

Bereitstellungsleitfaden von RHOSP auf dem UCS für Mobilitäts-Workloads im getrennten Modus

Inhalt

[Einleitung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Hintergrundinformationen](#)

[RHOSP im Überblick](#)

[RHOSP-Architektur](#)

[Unterwolke](#)

[Komponenten der Cloud](#)

[ÜberCloud](#)

[Controller](#)

[Computing](#)

[Storage](#)

[Lokales Repository \(REPO-Server\)](#)

[RHOSP-Netzwerk](#)

[Physische RHOSP-Verbindungen](#)

[Logische RHOSP-Verbindungen](#)

[Hardwareparameter-Optimierung](#)

[Hypervisor-KVM-Installation und Netzwerkerstellung](#)

[Erstellung von REPO und Director VM](#)

[Voraussetzungen für die Erstellung von REPO und Director VM](#)

[REPO VM-Erstellung](#)

[Erstellung von Director VMs](#)

[Lokale Offline-REPO-Erstellung](#)

[RHOSP-Cloud-Bereitstellung](#)

[Voraussetzungen](#)

[Aktualisierung von Eingabedateien](#)

[Bereitstellung in der Cloud](#)

[Overcloud-Bereitstellung](#)

[Zugriff auf das Horizon Dashboard](#)

[Statusprüfung des RHOSP-Clusters](#)

[Zusammenfassung](#)

Einleitung

In diesem Dokument wird ein Framework für die Bereitstellung von RHOSP auf C220 M6 UCS-Servern zur Unterstützung von Cisco VPC-DI beschrieben.

Voraussetzungen

Anforderungen

Cisco empfiehlt, dass Sie über Kenntnisse der Red Hat OpenStack Platform (RHOSP) verfügen und über fundierte Kenntnisse in Red Hat Enterprise Linux (RHEL) verfügen. Darüber hinaus ist ein fundiertes Verständnis der Virtualisierungs- und Netzwerkkonzepte erforderlich.

Verwendete Komponenten

Dieses Dokument ist nicht auf bestimmte Software- und Hardware-Versionen beschränkt.

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die möglichen Auswirkungen aller Befehle kennen.

Hintergrundinformationen

In diesem Leitfaden wird die Integration von RHOSP in die UCS-Infrastruktur (Unified Computing System) beschrieben. Der Schwerpunkt liegt dabei auf der Skalierbarkeit, Zuverlässigkeit und Leistungsoptimierung.

Es beschreibt Best Practices und nutzt eine ansible skriptbasierte Automatisierung für die Bereitstellung von OpenStack TripleO mit Undercloud- und Overcloud-Architektur.

Mit diesem Bereitstellungsleitfaden erhalten Unternehmen eine robuste und effiziente RHOSP-Cloud-Infrastruktur, die auf Cisco Virtual Packet Core - Distributed Instance (VPC-DI)-basierte Mobility Virtual Network-Funktionen (VNF) zugeschnitten ist.

RHOSP im Überblick

RHOSP ist eine Private Cloud-Lösung der Enterprise-Klasse, die auf dem Open-Source-OpenStack-Projekt basiert und von Red Hat integriert und unterstützt wird. Sie ermöglicht die

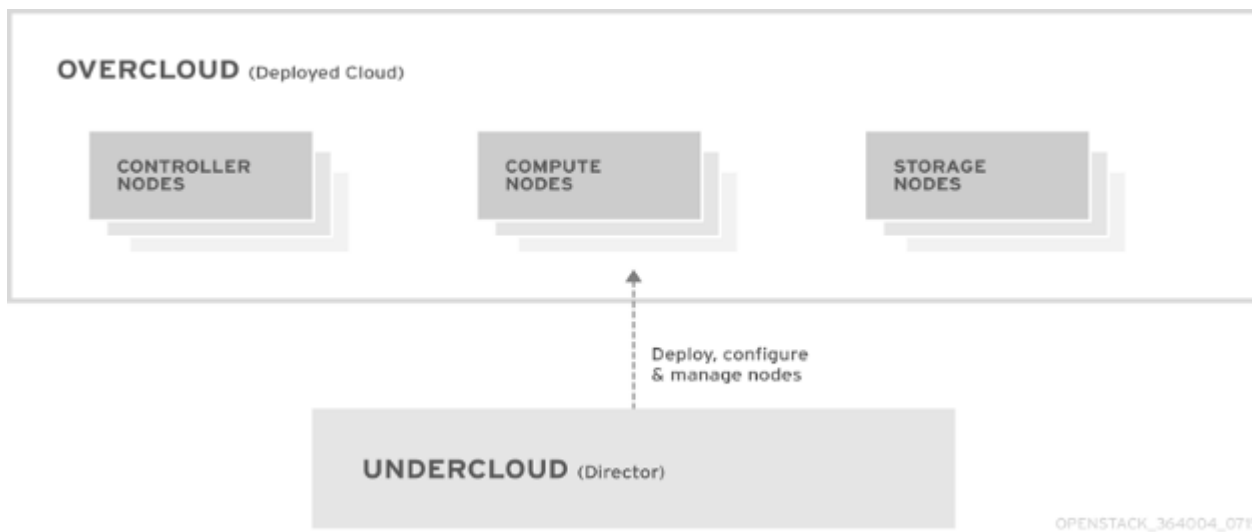
bedarfsgerechte Bereitstellung und Verwaltung von Infrastructure-as-a-Service (IaaS) für virtuelle Systeme, Netzwerke und Speicher.

Er bietet Funktionen wie Hochverfügbarkeit (HA), Virtualisierung von Netzwerkfunktionen und anpassbare Bereitstellungen.

RHOSP-Architektur

Das RHOSP basiert primär auf dem OpenStack TripleO Projekt. OpenStack verwendet Director, ein Toolset zur Installation und Verwaltung einer kompletten RHOSP-Umgebung.

RHOSP wurde entwickelt, um eine skalierbare und flexible Cloud-Infrastruktur bereitzustellen. Die Architektur besteht aus zwei Hauptkomponenten: Undercloud und Overcloud.



Unterwolke

Die Undercloud ist der Hauptmanagementknoten, der das RHOSP Director-Toolset enthält. Es handelt sich um eine OpenStack-Installation mit einem System, die Komponenten für die Bereitstellung und Verwaltung der OpenStack-Knoten umfasst, die die OpenStack-Umgebung (die Overcloud) bilden.

Komponenten der Cloud

Die Undercloud verwendet OpenStack-Komponenten als Basis-Toolset. Jede Komponente arbeitet in einem separaten Container in der Unterwolke:

- OpenStack Identity (Keystone) - Ermöglicht Authentifizierung und Autorisierung für die Director-Komponenten
- OpenStack Bare Metal (ironisch) - Verwaltung von Bare-Metal-Knoten
- OpenStack Networking (Neutronen) und Open vSwitch - Steuernetzwerke für Bare-Metal-Knoten
- OpenStack Orchestration (Ephemeral Heat) - Ermöglicht die Orchestrierung von Knoten, nachdem der Director das Overcloud-Image auf die Festplatte geschrieben hat

ÜberCloud

Die Overcloud ist die resultierende RHOSP-Umgebung, die mithilfe der Undercloud erstellt wird. Dies umfasst verschiedene Knotenrollen, die basierend auf der OpenStack Platform (OSP)-Umgebung definiert werden, die der Kunde erstellen möchte.

Controller

Controller-Knoten bieten Administration, Netzwerk und HA für die OpenStack-Umgebung. Eine empfohlene OpenStack-Umgebung umfasst drei Controller-Knoten in einem HA-Cluster.

Computing

Computing-Knoten stellen Computing-Ressourcen für die OpenStack-Umgebung bereit. Die Rechenknoten können je nach den Netzwerkanforderungen im Laufe der Zeit ein- und auskaliert werden. Ein Standardrechenknoten enthält die folgenden Komponenten:

- OpenStack Compute (Nova)
- Kernel-basiertes VM (KVM)/Quick Emulator (QEMU)
- Offener vSwitch

Storage

Storage-Knoten stellen Speicher für die OpenStack-Umgebung bereit.



Anmerkung: Es gibt mehrere Ansätze für die Bereitstellung von Undercloud/Director und Offline Repository (REPO) im Kundennetzwerk, die direkt auf Bare-Metal-Knoten oder als VM auf dem KVM-Hypervisor bereitgestellt werden können. Im aktuellen Bereitstellungsleitfaden hostet der Director UCS-Server KVM (Hypervisor), um mehrere

virtuelle Systeme im Vordergrund bereitzustellen. Der RHOSP Director-Knoten und die Offline-REPO-Knoten werden als VM auf dem KVM-Hypervisor bereitgestellt.

Lokales Repository (REPO-Server)

Redhat stellt ein Dienstprogramm namens reposync bereit, mit dem die Pakete vom [Content Delivery Network \(CDN\)](#) heruntergeladen werden können. Um alle Pakete von einem bestimmten Kanal herunterzuladen, muss das System diesen Kanal abonnieren. Wenn das System nicht für den erforderlichen Kanal abonniert ist, kann reposync diese Pakete nicht auf das lokale System herunterladen und synchronisieren.

Repositorys werden in `/etc/yum.repos.d/path` durch Dateien konfiguriert, die mit der Erweiterung `.repo` enden. Mehrere Repositories können in derselben Datei definiert werden.

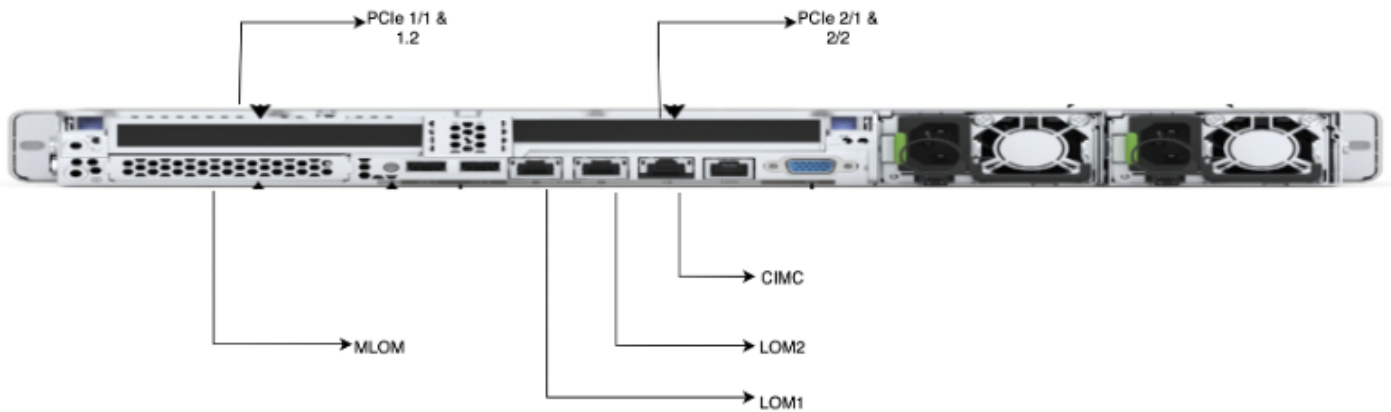
RHOSP-Netzwerk

Der Netzwerkservice (Neutron) ist die SDN-Komponente (Software-Defined Networking) von RHOSP. Der RHOSP-Netzwerkservice verwaltet den internen und externen Datenverkehr von und zu VM-Instanzen und stellt Kernservices wie Routing, Segmentierung, DHCP und Metadaten bereit. Sie stellt die API für virtuelle Netzwerkfunktionen und das Management von Switches, Routern, Ports und Firewalls bereit.

RHOSP Director ordnet OpenStack-Services verschiedenen isolierten Netzwerken zu. Die Netzwerke, die die einzelnen Datenverkehrstypen übertragen, sind: Cisco Integrated Management Controller (CIMC), Bereitstellung, interne API, Speicherdaten, Storage-Management, Tenant und extern (Secure Shell (SSH) und Operations, Administration and Maintenance (OAM)).

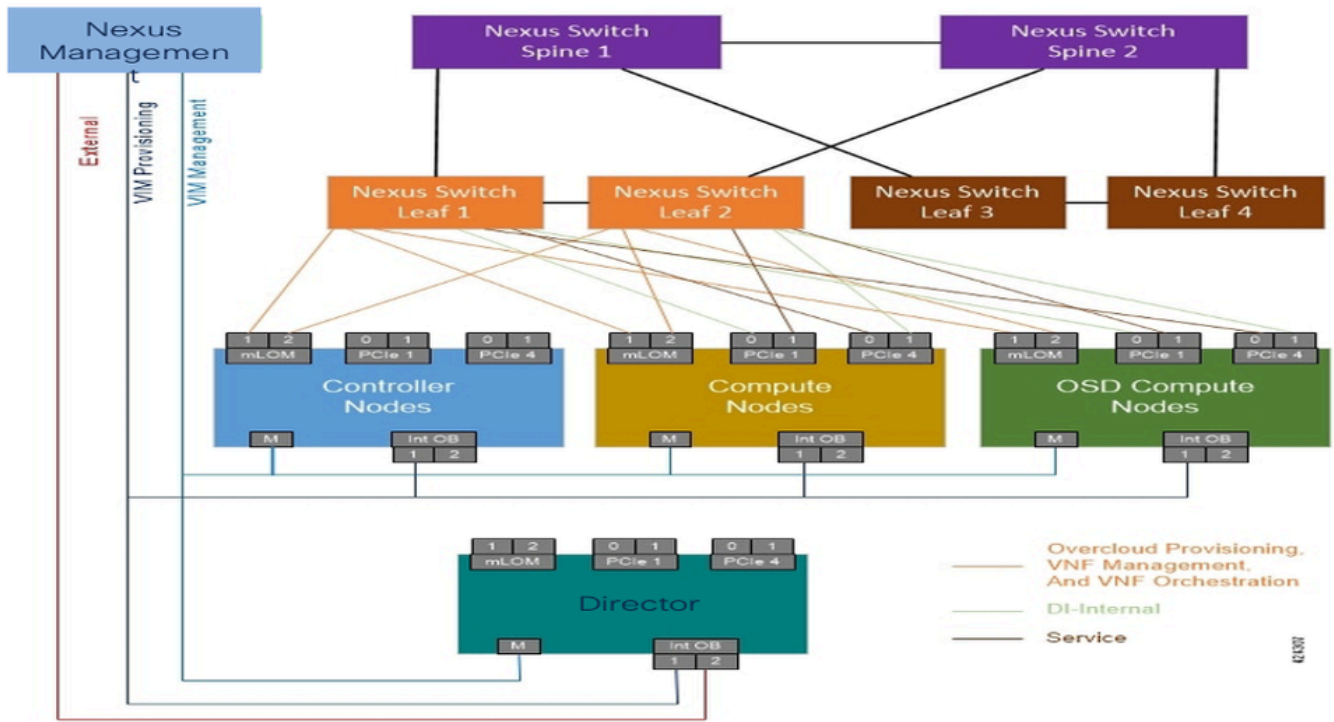
Physische RHOSP-Verbindungen

Bei der RHOSP-Bereitstellung werden unterschiedliche physische Ports der Cisco UCS C220 M6-Server für unterschiedliche Verbindungszwecke verwendet.



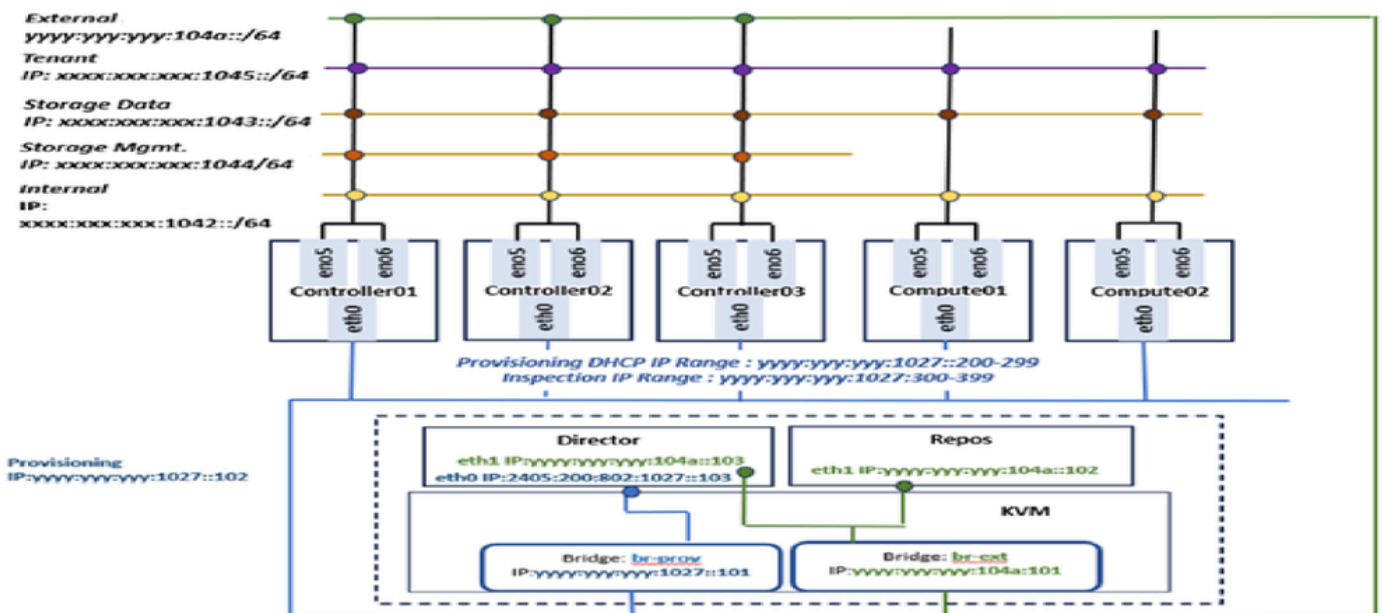
Seriennummer	Physische Ports	Details
1.	CIMC	CIMC bietet Out-of-Band-Verbindungen für die Serverbereitstellung und -verwaltung.
2.	Single Root I/O Virtualization (SR-IOV)/Peripheral Component Interconnect Express (PCIe)	Die PCIe-Netzwerkschnittstellenkarte (NIC) wird auf den Rechenknoten für die DI-internen Netzwerke und die Servicenetzwerke für die VNF verwendet.
3.	Modulares LAN auf Motherboard (MLOM)	MLOM-Ports werden als Bonds konfiguriert. osp_external, osp_internal, osp_tenant, osp_external, osp_storage_data, osp_storage_mgmt verwendet den MLOM-Port für die interne Kommunikation.
4.	Lan on MotherBoard (LOM)	Der Director verwendet LOM1- und LOM2-Ports, während Computing- und Controller-Ressourcen nur LOM1-Ports verwenden. LOM1 wird zur Bereitstellung oder Bereitstellung von OpenStack auf allen Servern verwendet. LOM2 wird als OAM (External Network) auf dem Director Switch verwendet.

Das Diagramm zeigt die physische Verbindung mit den Servern.



Logische RHOSP-Verbindungen

Das RHOSP-Netzwerk verfügt über mehrere Subnetze, die verschiedene Services innerhalb der Cloud verarbeiten.



- OSP_CIMC:

CIMC ist die Intelligent Programming Management Interface (IPMI), die die Verwaltung aller UCS-

Server steuert. Dieses CIMC-Netzwerk wird auf dem eigenständigen CIMC-Port aller UCS-Server konfiguriert.

- OSP_Bereitstellung:

Dieses Netzwerk ist für die Bereitstellung und die PXE-Bootverwaltung (Preboot Execution Environment) der Computer und Controller-Server während der Overcloud-Bereitstellung sowie für den Abruf der DHCP-IP zuständig. Das Provisioning-Netzwerk wird aus Gründen der Einfachheit und der Kompatibilität als natives VLAN auf dem LOM1-Port aller UCS-Server konfiguriert. Dieses Bereitstellungsnetzwerk ist für die Bereitstellung der Cloud auf allen Servern verantwortlich.

Aufgrund der Virtualisierung auf dem Director Server musste auf dem KVM ein Bridge-Netzwerk erstellt werden, damit der Director VM mit den anderen Servern kommunizieren kann.

- OSP_Intern:

Das interne API-Netzwerk wird für die Kommunikation zwischen den OpenStack-Services wie Neutron, Nova, Keystone usw. verwendet.

Das Netzwerk OSP_Internal wird auf verbundenen MLOM-Ports auf Controller- und Computing-Knoten konfiguriert.

- OSP_Tenant:

Das Tenant-Netzwerk wird standardmäßig im Rahmen von Cloud-Projekten für das VNF-Management erstellt. In der aktuellen Konfiguration wird nur ein OpenStack-Projekt für die VNF-Bereitstellung erstellt.

Das OSP_Tenant-Netzwerk wird auf verbundenen MLOM-Ports auf Controller- und Computing-Knoten konfiguriert.

- OSP_Extern:

Das externe Netzwerk wird für alle externen Zugriffe (wie SSH) und API-Netzwerke verwendet.

Das Netzwerk OSP_External wird auf dem LOM2-Port des Director-Knotens und auf verbundenen MLOM-Ports der Controller- und Computing-Knoten konfiguriert.

- OSP_Speicherdaten:

Das Netzwerk OSP_Storage wird für alle Vorgänge verwendet, die mit dem Zugriff auf den Speicher zusammenhängen. Dies ist für die Kommunikation zwischen dem CEPH-Service und VNF erforderlich, die auf den Speicher zugreifen müssen. Er wird von Controller, Rechenknoten und CEPH verwendet.

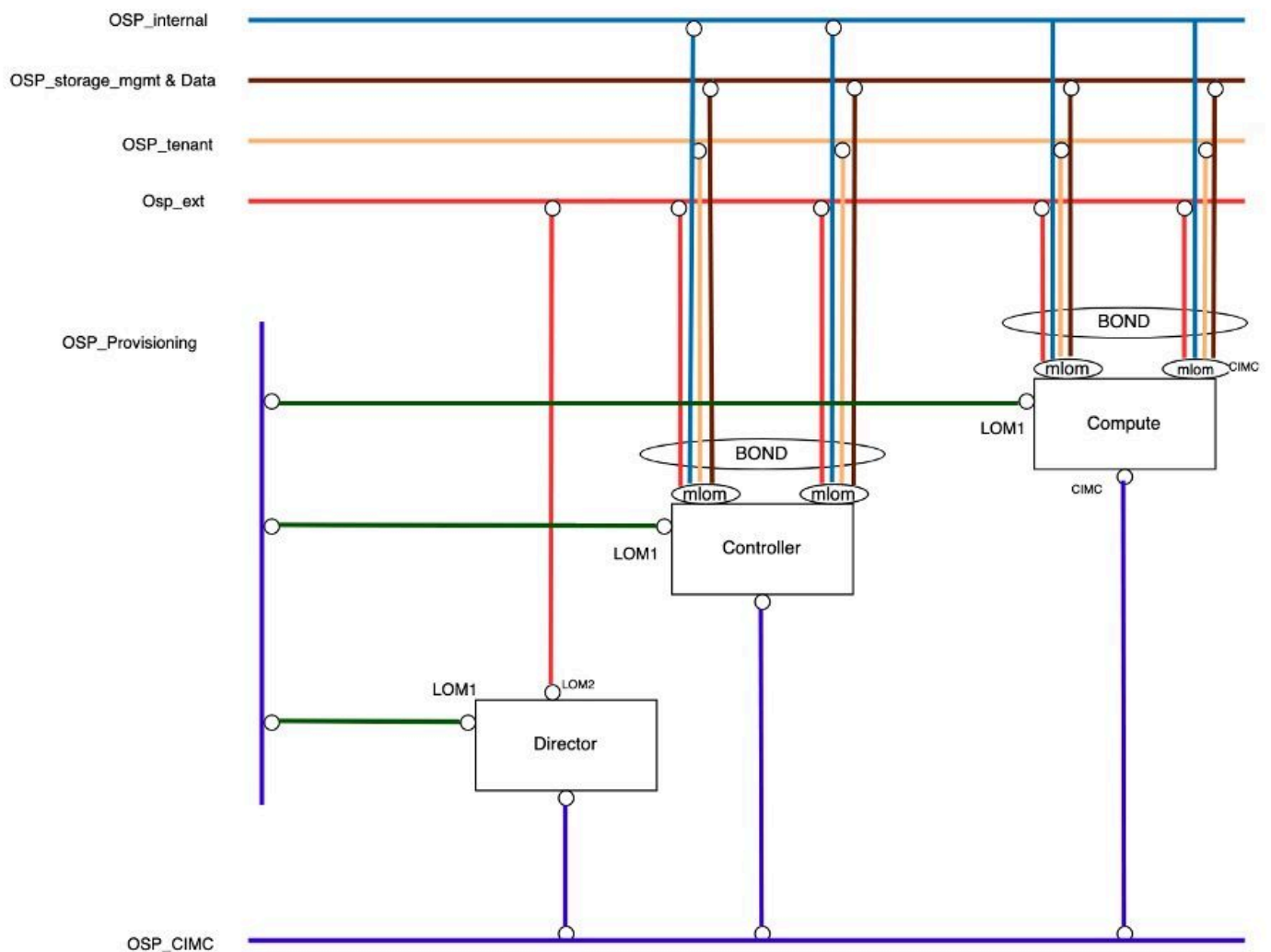
Das Netzwerk OSP_Storage_Data wird auf verbundenen MLOM-Ports auf Controller- und Computing-Knoten konfiguriert.

- OSP_Speicher_Management:

OpenStack Object Storage verwendet dieses Netzwerk zum Synchronisieren von Datenobjekten zwischen den beteiligten Replikatknoten im Speichercluster, die zwischen Controller-Compute-Knoten gebildet werden.

Das Netzwerk OSP_Storage_Mgmt wird auf verbundenen MLOM-Ports auf Controller- und Computing-Knoten konfiguriert.

Das Diagramm zeigt, wie die logischen Untercloud-Netzwerke mit den einzelnen Knotenarten im RHOSP-Cluster verbunden sind.



Hardwareparameter-Optimierung

- Stellen Sie sicher, dass IP aktiviert und mit der richtigen Management-Adresse gemäß Design konfiguriert ist.
- Stellen Sie sicher, dass der Admin-Benutzer und sein Kennwort für den gesamten CIMC-Anmeldezugriff identisch sein müssen.
- Stellen Sie sicher, dass der Bootstrap (Boot)-Modus auf Unified Extensible Firmware Interface (UEFI) eingestellt ist.
- Stellen Sie sicher, dass die BIOS-Einstellung mit den empfohlenen Parametern übereinstimmt.
- Bootreihenfolge festlegen auf:
 1. LOM-PXE (eno1)
 2. BOOT-Festplattenlaufwerk (HDD)
- Legen Sie die MAC-Adresse von LOM1 (eno1) fest.
- Stellen Sie sicher, dass alle UCS-/CIMC-Server mit der neuesten empfohlenen Firmware-Version aktualisiert wurden.

Hypervisor-KVM-Installation und Netzwerkerstellung

Es gibt verschiedene Methoden, um den Undercloud/Director- und Offline-REPO im Kundennetzwerk bereitzustellen. Diese können entweder direkt auf einem Bare-Metal-Knoten oder als VMs auf einem KVM-Hypervisor bereitgestellt werden.

Im vorliegenden Bereitstellungsleitfaden ist der Director UCS-Server für das Hosten des KVM-Hypervisors konfiguriert, wodurch die Erstellung mehrerer virtueller Systeme vereinfacht wird. Der RHOSP Director-Knoten und der Offline-REPO-Knoten werden als VMs auf diesem KVM-Hypervisor bereitgestellt.



Anmerkung: Die RHEL KVM-Standardinstallationsschritte müssen eingehalten werden, um den KVM-Hypervisor bereitzustellen.

-
- Sobald die KVM aktiv ist, aktualisieren Sie den Hostnamen:
`hostnamectl set-hostname <Hostname> —static`
 - Konfigurieren Sie die externen Bridges und Provisioning Bridges, und binden Sie die Schnittstellen.

br-prov: eth0

br-ext: eth1

Diese Bridges müssen über die Network Manager Text User Interface (NMTUI)-GUI erstellt werden.

- Standardmäßig werden eno1- und eno2-Ports in KVM für physische LOM1- und LOM2-Ports erstellt. Überprüfen Sie diese mithilfe der MAC-Adresse aus der CIMC-Protokollierung.
- Erstellen Sie ein Bridge-Netzwerk, und fügen Sie ihm Slave-Ports hinzu, indem Sie die richtige MAC-Adresse zuordnen.
- Stellen Sie nach der Erstellung des Bridge-Netzwerks sicher, dass die Bereitstellung und das externe Gateway über KVM erreichbar sind.

Erstellung von REPO und Director VM

Voraussetzungen für die Erstellung von REPO und Director VM

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

- Installieren Sie die erforderlichen Pakete in KVM für REPO und Director VM Creation.

```
! command not found...
[root@RHHEKVM01 /]#
[root@RHHEKVM01 /]# 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
```

- Erstellen Sie die erforderlichen Verzeichnisse auf KVM.

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

- Kopieren Sie die Dateien in die Verzeichnisse.

```
# 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/
```

- Einbinden des ISO in /mnt/iso.

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

- Erstellen Sie die REPO-Datei unter /etc/yum.repos.d.

```
# 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/
```

- Überprüfen Sie den Repolist, und stellen Sie sicher, dass die Appstream- und Baseos-REPOs zugeordnet sind.

```
# dnf repolist
```

```
[root@MUMBKVM01 ~]#
[root@MUMBKVM01 ~]#
[root@MUMBKVM01 ~]# 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@MUMBKVM01 ~]#
[root@MUMBKVM01 ~]#
[root@MUMBKVM01 ~]#
```

REPO VM-Erstellung

```
$ 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@MUMBKVM01 images]#
[root@MUMBKVM01 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@MUMBKVM01 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]#
```

- Erstellen Sie das Image des REPO-Servers.

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

```
[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
[  9.0] 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 299.9G. 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_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/OSP
```

```
[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/OSP_REPO_RHEL_84.qcow2
Formatting '/data/images/OSP_REPO_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/OSP_REPO_RHEL_84.qcow2 -i ln-sf /dev/null /etc/systemd/system/cloud-init.ser
```

```
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]# guestfish -a /data/images/OSPREPO_RHEL_84.qcow2 -t 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/1835/0100 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/OSPREPO_RHEL_84.qcow2

```
[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/1835/0100 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]#
```

Erstellung von Director VMs

- Erstellen Sie das Image des Director Servers.

```
# 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/images/rhel_84_ospdirector.qcow2
```

```
[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 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_ospdirector.qcow2 -F qcow2 /data/images/OSPDI
```

```
[root@MUMBKVMC01 images]#  
[root@MUMBKVMC01 images]# qemu-img create -f qcow2 -b /var/lib/libvirt/images/rhel_84_ospdirector.qcow2 -F qcow2 /data/images/OSPDI  
RHEL_84.qcow2  
Formatting '/data/images/OSPDI_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]#
```

```
# guestfish -a /data/images/OSPDI_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/OSPDI
```

```
[root@MUMBKVMC01 images]#  
[root@MUMBKVMC01 images]# virt-install --cpu host --memory 131072 --vcpus 32 --os-variant rhel8.4 --disk path=/data/images/OSPDI_RHEL_84.qcow2,device=disk,bus=virtio,format=qcow2 --import --noautoconsole --vnc --network bridge:br-prov --network bridge:br-ext --name OSPDI_RHEL_84  
** (process:49762): WARNING **: 17:15:52.000: Entity http://pcisig.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@MAGKVMC02 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/1836/0100 referenced but not defined
Starting install...
Retrieving file vmlinuz... | 9.6 MB 00:00:00
Retrieving file initrd.img... | 72 MB 00:00:00
Allocating 'OSPDIRECTORCL02_RHEL_84.qcow2' | 500 GB 00:00:00
(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'

```

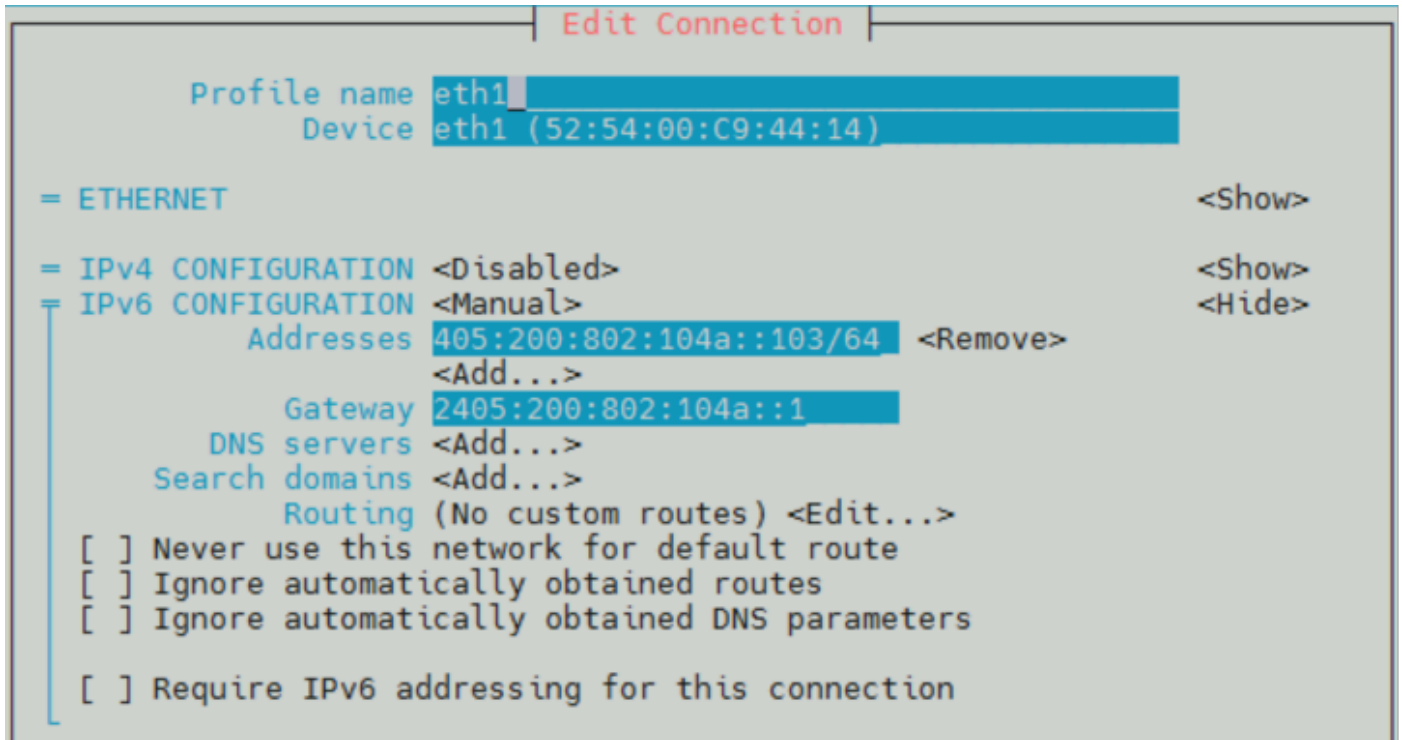
- Öffnen Sie ein neues Fenster, und melden Sie sich bei der VM über die Konsole an.

```

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

```

- Nach der Bereitstellung von Director VM werden zwei eth-Schnittstellen eth0 und eth1 erstellt.
 - eth0 wird für OSP_provisioning-Netzwerke verwendet, bei denen das br-ctlplane-Netzwerk während der Undercloud-Bereitstellung erstellt wird.
 - eth1 wird für externe Verbindungen für den SSH-Zugriff verwendet. Es wird also die IP-Adresse des OSP_external-Subnetzes zugewiesen.



Lokale Offline-REPO-Erstellung

REPO Server muss bei Redhat CDN registriert sein und über das Repository aller für die Bereitstellung erforderlichen verfügbaren Pakete von RHOSP 16.2 verfügen. RHEL RPM-Pakete und RHOSP-Container-Images müssen mithilfe eines Proxys auf REPO VM heruntergeladen werden.

RHOSP-Cloud-Bereitstellung

RHOSP 16.2 wird im Kundennetzwerk durch Automatisierung bereitgestellt. Transparente Skripte dienen zur Automatisierung der Bereitstellung in der Undercloud und in der Overcloud.

Voraussetzungen

Vor Beginn der eigentlichen Cloud-Bereitstellung zu befolgende Schritte:

1. Sicherstellung der Konnektivität des KVM-Knotens mit lokalem REPO VM und Director VM im OSP_EXT-Netzwerk.
2. Stellen Sie sicher, dass alle ansible Skripte und Automatisierungsskripte auf den KVM-Host im angegebenen Ordner hochgeladen werden.
3. Erstellen Sie Ordner mit Namen wie 'cisco' und 'automation' und legen Sie den Tarball von ansible Skripte.

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

Der Tarball besteht aus drei Ordnerverzeichnisstrukturen mit dem Namen:

- Skripte: besteht aus Shell-Skripten, die für die Bare-Metal-Knotenkonfiguration verwendet werden.
- RPM: besteht aus Intel Ethernet Connection (ICE) Treibern, Intel Treibern und Red-Hat Package Manager (RPM) Paketen.
- Zuständig: besteht aus Input vars-Datei, ansible yaml-Dateien für die Bereitstellung und jinja-Templates.

4. Installieren Sie sshpass package. sshpass ist ein Befehlszeilenprogramm, mit dem Passwörter für ssh ohne Interaktivität bereitgestellt werden. Es wird hauptsächlich in Skript- oder Automatisierungsszenarien verwendet, in denen eine manuelle Passworteingabe nicht möglich ist.

- Laden Sie das sshpass-Paket sshpass.tar.gz vom Internet/bestehenden Server herunter.
- Installieren Sie das GNU Compiler Collection (GCC)-Paket.

```
# yum install gcc
```

- Installieren von make-Paketen.

```
# yum install make
```

- Deaktivieren Sie das sshpass-Paket, und installieren Sie SSH pass.

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

5. Der Director-Installationsprozess erfordert, dass ein Benutzer ohne Root Befehle ausführt. Stack-Benutzer muss in Director VM mit Sudo-Zugriff erstellt werden.

```
# 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. Kopieren Sie die Datei rootCA.crt vom REPO-Server auf die VM Director und KVM unter dem angegebenen Pfad. Aktualisieren Sie außerdem das Zertifikat des REPO VM in der Vertrauensliste.

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

7. Aktualisieren Sie die Details des lokalen REPO-Server-Hostnamens in Director VM und KVM in /etc/hosts-Datei.

8. Installieren Sie auf KVM und Director VM zusätzliche Pakete wie python, ansible usw., um ansible Automatisierungsskripts auszuführen.

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

9. Das CIMC-Subnetz muss über das Provisioning-Netzwerk von Director erreichbar sein, um die Bereitstellung während der Cloud-Bereitstellung zu ermöglichen. Falls erforderlich, fügen Sie für dieselbe statische Route hinzu.

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

10. Erstellen Sie in KVM und Director VM eine Host-Datei unter dem Ordner /Ansible, und fügen Sie bei Bedarf stapelspezifische Details hinzu.

```
[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. Stellen Sie sicher, dass alle verfügbaren Playbooks und Eingabedateien im Director VM im Ordner /home/stack aufbewahrt werden müssen.

Aktualisierung von Eingabedateien

Es gibt eine Datei mit einer Eingangsvariablen, die aus kundennetzwerkspezifischen Details besteht und für die Cloud-Bereitstellung vorbereitet werden muss.

Pfad: /home/cisco/automation/ansible/podvars

Dateiname: <Stapelname>_vars.yml

Aktualisieren Sie die hervorgehobenen Parameter gemäß dem standortspezifischen IP-Plan/Low-Level-Design-Dokument.



Anmerkung: Dummy-IP-Adressen werden nur für Darstellungszwecke verwendet.

<#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::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
# #####

```

Bereitstellung in der Cloud

Undercloud wird mithilfe von ansible Scripts in sieben Schritten bereitgestellt. Alle Schritte müssen von einem KVM-Host ausgeführt werden, der in diesem Fall als Jump-Host fungiert.

Schritt	Tag	Beschreibung	Strategischer Leitfaden YAML
---------	-----	--------------	------------------------------

Schritt 1:	Prüfdateien	Überprüfen der erforderlichen strategischen Leitfäden, Skripte und RPMs auf OpenStack Platform Director (OSPD)	osp16_published_playbooks_verify.yml
Schritt 2:	Genpodvars	Generieren Sie POD-spezifische Variablendateien in Bezug auf Hardware, RHEL usw., wie common_vars.yml (Hardware, Software, Netzwerkdetails), hw_m6_vars.yml (CPU, Speicher, Webseiten, Festplatte, NIC usw.), rhel_84_vars.yml (RHEL, Kernel, ICE Treiber, NIC-Version), pf_esc_vars.yml (Details zum Elastic Services Controller (ESC)), osp_16_vars.yml (OSP-Version, Zeitzone, IP-Typ, VLAN-ID, IP, Neutronendetails).	osp16_generate_pod_specific_vars.yml
Schritt 3:	vorbereiten	Konfigurieren von FQDN (Fully Qualified Domain Name) und NTP (Network Time Protocol) und Aktualisieren aller Pakete auf dem Director-Knoten	osp16_pre_undercloud_deploy.yml
Schritt 4:	firstreboot	Führt nach der vorherigen Konfiguration und Paketinstallation einen ersten Neustart des Undercloud Director-Knotens durch.	osp16_undercloud_deploy.yml
Schritt 5:	UCStreib	Installation des Undercloud-Stacks auf Director	osp16_undercloud_tuning.yml
Schritt 6:	Cstate	Konfigurieren Sie die BIOS-CPU-C-State-Einstellungen auf dem Director-Knoten.	osp16_cstate.yml
Schritt 7:	Sekundenstart	Nach BIOS-Änderungen zweiter Neustart auf dem Undercloud-	-

		Director.	
--	--	-----------	--

Die Datei mit dem Namen osp16_auto_undercloud_deploy.yml ist ein praktisches Handbuch, das in einer Iteration ausgeführt werden kann. Es ist jedoch ratsam, das Handbuch schrittweise mit verschiedenen Tags auszuführen, um die Fehlerbehebung bei Bereitstellungsproblemen zu vereinfachen.

```
<#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
```

Overcloud-Bereitstellung

Overcloud wird mit mindestens drei Controllern im HA-Modus und einem Computing bereitgestellt. Overcloud wird mithilfe von ansible Scripts in 17 Schritten bereitgestellt. Alle Schritte müssen von Director-VM ausgeführt werden, die in diesem Fall als Jump-Host fungiert.

Schritt	Tag	Beschreibung	Strategischer Leitfaden YAML
Schritt 1:	Genpodvars	Generieren Sie POD-spezifische Variablendateien für Overcloud in Bezug auf Hardware, RHEL usw., wie <code>common_vars.yml</code> (Hardware, Software, Netzwerkdetails), <code>hw_m6_vars.yml</code> (CPU, Speicher, Webseiten, Festplatte, NIC usw.), <code>rhel_84_vars.yml</code> (RHEL, Kernel, ICE-Treiber, NIC-Version), <code>pf_esc_vars.yml</code> (ESC-Details), <code>osp_16_vars.yml</code> (OSP-Version, Zeitzone, IP-Typ, VLAN-ID, IP, Neutronendetails).	<code>osp16_generate_pod_specific_vars.yml</code>
Schritt 2:	Geninstack	Generieren Sie Instackenv JSON-Datei von <code>/var/common_vars.yml</code> erstellt im vorherigen Schritt. Der Director benötigt eine Knotendefinitionsvorlage, die manuell erstellt wird. Diese Datei <code>instackenv.json</code> verwendet das JSON-Format und enthält alle Hardware- und Energiemanagement-Details für die Knoten. In diesem Schritt wird auch die Hardwarekonfiguration auf dem UCS-Server validiert, bevor die Datei generiert wird.	<code>osp16_generate_instackenv.yml</code>
Schritt 3:	cimc vd	Konfigurieren Sie die CIMC-	<code>osp16_cimc_vd_configure.yml</code>

		Einstellung und virtuelle Laufwerke (Virtual Disks, VDs) auf jedem Server, der auf <code>common_vars.yml</code> , <code>hw_m6_vars.yaml</code> und <code>rhel_84_vars.yml</code> verweist.	
Schritt 4:	vorkodieren	In diesem Schritt werden alle Voraussetzungen für die Bereitstellung der Overcloud erfüllt. Es legt den FQDN, NTP fest und aktualisiert alle Pakete. Push-Abbild wird an den Pfad für die Bereitstellung weitergeleitet.	<code>osp16_pre_overcloud_deploy.yml</code>
Schritt 5	Importknoten	In diesem Schritt werden die Server-CPU, der Arbeitsspeicher, die Netzwerkkarte sowie die Schnittstelle und die Ports an den Netzwerk-Switches eingebunden. Die angeschlossenen Netzwerk-Switches werden für alle Controller und Computing-Ressourcen einer Introspektion unterzogen.	<code>osp16_import_ironic_nodes.yml</code>
Schritt 6:	Gentemplates	Generieren Sie benutzerdefinierte Vorlagendateien für Controller und Computer. Definieren Sie in der benutzerdefinierten Vorlage Controller- und Computing-Rollen für alle darauf ausgeführten Dienste. Außerdem wird das System gehärtet, indem Zertifikate, Routen usw. angewendet werden.	<code>osp16_generate_custom_templates.yml</code>
Schritt 7.	kodisponieren	In diesem Schritt wird eine	<code>osp16_overcloud_deploy.yml</code>

		OpenStack Overcloud-Bereitstellung durchgeführt. Führen Sie <code>deploy.sh</code> aus, das von Red Hat für die RHOSP-Bereitstellung bereitgestellt wird.	
Schritt 8	Geninventar	In diesem Schritt wird eine Inventar-MML-Datei zur Verwendung durch Ansible generiert, in der Provisioning-IP, IPMI (CIMC)-IP und Anmeldeinformationen gespeichert und dem Controller zugeordnet werden. Außerdem wird eine automatische Anmeldung und Ausführung der weiteren Schritte im System berechnet.	<code>osp16_build_Inventory_v3.py</code>
Schritt 9.	Offlinerepo	Konfigurieren Sie Overcloud Offline REPO in der Datei <code>/etc/yum.repo.d/offline.repo</code> verweisen Sie über das externe Netzwerk auf den Repo-Server.	<code>osp16_config_offline_repo.yml</code>
Schritt 10.	Einzäunung	Konfigurieren Sie Fencing auf allen Controller-Knoten mit Shoot The Other Node In The Head (eine Fencing-Technik in HA-Clustern) (STONITH).	<code>osp16_config_fencing.yml</code>
Schritt 11.	RAID-Cache	Konfigurieren Sie die RAID-Cache-Einstellung für alle Controller, und konfigurieren Sie die SWIFT-Speichereinstellungen.	<code>osp16_raid_cache_tuning.yml</code>
Schritt 12:	<code>dnfupdate</code>	DNF-Updates für alle Pakete auf allen Knoten ausführen.	<code>dnf_update_all_packages.yml</code>
Schritt 13:	Setiplink	In diesem Schritt wird die	<code>osp16_setlpLink.yml</code>

		Steuerung des Vertrauensmodus für SR-IOV-Ports für den internen und Datenverkehr des Evolved Packet Data Gateway (EPDG) aktiviert. Die Unterstützung für SR-IOV-Ports ist neutronenbasiert und ermöglicht VMs den Zugriff auf das Netzwerk über virtuelle SR-IOV-Funktionen.	
Schritt 14:	Wachhund	In diesem Schritt wird die IPMI-Einstellung auf dem Director-Knoten für Managementaufgaben auf allen Servern über Out-of-Band-Verbindungen konfiguriert.	osp16_config_ipmi_watchdog.yml
Schritt 15:	Eislaufstaucher	Aktualisieren Sie den Intel E810 ICE-Treiber für Peripheral Component Interconnect (PCI)-Karten auf Version 1.12.6 für EPDG, um Intel NIC-Ports als SR-IOV zu verwenden.	osp16_ice_driver_install.yml
Schritt 16:	Starten Sie erneut	Starten Sie alle Overcloud-Knoten nach der Ausführung der früheren Schritte neu.	osp16_reboot_overcloud_hosts.yml
Schritt 17:	Verifyrhop	Überprüfen der Konfiguration und des Zustands der RHOSP-Bereitstellung	osp16_rhosp_verify.yml

Zur Bereitstellung von Overcloud-Knoten verwendet Undercloud "overcloud-hardened-uefi-full.qcow2". Vor Beginn der Overcloud-Bereitstellung muss das Image daher unter einem festgelegten Pfad in der Undercloud/im Director gespeichert werden.

Kopieren Sie die Datei Overcloud qcow2 vom Remote-Standort.

<#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=verifyrhosp
```

Verwenden Sie die neueste Protokolldatei, um die Bereitstellungsprotokolle zu überwachen.

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

Stellen Sie sicher, dass alle 17 Schritte erfolgreich durchgeführt wurden.

Fehler bei Prüfungen => Der Wert muss 00 sein.

Protokolle => /home/stack/autologs/osp16_rhosp_verify.yml_20200703T042257.log

```
#####
```

```
# SCHRITT | TAG          | BESCHREIBUNG                                | STRATEGISCHER
LEITFADEN
```

```
#####
```

```
# Schritt 1 | genpodvars    | POD-spezifische Variablendateien generieren          |
osp16_generate_pod_specific_vars.yml -e podname=
```

```
# Schritt 2 | Geninstack    | Instackenv JSON-Datei generieren                    |
osp16_generate_instackenv.yml -e podname=
```

```
# Schritt 3 | cimc vd       | CIMC-VDs konfigurieren                              |
osp16_cimc_vd_configure.yml
```

```
# Schritt 4 | preocdeploy   | Konfiguration vor der Overcloud-Bereitstellung      |
osp16_pre_overcloud_deploy.yml
```

```
# Schritt 5 | Importknoten  | OpenStack Baremetal Ironic-Knoten importieren      |
osp16_import_ironic_nodes.yml
```

```
# Schritt 6 | Gentemplates  | Benutzerdefinierte Vorlagen generieren              |
```

osp16_generate_custom_templates.yml

Schritt 7 | codeploy | OpenStack Overcloud-Bereitstellung |
osp16_overcloud_deploy.yml

Schritt8 | generischer Bestand | Inventardatei generieren |
osp16_build_Inventory_v3.py —ipmipass

Schritt 9 | Offlinerepo | Overcloud Offline-Repo aus Offline-TAR-Datei konfigurieren |
osp16_config_offline_repo.yml

Anzahl Schritt 10 | Zaun | POST-BEREITSTELLUNGSMOPP VOR Reboot - Konfigurieren der
Umzäunung | osp16_config_fencing.yml

Anzahl Schritt 11 | RAID-Cache | POST-Bereitstellung MOPP vor Neustart - RAID-Cache und
PR-Optimierung | osp16_raid_cache_tuning.yml

Anzahl Schritt 12 | dnfupdate | Bereitstellen - MOPP vor dem Neustart - DNF-
Aktualisierungspakete | dnf_update_all_packages.yml

Anzahl Schritt 13 | setiplink | POST-BEREITSTELLUNGSMOPP VOR Reboot - VF IP Link
Trust AUF | osp16_setIpLink.yml

Anzahl Schritt 14 | Watchdog | POST-Bereitstellung MOPP vor Neustart - IPMI-Watchdog
konfigurieren | osp16_config_ipmi_watchdog.yml

Anzahl Schritt 15 | EisFluss | POST-Bereitstellung MOPP vor Neustart - E810 ICE-Treiber
aktualisieren | osp16_ice_driver_install.yml

Anzahl Schritt 16 | Neustart | Alle Overcloud-Knoten neu starten |
osp16_reboot_overcloud_hosts.yml

Anzahl Schritt 17 | verifyrhofs | RHOSP-Bereitstellungskonfiguration und -status überprüfen
| osp16_rhosp_verify.yml -e podname=

#=====

ALLE HOSTS ERREICHBAR

=====

ABGESCHLOSSENE PRÜFUNGEN => 17

BESTANDTE PRÜFUNGEN => 17

FEHLER BEI PRÜFUNGEN => 00

=====

GESAMTSTATUS => BESTANDEN !!

=====

Nach der erfolgreichen Bereitstellung von Overcloud muss der Zugriff auf das Horizon Dashboard sichergestellt sein.

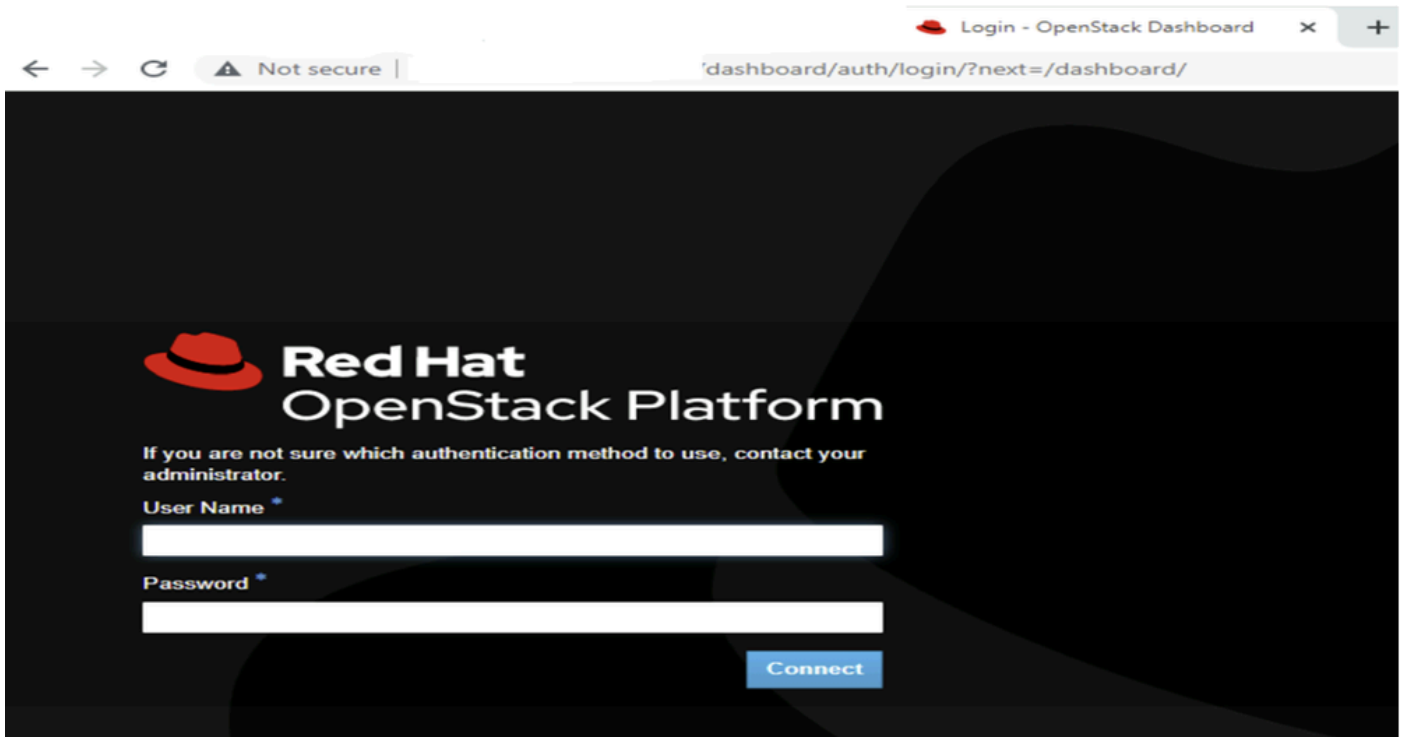
Zugriff auf das Horizon Dashboard

Verwenden Sie für die Horizon Dashboard-URL 'OS_AUTH_URL' von 'overcloudrc'.

```
[stack@MUMBMUMBTCUDR201C0-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='[,.]'
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 Dashboard:



Statusprüfung des RHOSP-Clusters

<#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
```

Zusammenfassung

Der RHOSP 16.2-Bereitstellungsleitfaden enthält umfassende, schrittweise Anleitungen für die Bereitstellung einer skalierbaren und produktionsbereiten OpenStack Cloud-Umgebung unter Verwendung der bewährten Tools und Methoden von Red Hat. Dieser Leitfaden richtet sich an Systemadministratoren und Cloud-Architekten und konzentriert sich auf die Bereitstellung von RHOSP 16.2 mit dem OpenStack Director, der auf TripleO (OpenStack on OpenStack) basiert.

Der Leitfaden deckt alle kritischen Phasen der Bereitstellung ab, darunter:

- Infrastrukturplanung und Voraussetzungen
- Umgebungsvorbereitung und Netzwerkkonfiguration
- Installation und Konfiguration unter Cloud
- Schritte zur Übercloud-Bereitstellung und danach
- Optionen für HA, Storage und Service-Skalierung

Dieser Leitfaden ist von entscheidender Bedeutung für Teams, die eine zuverlässige Cloud-Plattform der Enterprise-Klasse mit der Integration und Unterstützung von Red Hat in das Partnernetzwerk benötigen.

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.