Installing Cisco VTS 2.6.1 Components in OpenStack using Red Hat Enterprise Linux OpenStack Director

The following sections provide details about installing Cisco VTS 2.6.1 components in OpenStack using Red Hat Enterprise Linux OpenStack Director.

Installing Cisco VTS Components in OpenStack using Red Hat Enterprise Linux OpenStack Director

The Red Hat OpenStack Platform director (RHOSPD) is a toolset for installing and managing a complete OpenStack environment. It is based primarily on the OpenStack project TripleO, which is the abbreviation for OpenStack-On-OpenStack. Redhat also has a program for partners to integrate their solution into OpenStack deployment using the framework provided by Red Hat OpenStack Platform director.

Cisco VTS follows the Red Hat Partner Integration document to introduce VTS specific content into the OpenStack installation. See Red Hat OpenStack Platform 10 Partner Integration document for details. As of this release (VTS 2.6.0), the integration has been qualified with Red Hat OpenStack 10 platform (corresponding to OpenStack Newton Release).

Installation and setup of the director node and the necessary networking needed to manage the hardware (that would take roles of Controller, Compute, or Storage) is documented in the Red Hat documentation referenced above. Note that these procedure are dependent on the type of hardware used and the specific configuration of each deployment. If the deployment involves hosting NFV workloads, additional configuration is needed for reserving CPU capacity, huge-pages, and libvirt settings. This needs to be taken into consideration. Red Hat documentation on NFV provides an overview of these configuration options. See the Red Hat OpenStack Platform 10 Network Functions Virtualization Configuration Guide for details.

Prerequisites

Ensure that:

- The director node is equipped with the right set of software for undercloud installation. See Installing the Undercloud chapter of the Red Hat OpenStack Platform 10 Director Installation and Usage document, for details.
- You perform the node introspection procedures. See Configuring Basic Overcloud Requirements with the CLI Tools chapter of the Red Hat OpenStack Platform 10 Director Installation and Usage document, for details.
- The packages needed for overcloud image installation are obtained from the director node using:
  ```
  sudo yum install rhosp-director-images rhosp-director-images-ipa
  ```
  The overcloud-full image is the standard OpenStack overcloud image shipped with OSPD 10. However, this would be customized with Cisco VTS RPMs.
- The OSPD deployment, undercloud, and overcloud nodes have access to the yum repositories and RHEL registration, including any proxy setup. See the Overcloud Registration chapter of the Red Hat OpenStack Platform 10 Advanced Overcloud Customization document, for details.

In order to integrate Cisco VTS components, the following steps are required:

- Install the Cisco VTS related RPMs in the overcloud image.
- Introduce the Heat templates and environmental files in the director, for VTS services
- Proceed with overcloud deployment including the Cisco VTS environmental files.

Note This document covers installation of only Cisco VTS packages into a default OSPD overcloud image. Any further changes to the overcloud image are outside the scope of the document and you need to see the Red Hat overcloud customization document for details.

Usage of HTTP/HTTPS Proxies—In deployments where HTTP/HTTPS proxy is in use, ensure that the director node's **http_proxy**, **https_proxy**, and **no_proxy** environment variables are set. Additionally, ensure that the overcloud nodes have their proxy settings set correctly. This is needed for performing overcloud package updates during steady-state operation. The latter is usually accomplished by following Red Hat's recommendation for RHEL registration See the Overcloud Registration chapter of the Red Hat OpenStack Platform 10 Advanced Overcloud Customization document, for details.

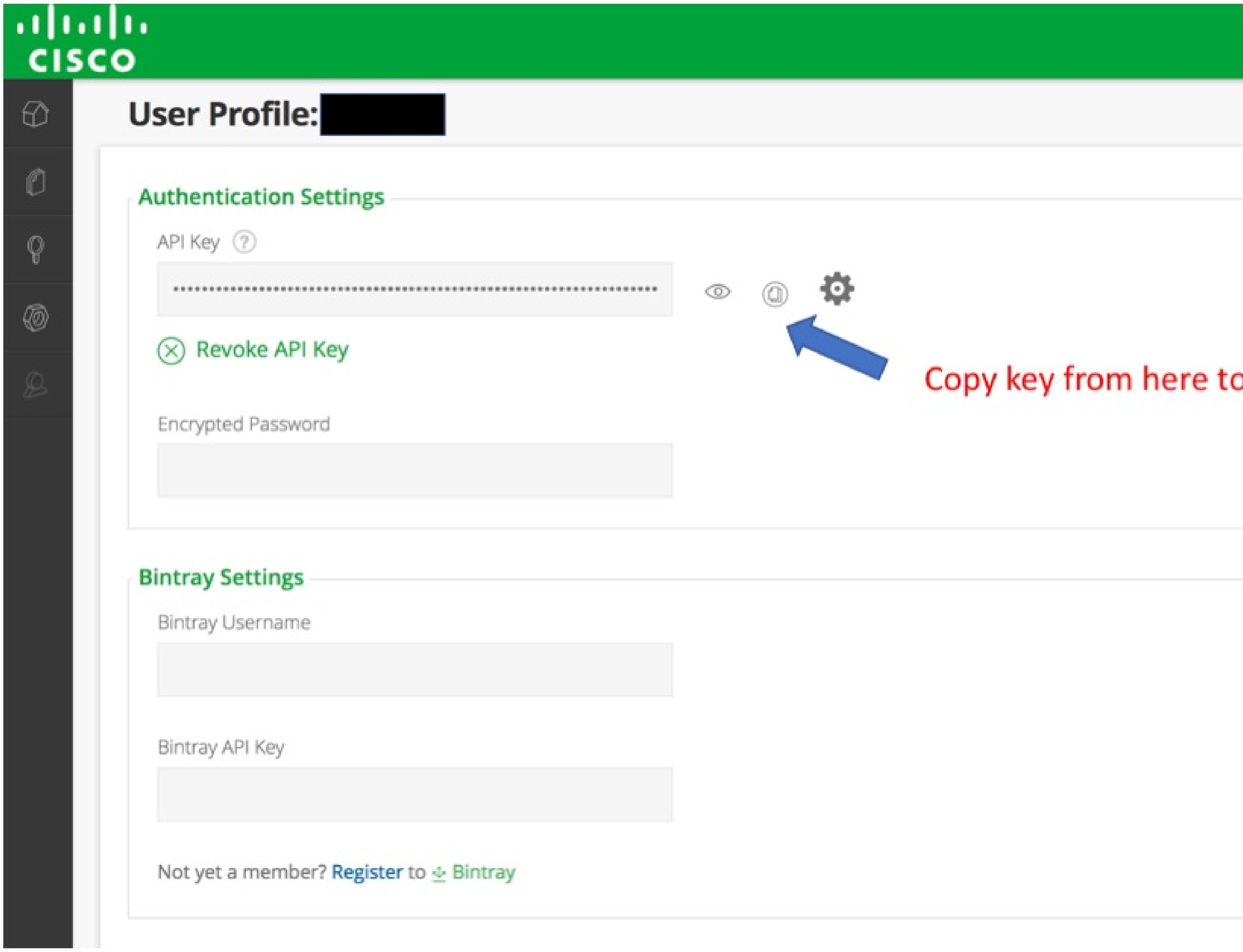
Installing and Configuring Cisco VTS Packages

Note This document is written assuming the OpenStack Overcloud nodes can retrieve VTS-specific packages from Cisco's YUM repository at devhub.cisco.com. The exact procedure may vary depending on the customer deployment preferences. Some deployments may have an intermediary satellite repository, which can host RPMs from multiple external YUM repositories. The satellite repository may host RPMs that have been thoroughly validated in a lab environment, prior to using them in production deployment.

Step 1 Install the vts-overcloud installer RPM on the director node.

Note To access the Cisco VTS YUM repositories, you need a cisco.com account that needs to be authorized to access the repository. Contact your Cisco Account team to request access, mentioning your cisco.com ID.

1 Obtain the VTS Repo credentials by logging in to https://devhub.cisco.com/.
2 Click the login name in the upper right corner and log in with CEC/SSO credentials.
3 Collect the access token generating an API key for use as the password. Click on the eye icon to view the API key.



User Profile: [redacted]

Authentication Settings

API Key

••

Revoke API Key

Encrypted Password

Bintray Settings

Bintray Username

Bintray API Key

Not yet a member? Register to Bintray

Copy key from here to

4 Run the "install-vts-helpers.sh" script, using sudo. This script will create a YUM configuration entry on the OSPD director node for the Cisco devhub repository with the collected credentials, and install the overcloud image customization helper script ('install-vts-in-overcloud'), the password utility ('encpwd'), and Cisco VTS Heat templates package.

```
sudo ./install-vts-helpers.sh --RepoUser=<username> --RepoPass=<devhub token> --Proxy=<proxyIP:port>
```

The **install-vts-helpers.sh** script is available in the Appendix appendix.

Step 2 Install cisco-vts packages in the OSPD overcloud image using virt-install wrapped in the Cisco package install helper script.

```
sh-4.2# pwd
```

```
/home/stack/image
sh-4.2# install-vts-in-overcloud --RepoUser=<username> --RepoPass=<devhub token>
--RhelPool=<pool> --RhelUser=<rhel_user> --RhelPass=<rhel_password>
```

Note Proxy settings should be also be used if the system is not configured for the proxy.

Step 3 Upload/Update the overcloud image in glance. Upload the modified image by running the following command from the directory containing the overcloud image:

Note Run the **source stackrc** command and unset any http_proxy setting before you run the following command.

```
openstack overcloud image upload --image-path ./ --update-existing
```

Step 4 On the director, copy the following environment file into your ~/templates directory (The Heat templates were installed in Step 2).

```
cp /usr/share/openstack-tripleo-heat-templates/environments/neutron-cisco-vts.yaml ~/templates
```

Note When deploying add it to the heat deploy command or to the deploy_overcloud.sh scriptwith a -e option.

Step 5 This step involves customizing the environment files neutron-cisco-vts.yaml for different modes of deployments. See section Configuring the neutron-cisco-vts.yaml File, on page 5.

For purpose of illustration, this document includes two sample configurations of neutron-cisco-vts.yaml:

- Sample configuration for deploying Cisco VTS with OVS on Compute
- Sample configuration for deploying Cisco VTS with VTF on Controller/Compute

See appendix Sample neutron-cisco-vts.yaml Configuration, for details.

Step 6 This step involves editing the nic configuration for controllers and compute nodes to designate specific interfaces, bridge settings, bonding configuration and other values.

- For pure OVS based deployments, See Configuring Basic Overcloud Requirements with the CLI Tools chapter of the Red Hat OpenStack Platform 10 Director Installation and Usage document for details about overcloud customization.
- For pure VTF based deployments, the nic-config/compute.yaml and nic-config/controller.yaml files (by default, available at /home/stack/templates) for a specific deployment needs to incorporate the following VPP interface types:

```
For a VPP bonded interface following list item will be added to controller and/or compute node
 nic-templates.
           -
             type: vpp_bond
             name: eth_bond101
             bonding_options: "mode=2,xmit_policy=l34"
             members:
               -
                 type: vpp_interface
                 name: enp7s0f0
                 uio_driver: vfio-pci
                 options: vlan-strip-offload off
               -
                 type: vpp_interface
                 name: enp7s0f1
                 uio_driver: vfio-pci
                 options: vlan-strip-offload off

For a single interface, following list item will be added to controller and/or compute nodes
```

```
-
  type: vpp_interface
  name: enp7s0f0
  uio_driver: uio_pci_generic
```

Step 7 The activation of VTS-specific services is controlled by adding these services to the role list defined for each node. Enable VTS-specific services on the controller or compute nodes depending on the type of deployment.

- For OVS deployment—The following lines need to be added to the roles_data.yaml file found usually in the director node under /usr/share/openstack-tripleo-heat-templates:

```
#Add VTS agent role for Controller
- OS::TripleO::Services::VTSAgent

#Add VTS agent role for Compute
- OS::TripleO::Services::VTSAgent
```

- For VTF deployment—The following lines need to be added to the roles_data.yaml file found usually in the director node under /usr/share/openstack-tripleo-heat-templates:

```
# For controllers

 - OS::TripleO::Services::VppController
 - OS::TripleO::Services::CiscoVpfaController


# For computes:

 - OS::TripleO::Services::VppCompute
 - OS::TripleO::Services::CiscoVpfaCompute
```

See Red Hat documentation on setting services on computes and controllers based on roles.
See the Troubleshooting Director Issues chapter of the Red Hat OpenStack Platform 10 Director Installation and Usage document, for troubleshooting information in case you face problems with the deployment.

Configuring the neutron-cisco-vts.yaml File

All of the configuration sections below apply to the neutron-cisco-vts environment file.

Cisco VTS Openstack Neutron ML2 plugin

All forms of VTS deployments rely on integration with OpenStack via the VTS ML2 Plugin. A general description of all relevant parameters to enable the ML2 Plugin is documented here.
Cisco VTS General Parameters

This section provides details about the Cisco VTS general parameters.

```
##################
### VTS General ###
### VTSPassword: 'Cisco123!'
```

```
###################

VTSUsername: 'admin'
VTSPassword: 'YLdKnf3qSsKA2JWQT9a0Sg=='
VTSServer: '30.30.30.3'
VTSVMMID: 'c958399f-5af6-3517-b902-771b7c1d37e6'
```

- VTSUsername/VTSPassword—Login name and password used by the Cisco VTS components in OpenStack in order to establish https connection with VTS. Password entered here is an encrypted form of the clear text password. For details about how to create an encrypted password, see Running the Password Encryption Script, on page 13.

- VTSService—IPv4/IPv6 address of the VTS. This is the virtual IP address assigned to the VTS VMs during VTS installation.

- VTSVMMID—This is the UUID associated with this OpenStack installation. VTS can manage multiple VMM instances. This UUID helps disambiguate the specific VMM instance. The UUID entered here can be generated by any UUID generator tool (For example; https://www.uuidgenerator.net/). However, the value entered in this field should match the UUID specified in the VTS UI as part of VMM registration.

Neutron ML2 Parameters

This section provides details about the Neutron ML2 parameters.

```
###################
### Neutron ML2 ###
###################
NeutronCorePlugin: 'neutron.plugins.ml2.plugin.Ml2Plugin'
NeutronMechanismDrivers: 'sriovnicswitch,cisco_vts'
NeutronTypeDrivers: 'vxlan,vlan,flat'
NeutronServicePlugins: 'cisco_vts_router,trunk'


#NeutronInterfaceDriver: ‘cisco_controller.drivers.agent.linux.interface.NamespaceDriver'
```

- NeutronCorePlugin—This is the Core neutron plugin for neutron tenant network. Default value is “neutron.plugins.ml2.plugin.Ml2Plugin”.
- NeutronMechanismDrivers—These are the mechanism drivers for neutron tenant network. To specify multiple values, use a comma-separated string. To enable VTS-specific mechanism driver, add cisco_vts to this list. For enabling SR-IoV interfaces on the compute, add sriovnicswitch.
- NeutronTypeDrivers—These are the tenant network types for neutron tenant network. To specify multiple values, use a comma-separated string.
- NeutronServicePlugins—This is the neutron service plugin for neutron tenant network. To enable L3 networking, add cisco_vts_router. To enable trunk mode operation (VLAN aware VMs) add trunk to the list of type drivers.
- NeutronInterfaceDriver—Specifies the interface driver used by the Neutron DHCP Agent. When deploying the VTF on nodes running the Neutron DHCP Agent this setting needs to be passed (uncommented). Valid values are (default) ‘neutron.agent.linux.interface.OVSInterfaceDriver’ and ‘cisco_controller.drivers.agent.linux.interface.NamespaceDriver'.

VTS Agent Configuration Parameters

This section provides details about the VTS Agent parameters.

```
########################
### VTS Agent Config ###
########################
VTSPhysicalNet: 'physnet101'
VTSRetries: 15
VTSTimeout:
VTSPollingInterval: 6
```

- VTSPhysicalNet—VTSPhysicalNet should be set to the ‘physnet’ used for the tenant networks for OVS on the compute. The environment file in the Heat templates should have the mapping of the tenant OVS bridge to the physnet name.
- VTSRetries—Number of times VTS agent retries a neutron request. Default is 15.
- VTSTimeout—Cisco VTS agent times out a request. Default value is 120 seconds.
- VTSPollingInterval—Cisco VTS agent polling interval for a request. Default value is 6 seconds.

Cisco VTS with VTF on Controller/Compute

VPFA Configuration Parameters

This section provides details about VPFA configuration parameters. These are mandatory to be configured

```
###################
### VPFA Config ###
###################
UnderlayIpNewtorksList: '21.0.0.0/8,10.10.10.0/24,50.50.0.0/16,40.40.0.0/16,42.42.42.0/24'

VTSR_u_IpAddressList: '10.10.10.133,10.10.10.134'
VPFAHostname: '<some_name>'
NetworkConfigMethod: 'static'
NetworkNameServerIP: ''
VifTypeCompute: 'vhostuser'
VifTypeController: 'tap'
```

- UnderlayIpNetworksList—List of underlay IP networks for VTF reachability. To specify multiple values, use a comma-separated string.
- VTSR_u_IpAddressList—List of underlay IP address assigned to VTS.
- VPFAHostname—Hostname assigned to VPFA.
- NetworkConfigMethod—VPFA network configuration method. Default value is “static”.
- NetworkNameServerIP—DNS IP assigned to VPFA.
- VifTypeCompute—VPFA VIF type for compute is “vhostuser”.
- VifTypeController—VPFA VIF type for Controller node is “tap”.

VPP Configuration Parameters

This section provides details about VPP configuration parameters.

```
#####################################
### VPP Configuration Parameters ###
#####################################
## MTU for Tun/tap interfaces
VppTunTapMtu: '9000'
##The CPUs listed below need to be part of the grub isol CPU list (configured elsewhere)
VppCpuMainCoreController: '6'
VppCpuMainCoreCompute: '6'
## Comma delimited workers list
VppCpuCorelistWorkersCompute: '7,8,9'
VppCpuCorelistWorkersController: '7,8,9'
## Avoid dumping vhost-user shared memory segments to core files
VppVhostUserDontDumpMem: True
```

Note All CPU values given above are examples and need to be adapted to the actual deployment, or left commented out (that is, these are optional).

- VppTunTapMtu—MTU for VPP tap interface.
- VppCpuMainCoreController—Pin VPP to a CPU core on controller.
- VppCpuMainCoreCompute—Pin VPP to a CPU core on compute.
- VppCpuCorelistWorkersCompute—Pin VPP worker threads to a CPU core on compute.
- VppCpuCorelistWorkersController—Pin VPP worker threads to a CPU core on controller
- VppVhostUserDontDumpMem—Do not dump vhost-user memory segments in core files.

PerNodeData Parameters

Collecting Node-specific UUID

1. Gather baremetal (ironic) UUID for overcloud nodes where VTF needs to be deployed.

   `“openstack baremetal node list”`

2. The node-specific hieradata is provisioned based on the node UUID, which is hardware dependent and immutable across reboots/reinstalls. Value returned will be unique and immutable machine UUID not related to the baremetal node UUID. Extract the machine unique UUID from the command below by substituting <baremetal-UUID> from the previous step. Run:

   ```
   “openstack baremetal introspection data save <baremetal-UUID> | jq
   .extra.system.product.uuid”
   ```

3. Populate “PerNodeData” parameters for each node where VTF is intended to be deployed in the neutron-cisco-vts.yaml. For example:

```
PerNodeData: |
     {
       "< Node1 UUID >": {
            "cisco_vpfa::vtf_underlay_ip_v4": "10.0.0.2",
            "cisco_vpfa::vtf_underlay_mask_v4": "24",
            "cisco_vpfa::network_ipv4_gateway": "10.0.0.1"},
       "< Node2 UUID >": {
            "cisco_vpfa::vtf_underlay_ip_v4": "10.0.0.3",
            "cisco_vpfa::vtf_underlay_mask_v4": "24",
            "cisco_vpfa::network_ipv4_gateway": "10.0.0.1"}
     }
```

- UUID—Immutable machine UUID derived from Step 2 for the overcloud node.
- “cisco_vpfa::vtf_underlay_ip_v4”—Underlay IPv4 address assigned to VTF.
- "cisco_vpfa::vtf_underlay_mask_v4"—Underlay IPv4 netmask assigned to VTF.
- "cisco_vpfa::network_ipv4_gateway"—Underlay IPv4 network gateway assigned to VTF.

Monit Agent Configuration

This section provides details about Monit agent configuration parameters.

```
##################################
### Monit Agent Configuration  ###
##################################

# IMPORTANT: To enable the Monit Agent config, add the VPFA specific
"OS::TripleO::Services::MonitVpfaAgent"
# or generic "OS::TripleO::Services::MonitAgent" to the corresponding nodes role data
configuration.


## General settings. Applied to all Monit Agents
## Credentials
MonitUser: ''
MonitPassword:
```

- MonitUser—The Monit username.
- MonitPassword—The Monit password.

collectd Agent Configuration

This section provides details about collectd Agent Configuration

```
######################################
### Collectd Agent Configuration  ###
######################################
```

```
# IMPORTANT: To enable the Collectd Agent config, add the
"OS::TripleO::Services::CollectDAgent"


##Enable or disable collectd (default is true)
# CollectDEnable: true

## Purge default/previous configurations
CollectDPurge: true

## CollectD Plugin configurations
## Each named plugin should have its own named dictionary entry, followed by a "content"
element containing the
## plugin's XML configuration stanza, in JSON list format.
## The configuration content is the native collectd configuration for the plugin
CollectDPluginConfigs: |
  {
    "memory":
    {
     "content":
      [
      "<Plugin memory>",
        "ValuesAbsolute true",
        "ValuesPercentage false",
      "</Plugin>"
      ]
    },
    "cpu":
    {
      "content":
      [
        "<Plugin cpu>",
           "ReportByCpu true",
           "ReportByState true",
           "ValuesPercentage false",
           "ReportNumCpu false",
           "ReportGuestState false",
           "SubtractGuestState true",
         "</Plugin>"
      ]
    },
    "python":
    {
      "content":
      [
        "<Plugin python>",
        "ModulePath \"/opt/cisco/vpe/collectd/\"",
        "LogTraces true",
        "Import \"cisco-vpfa-collectd-plugin\"",
        "</Plugin>"
      ]
    },
    "write_log":
    {
      "content":
      [
        "<Plugin write_log>",
        "Format JSON",
        "</Plugin>"
      ]
    },
    "interface":
    {
      "content":
      [
        "<Plugin interface>",
        "Interface \"br-ctlplane\"",
        "Interface \"br-ex\"",
        "Interface \"br-tenant\"",
        "Interface \"lo\"",
        "IgnoreSelected false",
        "ReportInactive true",
        "UniqueName false",
          "</Plugin>"
        ]
      },
      "disk":
      {
        "content":
        []
      },
      "load":
      {
        "content":
        [
          "<Plugin load>",
          "ReportRelative true",
          "</Plugin>"
        ]
      }
    }
```

- CollectDEnable—Enable or disable collectd, By default, this is set to true.
- CollectDPurge— Purge default or previous configurations. By default, this is set to true.
- CollectD Plugin configurations—Modify this for changing the VTF collectd plugin configuration. See *Monitoring Cisco VTS* chapter in the *Cisco VTS 2.6.1 User Guide* for details about collectd plugins.

Rsyslog settings for computes with VTF

Logs from VTF compute nodes can directed to a remote syslog server using rSyslog. To do this, certain parameters need to be configured the neutron-cisco-vts.yaml file. For example:

```
# IMPORTANT: Add OS::TripleO::Services::RSyslogClient to the role data catalogue for the
service to come into effect

# ****** EDIT THE SYSLOG SERVER IP ADDRESS AND PORT IN ClientLogFilters and add/remove
entries as needed! *******
#The default template below uses UDP (@) servers on port 514. To add a TCP server, add an
extra stanza prefixing
# with @@ the server's IP address

ClientLogFilters: | [
{
"expression": "$syslogfacility-text == 'local3' and $syslogseverity-text == 'crit'", "action":
 "@[192.168.128.2]:514;forwardFormat"
},
{
"expression": "$syslogfacility-text == 'local3' and $syslogseverity-text == 'err'", "action":
 "@[192.168.128.2]:514;forwardFormat"
},
{
"expression": "$syslogfacility-text == 'local3' and $syslogseverity-text == 'warning'",

"action": "@[192.168.128.2]:514;forwardFormat"
},

{
"expression": "$syslogfacility-text == 'local3' and $syslogseverity-text == 'info'", "action":
 "@[192.168.128.2]:514;forwardFormat"
}

]
```

Note In this example, 192.168.128.2 is the IP address of the Syslog server, and 514 is the UDP port.

Additionally, the rsyslog client service on the computes and controller may need to be enabled in the roles_data.yaml file.

```
##Add rsyslog client under Controller role:
- OS::TripleO::Services::RSyslogClient

#Add rsyslog client under Compute role:
- OS::TripleO::Services::RSyslogClient
```

Updating VTS RPMs in Overcloud

Ensure that the YUM repositories referred to by the Overcloud nodes contain the latest relevant set of RPMs. In case of deployments where Satellites are in use, the Satellite should contain the latest set of RPMs.
To be able to update packages, Red Hat recommends the use of activation keys. To do this, the overcloud nodes need to be registered using environment files. See Registering the Overcloud with an Environment File section of the Red Hat OpenStack Platform 10 Advanced Overcloud Customization document, for details.

After these are setup, you can update overcloud nodes with the latest set of RPMs initiated from the OpenStack director node, by following the procedures documented in the Updating the Overcloud Packages section of the Red Hat OpenStack Platform 10 Upgrading Red Hat OpenStack Platform document.

Running the Password Encryption Script

Ensure that the system has the cisco-vts-overcloud-installer package installed. See **Step 1**, Installing and Configuring Cisco VTS Packages , on page 2.

```
sudo yum install cisco-vts-overcloud-installer
```

Run the following command:

```
$ encpwd <clearTextPassword>
```

Note Any special characters in the password need to be preceded with \. For example, Cisco123! should be entered as Cisco123\!. For security reasons, we recommend that you clear the history from the command line to avoid the clear text password from getting displayed at a later point in time, by running the following command:

```
history -cw
```