



Red Hat OpenStack Platform 16.2

Networking サービスの ML2/OVN メカニズムド ライバーへの移行

Networking サービス(neutron)を ML2/OVS メカニズムドライバーから ML2/OVN メカニズムドライバーに移行します。

Red Hat OpenStack Platform 16.2 Networking サービスの ML2/OVN メカニズムドライバーへの移行

Networking サービス(neutron)を ML2/OVS メカニズムドライバーから ML2/OVN メカニズムドライバーに移行します。

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

法律上の通知

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Migrating_the_Networking_Service_to_the_ML2OVN_Mechanism_Driver.ent file |.

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.

概要

これは、Open vSwitch メカニズムドライバーを備えた Modular Layer 2 プラグインから、Open Virtual Networking を備えた Modular Layer 2 プラグインに、Red Hat OpenStack Platform Networking サービス (neutron) を移行するためのガイドです。

目次

はじめに	3
多様性を受け入れるオープンソースの強化	4
RED HAT ドキュメントへのフィードバックの提供	5
第1章 ML2/OVS から ML2/OVN への移行	6
1.1. ML2/OVN メカニズムドライバーの制約	7
1.1.1. ML2/OVN ではまだサポートされていない ML2/OVS 機能	7
1.1.2. OVN に関する主な制約	7
1.2. ML2/OVS から ML2/OVN へのインプレースマイグレーション: 検証済みのシナリオおよび禁止されるシナリオ	8
1.2.1. 検証済みの ML2/OVS から ML2/OVN への移行シナリオ	8
1.2.2. 検証されていない ML2/OVS から ML2/OVN へのインプレースマイグレーションのシナリオ	8
1.2.3. ML2/OVS から ML2/OVN へのインプレースマイグレーションおよびセキュリティーグループルール	9
1.3. ML2/OVS から ML2/OVN への移行の準備	9
1.4. ML2/OVS から ML2/OVN への移行	16

はじめに

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、[弊社の CTO、Chris Wright のメッセージ](#) を参照してください。

RED HAT ドキュメントへのフィードバックの提供

弊社ドキュメントに対するご意見をお聞かせください。ドキュメントの改善点があればお知らせください。

ドキュメントへのダイレクトフィードバック (DDF) 機能の使用 (英語版のみ)

特定の文章、段落、またはコードブロックに対して直接コメントを送付するには、DDF の **Add Feedback** 機能を使用してください。なお、この機能は英語版のドキュメントでのみご利用いただけます。

1. **Multi-page HTML** 形式でドキュメントを表示します。
2. ドキュメントの右上隅に **Feedback** ボタンが表示されていることを確認してください。
3. コメントするテキスト部分をハイライト表示します。
4. **Add Feedback** をクリックします。
5. **Add Feedback** フィールドにコメントを入力します。
6. (オプション) ドキュメントチームが連絡を取り問題についてお伺いできるように、ご自分のメールアドレスを追加します。
7. **Submit** をクリックします。

第1章 ML2/OVS から ML2/OVN への移行

RHOSP 16.0 以降のすべての新規デプロイメントについて、Red Hat では ML2/OVN をデフォルトのメカニズムドライバーとして選択しました。これは、今日のほとんどのお客様にとって ML2/OVS メカニズムドライバー以上のメリットが即座に得られるためです。継続して ML2/OVN 機能セットの拡張および改善を行っているため、これらのメリットはリリースと共に拡大します。

既存の Red Hat OpenStack Platform (RHOSP) デプロイメントで ML2/OVS メカニズムドライバーが使用されている場合、ML2/OVS メカニズムドライバーを ML2/OVN メカニズムドライバーに置き換えるメリットおよび現実性の評価を今すぐ開始してください。



注記

ML2/OVS から ML2/OVN への移行を試みる前に、事前サポートケースを作成する必要があります。事前サポートケースを作成しない場合、Red Hat では移行をサポートしません。事前ケースを送信する方法を参照してください。

この評価の初期段階で、Red Hat Technical Account Manager または Red Hat Global Professional Services との連携を始めてください。移行を選択した場合、必要な事前サポートケースの作成を支援するのに加えて、以下の基本的な質問から作業を開始して、Red Hat は計画および準備をお手伝いすることができます。

移行すべきか？

Red Hat では、ほとんどのデプロイメントで ML2/OVN が適切な選択であると考えています。さまざまな理由により、一部のデプロイメントでは、ML2/OVS の使用がより適切です。「[ML2/OVN メカニズムドライバーの制約](#)」および「[ML2/OVS から ML2/OVN へのインプレースマイグレーション：検証済みのシナリオおよび禁止されるシナリオ](#)」を参照してください。

いつ移行すべきか？

タイミングは、お客様のビジネスニーズや Red Hat による ML2/OVN オファリングに対する継続的な改善のステータスなど、多くの要素に依存します。たとえば、セキュリティーグループのログインは、RHOSP の今後のリリースで予定されています。この機能が必要な場合は、この機能が利用可能になった後に移行を計画する方が望ましいでしょう。「[ML2/OVN メカニズムドライバーの制約](#)」を参照してください。

インプレースマイグレーションか並列移行か？

さまざまな要因に応じて、以下の移行の基本アプローチのいずれかを選択できます。

- 並列移行:ML2/OVN を使用する新しい並列デプロイメントを作成し、運用をそのデプロイメントに移動します。
- インプレースマイグレーション。本書で説明するように、`ovn-migration.sh` スクリプトを使用します。Red Hat は、RHOSP director が管理するデプロイメントでのみ `ovn_migration.sh` スクリプトをサポートする点に注意してください。

ovs-firewall ファイアウォールドライバーを使用した状態で、ML2/OVS から ML2/OVN メカニズムドライバーに移行することができます。iptables_hybrid ファイアウォールドライバーを使用する移行はサポートされていません。iptables_hybrid デプロイメントで使用される中間 linux_bridge インターフェースは、移行ツールと互換性がありません。



警告

ML2/OVS から ML2/OVN への移行により、環境が完全に回復できない可能性がある方法で環境が変更されます。移行が失敗した場合や中断した場合には、OpenStack 環境が動作不能のままになる可能性があります。実稼働環境で移行を行う前に、事前サポートケースを作成します。次に、Red Hat Technical Account Manager または Red Hat Global Professional Services と連携してバックアップおよび移行プランを作成し、実稼働環境に極めて近いステージ環境で移行をテストします。

1.1. ML2/OVN メカニズムドライバーの制約

ML2/OVS メカニズムドライバーで利用可能な機能の一部は、ML2/OVN メカニズムドライバーではまだサポートされていません。

1.1.1. ML2/OVN ではまだサポートされていない ML2/OVS 機能

機能	備考	本機能の経緯
VLAN プロジェクト (テナント) ネットワーク上での分散仮想ルーター (DVR) と OVN の組み合わせ	<p>FIP トラフィックは、ML2/OVN および DVR を使用する VLAN テナントネットワークに渡されません。</p> <p>DVR は、新規の ML2/OVN デプロイメントおよび DVR が有効化された ML2/OVS デプロイメントから移行された ML2/OVN デプロイメントでデフォルトで有効にされます。VLAN テナントネットワークで OVN を使用する必要がある場合は、DVR を無効にすることができます。DVR を無効にするには、環境ファイルに以下の行を追加します。</p> <pre>parameter_defaults: NeutronEnableDVR: false</pre>	Bug 1704596 Bug 1766930
OVN と DHCP の組み合わせでのベアメタルマシンのプロビジョニング	<p>OVN 上の組み込み型 DHCP サーバーは、現状ベアメタルノードをプロビジョニングすることができません。プロビジョニングネットワーク用に、DHCP を提供することができません。iPXE のチェーンブートにはタグ付け (dnsmasq の --dhcp-match) が必要ですが、OVN DHCP サーバーではサポートされていません。</p>	Bug 1622154

1.1.2. OVN に関する主な制約

外部ポートは論理ルーターのゲートウェイポートと共存しないため、VLAN テナントネットワークでは、VF (直接) ポートでの North-South ルーティングは SR-IOV では機能しない。[Bug #1875852](#) を参照してください。

1.2. ML2/OVS から ML2/OVN へのインプレースマイグレーション: 検証済みのシナリオおよび禁止されるシナリオ

Red Hat では、インプレースマイグレーションのシナリオのテストと改良を続けています。Red Hat Technical Account Manager または Global Professional Services と連携して、OVS デプロイメントが有効なインプレースマイグレーションのシナリオの条件を満たしているかどうかを判断します。

1.2.1. 検証済みの ML2/OVS から ML2/OVN への移行シナリオ

DVR から DVR へ

開始時点: RHOSP 16.1.1 以降と OVS および DVR の組み合わせ。

終了時点: 同じ RHOSP バージョンおよびリリースと OVS および DVR の組み合わせ。

SR-IOV は開始環境には存在せず、移行中または移行後に追加されませんでした。

集中ルーティングと SR-IOV および Virtual Function (VF) ポートのための組み合わせ

開始時点: RHOSP 16.1.1 以降と OVS (DVR なし) および SR-IOV の組み合わせ。

終了時点: 同じ RHOSP バージョンおよびリリースと OVS (DVR なし) および SR-IOV の組み合わせ。

負荷は SR-IOV Virtual Function (VF) ポートだけを使用しました。SR-IOV Physical Function (PF) ポートにより、移行に失敗していました。

1.2.2. 検証されていない ML2/OVS から ML2/OVN へのインプレースマイグレーションのシナリオ

Red Hat から根本の問題が解決されると通知があるまで、以下のシナリオでは ML2/OVS から ML2/OVN へのインプレースマイグレーションを行うことはできません。

OVS デプロイメントで VXLAN が使用される、ターゲットデプロイメント: RHOSP 16.2.0

RHOSP では、ML2/OVN と VXLAN ネットワークの組み合わせはまだサポートされていません。移行プロセスには、VXLAN ネットワークを Geneve に変換するステップが含まれます。移行のターゲットバージョンが RHOSP 16.2.0 の場合、バグにより予想される VXLAN から Geneve への変換が阻止され、ネットワークは VXLAN として設定されたままになります。

https://bugzilla.redhat.com/show_bug.cgi?id=2003708 を参照してください。

このバグは、RHOSP 16.2 上の ML2/OVN への移行にのみ影響します。RHOSP 16.1 上の ML2/OVN への移行には影響しません。

OVS デプロイメントで iptables_hybrid ファイアウォールドライバーを使用する

openvswitch ファイアウォールドライバーを使用して ML2/OVS から ML2/OVN メカニズムドライバーに移行することができませんが、iptables_hybrid ファイアウォールドライバーではサポートされません。iptables_hybrid ファイアウォールドライバーを使用した移行はサポートされていません。

詳細は、https://bugzilla.redhat.com/show_bug.cgi?id=2011450 を参照してください。

OVS デプロイメントでネットワーク機能仮想化 (NFV) が使用される

Red Hat は ML2/OVN および NFV を使用する新しいデプロイメントをサポートしますが、ML2/OVS および NFV を使用するデプロイメントから ML2/OVN への移行は正常にテストされていません。この問題の進捗を確認するには、[Bug 1925290](#) を参照してください。

SR-IOV と Physical Function (PF) ポートの組み合わせ

いずれの負荷が SR-IOV PF ポートを使用する場合でも、移行テストが失敗しました。この問題の進捗を確認するには、[Bug 1879546](#) を参照してください。

OVS がトランクポートを使用する

ML2/OVS デプロイメントでトランクポートが使用される場合は、ML2/OVS から ML2/OVN への移行を実施しないでください。OVN 環境では、移行によりトランキングされたポートが適切に設定されません。この問題の進捗を確認するには、[Bug 1857652](#) を参照してください。

DVR を使用する VLAN プロジェクト (テナント) ネットワーク

DVR および VLAN プロジェクトネットワークを使用する ML2/OVN の構成に移行しないでください。集中ルーティングを使用する ML2/OVN に移行することができます。この問題の進捗を確認するには、[Bug 1766930](#) を参照してください。

1.2.3. ML2/OVS から ML2/OVN へのインプレースマイグレーションおよびセキュリティーグループルール

元の ML2/OVS デプロイメントのカスタムセキュリティーグループルールがターゲットの ML2/OVN デプロイメントと互換性があることを確認します。

たとえば、デフォルトのセキュリティーグループには、DHCP サーバーへの送信を許可するルールが含まれます。ML2/OVS デプロイメントでこれらのルールを削除した場合、ML2/OVS は DHCP サーバーへの送信を許可する暗黙的なルールを自動的に追加します。これらの暗黙的なルールは ML2/OVN ではサポートされません。そのため、ターゲットの ML2/OVN 環境では、DHCP およびメタデータのトラフィックは DHCP サーバーに到達せず、インスタンスは起動しません。この場合は、DHCP アクセスを回復するには、以下のルールを追加します。

```
# 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. ML2/OVS から ML2/OVN への移行の準備

環境の評価と準備は、移行を成功させるために重要です。Red Hat Technical Account Manager または Global Professional Services は、以下の手順の実施をガイドします。

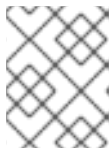
前提条件

- 移行前のデプロイメントが、Red Hat OpenStack Platform (RHOSP) 16.1 以降である。
- 移行前のデプロイメントが、**iptables_hybrid** ファイアウォールドライバーを使用していない。**iptables_hybrid** デプロイメントで使用される中間 **linux_bridge** インターフェースは、移行ツールと互換性がありません。

- RHOSP のデプロイメントが最新化されている。つまり、OpenStack バージョンのアップグレードまたは更新が必要な場合は、まずアップグレードまたは更新を実行し、続いて ML2/OVS から ML2/OVN への移行を実施します。
- Red Hat Technical Account Manager または Global Professional Services と連携して移行を計画し、事前サポートケースを作成している。[事前ケースを送信する方法](#) を参照してください。

手順

1. ML2/OVN ステージデプロイメントを作成して、ターゲット ML2/OVN デプロイメントのベースライン設定を取得し、ターゲットデプロイメントの実現可能性をテストします。計画された移行後の本番デプロイメントと同じ基本的なロール、ルーティング、およびトポロジーを使用して、ステージデプロイメントを設計します。**overcloud-deploy.sh** ファイルと、環境ファイルなど、デプロイメントによって参照されるすべてのファイルを保存します。移行ターゲット環境を設定するには、この手順の後半でこれらのファイルが必要になります。



注記

これらのファイルは、ステージデプロイメントの作成および移行でのみ使用してください。移行後に再利用しないでください。

2. ML2/OVS デプロイメントで VXLAN または GRE プロジェクトネットワークを使用する場合は、`setup-mtu-tl` のステップ後に最大 24 時間待機します。
 - この待機時間により、仮想マシンインスタンスは DHCP リースを更新し、新しい MTU 値を受け取ることができます。この間に、一部のインスタンスに MTU を手動で設定し、一部のインスタンスを再起動する必要がある場合があります。
 - 24 時間はデフォルト設定の 86400 秒に基づいた時間です。実際の時間は、`/var/lib/config-data/puppet-generated/neutron/etc/neutron/dhcp_agent.ini` の `dhcp_renewal_time` および `/var/lib/config-data/puppet-generated/neutron/etc/neutron/neutron.conf` の `dhcp_lease_duration` のパラメーターにより異なります。
3. `python3-networking-ovn-migration-tool` をインストールします。

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

@container-tools 引数は、コンテナツールがまだ存在しない場合は、それらもインストールします。

4. アンダークラウドにディレクトリーを作成し、Ansible Playbook をコピーします。

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

5. ML2/OVN ステージデプロイメントファイルを `~/ovn_migration` などの移行ホームディレクトリーにコピーします。ステージ移行デプロイメントファイルには、**overcloud-deploy.sh** と、環境ファイルなど、デプロイメントによって参照されるすべてのファイルが含まれます。**overcloud-deploy.sh** のコピーの名前を **overcloud-deploy-ovn.sh** に変更します。このスクリプトは移行にのみ使用してください。他の目的には使用しないでください。

6. 以下の一覧で移行シナリオを確認し、**overcloud-deploy-ovn.sh** の **openstack deploy** コマンドをカスタマイズするための適切な手順を実施します。

シナリオ 1: DVR から DVR へ コンピュートノードが外部ネットワークに接続できる

- 以下の環境ファイルを **overcloud-deploy-ovn.sh** の **openstack deploy** コマンドに追加します。環境ファイルは以下の順序で追加します。このコマンドの例では、デフォルトの **neutron-ovn-dvr-ha.yaml** ファイルを使用します。別のファイルを使用する場合は、コマンドのファイル名を置き換えます。

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

シナリオ 2: 集中ルーティングから集中ルーティングへ (DVR なし)

- デプロイメントで SR-IOV が使用されている場合は、**roles_data.yaml** ファイルの **OS::TripleO::Services::OVNMetadataAgent** を Controller ロールに追加します。
- 移行前のカスタムブリッジマッピングを維持します。
 - コントローラーノードで以下のコマンドを実行し、現在のブリッジマッピングを取得します。

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

出力例

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

- アンダークラウドで、ブリッジマッピング用の環境ファイル (**/home/stack/neutron_bridge_mappings.yaml**) を作成します。
- 環境ファイルでデフォルト値を設定します。以下に例を示します。

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

- 以下の環境ファイルを **overcloud-deploy-ovn.sh** の **openstack deploy** コマンドに追加します。環境ファイルは以下の順序で追加します。お使いの環境で SR-IOV が使用されない場合は、**neutron-ovn-sriov.yaml** ファイルを省略します。**ovn-extras.yaml** ファイルは存在していませんが、**openstack deploy** コマンドを実行する前に **ovn_migration.sh** スクリプトにより作成されます。

```
-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
```

- カスタムのネットワーク変更は、移行前と同じままにします。

シナリオ 3: 集中ルーティングから DVR へ (Geneve タイプドライバーおよび br-ex 経由で外部ネットワークに接続されたコンピュータードを使用)



警告

ML2/OVS デプロイメントで集中ルーティングおよび VLAN プロジェクト (テナント) ネットワークが使用される場合は、DVR を使用する ML2/OVN に移行しないでください。集中ルーティングを使用する ML2/OVN に移行することができます。この制限の進捗を追跡するには、[Bug 1766930](#) を参照してください。

- コンピュータードが **br-ex** ブリッジを介して外部ネットワークに接続されていることを確認します。たとえば、`compute-dvr.yaml` 等の環境ファイルで以下のように設定します。

```
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. すべてのユーザーに **overcloud-deploy-ovn.sh** ファイルの実行権限があることを確認します。このスクリプトには、移行プロセス中に実行権限が必要です。

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

8. **export** コマンドを使用して、以下の移行関連の環境変数を設定します。以下に例を示します。

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

- `STACKRC_FILE`: アンダークラウドの `stackrc` ファイル
デフォルト: `~/stackrc`
- `OVERCLOUDRC_FILE`: アンダークラウドの `overcloudrc` ファイル
デフォルト: `~/overcloudrc`
- `OVERCLOUD_OVN_DEPLOY_SCRIPT`: デプロイメントスクリプト
デフォルト: `~/overcloud-deploy-ovn.sh`
- `PUBLIC_NETWORK_NAME`: パブリックネットワークの名前
デフォルト: `public`
- `IMAGE_NAME`: テストサーバーのブートに使用する glance イメージの名前または ID

デフォルト: **cirros**

イメージは、検証前/検証後のプロセス時に自動的にダウンロードされます。

- **VALIDATE_MIGRATION**: 移行リソースを作成して移行を検証します。移行スクリプトは、移行開始前にサーバーを起動し、移行後にサーバーが到達可能であることを検証します。
デフォルト: **True**



警告

移行の検証には、少なくとも2つの利用可能な Floating IP アドレス、2つのネットワーク、2つのサブネット、2つのインスタンス、および2つの admin ルーターが必要です。

また、**PUBLIC_NETWORK_NAME** で指定されるネットワークには、利用可能な Floating IP アドレスが必要で、アンダークラウドから ping できる必要があります。

お使いの環境がこれらの要件を満たさない場合には、**VALIDATE_MIGRATION** を **False** に設定します。

- **SERVER_USER_NAME**: 移行インスタンスへのログインに使用するユーザー名。
デフォルト: **cirros**
 - **DHCP_RENEWAL_TIME**: DHCP エージェント設定ファイルで設定する DHCP 更新時間 (秒単位)。
デフォルト: 30
9. **ovn-migration** ディレクトリーにあり、**ovn_migration.sh generate-inventory** コマンドを実行して、インベントリーファイル **hosts_for_migration** および **ansible.cfg** ファイルを生成します。
- ```
$ ovn_migration.sh generate-inventory | sudo tee -a /var/log/ovn_migration_output.txt
```
10. **hosts\_for\_migration** ファイルで正確性を確認してください。
- 一覧がお使いの環境と一致していることを確認します。
  - 各ノードに ovn コントローラーがあることを確認します。
  - リスト見出し ([**ovn-controllers**] など) がリスト項目に含まれていないことを確認します。
  - ovn 移行ディレクトリーから、**ansible -i hosts\_for\_migration -m ping all** コマンドをすべて ping します。
11. 元の ML2/OVS デプロイメントで VLAN プロジェクトネットワークを使用している場合には、ステップ 18 に進みます。
12. **ovn\_migration.sh setup-mtu-t1** を実行します。これにより、DHCP エージェントが実行しているすべてのノードで、**/var/lib/config-data/puppet-generated/neutron/etc/neutron/dhcp\_agent.ini** 内の **dhcp\_renewal\_time** を設定する内部

neutron DHCP サーバーの T1 パラメーターは短くなります。

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

13. 元の OVS デプロイメントで VXLAN または GRE プロジェクトネットワークを使用している場合には、DHCP リースがすべての仮想マシンインスタンスで更新されるまでお待ちください。リース更新設定およびインスタンス数によっては、最大 24 時間かかる場合があります。
14. VXLAN または GRE プロジェクトネットワークに静的 IP を割り当てるインスタンスがある場合は、それらのインスタンスの設定を手動で変更し、新しい Geneve MTU (現在の VXLAN MTU から 8 バイト減) を設定する必要があります。たとえば、VXLAN ベースの MTU が 1450 の場合は、これを 1442 に変更します。



### 注記

このステップは、VXLAN または GRE プロジェクトネットワークに静的 IP の割り当てと MTU 設定を手動で提供している場合にのみ行います。デフォルトでは、DHCP が IP 割り当てと MTU 設定を提供します。

15. T1 パラメーターが既存の仮想マシンに伝播されていることを確認します。
  - コンピュートノードのいずれかに接続します。
  - プロジェクトネットワークに接続された仮想マシンタップのいずれかで tcpdump を実行します。  
T1 が正常に伝搬された場合、約 30 秒間隔で要求が実行されるはずです。

```
[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 300
13:17:56.249899 IP 192.168.99.3.bootps > 192.168.99.5.bootpc: BOOTP/DHCP, Reply, length 355
```



### 注記

cirros 仮想マシンでは、この検証はできません。cirros udhcpd 実装は、DHCP オプション 58 (T1) に応答しません。完全な Linux 仮想マシンに属するポートで、この検証を試みます。Red Hat では、お使いのシステムが実行するさまざまなタイプのワークロードを確認することを推奨しています (Windows、Linux の異なるフレーバーなど)。

16. DHCP の T1 パラメーターへの変更を反映するように仮想マシンインスタンスが更新されていない場合は、再起動します。
17. 移行前の VXLAN および GRE ネットワークの MTU を減らします。

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

このステップでは、ネットワークごとに MTU を減らし、`adapted_mtu` で完了したネットワークにタグを付けます。このツールは、VXLAN/GRE 以外のネットワークを無視します。そのため、プロジェクトネットワークに VLAN を使用しても、このステップでは値を変更することは想定されていません。

18. ML2/OVN への移行後に使用する新しいコンテナイメージを準備します。

- a. ホームディレクトリーに **containers-prepare-parameter.yaml** ファイルがない場合は作成します。

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

- b. **containers-prepare-parameter.yaml** が `$HOME/overcloud-deploy-ovn.sh` ファイルおよび `$HOME/overcloud-deploy.sh` ファイルの末尾にあることを確認します。

- c. **containers-prepare-parameter.yaml** ファイルの `neutron_driver` を `ovn` に変更します。

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

- d. `neutron_driver` への変更を確認します。

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

- e. イメージを更新します。

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



### 注記

**containers-prepare-parameter.yaml** ファイルへの完全パスを指定します。そうでない場合には、イメージ一覧を更新したり、エラーメッセージを表示したりせずに、コマンドは非常に迅速に完了します。

19. アンダークラウドにおいて、更新されたイメージを検証します。

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

リストは以下の例のようになります。これには、OVN データベース、OVN コントローラー、メタデータエージェント、および `neutron` サーバーエージェント用のコンテナが含まれません。

```
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

```

## 1.4. ML2/OVS から ML2/OVN への移行

`ovn-migration` スクリプトは、ML2/OVS から ML2/OVN へのインプレースマイグレーションに関する環境設定、移行、およびクリーンアップタスクを実行します。

### 前提条件

- 「[ML2/OVS から ML2/OVN への移行の準備](#)」の手順が完了していること。

### 手順

- `ovn_migration.sh start-migration` を実行して移行プロセスを開始します。tee コマンドは、トラブルシューティング目的のスクリプト出力のコピーを作成します。

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

### 結果

スクリプトは以下のアクションを実行します。

- 移行前のリソース (ネットワークおよび仮想マシン) を作成して、既存のデプロイメントと最終移行を検証します。
- `br-int` ではなく、一時ブリッジ `br-migration` を使用して、参照実装サービスと共に OVN をデプロイするために、オーバークラウドスタックを更新します。一時ブリッジは、移行時のダウンタイムを抑えるのに役立ちます。
- `neutron-ovn-db-sync-util` を実行して OVN ノースバウンドデータベースを生成します。このユーティリティーは、OVN ノースバウンドデータベースで等価なリソースを作成するために Neutron データベースを調べます。
- `br-int` から `br-migration` に既存のリソースのクローンを作成し、`ovn` が `br-migration` で同じリソース UUID を検出できるようにします。
- `br-migration` ではなく `br-int` に `ovn-controller` を再度割り当てます。
- 以下に示す ML2/OVN が使用していないノードリソースを削除します。
  - ネットワーク名前空間 (`fip`、`snat`、`qrouter`、`qdhcp`) をクリーンアップします。
  - br-int** の不要なパッチポートを削除します。
  - br-tun** および **br-migration** ovs ブリッジを削除します。
  - br-int** から **qr-**、**ha-**、および **qg-** で始まるポートを削除します (`neutron-netns-cleanup` を使用)。

- Networking サービス API を使用して、Networking サービス (neutron) エージェントおよび Networking サービス HA の内部ネットワークをデータベースから削除します。
- 移行前のリソースで接続を検証します。
- 移行前のリソースを削除します。
- 移行後のリソースを作成します。
- 移行後のリソースで接続を検証します。
- 移行後のリソースをクリーンアップします。
- デプロイメントツールを再度実行して、**br-int** 上の OVN を更新します。