Red Hat OpenStack Platform 16.2

Migrating the Networking Service to the ML2/OVN Mechanism Driver

Migrate the Networking service (neutron) from the ML2/OVS mechanism driver to the ML2/OVN mechanism driver

Red Hat OpenStack Platform 16.2 Migrating the Networking Service to the ML2/OVN Mechanism Driver

Migrate the Networking service (neutron) from the ML2/OVS mechanism driver to the ML2/OVN mechanism driver

OpenStack Team
rhos-docs@redhat.com

Legal Notice

Abstract

This is an instructional guide for migrating the Red Hat OpenStack Platform Networking service (neutron) from the Modular Layer 2 plug-in with Open vSwitch mechanism driver to Modular Layer 2 plug-in with Open Virtual Networking.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright's message .

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your input on our documentation. Tell us how we can make it better.

Providing documentation feedback in Jira

Use the Create Issue form to provide feedback on the documentation. The Jira issue will be created in the Red Hat OpenStack Platform Jira project, where you can track the progress of your feedback.

1. Ensure that you are logged in to Jira. If you do not have a Jira account, create an account to submit feedback.

2. Click the following link to open a the **Create Issue** page: Create Issue

3. Complete the **Summary** and **Description** fields. In the **Description** field, include the documentation URL, chapter or section number, and a detailed description of the issue. Do not modify any other fields in the form.

4. Click **Create**.

CHAPTER 1. PLANNING YOUR MIGRATION OF THE ML2 MECHANISM DRIVER FROM OVS TO OVN

NOTE

Red Hat strongly recommends that you upgrade from RHOSP 16.2 to RHOSP 17.1 before you migrate from the OVS mechanism driver to the OVN mechanism driver.

RHOSP 17.1 affords a better baseline for migration in most cases. For example, in RHOSP 16.2, you cannot migrate from the OVS mechanism driver with trunk ports or iptables hybrid firewalls to the OVN mechanism driver. In RHOSP 17.1, you can.

In addition, some improvements in the RHOSP 17.1 version of the OVN migration tools that are not available in the 16.2 version of the migration tools. For example, the RHOSP 17.1 migration tools feature the ability to revert back to the OVS mechanism driver in case of a failure during the migration.

Red Hat chose ML2/OVN as the default mechanism driver for all new deployments starting with RHOSP 15.0 because it offers immediate advantages over the ML2/OVS mechanism driver for most customers today. Those advantages multiply with each release while we continue to enhance and improve the ML2/OVN feature set.

The ML2/OVS mechanism driver is deprecated in RHOSP 17.0. Over several releases, Red Hat is replacing ML2/OVS with ML2/OVN.

Support is available for the deprecated ML2/OVS mechanism driver through the RHOSP 17 releases. During this time, the ML2/OVS driver remains in maintenance mode, receiving bug fixes and normal support, and most new feature development happens in the ML2/OVN mechanism driver.

In RHOSP 18.0, Red Hat plans to completely remove the ML2/OVS mechanism driver and stop supporting it.

If your existing Red Hat OpenStack Platform (RHOSP) deployment uses the ML2/OVS mechanism driver, start now to evaluate the benefits and feasibility of replacing the ML2/OVS mechanism driver with the ML2/OVN mechanism driver. Migration is supported in RHOSP 16.2 and will be supported in RHOSP 17.1. Migration tools are available in RHOSP 17.0 for test purposes only.

NOTE

Red Hat requires that you file a proactive support case before attempting a migration from ML2/OVS to ML2/OVN. Red Hat does not support migrations without the proactive support case. See How to submit a Proactive Case .

Engage your Red Hat Technical Account Manager or Red Hat Global Professional Services early in this evaluation. In addition to helping you file the required proactive support case if you decide to migrate, Red Hat can help you plan and prepare, starting with the following basic questions.

When should you migrate?

Timing depends on many factors, including your business needs and the status of our continuing improvements to the ML2/OVN offering. See Feature support in OVN and OVS mechanism drivers and ML2/OVS to ML2/OVN in-place migration: validated and prohibited scenarios .

In-place migration or parallel migration?

Depending on a variety of factors, you can choose between the following basic approaches to migration.

- Parallel migration. Create a new, parallel deployment that uses ML2/OVN and then move your operations to that deployment.

- In-place migration. Use the *ovn_migration.sh* script as described in this document. Note that Red Hat supports the **ovn_migration.sh** script only in deployments that are managed by RHOSP director.

> ⚠️ **WARNING**
>
> An ML2/OVS to ML2/OVN migration alters the environment in ways that might not be completely reversible. A failed or interrupted migration can leave the OpenStack environment inoperable. Before migrating in a production environment, file a proactive support case. Then work with your Red Hat Technical Account Manager or Red Hat Global Professional Services to create a backup and migration plan and test the migration in a stage environment that closely resembles your production environment.

1.1. FEATURE SUPPORT IN OVN AND OVS MECHANISM DRIVERS

Review the availability of Red Hat OpenStack Platform (RHOSP) features as part of your OVS to OVN mechanism driver migration plan.

Feature	OVN RHOSP 16.2	OVN RHOSP 17.1	OVS RHOSP 16.2	OVS RHOSP 17.1	Additional information
Provisioning Baremetal Machines with OVN DHCP	No	No	Yes	Yes	The built-in DHCP server on OVN presently can not provision baremetal nodes. It cannot serve DHCP for the provisioning networks. Chainbooting iPXE requires tagging (**--dhcp-match** in dnsmasq), which is not supported in the OVN DHCP server. See https://bugzilla.redhat.com/show_bug.cgi?id=1622154.
North/south routing on VF(direct) ports on VLAN project (tenant networks)	No	No	Yes	Yes	Core OVN limitation. See https://bugs.launchpad.net/neutron/+bug/1875852.
Reverse DNS for internal DNS records	No	Yes	Yes	Yes	See https://bugzilla.redhat.com/show_bug.cgi?id=2211426.

Feature	OVN RHOSP 16.2	OVN RHOSP 17.1	OVS RHOSP 16.2	OVS RHOSP 17.1	Additional information
Internal DNS resolution for isolated networks	No	No	Yes	Yes	OVN does not support internal DNS resolution for isolated networks because it does not allocate ports for DNS service. This does not affect OVS deployments because OVS uses dnsmasq. See https://issues.redhat.com/browse/OSP-25661.
Security group logging	Tech Preview	Yes	No	No	RHOSP does not support security group logging with the OVS mechanism driver.
Stateless security groups	No	Yes	No	No	See Configuring security groups.
Load-balancing service distributed virtual routing (DVR)	Yes	Yes	No	No	The OVS mechanism driver routes Load-balancing service traffic through Controller or Network nodes even with DVR enabled. The OVN mechanism driver routes Load-balancing service traffic directly through the Compute nodes.
IPv6 DVR	Yes	Yes	No	No	With the OVS mechanism driver, RHOSP does not distribute IPv6 traffic to the Compute nodes, even when DVR is enabled. All ingress/egress traffic goes through the centralized Controller or Network nodes. If you need IPv6 DVR, use the OVN mechanism driver.
DVR and layer 3 high availability (L3 HA)	Yes	Yes	No	No	RHOSP deployments with the OVS mechanism driver do not support DVR in conjunction with L3 HA. If you use DVR with RHOSP director, L3 HA is disabled. This means that the Networking service still schedules routers on the Network nodes and load-shares them between the L3 agents. However, if one agent fails, all routers hosted by this agent also fail. This affects only SNAT traffic. Red Hat recommends using the **allow_automatic_l3agent_failover** feature in such cases, so that if one Network node fails, the routers are rescheduled to a different node.

1.2. ML2/OVS TO ML2/OVN IN-PLACE MIGRATION: VALIDATED AND PROHIBITED SCENARIOS

Red Hat continues to test and refine in-place migration scenarios. Work with your Red Hat Technical Account Manager or Global Professional Services to determine whether your OVS deployment meets the criteria for a valid in-place migration scenario.

1.2.1. Validated ML2/OVS to ML2/OVN migration scenarios

DVR to DVR

Start: RHOSP 16.1.1 or later with OVS with DVR.
End: Same RHOSP version and release with OVN with DVR.

SR-IOV was not present in the starting environment or added during or after the migration.

Centralized routing + SR-IOV with virtual function (VF) ports only

Start: RHOSP 16.1.1 or later with OVS (no DVR)and SR-IOV.
End: Same RHOSP version and release with OVN (no DVR) and SR-IOV.

Workloads used only SR-IOV virtual function (VF) ports. SR-IOV physical function (PF) ports caused migration failure.

1.2.2. ML2/OVS to ML2/OVN in-place migration scenarios that have not been verified

You cannot perform an in-place ML2/OVS to ML2/OVN migration in the following scenarios until Red Hat announces that the underlying issues are resolved.

OVS deployment uses network functions virtualization (NFV)

Red Hat supports new deployments with ML2/OVN and NFV, but has not successfully tested migration of an ML2/OVS and NFV deployment to ML2/OVN. To track progress on this issue, see https://bugzilla.redhat.com/show_bug.cgi?id=1925290.

SR-IOV with physical function (PF) ports

Migration tests failed when any workload uses an SR-IOV PF port. To track progress on this issue, see https://bugzilla.redhat.com/show_bug.cgi?id=1879546.

OVS uses trunk ports

If your ML2/OVS deployment uses trunk ports, do not perform an ML2/OVS to ML2/OVN migration. The migration does not properly set up the trunked ports in the OVN environment. To track progress on this issue, see https://bugzilla.redhat.com/show_bug.cgi?id=1857652.

DVR with VLAN project (tenant) networks

Do not migrate to ML2/OVN with DVR and VLAN project networks. You can migrate to ML2/OVN with centralized routing. To track progress on this issue, see https://bugzilla.redhat.com/show_bug.cgi?id=1766930.

1.2.3. ML2/OVS to ML2/OVN in-place migration and security group rules

Ensure that any custom security group rules in your originating ML2/OVS deployment are compatible with the target ML2/OVN deployment.

For example, the default security group includes rules that allow egress to the DHCP server. If you deleted those rules in your ML2/OVS deployment, ML2/OVS automatically adds implicit rules that allow egress to the DHCP server. Those implicit rules are not supported by ML2/OVN, so in your target ML2/OVN environment, DHCP and metadata traffic would not reach the DHCP server and the instance would not boot. In this case, to restore DHCP access, you could add the following rules:

```
# Allow VM to contact dhcp server (ipv4)
   openstack security group rule create --egress --ethertype IPv4 --protocol udp --dst-port 67
${SEC_GROUP_ID}
   # Allow VM to contact metadata server (ipv4)
   openstack security group rule create --egress --ethertype IPv4 --protocol tcp --remote-ip
169.254.169.254 ${SEC_GROUP_ID}


   # Allow VM to contact dhcp server (ipv6, non-slaac). Be aware that the remote-ip may vary
depending on your use case!
   openstack security group rule create --egress --ethertype IPv6 --protocol udp --dst-port 547 --
remote-ip ff02::1:2 ${SEC_GROUP_ID}
   # Allow VM to contact metadata server (ipv6)
   openstack security group rule create --egress --ethertype IPv6 --protocol tcp --remote-ip
fe80::a9fe:a9fe ${SEC_GROUP_ID}
```

CHAPTER 2. MIGRATING THE ML2 MECHANISM DRIVER FROM OVS TO OVN

2.1. PREPARING THE ENVIRONMENT FOR MIGRATION OF THE ML2 MECHANISM DRIVER FROM OVS TO OVN

Environment assessment and preparation is critical to a successful migration. Your Red Hat Technical Account Manager or Global Professional Services will guide you through these steps.

Prerequisites

- Your deployment is the latest RHOSP 16.2 version. In other words, if you need to upgrade or update your OpenStack version, perform the upgrade or update first, and then perform the ML2/OVS to ML2/OVN migration.

- At least one IP address is available for each subnet pool.
  The OVN mechanism driver creates a metadata port for each subnet. Each metadata port claims an IP address from the IP address pool.

- You have worked with your Red Hat Technical Account Manager or Global Professional Services to plan the migration and have filed a proactive support case. See How to submit a Proactive Case.

Procedure

1. Create an ML2/OVN stage deployment to obtain the baseline configuration of your target ML2/OVN deployment and test the feasibility of the target deployment.
   Design the stage deployment with the same basic roles, routing, and topology as the planned post-migration production deployment. Save the **overcloud-deploy.sh** file and any files referenced by the deployment, such as environment files. You need these files later in this procedure to configure the migration target environment.

   > **NOTE**
   >
   > Use these files only for creation of the stage deployment and in the migration. Do not re-use them after the migration.

2. If your ML2/OVS deployment uses VXLAN or GRE project networks, schedule for a waiting period of up to 24 hours after the setup-mtu-t1 step.

   - This waiting period allows the VM instances to renew their DHCP leases and receive the new MTU value. During this time you might need to manually set MTUs on some instances and reboot some instances.

   - 24 hours is the time based on default configuration of 86400 seconds. The actual time depends on /var/lib/config-data/puppet-generated/neutron/etc/neutron/dhcp_agent.ini dhcp_renewal_time and /var/lib/config-data/puppet-generated/neutron/etc/neutron/neutron.conf dhcp_lease_duration parameters.

3. Install python3-networking-ovn-migration-tool.

   ```
   sudo dnf install python3-networking-ovn-migration-tool @container-tools
   ```

The **@container-tools** argument also installs the container tools if they are not already present.

4. Create a directory on the undercloud, and copy the Ansible playbooks:

```
mkdir ~/ovn_migration
cd ~/ovn_migration
cp -rfp /usr/share/ansible/networking-ovn-migration/playbooks .
```

5. Copy your ML2/OVN stage deployment files to the migration home directory, such as ~/**ovn_migration**.
The stage migration deployment files include **overcloud-deploy.sh** and any files referenced by the deployment, such as environment files. Rename the copy of **overcloud-deploy.sh** to **overcloud-deploy-ovn.sh**. Use this script for migration only. Do not use it for other purposes.

6. Find your migration scenario in the following list and perform the appropriate steps to customize the **openstack deploy** command in **overcloud-deploy-ovn.sh**.

   Scenario 1: DVR to DVR, compute nodes have connectivity to the external network

   - Add the following environment files to the **openstack deploy** command in overcloud-deploy-ovn.sh. Add them in the order shown. This command example uses the default **neutron-ovn-dvr-ha.yaml** file. If you use a different file, replace the file name in the command.

     ```
     -e /usr/share/openstack-tripleo-heat-templates/environments/services/neutron-ovn-dvr-ha.yaml \
     -e $HOME/ovn-extras.yaml
     ```

   Scenario 2: Centralized routing to centralized routing (no DVR)

   - If your deployment uses SR-IOV, add the service definition **OS::TripleO::Services::OVNMetadataAgent** to the Controller role in the file **roles_data.yaml**.

   - Preserve the pre-migration custom bridge mappings.

     - Run this command on a networker or combined networker/controller node to get the current bridge mappings:

       ```
       sudo podman exec -it neutron_ovs_agent crudini --get /etc/neutron/plugins/ml2/openvswitch_agent.ini ovs bridge_mappings
       ```

       Example output

       ```
       datacentre:br-ex,tenant:br-isolated
       ```

     - On the undercloud, create an environment file for the bridge mappings: **/home/stack/neutron_bridge_mappings.yaml**.

     - Set the defaults in the environment file. For example:

       ```
       parameter_defaults:
         ComputeParameters:
           NeutronBridgeMappings: "datacentre:br-ex,tenant:br-isolated"
       ```

- Add the following environment files to the **openstack deploy** command in overcloud-deploy-ovn.sh. Add them in the order shown. If your environment does not use SR-IOV, omit the neutron-ovn-sriov.yaml file. The file ovn-extras.yaml does not exist yet but it is created by the script ovn_migration.sh before the **openstack deploy** command is run.

  ```
  -e /usr/share/openstack-tripleo-heat-templates/environments/services/neutron-ovn-
  ha.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/services/neutron-ovn-
  sriov.yaml \
  -e /home/stack/ovn-extras.yaml  \
  -e /home/stack/neutron_bridge_mappings.yaml
  ```

- Leave any custom network modifications the same as they were before migration.

Scenario 3: Centralized routing to DVR, with Geneve type driver, and compute nodes connected to external networks through br-ex

> **⚠ WARNING**
>
> If your ML2/OVS deployment uses centralized routing and VLAN project (tenant) networks, do not migrate to ML2/OVN with DVR. You can migrate to ML2/OVN with centralized routing. To track progress on this limitation, see https://bugzilla.redhat.com/show_bug.cgi?id=1766930.

- Ensure that compute nodes are connected to the external network through the **br-ex** bridge. For example, in an environment file such as compute-dvr.yaml, set the following:

  ```
  type: ovs_bridge
      # Defaults to br-ex, anything else requires specific # bridge mapping entries for it to
  be used.
      name: bridge_name
      use_dhcp: false
      members:
      -
       type: interface
       name: nic3
       # force the MAC address of the bridge to this interface
       primary: true
  ```

7. Ensure that all users have execution privileges on the file **overcloud-deploy-ovn.sh**. The script requires execution privileges during the migration process.

   ```
   $ chmod a+x ~/overcloud-deploy-ovn.sh
   ```

8. Use **export** commands to set the following migration-related environment variables. For example:

```
$ export PUBLIC_NETWORK_NAME=my-public-network
```

- STACKRC_FILE - the stackrc file in your undercloud.
  Default: ~/stackrc

- OVERCLOUDRC_FILE - the overcloudrc file in your undercloud.
  Default: ~/overcloudrc

- OVERCLOUD_OVN_DEPLOY_SCRIPT - the deployment script.
  Default: ~/overcloud-deploy-ovn.sh

- PUBLIC_NETWORK_NAME - the name of your public network.
  Default: *public*.

- IMAGE_NAME - the name or ID of the glance image to use to boot a test server.
  Default: *cirros*.

  The image is automatically downloaded during the pre-validation / post-validation process.

- VALIDATE_MIGRATION - Create migration resources to validate the migration. Before starting the migration, the migration script boots a server and validates that the server is reachable after the migration.
  Default: True.

> **WARNING**
>
> Migration validation requires at least two available floating IP addresses, two networks, two subnets, two instances, and two routers as admin.
>
> Also, the network specified by PUBLIC_NETWORK_NAME must have available floating IP addresses, and you must be able to ping them from the undercloud.
>
> If your environment does not meet these requirements, set VALIDATE_MIGRATION to False.

- SERVER_USER_NAME - User name to use for logging to the migration instances.
  Default: *cirros*.

- DHCP_RENEWAL_TIME - DHCP renewal time in seconds to configure in DHCP agent configuration file.
  Default: 30

9. Ensure you are in the ovn-migration directory and run the command **ovn_migration.sh generate-inventory** to generate the inventory file **hosts_for_migration** and the **ansible.cfg** file.

```
$ ovn_migration.sh generate-inventory   | sudo tee -a /var/log/ovn_migration_output.txt
```

10. Review the **hosts_for_migration** file for accuracy.

    a. Ensure the lists match your environment.

    b. Ensure there are ovn controllers on each node.

    c. Ensure there are no list headings (such as [ovn-controllers]) that do not have list items under them.

    d. From the ovn migration directory, run the command *ansible -i hosts_for_migration -m ping all*

11. If your original deployment uses VXLAN or GRE, you need to adjust maximum transmission unit (MTU) values. Proceed to Adjusting MTU for migration from the OVS mechanism driver to the OVN mechanism driver.
    If your original deployment uses VLAN networks, you can skip the MTU adjustments and proceed to Preparing container images for migration from the OVS mechanism driver to the OVN mechanism driver.

2.2. ADJUSTING MTU FOR MIGRATION OF THE ML2 MECHANISM DRIVER FROM OVS TO OVN

If you are migrating from RHOSP 16.2 with the OVN mechanism driver with VXLAN or GRE to the OVN mechanism driver with Geneve, you must ensure that the maximum transmission unit (MTU) settings are smaller than or equal to the smallest MTU in the network.

If your current deployment uses VLAN instead of VXLAN or GRE, skip this procedure and proceed to Preparing container images for migration from the OVS mechanism driver to the OVN mechanism driver.

Prerequisites

- You have completed the steps in Preparing the environment for migration from the OVS mechanism driver to the OVN mechanism driver.

- Your pre-migration deployment is the latest RHOSP 16.2 version, with VXLAN.

Procedure

1. Run **ovn_migration.sh setup-mtu-t1**. This lowers the T1 parameter of the internal neutron DHCP servers that configure the **dhcp_renewal_time** in /var/lib/config-data/puppet-generated/neutron/etc/neutron/dhcp_agent.ini in all the nodes where DHCP agent is running.

   ```
   $ ovn_migration.sh setup-mtu-t1   | sudo tee -a /var/log/ovn_migration_output.txt
   ```

2. If your original OVS deployment uses VXLAN or GRE project networking, wait until the DHCP leases have been renewed on all VM instances. This can take up to 24 hours depending on lease renewal settings and the number of instances.

3. Verify that the T1 parameter has propagated to existing VMs.

   - Connect to one of the compute nodes.

   - Run **tcpdump`** over one of the VM taps attached to a project network.
     If T1 propagation is successful, expect to see requests occur approximately every 30 seconds:

```
[heat-admin@overcloud-novacompute-0 ~]$ sudo tcpdump -i tap52e872c2-e6 port 67 or
port 68 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on tap52e872c2-e6, link-type EN10MB (Ethernet), capture size 262144 bytes
13:17:28.954675 IP 192.168.99.5.bootpc > 192.168.99.3.bootps: BOOTP/DHCP,
Request from fa:16:3e:6b:41:3d, length 300
13:17:28.961321 IP 192.168.99.3.bootps > 192.168.99.5.bootpc: BOOTP/DHCP, Reply,
length 355
13:17:56.241156 IP 192.168.99.5.bootpc > 192.168.99.3.bootps: BOOTP/DHCP,
Request from fa:16:3e:6b:41:3d, length 30013:17:56.249899 IP 192.168.99.3.bootps >
192.168.99.5.bootpc: BOOTP/DHCP, Reply, length 355
```

> **NOTE**
>
> This verification is not possible with cirros VMs. The cirros **udhcpc`** implementation does not respond to DHCP option 58 (T1). Try this verification on a port that belongs to a full Linux VM. Red Hat recommends that you check all the different operating systems represented in your workloads, such as variants of Windows and Linux distributions.

4. If any VM instances were not updated to reflect the change to the T1 parameter of DHCP, reboot them.

5. Lower the MTU of the pre-migration VXLAN and GRE networks:

   ```
   $ ovn_migration.sh reduce-mtu   | sudo tee -a /var/log/ovn_migration_output.txt
   ```

   This step reduces the MTU network by network and tags the completed network with adapted_mtu. The tool acts only on VXLAN and GRE networks. This step will not change any values if your deployment has only VLAN project networks.

6. If you have any instances with static IP assignment on VXLAN or GRE project networks, manually modify the configuration of those instances to configure the new Geneve MTU, which is the current VXLAN MTU minus 8 bytes. For example, if the VXLAN-based MTU was 1450, change it to 1442.

> **NOTE**
>
> Perform this step only if you have manually provided static IP assignments and MTU settings on VXLAN or GRE project networks. By default, DHCP provides the IP assignment and MTU settings.

7. Proceed to Preparing container images for migration from the OVS mechanism driver to the OVN mechanism driver.

2.3. PREPARING CONTAINER IMAGES FOR MIGRATION OF THE ML2 MECHANISM DRIVER FROM OVS TO OVN

Environment assessment and preparation is critical to a successful migration. Your Red Hat Technical Account Manager or Global Professional Services will guide you through these steps.

Prerequisites

- You have completed the steps in Preparing the environment for migration of the ML2 mechanism driver from OVS to OVN

- If your original deployment uses VXLAN or GRE, you also completed the steps in Adjusting MTU for migration from the OVS mechanism driver to the OVN mechanism driver.

Procedure

1. Prepare the new container images for use after the migration to ML2/OVN.

   a. Create **containers-prepare-parameter.yaml** file in the home directory if it is not present.

   ```
   $ test -f $HOME/containers-prepare-parameter.yaml || sudo openstack tripleo container image prepare default \
   --output-env-file $HOME/containers-prepare-parameter.yaml
   ```

   b. Verify that **containers-prepare-parameter.yaml** is present at the end of your $HOME/overcloud-deploy-ovn.sh and $HOME/overcloud-deploy.sh files.

   c. Change the neutron_driver in the **containers-prepare-parameter.yaml** file to ovn:

   ```
   $ sed -i -E 's/neutron_driver:([ ]\w+)/neutron_driver: ovn/' $HOME/containers-prepare-parameter.yaml
   ```

   d. Verify the changes to the neutron_driver:

   ```
   $ grep neutron_driver $HOME/containers-prepare-parameter.yaml
   neutron_driver: ovn
   ```

   e. Update the images:

   ```
   $ sudo openstack tripleo container image prepare \
   --environment-file /home/stack/containers-prepare-parameter.yaml
   ```

   > **NOTE**
   >
   > Provide the full path to your **containers-prepare-parameter.yaml** file. Otherwise, the command completes very quickly without updating the image list or providing an error message.

2. On the undercloud, validate the updated images.

   ```
   . Log in to the undercloud as the user `stack` and source the stackrc file.
   $ source ~/stackrc
   $ openstack tripleo container image list | grep  '\-ovn'
   ```

   Your list should resemble the following example. It includes containers for the OVN databases, OVN controller, the metadata agent, and the neutron server agent.

   ```
   docker://undercloud-0.ctlplane.redhat.local:8787/rh-osbs/rhosp16-openstack-ovn-northd:16.2_20211110.2
   docker://undercloud-0.ctlplane.redhat.local:8787/rh-osbs/rhosp16-openstack-ovn-sb-db-server:16.2_20211110.2
   ```

> docker://undercloud-0.ctlplane.redhat.local:8787/rh-osbs/rhosp16-openstack-ovn-controller:16.2_20211110.2
> docker://undercloud-0.ctlplane.redhat.local:8787/rh-osbs/rhosp16-openstack-neutron-server-ovn:16.2_20211110.2
> docker://undercloud-0.ctlplane.redhat.local:8787/rh-osbs/rhosp16-openstack-ovn-nb-db-server:16.2_20211110.2
> docker://undercloud-0.ctlplane.redhat.local:8787/rh-osbs/rhosp16-openstack-neutron-metadata-agent-ovn:16.2_20211110.2

3. Proceed to Migrating from ML2/OVS to ML2/OVN .

2.4. MIGRATING THE ML2 MECHANISM DRIVER FROM OVS TO OVN

The *ovn-migration* script performs environmental setup, migration, and cleanup tasks related to the in-place migration of the ML2 mechanism driver from OVS to OVN.

Prerequisites

- You have completed the steps in Preparing to migrate from ML2/OVS to ML2/OVN

- If your original deployment uses VXLAN or GRE, you also completed the steps in Adjusting MTU for migration from the OVS mechanism driver to the OVN mechanism driver.

- You also completed all required migration steps through Preparing container images for migration from the OVS mechanism driver to the OVN mechanism driver.

Procedure

1. Stop all operations that interact with the Networking Service (neutron) API, such as creating new networks, subnets, or routers, or migrating virtual machine instances between compute nodes.
   Interaction with Networking API during migration can cause undefined behavior. You can restart the API operations after completing the migration.

2. Run **ovn_migration.sh start-migration** to begin the migration process. The **tee** command creates a copy of the script output for troubleshooting purposes.

   > $ ovn_migration.sh start-migration | sudo tee -a /var/log/ovn_migration_output.txt

Result

The script performs the following actions.

- Creates pre-migration resources (network and VM) to validate existing deployment and final migration.

- Updates the overcloud stack to deploy OVN alongside reference implementation services using the temporary bridge br-migration instead of br-int. The temporary bridge helps to limit downtime during migration.

- Generates the OVN northbound database by running neutron-ovn-db-sync-util. The utility examines the Neutron database to create equivalent resources in the OVN northbound database.

- Clones the existing resources from br-int to br-migration, to allow ovn to find the same resource UUIDS over br-migration.

- Re-assigns ovn-controller to br-int instead of br-migration.

- Removes node resources that are not used by ML2/OVN, including the following.

  - Cleans up network namespaces (fip, snat, qrouter, qdhcp).

  - Removes any unnecessary patch ports on **br-int**.

  - Removes **br-tun** and **br-migration** ovs bridges.

  - Deletes ports from **br-int** that begin with **qr-**, **ha-**, and **qg-** (using neutron-netns-cleanup).

- Deletes Networking Service (neutron) agents and Networking Service HA internal networks from the database through the Networking Service API.

- Validates connectivity on pre-migration resources.

- Deletes pre-migration resources.

- Creates post-migration resources.

- Validates connectivity on post-migration resources.

- Cleans up post-migration resources.

- Re-runs the deployment tool to update OVN on **br-int**.