Abstract

This is an instructional guide for using OVN in OpenStack Networking Tasks.
# Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE .......................................................... 3  

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION ............................... 4  

CHAPTER 1. MIGRATING FROM ML2/OVS TO ML2/OVN ................................ 5  
  1.1. LIMITATIONS OF THE ML2/OVN MECHANISM DRIVER ......................... 6  
    1.1.1. ML2/OVS features not yet supported by ML2/OVN ......................... 6  
    1.1.2. Core OVN limitations ............................................................. 7  
  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 verified .......................................................... 7  
    1.2.3. ML2/OVS to ML2/OVN in-place migration and security group rules .... 8  
  1.3. PREPARING TO MIGRATE FROM ML2/OVS TO ML2/OVN ...................... 9  
  1.4. MIGRATING FROM ML2/OVS TO ML2/OVN ....................................... 15
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.

Using the Direct Documentation Feedback (DDF) function

Use the Add Feedback DDF function for direct comments on specific sentences, paragraphs, or code blocks.

1. View the documentation in the Multi-page HTML format.

2. Ensure that you see the Feedback button in the upper right corner of the document.

3. Highlight the part of text that you want to comment on.

4. Click Add Feedback.

5. Complete the Add Feedback field with your comments.

6. Optional: Add your email address so that the documentation team can contact you for clarification on your issue.

7. Click Submit.
CHAPTER 1. MIGRATING FROM ML2/OVS TO ML2/OVN

Red Hat chose ML2/OVN as the default mechanism driver for all new deployments starting with RHOSP 16.0 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.

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.

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.

Should you migrate?

Red Hat believes that ML2/OVN is the right choice for most deployments. For various reasons, some deployments are better served by ML2/OVS. See Limitations of the ML2/OVN mechanism driver and ML2/OVS to ML2/OVN in-place migration: validated and prohibited scenarios.

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. For instance, security groups logging is planned for a future RHOSP release. If you need that feature, you might plan for a migration after the feature is available. See Limitations of the ML2/OVN mechanism driver.

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.

You can migrate from the ML2/OVS to the ML2/OVN mechanism driver with the ovs-firewall firewall driver. Migration with the iptables_hybrid firewall driver is not supported. The intermediate linux_bridge interface used in iptables_hybrid deployments is not compatible with the migration tool.
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. LIMITATIONS OF THE ML2/OVN MECHANISM DRIVER

Some features available with the ML2/OVS mechanism driver are not yet supported with the ML2/OVN mechanism driver.

1.1.1. ML2/OVS features not yet supported by ML2/OVN

<table>
<thead>
<tr>
<th>Feature</th>
<th>Notes</th>
<th>Track this Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributed virtual routing (DVR) with OVN on VLAN project (tenant) networks.</td>
<td>FIP traffic does not pass to a VLAN tenant network with ML2/OVN and DVR. DVR is enabled by default in new ML2/OVN deployments and in ML2/OVN deployments that were migrated from ML2/OVS deployments that had DVR enabled. If you need VLAN tenant networks with OVN, you can disable DVR. To disable DVR, include the following lines in an environment file:</td>
<td><a href="https://bugzilla.redhat.com/show_bug.cgi?id=1704596">https://bugzilla.redhat.com/show_bug.cgi?id=1704596</a> <a href="https://bugzilla.redhat.com/show_bug.cgi?id=1766930">https://bugzilla.redhat.com/show_bug.cgi?id=1766930</a></td>
</tr>
</tbody>
</table>
| Fragmentation of packets on east/west UDP/ICMP traffic | In east/west traffic OVN does not yet support fragmentation of packets that are larger than the smallest MTU on the east/west path. For example:  
- VM1 is on Network1 with an MTU of 1300.  
- VM2 is on Network2 with an MTU of 1200.  
- A ping in either direction between VM1 and VM2 with a size of 1171 or less succeeds. A ping with a size greater than 1171 results in 100 percent packet loss. | [https://bugzilla.redhat.com/show_bug.cgi?id=1891591](https://bugzilla.redhat.com/show_bug.cgi?id=1891591) |
The built-in DHCP server on OVN presently cannot 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.

---

### 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 VXLAN, target deployment RHOSP 16.2.0**

RHOSP does not yet support ML2/OVN with VXLAN networks. The migration process includes steps to convert VXLAN networks to Geneve. When the migration target version is RHOSP 16.2.0, a bug prevents the expected VXLAN to Geneve conversion, and the networks remain configured as VXLAN. See [bugzilla.redhat.com/show_bug.cgi?id=2003708](https://bugzilla.redhat.com/show_bug.cgi?id=2003708).
This bug affects only migrations to ML2/OVN on RHOSP 16.2. It does not affect migrations to ML2/OVN on RHOSP 16.1.

**OVS deployment uses iptables_hybrid firewall driver**

You can migrate from the ML2/OVS to the ML2/OVN mechanism driver with the openvswitch firewall driver, but not with the iptables_hybrid firewall driver. Migration with the iptables_hybrid firewall driver is not supported. For more information, see https://bugzilla.redhat.com/show_bug.cgi?id=2011450.

**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
```
1.3. PREPARING TO MIGRATE FROM ML2/OVS TO ML2/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 pre-migration deployment is Red Hat OpenStack Platform (RHOSP) 16.1 or later.
- Your pre-migration deployment does not use the `iptables_hybrid` firewall driver. The intermediate `linux_bridge` interface used in `iptables_hybrid` deployments is not compatible with the migration tool.
- Your RHOSP deployment is up to date. 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.
- 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`.

   ```bash
   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:
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 controller node to get the current bridge mappings:

    sudo podman exec -it neutron_api 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 ML2/OVS deployment uses VLAN project networks, skip to step 18.

12. 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.

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

13. 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.

14. If you have any instances with static IP assignment on VXLAN or GRE project networks, you must 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.

15. 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 that requests happen on an interval of approximately 30 seconds:

   ```bash
   [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
   ```
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 types of workloads that your system runs (Windows, different flavors of Linux, etc.).

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

17. 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 ignores non-VXLAN/GRE networks, so if you use VLAN for project networks, this step is not expected to change any values.

18. 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 \n      --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 \n      --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.

19. 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`

### 1.4. MIGRATING FROM ML2/OVS TO ML2/OVN

The `ovn-migration` script performs environmental setup, migration, and cleanup tasks related to the in-place migration from ML2/OVN to ML2/OVS.

**Prerequisites**

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

**Procedure**

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

  ```bash
  $ 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.