Guide de déploiement de RHOSP sur UCS pour la charge de travail de mobilité en mode déconnecté

Table des matières

- Introduction
- Conditions préalables
  - Exigences
  - Composants utilisés
- Informations générales
- Présentation du RHOSP
  - Architecture RHOSP
  - Sous-Nuage
    - Composants sous-cloud
  - Surnuage
    - Contrôleur
    - Calculer
    - Stockage
  - Référentiel local (serveur REPO)
  - Mise en réseau RHOSP
    - Connectivité physique RHOSP
    - Connectivité logique RHOSP
- Réglage des paramètres matériels
- Installation de l'hyperviseur-KVM et création du réseau
- Création de VM REPO et Director
  - Conditions préalables à la création de VM REPO et Director
  - Création de VM REPO
  - Création de VM Director
- Création de REPO hors connexion locale
- Déploiement cloud RHOSP
  - Conditions préalables
  - Mise à jour des fichiers d'entrée
  - Déploiement sous-cloud
  - Déploiement Overcloud
  - Accès au tableau de bord Horizon
- Vérification du fonctionnement du cluster RHOSP
- Résumé

Introduction

Ce document décrit une structure pour déployer RHOSP sur les serveurs UCS C220 M6 afin de prendre en charge Cisco VPC-DI.

Conditions préalables

Exigences

Cisco vous recommande d'avoir des connaissances de Red Hat OpenStack Platform (RHOSP) et de posséder de solides compétences en Red Hat Enterprise Linux (RHEL). En outre, une bonne compréhension des concepts de virtualisation et de mise en réseau est requise.

Composants utilisés

Ce document n'est pas limité à des versions de matériel et de logiciel spécifiques.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l’incidence possible des commandes.

Informations générales

Ce guide décrit l'intégration de RHOSP à l'infrastructure UCS (Unified Computing System), en mettant l'accent sur l'évolutivité, la fiabilité et l'optimisation des performances.

Il détaille les meilleures pratiques et utilise une automatisation basée sur des scripts pour le déploiement d'OpenStack TripleO comprenant une architecture Undercloud et Overcloud.

Grâce à ce guide de déploiement, les entreprises peuvent mettre en place une infrastructure cloud RHOSP robuste et efficace, adaptée aux fonctions de mobilité et de réseau virtuel (VNF) basées sur Cisco Virtual Packet Core - Distributed Instance (VPC-DI).

Présentation du RHOSP

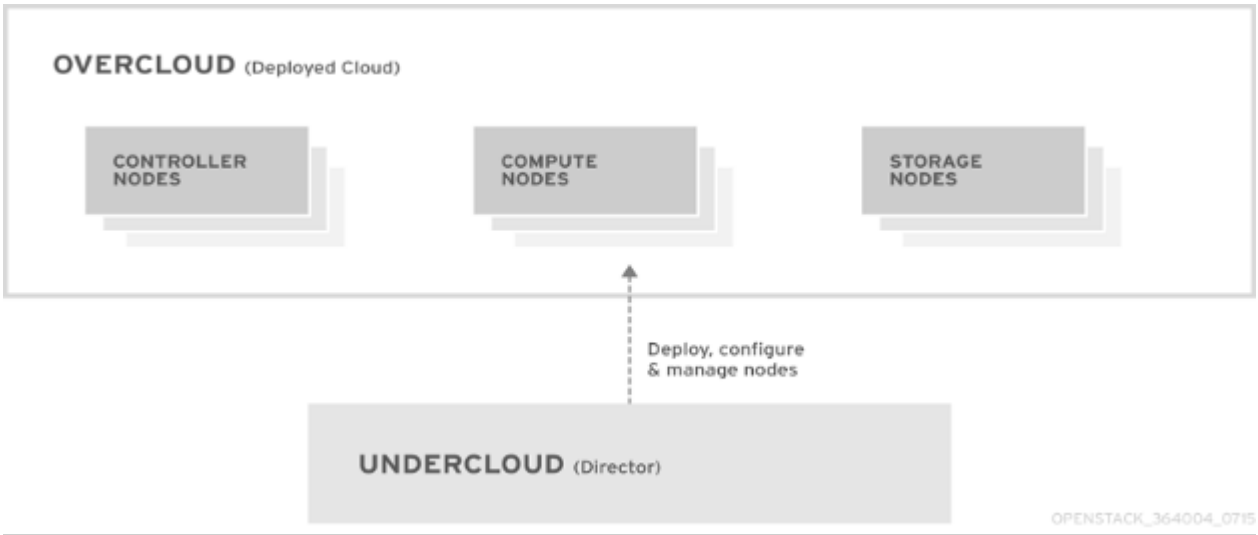
RHOSP est une solution de cloud privé de niveau entreprise basée sur le projet OpenStack open source, intégrée et prise en charge par Red Hat. Elle permet aux entreprises de déployer et de gérer l'infrastructure en tant que service (IaaS) pour les machines virtuelles (VM), la mise en réseau et le stockage à la demande.

Il offre des fonctionnalités telles que la haute disponibilité, la virtualisation des fonctions réseau et les déploiements personnalisables.

Architecture RHOSP

Le RHOSP est basé principalement sur le projet OpenStack TripleO. Openstack utilise Director qui agit comme un ensemble d'outils pour installer et gérer un environnement RHOSP complet.

RHOSP est conçu pour fournir une infrastructure cloud évolutive et flexible. Son architecture se compose de deux éléments principaux : Sous-cloud et sur-cloud.



Sous-Nuage

Le sous-cloud est le noeud de gestion principal qui contient l'ensemble d'outils du directeur RHOSP. Il s'agit d'une installation OpenStack à système unique qui inclut des composants pour le provisionnement et la gestion des noeuds OpenStack qui forment l'environnement OpenStack (Overcloud).

Composants sous-cloud

Le sous-cloud utilise des composants OpenStack comme jeu d'outils de base. Chaque composant fonctionne dans un conteneur distinct sur le nuage inférieur :

- Identité OpenStack (clé de voûte) : fournit l'authentification et l'autorisation des composants du directeur
- OpenStack Bare Metal (ironique) - Gère les noeuds sans système d'exploitation
- OpenStack Networking (neutron) et Open vSwitch : contrôle du réseau pour les noeuds sans

système d'exploitation
- Orchestration OpenStack (chaleur éphémère) - Assure l'orchestration des noeuds après que le directeur a écrit l'image Overcloud sur le disque

Surnuage

L'Overcloud est l'environnement RHOSP résultant créé à l'aide du sous-cloud. Cela inclut différents rôles de noeud définis en fonction de l'environnement OpenStack Platform (OSP) que le client souhaite créer.

Contrôleur

Les noeuds de contrôleur assurent l'administration, la mise en réseau et la haute disponibilité de l'environnement OpenStack. Un environnement OpenStack recommandé contient trois noeuds de contrôleur dans un cluster haute disponibilité.

Calculer

Les noeuds de calcul fournissent des ressources informatiques pour l'environnement OpenStack. Les noeuds de calcul peuvent être évolutifs en fonction des besoins du réseau au fil du temps. Un noeud de calcul par défaut contient les composants mentionnés ci-dessous :

i. OpenStack Compute (nova)
ii. Machine virtuelle basée sur un noyau (KVM)/émulateur rapide (QEMU)
iii. Ouvrir vSwitch

Stockage

Les noeuds de stockage fournissent un stockage pour l'environnement OpenStack.



Remarque : Il existe plusieurs approches pour le déploiement de sous-cloud/Director et de référentiel hors ligne (REPO) dans le réseau du client. Elles peuvent être déployées directement sur un noeud sans système d'exploitation ou en tant que machine virtuelle sur l'hyperviseur KVM. Dans le guide de déploiement actuel, le serveur Director UCS héberge un KVM (hyperviseur) pour déployer plusieurs machines virtuelles au-dessus. Les noeuds RHOSP Director et Offline-REPO sont déployés en tant que VM sur l'hyperviseur KVM.

Référentiel local (serveur REPO)

Red Hat fournit un utilitaire appelé reposync qui peut être utilisé pour télécharger les packages à partir du réseau de diffusion de contenu (CDN). Pour télécharger tous les packages à partir d'un canal spécifique, le système doit être abonné à ce canal. Si le système n'est pas abonné au canal requis, la fonction de resynchronisation ne peut pas télécharger et synchroniser ces packages sur le système local.

Les référentiels sont configurés dans /etc/yum.repos.d/ chemin par des fichiers qui se terminent par l'extension .repo. Plusieurs référentiels peuvent être définis dans le même fichier.

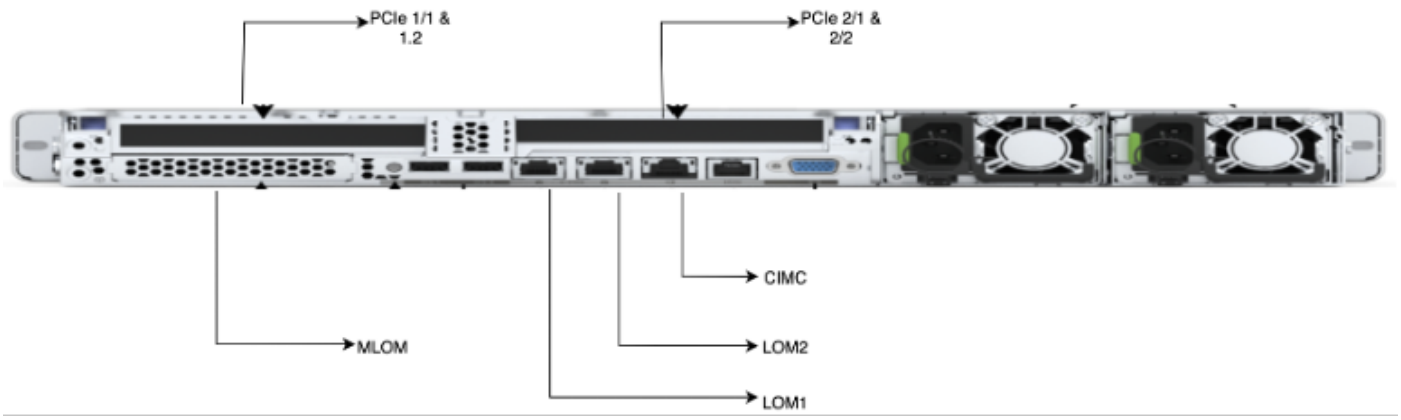
Mise en réseau RHOSP

Le service réseau (neutron) est le composant SDN (Software-Defined Networking) du protocole RHOSP. Le service de mise en réseau RHOSP gère le trafic interne et externe en provenance et à destination des instances de VM et fournit des services de base tels que le routage, la segmentation, le protocole DHCP et les métadonnées. Il fournit l'API pour les fonctionnalités de mise en réseau virtuelle et la gestion des commutateurs, des routeurs, des ports et des pare-feu.

Le directeur RHOSP mappe les services OpenStack à différents réseaux isolés. Les réseaux qui acheminent chaque type de trafic sont les suivants : Cisco Integrated Management Controller (CIMC), Provisioning, Internal API, Storage Data, Storage Management, Tenant and External (SSH) et Operations, Administration, and Maintenance (OAM).

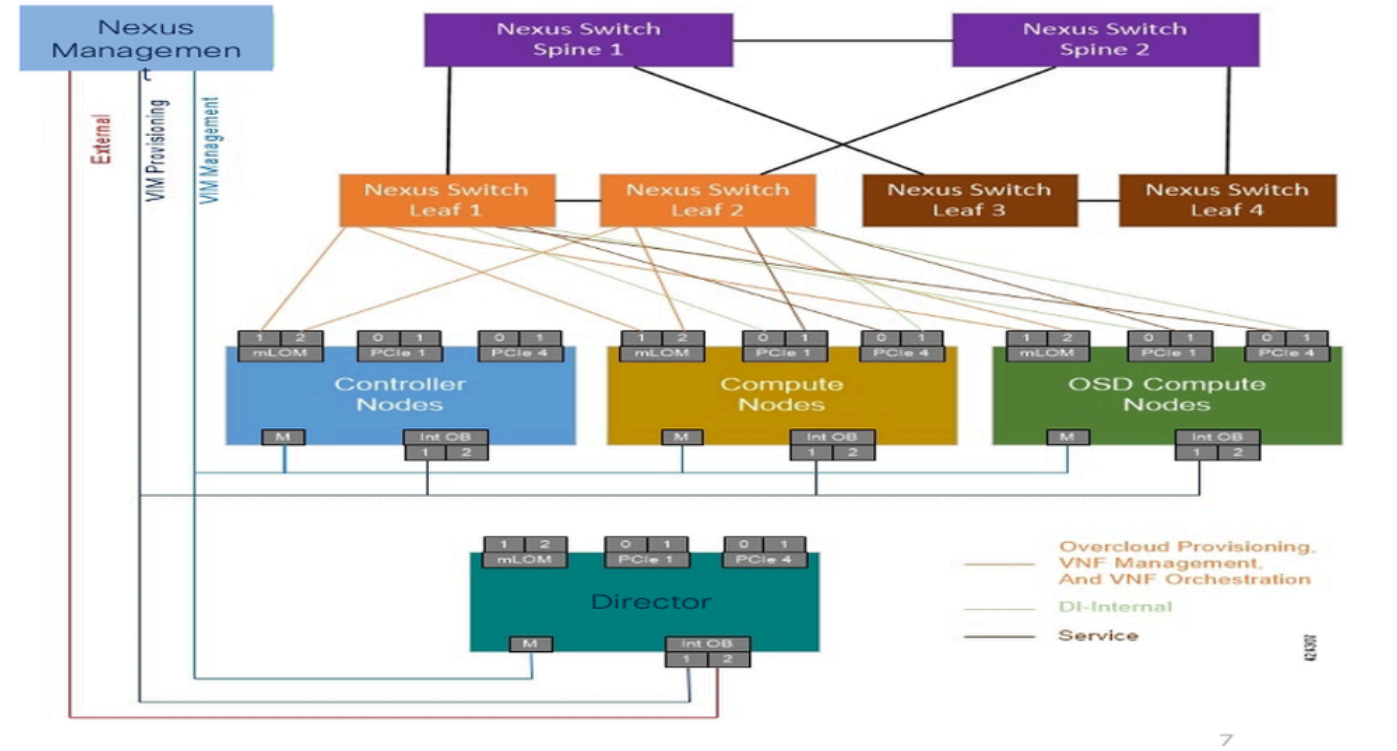
Connectivité physique RHOSP

Le déploiement RHOSP utilise différents ports physiques des serveurs Cisco UCS C220 M6 à des fins de connectivité différentes.



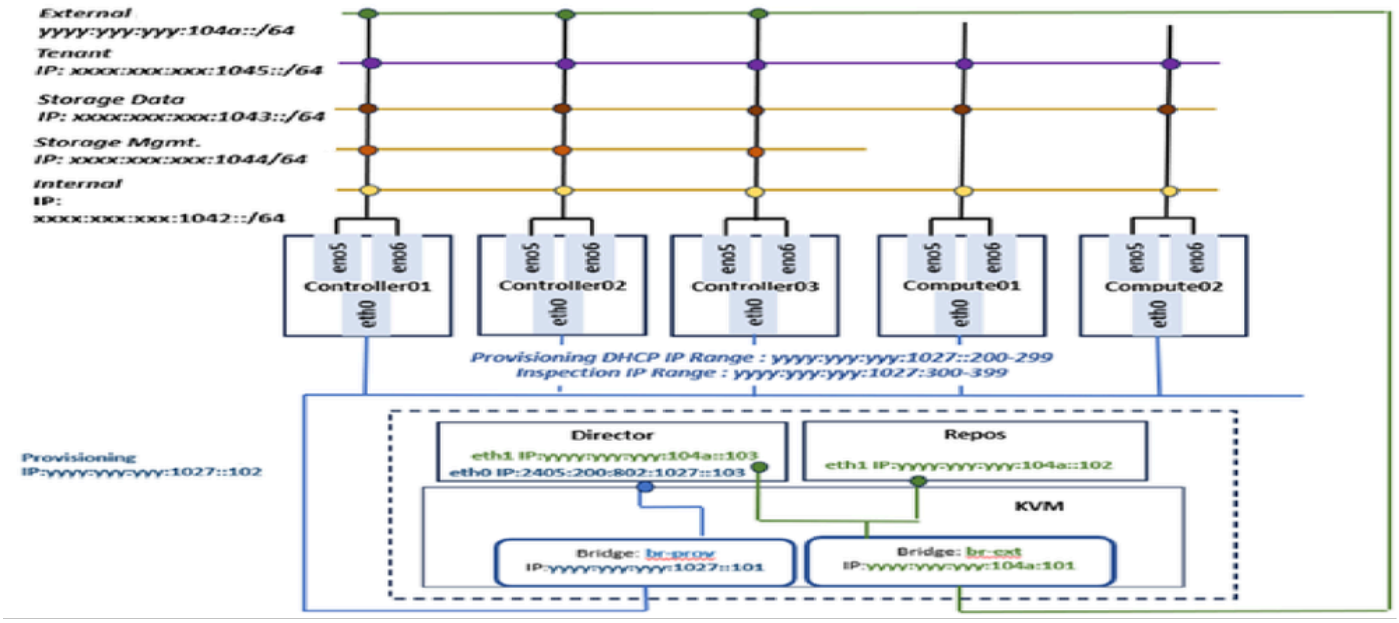
Numéro de série	Ports physiques	Détails
1.	CIMC	CIMC fournit une connectivité hors bande pour le provisionnement et la gestion des serveurs.
2.	Virtualisation d'E/S racine unique (SR-IOV)/PCIe (Peripheral Component Interconnect Express)	Les cartes d'interface réseau (NIC) PCIe sont utilisées sur les noeuds de calcul pour les réseaux DI-interne et de service pour le VNF.
3.	Réseau local modulaire sur carte mère (MLOM)	Les ports MLOM sont configurés en tant que liaisons. osp_external, osp_internal, osp_tenant, osp_external, osp_storage_data, osp_storage_mgmt utilise le port MLOM pour la communication interne.
4.	LAN sur carte mère (LOM)	Le directeur utilise les ports LOM1 et LOM2, tandis que les ordinateurs et les contrôleurs utilisent uniquement le port LOM1. LOM1 est utilisé pour déployer ou provisionner la pile Open sur tous les serveurs. LOM2 est utilisé comme OAM (réseau externe) sur le directeur.

Le schéma suivant illustre la connectivité physique avec les serveurs.



Connectivité logique RHOSP

Le réseau RHOSP comporte plusieurs sous-réseaux desservant différents services au sein du cloud.



- OSP_CIMC :

CIMC est l'interface IPMI (Intelligent Programming Management Interface) qui contrôle la gestion

de tous les serveurs UCS. Ce réseau CIMC est configuré sur le port CIMC autonome de tous les serveurs UCS.

- OSP_Provisioning :

Ce réseau est responsable de la mise en service et de la gestion du démarrage PXE (Preboot Execution Environment) des ordinateurs et des serveurs contrôleurs pendant le déploiement Overcloud, ainsi que de l'obtention de l'adresse IP DHCP. Le réseau Provisioning est configuré en tant que VLAN natif sur le port LOM1 de tous les serveurs UCS pour plus de simplicité et de compatibilité. Ce réseau de mise en service est responsable du déploiement du cloud sur tous les serveurs.

En raison de la virtualisation sur le serveur Director, un réseau de pont doit être créé sur le KVM pour la machine virtuelle Director afin de communiquer avec les autres serveurs.

- OSP_Interne :

Le réseau API interne est utilisé pour la communication entre les services OpenStack tels que neutron, nova, keystone, etc.

Le réseau OSP_Internal est configuré sur les ports MLOM liés sur les noeuds de contrôleur et de calcul.

- OSP_Tenant :

Le réseau du locataire est créé par défaut dans les projets cloud pour la gestion VNF. Dans la configuration actuelle, seul un projet Openstack est créé pour le déploiement VNF.

Le réseau OSP_Tenant est configuré sur des ports MLOM liés sur des noeuds de contrôleur et de calcul.

- OSP_External :

Le réseau externe est utilisé pour tous les réseaux d'accès externe (comme SSH) et d'API.

Le réseau OSP_External est configuré sur le port LOM2 sur le noeud Director et sur les ports MLOM liés sur les noeuds Controller et Compute.

- Données_Stockage_OSP :

Le réseau OSP_Storage est utilisé pour toutes les opérations liées à l'accès au stockage. Ceci est requis pour la communication entre le service CEPH et VNF qui ont besoin d'accéder au stockage. Il est utilisé par le contrôleur, les noeuds de calcul et le CEPH.

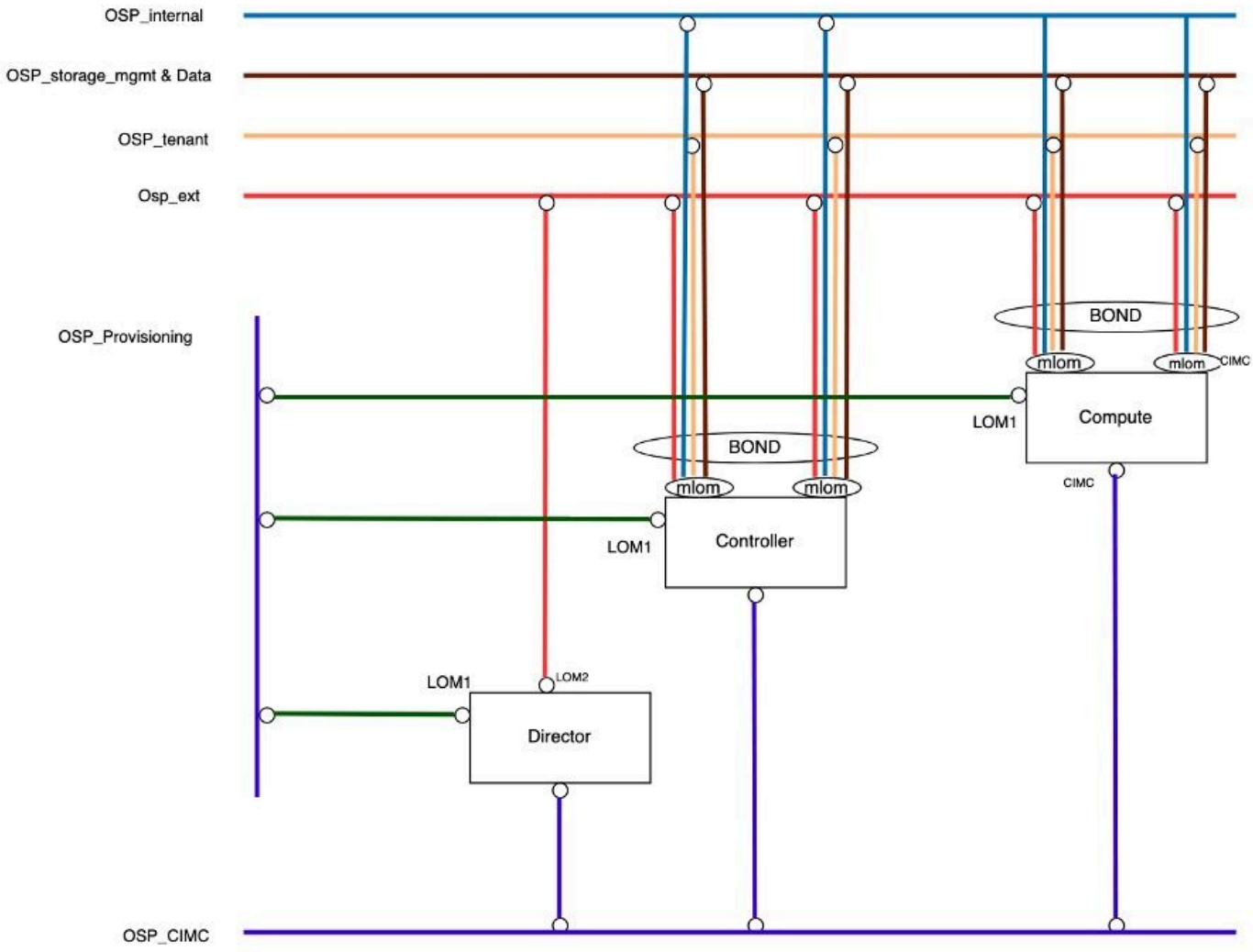
Le réseau OSP_Storage_Data est configuré sur des ports MLOM liés sur des noeuds de contrôleur et de calcul.

- OSP_Storage_Management :

OpenStack Object Storage utilise ce réseau pour synchroniser les objets de données entre les noeuds de réplica participants dans le cluster de stockage, formé entre les noeuds de calcul du contrôleur.

Le réseau OSP_Storage_Mgmt est configuré sur des ports MLOM liés sur des noeuds de contrôleur et de calcul.

Le schéma montre comment les réseaux logiques sous-cloud sont connectés à chaque type de noeuds dans le cluster RHOSP.



Réglage des paramètres matériels

- Assurez-vous que le protocole IP est activé et configuré avec une adresse de gestion correcte, conformément à la conception.
- Assurez-vous que l'utilisateur admin et son mot de passe doivent être identiques pour tous les accès de connexion CIMC.
- Assurez-vous que le mode Bootstrap (Boot) est défini sur Unified Extensible Firmware Interface (UEFI).
- Assurez-vous de définir les paramètres du BIOS avec les paramètres recommandés.
- Définissez l'ordre de démarrage sur :
    a. LOM-PXE (eno1)
    b. Boot-Disque dur (HDD)

- Définissez l'adresse MAC de LOM1 (eno1).
- Assurez-vous que tous les serveurs UCS/CIMC sont mis à niveau avec la dernière version recommandée du micrologiciel.

Installation de l'hyperviseur-KVM et création du réseau

Il existe plusieurs méthodes pour déployer le sous-cloud/Director et le REPO hors ligne dans le réseau du client. Ils peuvent être déployés directement sur un noeud sans système d'exploitation ou en tant que machines virtuelles exécutées sur un hyperviseur KVM.
Dans le guide de déploiement actuel, le serveur Director UCS est configuré pour héberger l'hyperviseur KVM, ce qui facilite la création de plusieurs machines virtuelles. Le noeud Directeur RHOSP et le noeud REPO hors ligne sont déployés en tant que machines virtuelles sur cet hyperviseur KVM.



Remarque : Pour déployer l'hyperviseur KVM, vous devez respecter les étapes d'installation KVM RHEL standard.

- Une fois le KVM activé, mettez à jour le nom d'hôte :
  hostnamectl set-hostname <hostname> —static

- Configurez les ponts externes et de mise en service et liez les interfaces.

br-prov : eth0

br-ext : eth1

Ces ponts doivent être créés via l'interface utilisateur graphique NMTUI (Network Manager Text User Interface).

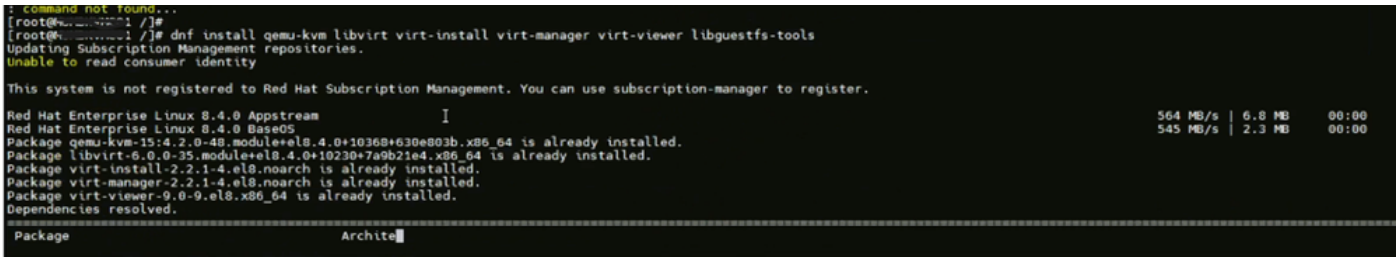
- Par défaut, les ports eno1 et eno2 sont créés dans KVM pour les ports LOM1 et LOM2 physiques, et vérifient la même chose à l'aide de l'adresse MAC de la journalisation CIMC.
- Créez un réseau en pont et ajoutez-y des ports esclaves en mappant une adresse MAC correcte.
- Une fois le pont créé, assurez-vous que le provisionnement et la passerelle externe sont accessibles depuis KVM.

Création de VM REPO et Director

Conditions préalables à la création de VM REPO et Director

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

- Installez les packages requis dans KVM pour REPO et Director VM Creation.



Installer les packages libvirt et qemu

- Créez les répertoires requis sur KVM.

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

- Copiez les fichiers dans les répertoires.

```
# 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/
```

- Montez l'ISO sur /mnt/iso.

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

- Créez le fichier REPO à l'adresse /etc/yum.repos.d path.

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

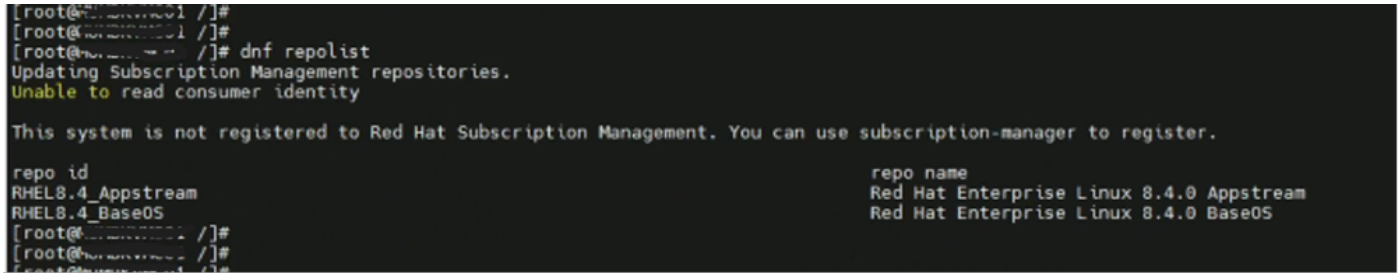
[RHEL8.4_Appstream]
name=Red Hat Enterprise Linux 8.4.0 Appstream
mediaid=None
```

```
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/
```

- Vérifiez la liste de répolation et assurez-vous que les RÉPO Appstream et Baseos sont mappés.

```
# dnf repolist
```



Repoliste

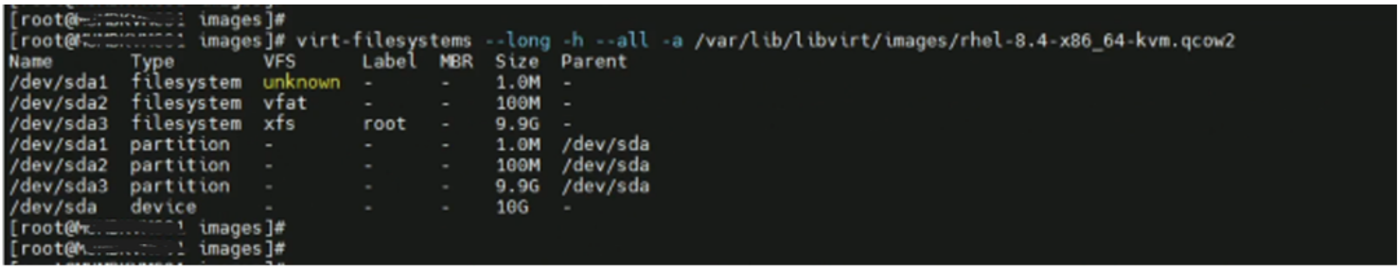
Création de VM REPO

```
$ 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@12
```



Personnalisation d'image RHEL

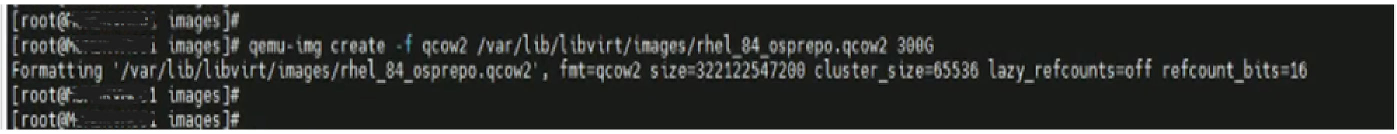
```
$ virt-filesystems --long -h --all -a /var/lib/libvirt/images/rhel-8.4-x86_64-kvm.qcow2
```



Mappage du système de fichiers

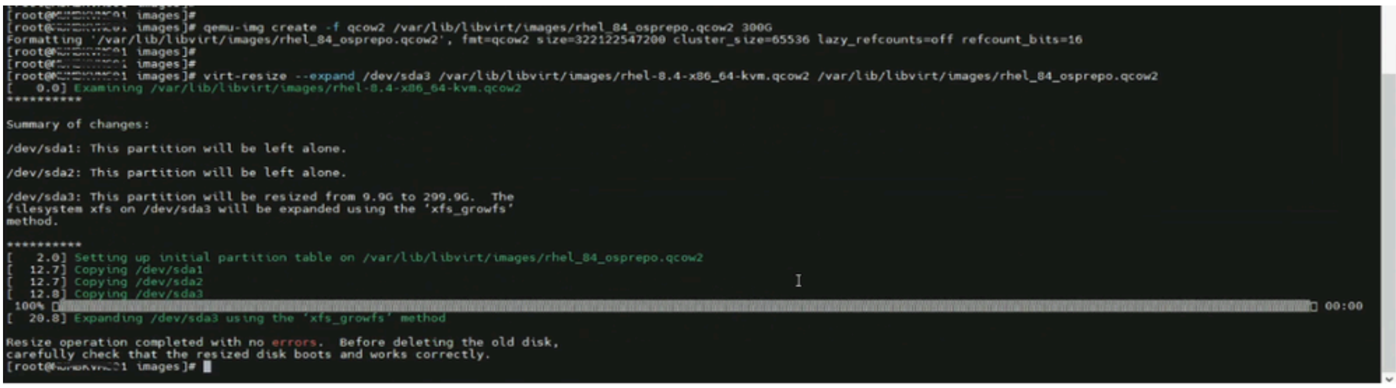
- Créez l'image du serveur REPO.

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



Création d'image Repo

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



Installation Repo

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

```
[root@[redacted] images]#
[root@[redacted] images]#
[root@[redacted] images]# qemu-img create -f qcow2 -b /var/lib/libvirt/images/rhel_84_osprepo.qcow2 -F qcow2 /data/images/OSPREPO_RHEL_84.qcow2
Formatting '/data/images/OSPREPO_RHEL_84.qcow2', fmt=qcow2 size=322122547200 backing_file=/var/lib/libvirt/images/rhel_84_osprepo.qcow2 backing_fmt=qcow2 cluster_size=65536 lazy_refcounts=off refcount_bits=16
[root@[redacted] images]#
[root@[redacted] images]#
```

Création d'image Repo

```
$ guestfish -a /data/images/OSPREPO_RHEL_84.qcow2 -i ln-sf /dev/null /etc/systemd/system/cloud-init.ser
```

```
[root@[redacted] images]#
[root@[redacted] images]#
[root@[redacted] images]# guestfish -a /data/images/OSPREPO_RHEL_84.qcow2 -i ln-sf /dev/null /etc/systemd/system/cloud-init.service
[root@[redacted] images]#
[root@[redacted] images]#
[root@[redacted] images]#
```

Guestfish OSP Repo

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

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

** (osinfo-query:49279): WARNING **: 17:12:33.856: Entity http://pcisig.com/pci/1B36/0100 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@[redacted] images]#
[root@[redacted] images]#
[root@[redacted] images]#
```

Packages RHEL

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

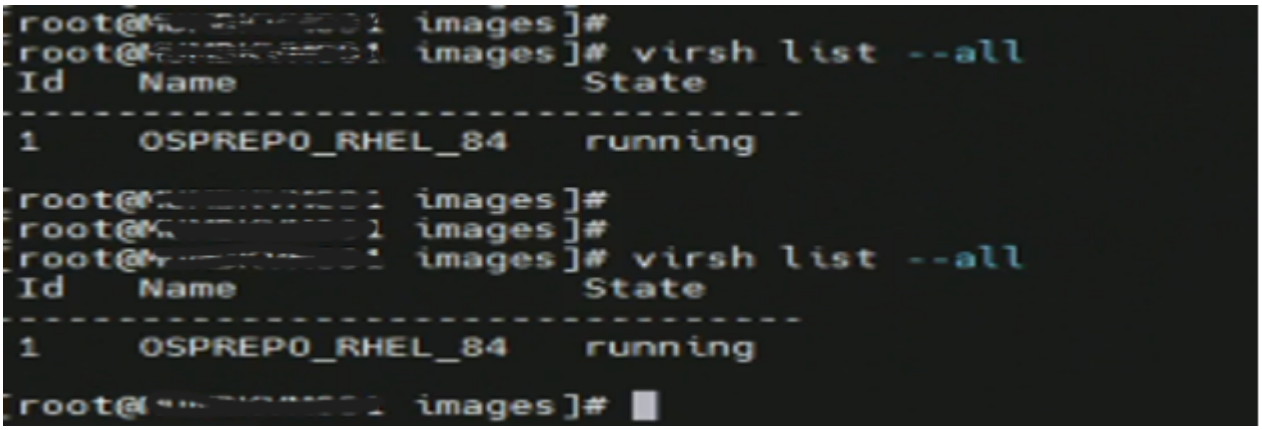
```
[root@[redacted] images]#
[root@[redacted] images]# virt-install --cpu host --memory 32768 --vcpus 16 --os-variant rhel8.4 --disk path=/data/images/OSPREPO_RHEL_84.qcow2,device=disk,bus=virtio,format=qcow2
--import --noautoconsole --vnc --network bridge:br-ext --name OSPREPO_RHEL_84

** (process:49296): WARNING **: 17:13:15.813: Entity http://pcisig.com/pci/1B36/0100 referenced but not defined

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

Installation du serveur Repo

```
$ virsh list --all
```

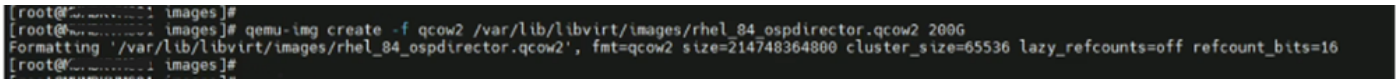


État du serveur Repo

Création de VM Director

- Créez l'image du serveur directeur.

```
# qemu-img create -f qcow2 /var/lib/libvirt/images/rhel_84_ospdirector.qcow2 500G
```



Création d'images Director

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



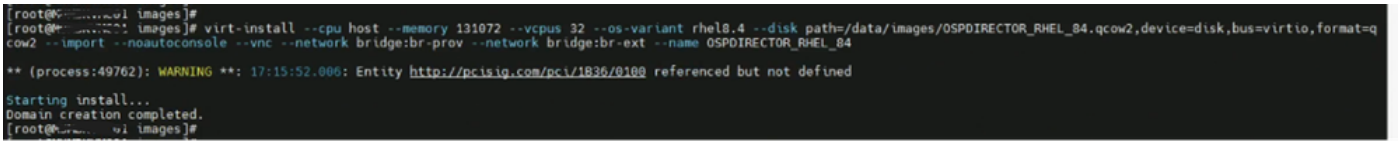
virt-resize de /dev/sda3

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



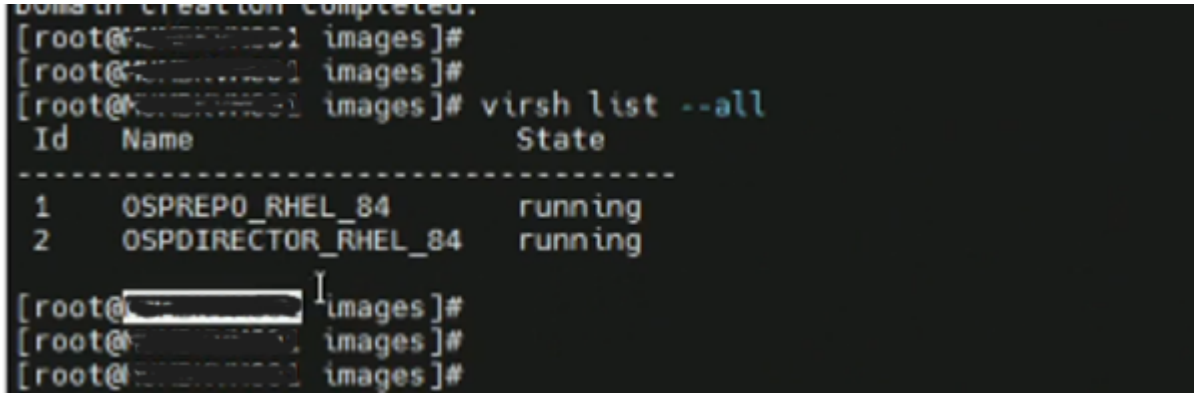
Création d'images Director

```
# 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
```

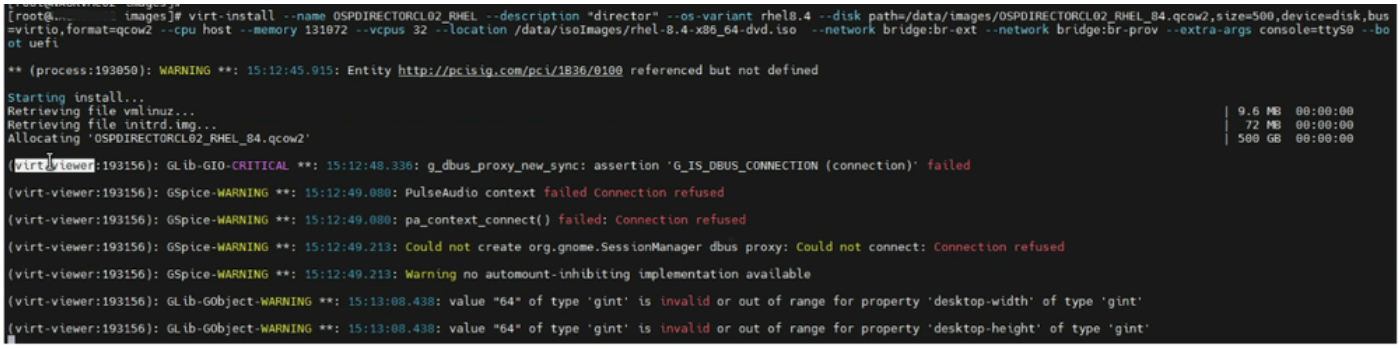


Création de VM Director

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



Création de VM Director

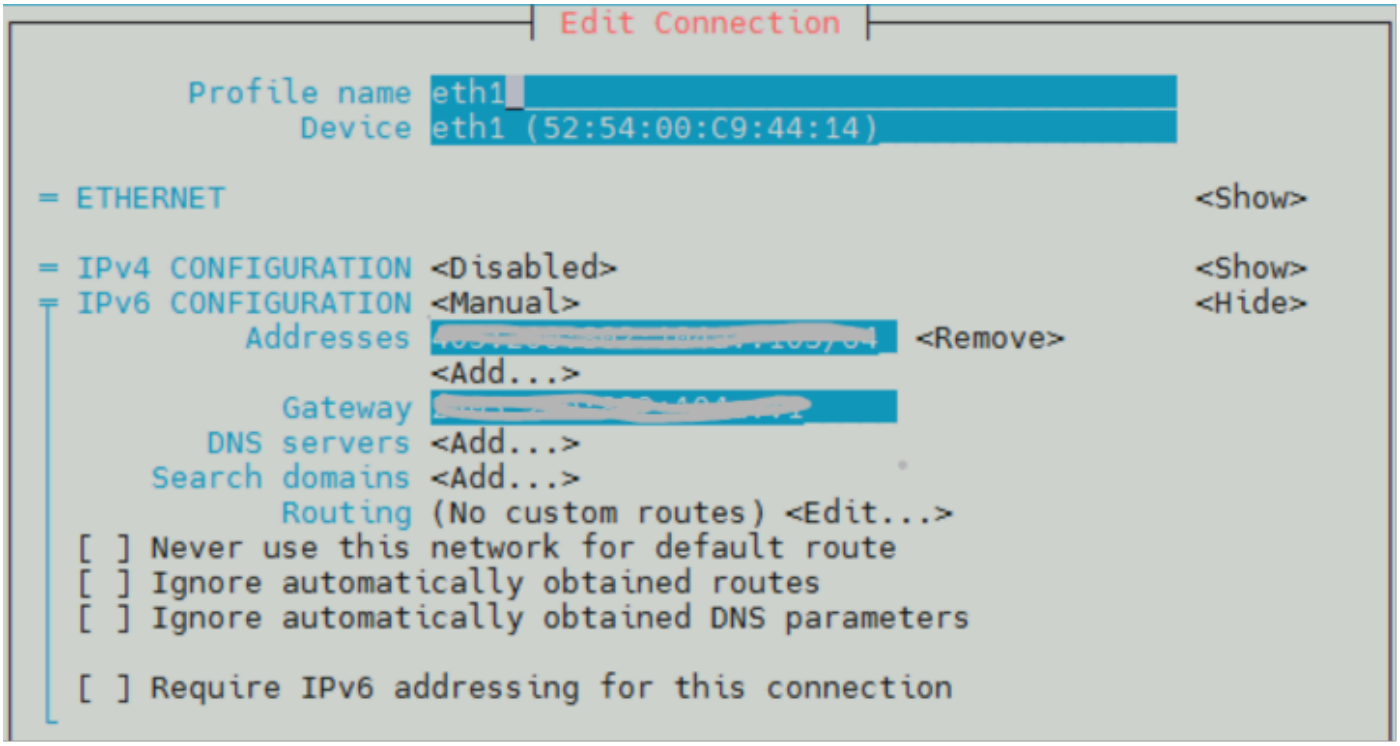


Installation de Director VM

- Ouvrez une nouvelle fenêtre et ouvrez la connexion console de la machine virtuelle.

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

- Après le déploiement de la machine virtuelle Director, deux interfaces eth0 et eth1 sont créées.
    - eth0 est utilisé pour le réseau OSP_provisioning, dans lequel pendant le déploiement br-ctlplane réseau sont créés pendant le sous-nuage.
    - eth1 est utilisé pour la connectivité externe pour l'accès SSH. Ainsi, l'adresse IP de sous-réseau OSP_external est attribuée.



Configuration réseau de Director

Création de REPO hors connexion locale

Le serveur REPO doit être enregistré auprès de Red Hat CDN et doit disposer du référentiel de tous les packages RHOSP 16.2 disponibles nécessaires au déploiement. Les packages RPM RHEL et les images du conteneur RHOSP doivent être téléchargés sur la machine virtuelle REPO à l'aide d'un proxy.

Déploiement cloud RHOSP

RHOSP 16.2 est déployé sur le réseau du client via l'automatisation. Les scripts réactifs sont utilisés pour automatiser le déploiement des applications Undercloud et Overcloud.

Conditions préalables

Étapes à observer avant de commencer le déploiement réel du cloud :

1. Assurer la connectivité du noeud KVM avec la machine virtuelle REPO locale et la machine virtuelle Director sur le réseau OSP_EXT.

2. Assurez-vous que tous les scripts valides et d'automatisation sont chargés sur l'hôte KVM dans le dossier désigné.

3. Créez des dossiers avec des noms comme 'cisco' et 'automation' et mettez l'archive des scripts ansibles.

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

L'archive tar se composera de trois structures de répertoires, nommées ainsi :

- scripts : se compose de scripts shell utilisés pour la configuration de noeud sans système d'exploitation.
- rpms : se compose de pilotes de connexion Ethernet Intel (ICE), de pilotes Intel et de packages Red Hat Package Manager (RPM).
- ansible : se compose du fichier Input vars, des fichiers ansible yaml pour le déploiement et des modèles jinja.

4. Installez le package sshpass. sshpass est un utilitaire de ligne de commande utilisé pour fournir des mots de passe à ssh de manière non interactive. Il est principalement utilisé dans les scénarios de script ou d'automatisation où la saisie manuelle de mot de passe n'est pas possible.

- Téléchargez le package sshpass sshpass.tar.gz à partir d'Internet/du serveur existant.
- Installez le package GNU Compiler Collection (GCC).

```
# yum install gcc
```

- Installez les packages make.

```
# yum install make
```

- Décompressez le paquet shpass et installez SSH pass.

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

5. Le processus d'installation du répertoire nécessite l'exécution de commandes par un utilisateur non racine. L'utilisateur « Stack » doit être créé dans Director VM avec un accès sudo.

```
# 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. Copiez le fichier rootCA.crt du serveur REPO vers la machine virtuelle Director et la machine virtuelle KVM au chemin indiqué. Mettez également à jour le certificat de la machine virtuelle REPO dans la liste de confiance.

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

7. Mettez à jour les détails du nom d'hôte du serveur REPO local dans Director VM et KVM dans le fichier /etc/hosts.

8. Sur KVM et Director VM, installez des packages supplémentaires tels que python, ansible, etc. afin d'exécuter des scripts d'automatisation ansible.

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

9. Le sous-réseau CIMC doit être accessible à partir du réseau de mise en service de Director

pour permettre la mise en service lors du déploiement en cloud. Si nécessaire, ajoutez une route statique pour la même route.

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

10. Dans KVM et Director VM, créez un fichier hôte sous le dossier /ansible et ajoutez les détails spécifiques à la pile, le cas échéant.

```
[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. Assurez-vous que tous les guides et fichiers d'entrée valides doivent être conservés dans Director VM sous le dossier /home/stack.

Mise à jour des fichiers d'entrée

Il existe un fichier de variables d'entrée composé de détails spécifiques au réseau du client qui doivent être préparés pour le déploiement en cloud.

Chemin : /home/cisco/automation/ansible/podvars

Nom du fichier : **<**nom-pile**>**_vars.yml

Mettez à jour les paramètres mis en surbrillance conformément au plan IP spécifique au site/document de conception de bas niveau.



Remarque : Les adresses IP factices sont utilisées à des fins de représentation uniquement.

<#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: '<stack name>'



# OSPD full hostname

fqdn_hostname: '<stack name>.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 pasword 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: '<Cloud-name>.<domain-name>',
    port: '5000',
    container_tag: '16.2.6',
    user: 'ciscoadmin',
    password: '******'
}

# Custom cloud domain details
domain_name: {
    domain: '<domain-name>',
    cloudshortname: 'nl'
}

# 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 verison
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
# ##################################
```

Déploiement sous-cloud

Undercloud est déployé à l'aide de scripts tangibles en sept étapes. Toutes les étapes doivent être exécutées à partir de l'hôte KVM qui agit en tant qu'hôte de saut dans ce cas.

Étape	Étiquette	Description	Guide des YAML
Étape 1.	fichiers de contrôle	Vérifiez les guides, scripts et RPM requis sur Openstack Platform Director (OSPD).	osp16_publisher_playbooks_verify.yml
Étape 2.	genpodvars	Générez des fichiers variables spécifiques à POD relatifs au matériel, RHEL, etc., comme common_vars.yml (hardware,	osp16_generate_pod_specific_vars.yml

		software, network details), hw_m6_vars.yml (CPU, memory, hugepages, disk, NIC, etc), rhel_84_vars.yml (RHEL, kernel, ICE driver, NIC version), pf_esc_vars.yml (Elastic Services Controller (ESC) details), osp_16_vars.yml (version OSP, fuseau horaire, type IP, ID VLAN, IP, détails neutron).	
Étape 3.	précuploïde	Configurez le nom de domaine complet (FQDN), le protocole NTP (Network Time Protocol) et mettez à jour tous les packages sur le noeud directeur.	osp16_pre_undercloud_deploy.yml
Étape 4.	bottillon	Effectue le premier redémarrage du noeud directeur Undercloud après la configuration et l'installation du package précédents.	osp16_undercloud_deploy.yml
Étape 5.	ucdeploy	Installer la pile sous-cloud sur le directeur	osp16_undercloud_tuning.yml
Étape 6.	cstate	Configurez les paramètres d'état C du processeur du BIOS sur le noeud directeur.	osp16_cstate.yml
Étape 7.	secondreboot	Deuxième redémarrage sur le directeur Undercloud après les modifications du BIOS.	S/O

Le fichier nommé osp16_auto_undercloud_deploy.yml est le principal manuel ansible-playbook qui peut être exécuté en une seule itération, mais il est conseillé d'exécuter le manuel par étapes en utilisant différentes balises pour faciliter le dépannage en cas de problèmes de déploiement.

<#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
```

Déploiement Overcloud

Le surcloud est déployé avec au moins trois contrôleurs en mode haute disponibilité et un ordinateur. Overcloud est déployé à l'aide de scripts tangibles en 17 étapes. Toutes les étapes doivent être exécutées à partir de Director-VM qui agit en tant qu'hôte de saut dans ce cas.

Étape	Étiquette	Description	Guide des YAML
Étape 1.	genpodvars	Générez des fichiers variables spécifiques à POD pour	osp16_generate_pod_specific_vars.yml

		Overcloud liés au matériel, RHEL, etc., comme common_vars.yml (hardware, software, network details), hw_m6_vars.yml (CPU, memory, hugepages, disk, NIC, etc), rhel_84_vars.yml (RHEL, kernel, ICE driver, NIC version), pf_esc_vars.yml (ESC details), osp_16_vars.yml. (Version OSP, fuseau horaire, type IP, ID VLAN, IP, détails neutron).	
Étape 2.	pile de gènes	Générez le fichier JSON Instackenv à partir de /var/common_vars.yml créé à l'étape précédente. Le directeur nécessite un modèle de définition de noeud, qui est créé manuellement. Ce fichier instackenv.json utilise le format JSON et contient tous les détails de gestion du matériel et de l'alimentation pour les noeuds. Cette étape valide également la configuration matérielle sur le serveur UCS avant de générer le fichier.	osp16_generate_instackenv.yml
Étape 3.	cimcvd	Configurez les paramètres CIMC et les disques virtuels (VD) sur chaque serveur faisant référence aux common_vars.yml, hw_m6_vars.yaml et rhel_84_vars.yml.	osp16_cimc_vd_configure.yml
Étape 4.	prédéployer	Cette étape remplit toutes les conditions requises pour déployer l'Overcloud. Il définit le FQDN, le NTP, et met à jour tous les packages, transmet l'image au chemin pour le déploiement.	osp16_pre_overcloud_deploy.yml

Étape 5.	noeuds d'importation	Dans cette étape, le processeur du serveur est inspecté, la mémoire, la carte réseau avec l'interface et les ports sur les commutateurs réseau. L'introspection est effectuée sur les commutateurs réseau connectés pour tous les contrôleurs et ordinateurs.	osp16_import_ironic_nodes.yml
Étape 6.	gabarits	Générez des fichiers modèles personnalisés pour les contrôleurs et les ordinateurs. Dans le modèle personnalisé, définissez les rôles de contrôleur et de calcul pour tous les services qui s'exécutent sur eux. Il effectue également le durcissement du système en appliquant des certificats, des routes, etc.	osp16_generate_custom_templates.yml
Étape 7.	codéployer	Dans cette étape, un déploiement OpenStack Overcloud est effectué. Exécutez deploy.sh fourni par Red Hat pour le déploiement RHOSP.	osp16_overcloud_deploy.yml
Étape 8.	geninventory	Dans cette étape, un fichier d'inventaire yml destiné à être utilisé par ansible est généré, dans lequel l'IP d'approvisionnement, l'IP IPMI (CIMC) et les informations d'identification sont stockés et mappés avec le contrôleur et les ordinateurs pour l'automatisation afin de se connecter au système et d'exécuter les étapes suivantes.	osp16_build_inventory_v3.py
Étape 9.	hors ligne	Configurez Overcloud Offline REPO dans le fichier /etc/yum.repo.d/offline.repo	osp16_config_offline_repo.yml

		pointez vers le serveur REPO via le réseau externe.	
Étape 10.	clôture	Configurez la délimitation sur tous les noeuds du contrôleur avec « Shoot The Other Node In The Head » (une technique de délimitation dans les clusters HA) (STONITH).	osp16_config_fencing.yml
Étape 11.	cache radio	Configurer le paramètre Raid Cache pour tous les contrôleurs et ordinateurs et configurer également les paramètres de stockage SWIFT.	osp16_raid_cache_tuning.yml
Étape 12.	dnifurpdate	Exécutez les mises à jour DNF pour tous les packages sur tous les noeuds.	dnf_update_all_packages.yml
Étape 13.	sétiplink	Au cours de cette étape, le contrôle du mode de confiance des ports SR-IOV est activé pour le trafic interne et le trafic de données de la passerelle EDP (Evolved Packet Data Gateway). La prise en charge des ports SR-IOV est disponible en neutron et permet aux machines virtuelles d'accéder au réseau via les fonctions virtuelles SR-IOV.	osp16_setIpLink.yml
Étape 14.	chien de garde	Au cours de cette étape, le paramètre IPMI sur le noeud directeur est configuré pour les tâches de gestion sur tous les serveurs via une connexion hors bande.	osp16_config_ipmi_watchdog.yml
Étape 15.	icedriver	Mettez à jour le pilote ICE Intel E810 pour cartes PCI (Peripheral Component Interconnect) vers la	osp16_ice_driver_install.yml

		version 1.12.6 pour EPDG afin d'utiliser les ports de carte réseau Intel comme SR-IOV.	
Étape 16.	redémarrez	Redémarrez tous les noeuds Overcloud après l'exécution des étapes précédentes.	osp16_reboot_overcloud_hosts.yml
Étape 17.	verifyrops	Vérifiez la configuration et l'intégrité du déploiement RHOSP.	osp16_rops_verify.yml

Pour le provisionnement des noeuds Overcloud, Undercloud utilise « overcloud-hardened-uefi-full.qcow2 ». Ainsi, avant de commencer le déploiement sur le cloud, l'image doit être stockée dans un chemin désigné dans le sous-cloud/directeur.

Copiez le fichier Overcloud qcow2 à partir du site distant.

<#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_NAM
```

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=verifyrhosp
```

Afin de surveiller les journaux de déploiement, utilisez le fichier journal le plus récent.

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

Assurez-vous que toutes les 17 étapes ont été effectuées.

Echec des vérifications = Le nombre doit être 00.

Journaux = /home/stack/autologs/osp16_rhosp_verify.yml _20200703T042257.log

#==========

ÉTAPE | BALISE | DESCRIPTION | GUIDE

#==========

étape 1 | genpodvars | Générer des fichiers de variables spécifiques à POD | osp16_generate_pod_specific_vars.yml -e podname=

étape2 | geninstack | Générer un fichier JSON Instackenv | osp16_generate_instackenv.yml -e podname=

étape3 | cimcvd | Configurer les versions virtuelles CIMC | osp16_cimc_vd_configure.yml

étape4 | preocdeploy | Configurer le déploiement pré-overcloud | osp16_pre_overcloud_deploy.yml

étape 5 | importnodes | Importer des noeuds ironiques sans système d'exploitation | osp16_import_ironic_nodes.yml

étape6 | genttemplate | Générer des modèles personnalisés | osp16_generate_custom_templates.yml

étape7 | codéployer | Déploiement Overcloud Openstack | osp16_overcloud_deploy.yml

étape8 | geninventory | Générer un fichier d'inventaire | osp16_build_inventory_v3.py —ipmipass

étape9 | hors ligne | Configurer Overcloud Repo hors connexion à partir du fichier TAR hors connexion | osp16_config_offline_repo.yml

étape10 | escrime | Post-déploiement de la MOP avant le redémarrage - Configuration de la délimitation | osp16_config_fencing.yml

étape11 | raidcache | Post-déploiement MOP avant redémarrage - Raid Cache et réglage PR | osp16_raid_cache_tuning.yml

étape12 | dnfupdate | Post-déploiement MOP avant redémarrage - Paquets de mise à jour Dnf | dnf_update_all_packages.yml

étape13 | setiplink | Post-déploiement MOP avant redémarrage - Activer la confiance de liaison IP VF | osp16_setIpLink.yml

étape14 | chien de garde | Post-déploiement MOP avant redémarrage - Config IPMI Watchdog | osp16_config_ipmi_watchdog.yml

étape15 | icedriver | Post Déploiement de MOP avant redémarrage - Mise à jour du pilote ICE E810 | osp16_ice_driver_install.yml

étape16 | redémarrer | Redémarrer tous les noeuds de surcloud | osp16_reboot_overcloud_hosts.yml

étape17 | verifyrhops | Vérification de la configuration et de l'intégrité du déploiement RHOSP | osp16_rops_verify.yml -e podname=

#==========

TOUS LES HÔTES ACCESSIBLES

==========

VÉRIFICATIONS EFFECTUÉES => 17

VÉRIFICATIONS RÉUSSIES => 17

ÉCHEC DES VÉRIFICATIONS => 0

==========

ÉTAT GLOBAL => RÉUSSI !!

==========

Après le déploiement réussi d'Overcloud, assurez-vous que le tableau de bord Horizon est accessible.

Accès au tableau de bord Horizon

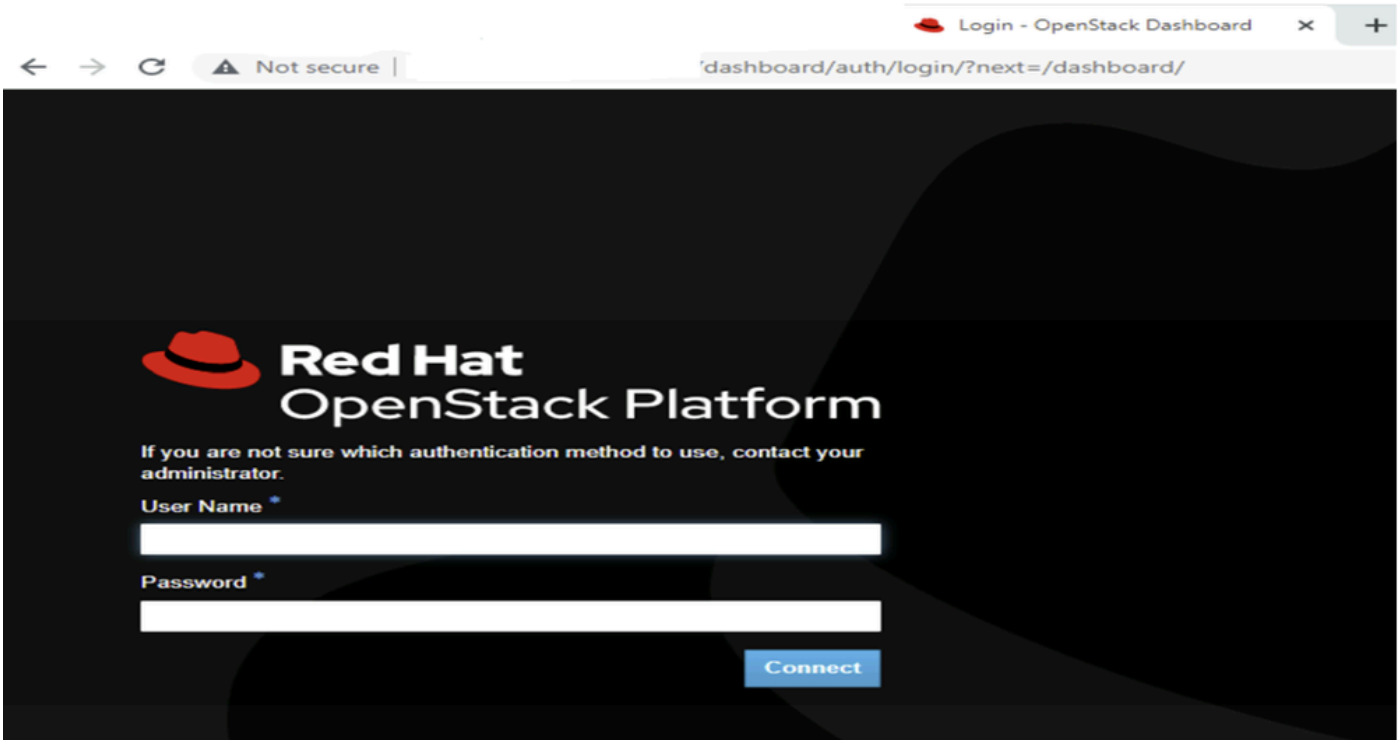
Pour l'URL du tableau de bord d'horizon, utilisez 'OS_AUTH URL'from 'overcloud'.

```
[stack@[redacted]-ospd ~]$ cat [redacted]rc
# 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=[redacted]
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
```

création de fichiers overcloud

Tableau de bord :



Vérification du fonctionnement du cluster RHOSP

<#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
```

Résumé

Le guide de déploiement RHOSP 16.2 fournit des instructions détaillées pour déployer un environnement cloud OpenStack évolutif et prêt à la production à l'aide des outils et méthodologies éprouvés de Red Hat. Ce guide est conçu pour les administrateurs système et les architectes cloud et se concentre sur le déploiement de RHOSP 16.2 à l'aide du directeur OpenStack, basé sur TripleO (OpenStack sur OpenStack).

Ce guide couvre toutes les phases critiques du déploiement, notamment :

- Planification de l'infrastructure et conditions préalables
- Préparation de l'environnement et configuration du réseau
- Installation et configuration sous le cloud
- Étapes de déploiement et de post-déploiement du cloud
- Options de haute disponibilité, de stockage et d'évolutivité des services

Ce guide est essentiel pour les équipes à la recherche d'une plate-forme cloud d'entreprise fiable, intégrant l'écosystème et prenant en charge Red Hat.

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.