

# 非接続モードでのモビリティワークロードのためのUCSでのRHOSPの導入ガイド

## 内容

---

[はじめに](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[バックグラウンド情報](#)

[RHOSPについて](#)

[RHOSPアーキテクチャ](#)

[クラウド下](#)

[Undercloudコンポーネント](#)

[クラウド超過](#)

[コントローラ](#)

[計算](#)

[ストレージ](#)

[ローカルリポジトリ \( REPOサーバ \)](#)

[RHOSPネットワーキング](#)

[RHOSP物理接続](#)

[RHOSP論理接続](#)

[ハードウェアパラメータの調整](#)

[ハイパーバイザー-KVMのインストールとネットワーク作成](#)

[REPOおよびDirector VMの作成](#)

[REPOおよびDirector VM作成の前提条件](#)

[リポジトリVMの作成](#)

[ディレクタVMの作成](#)

[ローカル・オフラインREPO作成](#)

[RHOSPクラウドの導入](#)

[前提条件](#)

[入カファイルの更新](#)

[アンダークラウドの導入](#)

[クラウドの導入](#)

[Horizon Dashboardへのアクセス](#)

[RHOSPクラスターのヘルスチェック](#)

[要約](#)

---

## はじめに

このドキュメントでは、Cisco VPC-DIをサポートするためにC220 M6 UCSサーバにRHOSPを導入するためのフレームワークについて説明します。

## 前提条件

### 要件

Red Hat OpenStack Platform(RHOSP)に関する知識とRed Hat Enterprise Linux(RHEL)に関する高度なスキルを持っていることが推奨されます。また、仮想化とネットワーキングの概念を確実に理解することも必要です。

### 使用するコンポーネント

このドキュメントの内容は、特定のソフトウェアやハードウェアのバージョンに限定されるものではありません。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されたものです。このドキュメントで使用するすべてのデバイスは、クリアな(デフォルト)設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

## バックグラウンド情報

このガイドでは、RHOSPとUnified Computing System(UCS)インフラストラクチャの統合の概要を説明し、拡張性、信頼性、およびパフォーマンスの最適化に重点を置いています。

ベストプラクティスの詳細を説明し、OpenStack TripleOの導入にスクリプトベースの自動化を使用して、UndercloudおよびOvercloudアーキテクチャで構成されます。

この導入ガイドを使用することで、組織はCisco Virtual Packet Core - Distributed Instance(VPC-DI)ベースのモビリティ仮想ネットワーク機能(VNF)に対応するように調整された、堅牢で効率的なRHOSPクラウドインフラストラクチャを実現できます。

## RHOSPについて

RHOSPは、オープンソースのOpenStackプロジェクト上に構築され、Red Hatによって統合およびサポートされている、エンタープライズグレードのプライベートクラウドソリューションです。

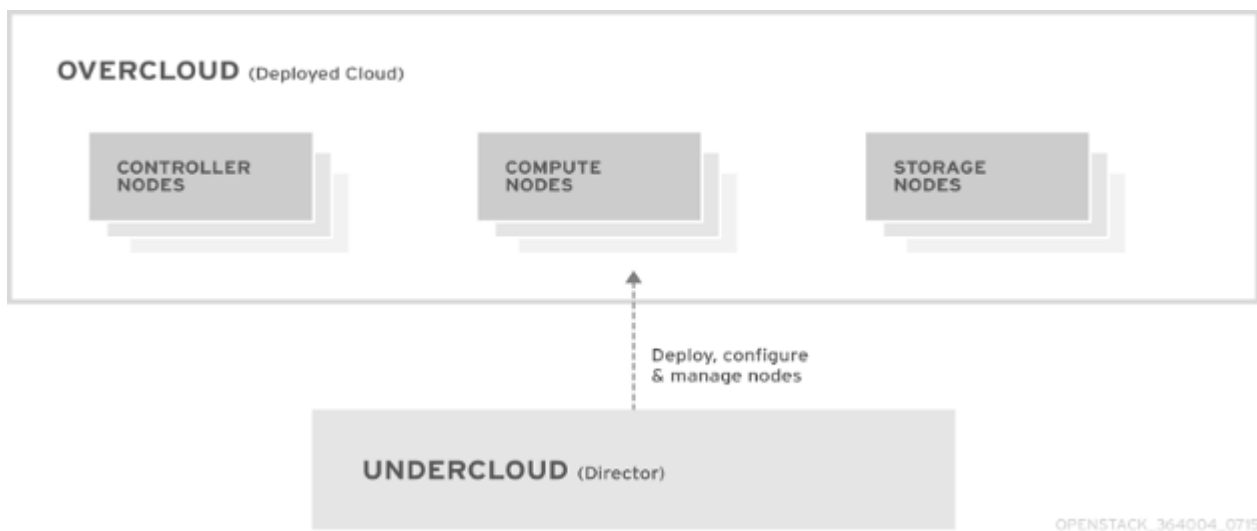
。組織は、仮想マシン(VM)、ネットワーキング、およびストレージのInfrastructure-as-a-Service(IaaS)をオンデマンドで導入および管理できます。

高可用性(HA)、ネットワーク機能の仮想化、カスタマイズ可能な導入などの機能を提供します。

## RHOSPアーキテクチャ

RHOSPは、主にOpenStack TripleOプロジェクトに基づいています。Openstackは、完全なRHOSP環境をインストールおよび管理するためのツールセットとして機能するDirectorを使用します。

RHOSPは、スケーラブルで柔軟なクラウドインフラストラクチャを提供するように設計されています。このアーキテクチャは、UndercloudとOvercloudという2つの主要コンポーネントで構成されています。



## クラウド下

アンダークラウドは、RHOSPディレクタツールセットを含むメイン管理ノードです。これは、OpenStack環境 (オーバークラウド) を形成するOpenStackノードをプロビジョニングおよび管理するためのコンポーネントを含む、単一システムのOpenStackインストールです。

## Undercloudコンポーネント

アンダークラウドでは、ベースツールセットとしてOpenStackコンポーネントが使用されています。各コンポーネントは、アンダークラウド上の個別のコンテナ内で動作します。

- OpenStack Identity (キーストーン) : ディレクタ・コンポーネントの認証と承認を行います。
- OpenStack Bare Metal(ironic) : ベアメタルノードを管理します。
- OpenStack ネットワーキング(neutron)およびOpen vSwitch – ベアメタルノードの制御ネットワークワーキング
- OpenStack オークストレーション (一時的な熱) : ディレクタがオーバークラウドイメージをディスクに書き込んだ後に、ノードのオークストレーションを行います。

## クラウド超過

オーバークラウドとは、アンダークラウドを使用して作成された結果のRHOSP環境です。これには、お客様が作成を目指すOpenStack Platform(OSP)環境に基づいて定義されるさまざまなノードロールが含まれます。

## コントローラ

コントローラノードは、OpenStack環境に管理、ネットワーク、およびHAを提供します。推奨されるOpenStack環境には、3つのコントローラノードがHAクラスタに含まれています。

## 計算

コンピューティングノードは、OpenStack環境にコンピューティングリソースを提供します。コンピュートノードは、ネットワーク要件に基づいて徐々にスケールイン/スケールアウトできます。デフォルトのコンピュートノードには、次のコンポーネントが含まれています。

- OpenStackコンピューティング(nova)
- カーネルベースのVM(KVM)/クイックエミュレータ(QEMU)
- vSwitchを開く

## ストレージ

ストレージノードは、OpenStack環境にストレージを提供します。



注：お客様のネットワークでUndercloud/DirectorおよびOffline Repository(REPO)を導入するには、複数のアプローチがあります。ベアメタルノードに直接導入することも、KVMハイパーバイザ上のVMとして導入することもできます。現在の導入ガイドでは、Director UCSサーバがKVM (ハイパーバイザ) をホストして、複数のVMを最上位に導入

---

---

します。RHOSP DirectorノードとOffline-REPOノードは、KVM Hypervisor上のVMとして導入されます。

---

## ローカルリポジトリ ( REPOサーバ )

Redhatは、[コンテンツ配信ネットワーク](#)(CDN)からパッケージをダウンロードするために使用できるreposyncというユーティリティを提供しています。特定のチャンネルからすべてのパッケージをダウンロードするには、システムがそのチャンネルにサブスクライブされている必要があります。システムが必要なチャンネルにサブスクライブされていない場合、reposyncはローカルシステムでこれらのパッケージをダウンロードおよび同期できません。

リポジトリは、.repo拡張子で終わるファイルによって、/etc/yum.repos.d/パスに設定されます。同じファイルに複数のリポジトリを定義できます。

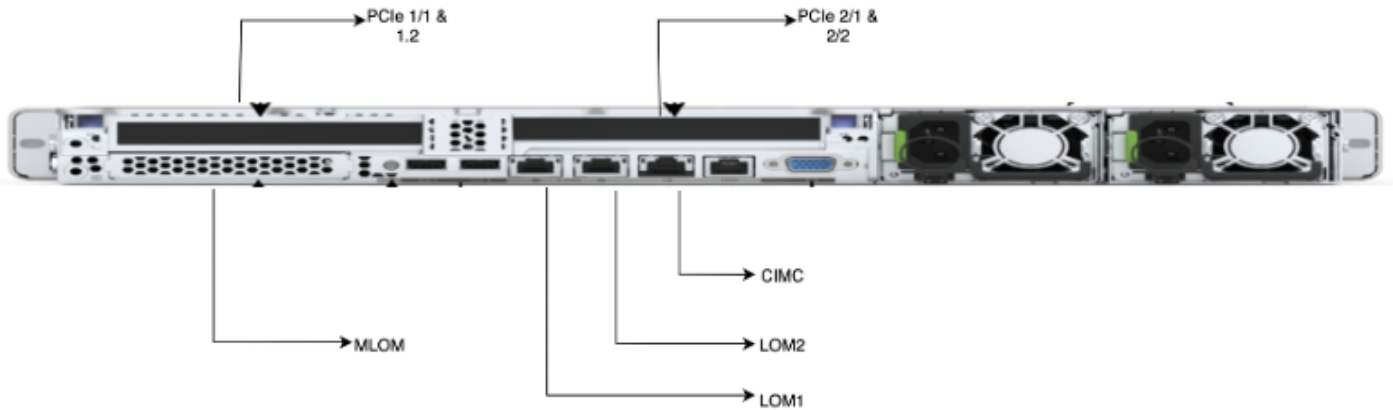
## RHOSPネットワーク

ネットワークサービス(neutron)は、RHOSPのソフトウェア定義型ネットワーク(SDN)コンポーネントです。RHOSPネットワークサービスは、VMインスタンスとの間の内部および外部トラフィックを管理し、ルーティング、セグメンテーション、DHCP、メタデータなどのコアサービスを提供します。仮想ネットワーク機能のAPIと、スイッチ、ルータ、ポート、およびファイアウォールの管理を提供します。

RHOSPディレクタは、OpenStackサービスを異なる独立したネットワークにマッピングします。各トラフィックタイプを伝送するネットワークは、Cisco Integrated Management Controller(CIMC)、プロビジョニング、内部API、ストレージデータ、ストレージ管理、テナントおよび外部(セキュアシェル(SSH)およびOperations, Administration, and Maintenance(OAM))です。

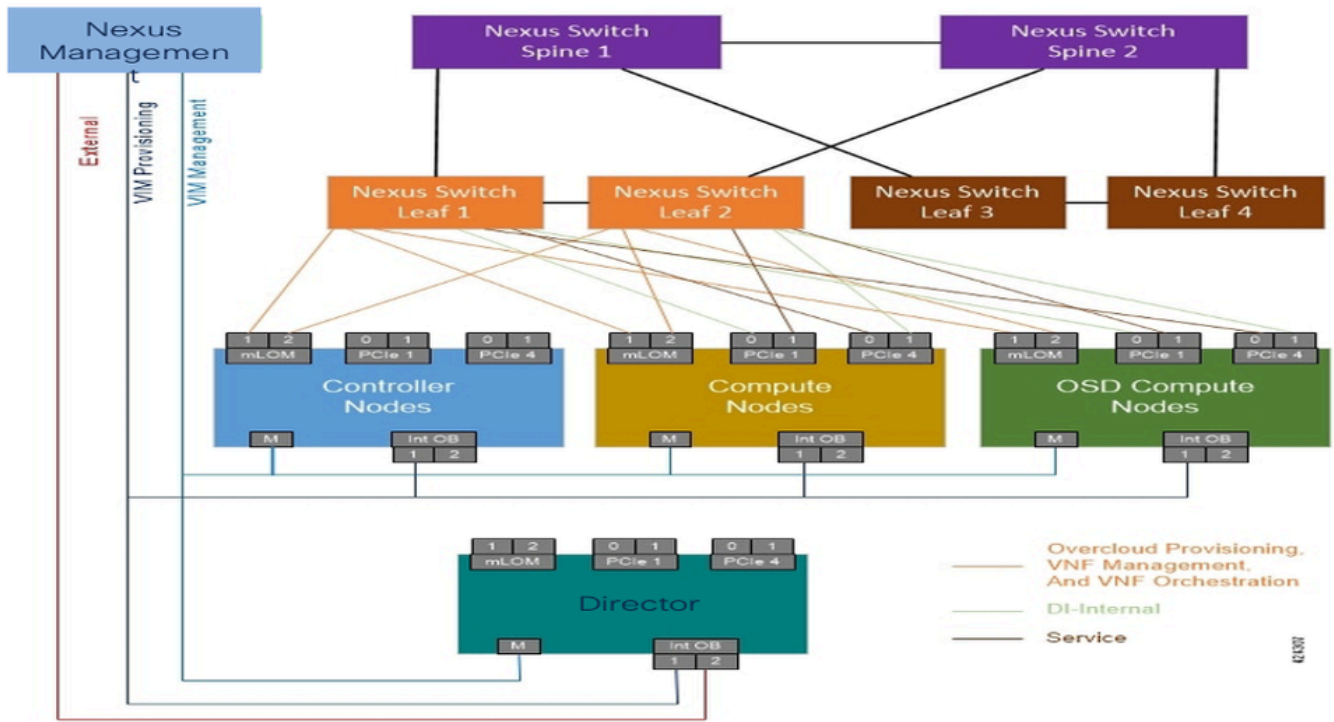
## RHOSP物理接続

RHOSPの導入では、Cisco UCS C220 M6サーバの異なる物理ポートを異なる接続の目的で使用します。



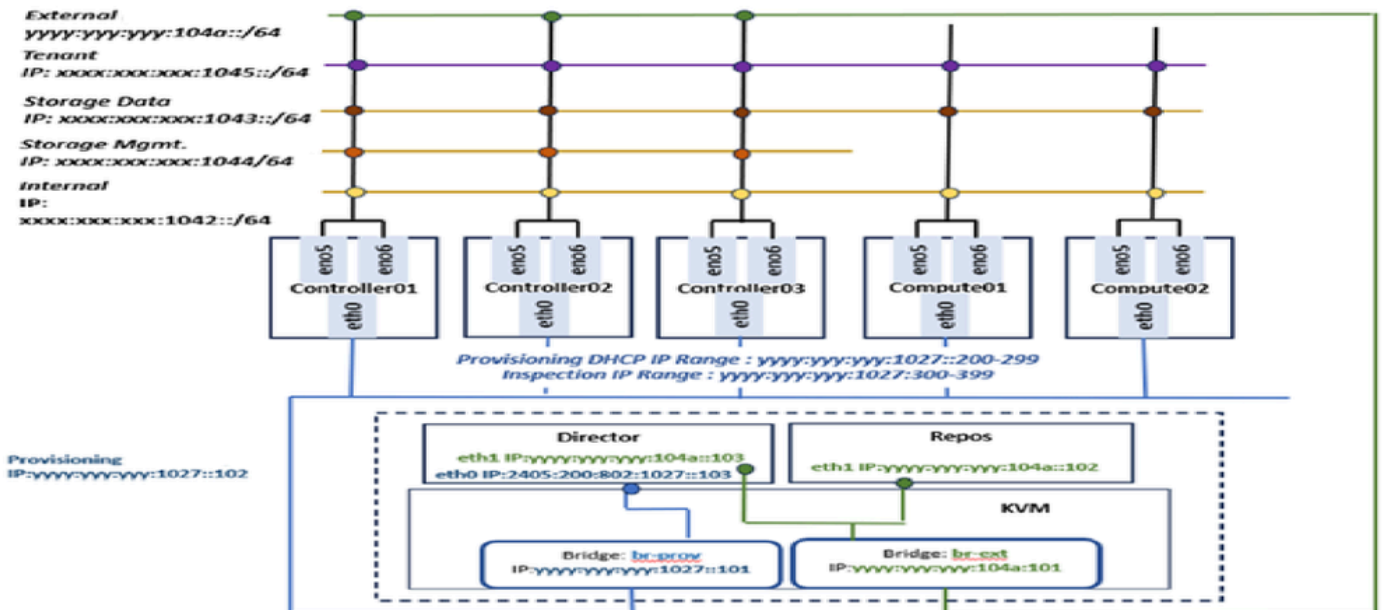
シリアル番号	物理ポート	詳細
1.	CIMC	CIMCは、サーバのプロビジョニングと管理にアウトオブバンド接続を提供します。
2.	シングルルートI/O仮想化(SR-IOV)/Peripheral Component Interconnect Express(PCIe)	PCIeネットワークインターフェイスカード(NIC)は、DI内部のコンピュートノードとVNFのサービスネットワークで使用されます。
を選択します。	マザーボード上のモジュラ Lan(MLOM)	MLOMポートはボンドとして設定されます。 osp_external、osp_internal、osp_tenant、osp_external、osp_storage_data、osp_storage_mgmtは、内部通信にMLOMポートを使用します。
4.	マザーボード上のLAN(LOM)	ダイレクタはLOM1およびLOM2ポートを使用し、演算とコントローラはLOM1ポートのみを使用します。 LOM1は、すべてのサーバでOpenstackを導入またはプロビジョニングするために使用されます。 LOM2は、ダイレクタ上でOAM (外部ネットワーク)として使用されます。

この図は、サーバとの物理的な接続を示しています。



## RHOSP論理接続

RHOSPネットワークには、クラウド内の異なるサービスに対応する複数のサブネットがあります。



- OSP\_CIMC:

CIMCは、すべてのUCSサーバの管理を制御するインテリジェントプログラミング管理インター

フェイス(IPMI)です。このCIMCネットワークは、すべてのUCSサーバのスタンドアロンCIMCポートに設定されます。

- OSP\_Provisioning:

このネットワークは、オーバークラウド導入時のコンピューティングサーバとコントローラサーバのプロビジョニングとPreboot Execution Environment(PXE)ブート管理、およびDHCP IPの取得を担当します。簡素化と互換性のために、プロビジョニングネットワークはすべてのUCSサーバのLOM1ポートでネイティブVLANとして設定されています。このプロビジョニングネットワークは、すべてのサーバへのクラウドの導入を担当します。

ディレクタサーバの仮想化により、ディレクタVMが他のサーバと通信するには、ブリッジネットワークをKVM上に作成する必要がありました。

- OSP\_内部 :

内部APIネットワークは、neutron、nova、keystoneなどのOpenStackサービス間の通信に使用されます。

OSP\_Internalネットワークは、コントローラおよびコンピュータード上の結合されたMLOMポートで設定されます。

- OSP\_テナント :

テナントネットワークは、VNF管理用のクラウドプロジェクト内にデフォルトで作成されます。現在のセットアップでは、VNF導入用に単一のOpenstackプロジェクトのみが作成されます。

OSP\_Tenantネットワークは、コントローラおよびコンピュータード上の結合されたMLOMポートで設定されます。

- OSP\_外部 :

外部ネットワークは、すべての外部アクセス (SSHなど) とAPIネットワークに使用されます。

OSP\_Externalネットワークは、ディレクタノードのLOM2ポートと、コントローラおよびコンピュータードの結合MLOMポートに設定されます。

- OSP\_Storage\_Data:

OSP\_Storageネットワークは、ストレージへのアクセスに関連するすべての操作に使用されます。これは、ストレージにアクセスする必要があるCEPHサービスとVNF間の通信に必要です。コントローラ、コンピュートノード、およびCEPHで使用されます。

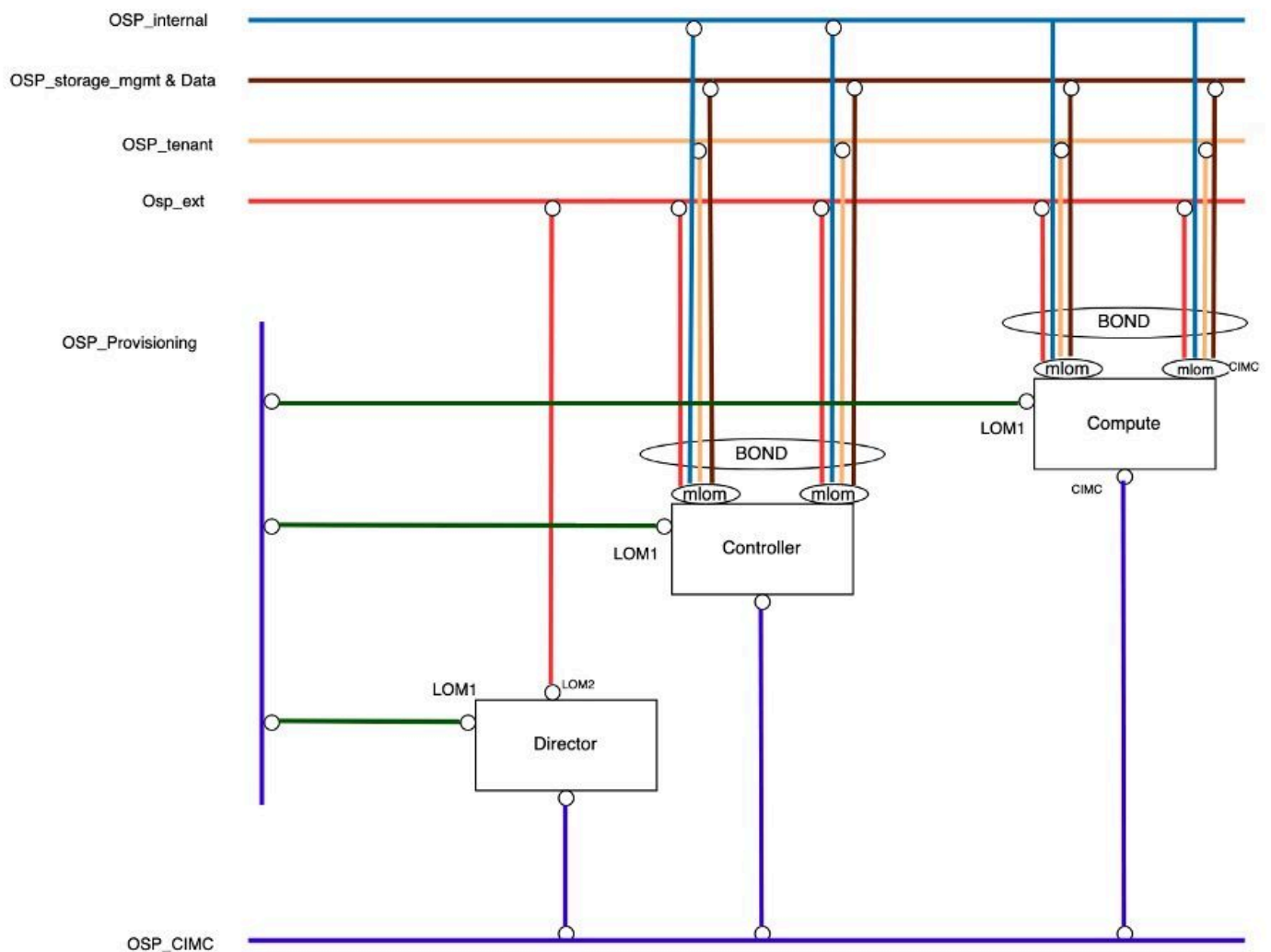
OSP\_Storage\_Dataネットワークは、コントローラおよびコンピュートノード上の結合されたMLOMポートで構成されます。

- OSP\_Storage\_Management:

OpenStackオブジェクトストレージは、このネットワークを使用して、コントローラとコンピュートノードの間で形成されるストレージクラスタ内の参加レプリカノード間でデータオブジェクトを同期します。

OSP\_Storage\_Mgmtネットワークは、コントローラおよびコンピュートノード上の結合されたMLOMポートで設定されます。

この図は、アンダークラウド論理ネットワークがRHOSPクラスタ内の各タイプのノードにどのように接続されているかを示しています。



## ハードウェアパラメータの調整

- 設計に従って、IPが有効になっており、正しい管理アドレスが設定されていることを確認します。
- すべてのCIMCログインアクセスでadminユーザとそのパスワードが同じである必要があります。
- ブートストラップ ( ブート ) モードがUnified Extensible Firmware Interface(UEFI)に設定されていることを確認します。
- BIOSの設定には、必ず推奨パラメータを設定してください。
- Boot-Orderを次のように設定します。
  1. LOM-PXE(eno1)
  2. ブートハードディスクドライブ(HDD)
- LOM1(eno1)のMACアドレスを設定します。
- すべてのUCS/CIMCサーバが、推奨される最新のファームウェアバージョンでアップグレードされていることを確認します。

# ハイパーバイザ – KVMのインストールとネットワーク作成

お客様のネットワークにUndercloud/DirectorおよびOffline REPOを導入するには、いくつかの方法があります。これらは、ベアメタルノードに直接導入することも、KVMハイパーバイザ上で実行されるVMとして導入することもできます。

現在の導入ガイドでは、Director UCSサーバがKVMハイパーバイザをホストするように設定されているため、複数のVMを容易に作成できます。RHOSP DirectorノードとOffline REPOノードは、このKVMハイパーバイザ上にVMとして導入されます。



注:KVMハイパーバイザを導入するには、標準のRHEL KVMインストール手順を実行する必要があります。

- KVMが起動したら、ホスト名を更新します。  
`hostnamectl set-hostname <hostname> – スタティック`
- 外部ブリッジとプロビジョニングブリッジを設定し、インターフェイスをバインドします。

`br-prov:eth0`

`br-ext:eth1`

これらのブリッジは、ネットワークマネージャテキストユーザインターフェイス(NMTUI)GUIを使用して作成する必要があります。

- デフォルトでは、物理LOM1およびLOM2ポート用にeno1およびeno2ポートがKVMで作成され、CIMCロギングのMACアドレスを使用して同じことを相互検証します。
- ブリッジネットワークを作成し、正しいMACアドレスをマッピングしてスレーブポートを追加します。
- ブリッジネットワークの作成後に、プロビジョニングと外部ゲートウェイがKVMから到達可能であることを確認します。

## REPOおよびDirector VMの作成

### REPOおよびDirector VM作成の前提条件

```
# dnf install qemu-kvm libvirt virt-install virt-manager virt-viewer libguestfs-tools
```

- KVM for REPOおよびDirector VM Creationに必要なパッケージをインストールします。

```
! command not found...
[root@RH08KVM01 ~]#
[root@RH08KVM01 ~]# dnf install qemu-kvm libvirt virt-install virt-manager virt-viewer libguestfs-tools
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.

Red Hat Enterprise Linux 8.4.0 Appstream |
Red Hat Enterprise Linux 8.4.0 BaseOS |
Package qemu-kvm-15:4.2.0-48.module+el8.4.0+10360+630e003b.x86_64 is already installed. 564 MB/s | 6.8 MB 00:00
Package libvirt-6.0.0-35.module+el8.4.0+10230+7a9b21e4.x86_64 is already installed. 545 MB/s | 2.3 MB 00:00
Package virt-install-2.2.1-4.el8.noarch is already installed.
Package virt-manager-2.2.1-4.el8.noarch is already installed.
Package virt-viewer-9.0-9.el8.x86_64 is already installed.
Dependencies resolved.
-----
Package                               Archite
```

- KVM上に必要なディレクトリを作成します。

```
# mkdir /data # mkdir /data/offlineRepos # mkdir /data/isoImages # mkdir /data/qcow2Images # mkdir /data/images
```

- ファイルをディレクトリにコピーします。

```
# scp -r root@[remote-IP]:/root/rhel-8.4-x86_64-dvd.iso /data/isoImages/
# scp -r root@[remote-IP]:/root/offlineRepos/RHEL8.4 /data/offlineRepos/
# scp -r root@[remote-IP]:/etc/yum.repos.d/offlinedvd.repo /etc/yum.repos.d/
# scp -r root@[remote-IP]:/root/rhel-8.4-x86_64-kvm.qcow2 /data/qcow2Images/
# scp -r root@[remote-IP]:/root/OSREPO_RHEL_84.qcow2 /data/images/
# scp -r root@[remote-IP]:/root/OSREPO_DIRECTOR_84.qcow2 /data/images/
```

- ISOを/mnt/isoにマウントします。

```
# mount -t iso9660 -o loop /data/isoImages/rhel-8.4-x86_64-dvd.iso /mnt/iso
```

- /etc/yum.repos.d pathにREPOファイルを作成します。

```
# cat /etc/yum.repos.d/offlinedvd.repo

[RHEL8.4_Appstream]
name=Red Hat Enterprise Linux 8.4.0 Appstream
mediaid=None
metadata_expire=-1
gpgcheck=0
enabled=1
baseurl=file:///data/offlineRepos/RHEL8.4/AppStream/

[RHEL8.4_BaseOS]
```

```
name=Red Hat Enterprise Linux 8.4.0 BaseOS
mediaid=None
metadata_expire=-1
gpgcheck=0
enabled=1
baseurl=file:///data/offlineRepos/RHEL8.4/BaseOS/
```

- リポリストを確認し、AppstreamおよびBaseos REPOがマッピングされていることを確認します。

```
# dnf repolist
```

```
[root@MUMBKVMC01 /]#
[root@MUMBKVMC01 /]#
[root@MUMBKVMC01 /]# dnf repolist
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.

repo id                                repo name
RHEL8.4_Appstream                       Red Hat Enterprise Linux 8.4.0 Appstream
RHEL8.4_BaseOS                           Red Hat Enterprise Linux 8.4.0 BaseOS
[root@MUMBKVMC01 /]#
[root@MUMBKVMC01 /]#
[root@MUMBKVMC01 /]#
```

## リポジトリVMの作成

```
$ cd /var/lib/libvirt/images/
$ export LIBGUESTFS_BACKEND=direct
$ virt-customize -a /var/lib/libvirt/images/rhel-8.4-x86_64-kvm.qcow2 --root-password password:Cisco@123
```

```
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]# virt-customize -a /var/lib/libvirt/images/rhel-8.4-x86_64-kvm.qcow2 --root-password password:Cisco@123
[ 0.0] Examining the guest ...
[ 6.6] Setting a random seed
[ 6.7] Setting the machine ID in /etc/machine-id
[ 6.7] Setting passwords
[ 7.7] Finishing off
[root@MUMBKVMC01 images]#
```

```
$ virt-filesystems --long -h --all -a /var/lib/libvirt/images/rhel-8.4-x86_64-kvm.qcow2
```

```
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]# virt-filesystems --long -h --all -a /var/lib/libvirt/images/rhel-8.4-x86_64-kvm.qcow2
Name      Type      VFS      Label MBR  Size Parent
/dev/sda1 filesystem unknown  -    -    1.0M -
/dev/sda2 filesystem vfat    -    -    100M -
/dev/sda3 filesystem xfs     root  -    9.9G -
/dev/sda1 partition -        -    -    1.0M /dev/sda
/dev/sda2 partition -        -    -    100M /dev/sda
/dev/sda3 partition -        -    -    9.9G /dev/sda
/dev/sda  device   -        -    -    10G  -
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]#
```

- REPOサーバのイメージを作成します。

```
$ qemu-img create -f qcow2 /var/lib/libvirt/images/rhel_84_osprepo.qcow2 500G
```

```
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]# qemu-img create -f qcow2 /var/lib/libvirt/images/rhel_84_osprepo.qcow2 300G
Formatting '/var/lib/libvirt/images/rhel_84_osprepo.qcow2', fmt=qcow2 size=322122547200 cluster_size=65536 lazy_refcounts=off refcount_bits=16
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]#
```

```
$ virt-resize --expand /dev/sda3 /var/lib/libvirt/images/rhel-8.4-x86_64-kvm.qcow2 /var/lib/libvirt/ima
```

```
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]# qemu-img create -f qcow2 /var/lib/libvirt/images/rhel_84_osprepo.qcow2 300G
Formatting '/var/lib/libvirt/images/rhel_84_osprepo.qcow2', fmt=qcow2 size=322122547200 cluster_size=65536 lazy_refcounts=off refcount_bits=16
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]# virt-resize --expand /dev/sda3 /var/lib/libvirt/images/rhel-8.4-x86_64-kvm.qcow2 /var/lib/libvirt/images/rhel_84_osprepo.qcow2
[  0.0] Examining /var/lib/libvirt/images/rhel-8.4-x86_64-kvm.qcow2
*****
Summary of changes:
/dev/sda1: This partition will be left alone.
/dev/sda2: This partition will be left alone.
/dev/sda3: This partition will be resized from 9,96 to 299,96. The
filesystem xfs on /dev/sda3 will be expanded using the 'xfs_growfs'
method.
*****
[  2.0] Setting up initial partition table on /var/lib/libvirt/images/rhel_84_osprepo.qcow2
[ 12.7] Copying /dev/sda1
[ 12.7] Copying /dev/sda2
[ 12.8] Copying /dev/sda3
100% |
[ 20.8] Expanding /dev/sda3 using the 'xfs_growfs' method
00:00
Resize operation completed with no errors. Before deleting the old disk,
carefully check that the resized disk boots and works correctly.
[root@MUMBKVMC01 images]#
```

```
$ qemu-img create -f qcow2 -b /var/lib/libvirt/images/rhel_84_osprepo.qcow2 -F qcow2 /data/images/OSPRED
```

```
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]# qemu-img create -f qcow2 -b /var/lib/libvirt/images/rhel_84_osprepo.qcow2 -F qcow2 /data/images/OSPRED_RHEL_84.qcow2
Formatting '/data/images/OSPRED_RHEL_84.qcow2', fmt=qcow2 size=322122547200 backing_file=/var/lib/libvirt/images/rhel_84_osprepo.qcow2 backing_fmt=qcow2 cluster_size=65536 lazy_re
fcounts=off refcount_bits=16
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]#
```

```
$ guestfish -a /data/images/OSPRED_RHEL_84.qcow2 -i ln-sf /dev/null /etc/systemd/system/cloud-init.service
```

```
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]# guestfish -a /data/images/OSPRED_RHEL_84.qcow2 -i ln-sf /dev/null /etc/systemd/system/cloud-init.service
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]#
```

```
$ osinfo-query os | grep rhel8
```

```
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]# osinfo-query os | grep rhel8

** (osinfo-query:49279): WARNING **: 17:12:33.856: Entity http://pcisig.com/pci/1836/0100 referenced but not defined
rhel8-unknown | Red Hat Enterprise Linux 8 Unknown | 8-unknown | http://redhat.com/rhel/8-unknown
rhel8.0 | Red Hat Enterprise Linux 8.0 | 8.0 | http://redhat.com/rhel/8.0
rhel8.1 | Red Hat Enterprise Linux 8.1 | 8.1 | http://redhat.com/rhel/8.1
rhel8.2 | Red Hat Enterprise Linux 8.2 | 8.2 | http://redhat.com/rhel/8.2
rhel8.3 | Red Hat Enterprise Linux 8.3 | 8.3 | http://redhat.com/rhel/8.3
rhel8.4 | Red Hat Enterprise Linux 8.4 | 8.4 | http://redhat.com/rhel/8.4
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]#
```

\$ virt-install --cpu host --memory 32768 --vcpus 16 --os-variant rhel8.4 --disk path=/data/images/OSPRE

```
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]# virt-install --cpu host --memory 32768 --vcpus 16 --os-variant rhel8.4 --disk path=/data/images/OSPREPO_RHEL_84.qcow2,device=disk,bus=virtio,format=qcow2
--import --noautoconsole --vnc --network bridge:br-ext --name OSPREPO_RHEL_84

** (process:49296): WARNING **: 17:13:15.813: Entity http://pcisig.com/pci/1836/0100 referenced but not defined

Starting install...
Domain creation completed.
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]#
```

\$ virsh list --all

```
root@MUMBKVMC01 images]#
root@MUMBKVMC01 images]# virsh list --all
Id Name State
-----
1 OSPREPO_RHEL_84 running

root@MUMBKVMC01 images]#
root@MUMBKVMC01 images]#
root@MUMBKVMC01 images]# virsh list --all
Id Name State
-----
1 OSPREPO_RHEL_84 running

root@MUMBKVMC01 images]#
```

## ディレクタVMの作成

- ディレクタサーバのイメージを作成します。

# qemu-img create -f qcow2 /var/lib/libvirt/images/rhel\_84\_ospdirector.qcow2 500G

```
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]# qemu-img create -f qcow2 /var/lib/libvirt/images/rhel_84_ospdirector.qcow2 200G
Formatting '/var/lib/libvirt/images/rhel_84_ospdirector.qcow2': fmt=qcow2 size=214748364800 cluster_size=65536 lazy_refcounts=off refcount_bits=16
[root@MUMBKVMC01 images]#
```

```
# virt-resize --expand /dev/sda3 /var/lib/libvirt/images/rhel-8.4-x86_64-kvm.qcow2 /var/lib/libvirt/ima
```

```
root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]# virt-resize --expand /dev/sda3 /var/lib/libvirt/images/rhel-8.4-x86_64-kvm.qcow2 /var/lib/libvirt/images/rhel_84_ospdirector.qcow2
[  0.0] Examining /var/lib/libvirt/images/rhel-8.4-x86_64-kvm.qcow2
*****
Summary of changes:
/dev/sda1: This partition will be left alone.
/dev/sda2: This partition will be left alone.
/dev/sda3: This partition will be resized from 9.9G to 199.9G. The
filesystem xfs on /dev/sda3 will be expanded using the 'xfs_growfs'
method.
*****
[  2.0] Setting up initial partition table on /var/lib/libvirt/images/rhel_84_ospdirector.qcow2
[ 12.7] Copying /dev/sda1
[ 12.7] Copying /dev/sda2
[ 12.7] Copying /dev/sda3
100% |
[ 20.6] Expanding /dev/sda3 using the 'xfs_growfs' method
Resize operation completed with no errors. Before deleting the old disk,
carefully check that the resized disk boots and works correctly.
[root@MUMBKVMC01 images]#
```

```
# qemu-img create -f qcow2 -b /var/lib/libvirt/images/rhel_84_ospdirector.qcow2 -F qcow2 /data/images/O
```

```
root@MUMBKVMC01 images]#
root@MUMBKVMC01 images]# qemu-img create -f qcow2 -b /var/lib/libvirt/images/rhel_84_ospdirector.qcow2 -F qcow2 /data/images/OSPDIRECTOR_RHEL_84.qcow2
qformatting /data/images/OSPDIRECTOR_RHEL_84.qcow2, fmt=qcow2 size=214748364800 backing_file=/var/lib/libvirt/images/rhel_84_ospdirector.qcow2 backing_fmt=qcow2 cluster_size=65536
lazy_refcounts=off refcount_bits=16
root@MUMBKVMC01 images]#
root@MUMBKVMC01 images]#
```

```
# guestfish -a /data/images/OSPDIRECTOR_RHEL_84.qcow2 -i ln-sf /dev/null /etc/systemd/system/cloud-init
# virt-install --cpu host --memory 131072 --vcpus 32 --os-variant rhel8.4 --disk path=/data/images/OSPD
```

```
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]# virt-install --cpu host --memory 131072 --vcpus 32 --os-variant rhel8.4 --disk path=/data/images/OSPDIRECTOR_RHEL_84.qcow2,device=disk,bus=virtio,format=q
cow2 --import --noautoconsole --vnc --network bridge:br-prov --network bridge:br-ext --name OSPDIRECTOR_RHEL_84
** (process:49762): WARNING **: 17:15:52.006: Entity http://pcr.sig.com/pci/1836/0100 referenced but not defined
Starting install...
Domain creation completed.
[root@MUMBKVMC01 images]#
```

```
# virsh list --all
```

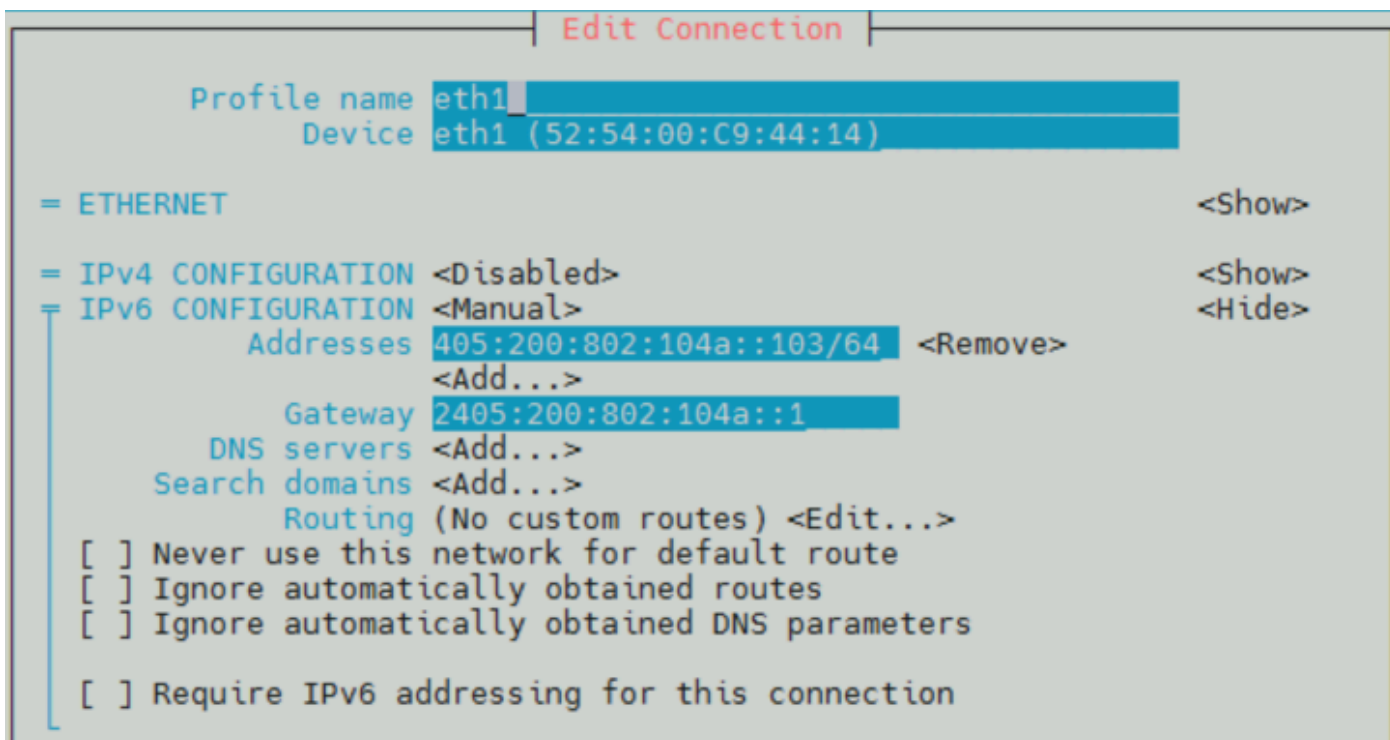
```
Domain creation completed.
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]# virsh list --all
 Id   Name                               State
-----
  1   OSPREPO_RHEL_84                    running
  2   OSPDIRECTOR_RHEL_84                running
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]#
```

```
[root@NAGVMK02 images]# virt-install --name OSPDIRECTORCL02_RHEL --description "director" --os-variant rhel8.4 --disk path=/data/images/OSPDIRECTORCL02_RHEL_84.qcow2,size=500,device=disk,bus=virtio,format=qcow2 --cpu host --memory 131072 --vcpus 32 --location /data/isoImages/rhel-8.4-x86_64-dvd.iso --network bridge:br-ext --network bridge:br-prov --extra-args console=ttyS0 --boot uefi
** (process:193050): WARNING **: 15:12:45.915: Entity http://pcisig.com/pci/1B36/0100 referenced but not defined
Starting install...
Retrieving file vmlinuz...
Retrieving file initrd.img...
Allocating 'OSPDIRECTORCL02_RHEL_84.qcow2'
(virt-viewer:193156): GLib-GIO-CRITICAL **: 15:12:48.336: g_dbus_proxy_new_sync: assertion 'G_IS_DBUS_CONNECTION (connection)' failed
(virt-viewer:193156): GSpice-WARNING **: 15:12:49.080: PulseAudio context failed Connection refused
(virt-viewer:193156): GSpice-WARNING **: 15:12:49.080: pa_context_connect() failed: Connection refused
(virt-viewer:193156): GSpice-WARNING **: 15:12:49.213: Could not create org.gnome.SessionManager dbus proxy: Could not connect: Connection refused
(virt-viewer:193156): GSpice-WARNING **: 15:12:49.213: Warning no automount-inhibiting implementation available
(virt-viewer:193156): GLib-GObject-WARNING **: 15:13:08.438: value "64" of type "gint" is invalid or out of range for property 'desktop-width' of type 'gint'
(virt-viewer:193156): GLib-GObject-WARNING **: 15:13:08.438: value "64" of type "gint" is invalid or out of range for property 'desktop-height' of type 'gint'
```

- 新しいウィンドウを開き、VMのコンソールログインを実行します。

```
# virsh list --all
# virsh console <domain-id>
```

- Post Director VMの導入では、2つのethインターフェイスeth0とeth1が作成されます。
  - eth0はOSP\_provisioningネットワークに使用され、このネットワークではbrctlplaneネットワークがUndercloud導入時に作成されます。
  - eth1は、SSHアクセスの外部接続に使用されます。そのため、OSP\_externalサブネットのIPアドレスが割り当てられます。



## ローカル・オフラインREPO作成

REPOサーバはRedhat CDNに登録されている必要があり、導入に必要なRHOSP 16.2のすべての使用可能なパッケージのリポジトリが必要です。RHEL RPMパッケージおよびRHOSPコンテナイメージは、プロキシを使用してREPO VMにダウンロードする必要があります。

## RHOSPクラウドの導入

RHOSP 16.2は、自動化によってお客様のネットワークに導入されます。Ansibleスクリプトは、UndercloudおよびOvercloudの導入を自動化するために使用されます。

### 前提条件

実際のクラウド導入を開始する前に従うべき手順：

1. KVMノードとOSP\_EXTネットワーク上のローカルREPO VMおよびDirector VMとの接続を確認する。
2. 指定されたフォルダのKVMホストにアップロードされたすべての実行可能なスクリプトと自動化スクリプトを確認します。
3. 「cisco」および「automation」という名前のフォルダを作成し、ansibleスクリプトのtarballを配置します。

```
# cd /home
# mkdir cisco
# cd /home/cisco
# mkdir automation
# cd /home/cisco/automation
```

tarballは、次の3つのフォルダのディレクトリ構造で構成されます。

- スクリプト：ベアメタルノードの設定に使用されるシェルスクリプトで構成されます。
- rpms: Intel Ethernet Connection(ICE)ドライバ、Intelドライバ、およびRed-hat Package Manager(RPM)パッケージで構成されます。
- ansible：入力varsファイル、展開用のansible yamlファイル、およびjinja-templateで構成されます。

4. sshpassパッケージをインストールします。sshpassは、sshにパスワードを非対話的に提供するために使用されるコマンドラインユーティリティです。主に、手動によるパスワード入力が不可能なスクリプトまたは自動化のシナリオで使用されます。

- インターネットまたは既存のサーバからsshpassパッケージsshpass.tar.gzをダウンロード

します。

- GNUコンパイラコレクション(GCC)パッケージをインストールします。

```
# yum install gcc
```

- makeパッケージをインストールします。

```
# yum install make
```

- sshpassパッケージを解凍し、SSH passをインストールします。

```
# tar -xvzf sshpass.tar.gz
# cd sshpass-1.10/
# ./configure
# sudo make install
# sshpass -V
```

5. ディレクタのインストール・プロセスでは、root以外のユーザーがコマンドを実行する必要があります。'Stack'ユーザは、sudoアクセスを使用してDirector VMに作成する必要があります。

```
# useradd stack
# passwd stack
```

```
Disable password requirements for the 'stack' user when using sudo.
# echo "stack ALL=(root) NOPASSWD:ALL" | tee -a /etc/sudoers.d/stack
# chmod 0440 /etc/sudoers.d/stack
```

6. REPOサーバからディレクタVMおよびKVMにrootCA.crtファイルを所定のパスでコピーします。また、信頼リストのREPO VM証明書を更新します。

```
# /etc/pki/ca-trust/source/anchors
# update ca-trust
```

7. /etc/hostsファイル内のDirector VMおよびKVMのローカルREPOサーバホスト名の詳細を更新します。

8. KVMおよびDirector VMで、Pythonやansibleなどの追加パッケージをインストールして、ansible自動化スクリプトを実行します。

```
# dnf install python3 python3-devel ansible httpd -y
# update-alternatives --set python /usr/bin/python3
```

9. クラウド導入時にプロビジョニングを有効にするには、DirectorのプロビジョニングネットワークからCIMCサブネットに到達できる必要があります。必要に応じて、同じスタティックルートを追加します。

```
# ip -6 route add <CIMC Subnet> via <Provisioning Subnet>
```

10. KVMおよびDirector VMで、/ansibleフォルダにホストファイルを作成し、必要に応じてスタック固有の詳細を追加します。

```
[ospd]
```

```
# <PODNAME> ansible_host=<OSPD IP> ansible_ssh_user=stack ansible_ssh_pass='<STACKPASSWD>' ansible_ssh_
```

<podname> - Stack Name of the Cloud.

<OSPD IP> - Baremetal OSPD Node IP Address

<STACKPASSWD> - OSPD Node password for 'stack' user

11. 使用可能なすべてのプレイブックと入力ファイルがディレクタVMの/home/stackフォルダに保存されていることを確認します。

## 入力ファイルの更新

顧客のネットワーク固有の詳細で構成される入力変数ファイルがあり、クラウドの導入に備えて準備する必要があります。

パス : /home/cisco/automation/ansible/podvars

ファイル名 : <スタック名>\_vars.yml

サイト固有のIP計画/詳細設計文書に従って、強調表示されているパラメータを更新します。



---

注：ダミーIPアドレスは、表現の目的でのみ使用されます。

---

<#root>

```
# #####  
# XR21 Specific Variables  
# #####
```

```
# =====  
# Common Variables  
# =====
```

```
# UCS hardware type: 'm4/m5/m6'  
hardware: m6
```

```
# Platform type: 'epc/pcrf'  
platform: epc
```

```
# RHEL version  
rhel: { version: 84, tag: 8.4 }
```

```
# Openstack version  
osp: { version: 16, major: 2 }
```

```
# Container version  
container: { tag: 16.2, tools: 3.0 }
```

```
# Overcloud stack name
```

```
stack_name: '
```

```
,
```

```
# OSPD full hostname
```

```
fqdn_hostname: '
```

```
.epdg.ap.hamb.a6.cloud.com'
```

```
# OSPD host login
ospd_host: { ip:

'2405:XXXX:089:1054::11'

, username: 'stack', password: '*****' }

# OSPD cimc login
ospd_cimc: { ip:

'2405:XXXX:089:1054::11'

, username: 'admin', password: '*****' }

# CIMC username and password must be same across all Overcloud nodes
cimc: { username: 'admin', password: '*****', ip_pool: '2405:XXXX:089:1055::/64' }

# Undercloud-Overcloud provision
internal_network: {
  ip_type: 'v6',
  local_interface: 'eth0',

  local_ip: `2405:XXXX:089:1041::103`,

  undercloud_public_host: `2405:XXXX:089:1041::105`,

  undercloud_admin_host: `2405:XXXX:089:1041::104`,

  cidr: `2405:XXXX:089:1041::/64`,

  dhcp_start: `2405:XXXX:089:1041::200`,

  dhcp_end: `2405:XXXX:089:1041::299`,

  gateway: `2405:XXXX:089:1041::199`,

  # nexthop: `2405:XXXX:089:1041::1`,
```

```
inspection_iprange_start: '2405:XXXX:089:1041::300',
```

```
inspection_iprange_end: '2405:XXXX:089:1041::399',
```

```
}
```

```
# DNS
```

```
dns_ips: [ '2405:YYYY:a10:f100::1' ]
```

```
dns_search_domains: [ 'cloud.com' ]
```

```
# NTP
```

```
ntp_ips: [ '2405:YYYY:801:700::afa', '2405:YYYY:801:700::afb' ]
```

```
# Deployment type: 'offline/online'
```

```
repos: { rhel: 'offline', container: 'offline' }
```

```
# Offline details if repos is 'offline'
```

```
offline: {
```

```
    environment: 'v01_00',
```

```
    deliverymedia: '/home/stack/deliverymedia/'
```

```
}
```

```
# Satellite details if repos is 'online'
```

```
satellite: {
```

```
    fqdn_name: 'rh-satellite2.mitg-bxb300.cisco.com',
```

```
    ip: '10.XX.XX.XX',
```

```
    org: 'MITG',
```

```
    user: 'admin',
```

```
    password: '*****',
```

```
    environment: 'production',
```

```
    activation_key: 'ak-rhel{{rhel.version}}-osp{{osp.version}}{{osp.major}}',
```

```
    repos_file: 'rhel{{rhel.version}}osp{{osp.version}}{{osp.major}}.yaml'
```

```
}
```

```
# Offline container registry details
```

```
offline_registry: {
```

```
    ip: '2405:XXXX:089:1055::100',
```

```
    name: '
```

```
.
```

```
',
```

```
    port: '5000',
    container_tag: '16.2.6',
    user: 'ciscoadmin',
    password: '*****'
}
```

```
# Custom cloud domain details
domain_name: {
    domain: '
```

```
,
```

```
    cloudshortname: 'n1'
}
```

```
# Container images namespace
```

```
container_namespace: 'mitg-{{satellite.environment}}-cv-rhel{{rhel.version}}-osp{{osp.version}}{{osp.ma
```

```
# List of cimc IPs
```

```
ctrl_cimc_ip:
```

```
- 2405:xxxx:yyyy:1036::12
```

```
- 2405:xxxx:yyyy:1036::13
```

```
- 2405:xxxx:yyyy:1036::14
```

```
osdc_cimc_ip:
```

```
cmpt_cimc_ip:
```

- 2405:XXXX:YYYY:1037::17

- 2405:XXXX:YYYY:1038::18

- 2405:XXXX:YYYY:1038::19

- 2405:XXXX:YYYY:1038::20

mgmt\_cimc\_ip:

- 2405:XXXX:YYYY:1051::15

- 2405:XXXX:YYYY:1051::16

# =====  
# Hardware Specific Variables  
# =====

# Isolcpu for cpu pinning  
isolcpus: { osdc: '4-31,36-63', cmpt: '2-31,34-63', mgmt: '2-31,34-63' }

# Hugepages in 1G Pages  
hugepages: { osdc: 428, cmpt: 448, mgmt: 448 }

# Reserved host memory in MB  
reserved\_host\_memory: { osdc: 84000, cmpt: 64000, mgmt: 64000 }

# Number of VFs per SR-IOV port  
sriov\_vfs\_per\_port: 16

# List of SR-IOV ports  
sriov\_port\_list: [ens1f0, ens1f1, ens9f0, ens9f1]

# List of OVS bonding interface  
ovs\_bond\_interface: [eno5, eno6]

# Physical networks  
physical\_network: [phys\_pcie1\_0, phys\_pcie1\_1, phys\_pcie2\_0, phys\_pcie2\_1]

# Boot disk size  
boot\_disk\_mb\_size: { ctrl: 761985, osdc: 761985, cmpt: 761985, mgmt: 1524925 }

```
# Boot disk PD slot number
boot_disk_pd_slot: { ctrl: [1,2], osdc: [1,2], cmpt: [1,2], mgmt: [1,2] }

# Boot disk VD slot number
boot_disk_vd_slot: { ctrl: 237, osdc: 235, cmpt: 239, mgmt: 239 }

# Storage backend 'swift' or 'ceph'
storage_backend: 'swift'

# Storage disk size
storage_disk_mb_size: { swift: 761985, ceph: 914573, journal: 0 }

# Storage disk PD slot number
storage_disk_pd_slot: { swift: [6,7], ceph: [3,4,5,6], journal: [0] }

# Storage disk VD slot number
storage_disk_vd_slot: { swift: [238,239], ceph: [236,237,238,239], journal: [0] }

# Firmware version 'yes' or 'no' ???
firmware: { check: 'no', bios_version: '4.2.3c', cimc_version: '4.2(3e)' }

# =====
# OSP Specific Variables
# =====

# Timezone for overcloud nodes

timezone: 'Asia/Kolkata'

# Overcloud node count to deploy

node_count: { ctrl: 3, osdc: 0, cmpt: 11, mgmt: 2 }

local_network: {
  ip_type: 'v6',
  tenant_vlan_id: 1045,
  tenant_net_cidr: '240f:ppp:rr:1045::/64',
  tenant_alloc_pools_start: '240f:ppp:rr:1045::10',
  tenant_alloc_pools_end: '240f:ppp:rr:1045:ffff:ffff:ffff:fffe',

  storage_vlan_id: 1043,
  storage_net_cidr: '240f:ppp:rr:1043::/64',
  storage_alloc_pools_start: '240f:ppp:rr:1043::10',
  storage_alloc_pools_end: '240f:ppp:rr:1043:ffff:ffff:ffff:fffe',

  storage_mgmt_vlan_id: 1044,
  storage_mgmt_net_cidr: '240f:ppp:rr:1044::/64',
  storage_mgmt_alloc_pools_start: '240f:ppp:rr:1044::10',
```

```
storage_mgmt_alloc_pools_end: '240f:ppp:rr:1044:ffff:ffff:ffff:fffe',

internal_api_vlan_id: 1042,
internal_api_net_cidr: '240f:ppp:rr:1042::/64',
internal_api_alloc_pools_start: '240f:ppp:rr:1042::10',
internal_api_alloc_pools_end: '240f:ppp:rr:1042:ffff:ffff:ffff:fffe'
}
```

```
# External VLAN and IP configs
```

```
external_network: {
```

```
    ip_type: 'v6',
```

```
    vlan_id: 1046,
```

```
    default_route: '2405:XXXX:YYYY:1055::1',
```

```
    network_cidr: '2405:XXXX:YYYY:1055::/64',
```

```
    alloc_pool_start: '2405:XXXX:YYYY:1055::100',
```

```
    alloc_pool_end: '2405:XXXX:YYYY:1055::200',
```

```
    horizon_ip: '2405:XXXX:YYYY:1055::107'
```

```
}
```

```
# Neutron mechanism driver 'ovs' or 'ovn'
```

```
neutron: {
```

```
    driver: 'ovs',
```

```
    dvr: false,
```

```
    datacenter_vlan_start: 1050,
```

```
    datacenter_vlan_end: 1070
```

```
}
```

```
# =====
```

```
# OS Specific Variables
```

```
# =====
```

```
# RHEL kernel version
```

```
kernelversion: '4.18.0-305.88.1.el8_4.x86_64'
```

```
# E810 ICE driver
ice_driver: { check: 'yes', version: 1.12.6, intel_aux_version: 1.0.1 }

# ENIC and FNIC version
nic_version: { enic: '2.3.0.53', fnic: '1.6.0.53' }

# IPMI watchdog timer config
watchdog: { action: enabled, version: '2.0.31-3.el8.x86_64', timer: 250 }

# StorCLI raid management
storcliver: '007.2612.0000.0000-1.noarch'

# =====
# Platform Specific Variables
# =====

# Buffer pool size based on platform type
innodb_buffer_pool_size: 1610
# #####
# END - XR21 Specific Variables
# #####
```

## アンダークラウドの導入

Undercloudは、7つの手順で実行可能なスクリプトを使用して導入されます。この場合、すべての手順は、ジャンプホストとして機能するKVMホストから実行する必要があります。

手順	TAG	説明	プレイブックYAML
ステップ 1:	チェックファイル	Openstack Platform Director(OSPD)に必要なプレイブック、スクリプト、およびRPMを確認します。	osp16_published_playbooks_verify.yml (公開済みプレイブック検証.yml)
ステップ 2	下顎骨	common_vars.yml (ハードウェア、ソフトウェア、ネットワークの詳細)、 hw_m6_vars.yml (CPU、メモリ、Hugepages、ディスク、NICなど)、 rhel_84_vars.yml (RHEL、カーネル、ICEドライバ、NICバージョン)、 pf_esc_vars.yml(Elastic Services Controller(ESC)の詳細	osp16_generate_pod_specific_vars.yml (特定の変数を生成)

		)、osp_16など、ハードウェア、POD固有の変数ファイルを生成します。 vars.yml ( OSPバージョン、タイムゾーン、IPタイプ、VLAN ID、IP、neutron詳細 )	
ステップ 3	preucdeploy	完全修飾ドメイン名(FQDN)、ネットワークタイムプロトコル(NTP)を設定し、ディレクタノード上のすべてのパッケージを更新します。	osp16_pre_undercloud_deploy.yml ( クラウド導入前に導入 )
ステップ 4	初起動	以前の設定とパッケージのインストール後に、Undercloudディレクタノードの最初のレポートを実行します。	osp16_アンダークラウド_導入.yml
ステップ 5	ucdeploy	ディレクタへのUndercloudスタックのインストール	osp16_アンダークラウド_チューニング.yml
ステップ 6	状態	ディレクタノードでBIOS CPU Cステータを設定します。	osp16_cstate.yml
ステップ 7	セカンダリレポート	BIOS変更後、Undercloud Directorで2回目のレポートを行います。	N/A

osp16\_auto\_undercloud\_deploy.ymlという名前のファイルは単一のイテレーションで実行できる主要な実行可能プレイブックですが、展開に問題が発生した場合のトラブルシューティングを容易にするために、異なるタグを使用してプレイブックを段階的に実行することをお勧めします。

```
<#root>
```

```
#
```

```
cd /home/stack/ansible/
```

```
# ansible-playbook -i hosts osp16_auto_undercloud_deploy.yml -e podname=<> --tags=
```

TAG

For Ex -

#

```
ansible-playbook -i hosts osp16_auto_undercloud_deploy.yml -e podname=<> --tags=checkfiles
# ansible-playbook -i hosts osp16_auto_undercloud_deploy.yml -e podname=<> --tags=genpodvars
# ansible-playbook -i hosts osp16_auto_undercloud_deploy.yml -e podname=<> --tags=preucdeploy
# ansible-playbook -i hosts osp16_auto_undercloud_deploy.yml -e podname=<> --tags=firstreboot
# ansible-playbook -i hosts osp16_auto_undercloud_deploy.yml -e podname=<> --tags=ucdeploy
# ansible-playbook -i hosts osp16_auto_undercloud_deploy.yml -e podname=<> --tags=cstate
# ansible-playbook -i hosts osp16_auto_undercloud_deploy.yml -e podname=<> --tags=secondreboot
```

Note :-

Deployment Logs would be generated in “/home/stack/autologs” in Director-VM.

Post-Checks for Verification of Undercloud Deployment.

“stackrc” & “undercloud.conf” file must be generated in /home/stack folder.

```
# sudo podman ps -a
# source stackrc
# openstack stack list
# openstack stack show <stack-name> --fit
# openstack server list
# openstack network list
# openstack subnet list
```

## クラウドの導入

HAモードのコントローラが3つ以上、コンピューティングが1つ以上の状態でオーバークラウドが導入されます。Overcloudは、ansibleスクリプトを使用して17ステップで導入されます。この場合、すべてのステップは、Jump-hostとして機能するDirector-VMから実行する必要があります。

手順	TAG	説明	プレイブックYAML
ステップ 1:	下顎骨	common_vars.yml ( ハードウェア、ソフトウェア、ネットワークの詳細 )、 hw_m6_vars.yml ( CPU、メモリ、hugepages、ディスク、	osp16_generate_pod_specific_vars.yml ( 特定の変数を生成 )

		<p>NICなど)、  rhel_84_vars.yml ( RHEL、カーネル、ICEドライバ、NICバージョン)、  pf_esc_vars.yml ( ESCの詳細)、osp_16_vars.ymlなど、ハードウェア、RHELに関連するPOD固有の変数ファイルを生成します。ML ( OSPバージョン、タイムゾーン、IPタイプ、VLAN ID、IP、neutron詳細 )</p>	
ステップ 2	genystack ( 必須 )	<p>前の手順で作成した /var/common_vars.ymlから Instackenv JSONファイルを生成します。</p> <p>ダイレクタには、手動で作成されたノード定義テンプレートが必要です。このファイル instackenv.jsonはJSON形式を使用し、ノードのすべてのハードウェアと電源管理の詳細が含まれています。この手順では、ファイルを生成する前に、UCSサーバのハードウェア構成も検証します。</p>	osp16_generate_instackenv.yml ( 生成 )
ステップ 3	CIMCVD	<p>common_vars.yml、hw_m6_vars.yaml、および rhel_84_vars.ymlを参照して、各サーバでCIMC設定と仮想ディスク(VD)を設定します。</p>	osp16_cimc_vd_configure.yml
ステップ 4	導入前	<p>この手順では、Overcloudを導入するための前提条件をすべて実行します。FQDN、NTPを設定し、すべてのパッケージを更新し、イメージをパスにプッシュして展開します。</p>	osp16_pre_overcloud_deploy.yml
手順 5	インポートノード	<p>このステップでは、サーバのCPU、メモリ、NIC、インター</p>	osp16_import_ironic_nodes.yml

		フェイス、およびネットワークスイッチ上のポートをイントロスペクトします。イントロスペクションは、接続されているネットワークスイッチですべてのコントローラとコンピュータに対して実行されます。	
ステップ 6	ジェンテンプレート	コントローラおよび計算用のカスタムテンプレートファイルを生成します。カスタムテンプレートで、コントローラで実行されているすべてのサービスのコントローラおよびコンピューティングロールを定義します。また、証明書やルートなどを適用してシステムを強化します。	osp16_generate_custom_templates.yml ( カスタムテンプレートの生成 )
ステップ 7	ocdeploy	この手順では、Openstack Overcloudの導入を行います。Red Hatが提供する、RHOSP導入用のdeploy.shを実行します。	osp16_overcloud_deploy.yml ( オフラインでのクラウド導入 )
手順 8	geninventoryの	このステップでは、ansibleで使用するinventory ymlファイルを生成します。このファイルには、プロビジョニングIP、IPMI(CIMC)IP、およびクレデンシャルが保存され、コントローラでマッピングされて、システムにログインして次のステップを実行するための自動計算用に計算されます。	osp16_build_inventory_v3.py ( ビルド_インベントリ_v3.py )
ステップ 9	オフライン報告	ファイル /etc/yum.repo.d/offline.repo内のOvercloud Offline REPOが、外部ネットワーク経由でサーバをポイントするように設定します。	osp16_config_offline_repo.yml ( オフライン再オープン )
ステ	フェンシング	Shoot The Other Node In the	osp16_config_フェンシング.yml

ステップ 10		Head ( HAクラスタのフェンシングテクニック ) (STONITH)を使用して、すべてのコントローラノードでフェンシングを設定します。	
ステップ 11	raidcache	すべてのコントローラのRAIDキャッシュ設定を構成し、SWIFTストレージ設定も構成します。	osp16_raid_cache_tuning.yml
ステップ 12	dnfupdate	すべてのノードのすべてのパッケージに対してDNF更新を実行します。	dnf_update_all_packages.yml
ステップ 13	setiplink	この手順では、Evolved Packet Data Gateway(EPDG)の内部トラフィックとデータトラフィックに対して、SR-IOVポートの信頼モード制御を有効にします。SR-IOVポートのサポートはneutronで使用でき、VMはSR-IOV仮想機能を介してネットワークにアクセスできます。	osp16_セットIpLink.yml
手順 14 :	ウォッチドッグ	このステップでは、ディレクタノードのIPMI設定を、帯域外接続を介した全サーバの管理タスクに対して設定します。	osp16_config_ipmi_watchdog.yml
手順 15 :	川	Intel NICポートをSR-IOVとして使用するには、Peripheral Component Interconnect(PCI)カード用のIntel E810 ICEドライバをEPDG用のバージョン1.12.6にアップデートします。	osp16_ice_driver_install.yml
ステップ 16 :	reboot	前述の手順を実行した後で、すべてのOvercloudノードをリブートします。	osp16_reboot_overcloud_hosts.yml ( 全バージョン )

ステップ 17:	ベリフィス	RHOSP導入の設定と状態を確認します。	osp16_rhosp_verify.yml
-------------	-------	----------------------	------------------------

Overcloudノードのプロビジョニングでは、Undercloudは「overcloud-hardened-uefi-full.qcow2」を使用します。そのため、Overcloudの導入を開始する前に、イメージをアンダークラウド/ディレクトリ内の指定されたパスに保存する必要があります。

リモートサイトからOvercloud qcow2ファイルをコピーします。

<#root>

```
# su - stack
# cd /home/stack
# mkdir deliverymedia
# cd deliverymedia
```

### Copy overcloud-hardened-uefi-full.qcow2 to deliverymedia ###

```
# scp overcloud-hardened-uefi-full.qcow2 stack@[Director-IP]:/home/stack/deliverymedia
[stack@[stack@ Undercloud ~]$ cd /home/stack/ansible/
[stack@[stack@ Undercloud ansible]$ ansible-playbook osp16_auto_overcloud_deploy.yml -e podname=POD_NAME]
```

For Ex -

```
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=genpodvars
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=geninstack
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=cimcvd
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=preocdeploy
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=importnodes
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=gentemplates
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=ocdeploy
```

#### Push & Update the rootCA.pem in all the Controllers & Computes ####

```
# for node in $(nova list | grep -i active | awk '{print $12}' | awk -F "=" '{print $2}'); do scp -o S...
```

#### Append the Director Entry in "/etc/hosts" file ####

```
# for node in $(nova list | grep -i active | awk '{print $12}' | awk -F "=" '{print $2}'); do ssh -o S...
```

```
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=geninventory
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=offlinerepo
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=fencing
```

### In case of Fencing Failures, please check the reachability of CIMC Subnet from Controllers #####

## If CIMC Subnet is not pinging, Do add the static Route ##

```
# ip -6 route add <CIMC Subnet> via <Provisioning Subnet>
Ex: ip -6 route add 2405:XXXX:YYY:9999::/64 via 2405:XXXX:YYY:9999:1
```

```
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=raidcache
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=dnfupdate
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=setiplink
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=watchdog
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=icedriver
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=reboot
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=verifyrhaps
```

配置ログを監視するには、最新のログファイルを使用します。

```
# tail -F </home/stack/autologs/osp16_auto_overcloud_deploy_*.log>
```

すべての17ステップを通過したことを確認します。

失敗したチェック=>カウントは00である必要があります。

ログ=> /home/stack/autologs/osp16\_rhosp\_verify.yml\_20200703T042257.log

#####

#ステップ|タグ|説明|プレイブック

#####

```
#ステップ1 | genpodvars | POD固有変数ファイルの生成| osp16_generate_pod_specific_vars.yml
-e podname=
```

```
#ステップ2 | geninstack | Instackenv JSONファイルの生成| osp16_generate_instackenv.yml -e
podname=
```

#ステップ3 | cimc vd | CIMC VDの設定| osp16\_cimc\_vd\_configure.yml

#ステップ4 | preocdeploy | Overcloud導入前の設定| osp16\_pre\_overcloud\_deploy.yml

#ステップ5 | importnodes | OpenstackベアメタルIronicノードのインポート|  
osp16\_import\_ironic\_nodes.yml

#ステップ6 | gentemplates | カスタムテンプレートの生成| osp16\_generate\_custom\_templates.yml

#ステップ7 | ocdeploy | Openstack Overcloudの導入| osp16\_overcloud\_deploy.yml

#ステップ8 | geninventory | インベントリファイルの生成| osp16\_build\_inventory\_v3.py  
—ipmipass

#ステップ9 | offlinerepo | オフラインTARファイルからのOvercloudオフラインリポジトリの設定|  
osp16\_config\_offline\_repo.yml

# step10 | fencing | Post Deploy MOP Before Reboot – フェンシングの設定|  
osp16\_config\_fencing.yml

#ステップ11 | raidcache |再起動前のMOPの導入後 – RAIDキャッシュとPRチューニング|  
osp16\_raid\_cache\_tuning.yml

#ステップ12 | dnfupdate |再起動前の導入後MOP - Dnf更新パッケージ|  
dnf\_update\_all\_packages.yml

#ステップ13 | setiplink |再起動前のMOPの導入後 – VF IPリンク信頼をオンに設定|  
osp16\_setIpLink.yml

# step14 | watchdog | Post Deploy MOP Before Reboot - Config IPMI Watchdog |  
osp16\_config\_ipmi\_watchdog.yml

#ステップ15 | icedriver |再起動前のMOPの導入後 – E810 ICEドライバの更新|  
osp16\_ice\_driver\_install.yml

# step16 | reboot | Reboot All Overcloud Nodes | osp16\_reboot\_overcloud\_hosts.yml

#ステップ17 | verifyrhosp | RHOSP導入構成と健全性の確認| osp16\_rhosp\_verify.yml -e  
podname=

#=====

到達可能なすべてのホスト

=====

実行されたチェック=> 17

合格チェック=> 17

失敗したチェック=> 00

=====

全体的なステータス=>合格!!

=====

Overcloudの導入が成功したら、Horizonダッシュボードにアクセスできることを確認します。

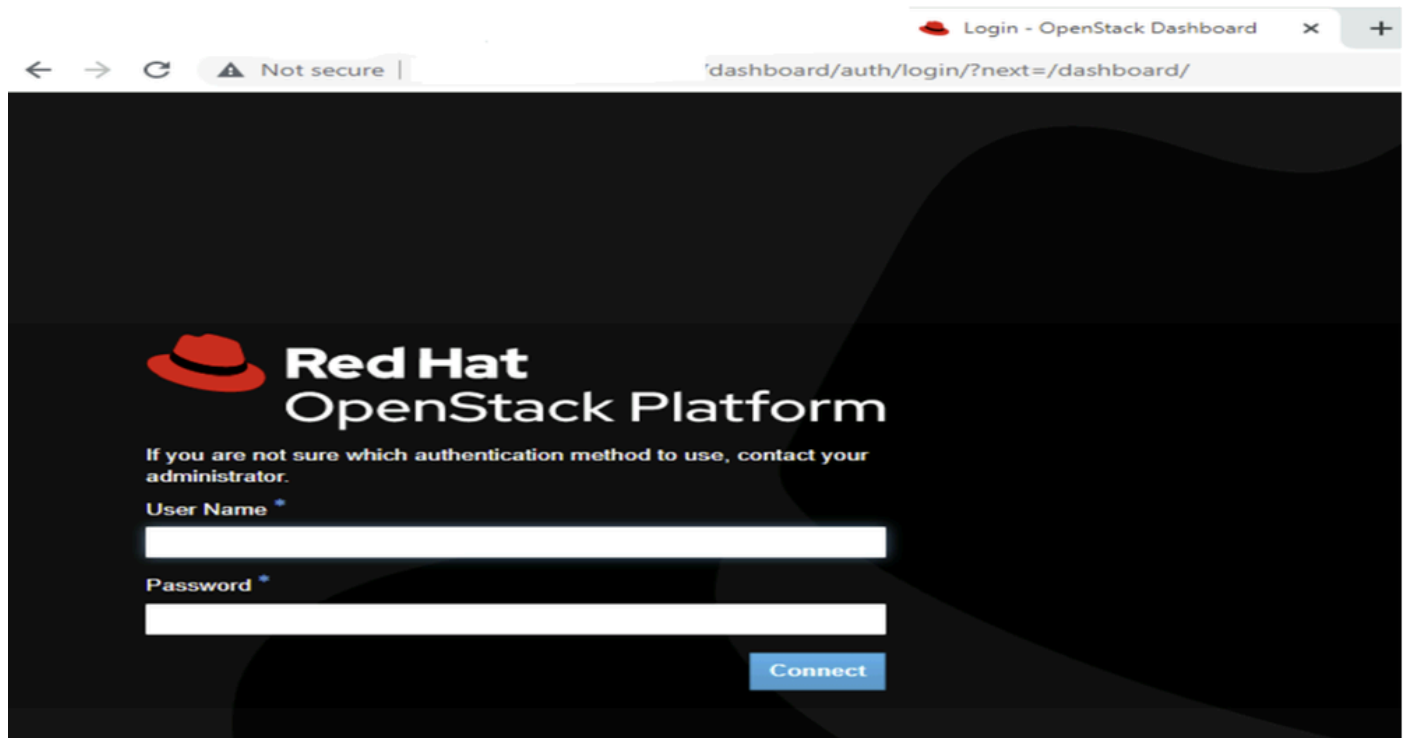
Horizon Dashboardへのアクセス

horizon dashboard URLには、「overcloudrc」の「OS\_AUTH\_URL」を使用します。

```
[stack@MUMBMUMBTCUDR201CO-ospd ~]$ cat MUMBMUMBTCUCL200C0rc
# Clear any old environment that may conflict.
for key in $( set | awk '{FS="="} /^OS_/ {print $1}' ); do unset $key ; done
export NOVA_VERSION=1.1
export COMPUTE_API_VERSION=1.1
export OS_USERNAME=admin
export OS_PROJECT_NAME=admin
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_DOMAIN_NAME=Default
export OS_NO_CACHE=True
export OS_CLOUDNAME=MUMBMUMBTCUCL200C0
export no_proxy=', [REDACTED], [REDACTED]'
export PYTHONWARNINGS='ignore:Certificate has no, ignore:A true SSLContext object is not available'
export OS_AUTH_TYPE=password
export OS_PASSWORD=openstack
export OS_AUTH_URL='http://[REDACTED]:5000'
export OS_IDENTITY_API_VERSION=3
export OS_COMPUTE_API_VERSION=2.latest
export OS_IMAGE_API_VERSION=2
export OS_VOLUME_API_VERSION=3
export OS_REGION_NAME=regionOne

# Add OS_CLOUDNAME to PS1
if [ -z "${CLOUDPROMPT_ENABLED:-}" ]; then
  export PS1=${PS1:-""}
  export PS1=\${OS_CLOUDNAME:+"(\${OS_CLOUDNAME})\ } $PS1
  export CLOUDPROMPT_ENABLED=1
fi
```

Horizonダッシュボード :



## RHOSPクラスタのヘルスチェック

```
<#root>
```

```
### Check OpenStack Services Status ###
```

```
# openstack compute service list  
# openstack network agent list  
# openstack volume service list  
# openstack orchestration service list  
# openstack identity service list  
# openstack endpoint list  
# openstack server list  
# openstack image list
```

## 要約

RHOSP 16.2導入ガイドでは、Red Hatの実績あるツールと手法を使用して、スケーラブルで実稼働対応のOpenStackクラウド環境を導入するための手順を包括的に説明します。このガイドは、システム管理者およびクラウドアーキテクト向けに作成されており、TripleO(OpenStack on OpenStack)に基づくOpenStackディレクタを使用したRHOSP 16.2の導入に焦点を当てています。

。

このガイドでは、次のような導入の重要なフェーズをすべて取り上げます。

- インフラストラクチャの計画と前提条件
- 環境の準備とネットワーク構成
- クラウド下でのインストールと設定
- クラウドの過剰導入と導入後の手順
- HA、ストレージ、サービスの拡張オプション

このガイドは、エコシステムの統合とRed Hatのサポートにより、信頼性の高いエンタープライズグレードのクラウドプラットフォームを求めるチームに不可欠です。

## 翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。