



Red Hat OpenStack Platform 17.1

迁移到 OVN 机制驱动程序

将 Red Hat OpenStack Platform Networking 服务(neutron)从 ML2/OVS 机制驱动程序迁移到 ML2/OVN 机制驱动程序

Red Hat OpenStack Platform 17.1 迁移到 OVN 机制驱动程序

将 Red Hat OpenStack Platform Networking 服务(neutron)从 ML2/OVS 机制驱动程序迁移到 ML2/OVN 机制驱动程序

OpenStack Team
rhos-docs@redhat.com

法律通告

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.

摘要

有关使用 Open Virtual Networking (ML2/OVN)的 Modular Layer 2 插件将 Red Hat OpenStack Platform Networking 服务(neutron)从 Modular Layer 2 插件迁移到 Modular Layer 2 插件。

目录

使开源包含更多	3
对红帽文档提供反馈	4
第 1 章 规划从 OVS 迁移到 OVN 的 ML2 机制驱动程序	5
1.1. OVN 和 OVS 机制驱动程序中的功能支持	5
1.2. ML2/OVS 到 ML2/OVN 的原位迁移：经过验证并禁止的场景	7
第 2 章 将 ML2 机制驱动程序从 OVS 迁移到 OVN	9
2.1. 准备迁移到 OVN 机制驱动程序的环境	9
2.2. 准备将 ML2 机制驱动程序从 OVS 迁移到 OVN 的容器镜像	12
2.3. 降低从 VXLAN OVS 部署迁移的 MTU	13
2.4. 将 ML2 机制驱动程序从 OVS 迁移到 OVN	15
第 3 章	17
3.1.	17

使开源包含更多

红帽致力于替换我们的代码、文档和 Web 属性中存在问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。详情请查看 [CTO Chris Wright 的信息](#)。

对红帽文档提供反馈

我们感谢您对文档提供反馈信息。与我们分享您的成功秘诀。

在 JIRA 中提供文档反馈

使用 [Create Issue](#) 表单对文档提供反馈。JIRA 问题将在 Red Hat OpenStack Platform Jira 项目中创建，您可以在其中跟踪您的反馈进度。

1. 确保您已登录到 JIRA。如果您没有 JIRA 帐户，请创建一个帐户来提交反馈。
2. 点击以下链接打开 **Create Issue** 页面：[Create Issue](#)
3. 完成 **Summary** 和 **Description** 字段。在 **Description** 字段中，包含文档 URL、章节或章节号以及问题的详细描述。不要修改表单中的任何其他字段。
4. 点 **Create**。

第1章 规划从 OVS 迁移到 OVN 的 ML2 机制驱动程序

红帽选择 ML2/OVN 作为从 RHOSP 15.0 开始的所有新部署的默认机制驱动程序，因为它现在为大多数客户提供了与 ML2/OVS 机制驱动程序直接的好处。当我们继续增强和改进 ML2/OVN 功能集时，每个版本都会有多大优势。

在 RHOSP 17.0 中弃用了 ML2/OVS 机制驱动程序。在多个版本中，红帽将 ML2/OVS 替换为 ML2/OVN。

通过 RHOSP 17 发行版本，支持已弃用的 ML2/OVS 机制驱动程序。在这段时间中，ML2/OVS 驱动程序处于维护模式，接收程序错误修复和常规支持。ML2/OVN 机制驱动程序中大多数新功能开发都发生。

在 RHOSP 18.0 中，红帽计划完全删除 ML2/OVS 机制驱动程序并停止支持它。

如果您的现有 Red Hat OpenStack Platform (RHOSP)部署使用 ML2/OVS 机制驱动程序，则开始评估将 ML2/OVS 机制驱动程序替换为 ML2/OVN 机制驱动程序的好处和可行性。RHOSP 16.2 和 RHOSP 17.1 支持迁移。



注意

红帽需要在尝试从 ML2/OVS 迁移到 ML2/OVN 前提交主动支持问题单。红帽不支持在没有主动支持问题单的情况下进行迁移。请参阅 [link:https://access.redhat.com/solutions/2186261](https://access.redhat.com/solutions/2186261)。

在此评估早期，请联系您的红帽大客户经理或红帽全球支持服务。除了帮助您提交所需的主动支持问题单（如果您决定迁移）之外，红帽还可以帮助您规划和准备，从以下基本问题开始。

何时应该迁移？

时间取决于许多因素，包括您的业务需要以及对 ML2/OVN 产品持续改进的状态。请参阅 [OVN 和 OVS 机制驱动程序](#) 和 [ML2/OVS 中的功能支持到 ML2/OVN 原位迁移：验证和禁止的情况](#)。

原位迁移或并行迁移？

根据各种因素，您可以选择以下用于迁移的基本方法。

- 并行迁移。创建一个使用 ML2/OVN 的新并行部署，然后将操作移到该部署中。
- 原位迁移。使用 `ovn_migration.sh` 脚本，如本文档所述。请注意，红帽仅在由 RHOSP director 管理的部署中支持 `ovn_migration.sh` 脚本。



警告

ML2/OVS 到 ML2/OVN 迁移以可能无法完全解析的方式改变环境。

1.1. OVN 和 OVS 机制驱动程序中的功能支持

检查 Red Hat OpenStack Platform (RHOSP)功能的可用性，作为 OVS 到 OVN 机制驱动程序迁移计划的一部分。

功能	OVN RHOS P 16.2	OVN RHOS P 17.1	OVS RHOS P 16.2	OVS RHOS P 17.1	附加信息
使用 OVN DHCP 置备裸机	否	否	是	是	OVN 上的内置 DHCP 服务器目前无法置备裸机节点。它无法为 provisioning 网络提供 DHCP。Chainbooting iPXE 需要标记(dnsmasq 中的 --dhcp-match), 这是 OVN DHCP 服务器不支持的。See https://bugzilla.redhat.com/show_bug.cgi?id=1622154 .
VLAN 项目 (租户网络) 端口的 VF (直接) 端口的北/南路由	否	否	是	是	核心 OVN 限制。请参阅 https://bugs.launchpad.net/neutron/+bug/1875852 。
内部 DNS 记录的反向 DNS	否	是	是	是	See https://bugzilla.redhat.com/show_bug.cgi?id=2211426 .
隔离网络的内部 DNS 解析	否	否	是	是	OVN 不支持隔离网络的内部 DNS 解析, 因为它不为 DNS 服务分配端口。这不会影响 OVS 部署, 因为 OVS 使用 dnsmasq。请参阅 https://issues.redhat.com/browse/OSP-25661 。
安全组日志记录	技术预览	是	否	否	RHOSP 不支持使用 OVS 机制驱动程序的安全组日志记录。
无状态安全组	否	是	否	否	请参阅 配置安全组 。
负载均衡服务分布式虚拟路由(DVR)	是	是	否	否	OVS 机制驱动程序通过 Controller 或网络节点路由负载均衡服务流量, 即使启用了 DVR。OVN 机制驱动程序通过 Compute 节点直接路由负载均衡服务流量。
IPv6 DVR	是	是	否	否	使用 OVS 机制驱动程序时, RHOSP 不会将 IPv6 流量分发到 Compute 节点, 即使启用了 DVR。所有入口/出口流量都通过集中式 Controller 或 Network 节点。如果您需要 IPv6 DVR, 请使用 OVN 机制驱动程序。

功能	OVN RHOS P 16.2	OVN RHOS P 17.1	OVS RHOS P 16.2	OVS RHOS P 17.1	附加信息
DVR 和第 3 层高可用性(L3 HA)	是	是	否	否	使用 OVS 机制驱动程序的 RHOSP 部署不支持 DVR 与 L3 HA 结合使用。如果您将 DVR 与 RHOSP director 搭配使用，则禁用 L3 HA。这意味着网络服务仍然在网络节点上调度路由器，并在 L3 代理间共享它们。但是，如果一个代理失败，由此代理托管的所有路由器也会失败。这只会影响 SNAT 流量。红帽建议在这种情况下使用 allow_automatic_l3agent_failover 功能，以便在一个网络节点失败时，路由器会重新调度到不同的节点。

1.2. ML2/OVS 到 ML2/OVN 的原位迁移：经过验证并禁止的场景

红帽继续测试和优化原地迁移方案。与红帽大客户经理或全球专业服务合作，确定您的 OVS 部署是否满足有效迁移方案的条件。

1.2.1. 验证的 ML2/OVS 到 ML2/OVN 迁移场景

红帽测试了以下迁移路径：

- 分布式虚拟路由(DVR)到 DVR
- 集中式路由(no-DVR)到 no-DVR
- No-DVR 到 DVR

成功测试包括以下端口配置的工作负载：

- 标准端口
- SR-IOV 端口
- 中继端口

成功测试还包括 iptables_hybrid 和 Open vSwitch 防火墙驱动程序。



注意

在每个测试中，迁移前环境都是以 greenfield ML2/OVS 部署创建的。

1.2.2. ML2/OVS 到 ML2/OVN 的原位迁移方案尚未验证

在红帽宣布根本问题解决之前，您无法在以下情况下执行从 ML2/OVS 到 ML2/OVN 的原位升级。

- 使用 GRE 网络的 OVS 预迁移部署
- 使用 VXLAN 的 OVS 到使用 VXLAN 的 OVN
- OVS 预迁移网络具有 VLAN 项目网络和 DVR

- 使用 SR-IOV 和 DVR 到 OVN 的 OVS 预迁移部署
- 使用 iptables 混合防火墙和中继端口的 OVS 预迁移部署。
在迁移的环境中，如果您在发生硬重启、启动和停止节点或节点重启后重新创建带有中继的实例，则实例网络问题。作为临时解决方案，您可以：
 - 在迁移前，从 iptables 混合防火墙切换到 OVS 防火墙。
 - 延迟 OVN 迁移，直到问题解决为止。要监控问题的进度，请参阅 https://bugzilla.redhat.com/show_bug.cgi?id=2218596。

1.2.3. ML2/OVS 到 ML2/OVN 的原位迁移和安全组规则

确保源自 ML2/OVS 部署中的任何自定义安全组规则都与目标 ML2/OVN 部署兼容。

第 2 章 将 ML2 机制驱动程序从 OVS 迁移到 OVN

2.1. 准备迁移到 OVN 机制驱动程序的环境

环境评估和准备对于成功迁移至关重要。您的红帽大客户经理或全球专业服务将指导您完成这些步骤。

先决条件

- 您的部署是最新的 RHOSP 17.1 版本。换句话说，如果您需要升级或更新 OpenStack 版本，请首先执行升级或更新，然后执行 ML2/OVS 到 ML2/OVN 迁移。
- 每个子网池至少有一个 IP 地址。
OVN 机制驱动程序为每个子网创建一个元数据端口。每个元数据端口从 IP 地址池声明 IP 地址。
- 您已与红帽大客户经理或全球专业服务合作计划迁移，并提交了一个主动的支持问题单。[请参阅如何提交主动问题单。](#)
- 如果您的 ML2/OVS 部署使用 VXLAN 项目网络，请查看 [第 2.3 节“降低从 VXLAN OVS 部署迁移的 MTU”](#) 中描述的潜在调整。

流程

1. 创建一个 ML2/OVN 阶段部署，以获取目标 ML2/OVN 部署的基准配置，并测试目标部署的可行性。
使用与计划迁移后部署相同的基本角色、路由和拓扑设计阶段部署。将完整的 **openstack overcloud deploy** 命令和所有部署参数保存到名为 **overcloud-deploy.sh** 的文件中。另外，也保存由 **openstack overcloud deploy** 命令引用的任何文件，如环境文件。此流程稍后需要这些文件来配置迁移的目标 ML2/OVN 环境。



注意

仅使用这些文件来创建阶段部署和迁移。迁移后请勿重新使用它们。

2. `sudo dnf install openstack-neutron-ovn-migration-tool`
3. 复制在第 1 步中创建的 `overcloud-deploy.sh` 脚本，并将副本重命名为 `overcloud-migrate-ovn.sh`。确认 `overcloud-migrate-ovn.sh` 中 `overcloud deploy` 命令的所有路径仍然正确。在后续步骤中，您可以自定义 `overcloud-migrate-ovn.sh` 脚本中的一些参数。
4. 在以下列表中找到迁移场景，并执行适当的步骤以在 `overcloud-migrate-ovn.sh` 中自定义 **openstack deploy** 命令。



警告

在部署命令中，注意添加环境文件的 `-e` 参数的顺序。带有通用默认值（如 `neutron-ovn-dvr-ha.yaml`）的环境文件前必须加上 `-e` 参数来指定带有自定义网络环境设置的文件，如网桥映射。

场景 1: DVR 到 DVR, 计算节点与外部网络连接

- 在 `overcloud-migrate-ovn.sh` 中, 添加自定义 heat 模板文件参数到 `openstack overcloud deploy` 命令。在核心模板文件参数后添加它们。
以下命令示例使用默认的 `neutron-ovn-dvr-ha.yaml` heat 模板文件。您的部署可能会使用多个 heat 文件来定义 OVN 环境。使用单独的 `-e` 参数添加每个。

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

场景 2 : 集中路由到集中式路由 (没有 DVR)

- 如果您的部署使用 SR-IOV 和其他 NFV 功能, 在 `overcloud-migrate-ovn.sh` 中, 请使用 `-e` 参数将 SR-IOV 环境参数添加到 `openstack overcloud deploy` 命令中。在核心模板环境文件参数和其他自定义环境文件参数后, 添加 SR-IOV 环境文件文件。有关 SR-IOV 环境文件的示例, 请参阅 `/usr/share/openstack-tripleo-heat-templates/environments/services/neutron-ovn-sriov.yaml`。
- 保留任何自定义网络修改与迁移前相同。

场景 3 : 到一个 DVR 的集中式路由, 以及通过 br-ex 连接到外部网络的 Compute 节点

- 确保 Compute 节点通过 `br-ex` 网桥连接到外部网络。例如, 在 `compute-dvr.yaml` 等环境文件中设置以下参数。然后使用 `-e` 将环境文件添加到脚本 `overcloud-migrate-ovn.sh` 中的 `openstack overcloud deploy` 命令中 :

```
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
```

- 在 `overcloud-migrate-ovn.sh` 中的 `overcloud deploy` 命令末尾添加以下参数 :

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

- 如果路由器显示为任何环境文件或模板中的 `NeutronService PluginExtensions` 或 `NeutronPluginExtensions` 的值, 请将值 `router` 替换为 `ovn-router`。例如, 在 `tripleo-heat-templates/environments/services/neutron-ovn-dvr-ha.yaml` 中 :

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

7. 确保所有用户对 **overcloud-migrate-ovn.sh** 文件具有执行权限。该脚本需要在迁移过程中执行特权。

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

8. 使用 **export** 命令设置以下与迁移相关的环境变量。例如：

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

STACKRC_FILE

undercloud 中的 stackrc 文件。

默认：~/stackrc

OVERCLOUDRC_FILE

undercloud 中的 overcloudrc 文件。

Default: ~/overcloudrc

OVERCLOUD_OVN_DEPLOY_SCRIPT

部署脚本。

Default: ~/overcloud-migrate-ovn.sh

DHCP_RENEWAL_TIME

DHCP 续订时间（以秒为单位）在 DHCP 代理配置文件中配置。

默认：30

BACKUP_MIGRATION_IP

BACKUP_MIGRATION_CTL_PLANE_CIDRS

您可以在 `/usr/bin/ovn_migraton.sh` 文件的开头看到所有相关环境变量的列表。

9. 确保您位于 `ovn-migration` 目录中，并运行 **ovn_migration.sh generate-inventory** 命令来生成清单文件 **hosts_for_migration** 和 **ansible.cfg** 文件。

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

10. 检查 **hosts_for_migration** 文件是否准确。

- a. 确保列表与您的环境匹配。
- b. 确保每个节点上都有 `ovn` 控制器。
- c. 确保没有列表标题（如 `[ovn-controllers]`）没有列表项。
- d. 在 `ovn migration` 目录中，运行命令 `ansible -i hosts_for_migration -m ping all`

11. a.

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

- b.

```
$ ovn_migration.sh backup
```

-
- c.
- d.
- e.

```
parameter_defaults:
  ForceNeutronDriverUpdate: true
```
- f.

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

12. 继续 [第 2.2 节“准备将 ML2 机制驱动程序从 OVS 迁移到 OVN 的容器镜像”](#)。

2.2. 准备将 ML2 机制驱动程序从 OVS 迁移到 OVN 的容器镜像

环境评估和准备对于成功迁移至关重要。您的红帽大客户经理或全球专业服务将指导您完成这些步骤。

前提条件

- 您已完成了准备将 [ML2 机制驱动程序从 OVS 迁移到 OVN](#) 的步骤

流程

1. 准备新的容器镜像，以便在迁移到 ML2/OVN 后使用。

a. 如果不存在，请在主目录中创建 **containers-prepare-parameter.yaml** 文件。

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

b. 验证 `$HOME/overcloud-migrate-ovn.sh` 和 `$HOME/overcloud-deploy.sh` 文件末尾是否存在 **containers-prepare-parameter.yaml**。

c. 将 **containers-prepare-parameter.yaml** 文件中的 `neutron_driver` 更改为 `ovn`：

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

d. 验证 `neutron_driver` 的更改：

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

e. 更新镜像：

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




注意

提供 **containers-prepare-parameter.yaml** 文件的完整路径。否则，命令会在不更新镜像列表或提供错误消息的情况下快速完成。

2. 在 undercloud 上，验证更新的镜像。

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

您的列表应类似以下示例。它包括 OVN 数据库、OVN 控制器、元数据代理和 neutron 服务器代理的容器。

```
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. 如果您的原始部署使用 VXLAN，您可能需要调整最大传输单元(MTU)值。继续 [第 2.3 节“降低从 VXLAN OVS 部署迁移的 MTU”](#)。如果您的原始部署使用 VLAN 网络，您可以跳过 MTU 调整并继续 [第 2.4 节“将 ML2 机制驱动程序从 OVS 迁移到 OVN”](#)。

2.3. 降低从 VXLAN OVS 部署迁移的 MTU

如果您的预迁移 OVS 部署使用 VXLAN 隧道协议，您可能需要在迁移到 OVN（使用 Geneve 隧道协议）前，将网络最大传输单元(MTU)减少 8 字节。

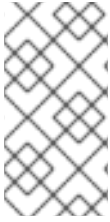


注意

在迁移前，请考虑在专用维护窗口期间执行这个步骤。

VXLAN 数据包为标头内容保留 50 字节的数据。这包括标准外部标头的 42 字节，再加上 8 字节 VXLAN 标头。如果物理网络使用标准以太网 MTU 1500 字节，您可以将 VXLAN 网络上的 MTU 设置为 1450，流量可以在不需要碎片的情况下传递。

Geneve 数据包为标头内容保留 58 字节数据。这包括标准外部标头的 42 字节，再加上 16 字节 Geneve 标头。因此，如果物理网络的 MTU 小于 1508，则必须将预迁移 OpenStack VXLAN 网络上的 MTU 减少 8 字节，以避免需要碎片。



注意

如果您的物理网络可以在没有碎片的情况下传输至少 58 字节超过 OpenStack VXLAN 网络 MTU，请跳过这个过程并继续 [第 2.4 节“将 ML2 机制驱动程序从 OVS 迁移到 OVN”](#)。例如，如果您的物理网络是为 9000 字节巨型帧配置的，并且 openstack 网络 MTU 为 8942 或 less，则可以跳过这个过程。

RHOSP OVN 迁移工具会在 VXLAN 和 GRE overcloud 网络上自动将 MTU 降低 8 字节。在以下步骤中，您可以使用该工具：

- 通过将 DHCP T1 计时器减少到 30 秒，增加 DHCP 续订的频率。
- 将现有 VXLAN 网络的 MTU 大小减少 8 字节。

如果您的部署没有使用 DHCP 来配置所有虚拟机实例，则必须在排除的实例上手动减少 MTU。

先决条件

- 您已完成这些步骤 [第 2.1 节“准备迁移到 OVN 机制驱动程序的环境”](#)
- 您已完成了 [第 2.2 节“准备将 ML2 机制驱动程序从 OVS 迁移到 OVN 的容器镜像”](#) 中的步骤。
- 您的预迁移部署是带有 VXLAN 的 Red Hat OpenStack Platform (RHOSP) 17.1 或更高版本。

流程

1. 运行 `ovn_migration.sh reduce-dhcp-t1`。这会降低在 `/var/lib/config-data/puppet-generated/neutron/etc/neutron/dhcp_agent.ini` 中配置 `dhcp_renewal_time` 的内部 neutron DHCP 服务器的 T1 参数。

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

2. 验证 T1 参数是否已传播到现有虚拟机。此过程可能需要四小时。

- 登录其中一个 Compute 节点。
- 在附加到项目网络的其中一个虚拟机上面运行 `tcpdump`。
如果 T1 propagation 成功，应该查看请求大约每 30 秒发生：

```
[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
```



注意

cirros 虚拟机无法进行这个验证。cirros **udhcpd** 的实现不会响应 DHCP 选项 58 (T1)。在属于完整 Linux 虚拟机的端口中尝试此验证。红帽建议您检查工作负载中代表的所有不同操作系统，如 Windows 和 Linux 发行版的不同操作系统。

3. 如果有任何虚拟机实例没有更新，以反映 DHCP 的 T1 参数的更改，请重新引导它们。
4. 降低预迁移 VXLAN 网络的 MTU：

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

此步骤通过网络减少 MTU 网络，并使用 `adapted_mtu` 标记已完成的网络。该工具仅适用于 VXLAN 网络。如果您的部署只有 VLAN 项目网络，则此步骤不会更改任何值。

5. 如果您在 VXLAN 项目网络上具有静态 IP 分配的任何实例，请手动将实例 MTU 减少 8 字节。例如，如果基于 VXLAN 的 MTU 是 1450，请将其更改为 1442。



注意

只有在 VXLAN 项目网络上手动提供静态 IP 分配和 MTU 设置时，才执行这个步骤。默认情况下，DHCP 提供 IP 分配和 MTU 设置。

6. 继续 [第 2.4 节“将 ML2 机制驱动程序从 OVS 迁移到 OVN”](#)。

2.4. 将 ML2 机制驱动程序从 OVS 迁移到 OVN

`ovn-migration` 脚本执行环境设置、迁移和清理与 ML2 机制驱动程序从 OVS 到 OVN 的原位升级相关的任务。

前提条件

- 您已完成了准备将 [ML2 机制驱动程序从 OVS 迁移到 OVN](#) 的步骤
- 如果您的原始部署使用 VXLAN 或 GRE，您也完成了 [Adjusting MTU](#) 中的步骤，以便从 OVS 机制驱动程序迁移到 OVN 机制驱动程序。
- 您还完成了为从 [OVS 机制驱动程序迁移到 OVN 机制驱动程序](#) 准备容器镜像的所有必要步骤。

流程

1. 停止与网络服务(neutron) API 交互的所有操作，如创建新网络、子网、路由器或实例，或在计算节点之间迁移实例。
在迁移过程中与网络 API 交互可能会导致未定义的行为。您可以在迁移后重启 API 操作。
2. 运行 `ovn_migration.sh start-migration` 以开始迁移过程。`tee` 命令创建脚本输出的副本，用于故障排除目的。

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

结果

脚本执行以下操作：

- 更新 overcloud 堆栈以部署 OVN，以及利用临时网桥 br-migration 而不是 br-int 的引用实施服务。临时网桥有助于限制迁移期间的停机时间。
- 通过运行 neutron-ovn-db-sync-util 来生成 OVN 北向数据库。实用程序检查 Neutron 数据库，以在 OVN 北向数据库中创建等效的资源。
- 将 ovn-controller 重新分配给 br-int，而不是 br-migration。
- 删除 ML2/OVN 不使用的节点资源，包括以下内容：
 - 清理网络命名空间(fip、snat、qrouter、qdhcp)。
 - 删除 **br-int** 中的任何不必要的补丁端口。
 - 删除 **br-tun** 和 **br-migration** ovs 网桥。
 - 从 **br-int** 中删除端口，这些端口以 **qr-**、**ha-** 和 **qg-** 开头（使用 neutron-netns-cleanup）。
- 通过网络服务 API 从数据库中删除网络服务(neutron)代理和网络服务 HA 内部网络。

第 3 章

- 1.
- 2.
- 3.



注意



警告

ML2/OVS 到 ML2/OVN 迁移以可能无法完全解析的方式改变环境。

3.1.

前提条件

-
-
-
-

流程

- 1.
- 2.
- 3.



注意

5. 在每个 control plane 节点上，以 **root** 用户身份登录并恢复节点：

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

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

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

7.

8. 以 **root** 用户身份登录 Controller 节点，然后输入以下命令：

```
# pcs status
```

9.

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

10.

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

11.

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

12.

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

故障排除

-

```
# pcs resource clean
```

-

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