



Red Hat OpenStack Platform 17.1

Migrating to the OVN mechanism driver

Migrate the Red Hat OpenStack Platform Networking service (neutron) from the ML2/OVS mechanism driver to the ML2/OVN mechanism driver

Red Hat OpenStack Platform 17.1 Migrating to the OVN mechanism driver

Migrate the Red Hat OpenStack Platform Networking service (neutron) from the ML2/OVS mechanism driver to the ML2/OVN mechanism driver

OpenStack Team
rhos-docs@redhat.com

Legal Notice

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

Instructions for migrating the Red Hat OpenStack Platform Networking service (neutron) from the Modular Layer 2 plug-in with Open vSwitch mechanism driver (ML2/OVS) to Modular Layer 2 plug-in with Open Virtual Networking (ML2/OVN).

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE	3
PROVIDING FEEDBACK ON RED HAT DOCUMENTATION	4
CHAPTER 1. PLANNING YOUR MIGRATION OF THE ML2 MECHANISM DRIVER FROM OVS TO OVN	5
1.1. FEATURE SUPPORT IN OVN AND OVS MECHANISM DRIVERS	6
1.2. ML2/OVS TO ML2/OVN IN-PLACE MIGRATION: VALIDATED AND PROHIBITED SCENARIOS	7
1.2.1. Validated ML2/OVS to ML2/OVN migration scenarios	7
1.2.2. ML2/OVS to ML2/OVN in-place migration scenarios that have not been validated	8
1.2.3. ML2/OVS to ML2/OVN in-place migration and security group rules	8
CHAPTER 2. MIGRATING THE ML2 MECHANISM DRIVER FROM OVS TO OVN	9
2.1. PREPARING THE ENVIRONMENT FOR MIGRATION TO THE OVN MECHANISM DRIVER	9
2.2. PREPARING CONTAINER IMAGES FOR MIGRATION OF THE ML2 MECHANISM DRIVER FROM OVS TO OVN	13
2.3. LOWERING MTU FOR MIGRATION FROM A VXLAN OVS DEPLOYMENT	14
2.4. MIGRATING THE ML2 MECHANISM DRIVER FROM OVS TO OVN	16
CHAPTER 3. REVERTING A MIGRATION FROM ML2/OVS TO ML2/OVN	18
3.1. RESTORING THE CONTROLLER NODES TO REVERT AN ML2/OVN MIGRATION	18

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

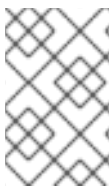
Red Hat chose ML2/OVN as the default mechanism driver for all new deployments starting with RHOSP 15.0 because it offers immediate advantages over the ML2/OVS mechanism driver for most customers today. Those advantages multiply with each release while we continue to enhance and improve the ML2/OVN feature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

The ML2/OVS mechanism driver was 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. 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 RHOSP 17.1.



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 [link:https://access.redhat.com/solutions/2186261](https://access.redhat.com/solutions/2186261).

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 be reverted if you follow the proper backup steps and revert instructions, but the reverted OVS environment might be altered from the original. 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. If you choose to prepare a backup for a potential migration revert, you should also test a migration revert in a stage 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 RHOS P 16.2	OVN RHOS P 17.1	OVS RHOS P 16.2	OVS RHOS P 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 .
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 .

Feature	OVN RHOS P 16.2	OVN RHOS P 17.1	OVS RHOS P 16.2	OVS RHOS P 17.1	Additional information
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

Red Hat tested the following migration paths.

- Distributed virtual routing (DVR) to DVR
- Centralized routing (no-DVR) to no-DVR
- no-DVR to DVR

Successful tests included the workloads with the following port configurations:

- standard ports
- SR-IOV ports
- trunk ports

Successful tests also included iptables_hybrid and Open vSwitch firewall drivers.



NOTE

In each test, the pre-migration environment was created as a greenfield ML2/OVS deployment.

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

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 pre-migration deployments with GRE networks
- OVS with VXLAN to OVN with VXLAN
- OVS pre-migration networks with VLAN project networks and DVR
- OVS pre-migration deployment to OVN with SR-IOV and DVR
- OVS pre-migration deployment with iptables hybrid firewall and trunk ports.
In the migrated environment, instance networking problems occur if you recreate an instance with trunks after an event such as a hard reboot, start and stop, or node reboot. As a workaround, you can either:
 - Switch from the iptables hybrid firewall to the OVS firewall before migrating.
 - Delay the OVN migration until the issue is resolved. To monitor progress on the issue, see https://bugzilla.redhat.com/show_bug.cgi?id=2218596.

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.

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

2.1. PREPARING THE ENVIRONMENT FOR MIGRATION TO THE OVN MECHANISM DRIVER

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 17.1 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](#).
- If your ML2/OVS deployment uses VXLAN project networks, review the potential adjustments described in [Section 2.3, “Lowering MTU for migration from a VXLAN OVS deployment”](#).

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 full **openstack overcloud deploy** command, along with all deployment arguments, into a file called **overcloud-deploy.sh**. Also save any files referenced by the **openstack overcloud deploy** command, such as environment files. You need these files later in this procedure to configure the migration’s target ML2/OVN 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. Install **openstack-neutron-ovn-migration-tool**.

```
sudo dnf install openstack-neutron-ovn-migration-tool
```

3. Copy the `overcloud-deploy.sh` script that you created in Step 1 and rename the copy to `overcloud-migrate-ovn.sh`. Confirm that all paths for the `openstack overcloud deploy` command inside the `overcloud-migrate-ovn.sh` are still correct. You customize some arguments in the `overcloud-migrate-ovn.sh` script in subsequent steps.
4. Find your migration scenario in the following list and perform the appropriate steps to customize the **openstack deploy** command in **overcloud-migrate-ovn.sh**.

**WARNING**

In the deployment command, pay careful attention to the order of the `-e` arguments that add environment files. The environment file with the generic defaults (such as `neutron-ovn-dvr-ha.yaml`) must precede the `-e` argument that specifies the file with custom network environment settings such as bridge mappings.

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

- In `overcloud-migrate-ovn.sh`, add custom heat template file arguments to the `openstack overcloud deploy` command. Add them after the core template file arguments.

The following command example uses the default `neutron-ovn-dvr-ha.yaml` heat template file. Your deployment might use multiple heat files to define your OVN environment. Add each with a separate `-e` argument.

```
openstack overcloud deploy \
--templates /usr/share/openstack-tripleo-heat-templates \
...
-e /usr/share/openstack-tripleo-heat-templates/environments/services/ \
neutron-ovn-dvr-ha.yaml
```

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

- If your deployment uses SR-IOV and other NFV features, in `overcloud-migrate-ovn.sh`, use `-e` arguments to add the SR-IOV environment parameters to the `openstack overcloud deploy` command. Add the SR-IOV environment files after the core template environment file arguments and other custom environment file arguments. For an example of SR-IOV environment file, see `/usr/share/openstack-tripleo-heat-templates/environments/services/neutron-ovn-sriov.yaml`.
- Leave any custom network modifications the same as they were before migration.

Scenario 3: Centralized routing to DVR, and Compute nodes connected to external networks through `br-ex`

- 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 parameters. Then use `-e` to add the environment file to the `openstack overcloud deploy` command in the script `overcloud-migrate-ovn.sh`:

```
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
```

5. Add the following arguments at the end of the **overcloud deploy** command in **overcloud-migrate-ovn.sh**:

```
-e /usr/share/openstack-tripleo-heat-templates/environments/disable-container-manage-
clean-orphans.yaml \
-e $HOME/ovn-extras.yaml
```

6. If **router** appears as a value for **NeutronServicePlugins** or **NeutronPluginExtensions** in any environment file or template, replace the value **router** with **ovn-router**. For example, in **tripleo-heat-templates/environments/services/neutron-ovn-dvr-ha.yaml**:

```
parameter_defaults:
  NeutronServicePlugins: "ovn-router,trunk,qos,placement"
```

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

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

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

```
$ export OVERCLOUDRC_FILE=~/myovercloudrc
```

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-migrate-ovn.sh

DHCP_RENEWAL_TIME

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

BACKUP_MIGRATION_IP

the IP address of the server where backup is stored.
Default: 192.168.24.1

BACKUP_MIGRATION_CTL_PLANE_CIDRS

A comma-separated string of control plane subnets in CIDR notation for all nodes that will be backed up.

Default: 192.168.24.0/24

You can see a list of all relevant environment variables in the beginning of `/usr/bin/ovn_migration.sh` file.

9. Ensure that 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. (Optional) Back up the deployment to prepare a potential migration revert in the case that something unexpected happens during migration.
 - a. Remove the following lines from the file `setup_rear_extra_vars.yaml` if they are present.

```
USER_INPUT_TIMEOUT: 5
KERNEL_CMDLINE: unattended
ISO_RECOVER_MODE: unattended
```

- b. Back up the control plane. Use the backup mechanism of your choice to back up the controller nodes. The supported choice is the default **`ovn-migration.sh backup`** command, which uses the ReaR backup tool.

```
$ ovn_migration.sh backup
```

- c. Back up the templates and environment files that were used to deploy the overcloud. The **`ovn-migration.sh backup`** command does not back up the overcloud. If you need to revert the controller nodes after a partial or failed migration, then you will need this backup to restore the OVS overcloud.
 - d. Copy the script that you used to deploy the original RHOSP 16.2 ML2/OVS deployment. For example, the original script might be named **`overcloud_deploy.sh`**. Name the copy **`overcloud-revert-ovs.sh`**.
 - e. Create a file **`/home/stack/ovs-extra.yml`** with the following contents:

```
parameter_defaults:
  ForceNeutronDriverUpdate: true
```

- f. Ensure that the final environment file argument in **`overcloud-revert-ovs.sh`** is the following.

■


```
-e /home/stack/ovs-extra.yml
```

- g. Store **overcloud-revert-ovs.sh** securely. You will need it if you revert a failed migration. .
12. Proceed to [Section 2.2, “Preparing container images for migration of the ML2 mechanism driver from OVS to OVN”](#).

2.2. 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](#)

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 `$HOME/overcloud-migrate-ovn.sh` and `$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. If your original deployment uses VXLAN, you might need to adjust maximum transmission unit (MTU) values. Proceed to [Section 2.3, “Lowering MTU for migration from a VXLAN OVS deployment”](#).
If your original deployment uses VLAN networks, you can skip the MTU adjustments and proceed to [Section 2.4, “Migrating the ML2 mechanism driver from OVS to OVN”](#).

2.3. LOWERING MTU FOR MIGRATION FROM A VXLAN OVS DEPLOYMENT

If your pre-migration OVS deployment uses the VXLAN tunneling protocol, you might need to reduce the network maximum transmission unit (MTU) by 8 bytes before migrating to OVN, which uses the Geneve tunneling protocol.



NOTE

Consider performing this procedure in a dedicated maintenance window period before the migration.

VXLAN packets reserve 50 bytes of data for header content. This includes 42 bytes of standard outer headers plus an 8-byte VXLAN header. If the physical network uses the standard ethernet MTU of 1500 bytes, you can set the MTU on your VXLAN networks to 1450 and traffic can pass without fragmentation.

Geneve packets reserve 58 bytes of data for header content. This includes the 42 bytes of standard outer headers plus a 16-byte Geneve header. Thus, if the physical network has an MTU less than 1508, you must reduce the MTU on your pre-migration OpenStack VXLAN networks by 8 bytes to avoid the need for fragmentation.



NOTE

If your physical network can transmit at least 58 bytes more than your OpenStack VXLAN network MTU without fragmentation, skip this procedure and proceed to [Section 2.4, “Migrating the ML2 mechanism driver from OVS to OVN”](#). For example, you can skip this procedure if your physical network is configured for 9000-byte jumbo frames and your openstack network MTU is 8942 or less.

The RHOSP OVN migration tool automatically lowers the MTU by 8 bytes on VXLAN and GRE overcloud networks. In the following procedure, you use the tool to:

- increase the frequency of DHCP renewals by reducing the DHCP T1 timer to 30 seconds.
- reduce the MTU size on existing VXLAN networks by 8 bytes.

If your deployment does not use DHCP to configure all VM instances, you must manually reduce MTU on the excluded instances.

Prerequisites

- You have completed the steps in [Section 2.1, “Preparing the environment for migration to the OVN mechanism driver”](#)
- You have completed the steps in [Section 2.2, “Preparing container images for migration of the ML2 mechanism driver from OVS to OVN”](#).
- Your pre-migration deployment is Red Hat OpenStack Platform (RHOSP) 17.1 or later with VXLAN.

Procedure

1. Run **ovn_migration.sh reduce-dhcp-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 reduce-dhcp-t1 | sudo tee -a /var/log/ovn_migration_output.txt
```

2. Verify that the T1 parameter has propagated to existing VMs. The process might take up to four hours.
 - Log in 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

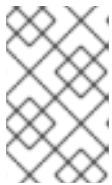
This verification is not possible with cirros VMs. The cirros `udhcpc` implementation does not respond to DHCP option 58 (T1). Try this verification on a port that belongs to a full Linux VM. Red Hat recommends that you check all the different operating systems represented in your workloads, such as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

3. If any VM instances were not updated to reflect the change to the T1 parameter of DHCP, reboot them.
4. Lower the MTU of the pre-migration VXLAN 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 networks. This step will not change any values if your deployment has only VLAN project networks.

5. If you have any instances with static IP assignment on VXLAN project networks, manually reduce the instance MTU by 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 project networks. By default, DHCP provides the IP assignment and MTU settings.

6. Proceed to [Section 2.4, “Migrating the ML2 mechanism driver from OVS to 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 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](#).
- 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, routers, or instances, or migrating 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.

- 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.
- 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.

CHAPTER 3. REVERTING A MIGRATION FROM ML2/OVS TO ML2/OVN

If you used **ovn-migration.sh backup** to back up the controller nodes as described in <xref>, and you also backed up the overcloud templates before starting the migration from ML2/OVS to ML2/OVN, you can revert the migration by performing these basic steps.

1. Restore the controller nodes.
2. Run `revert.yaml` to remove ML2/OVN artifacts.
3. Redeploy the overcloud using the ML2/OVS templates that you backed up.



NOTE

After you revert the migration and restore the compute nodes, you may experience significant network downtime of 20 minutes or more.



WARNING

An ML2/OVS to ML2/OVN migration alters the environment in ways that might not be completely reversible.

You can revert a failed or interrupted migration if you follow the proper backup steps and revert instructions, but the reverted OVS environment might be altered from the original. For example, if you migrate to the OVN mechanism driver, then migrate an instance to another Compute node, and then revert the OVN migration, the instance will be on the original Compute node. Also, a revert operation interrupts connection to the dataplane.

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.

If you choose to prepare a backup for a potential migration revert, you should also test a migration revert in a stage environment.

A migration revert is not a reverse migration. It is intended solely for use as a last resort in the case of a failed migration. If the migration passes validation, and you later identify operational issues in the post-migration environment, address those issues as bugs in the post-migration environment,

3.1. RESTORING THE CONTROLLER NODES TO REVERT AN ML2/OVN MIGRATION

If you used **ovn-migration.sh backup** to back up controller nodes before the migration, you can use the Relax and Recover tool (ReaR) to restore them in the event of a failed post-migration environment.

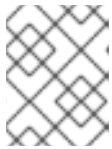
Prerequisites

Prerequisites

- You have filed a support case for the failed migration and have received instructions from Red Hat.
- You used **ovn-migration.sh backup** to back up controller nodes before the migration.
- You have access to the backup node.
- You backed up the overcloud templates needed to re-deploy the ML2/OVS overcloud.

Procedure

1. Stop all operations that interact with the control plane API, such as creating new networks, subnets, or routers, or migrating virtual machine instances between compute nodes.
2. Power off each control plane node. Ensure that the control plane nodes are powered off completely before you proceed.
3. Boot each control plane node with the corresponding backup ISO image.
4. When the **Relax-and-Recover** boot menu displays, on each control plane node, select **Recover <control_plane_node>**. Replace **<control_plane_node>** with the name of the corresponding control plane node.



NOTE

If your system uses UEFI, select the **Relax-and-Recover (no Secure Boot)** option.

5. On each control plane node, log in as the **root** user and restore the node:
The following message displays:

```
Welcome to Relax-and-Recover. Run "rear recover" to restore your system!
RESCUE <control_plane_node>:~ # rear recover
```

When the control plane node restoration process completes, the console displays the following message:

```
Finished recovering your system
Exiting rear recover
Running exit tasks
```

6. Power off the node:

```
RESCUE <control_plane_node>:~ # poweroff
```

7. Set the boot sequence to the normal boot device. On boot up, the node resumes its previous state.
8. To ensure that the services are running correctly, check the status of pacemaker. Log in to a Controller node as the **root** user and enter the following command:

```
# pcs status
```

9. Use Ansible to run the revert.yml file:

```
ansible-playbook -vv /usr/share/ansible/\
neutron-ovn-migration/playbooks/revert.yml \
-i hosts_for_migration
```

10. If **ovn-router** appears as a value for **NeutronServicePlugins** or **NeutronPluginExtensions** in any environment file or template, replace the value **ovn-router** with **router**. The parameter might appear in more than one file. For example, one such file is **tripleo-heat templates/environments/services/neutron-ovn-dvr-ha.yaml**:

```
parameter_defaults:
  NeutronServicePlugins: "router,trunk,qos,placement"
```

11. Update the overcloud to use the OVS templates:

```
bash ~/overcloud-revert-ovs.sh
```

12. If your original ML2/OVS deployment included instances that used trunk ports, run the following command to restore connectivity to those instances:

```
ansible-playbook -vv /usr/share/ansible/neutron-ovn-migration/playbooks/revert-rewire.yml \
-i hosts_for_migration
```

Troubleshooting

- Clear resource alarms that are displayed by **pcs status** by running the following command:

```
# pcs resource clean
```

- Clear STONITH fencing action errors that are displayed by **pcs status** by running the following commands:

```
# pcs resource clean
# pcs stonith history cleanup
```