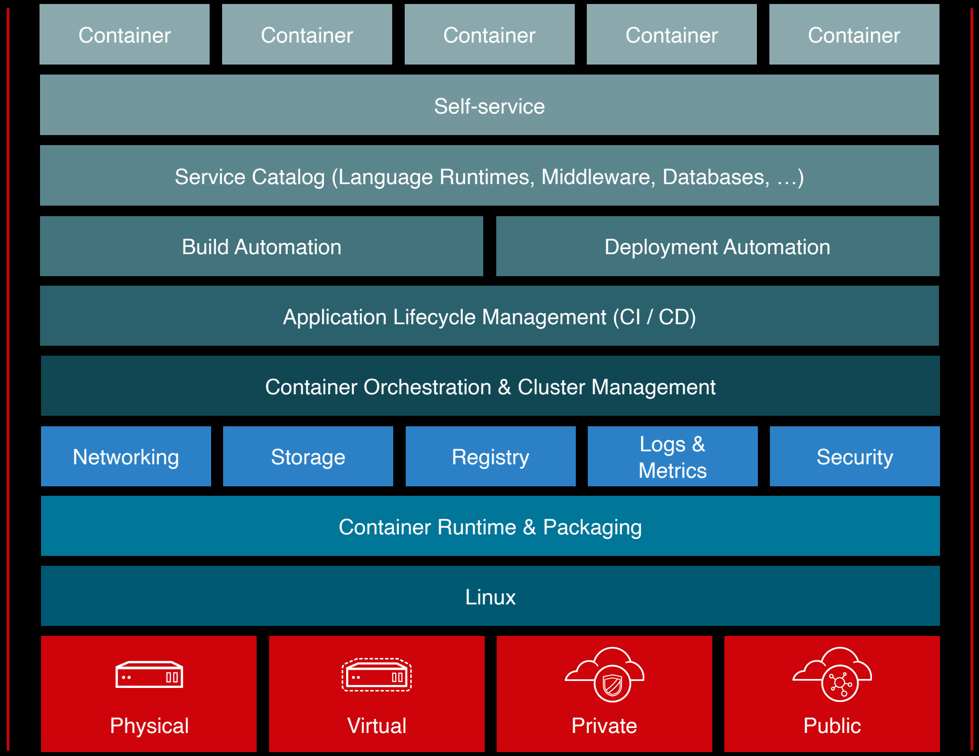
Red Hat OpenShift Container Platform

On IBM Z and LinuxONE

KVM Edition

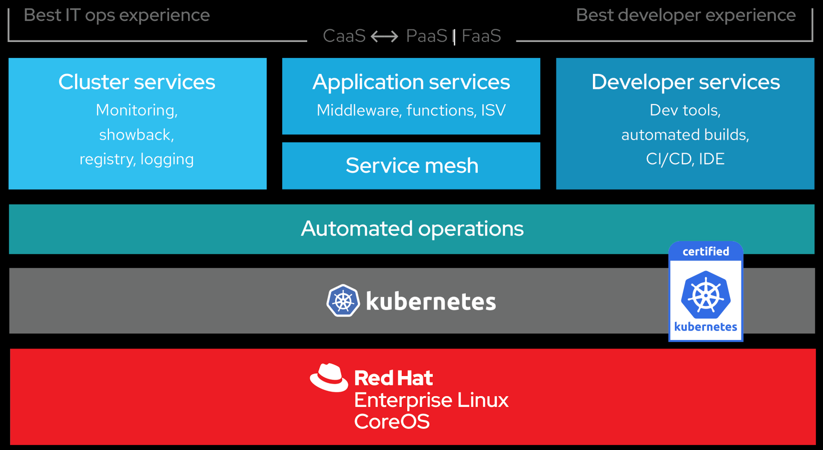
Filipe Miranda
Cloud Solutions Architect Leader, zAcceleration
Red Hat Synergy
fmiranda@ibm.com

Clients are facing DIY challenges



Container
Container
Container
Container
Container
Self-service
Service Catalog (Language Runtimes, Middleware, Databases, …)
Build Automation
Deployment Automation
Application Lifecycle Management (CI / CD)
Container Orchestration & Cluster Management
Networking
Storage
Registry
Logs & Metrics
Security
Container Runtime & Packaging
Linux
Physical
Virtual
Private
Public

Red Hat OpenShift Available on IBM Z and LinuxONE



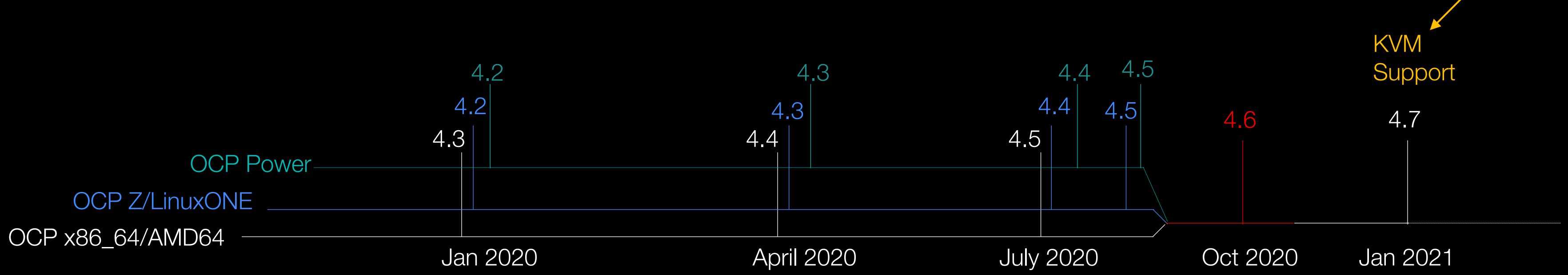
Best IT ops experience
Best developer experience
CaaS ⟷ PaaS | FaaS
Cluster services
Monitoring, showback, registry, logging
Application services
Middleware, functions, ISV
Service mesh
Developer services
Dev tools, automated builds, CI/CD, IDE
Automated operations
kubernetes
certified kubernetes
Red Hat Enterprise Linux CoreOS

Automated, full-stack installation from the container host to application services

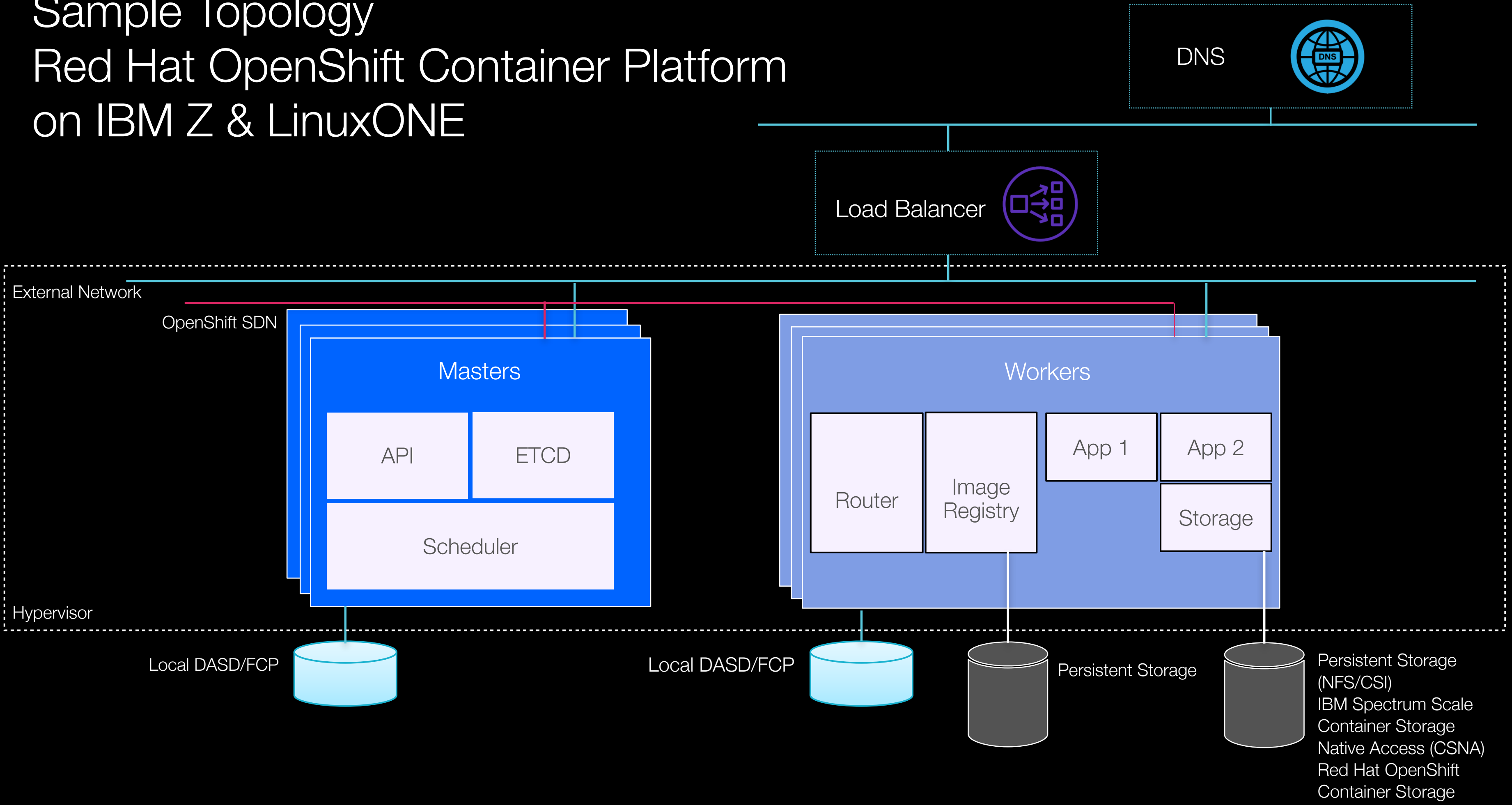
Seamless Kubernetes deployment to any cloud or on-premises environment

Autoscaling of cloud resources

One-click updates for platform, services, and applications

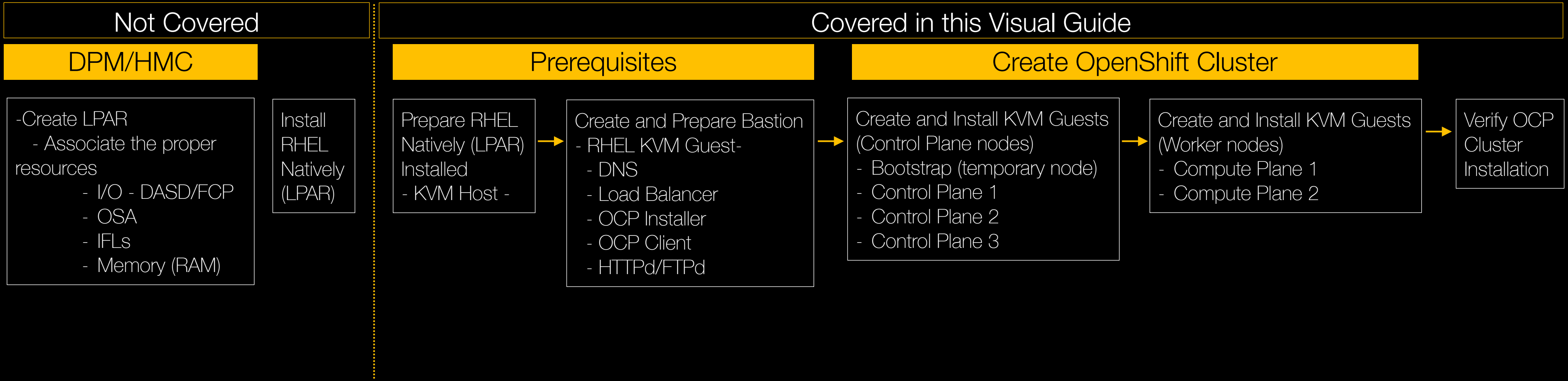


Sample Topology
Red Hat OpenShift Container Platform on IBM Z & LinuxONE



Red Hat OpenShift Container Platform Install Process

KVM



Not Covered

DPM/HMC

-Create LPAR
- Associate the proper resources
  - I/O - DASD/FCP
  - OSA
  - IFLs
  - Memory (RAM)

Install RHEL Natively (LPAR)

Covered in this Visual Guide

Prerequisites

Prepare RHEL Natively (LPAR) Installed
- KVM Host -

→

Create and Prepare Bastion
- RHEL KVM Guest-
  - DNS
  - Load Balancer
  - OCP Installer
  - OCP Client
  - HTTPd/FTPd

→

Create OpenShift Cluster

Create and Install KVM Guests (Control Plane nodes)
- Bootstrap (temporary node)
- Control Plane 1
- Control Plane 2
- Control Plane 3

→

Create and Install KVM Guests (Worker nodes)
- Compute Plane 1
- Compute Plane 2

→

Verify OCP Cluster Installation

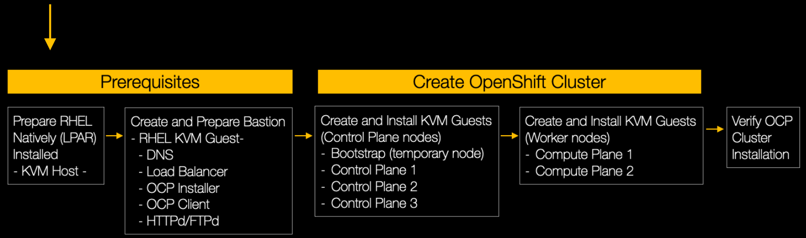
Red Hat OpenShift Container Platform Install Process

KVM

Red Hat Enterprise Linux (KVM Host)

Logical Partition (LPAR)

KVM Support installation



Prerequisites

Create OpenShift Cluster

Prepare RHEL Natively (LPAR) Installed - KVM Host -

Create and Prepare Bastion - RHEL KVM Guest-
- DNS
- Load Balancer
- OCP Installer
- OCP Client
- HTTPd/FTPd

Create and Install KVM Guests (Control Plane nodes)
- Bootstrap (temporary node)
- Control Plane 1
- Control Plane 2
- Control Plane 3

Create and Install KVM Guests (Worker nodes)
- Compute Plane 1
- Compute Plane 2

Verify OCP Cluster Installation

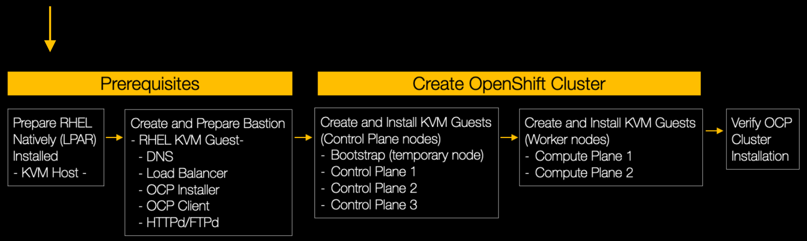
1 Make sure you make these packages installed and that libvirtd is started:

```
# yum install libvirt libvirt-devel libvirt-daemon-kvm qemu-kvm virt-manager libvirt-daemon-config-network
libvirt-client qemu-img
# systemctl enable --now libvirtd
# systemctl status libvirtd.service
```

2

```
# systemctl status libvirtd

● libvirtd.service - Virtualization daemon

   Loaded: loaded (/usr/lib/systemd/system/libvirtd.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2020-12-17 22:26:01 EST; 2 months 19 days ago
     Docs: man:libvirtd(8)
           https://libvirt.org

 Main PID: 933606 (libvirtd)
    Tasks: 18 (limit: 32768)
   Memory: 68.9M
   CGroup: /system.slice/libvirtd.service
           └─933606 /usr/sbin/libvirtd --timeout 120
```

Host KVM Network Preparation



Prerequisites

Create OpenShift Cluster

Prepare RHEL Natively (LPAR) Installed
- KVM Host -

Create and Prepare Bastion
- RHEL KVM Guest-
- DNS
- Load Balancer
- OCP Installer
- OCP Client
- HTTPd/FTPd

Create and Install KVM Guests (Control Plane nodes)
- Bootstrap (temporary node)
- Control Plane 1
- Control Plane 2
- Control Plane 3

Create and Install KVM Guests (Worker nodes)
- Compute Plane 1
- Compute Plane 2

Verify OCP Cluster Installation

1 Let's find details about our network interface:

```
…

3: enc4100: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 7e:4c:52:67:a6:f0 brd ff:ff:ff:ff:ff:ff
    inet <IP_ADDRESS>/24 brd <BROADCAST> scope global noprefixroute enc4100
      valid_lft forever preferred_lft forever

    inet6 fe80::7c4c:52ff:fe67:a6f0/64 scope link
       valid_lft forever preferred_lft forever
…
```

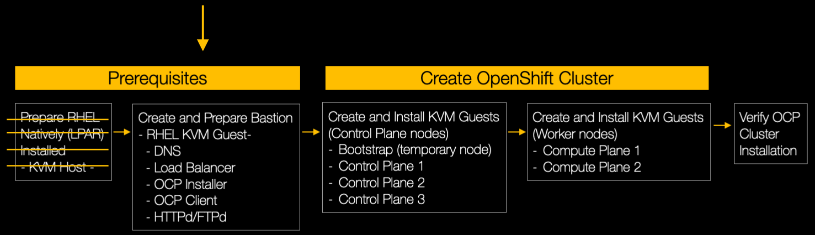
2 Create this xml file:

```
<network>
  <name>macvtap-net</name>
  <forward dev='enc4100' mode='bridge'>
    <interface dev='enc4100'/>
  </forward>
</network>
```

3 Set up a bridge to act as a macvtap interface to the network:

```
# virsh net-create macvtap.xml
# virsh net-start --network macvtap-net
# virsh net-autostart --network macvtap-net
# virsh net-list --all
```

Create and Configure Bastion



Prerequisites

Create OpenShift Cluster

~~Prepare RHEL Natively (LPAR) Installed - KVM Host -~~

Create and Prepare Bastion
- RHEL KVM Guest-
  - DNS
  - Load Balancer
  - OCP Installer
  - OCP Client
  - HTTPd/FTPd

Create and Install KVM Guests (Control Plane nodes)
- Bootstrap (temporary node)
- Control Plane 1
- Control Plane 2
- Control Plane 3

Create and Install KVM Guests (Worker nodes)
- Compute Plane 1
- Compute Plane 2

Verify OCP Cluster Installation

1 Download the RHEL ISO image to your RHEL KVM:

```
# wget /URL/rhel-8.3-s390x-dvd.iso
# mv rhel-8.3-s390x-dvd.iso rhel83.iso
```

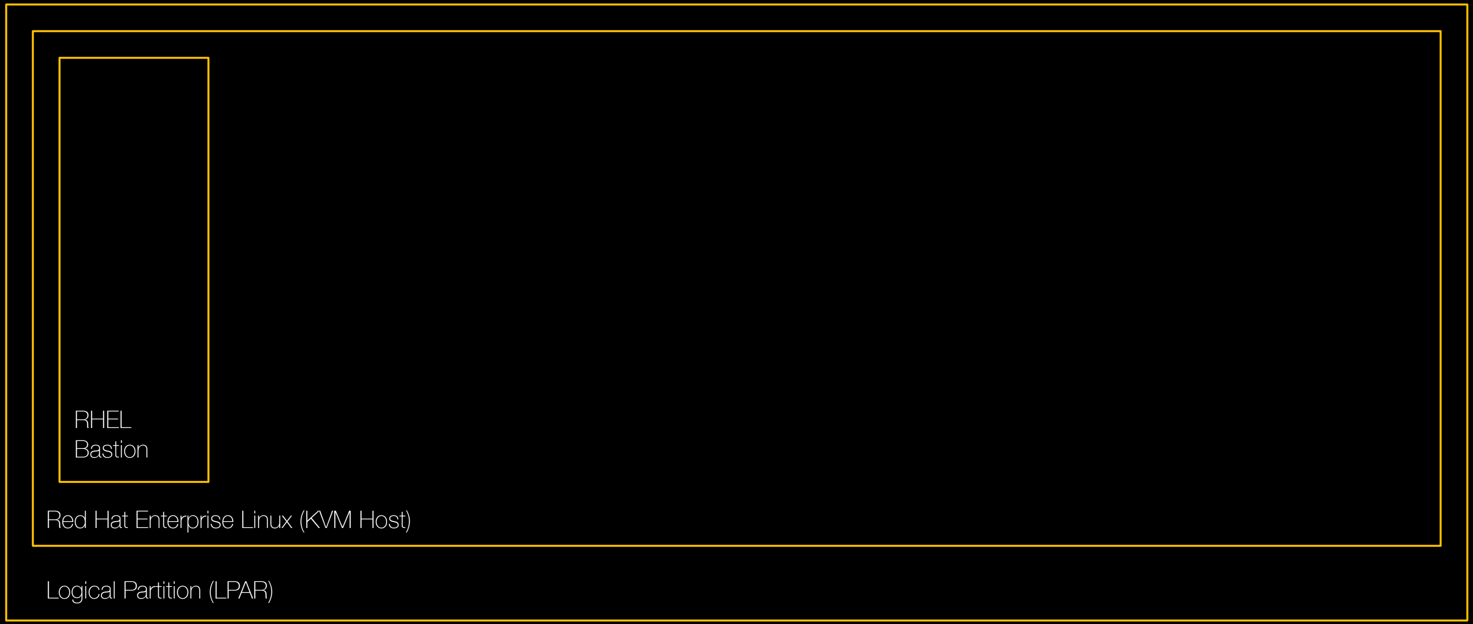
2 Start the install process

```
virt# virt-install --connect qemu:///system --name bastion --memory 4096 --vcpus 2 --disk size=20 --cdrom /var/lib/libvirt/images/rhel83.iso
--accelerate --import --network network=macvtap-net --extra-args "ip=172.16.10.212::172.16.10.1:255.255.255.0:bastion.ocp.home.local::none
nameserver=172.16.10.38 vnc vncpassword=12341234 inst.repo=hd:/dev/vda ipv6.disable=1" --location /rhcos-install --qemu-commandline="-drive
if=none,id=ignition,format=raw,file=/var/lib/libvirt/images/rhel83.iso,readonly=on -device virtio-blk,serial=ignition,drive=ignition" --
noautoconsole
```

3 Use a local (laptop) VNC Viewer to connect to the Bastion VM to complete the RHEL Bastion install process.

Red Hat OpenShift Container Platform Install Process

KVM



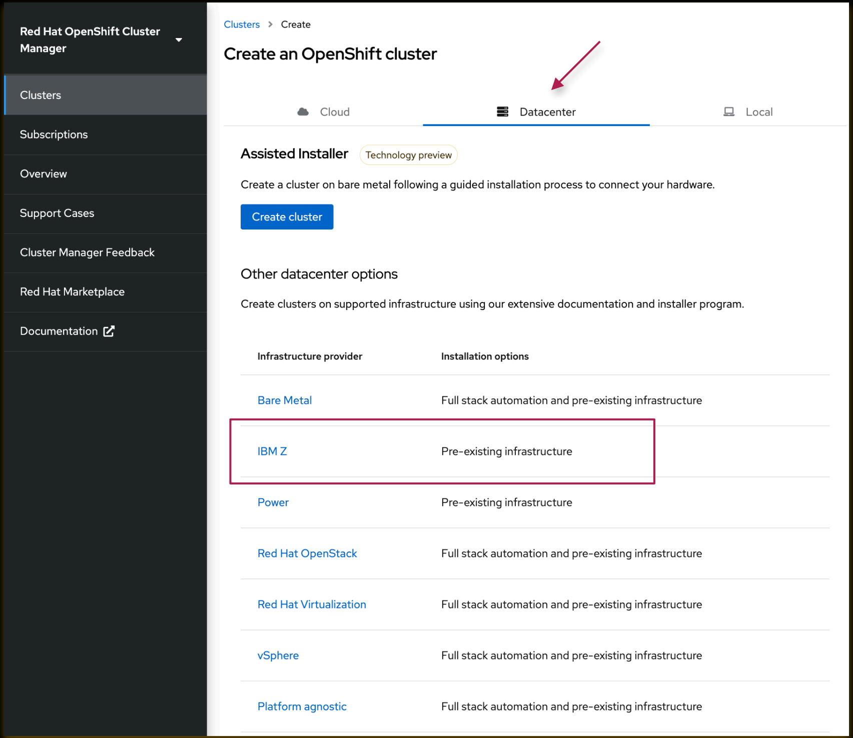
RHEL
Bastion

Red Hat Enterprise Linux (KVM Host)

Logical Partition (LPAR)

Download Software

https://cloud.redhat.com/



Red Hat OpenShift Cluster Manager

- Clusters
- Subscriptions
- Overview
- Support Cases
- Cluster Manager Feedback
- Red Hat Marketplace
- Documentation

Clusters > Create

Create an OpenShift cluster

Cloud | Datacenter | Local

Assisted Installer

Technology preview

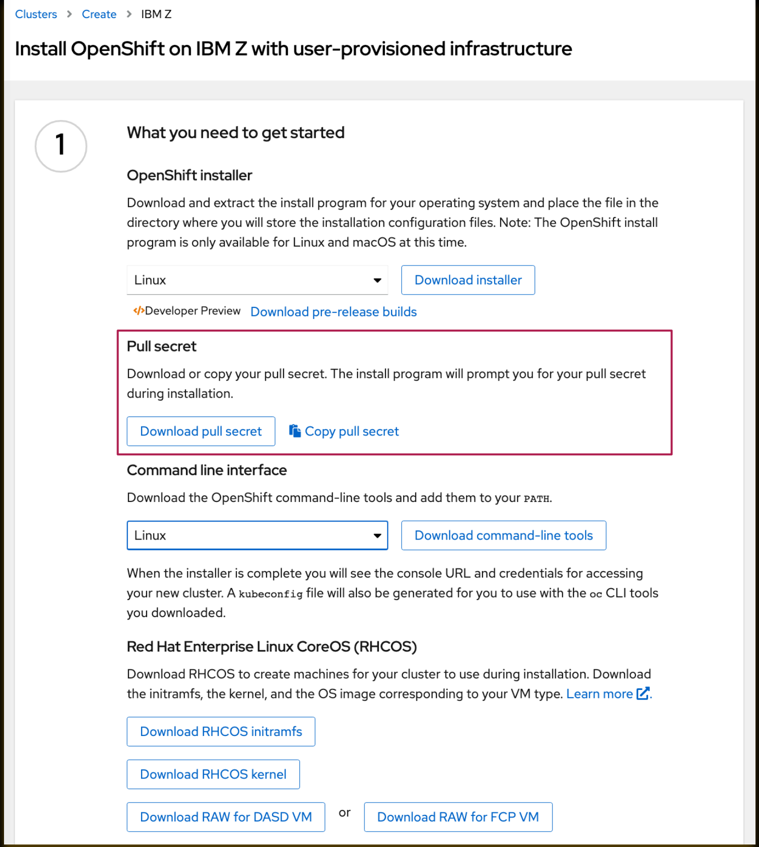
Create a cluster on bare metal following a guided installation process to connect your hardware.

Create cluster

Other datacenter options

Create clusters on supported infrastructure using our extensive documentation and installer program.

Infrastructure provider	Installation options
Bare Metal	Full stack automation and pre-existing infrastructure
IBM Z	Pre-existing infrastructure
Power	Pre-existing infrastructure
Red Hat OpenStack	Full stack automation and pre-existing infrastructure
Red Hat Virtualization	Full stack automation and pre-existing infrastructure
vSphere	Full stack automation and pre-existing infrastructure
Platform agnostic	Full stack automation and pre-existing infrastructure



Clusters > Create > IBM Z

Install OpenShift on IBM Z with user-provisioned infrastructure

1

What you need to get started

OpenShift installer

Download and extract the install program for your operating system and place the file in the directory where you will store the installation configuration files. Note: The OpenShift install program is only available for Linux and macOS at this time.

Linux | Download installer

Developer Preview Download pre-release builds

Pull secret

Download or copy your pull secret. The install program will prompt you for your pull secret during installation.

Download pull secret | Copy pull secret

Command line interface

Download the OpenShift command-line tools and add them to your `PATH`.

Linux | Download command-line tools

When the installer is complete you will see the console URL and credentials for accessing your new cluster. A `kubeconfig` file will also be generated for you to use with the `oc` CLI tools you downloaded.

Red Hat Enterprise Linux CoreOS (RHCOS)

Download RHCOS to create machines for your cluster to use during installation. Download the initramfs, the kernel, and the OS image corresponding to your VM type. Learn more.

Download RHCOS initramfs

Download RHCOS kernel

Download RAW for DASD VM or Download RAW for FCP VM

https://mirror.openshift.com/pub/openshift-v4/s390x/clients/ocp/latest/

https://mirror.openshift.com/pub/openshift-v4/s390x/dependencies/rhcos/latest/latest/

DNS Requirements and Configuration Example:

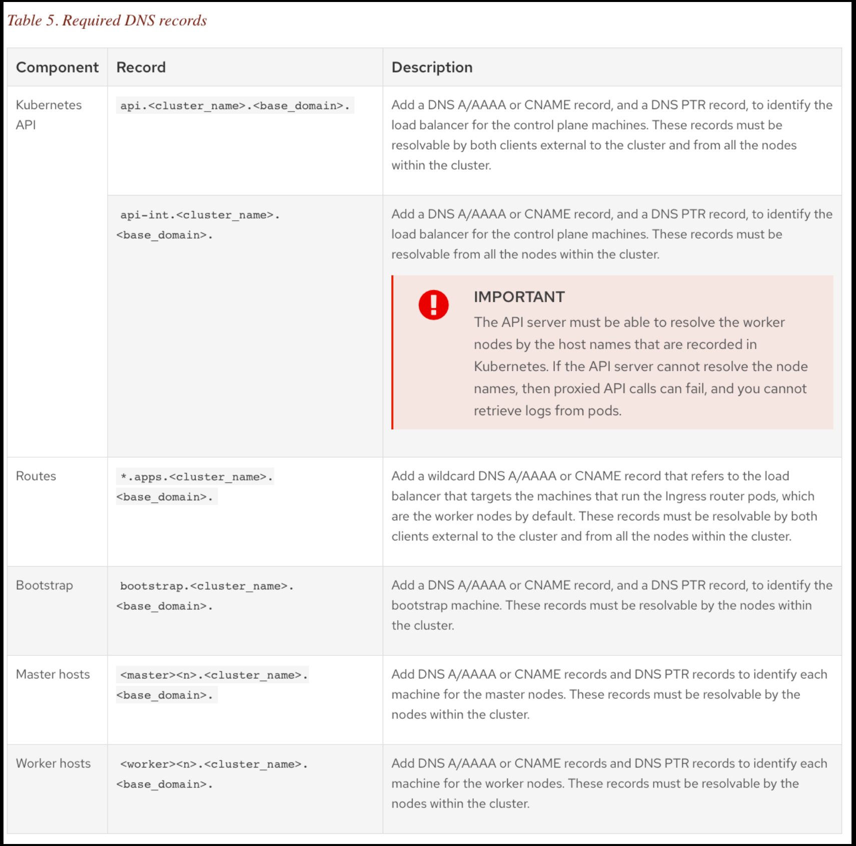


Table 5. Required DNS records

Component	Record	Description
Kubernetes API	`api.<cluster_name>.<base_domain>.`	Add a DNS A/AAAA or CNAME record, and a DNS PTR record, to identify the load balancer for the control plane machines. These records must be resolvable by both clients external to the cluster and from all the nodes within the cluster.
	`api-int.<cluster_name>.<base_domain>.`	Add a DNS A/AAAA or CNAME record, and a DNS PTR record, to identify the load balancer for the control plane machines. These records must be resolvable from all the nodes within the cluster. **IMPORTANT** The API server must be able to resolve the worker nodes by the host names that are recorded in Kubernetes. If the API server cannot resolve the node names, then proxied API calls can fail, and you cannot retrieve logs from pods.
Routes	`*.apps.<cluster_name>.<base_domain>.`	Add a wildcard DNS A/AAAA or CNAME record that refers to the load balancer that targets the machines that run the Ingress router pods, which are the worker nodes by default. These records must be resolvable by both clients external to the cluster and from all the nodes within the cluster.
Bootstrap	`bootstrap.<cluster_name>.<base_domain>.`	Add a DNS A/AAAA or CNAME record, and a DNS PTR record, to identify the bootstrap machine. These records must be resolvable by the nodes within the cluster.
Master hosts	`<master><n>.<cluster_name>.<base_domain>.`	Add DNS A/AAAA or CNAME records and DNS PTR records to identify each machine for the master nodes. These records must be resolvable by the nodes within the cluster.
Worker hosts	`<worker><n>.<cluster_name>.<base_domain>.`	Add DNS A/AAAA or CNAME records and DNS PTR records to identify each machine for the worker nodes. These records must be resolvable by the nodes within the cluster.

```
@ IN SOA ns1.<domain>. admin.<domain>. (
                                    2020021821 ;Serial
                                    3600 ;Refresh
                                    1800 ;Retry
                                    604800 ;Expire
                                    86400 ;Minimum TTL
)

;Name Server Information
@ IN NS ns1.<domain>.

;IP Address for Name Server
ns1 IN A <DNS_server_IP_address>

; entry for the bootstrap host.
bootstrap.<cluster_name>   IN   A   <bootstrap_IP_address>

; entry of your load balancer
haproxy    IN   A   <loadbalancer_IP_address>

;  entries for the master hosts
<control plane 1>.<cluster_name>    IN   A   <control_plane1_IP_address>
<control plane 2>.<cluster_name>    IN   A   <control_plane2_IP_address>
<control plane 3>.<cluster_name>    IN   A   <control_plane3_IP_address>

; entry for the bastion host
bastion     IN   A   <Infra_server_IP_address>

; entries for the workers hosts
<compute plane 1>.<cluster_name>    IN   A   <compute_plane1_IP_address>
<compute plane 2>.<cluster_name>    IN   A   <compute_plane2_IP_address>

; The api identifies the IP of your load balancer.
api.<cluster_name>     IN CNAME haproxy.<domain>.
api-int.<cluster_name> IN CNAME haproxy.<domain>.

; The wildcard also identifies the load balancer.
*.apps.<cluster_name>  IN CNAME haproxy.<domain>.
```

Load Balancer

HAProxy Example:

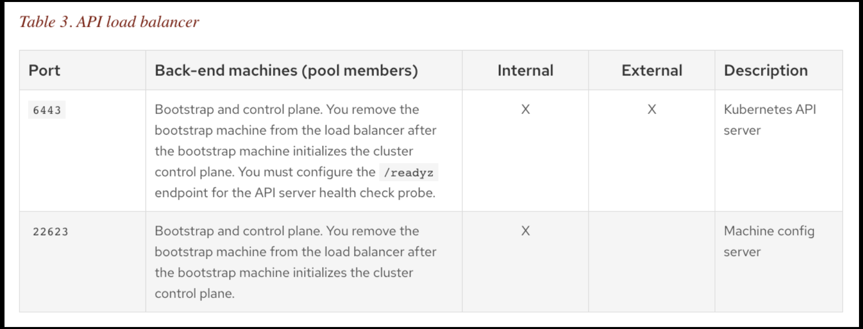


Table 3. API load balancer

Port	Back-end machines (pool members)	Internal	External	Description
`6443`	Bootstrap and control plane. You remove the bootstrap machine from the load balancer after the bootstrap machine initializes the cluster control plane. You must configure the `/readyz` endpoint for the API server health check probe.	X	X	Kubernetes API server
`22623`	Bootstrap and control plane. You remove the bootstrap machine from the load balancer after the bootstrap machine initializes the cluster control plane.	X		Machine config server

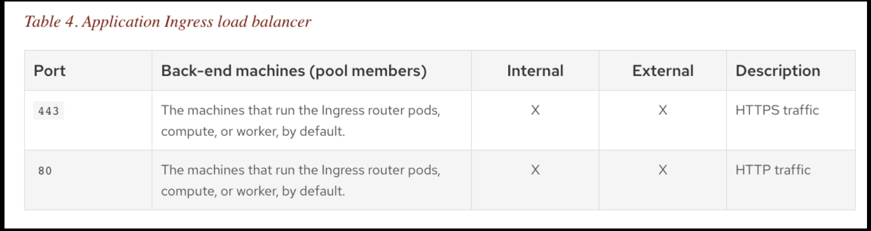


Table 4. Application Ingress load balancer

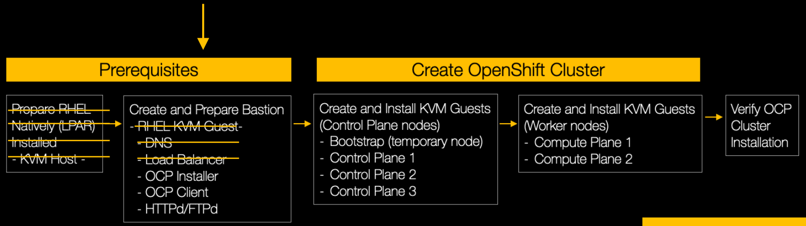
Port	Back-end machines (pool members)	Internal	External	Description
`443`	The machines that run the Ingress router pods, compute, or worker, by default.	X	X	HTTPS traffic
`80`	The machines that run the Ingress router pods, compute, or worker, by default.	X	X	HTTP traffic

1 `# dnf install -y haproxy`

2 `/etc/haproxy/haproxy.cfg`

```
listen ingress-http
    bind *:80
    mode tcp
    server worker0 <worker0_IP>:80 check
    server worker1 <worker1_IP>:80 check

listen ingress-https
    bind *:443
    mode tcp
    server worker0 <worker0_IP>:443 check
    server worker1 <worker1_IP>:443 check

listen api
    bind *:6443
    mode tcp
    server bootstrap <bootstrap_IP>:6443 check
    server master0 <master0_IP>:6443 check
    server master1 <master1_IP>:6443 check
    server master2 <master2_IP>:6443 check

listen api-int
    bind *:22623
    mode tcp
    server bootstrap <bootstrap_IP>:22623 check
    server master0 <master0_IP>:22623 check
    server master1 <master1_IP>:22623 check
    server master2 <master2_IP>:22623 check
```

Create and configure the HTTP server



Prerequisites

~~Prepare RHEL Natively (LPAR) Installed~~
~~- KVM Host -~~

Create and Prepare Bastion
- ~~RHEL KVM Guest~~-
- ~~DNS~~
- ~~Load Balancer~~
- OCP Installer
- OCP Client
- HTTPd/FTPd

Create OpenShift Cluster

Create and Install KVM Guests (Control Plane nodes)
- Bootstrap (temporary node)
- Control Plane 1
- Control Plane 2
- Control Plane 3

Create and Install KVM Guests (Worker nodes)
- Compute Plane 1
- Compute Plane 2

Verify OCP Cluster Installation

NOTE

```
Always check the latest versions of OpenShift, the file names will change as new versions
Are made available:
https://mirror.openshift.com/pub/openshift-v4/s390x/dependencies/rhcos/latest/latest/
```

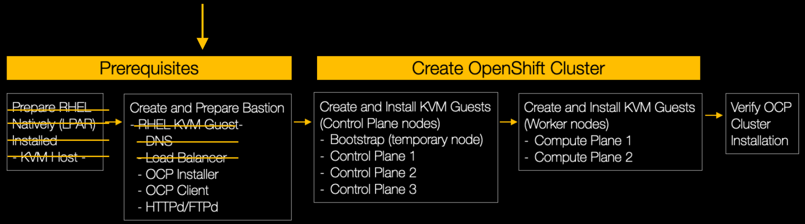
1
```
# dnf install -y httpd
```

2 Change default port to 8080

3
```
# mkdir /var/www/html/bin /var/www/html/bootstrap

# wget https://mirror.openshift.com/pub/openshift-v4/s390x/dependencies/rhcos/latest/latest/rhcos-4.7.7-s390x-live-kernel-s390x
-O /var/www/html/bin/rhcos-kernel

# wget https://mirror.openshift.com/pub/openshift-v4/s390x/dependencies/rhcos/latest/latest/rhcos-4.7.7-s390x-live-
initramfs.s390x.img -O /var/www/html/bin/rhcos-initramfs.img

# wget https://mirror.openshift.com/pub/openshift-v4/s390x/dependencies/rhcos/latest/latest/rhcos-4.7.7-s390x-live-
rootfs.s390x.img -O rhcos-rootfs.img
```

4
```
# systemctl enable --now httpd; systemctl status httpd
```

Installer and oc Client Tools



Prerequisites

~~Prepare RHEL Natively (LPAR) Installed~~
~~- KVM Host -~~

Create and Prepare Bastion
~~- RHEL KVM Guest -~~
~~- DNS~~
~~- Load Balancer~~
- OCP Installer
- OCP Client
- HTTPd/FTPd

Create OpenShift Cluster

Create and Install KVM Guests (Control Plane nodes)
- Bootstrap (temporary node)
- Control Plane 1
- Control Plane 2
- Control Plane 3

Create and Install KVM Guests (Worker nodes)
- Compute Plane 1
- Compute Plane 2

Verify OCP Cluster Installation

1

```
# wget https://mirror.openshift.com/pub/openshift-v4/s390x/clients/ocp/latest/openshift-client-linux.tar.gz
tar -xvzf openshift-client-linux.tar.gz
```

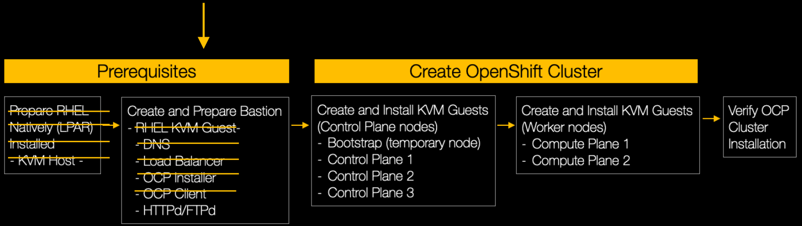
2

```
# wget https://mirror.openshift.com/pub/openshift-v4/s390x/clients/ocp/latest/openshift-install-linux.tar.gz
tar -xvzf openshift-install-linux.tar.gz
```

3

```
# chmod +x kubectl  oc  openshift-install
# mv kubectl  oc  openshift-install /usr/local/bin/
```

Install-config.yaml



Prerequisites

~~Prepare RHEL Natively (LPAR) Installed - KVM Host -~~

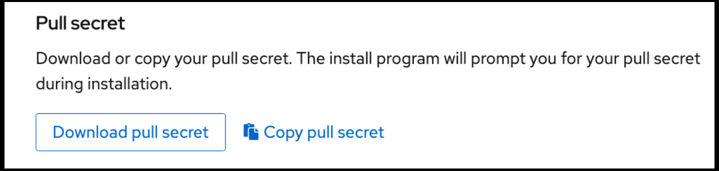
Create and Prepare Bastion
- ~~RHEL KVM Guest-~~
- ~~DNS~~
- ~~Load Balancer~~
- ~~OCP Installer~~
- ~~OCP Client~~
- HTTPd/FTPd

Create OpenShift Cluster

Create and Install KVM Guests (Control Plane nodes)
- Bootstrap (temporary node)
- Control Plane 1
- Control Plane 2
- Control Plane 3

Create and Install KVM Guests (Worker nodes)
- Compute Plane 1
- Compute Plane 2

Verify OCP Cluster Installation



Pull secret

Download or copy your pull secret. The install program will prompt you for your pull secret during installation.

install-config.yaml

```
apiVersion: v1
baseDomain: <domain>
compute:
- architecture: s390x
  hyperthreading: Enabled
  name: worker
  replicas: 0
controlPlane:
  architecture: s390x
  hyperthreading: Enabled
  name: master
  replicas: 3
metadata:
  name: <cluster_name>
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  networkType: OpenShiftSDN
  serviceNetwork:
  - 172.30.0.0/16
platform:
  none: {}
fips: false
pullSecret: '<pull-secret>'
sshKey: '<ssh-public-key>'
```

1

2

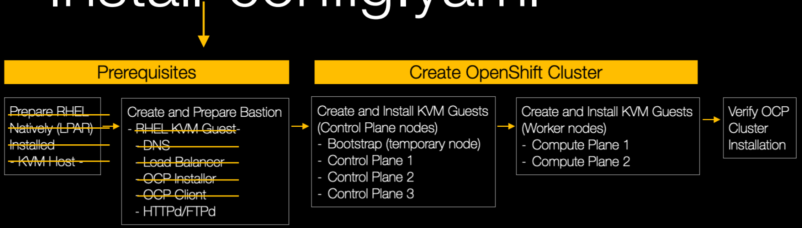
3

4

If you do not have an SSH key that is configured for password-less authentication on your computer, create one. For example, on a computer that uses a Linux operating system, run the following command:

```
# ssh-keygen -t rsa -b 4096 -N ''
```

Install-config.yaml



Prerequisites

~~Prepare RHEL Natively (LPAR) Installed - KVM Host -~~

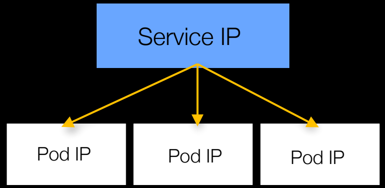
Create and Prepare Bastion
- ~~RHEL KVM Guest-~~
- ~~DNS~~
- ~~Load Balancer~~
- ~~OCP Installer~~
- ~~OCP Client~~
- HTTPd/FTPd

Create OpenShift Cluster

Create and Install KVM Guests (Control Plane nodes)
- Bootstrap (temporary node)
- Control Plane 1
- Control Plane 2
- Control Plane 3

Create and Install KVM Guests (Worker nodes)
- Compute Plane 1
- Compute Plane 2

Verify OCP Cluster Installation



Service IP

Pod IP

Pod IP

Pod IP

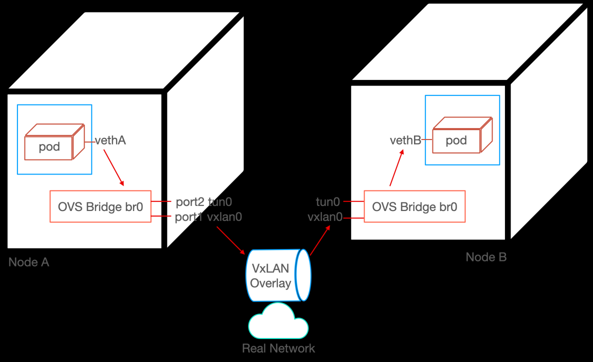
Pod's IP range

OpenShift SDN

Service's IP range

install-config.yaml

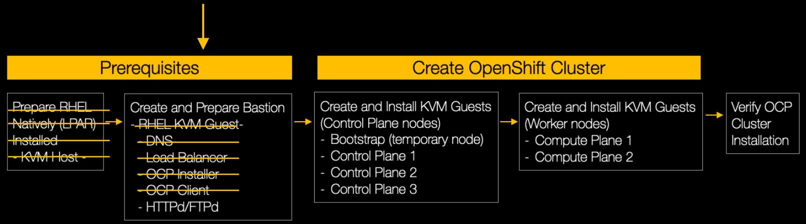
```
apiVersion: v1
baseDomain: <domain>
compute:
- architecture: s390x
  hyperthreading: Enabled
  name: worker
  replicas: 0
controlPlane:
  architecture: s390x
  hyperthreading: Enabled
  name: master
  replicas: 3
metadata:
  name: <cluster_name>
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  networkType: OpenShiftSDN
  serviceNetwork:
  - 172.30.0.0/16
platform:
  none: {}
fips: false
pullSecret: '<pull-secret>'
sshKey: '<ssh-public-key>'
```



Traffic Flow

eth0 - vethA - br0 - vxlan0 - network - vxlan0 - br0 - vethB - eth0

Generate the Ignition Files



Prerequisites

~~Prepare RHEL Natively (LPAR) Installed - KVM Host -~~

Create and Prepare Bastion
- ~~RHEL KVM Guest~~
- ~~DNS~~
- ~~Load Balancer~~
- ~~OCP Installer~~
- ~~OCP Client~~
- HTTPd/FTPd

Create OpenShift Cluster

Create and Install KVM Guests (Control Plane nodes)
- Bootstrap (temporary node)
- Control Plane 1
- Control Plane 2
- Control Plane 3

Create and Install KVM Guests (Worker nodes)
- Compute Plane 1
- Compute Plane 2

Verify OCP Cluster Installation

1

```
# ./openshift-install create manifests --dir=<installation_directory>
```

Modify the /<installation_directory>/manifests/cluster-scheduler-02-config.yml mastersSchedulable parameter and set its value to False

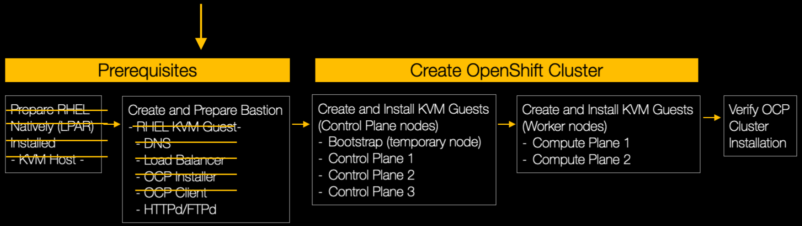
2

```
# ./openshift-install create ignition-configs --dir=<installation_directory>
```

3 Copy the bootstrap.ign, master.ign and worker.ign to your already pre-configured HTTPd

```
# cp <installation_directory>/*.ign /var/www/html/ignition
# chmod 775 /var/www/html/ignition/*.ign
```

Prepare the KVM OCP guests



Prerequisites

Create OpenShift Cluster

~~Prepare RHEL Natively (LPAR) Installed~~
~~- KVM Host -~~

Create and Prepare Bastion
~~-RHEL KVM Guest-~~
~~- DNS~~
~~- Load Balancer~~
~~- OCP Installer~~
~~- OCP Client~~
- HTTPd/FTPd

Create and Install KVM Guests (Control Plane nodes)
- Bootstrap (temporary node)
- Control Plane 1
- Control Plane 2
- Control Plane 3

Create and Install KVM Guests (Worker nodes)
- Compute Plane 1
- Compute Plane 2

Verify OCP Cluster Installation

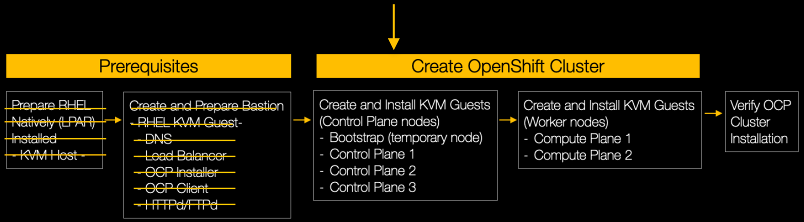
1

```
# wget https://mirror.openshift.com/pub/openshift-v4/s390x/dependencies/rhcos/latest/latest/rhcos-qemu.s390x.qcow2.gz
```

2

```
# dnf install -y gzip
# gunzip rhcos-qemu.s390x.qcow2.gz /var/lib/libvirt/images/
```

Create Bootstrap



Prerequisites

~~Prepare RHEL Natively (LPAR) Installed~~
~~- KVM Host -~~

~~Create and Prepare Bastion~~
~~- RHEL KVM Guest-~~
~~- DNS~~
~~- Load Balancer~~
~~- OCP Installer~~
~~- OCP Client~~
~~- HTTPd/FTPd~~

Create OpenShift Cluster

Create and Install KVM Guests (Control Plane nodes)
- Bootstrap (temporary node)
- Control Plane 1
- Control Plane 2
- Control Plane 3

Create and Install KVM Guests (Worker nodes)
- Compute Plane 1
- Compute Plane 2

Verify OCP Cluster Installation

1

```
# qemu-img create -f qcow2 -F qcow2 -b /var/lib/libvirt/images/rhcos-qemu.s390x.qcow2 /var/lib/libvirt/images/bootstrap.qcow2 120G
```

2

```
# virt-install --boot kernel=rhcos-kernel,initrd=rhcos-initramfs.img,kernel_args='rd.neednet=1
coreos.inst.install_dev=/dev/vda coreos.live.rootfs_url=http://<bastion_IP>:8080/bin/rhcos-rootfs.img
coreos.inst.ignition_url=http://<bastion_IP>:8080/ignition/bootstrap.ign ip=<bootstrap_IP>::<gateway>:<netmask>:::none
nameserver=<bastion_IP>' --connect qemu:///system --name bootstrap --memory 16384 --vcpus 4 --disk /var/lib/libvirt/
images/bootstrap.qcow2 --accelerate --import --network network=macvtap-net --qemu-commandline="-drive
if=none,id=ignition,format=raw,file=/var/www/html/ignition/bootstrap.ign,readonly=on -device virtio-
blk,serial=ignition,drive=ignition"
```

3 Wait the Bootstrap creation. To verify the install process:

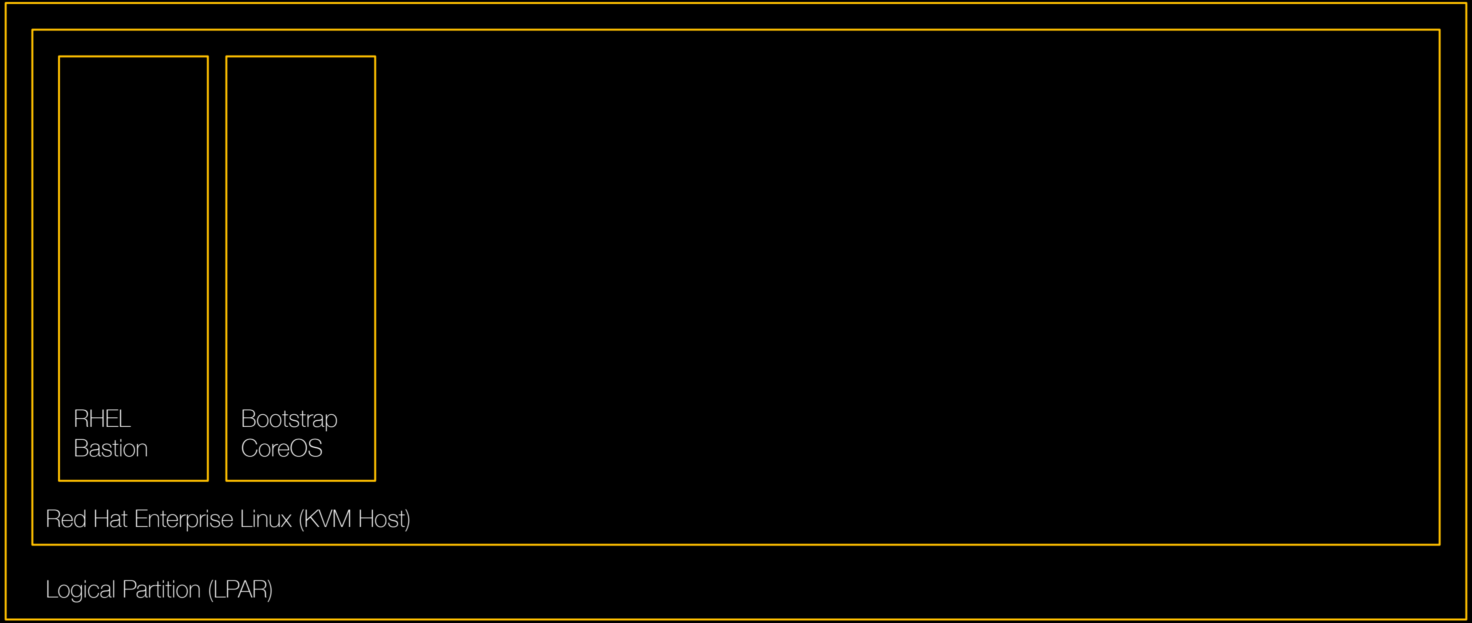
```
# virsh console bootstrap
# journalctl  -u bootkube.service
```

NOTE

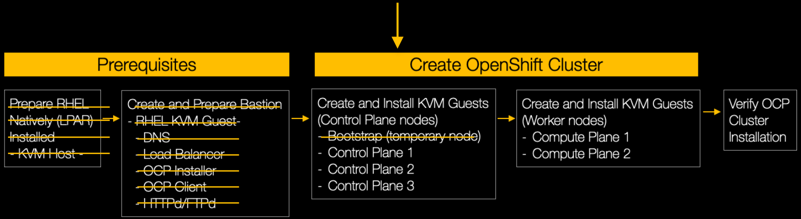
Expect many error messages from the bootstrap's log.
Wait for the message: **bootkube.service complete** once the **bootstrap** and **all control plane nodes** are **up** and **running**.

Red Hat OpenShift Container Platform Install Process

KVM



Create Masters



Prerequisites

~~Prepare RHEL Natively (LPAR) Installed~~
~~- KVM Host -~~

~~Create and Prepare Bastion~~
~~- RHEL KVM Guest-~~
~~-DNS~~
~~-Load Balancer~~
~~-OCP Installer~~
~~-OCP Client~~
~~-HTTPd/FTPd~~

Create OpenShift Cluster

Create and Install KVM Guests (Control Plane nodes)
~~- Bootstrap (temporary node)~~
- Control Plane 1
- Control Plane 2
- Control Plane 3

Create and Install KVM Guests (Worker nodes)
- Compute Plane 1
- Compute Plane 2

Verify OCP Cluster Installation

1

```
# qemu-img create -f qcow2 -F qcow2 -b /var/lib/libvirt/images/rhcos-qemu.s390x.qcow2 /var/lib/libvirt/images/master1.qcow2 120G
# qemu-img create -f qcow2 -F qcow2 -b /var/lib/libvirt/images/rhcos-qemu.s390x.qcow2 /var/lib/libvirt/images/master2.qcow2 120G
# qemu-img create -f qcow2 -F qcow2 -b /var/lib/libvirt/images/rhcos-qemu.s390x.qcow2 /var/lib/libvirt/images/master3.qcow2 120G
```

2

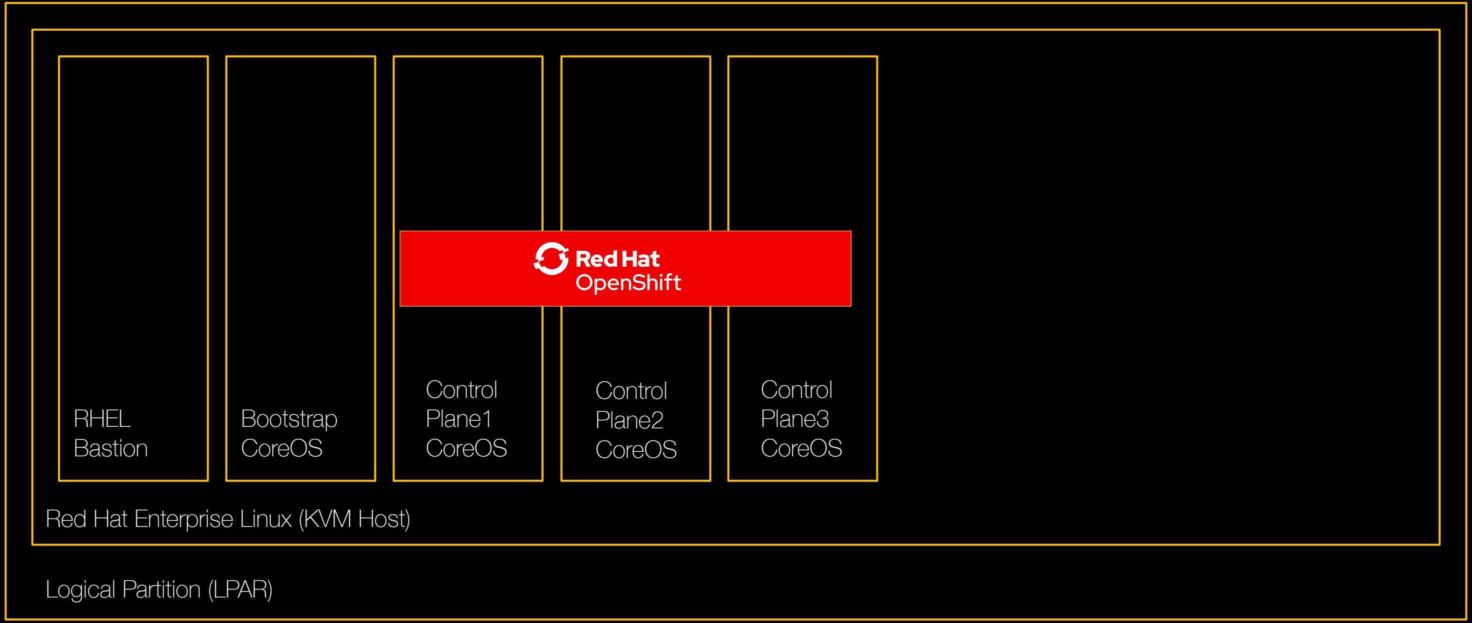
```
# virt-install --boot kernel=rhcos-kernel,initrd=rhcos-initramfs.img,kernel_args='rd.neednet=1 coreos.inst.install_dev=/dev/
vda coreos.live.rootfs_url=http://<bastion_IP>:8080/bin/rhcos-rootfs.img coreos.inst.ignition_url=http://<bastion_IP>:8080/
ignition/master.ign ip=<master1_IP>::<gateway>:<netmask>:::none nameserver=<bastion_IP>’ --connect qemu:///system --name
master1 --memory 16384 --vcpus 4 --disk /var/lib/libvirt/images/master1.qcow2 --accelerate --import --network
network=macvtap-net --qemu-commandline="-drive if=none,id=ignition,format=raw,file=/var/www/html/ignition/
master.ign,readonly=on -device virtio-blk,serial=ignition,drive=ignition"
```

```
# virt-install --boot kernel=rhcos-kernel,initrd=rhcos-initramfs.img,kernel_args='rd.neednet=1 coreos.inst.install_dev=/dev/
vda coreos.live.rootfs_url=http://<bastion_IP>:8080/bin/rhcos-rootfs.img coreos.inst.ignition_url=http://<bastion_IP>:8080/
ignition/master.ign ip=<master2_IP>::<gateway>:<netmask>:::none nameserver=bastion_IP>’ --connect qemu:///system --name
master2 --memory 16384 --vcpus 4 --disk /var/lib/libvirt/images/master2.qcow2 --accelerate --import --network
network=macvtap-net --qemu-commandline="-drive if=none,id=ignition,format=raw,file=/var/www/html/ignition/
master.ign,readonly=on -device virtio-blk,serial=ignition,drive=ignition"
```

```
# virt-install --boot kernel=rhcos-kernel,initrd=rhcos-initramfs.img,kernel_args='rd.neednet=1 coreos.inst.install_dev=/dev/
vda coreos.live.rootfs_url=http://<bastion_IP>:8080/bin/rhcos-rootfs.img coreos.inst.ignition_url=http://<bastion_IP>:8080/
ignition/master.ign ip=<master3_IP>::<gateway>:<netmask>:::none nameserver=<bastion_IP>’ --connect qemu:///system --name
master3 --memory 16384 --vcpus 4 --disk /var/lib/libvirt/images/master3.qcow2 --accelerate --import --network
network=macvtap-net --qemu-commandline="-drive if=none,id=ignition,format=raw,file=/var/www/html/ignition/
master.ign,readonly=on -device virtio-blk,serial=ignition,drive=ignition"
```

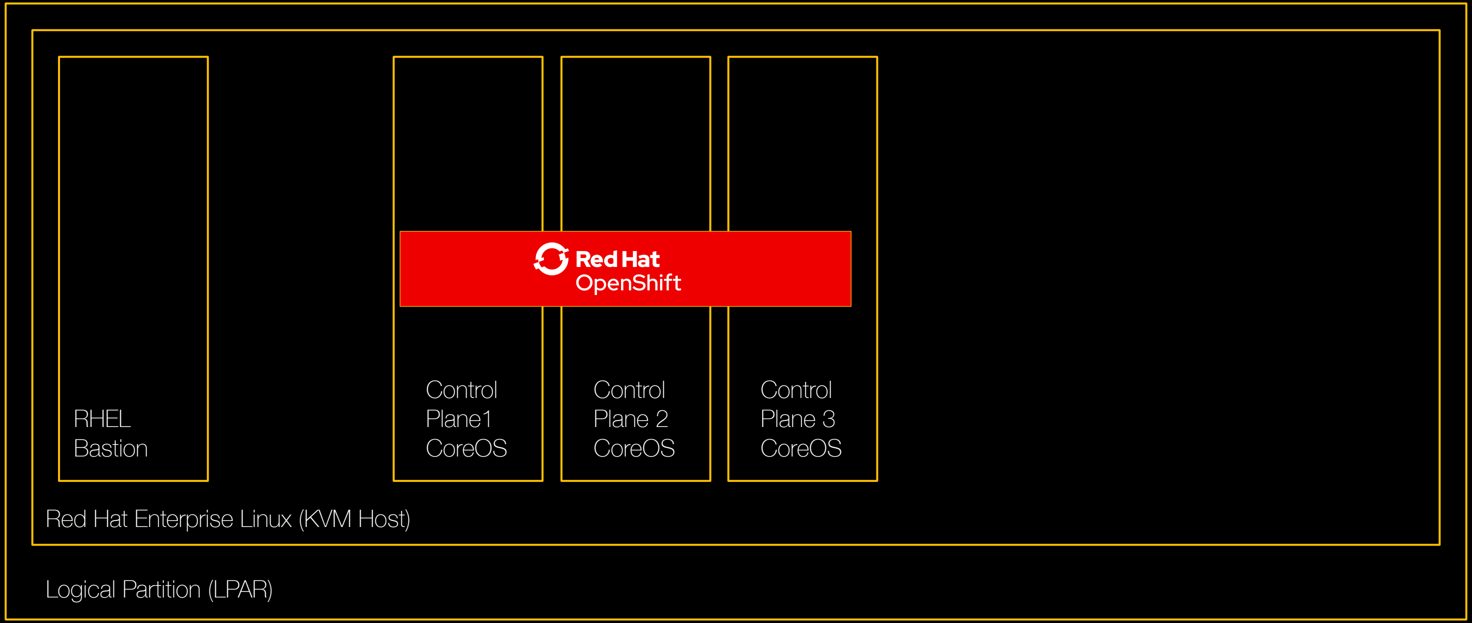
Red Hat OpenShift Container Platform Install Process

KVM



Red Hat OpenShift Container Platform Install Process

KVM



RHEL
Bastion

Control
Plane1
CoreOS

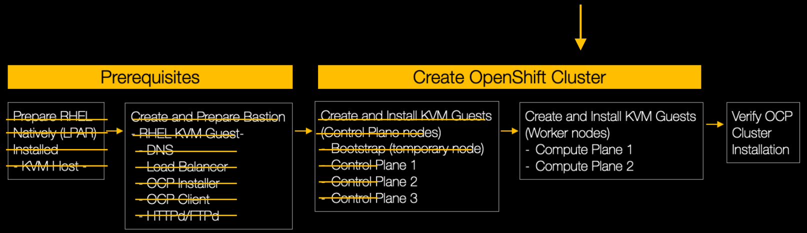
Control
Plane 2
CoreOS

Control
Plane 3
CoreOS

Red Hat Enterprise Linux (KVM Host)

Logical Partition (LPAR)

Create Workers



Prerequisites

~~Prepare RHEL Natively (LPAR) Installed - KVM Host -~~

~~Create and Prepare Bastion~~
- ~~RHEL KVM Guest-~~
- ~~DNS~~
- ~~Load Balancer~~
- ~~OCP Installer~~
- ~~OCP Client~~
- ~~HTTPd/FTPd~~

Create OpenShift Cluster

~~Create and Install KVM Guests (Control Plane nodes)~~
- ~~Bootstrap (temporary node)~~
- ~~Control Plane 1~~
- ~~Control Plane 2~~
- ~~Control Plane 3~~

Create and Install KVM Guests (Worker nodes)
- Compute Plane 1
- Compute Plane 2

Verify OCP Cluster Installation

1

```
# qemu-img create -f qcow2 -F qcow2 -b /var/lib/libvirt/images/rhcos-qemu.s390x.qcow2 /var/lib/libvirt/images/worker1.qcow2 120G
# qemu-img create -f qcow2 -F qcow2 -b /var/lib/libvirt/images/rhcos-qemu.s390x.qcow2 /var/lib/libvirt/images/worker2.qcow2 120G
```

2

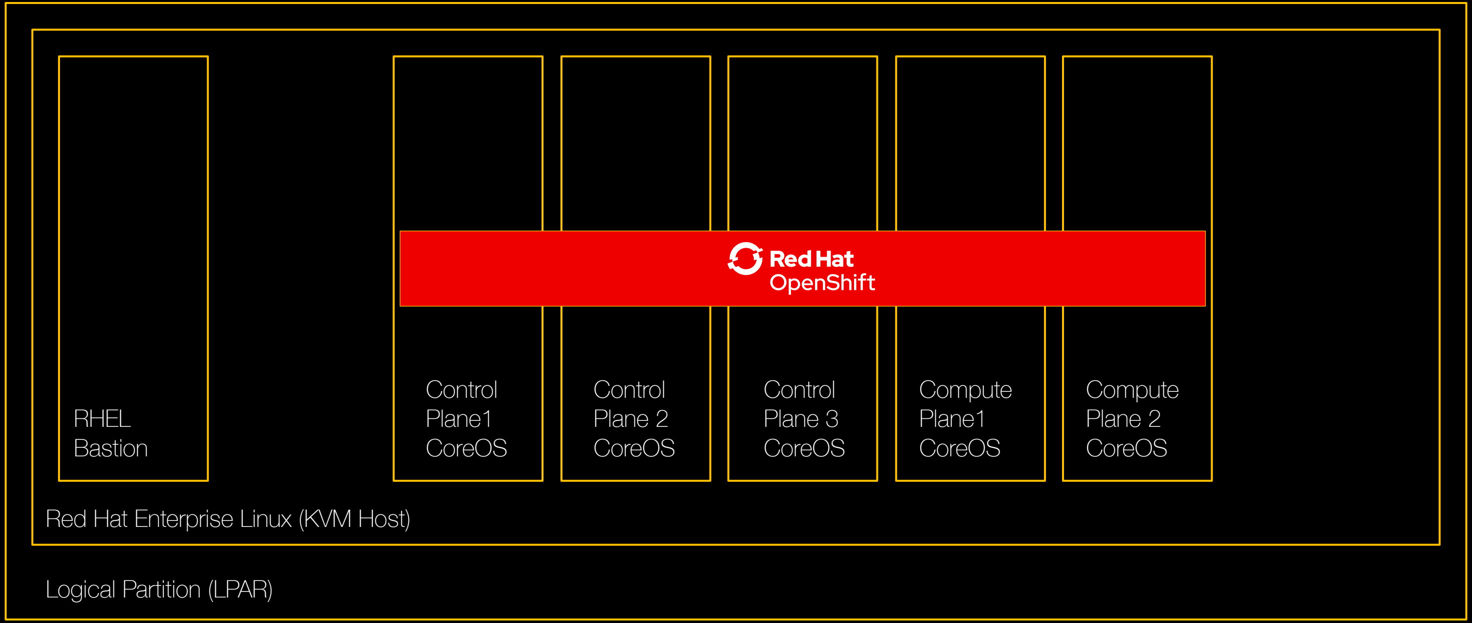
```
# virt-install --boot kernel=rhcos-kernel,initrd=rhcos-initramfs.img,kernel_args='rd.neednet=1 coreos.inst.install_dev=/dev/vda
coreos.live.rootfs_url=http://<bastion_IP>:8080/bin/rhcos-rootfs.img coreos.inst.ignition_url=http://<bastion_IP>:8080/ignition/
worker.ign ip=<worker1_IP>::<gateway>:<netmask>:::none nameserver=<bastion_IP>’ --connect qemu:///system --name worker1 --memory
16384 --vcpus 4 --disk /var/lib/libvirt/images/worker1.qcow2 --accelerate --import --network network=macvtap-net --qemu-
commandline="-drive if=none,id=ignition,format=raw,file=/var/www/html/ignition/worker.ign,readonly=on -device virtio-
blk,serial=ignition,drive=ignition"
```

```
# virt-install --boot kernel=rhcos-kernel,initrd=rhcos-initramfs.img,kernel_args='rd.neednet=1 coreos.inst.install_dev=/dev/vda
coreos.live.rootfs_url=http://<bastion_IP>:8080/bin/rhcos-rootfs.img coreos.inst.ignition_url=http://<bastion_IP>:8080/ignition/
worker.ign ip=<worker2_IP>::<gateway>:<netmask>:::none nameserver=<bastion_IP>’ --connect qemu:///system --name worker2 --memory
16384 --vcpus 4 --disk /var/lib/libvirt/images/worker2.qcow2 --accelerate --import --network network=macvtap-net --qemu-
commandline="-drive if=none,id=ignition,format=raw,file=/var/www/html/ignition/worker.ign,readonly=on -device virtio-
blk,serial=ignition,drive=ignition"
```

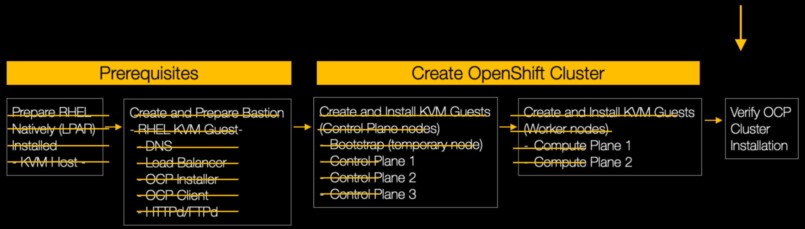
3 From your Bastion system, use the OC client to connect the OCP cluster and approve
Any pending csr certificate:

```
# oc get csr
# oc adm certificate approve <csr-name>
```

Red Hat OpenShift Container Platform Install Process



Verify the Installation



1 Monitor nodes and cluster operators

```
# oc get nodes
# oc get clusteroperators
```

```
NAME        STATUS  ROLES  AGE  VERSION
master0.<cluster_name>.<domain>   Ready  master  20m
master1.<cluster_name>.<domain>   Ready  master  20m
master2.<cluster_name>.<domain>   Ready  master  20m
worker0.<cluster_name>.<domain>   Ready  worker  20m
worker1.<cluster_name>.<domain>   Ready  worker  20m
```

```
NAME                    VERSION  AVAILABLE  PROGRESSING  DEGRADED  SINCE
authentication           4.7.7    True       False        False     20m
cloud-credential         4.7.7    True       False        False     20m
cluster-autoscaler       4.7.7    True       False        False     20m
console                  4.7.7    True       False        False     5h36m
...
```

1.1 Cluster operator does not come up

```
# oc describe co <clusteroperator>
# oc get pods -n <namespace>
# oc get all -n <namespace>
# oc logs <type>/<name> -n <namespace>
# oc describe nodes
```

2 Once all operators are marked as State TRUE, conclude the install process:

```
# ./openshift-install --dir=<installation_directory> wait-for install-complete
```

Final Step to complete OCP installation process

References:

Red Hat OpenShift Container Platform Reference Architecture
https://www.ibm.com/docs/en/linux-on-systems?topic=configuration-red-hat-openshift-reference

Red Hat OpenShift Container Platform (Static IP)
https://kvmonz.blogspot.com/2021/03/installing-red-hat-openshift-on-kvm-on-z.html

Special Thanks to:
Anderson Augusto Da Silveira
Mainframe Lab Services Field Technical Consultant
andersonz@br.ibm.com

Thank you
Grazie
Merci
Gracias
Obrigado
ありがとう
谢谢
Dankeschön

Worldwide zAcceleration

Chris Backer
zAcceleration Leader
cbacker@us.ibm.com
zAcceleration
Red Hat Synergy
Worldwide IBM Z

Filipe Miranda
Cloud Solutions Architect Leader
fmiranda@ibm.com
zAcceleration
Red Hat Synergy
Worldwide IBM Z

Elton de Souza
Chief Architect
Elton.desouza@ca.ibm.com
Cloud Native Client Success

Pat Fruth
Cloud Paks Leader
pfruth@us.ibm.com
zAcceleration
Red Hat Synergy
Worldwide IBM Z

Roberto Calderon
Cloud z/OS Integration Leader
rcalderon@us.ibm.com
zAcceleration
Red Hat Synergy
Worldwide IBM Z

Anna Shugol
Cloud Solutions Engineer
anna.shugol@ibm.com
zAcceleration
Red Hat Synergy
Worldwide IBM Z

Vic Cross
Cloud Solutions Engineer
viccross@au1.ibm.com
zAcceleration
Red Hat Synergy
Worldwide IBM Z