  heat_name: StorageMgmt1
  vlan: 1452
  cidr: "172.16.22.0/24"

```

```

gateway: 172.16.22.1
mtu: 9000
pool:
  start: 172.16.22.100
  end: 172.16.22.200
# List of roles that can be on this network
role:
  - CephStorage1Hw7
# The tenant network is used for tenant workload data
tenant:
  # Network name used by TripleO Heat Templates
  heat_name: Tenant
  # VLAN tag for the tenant network
  # Corresponds to the TenantNetworkVlanID heat property
  vlan: 1250
  supernet: "172.16.80.0/21"
  cidr: "172.16.81.0/24"
  gateway: 172.16.81.1
  vrouter_gateway: 172.16.81.1
  mtu: 9000
  pool:
    start: 172.16.81.100
    end: 172.16.81.200
  # List of roles that can be on this network
  role:
    - ContrailTsn
    - ComputeKernel
    - ComputeDpdk
    - ComputeSriov
# Leaf 0 subnet of the tenant network
tenant0:
  # Network name used by TripleO Heat Templates
  heat_name: Tenant0
  vlan: 1251
  cidr: "172.16.82.0/24"
  gateway: 172.16.82.1
  vrouter_gateway: 172.16.82.1
  mtu: 9000
  pool:
    start: 172.16.82.100
    end: 172.16.82.200
  # List of roles that can be on this network
  role:

```



```

    - ComputeKernel0Hw0
    - ComputeKernel0Hw1
    - ComputeDpdk0Hw2
    - ComputeSriov0Hw4
# Leaf 1 subnet of the tenant network
tenant1:
    # Network name used by TripleO Heat Templates
    heat_name: Tenant1
    vlan: 1252
    cidr: "172.16.83.0/24"
    gateway: 172.16.83.1
    vrouter_gateway: 172.16.83.1
    mtu: 9000
    pool:
        start: 172.16.83.100
        end: 172.16.83.200
    # List of roles that can be on this network
    role:
        - ComputeSriov1Hw5
        - ComputeKernel1Hw0
        - ComputeDpdk1Hw3
        - ComputeKernel1Hw1
# Contrail sepcific settings
#contrail:
#   aaa_mode: cloud-admin
#   vrouter:
#       contrail_settings:
#           # Settings per profile.
#           # Profile's contrail_settings replace default settings and should include
#           # all keys and values which are intended to be exported on given role.
#           # When leafs are used it implies per profile configuration as it defines
#           # VROUTER_GATEWAY for profile by quering node's tenant network for
#           # vrouter_gateway value.
#           default:
#               VROUTER_GATEWAY: 172.16.81.254
#               BGP_ASN: 64512
#               LACP_RATE: 1
#               ComputeKernel1Hw0:
#                   LACP_RATE: 1

# Information used to generate the SSL certificates for the public Openstack service APIs
tls:
    # SSL key size in KB for CA generation

```

```

ca_key_size: 2048
# SSL key size in KB for SSL of openstack external VIP
ssl_key_size: 2048
#countryName_default
country: "US"
#stateOrProvinceName_default
state: "CA"
#localityName_default
city: "Sunnyvale"
#organizationalUnitName_default
organization: "JNPR"
#commonName_default - this is typically the external VIP
common_name: "10.10.10.90"

```

ceph:

```

# Choice to enable Ceph storage in the overcloud.
# "true" means that Ceph will be deployed as the backed for Cinder and Glance services.
# "false" false means that Ceph will not be deployed.
enabled: true
# Ceph OSD disk configuration

```

osd:

```

# Update the Ceph crush map when OSDs are started
crush_update_on_start: true
# Ceph OSD disk assignments. The named disks will be exclusively used by Ceph for

```

persistence.

```

# Lvm is a default scenario for ceph deployment with bluestore as a backend.
# When all named disks are the same type, spinning or solid state, all of them will be used
# as ceph osds. When disks with mixed types are defined spinning disks will be used as osds
# and on solid state disks ceph db will be created. For mixed types of disks the automatic

```

pgp

```

# number calculation requires assigning key 'contents' with value 'db' to ssd disks.
# In below example disks sd[b-e] are spinning disks and sdf is solid state disk.

```

default:

disk:

```

'/dev/sdb':
'/dev/sdc':
'/dev/sdd':
'/dev/sde':
'/dev/sdf':
  contents: db

```

CephStorage0Hw6:

disk:

```

'/dev/sdb':

```

```

    '/dev/sdc':
    '/dev/sdd':
    '/dev/sde':
    '/dev/sdf':
    contents: db
CephStorage1Hw7:
  disk:
    '/dev/sdb':
    '/dev/sdc':
    '/dev/sdd':
    '/dev/sde':
    '/dev/sdf':
    contents: db
# By default, pgp number is calculated by contrail cloud. If you want, you can give this
parameter
# by yourself. Use the calculator on the website: https://ceph.com/pgcalc/. Calculator takes
into
# account also pool utilization. Calculated pgp_num can be introduced in configuration as
below.
# It's defined per used pool.
# pool:
#   vms:
#     pgp_num: 32
#   rbd:
#     pgp_num: 32
#   images:
#     pgp_num: 32
#   volumes:
#     pgp_num: 32
#   backups:
#     pgp_num: 32
#
# Rados Gateway when enabled, which is a default behaviour, creates it's own ceph pools
# not tracked by contrail cloud. Those pools can be predefined to better control
# their sizes. Below pools definitions are not an exhaustive, please consult with
# https://ceph.com/pgcalc/
# Pools should have enabled application according to their use.
# If not changed explicit, pools are created with 'rbd' application assigned.
# Available options are:
#   - rbd for the Ceph Block Device
#   - rgw for the Ceph Object Gateway
#   - cephfs for the Ceph Filesystem
# or user defined value for custom application.

```

```

# More details can be found on
# https://access.redhat.com/documentation/en-us/red\_hat\_ceph\_storage/3/html/
storage_strategies_guide/pools-1#enable-application
#   .rgw.root:
#       pgp_num: 16
#       enabled: true
#       replica: 3
#       application: rgw
#   default.rgw.control:
#       pgp_num: 16
#       enabled: true
#       replica: 3
#       application: rgw
#   default.rgw.meta:
#       pgp_num: 16
#       enabled: true
#       replica: 3
#       application: rgw
#   default.rgw.log:
#       pgp_num: 16
#       enabled: true
#       replica: 3
#       application: rgw
#   default.rgw.buckets.index:
#       pgp_num: 16
#       enabled: true
#       replica: 3
#       application: rgw
#   default.rgw.buckets.data:
#       pgp_num: 16
#       enabled: true
#       replica: 3
#       application: rgw
#   default.rgw.buckets.non-ec:
#       pgp_num: 16
#       enabled: true
#       replica: 3
#       application: rgw

appformix:
# Set to true if you have multiple control hosts which allows Apformix to run in HA mode
enable_ha: true
# Floating virtual IP for the Appformix APIs on the external network, used and required by HA

```

```

mode.
  vip: "10.10.10.101"
  # Floating virtual IP for the Appformix APIs on the internal network, used and required by HA
mode.
  secondary_vip: "172.16.0.89"
  keepalived:
    # Set which interface will be used for vrrp
    vrrp_interface: "enp2s0"
    # Set which interface will be used for second vrrp
    secondary_vrrp_interface: "vlan1200"

```

Sample inventory.yml Configuration File

```

# Copyright 2023 Juniper Networks, Inc. All rights reserved.
# Licensed under the Juniper Networks Script Software License (the "License").
# You may not use this script file except in compliance with the License, which is located at
# http://www.juniper.net/support/legal/scriptlicense/
# Unless required by applicable law or otherwise agreed to in writing by the parties,
# software distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
#
#
#

# Common values shared among group of nodes
ipmi_hardware1: &hardware1
  pm_type: "ipmi"
  pm_user: "{{ vault['inventory_nodes']['hardware1']['pm_user'] }}"
  pm_password: "{{ vault['inventory_nodes']['hardware1']['pm_password'] }}"
  capabilities: "boot_mode:uefi"

# List of baremetal server nodes that can be used for the deploying roles
# Each list item contains:
#   name: logical name to assign this resource (string)
#   pm_addr: IP address for resourceIPMI interface (string)
#   pm_type: Ironi driver to interface with this resource (typically ipmi) (string)
#   pm_user: IPMI user account (string)
#   pm_password: IPMI account user password (string)
#   capabilities: String of comma separated list of node capabilities.

```

```

#           Capabilities 'profile' and 'boot_option' are managed
#           by Contrail Cloud and will be omitted. (string)
#           e.g capabilities: "boot_mode:uefi" set boot mode to uefi
#
# Some values common for nodes can be moved to dedicated section like ipmi_hardware1
# and be referred like this:
#   <<: *hardware1
inventory_nodes:
  - name: "control-host1"
    pm_addr: "10.10.11.58"
    <<: *hardware1
  - name: "control-host2"
    pm_addr: "10.10.11.59"
    <<: *hardware1
  - name: "control-host3"
    pm_addr: "10.10.11.60"
    <<: *hardware1
  - name: "storage1"
    pm_addr: "10.10.11.61"
    <<: *hardware1
  - name: "storage2"
    pm_addr: "10.10.11.62"
    <<: *hardware1
  - name: "storage3"
    pm_addr: "10.10.11.63"
    <<: *hardware1
  - name: "computedpd1"
    pm_addr: "10.10.11.64"
    <<: *hardware1
  - name: "computedpd2"
    pm_addr: "10.10.11.65"
    <<: *hardware1
  - name: "compute1"
    pm_addr: "10.10.11.66"
    <<: *hardware1
  - name: "compute2"
    pm_addr: "10.10.11.67"
    <<: *hardware1
  - name: "compute3"
    pm_addr: "10.10.11.68"
    <<: *hardware1
  - name: "compute4"
    pm_addr: "10.10.11.69"

```

```

    <<: *hardware1
  - name: "computesriov1"
    pm_addr: "10.10.11.70"
    <<: *hardware1
  - name: "computesriov2"
    pm_addr: "10.10.11.71"
    <<: *hardware1

```

Sample control-host-nodes.yml Configuration File

```

# Copyright 2023 Juniper Networks, Inc. All rights reserved.
# Licensed under the Juniper Networks Script Software License (the "License").
# You may not use this script file except in compliance with the License, which is located at
# http://www.juniper.net/support/legal/scriptlicense/
# Unless required by applicable law or otherwise agreed to in writing by the parties,
# software distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
#
#
#

# List of nodes to use as control host role
# Each list item contains a set of variables which can be referenced
# with "{{ host.<variable> }}" in control_host_nodes_network_config below.
# Other ad-hoc variables can be added as needed.
#   name: name of a node in the inventory (string)
#   hostname: hostname to assign the node after it is imaged (string)
#   control_ip_netmask: static CIDR address on Control Plane network.
#                       Choose a value outside the DHCP range. (string)
#   dns_server1,dns_server2: dns server addresses (string)
#   max_mtu: The largest MTU supported by an interface
#

control_host_nodes:
  - name: "control-host1"
    control_ip_netmask: "192.168.213.5/24"
    max_mtu: 9216
  - name: "control-host2"
    control_ip_netmask: "192.168.213.6/24"

```

```

    max_mtu: 9216
  - name: "control-host3"
    control_ip_netmask: "192.168.213.7/24"
    max_mtu: 9216

# Template for network layout on all control host nodes
# This follows the syntax
# https://cloudinit.readthedocs.io/en/latest/topics/network-config-format-v1.html#network-config-
v1
# or
# https://cloudinit.readthedocs.io/en/latest/topics/network-config-format-v2.html#network-config-
v2
# variables from control_host_nodes can be referred with "{{ host.<variable> }}"

control_host_nodes_network_config:
  version: 1
  config:
    - type: physical
      name: eno1
      subnets:
        - type: dhcp
      mtu: "{{ host.max_mtu }}"
      nm_controlled: true
    - type: physical
      name: eno2
      mtu: "{{ host.max_mtu }}"
      nm_controlled: true
    - type: physical
      name: ens7f0
      mtu: "{{ host.max_mtu }}"
      nm_controlled: true
    - type: physical
      name: ens7f1
      mtu: "{{ host.max_mtu }}"
      nm_controlled: true
    - type: bond
      name: bond0
      mtu: "{{ host.max_mtu }}"
      nm_controlled: true
      bond_interfaces:
        - ens7f0
        - ens7f1
  params:

```



```

    bond-mode: "802.3ad"
    xmit_hash_policy: layer3+4
    lacp_rate: fast
    miimon: "100"

control_hosts:
  # The mapping from control host interfaces to the control VM interfaces
  # The first interface (enp1s0) must always be the Control Plane network to allow the VM to PXE
boot
  # VM interface names must be sequential with no gaps (e.g. enp1s0, enp2s0, enp3s0,...)
vm_interfaces:
  - interface: enp1s0
    physical_interface: eno1
  - interface: enp2s0
    physical_interface: eno2
  - interface: enp3s0
    physical_interface: bond0

```

Sample overcloud-nics.yml Configuration File

```

# Copyright 2023 Juniper Networks, Inc. All rights reserved.
# Licensed under the Juniper Networks Script Software License (the "License").
# You may not use this script file except in compliance with the License, which is located at
# http://www.juniper.net/support/legal/scriptlicense/
# Unless required by applicable law or otherwise agreed to in writing by the parties,
# software distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
#
#
#

Controller_network_config:
  - type: interface
    name: enp1s0
    dns_servers:
      get_param: DnsServers
    use_dhcp: false
    mtu:
      get_param: ControlPlaneMtu

```

```

addresses:
- ip_netmask:
    list_join:
      - '/'
      - - get_param: ControlPlaneIp
        - get_param: ControlPlaneSubnetCidr
- type: vlan
  device: enp1s0
  vlan_id:
    get_param: StorageNetworkVlanID
  mtu:
    get_param: StorageMtu
  addresses:
  - ip_netmask:
      get_param: StorageIpSubnet
    routes:
    -
      ip_netmask:
        get_param: StorageSupernet
      next_hop:
        get_param: StorageInterfaceDefaultRoute
- type: vlan
  device: enp1s0
  vlan_id:
    get_param: StorageMgmtNetworkVlanID
  mtu:
    get_param: StorageMgmtMtu
  addresses:
  - ip_netmask:
      get_param: StorageMgmtIpSubnet
    routes:
    -
      ip_netmask:
        get_param: StorageMgmtSupernet
      next_hop:
        get_param: StorageMgmtInterfaceDefaultRoute
- type: vlan
  device: enp1s0
  vlan_id:
    get_param: InternalApiNetworkVlanID
  mtu:
    get_param: InternalApiMtu
  addresses:

```

```

- ip_netmask:
    get_param: InternalApiIpSubnet
routes:
-
    ip_netmask:
        get_param: InternalApiSupernet
    next_hop:
        get_param: InternalApiInterfaceDefaultRoute
- type: interface
  name: enp2s0
  mtu:
    get_param: ExternalMtu
  addresses:
  - ip_netmask:
      get_param: ExternalIpSubnet
    routes:
    -
        default: True
        next_hop:
            get_param: ExternalInterfaceDefaultRoute
- type: interface
  name: enp3s0
  use_dhcp: false

AppformixController_network_config:
- type: interface
  name: enp1s0
  dns_servers:
    get_param: DnsServers
  use_dhcp: false
  mtu:
    get_param: ControlPlaneMtu
  addresses:
  - ip_netmask:
      list_join:
        - '/'
        - - get_param: ControlPlaneIp
          - get_param: ControlPlaneSubnetCidr
- type: vlan
  device: enp1s0
  vlan_id:
    get_param: InternalApiNetworkVlanID
  mtu:

```

```

    get_param: InternalApiMtu
addresses:
- ip_netmask:
    get_param: InternalApiIpSubnet
routes:
-
    ip_netmask:
        get_param: InternalApiSupernet
    next_hop:
        get_param: InternalApiInterfaceDefaultRoute
- type: interface
  name: enp2s0
  mtu:
    get_param: ExternalMtu
  addresses:
  - ip_netmask:
      get_param: ExternalIpSubnet
  routes:
  -
    default: True
    next_hop:
        get_param: ExternalInterfaceDefaultRoute
- type: interface
  name: enp3s0
  use_dhcp: false

ContrailTsn_network_config:
- type: interface
  name: enp1s0
  dns_servers:
    get_param: DnsServers
  mtu:
    get_param: ControlPlaneMtu
  addresses:
  - ip_netmask:
      list_join:
        - '/'
        - - get_param: ControlPlaneIp
          - get_param: ControlPlaneSubnetCidr
  use_dhcp: false
  routes:
  -
    default: True

```

```

    next_hop:
      get_param: ControlPlaneDefaultRoute
- type: vlan
  device: enp1s0
  vlan_id:
    get_param: InternalApiNetworkVlanID
  mtu:
    get_param: InternalApiMtu
  addresses:
- ip_netmask:
    get_param: InternalApiIpSubnet
  routes:
-
    ip_netmask:
      get_param: InternalApiSupernet
    next_hop:
      get_param: InternalApiInterfaceDefaultRoute
- type: interface
  name: enp2s0
  use_dhcp: false
- type: interface
  name: enp3s0
  use_dhcp: false
  mtu:
    get_param: TenantMtu
- type: vlan
  device: enp3s0
  vlan_id:
    get_param: TenantNetworkVlanID
  mtu:
    get_param: TenantMtu
  use_dhcp: false
- type: contrail_vrouter
  name: vhost0
  members:
-
    type: interface
    name:
      str_replace:
        template: vlanVLANID
      params:
        VLANID: {get_param: TenantNetworkVlanID}
    use_dhcp: false

```

```

mtu:
  get_param: TenantMtu
addresses:
- ip_netmask:
  get_param: TenantIpSubnet
routes:
-
  ip_netmask:
  get_param: TenantSupernet
  next_hop:
  get_param: TenantInterfaceDefaultRoute

ComputeKernel0Hw1_network_config:
- type: interface
  name: nic1
  dns_servers:
  get_param: DnsServers
  use_dhcp: false
  mtu:
  get_param: ControlPlaneMtu
  addresses:
  - ip_netmask:
    list_join:
    - '/'
    - - get_param: ControlPlaneIp
      - get_param: ControlPlaneSubnetCidr
  routes:
  -
    default: True
    next_hop:
    get_param: ControlPlaneDefaultRoute
- type: vlan
  device: nic1
  vlan_id:
  get_param: Storage0NetworkVlanID
  mtu:
  get_param: Storage0NetworkMtu
  addresses:
  - ip_netmask:
    get_param: Storage0IpSubnet
  routes:
  -
    ip_netmask:

```

```

        get_param: StorageSupernet
    next_hop:
        get_param: Storage0InterfaceDefaultRoute
- type: vlan
  device: nic1
  vlan_id:
    get_param: InternalApi0NetworkVlanID
  mtu:
    get_param: InternalApi0NetworkMtu
  addresses:
    - ip_netmask:
        get_param: InternalApi0IpSubnet
    routes:
      -
        ip_netmask:
          get_param: InternalApiSupernet
        next_hop:
          get_param: InternalApi0InterfaceDefaultRoute
- type: interface
  name: nic2
  use_dhcp: false
- type: linux_bond
  name: bond0
  use_dhcp: false
  bonding_options: "mode=802.3ad xmit_hash_policy=layer3+4 lacp_rate=fast updelay=1000
miimon=100"
  mtu:
    get_param: Tenant0NetworkMtu
  members:
    - type: interface
      name: nic3
      primary: true
      mtu:
        get_param: Tenant0NetworkMtu
    - type: interface
      name: nic4
      mtu:
        get_param: Tenant0NetworkMtu
- type: vlan
  vlan_id:
    get_param: Tenant0NetworkVlanID
  device: bond0
- type: contrail_vrouter

```

```

name: vhost0
use_dhcp: false
members:
  -
    type: interface
    name:
      str_replace:
        template: vlanVLANID
      params:
        VLANID: {get_param: Tenant0NetworkVlanID}
    use_dhcp: false
addresses:
  - ip_netmask:
      get_param: Tenant0IpSubnet
mtu:
  get_param: Tenant0NetworkMtu
routes:
  -
    ip_netmask:
      get_param: TenantSupernet
    next_hop:
      get_param: Tenant0InterfaceDefaultRoute

```

ComputeKernel0Hw0_network_config:

```

- type: interface
  name: nic1
  dns_servers:
    get_param: DnsServers
  use_dhcp: false
  mtu:
    get_param: ControlPlaneMtu
  addresses:
    - ip_netmask:
        list_join:
          - '/'
          - - get_param: ControlPlaneIp
            - get_param: ControlPlaneSubnetCidr
  routes:
    -
      default: True
      next_hop:
        get_param: ControlPlaneDefaultRoute
- type: vlan

```



```

device: nic1
vlan_id:
  get_param: Storage0NetworkVlanID
mtu:
  get_param: Storage0NetworkMtu
addresses:
- ip_netmask:
  get_param: Storage0IpSubnet
routes:
-
  ip_netmask:
  get_param: StorageSupernet
  next_hop:
  get_param: Storage0InterfaceDefaultRoute
- type: vlan
device: nic1
vlan_id:
  get_param: InternalApi0NetworkVlanID
mtu:
  get_param: InternalApi0NetworkMtu
addresses:
- ip_netmask:
  get_param: InternalApi0IpSubnet
routes:
-
  ip_netmask:
  get_param: InternalApiSupernet
  next_hop:
  get_param: InternalApi0InterfaceDefaultRoute
- type: interface
name: nic2
use_dhcp: false
- type: linux_bond
name: bond0
use_dhcp: false
bonding_options: "mode=802.3ad xmit_hash_policy=layer3+4 lacp_rate=fast updelay=1000
miimon=100"
mtu:
  get_param: Tenant0NetworkMtu
members:
- type: interface
name: nic3
primary: true

```

```

    mtu:
      get_param: Tenant0NetworkMtu
- type: interface
  name: nic4
  mtu:
    get_param: Tenant0NetworkMtu
- type: vlan
  vlan_id:
    get_param: Tenant0NetworkVlanID
  device: bond0
- type: contrail_vrouter
  name: vhost0
  use_dhcp: false
  members:
    -
      type: interface
      name:
        str_replace:
          template: vlanVLANID
          params:
            VLANID: {get_param: Tenant0NetworkVlanID}
      use_dhcp: false
  addresses:
    - ip_netmask:
        get_param: Tenant0IpSubnet
      mtu:
        get_param: Tenant0NetworkMtu
      routes:
        -
          ip_netmask:
            get_param: TenantSupernet
          next_hop:
            get_param: Tenant0InterfaceDefaultRoute

ComputeKernel1Hw0_network_config:
- type: interface
  name: nic1
  dns_servers:
    get_param: DnsServers
  use_dhcp: false
  mtu:
    get_param: ControlPlaneMtu
  addresses:

```

```

- ip_netmask:
  list_join:
    - '/'
    - - get_param: ControlPlaneIp
      - get_param: ControlPlaneSubnetCidr
  routes:
  -
    default: True
    next_hop:
      get_param: ControlPlaneDefaultRoute
- type: vlan
  device: nic1
  vlan_id:
    get_param: Storage1NetworkVlanID
  mtu:
    get_param: Storage1NetworkMtu
  addresses:
  - ip_netmask:
      get_param: Storage1IpSubnet
    routes:
    -
      ip_netmask:
        get_param: StorageSupernet
      next_hop:
        get_param: Storage1InterfaceDefaultRoute
- type: vlan
  device: nic1
  vlan_id:
    get_param: InternalApi1NetworkVlanID
  mtu:
    get_param: InternalApi1NetworkMtu
  addresses:
  - ip_netmask:
      get_param: InternalApi1IpSubnet
    routes:
    -
      ip_netmask:
        get_param: InternalApiSupernet
      next_hop:
        get_param: InternalApi1InterfaceDefaultRoute
- type: interface
  name: nic2
  use_dhcp: false

```

```

- type: linux_bond
  name: bond0
  use_dhcp: false
  bonding_options: "mode=802.3ad xmit_hash_policy=layer3+4 lacp_rate=fast updelay=1000
miimon=100"
  mtu:
    get_param: Tenant1NetworkMtu
  members:
- type: interface
  name: nic3
  primary: true
  mtu:
    get_param: Tenant1NetworkMtu
- type: interface
  name: nic4
  mtu:
    get_param: Tenant1NetworkMtu
- type: vlan
  vlan_id:
    get_param: Tenant1NetworkVlanID
  device: bond0
- type: contrail_vrouter
  name: vhost0
  use_dhcp: false
  members:
    -
      type: interface
      name:
        str_replace:
          template: vlanVLANID
          params:
            VLANID: {get_param: Tenant1NetworkVlanID}
      use_dhcp: false
  addresses:
- ip_netmask:
    get_param: Tenant1IpSubnet
  mtu:
    get_param: Tenant1NetworkMtu
  routes:
    -
      ip_netmask:
        get_param: TenantSupernet
      next_hop:

```

```

    get_param: Tenant1InterfaceDefaultRoute

ComputeKernel1Hw1_network_config:
- type: interface
  name: nic1
  dns_servers:
    get_param: DnsServers
  use_dhcp: false
  mtu:
    get_param: ControlPlaneMtu
  addresses:
  - ip_netmask:
    list_join:
      - '/'
      - - get_param: ControlPlaneIp
        - get_param: ControlPlaneSubnetCidr
    routes:
    -
      default: True
      next_hop:
        get_param: ControlPlaneDefaultRoute
- type: vlan
  device: nic1
  vlan_id:
    get_param: Storage1NetworkVlanID
  mtu:
    get_param: Storage1NetworkMtu
  addresses:
  - ip_netmask:
    get_param: Storage1IpSubnet
  routes:
  -
    ip_netmask:
      get_param: StorageSupernet
    next_hop:
      get_param: Storage1InterfaceDefaultRoute
- type: vlan
  device: nic1
  vlan_id:
    get_param: InternalApi1NetworkVlanID
  mtu:
    get_param: InternalApi1NetworkMtu
  addresses:

```

```

- ip_netmask:
  get_param: InternalApi1IpSubnet
routes:
-
  ip_netmask:
    get_param: InternalApiSupernet
  next_hop:
    get_param: InternalApi1InterfaceDefaultRoute
- type: interface
  name: nic2
  use_dhcp: false
- type: linux_bond
  name: bond0
  use_dhcp: false
  bonding_options: "mode=802.3ad xmit_hash_policy=layer3+4 lacp_rate=fast updelay=1000
miimon=100"
  mtu:
    get_param: Tenant1NetworkMtu
  members:
  - type: interface
    name: nic3
    primary: true
    mtu:
      get_param: Tenant1NetworkMtu
  - type: interface
    name: nic4
    mtu:
      get_param: Tenant1NetworkMtu
- type: vlan
  vlan_id:
    get_param: Tenant1NetworkVlanID
  device: bond0
- type: contrail_vrouter
  name: vhost0
  use_dhcp: false
  members:
  -
    type: interface
    name:
      str_replace:
        template: vlanVLANID
      params:
        VLANID: {get_param: Tenant1NetworkVlanID}

```

```

        use_dhcp: false
addresses:
- ip_netmask:
    get_param: Tenant1IpSubnet
mtu:
    get_param: Tenant1NetworkMtu
routes:
-
    ip_netmask:
        get_param: TenantSupernet
    next_hop:
        get_param: Tenant1InterfaceDefaultRoute

ComputeSriov0Hw4_network_config:
- type: interface
  name: nic1
  dns_servers:
    get_param: DnsServers
  use_dhcp: false
  mtu:
    get_param: ControlPlaneMtu
  addresses:
  - ip_netmask:
      list_join:
        - '/'
        - - get_param: ControlPlaneIp
          - get_param: ControlPlaneSubnetCidr
  routes:
  -
    default: True
    next_hop:
        get_param: ControlPlaneDefaultRoute
- type: vlan
  device: nic1
  vlan_id:
    get_param: Storage0NetworkVlanID
  mtu:
    get_param: Storage0NetworkMtu
  addresses:
  - ip_netmask:
      get_param: Storage0IpSubnet
  routes:
  -

```

```

    ip_netmask:
      get_param: StorageSupernet
    next_hop:
      get_param: Storage0InterfaceDefaultRoute
- type: vlan
  device: nic1
  vlan_id:
    get_param: InternalApi0NetworkVlanID
  mtu:
    get_param: InternalApi0NetworkMtu
  addresses:
  - ip_netmask:
      get_param: InternalApi0IpSubnet
    routes:
    -
      ip_netmask:
        get_param: InternalApiSupernet
      next_hop:
        get_param: InternalApi0InterfaceDefaultRoute
- type: interface
  name: nic2
  use_dhcp: false
- type: linux_bond
  name: bond0
  use_dhcp: false
  bonding_options: "mode=802.3ad xmit_hash_policy=layer3+4 lacp_rate=fast updelay=1000
miimon=100"
  mtu:
    get_param: Tenant0NetworkMtu
  members:
  - type: interface
    name: nic3
    primary: true
    mtu:
      get_param: Tenant0NetworkMtu
  - type: interface
    name: nic4
    mtu:
      get_param: Tenant0NetworkMtu
- type: vlan
  vlan_id:
    get_param: Tenant0NetworkVlanID
  device: bond0

```



```

- type: contrail_vrouter
  name: vhost0
  use_dhcp: false
  members:
    -
      type: interface
      name:
        str_replace:
          template: vlanVLANID
          params:
            VLANID: {get_param: Tenant0NetworkVlanID}
      use_dhcp: false
  addresses:
    - ip_netmask:
        get_param: Tenant0IpSubnet
      mtu:
        get_param: Tenant0NetworkMtu
    routes:
      -
        ip_netmask:
          get_param: TenantSupernet
        next_hop:
          get_param: Tenant0InterfaceDefaultRoute

ComputeSriov1Hw5_network_config:
- type: interface
  name: nic1
  dns_servers:
    get_param: DnsServers
  use_dhcp: false
  mtu:
    get_param: ControlPlaneMtu
  addresses:
    - ip_netmask:
        list_join:
          - '/'
          - - get_param: ControlPlaneIp
            - get_param: ControlPlaneSubnetCidr
    routes:
      -
        default: True
        next_hop:
          get_param: ControlPlaneDefaultRoute

```

```

- type: vlan
  device: nic1
  vlan_id:
    get_param: Storage1NetworkVlanID
  mtu:
    get_param: Storage1NetworkMtu
  addresses:
  - ip_netmask:
      get_param: Storage1IpSubnet
    routes:
    -
      ip_netmask:
        get_param: StorageSupernet
      next_hop:
        get_param: Storage1InterfaceDefaultRoute
- type: vlan
  device: nic1
  vlan_id:
    get_param: InternalApi1NetworkVlanID
  mtu:
    get_param: InternalApi1NetworkMtu
  addresses:
  - ip_netmask:
      get_param: InternalApi1IpSubnet
    routes:
    -
      ip_netmask:
        get_param: InternalApiSupernet
      next_hop:
        get_param: InternalApi1InterfaceDefaultRoute
- type: interface
  name: nic2
  use_dhcp: false
- type: linux_bond
  name: bond0
  use_dhcp: false
  bonding_options: "mode=802.3ad xmit_hash_policy=layer3+4 lacp_rate=fast updelay=1000
miimon=100"
  mtu:
    get_param: Tenant1NetworkMtu
  members:
  - type: interface
    name: nic3

```

```

    primary: true
    mtu:
      get_param: Tenant1NetworkMtu
  - type: interface
    name: nic4
    mtu:
      get_param: Tenant1NetworkMtu
  - type: vlan
    vlan_id:
      get_param: Tenant1NetworkVlanID
    device: bond0
  - type: contrail_vrouter
    name: vhost0
    use_dhcp: false
    members:
      -
        type: interface
        name:
          str_replace:
            template: vlanVLANID
          params:
            VLANID: {get_param: Tenant1NetworkVlanID}
        use_dhcp: false
    addresses:
      - ip_netmask:
          get_param: Tenant1IpSubnet
        mtu:
          get_param: Tenant1NetworkMtu
    routes:
      -
        ip_netmask:
          get_param: TenantSupernet
        next_hop:
          get_param: Tenant1InterfaceDefaultRoute

ComputeDpdk0Hw2_network_config:
  - type: interface
    name: nic1
    dns_servers:
      get_param: DnsServers
    use_dhcp: false
    mtu:
      get_param: ControlPlaneMtu

```

```

addresses:
- ip_netmask:
  list_join:
  - '/'
  - - get_param: ControlPlaneIp
    - get_param: ControlPlaneSubnetCidr
routes:
-
  default: True
  next_hop:
    get_param: ControlPlaneDefaultRoute
- type: vlan
  device: nic1
  vlan_id:
    get_param: Storage0NetworkVlanID
  mtu:
    get_param: Storage0NetworkMtu
  addresses:
  - ip_netmask:
    get_param: Storage0IpSubnet
  routes:
  -
    ip_netmask:
      get_param: StorageSupernet
    next_hop:
      get_param: Storage0InterfaceDefaultRoute
- type: vlan
  device: nic1
  vlan_id:
    get_param: InternalApi0NetworkVlanID
  mtu:
    get_param: InternalApi0NetworkMtu
  addresses:
  - ip_netmask:
    get_param: InternalApi0IpSubnet
  routes:
  -
    ip_netmask:
      get_param: InternalApiSupernet
    next_hop:
      get_param: InternalApi0InterfaceDefaultRoute
- type: interface
  name: nic2

```

```

    use_dhcp: false
- type: contrail_vrouter_dpd
  name: vhost0
  vlan_id:
    get_param: Tenant0NetworkVlanID
  driver: "{{ overcloud['contrail']['vrouter']['dpdk']['driver'] }}"
  bond_mode: 4
  bond_policy: layer2+3
  cpu_list: 1,2
  members:
    - type: interface
      name: nic3
    - type: interface
      name: nic4
  addresses:
    - ip_netmask:
        get_param: Tenant0IpSubnet
  mtu:
    get_param: Tenant0NetworkMtu
  routes:
    -
      ip_netmask:
        get_param: TenantSupernet
      next_hop:
        get_param: Tenant0InterfaceDefaultRoute

ComputeDpdk1Hw3_network_config:
- type: interface
  name: nic1
  dns_servers:
    get_param: DnsServers
  use_dhcp: false
  mtu:
    get_param: ControlPlaneMtu
  addresses:
    - ip_netmask:
        list_join:
          - '/'
          - - get_param: ControlPlaneIp
            - get_param: ControlPlaneSubnetCidr
  routes:
    -
      default: True

```

```

    next_hop:
        get_param: ControlPlaneDefaultRoute
- type: vlan
    device: nic1
    vlan_id:
        get_param: Storage1NetworkVlanID
    mtu:
        get_param: Storage1NetworkMtu
    addresses:
- ip_netmask:
        get_param: Storage1IpSubnet
    routes:
-
        ip_netmask:
            get_param: StorageSupernet
        next_hop:
            get_param: Storage1InterfaceDefaultRoute
- type: vlan
    device: nic1
    vlan_id:
        get_param: InternalApi1NetworkVlanID
    mtu:
        get_param: InternalApi1NetworkMtu
    addresses:
- ip_netmask:
        get_param: InternalApi1IpSubnet
    routes:
-
        ip_netmask:
            get_param: InternalApiSupernet
        next_hop:
            get_param: InternalApi1InterfaceDefaultRoute
- type: interface
    name: nic2
    use_dhcp: false
- type: contrail_vrouter_dpdn
    name: vhost0
    vlan_id:
        get_param: Tenant1NetworkVlanID
    driver: "{ overcloud['contrail']['vrouter']['dpdk']['driver'] }"
    bond_mode: 4
    bond_policy: layer2+3
    cpu_list: 1,2

```

```

members:
- type: interface
  name: nic3
- type: interface
  name: nic4
addresses:
- ip_netmask:
    get_param: Tenant1IpSubnet
mtu:
  get_param: Tenant1NetworkMtu
routes:
-
  ip_netmask:
    get_param: TenantSupernet
  next_hop:
    get_param: Tenant1InterfaceDefaultRoute

CephStorage0Hw6_network_config:
- type: interface
  name: nic1
  dns_servers:
    get_param: DnsServers
  use_dhcp: false
  mtu:
    get_param: ControlPlaneMtu
  addresses:
  - ip_netmask:
      list_join:
      - '/'
      - - get_param: ControlPlaneIp
        - get_param: ControlPlaneSubnetCidr
  routes:
  -
    default: True
    next_hop:
      get_param: ControlPlaneDefaultRoute
- type: vlan
  device: nic1
  vlan_id:
    get_param: Storage0NetworkVlanID
  mtu:
    get_param: Storage0NetworkMtu
  addresses:

```

```

- ip_netmask:
    get_param: Storage0IpSubnet
routes:
-
    ip_netmask:
        get_param: StorageSupernet
    next_hop:
        get_param: Storage0InterfaceDefaultRoute
- type: vlan
  device: nic1
  vlan_id:
    get_param: StorageMgmt0NetworkVlanID
  mtu:
    get_param: StorageMgmt0NetworkMtu
  addresses:
  - ip_netmask:
      get_param: StorageMgmt0IpSubnet
    routes:
    -
        ip_netmask:
            get_param: StorageMgmtSupernet
        next_hop:
            get_param: StorageMgmt0InterfaceDefaultRoute
- type: interface
  name: nic2
  use_dhcp: false
- type: interface
  name: nic3
  use_dhcp: false
- type: interface
  name: nic4
  use_dhcp: false

CephStorage1Hw7_network_config:
- type: interface
  name: nic1
  dns_servers:
    get_param: DnsServers
  use_dhcp: false
  mtu:
    get_param: ControlPlaneMtu
  addresses:
  - ip_netmask:

```



```

    list_join:
      - '/'
      - - get_param: ControlPlaneIp
        - get_param: ControlPlaneSubnetCidr
    routes:
      -
        default: True
        next_hop:
          get_param: ControlPlaneDefaultRoute
- type: vlan
  device: nic1
  vlan_id:
    get_param: Storage1NetworkVlanID
  mtu:
    get_param: Storage1NetworkMtu
  addresses:
    - ip_netmask:
        get_param: Storage1IpSubnet
    routes:
      -
        ip_netmask:
          get_param: StorageSupernet
        next_hop:
          get_param: Storage1InterfaceDefaultRoute
- type: vlan
  device: nic1
  vlan_id:
    get_param: StorageMgmt1NetworkVlanID
  mtu:
    get_param: StorageMgmt1NetworkMtu
  addresses:
    - ip_netmask:
        get_param: StorageMgmt1IpSubnet
    routes:
      -
        ip_netmask:
          get_param: StorageMgmtSupernet
        next_hop:
          get_param: StorageMgmt1InterfaceDefaultRoute
- type: interface
  name: nic2
  use_dhcp: false
- type: interface

```

```

    name: nic3
    use_dhcp: false
  - type: interface
    name: nic4
    use_dhcp: false

```

Sample compute-nodes.yml Configuration File

```

# Copyright 2023 Juniper Networks, Inc. All rights reserved.
# Licensed under the Juniper Networks Script Software License (the "License").
# You may not use this script file except in compliance with the License, which is located at
# http://www.juniper.net/support/legal/scriptlicense/
# Unless required by applicable law or otherwise agreed to in writing by the parties,
# software distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
#
#
#

# Each list item contains:
#   name: name of a node in the inventory (string)
#   profile: name of hardware profile, group of servers (optional, string)
#   leaf: leaf name (optional, string)

# List of nodes to use as compute role using Contrail DPDK vRouter
compute_nodes_dpdk:
  - name: computedpdk1
    leaf: '0'
    profile: hw2
  - name: computedpdk2
    leaf: '1'
    profile: hw3

# List of nodes to use as compute role using Sriov
compute_nodes_sriov:
  - name: computesriov1
    leaf: '0'
    profile: hw4
  - name: computesriov2

```

```

    leaf: '1'
    profile: hw5

# List of nodes to use as compute role using Contrail kernel vRouter
compute_nodes_kernel:
  - name: compute1
    leaf: '0'
    profile: hw0
  - name: compute2
    leaf: '0'
    profile: hw1
  - name: compute3
    leaf: '1'
    profile: hw1
  - name: compute4
    leaf: '1'
    profile: hw0

```

Sample storage-nodes.yml Configuration File

```

# Copyright 2023 Juniper Networks, Inc. All rights reserved.
# Licensed under the Juniper Networks Script Software License (the "License").
# You may not use this script file except in compliance with the License, which is located at
# http://www.juniper.net/support/legal/scriptlicense/
# Unless required by applicable law or otherwise agreed to in writing by the parties,
# software distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
#
#
#

# List of nodes to use as storage host role
# List item contains:
#   name: name of a node in the inventory (string)
#   profile: name of hardware profile, group of servers (optional, string)
#   leaf: leaf name (optional, string)

storage_nodes:
  - name: storage1

```

```

    leaf: '0'
    profile: hw6
  - name: storage2
    leaf: '0'
    profile: hw6
  - name: storage3
    leaf: '1'
    profile: hw7

```

Sample K8's-host-nodes.yml Configuration File

```

# Copyright 2023 Juniper Networks, Inc. All rights reserved.
# Licensed under the Juniper Networks Script Software License (the "License").
# You may not use this script file except in compliance with the License, which is located at
# http://www.juniper.net/support/legal/scriptlicense/
# Unless required by applicable law or otherwise agreed to in writing by the parties,
# software distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
#
#
# +-----+ +-----+ +-----+
# |Undercloud/Jumphost| |OpenStack Ctrl| |OpenStack Compute|
# +-----+ +-----+ +-----+
# |Provision & deploy | |Neutron      | |vrouter        |
# +-+-----+ |Heat      | |          |
# |           | |Keystone   | |          |
# |           | +-+-----+ |          |
# |           | |          | +-+-----+
# |           | |          | | |
# |           | |          | | |
# |           | |          | | +---+-----+
tenant_ip_netmask-----+
# |           |          | |          |
# |           |          | |          |
# |           | +-----+ +-----+ +-----+ +-----+
internalapi_ip_netmask-----+
# |           |          | |          | |          |
# |           |          | |          | |          |
# |           | +-----+ +-----+ +-----+

```



```

#           network to provision itself. K8s components utilises
#           this network for internal communication.
#
# * tenant_ip_netmask: IP address in CIDR notation in tenant network.
#           Through this network vrouter will communicate with
#           Contrail control.
#
# * external_ip_netmask: IP address in CIDR notation in external network.
#           This network will be used to setup external VIP
#           managed by keepalived. Haproxy will be configured
#           to expose services like Command, Contrail web ui and
#           k8s api through this VIP.
#
# * extra keys: nodes can define variables used later in network configuration
#           like dns servers, default gw etc.
common: &common
  dns_server1: "8.8.4.4"
  dns_server2: "8.8.8.8"
  gw: "192.2.0.254"

k8s_host_nodes:
  - name: k8s-contrail1
    hypervisor: controler1
    control_ip_netmask: "192.168.213.21/24"
    internalapi_ip_netmask: "172.16.0.21/24"
    tenant_ip_netmask: "172.17.131.21/24"
    external_ip_netmask: "192.2.0.230/25"
    <<: *common
  - name: k8s-contrail2
    hypervisor: controler2
    control_ip_netmask: "192.168.213.22/24"
    internalapi_ip_netmask: "172.16.0.22/24"
    tenant_ip_netmask: "172.17.131.22/24"
    external_ip_netmask: "192.2.0.231/25"
    <<: *common
  - name: k8s-contrail3
    hypervisor: controler3
    control_ip_netmask: "192.168.213.23/24"
    internalapi_ip_netmask: "172.16.0.23/24"
    tenant_ip_netmask: "172.17.131.23/24"
    external_ip_netmask: "192.2.0.232/25"
    <<: *common

```

```

# Template for network layout on all kubernetes cluster nodes
# This follows the syntax
# https://cloudinit.readthedocs.io/en/latest/topics/network-config-format-v1.html#network-config-v1
# or
# https://cloudinit.readthedocs.io/en/latest/topics/network-config-format-v2.html#network-config-v2
# variables from k8s_host_nodes can be referred with "{{ host.<variable> }}"
k8s_host_nodes_network_config:
  version: 1
  config:
    - type: physical
      name: enp1s0
      subnets:
        - type: static
          address: "{{ host.control_ip_netmask }}"
    - type: vlan
      name: "enp1s0.{{ overcloud.network.internal_api.vlan }}"
      vlan_link: enp1s0
      vlan_id: "{{ overcloud.network.internal_api.vlan }}"
      subnets:
        - type: static
          address: "{{ host.internalapi_ip_netmask }}"
          routes:
            - gateway: "{{ overcloud.network.internal_api.gateway }}"
              network: "{{ overcloud.network.internal_api.supernet | ipaddr('network') }}"
              netmask: "{{ overcloud.network.internal_api.supernet | ipaddr('netmask') }}"
    - type: physical
      name: enp2s0
      subnets:
        - type: static
          address: "{{ host.external_ip_netmask }}"
          gateway: "{{ host.gw }}"
          dns_nameservers:
            - "{{ host.dns_server1 }}"
            - "{{ host.dns_server2 }}"
    - type: physical
      name: enp3s0
    - type: vlan
      name: "enp3s0.{{ overcloud.network.tenant.vlan }}"
      vlan_link: enp3s0
      vlan_id: "{{ overcloud.network.tenant.vlan }}"
      subnets:

```

```

- type: static
  address: "{{ host.tenant_ip_netmask }}"
  routes:
  - gateway: "{{ overcloud.network.tenant.gateway }}"
    network: "{{ overcloud.network.tenant.supernet | ipaddr('network') }}"
    netmask: "{{ overcloud.network.tenant.supernet | ipaddr('netmask') }}"

```

Sample vault-data.yml Configuration File

```

# This config structure can be used to hold information that needs to be encrypted for privacy
# If there is a password stored in /var/lib/contrail_cloud/config/.vault_password then it will
be used
# Otherwise the password can be entered interactively
#
# This file can be edited with the "ansible-vault edit" command
# This file can be re-encrypted with a new password with the "ansible-vault rekey" command
vault:
  global:
    rhel:
      # Contrail Cloud Activation Key
      satellite:
        #SATELLITE_KEY
        key: "PUT_YOUR_KEY_HERE"
      # User account used for all Contrail Cloud automation
      # This account will be created on:
      #   - jumphost
      #   - control hosts
      #   - all overcloud roles
      #   - appformix controllers
    service_user:
      # Account Name
      name: "contrail"
      # Account Password
      password: "c0ntrail123"
      # Passphrase used to encrypt ssh key of service user.
      # If not defined ssh private key will not be encrypted.
      # ssh_key_passphrase: "c0ntrail123"
  rhvm:
    vm:

```



```

# rhvm user name
user: "contrail"
# password for the rhvm vm user
password: "c0ntrail123"
# root password for the rhvm VM
root_password: "c0ntrail123"
# webui admin password
admin_password: "c0ntrail123"
# Passphrase used to encrypt ssh key of rhvm user.
# If not defined ssh private key will not be encrypted.
# ssh_key_passphrase: "c0ntrail123"
vnc:
    # VNC console password for the rhvm VM
    password: "contrail123"
undercloud:
    #Administrator password - default is randomly generated
    #admin_password: "c0ntrail123"
vm:
    # undercloud user name
    user: "stack"
    # password for the undercloud vm user
    password: "contrail123"
    # root password for the undercloud VM
    root_password: "contrail123"
    # Passphrase used to encrypt ssh key of undercloud user.
    # If not defined ssh private key will not be encrypted.
    # ssh_key_passphrase: "c0ntrail123"
vnc:
    # VNC console password for the undercloud VM
    password: "contrail123"
overcloud:
    #Administrator password
    admin_password: "c0ntrail123"
    # Root password used for local login to overcloud nodes through console
    # root_password: "contrail123"
contrail:
    rabbitmq:
        # contrail rabbitmq user name
        user: "contrail_rabbitmq"
        # contrail rabbitmq user password
        password: "c0ntrail123"
registry:
    local_instance:

```

```

    openstack:
      username: "contrail"
      password: "c0ntrail123"
    contrail:
      username: "contrail"
      password: "c0ntrail123"
    contrail-insights:
      username: "contrail"
      password: "c0ntrail123"
control_hosts:
  vm:
    vnc:
      # VNC console password for all control VMs
      password: "contrail123"
appformix:
  mysql:
    # Approfmix MySQL user account
    user: "appformix"
    # Approfmix MySQL user password
    password: "c0ntrail123"
  rabbitmq:
    # Approfmix RabbitMQ user account
    user: "appformix"
    # Approfmix RabbitMQ user password
    password: "c0ntrail123"
# Credentials used to connect external ceph cluster
#ceph_external:
#  client_key: "CLIENT_KEY"
#  client_user: "openstack"

# List of inventory hardware types that can hold hardware-specific properties
# You can create similar configurations to allow reference from inventory-nodes.yml
inventory_nodes:
  # A sample configuration for a hardware type
  hardware1:
    # IPMI user account for Ironic inventory resources
    pm_user: "ADMIN"
    # IPMI user password for Ironic inventory resources
    pm_password: "ADMIN"
  # A sample configuration for a hardware type
  hardware2:
    # IPMI user account for Ironic inventory resource
    pm_user: "admin"

```

```
# IPMI user password for Ironio inventory resource
pm_password: "admin"
# User defined sensitive data can be stored under 'other' key.
# Schema validation will only check if key,value format is used.
#other:
# mykey: myvalue
```

RELATED DOCUMENTATION

[Deploy Contrail Cloud](#) | 9

Appendix B: Remove a Ceph Storage Node

Use this procedure to remove a Ceph storage node from a Ceph cluster. Removing Ceph storage is handled as a Red Hat process rather than an end-to-end Contrail Cloud process. However, this procedure will demonstrate the removal of a storage node from an environment in the context of Contrail Cloud.

Before you begin, ensure that the remaining nodes in the cluster is sufficient for keeping the required amount of pgs and replicas for your Ceph storage cluster. Ensure that both Ceph cluster and overcloud stack are healthy. For checking the health of your overcloud, see [Verify Quorum and Node Health](#).

All examples in this procedure come from a lab setting to demonstrate storage removal within the context of Contrail Cloud. Sample output in the provided examples will differ from the information in your specific cloud deployment. In the examples used for this procedure, “storage3” will be the targeted node for removal.

Remove the storage node:

1. Find the connection between the bare metal server and the overcloud server. The output from the command below shows us that the server we are looking for is “overcloud8st-cephstorageblue1-0”. This information will be used later in the procedure.

```
(undercloud) [stack@undercloud ~]$ openstack ccloud nodemap list
```

Name	IP	Hypervisor	Hypervisor IP
overcloud8st-afxctrl-0	192.168.213.69	controler2	192.168.213.6
overcloud8st-afxctrl-1	192.168.213.52	controler3	192.168.213.7

```
| overcloud8st-afxctrl-2 | 192.168.213.58 | controler1 | 192.168.213.5 |
| overcloud8st-ctrl-0 | 192.168.213.73 | controler2 | 192.168.213.6 |
| overcloud8st-ctrl-1 | 192.168.213.63 | controler1 | 192.168.213.5 |
| overcloud8st-ctrl-2 | 192.168.213.59 | controler3 | 192.168.213.7 |
| overcloud8st-cephstorageblue1-0 | 192.168.213.62 | storage3 | 192.168.213.62 |
| overcloud8st-compdpdk-0 | 192.168.213.56 | compute1 | 192.168.213.56 |
| overcloud8st-cephstorageblue2-0 | 192.168.213.61 | storage2 | 192.168.213.61 |
| overcloud8st-cephstorageblue2-1 | 192.168.213.80 | storage1 | 192.168.213.80 |
+-----+-----+-----+-----+
```

2. From the undercloud as the heat-admin user, SSH to any of the openstack controllers and then run `sudo ceph -s` to verify that the Ceph cluster is healthy:

```
[root@overcloud8st-ctrl-1 ~]# sudo ceph -s
cluster:
  id:      a98b1580-bb97-11ea-9f2b-525400882160
  health: HEALTH_OK
```

3. Find the OSDs that reside on the server to be removed (overcloud8st-cephstorageblue1-0). We identify osd.2, osd.3, osd.6, and osd.7 from the example below:

```
[root@overcloud8st-ctrl-1 ~]# sudo ceph osd tree
ID CLASS WEIGHT  TYPE NAME                STATUS REWEIGHT PRI-AFF
-1      10.91638 root default
-3      3.63879  host overcloud8st-cephstorageblue1-0
  2  hdd  0.90970      osd.2                up  1.00000 1.00000
  3  hdd  0.90970      osd.3                up  1.00000 1.00000
  6  hdd  0.90970      osd.6                up  1.00000 1.00000
  7  hdd  0.90970      osd.7                up  1.00000 1.00000
-7      3.63879  host overcloud8st-cephstorageblue2-0
  1  hdd  0.90970      osd.1                up  1.00000 1.00000
  4  hdd  0.90970      osd.4                up  1.00000 1.00000
  8  hdd  0.90970      osd.8                up  1.00000 1.00000
 10  hdd  0.90970      osd.10               up  1.00000 1.00000
-5      3.63879  host overcloud8st-cephstorageblue2-1
  0  hdd  0.90970      osd.0                up  1.00000 1.00000
  5  hdd  0.90970      osd.5                up  1.00000 1.00000
  9  hdd  0.90970      osd.9                up  1.00000 1.00000
 11  hdd  0.90970      osd.11               up  1.00000 1.00000
```

4. While still logged in to the openstack controller, mark osd.2, osd.3, osd.6, and osd.7 as non-operational:

```
[root@overcloud8st-ctrl-1 ~]# sudo ceph osd out 2
marked out osd.2.
[root@overcloud8st-ctrl-1 ~]# sudo ceph osd out 3
marked out osd.3.
[root@overcloud8st-ctrl-1 ~]# sudo ceph osd out 6
marked out osd.6.
[root@overcloud8st-ctrl-1 ~]# sudo ceph osd out 7
marked out osd.7.
```

From the undercloud as the heat-admin user, SSH to any of the openstack controllers and then run `sudo ceph -s` to verify that the Ceph cluster returns a “health_ok” state before you continue.

5. From the undercloud as the heat-admin user, SSH to Ceph node overcloud8st-cephstorageblue1-0, and stop the OSD services:

```
[root@overcloud8st-cephstorageblue1-0 ~]# sudo systemctl stop ceph-osd@2.service
[root@overcloud8st-cephstorageblue1-0 ~]# sudo systemctl stop ceph-osd@3.service
[root@overcloud8st-cephstorageblue1-0 ~]# sudo systemctl stop ceph-osd@6.service
[root@overcloud8st-cephstorageblue1-0 ~]# sudo systemctl stop ceph-osd@7.service
```

From the undercloud as the heat-admin user, SSH to any of the openstack controllers and then run `sudo ceph -s` to verify that the Ceph cluster returns a “health_ok” state before you continue.

6. From the undercloud as the heat-admin user, SSH back into the controller and remove further information about the OSDs from overcloud8st-cephstorageblue1-0:

```
[root@overcloud8st-ctrl-1 ~]# sudo ceph osd crush remove osd.2
removed item id 2 name 'osd.2' from crush map
[root@overcloud8st-ctrl-1 ~]# sudo ceph osd crush remove osd.3
removed item id 3 name 'osd.3' from crush map
[root@overcloud8st-ctrl-1 ~]# sudo ceph osd crush remove osd.6
removed item id 6 name 'osd.6' from crush map
[root@overcloud8st-ctrl-1 ~]# sudo ceph osd crush remove osd.7
removed item id 7 name 'osd.7' from crush map

[root@overcloud8st-ctrl-1 ~]# sudo ceph auth del osd.2
updated
[root@overcloud8st-ctrl-1 ~]# sudo ceph auth del osd.3
```

```

updated
[root@overcloud8st-ctrl-1 ~]# sudo ceph auth del osd.6
updated
[root@overcloud8st-ctrl-1 ~]# sudo ceph auth del osd.7
updated

[root@overcloud8st-ctrl-1 ~]# sudo ceph osd rm 2
removed osd.2
[root@overcloud8st-ctrl-1 ~]# sudo ceph osd rm 3
removed osd.3
[root@overcloud8st-ctrl-1 ~]# sudo ceph osd rm 6
removed osd.6
[root@overcloud8st-ctrl-1 ~]# sudo ceph osd rm 7
removed osd.7

[root@overcloud8st-ctrl-1 ~]# sudo ceph osd crush rm overcloud8st-cephstorageblue1-0

```

From the undercloud as the heat-admin user, SSH to any of the openstack controllers and then run `sudo ceph -s` to verify that the Ceph cluster returns a “health_ok” state before you continue.

7. From the undercloud VM find the ID of the Ceph storage node:

```

(undercloud) [stack@undercloud ~]$ openstack server list | grep overcloud8st-
cephstorageblue1-0
| 7ee9be4f-efda-4837-a597-a6554027d0c9 | overcloud8st-cephstorageblue1-0 | ACTIVE |
ctlplane=192.168.213.62 | overcloud-full | CephStorageBlue1

```

8. Initiate a removal using the node ID from the previous step:

```

(undercloud) [stack@undercloud ~]$ openstack overcloud node delete --stack overcloud
7ee9be4f-efda-4837-a597-a6554027d0c9

```

From the undercloud as the heat-admin user, SSH to any of the openstack controllers and then run `sudo ceph -s` to verify that the Ceph cluster returns a “health_ok” state before you continue.

9. Verify that the bare metal node is in a state of power off and available:

```

(undercloud) [stack@undercloud ~]$ openstack baremetal node list | grep storage3
| 05bbab4b-b968-4d1d-87bc-a26ac335303d | storage3 | None | power off | available | False |

```

10. From the jump host as the `contrail` user mark the storage node with 'status: deleting' so the Ceph profile will be removed from it. Add the 'status: deleting' to the `storage-nodes.yml` file for storage3 and then run the script `storage-nodes-assign.sh`.

```
[contrail@5a6s13-node1 contrail_cloud]$ cat config/storage-nodes.yml
storage_nodes:
  - name: storage1
    profile: blue2
  - name: storage2
    profile: blue2
  - name: storage3
    profile: blue1
    status: deleting

[contrail@5a6s13-node1 contrail_cloud]$ ./scripts/storage-nodes-assign.sh
```

From the undercloud as the `heat-admin` user, SSH to any of the openstack controllers and then run `sudo ceph -s` to verify that the Ceph cluster returns a "health_ok" state before you continue.

11. From the jump host as the `contrail` user, run `openstack-deploy.sh` to regenerate the templates to reflect the current state:

```
[contrail@5a6s13-node1 contrail_cloud]$ ./scripts/openstack-deploy.sh
```

From the undercloud as the `heat-admin` user, SSH to any of the openstack controllers and then run `sudo ceph -s` to verify that the Ceph cluster returns a "health_ok" state before you continue.

If the goal is to remove the bare metal node completely, use the following additional procedure:

1. Edit the `config/storage-nodes.yml` file and remove the bare metal node.
2. Edit the `inventory.yml` file and include the 'status: deleting' to the node to be removed:

```
[contrail@5a6s13-node1 contrail_cloud]$ cat config/inventory.yml
...
inventory_nodes:
  - name: "storage3"
    pm_addr: "10.84.129.184"
    status: deleting
    <<: *common
```

3. Run the `inventory-assign.sh` script:

```
[contrail@5a6s13-node1 contrail_cloud]$ ./scripts/inventory-assign.sh
```

From the undercloud as the `heat-admin` user, SSH to any of the openstack controllers and then run `sudo ceph -s` to verify that the Ceph cluster returns a “health_ok” state before you continue.

4. Verify the bare metal node has been removed. Enter the following command to view the list of nodes:

```
(undercloud) [stack@undercloud ~]$ openstack ccloud nodemap list |grep storage  
| overcloud8st-cephstorageblue2-1 | 192.168.213.80 | storage1 | 192.168.213.80 |  
| overcloud8st-cephstorageblue2-0 | 192.168.213.61 | storage2 | 192.168.213.61 |
```

RELATED DOCUMENTATION

[Removing a Node from the Overcloud](#)

[Handling a Node Failure](#)

[Reference Architecture for Contrail Cloud](#)