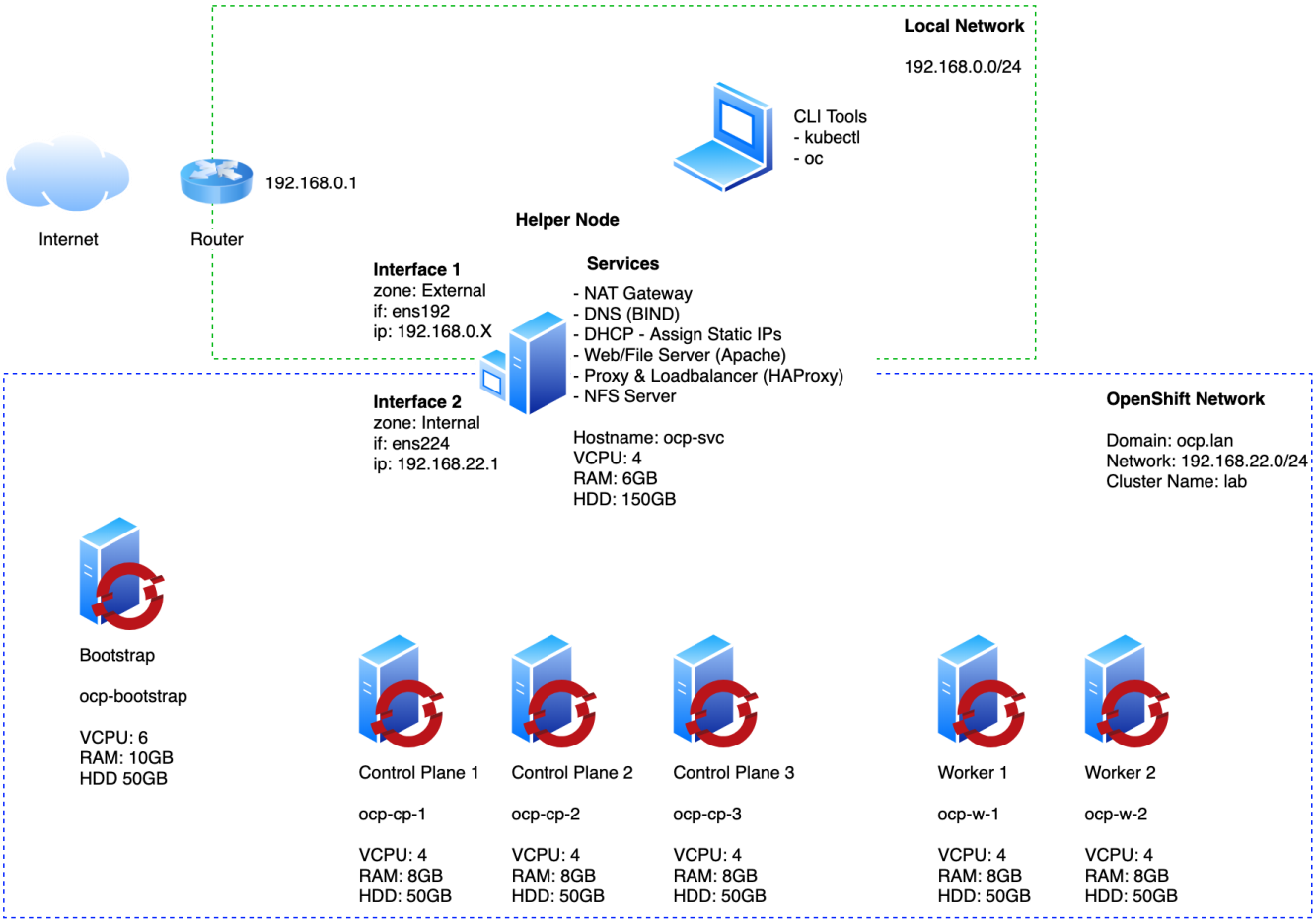
OPENSHIFT 4 BARE METAL INSTALL - UPI



If you use a vSphere version 6.5 instance, consider upgrading to 6.7U3 or 7.0 before you install OpenShift Container Platform

Also, if you plan to install OCS and Dynamic provisioning -

You must have at least three OpenShift Container Platform worker nodes in the cluster with locally attached storage devices on each of them.

Each of the three worker nodes must have at least one raw block device available to be used by OpenShift Container Storage.

For minimum starting node requirements -

CPU - 16

RAM - 64 GB

For dynamic storage deployments - 1 disk of size 0.5 TiB or 2 TiB or 4 TiB storage

Please refer - OpenShift - Install and configure OCS for Dynamic provisioning of persistent volumes using local storage

Please do the required changes accordingly -

Download Software

Download CentOS 8 x86_64 image

Login to RedHat OpenShift Cluster Manager (https://cloud.redhat.com/openshift)

Select 'Create Cluster' from the 'Clusters' navigation menu

Select 'RedHat OpenShift Container Platform'

Select 'Run on Bare Metal' → UPI

Download the following files:

Openshift Installer for Linux
Pull secret
Command Line Interface for Linux and your workstations OS
Red Hat Enterprise Linux CoreOS (RHCOS)
rhcos-X.X.X-x86_64-metal.x86_64.raw.gz
rhcos-X.X.X-x86_64-installer.x86_64.iso
Prepare the 'Bare Metal' environment
Copy the CentOS 8 iso to an ESXi datastore
Create a new Port Group called 'OCP' under Networking

IMP: Preferable to use a different vSwitch for this - this will avoid any conflicts with any of the existing DHCP servers

Create 3 Control Plane virtual machines with minimum settings:
Name: ocp-cp-# (Example ocp-cp-1)
4vcpu
8GB RAM
50GB HDD
NIC connected to the OCP network
Load the rhcos-X.X.X-x86_64-installer.x86_64.iso image into the CD/DVD drive

Create 2 Worker virtual machines (or more if you want) with minimum settings:
Name: ocp-w-# (Example ocp-w-1)
4vcpu
8GB RAM
50GB HDD
NIC connected to the OCP network
Load the rhcos-X.X.X-x86_64-installer.x86_64.iso image into the CD/DVD drive

Create a Bootstrap virtual machine (this vm will be deleted once installation completes) with minimum settings:
Name: ocp-boostrap
4vcpu
8GB RAM
50GB HDD
NIC connected to the OCP network
Load the rhcos-X.X.X-x86_64-installer.x86_64.iso image into the CD/DVD drive

Create a Services virtual machine with minimum settings:
Name: ocp-svc
4vcpu
4GB RAM
120GB HDD
NIC1 connected to the VM Network (LAN)
NIC2 connected to the OCP network
Load the CentOS_8.iso image into the CD/DVD drive

Boot all virtual machines so they each are assigned a MAC address
Shut down all virtual machines except for 'ocp-svc'
Use the VMware ESXi dashboard to record the MAC address of each vm, these will be used later to set static IPs
Configure Environmental Services
Install CentOS 8 on the ocp-svc host
Assign as much free storage to '/' as possible
Optionally you can install the 'Guest Tools' package to have monitoring and reporting in the VMware ESXi dashboard
Enable the LAN NIC only to obtain a DHCP address from the LAN network and make note of the IP address (ocp-svc_IP_address) assigned to the vm**
** For our labs - this will be the static IP address
Boot the ocp-svc VM

Move the files downloaded from the RedHat Cluster Manager site to the ocp-svc node :

Download ocp4-metal-install.tar (/root/ocp4-metal-install) for each of the services
(https://confluence.community.veritas.com/download/attachments/234347493/ocp4-metal-install.tar?version=1&modificationDate=1605858925000&api=v2)

scp ~/Downloads/openshift-install-linux.tar.gz ~/Downloads/openshift-client-linux.tar.gz ~/Downloads/rhcos-x.x.x-x86_64-installer.x86_64.iso ~/Downloads/ocp4-metal-install.tar root@{ocp-svc_IP_address}:/root/

tar xvf openshift-client-linux.tar.gz
mv oc kubectl /usr/local/bin

kubectl version
oc version

tar xvf openshift-install-linux.tar.gz
dnf update

Set a Static IP for OCP network interface `nmtui-edit ens224` or edit `/etc/sysconfig/network-scripts/ifcfg-ens224`
Address: 192.168.22.1
DNS Server: 127.0.0.1
Search domain: ocp.lan
Never use this network for default route
Automatically connect
(If changes aren't applied automatically you can bounce the NIC with `nmcli connection down ens224` and `nmcli connection up ens224`)

nmcli connection modify ens224 connection.zone internal
nmcli connection modify ens192 connection.zone external

firewall-cmd --get-active-zones

Set masquerading (source-nat) on the both zones.
So to give a quick example of source-nat - for packets leaving the external interface, which in this case is ens192 - after they have been routed they will have their source address altered to the interface address of ens192 so that return packets can find their way back to this interface where the reverse will happen.

firewall-cmd --zone=external --add-masquerade --permanent
firewall-cmd --zone=internal --add-masquerade --permanent

firewall-cmd --reload

firewall-cmd --list-all --zone=internal
firewall-cmd --list-all --zone=external

When masquerading is enabled so is ip forwarding which basically makes this host a router. Check:
cat /proc/sys/net/ipv4/ip_forward

dnf install bind bind-utils -y

cp ~/ocp4-metal-install/dns/named.conf /etc/named.conf
cp -R ~/ocp4-metal-install/dns/zones /etc/named/

IMP: The named.conf uses google forwarders, change it to use - our lab dns, as under:

```
        # Using Google DNS
        forwarders {
        172.16.8.32;
        172.16.8.33;
        };
```

Configure the firewall for DNS

```
firewall-cmd --add-port=53/udp --zone=internal --permanent
firewall-cmd --reload

systemctl enable named
systemctl start named
systemctl status named
```

At the moment DNS will still be pointing to the LAN DNS server. You can see this by testing with `dig ocp.lan`. Change the LAN nic (ens192) to use 127.0.0.1 for DNS AND ensure `Ignore automatically Obtained DNS parameters` is ticked

```
nmtui-edit ens192
systemctl restart NetworkManager
```

Confirm dig now sees the correct DNS results by using the DNS Server running locally

```
dig ocp.lan
# The following should return the answer ocp-bootstrap.lab.ocp.lan from the local server

dig -x 192.168.22.200

dnf install dhcp-server -y
```

Edit dhcpd.conf from the cloned git repo to have the correct mac address for each host and copy the conf file to the correct location for the DHCP service to use

```
cp ~/ocp4-metal-install/dhcpd.conf /etc/dhcp/dhcpd.conf
firewall-cmd --add-service=dhcp --zone=internal --permanent
firewall-cmd --reload

systemctl enable dhcpd
systemctl start dhcpd
systemctl status dhcpd

dnf install httpd -y
```

Change default listen port to 8080 in httpd.conf

```
sed -i 's/Listen 80/Listen 0.0.0.0:8080/' /etc/httpd/conf/httpd.conf
```

Configure the firewall for Web Server traffic

```
firewall-cmd --add-port=8080/tcp --zone=internal --permanent
firewall-cmd --reload

systemctl enable httpd
systemctl start httpd
systemctl status httpd
```

Making a GET request to localhost on port 8080 should now return the default Apache webpage
curl localhost:8080

dnf install haproxy -y
cp ~/ocp4-metal-install/haproxy.cfg /etc/haproxy/haproxy.cfg

Note: Opening port 9000 in the external zone allows access to HAProxy stats that are useful for monitoring and troubleshooting. The UI can be accessed at: `http://{ocp-svc_IP_address}:9000/stats`

```
firewall-cmd --add-port=6443/tcp --zone=internal --permanent          # kube-api-server on control plane nodes
firewall-cmd --add-port=6443/tcp --zone=external --permanent          # kube-api-server on control plane nodes
firewall-cmd --add-port=22623/tcp --zone=internal --permanent         # machine-config server
firewall-cmd --add-service=http --zone=internal --permanent           # web services hosted on worker nodes
firewall-cmd --add-service=http --zone=external --permanent           # web services hosted on worker nodes
firewall-cmd --add-service=https --zone=internal --permanent          # web services hosted on worker nodes
firewall-cmd --add-service=https --zone=external –permanent           # web services hosted on worker nodes
firewall-cmd --add-port=9000/tcp --zone=external --permanent          # HAProxy Stats
firewall-cmd –reload

setsebool -P haproxy_connect_any 1     # SELinux name_bind access
systemctl enable haproxy
systemctl start haproxy
systemctl status haproxy
```

Install and configure NFS for the OpenShift Registry. It is a requirement to provide storage for the Registry, emptyDir can be specified if necessary.

dnf install nfs-utils -y

Create the Share
Check available disk space and its location `df -h`

```
mkdir -p /shares/registry
chown -R nobody:nobody /shares/registry
chmod -R 777 /shares/registry

echo "/shares/registry  192.168.22.0/24(rw,sync,root_squash,no_subtree_check,no_wdelay)" > /etc/exports
exportfs -rv

firewall-cmd --zone=internal --add-service mountd --permanent
firewall-cmd --zone=internal --add-service rpc-bind --permanent
firewall-cmd --zone=internal --add-service nfs --permanent
firewall-cmd --reload

systemctl enable nfs-server rpcbind
systemctl start nfs-server rpcbind nfs-mountd
```

Generate and host install files
Generate an SSH key pair keeping all default options

```
ssh-keygen
mkdir ~/ocp-install
```

Copy the install-config.yaml included in the clones repository to the install directory

```
cp ~/ocp4-metal-install/install-config.yaml ~/ocp-install
```

Update the install-config.yaml with your own pull-secret and ssh key.

- Line 23 should contain the contents of your pull-secret.txt
- Line 24 should contain the contents of your '~/.ssh/id_rsa.pub'

vim ~/ocp-install/install-config.yaml

Generate Kubernetes manifest files
~/openshift-install create manifests --dir ~/ocp-install

(A warning is shown about making the control plane nodes schedulable. It is up to you if you want to run workloads on the Control Plane nodes. If you dont want to you can disable this with: `sed -i 's/mastersSchedulable: true/mastersSchedulable: false/' ~/ocp-install/manifests/cluster-scheduler-02-config.yml`. Make any other custom changes you like to the core Kubernetes manifest files.)

Generate the Ignition config and Kubernetes auth files
~/openshift-install create ignition-configs --dir ~/ocp-install/

Create a hosting directory to serve the configuration files for the OpenShift booting process
mkdir /var/www/html/ocp4

Copy all generated install files to the new web server directory
cp -R ~/ocp-install/* /var/www/html/ocp4

Move the Core OS image to the web server directory (later you need to type this path multiple times so it is a good idea to shorten the name)
mv ~/rhcos-X.X.X-x86_64-metal.x86_64.raw.gz /var/www/html/ocp4/rhcos

Change ownership and permissions of the web server directory
chcon -R -t httpd_sys_content_t /var/www/html/ocp4/
chown -R apache: /var/www/html/ocp4/
chmod 755 /var/www/html/ocp4/

Confirm you can see all files added to the `/var/www/html/ocp4/` dir through Apache
curl localhost:8080/ocp4/

Deploy OpenShift :

Power on the ocp-bootstrap host and ocp-cp-# hosts and select 'Tab' to enter boot configuration. Enter the following configuration:
Bootstrap Node - ocp-bootstrap
coreos.inst.install_dev=sda coreos.inst.image_url=http://192.168.22.1:8080/ocp4/rhcos
coreos.inst.ignition_url=http://192.168.22.1:8080/ocp4/bootstrap.ign coreos.inst.insecure=yes

Each of the Control Plane Nodes - ocp-cp-\#
coreos.inst.install_dev=sda coreos.inst.image_url=http://192.168.22.1:8080/ocp4/rhcos
coreos.inst.ignition_url=http://192.168.22.1:8080/ocp4/master.ign coreos.inst.insecure=yes
Power on the ocp-w-# hosts and select 'Tab' to enter boot configuration. Enter the following configuration:
Each of the Worker Nodes - ocp-w-\#
coreos.inst.install_dev=sda coreos.inst.image_url=http://192.168.22.1:8080/ocp4/rhcos
coreos.inst.ignition_url=http://192.168.22.1:8080/ocp4/worker.ign coreos.inst.insecure=yes

Monitor the Bootstrap Process

You can monitor the bootstrap process from the ocp-svc host at different log levels (debug, error, info)
~/openshift-install --dir ~/ocp-install wait-for bootstrap-complete --log-level=debug
Once bootstrapping is complete the ocp-boostrap node can be removed

Remove the Bootstrap Node

Remove all references to the `ocp-bootstrap` host from the `/etc/haproxy/haproxy.cfg` file

Two entries

vim /etc/haproxy/haproxy.cfg

Restart HAProxy - If you are still watching HAProxy stats console you will see that the ocp-boostrap host has been removed from the backends.

systemctl reload haproxy

The ocp-bootstrap host can now be safely shutdown and deleted from the VMware ESXi Console, the host is no longer required

Wait for installation to complete

IMPORTANT: if you set mastersSchedulable to false the worker nodes will need to be joined to the cluster to complete the installation. This is because the OpenShift Router will need to be scheduled on the worker nodes and it is a dependency for cluster operators such as ingress, console and authentication.

Collect the OpenShift Console address and kubeadmin credentials from the output of the install-complete event

~/openshift-install --dir ~/ocp-install wait-for install-complete

Continue to join the worker nodes to the cluster in a new tab whilst waiting for the above command to complete

Join Worker Nodes

Setup 'oc' and 'kubectl' clients on the ocp-svc machine

export KUBECONFIG=~/ocp-install/auth/kubeconfig

Test auth by viewing cluster nodes

oc get nodes

View and approve pending CSRs

Note: Once you approve the first set of CSRs additional 'kubelet-serving' CSRs will be created. These must be approved too. If you do not see pending requests wait until you do.

View CSRs

oc get csr

Approve all pending CSRs

oc get csr -o go-template='{{range .items}}{{if not .status}}{{.metadata.name}}{{"\n"}}{{end}}{{end}}' | xargs oc adm certificate approve

Wait for kubelet-serving CSRs and approve them too with the same command

oc get csr -o go-template='{{range .items}}{{if not .status}}{{.metadata.name}}{{"\n"}}{{end}}{{end}}' | xargs oc adm certificate approve

Watch and wait for the Worker Nodes to join the cluster and enter a 'Ready' status

This can take 5-10 minutes

watch -n5 oc get nodes

Configure storage for the Image Registry

A Bare Metal cluster does not by default provide storage so the Image Registry Operator bootstraps itself as 'Removed' so the installer can complete. As the installation has now completed storage can be added for the Registry and the operator updated to a 'Managed' state.

Create the 'image-registry-storage' PVC by updating the Image Registry operator config by updating the management state to 'Managed' and adding 'pvc' and 'claim' keys in the storage key:

oc edit configs.imageregistry.operator.openshift.io

```
    managementState: Managed
    storage:
      pvc:
        claim: # leave the claim blank
```

Confirm the 'image-registry-storage' pvc has been created and is currently in a 'Pending' state

oc get pvc -n openshift-image-registry

Create the persistent volume for the 'image-registry-storage' pvc to bind to

oc create -f ~/ocp4-metal-install/manifest/registry-pv.yaml

After a short wait the 'image-registry-storage' pvc should now be bound
oc get pvc -n openshift-image-registry

Create the first Admin user

Apply the `oauth-htpasswd.yaml` file to the cluster
This will create a user 'admin' with the password 'password'. To set a different username and password substitue the htpasswd key in the '~/ocp4-metal-install/manifest/oauth-htpasswd.yaml' file with the output of `htpasswd -n -B -b <username> <password>`
oc apply -f ~/ocp4-metal-install/manifest/oauth-htpasswd.yaml
Assign the new user (admin) admin permissions
oc adm policy add-cluster-role-to-user cluster-admin admin
Access the OpenShift Console
Wait for the 'console' Cluster Operator to become available
oc get co
Append the following to your local workstations `/etc/hosts` file:
From your local workstation If you do not want to add an entry for each new service made available on OpenShift you can configure the ocp-svc DNS server to serve externally and create a wildcard entry for *.apps.lab.ocp.lan

Open the hosts file
sudo vi /etc/hosts

Append the following entries:
<ocp-svc_IP_address> ocp-svc api.lab.ocp.lan console-openshift-console.apps.lab.ocp.lan oauth-openshift.apps.lab.ocp.lan downloads-openshift-console.apps.lab.ocp.lan alertmanager-main-openshift-monitoring.apps.lab.ocp.lan grafana-openshift-monitoring.apps.lab.ocp.lan prometheus-k8s-openshift-monitoring.apps.lab.ocp.lan thanos-querier-openshift-monitoring.apps.lab.ocp.lan
Navigate to the OpenShift Console URL and log in as the 'admin' user (https://console-openshift-console.apps.lab.ocp.lan/)

You will get self signed certificate warnings that you can ignore If you need to login as kubeadmin and need to the password again you can retrieve it with: `cat ~/ocp-install/auth/kubeadmin-password`
Troubleshooting
You can collect logs from all cluster hosts by running the following command from the 'ocp-svc' host:

./openshift-install gather bootstrap --dir ocp-install --bootstrap=192.168.22.200 --master=192.168.22.201 --master=192.168.22.202 --master=192.168.22.203

Modify the role of the Control Plane Nodes
If you would like to schedule workloads on the Control Plane nodes apply the 'worker' role by changing the value of 'mastersSchedulable' to true.
If you do not want to schedule workloads on the Control Plane nodes remove the 'worker' role by changing the value of 'mastersSchedulable' to false.
Remember depending on where you host your workloads you will have to update HAProxy to include or exclude the control plane nodes from the ingress backends.
oc edit schedulers.config.openshift.io cluster
Reference and due credits -
https://github.com/ryanhay/ocp4-metal-install