**:Open Shift Cluster Build:**

**Building openshift cluster on a bare metal server with user-provisioned-infrastructure method (UPI).**

**Prerequisites:**

1. Install the following on the helper:

* Git

#dnf update

#sudo dnf install -y git

* Ansible

#sudo dnf install -y ansible-core

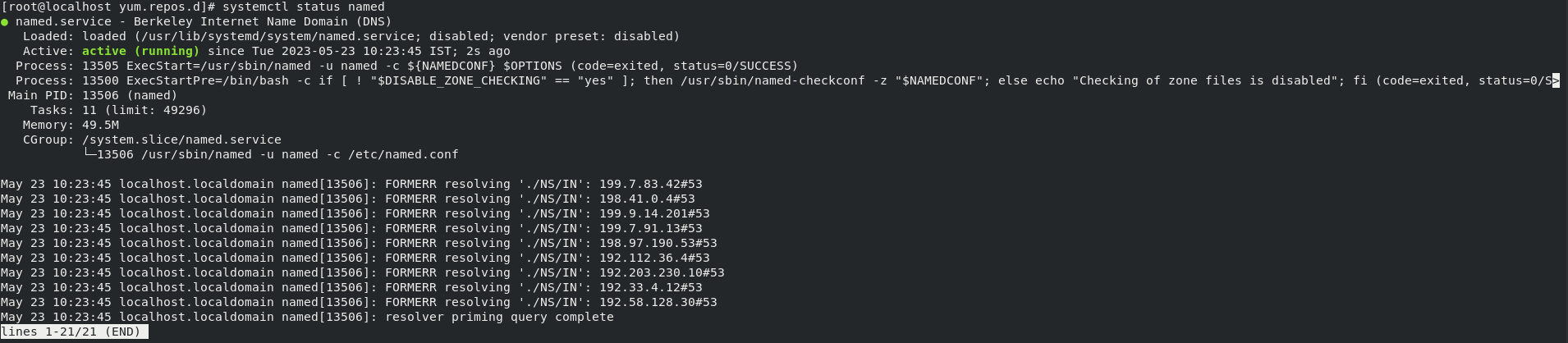
* Bind Package

#sudo dnf install -y bind

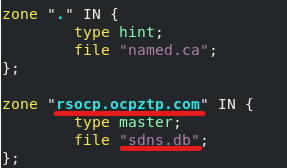
To configure git and ansible refer the “Using the playbook” section in the following link <https://github.com/redhat-cop/ocp4-helpernode>

2. Check if the named is active. Now make an entry in the named.conf file.

* #systemctl status named



* #vim /etc/named.conf



Here, “rsocp.ocpztp.com” is the name of the cluster, and “sdns.db” is the name of the file we make the DNS entry.

* In the options part of the file on the “listen-on port” line add your helper or DNS server IP



* In the options part of the file on the “allow-query” add “any” so that any query with the credentials will be allowed.



3. Now create the DNS file and make the entries.

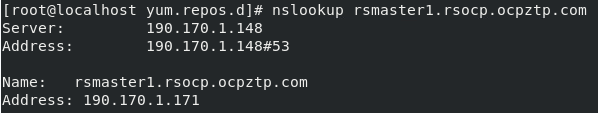
* #cd /var/named/
* #vim filename.db (sdns.db in our case)

Make the DNS entries like cluster name, api ip, api-init ip, apps ip, master and worker node’s name and ip as shown above.

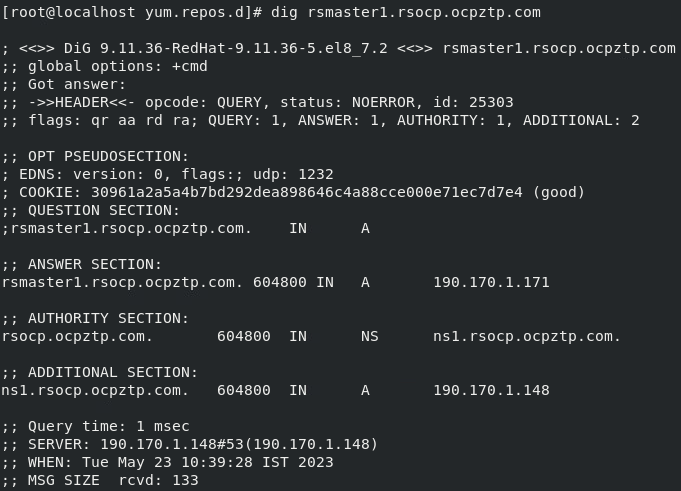
4. Now verify if the DNS is configured correctly.

* The named service should be up and running. To verify use “systemctl status named” (if it is inactive use “systemctl start named” to start the service).
* Run this command to get node details briefly

“#nslookup node.cluster.domain-name.com”



* Run this command to get node details “#dig node.cluster.domain-name.com”



5. Get the OC client openshift-installer and pull secret from the below link for latest version:

<https://access.redhat.com/downloads/content/290/ver=4.8/rhel---8/4.8.57/x86_64/product-software>

We can download any version based on our requirements. Here we have used 4.10.37:

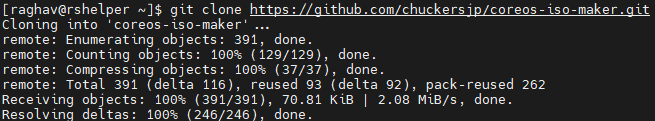
* [openshift-client-linux-4.10.37](https://mirror.openshift.com/pub/openshift-v4/clients/ocp/4.10.37/openshift-client-linux-4.10.37.tar.gz)
* [openshift-install-linux-4.10.37](https://mirror.openshift.com/pub/openshift-v4/clients/ocp/4.10.37/openshift-install-linux-4.10.37.tar.gz)
* [iso\_checksum](https://mirror.openshift.com/pub/openshift-v4/dependencies/rhcos/4.10/4.10.37/sha256sum.txt)

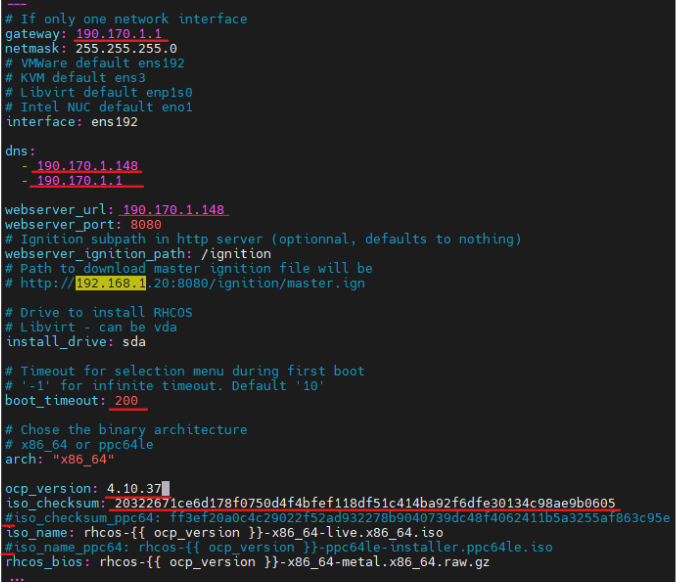
Now, we will have the openshift-client-linux-4.10.37, openshift-install-linux-4.10.37 which we have to unzip using the following commands:

* #tar -xvzf openshift-client-linux-4.10.37.tar.gz
* #tar -xvzf openshift-install-linux-4.10.37.tar.gz

Now give “which oc” command whit the output being a path. Now move the extracted files to that path, and verify by giving “oc” and “openshift-installer” commands.

6. Now we get a core-iso-maker to generate an iso image. Here we are getting the core-iso-maker from a git repository.

* #git clone [https://github.com/chuckersjp/coreos-iso-maker.g](https://github.com/chuckersjp/coreos-iso-maker.git)
* #cd coreos-iso-maker/
* #cd group\_vars/ (Here, we have the “all.yml” file)
* #vim all.yam



We get the checksum from “sha256sum.txt” file in the same directory in the above link.

The checksum we need will be in this format“rhcos-<version>-x86\_64-live.x86\_64.iso”

7. Now enter the cluster node details in the inventory.yml file in the core-iso-maker directory:

* #cd coreos-iso-maker/
* #vim inventory.yml

Make the following changes in the file.



8. Now to create the iso, run the “playbook-single.yml” ansible script located in the core-iso-maker directory:

* #cd core-iso-maker
* #ansible-playbook playbook-single.yml

(After the ansible is completed the iso will be available in the “/tmp” directory.)

Copy the iso to the Downloads directory from the /tmp directory.

* #cd /tmp
* #cp rhcos-install-cluster.iso /home/user/Downloads/

Now upload the iso (rhcos-install-cluster.iso) from the local machine to ESXI.

9. Now make a directory to make the “install-config.yaml” file and enter the cluster details, pull secret, and the ssh key:

* #mkdir dir-name
* #cd dir-name
* #vim install-config.yaml

Go to “[docs.openshift.com](https://docs.openshift.com/container-platform/4.10/installing/installing_bare_metal/installing-bare-metal.html)” and select the version and go to installation and select the installation method (Installing on bare metal) and type of installation (User-provisioned cluster) here you can use the sample install-config.yaml file to write your own.

* #ssh-keygen (generate an ssh key and put the public key in the install-c

Go to “console.redhat.com” and select Red Hat Openshift on the top left select “cluster”, click on create cluster and select “Datacenter” and select the Infrastructure provider in our case it is “[Bare Metal (x86\_64)](https://console.redhat.com/openshift/install/metal)”, now select “Full Control” now download/copy the pull secrete.

After generating the ssh-key and getting the pull secret enter these in the install-config.yaml file. Before proceeding further make a copy of the install-config.yaml file in the same directory and name it install-config.

10. Now we create the Kubernetes manifest and ignition config files.

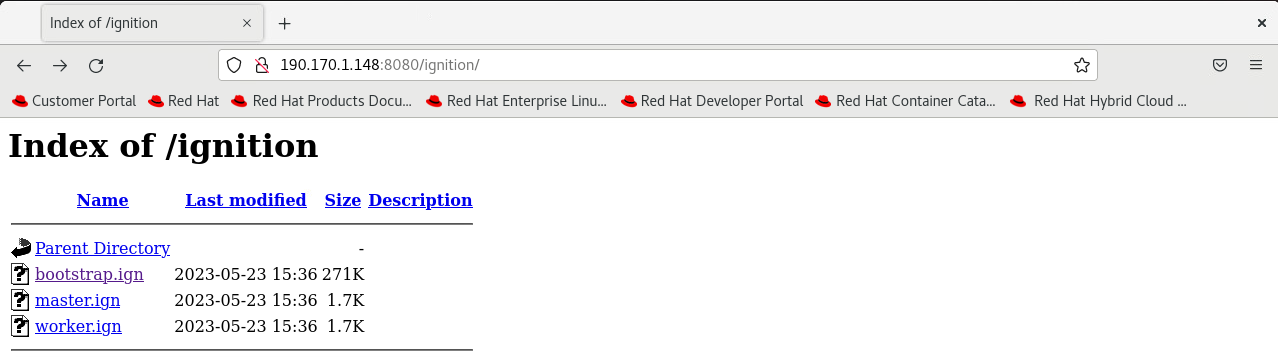
* #cd <dir with install-config.yaml file>
* #openshift-install create manifests --dir .
* Remove the Kubernetes manifest file if there is any in the manifest or openshift directories that we created by the above command.:

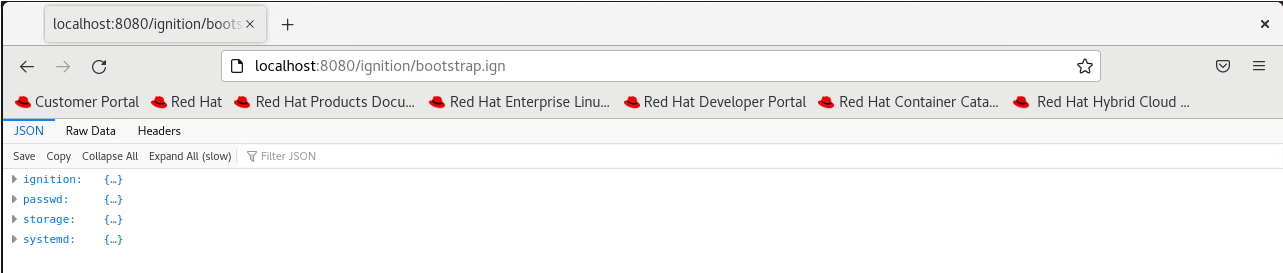
#rm -f OpenShift/99\_openshift-cluster-api\_master-machines-\*.yaml

#rm -f OpenShift/99\_openshift-cluster-api\_worker-machineset-\*.yaml

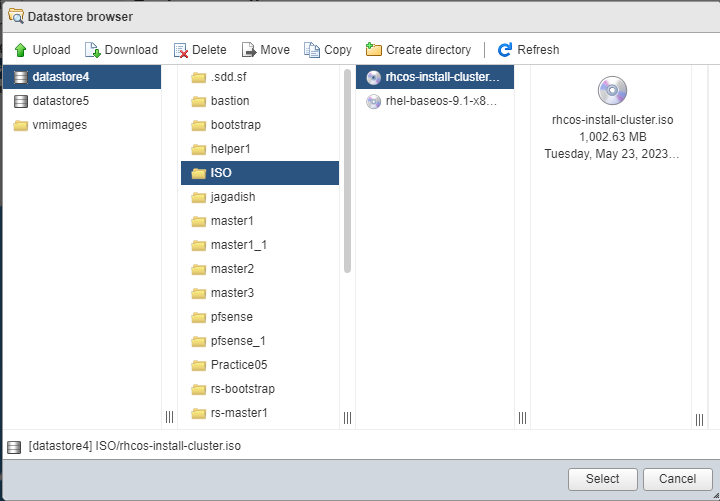
* #openshift-install create ignition-configs --dir <installation\_directory>
* #yum install -y httpd
* #systemctl enable –now httpd
* #mkdir /var/www/html/ignition
* #cp /home/raghav/rscluster\*.ign /var/www/html/ignition/
* #chmod 755 \*.ign
* #vim /etc/httpd/conf/httpd.conf (change the listening port from 80 to 8080)

Now in our helper search “localhost:8080/ignition” we should be able to see the page below:





11. Now create a VM for our bootstrap and select the cluster iso we generated earlier in step-8 and power on the bootstrap VM and select the bootstrap kernel while the VM is booting and create more VMs for your masters and workers and upload the same cluster iso for the masters and workers and select the respective kernels and let them boot up.





12. Once all the masters and workers are done booting up, go to your helper and verify if all

The output of the above command should be as shown in the above image containing the details of all the nodes.

13. If you are unable to view any nodes in the output of the above command that may be because the csr certificates might be pending to verify this use the following command:

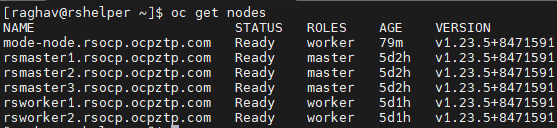
* #oc get csr (This command gives you all the csr that are approved and pending.)
* #oc get csr | grep -i pending (This command gives you all the pending csr certificates)
* #oc adm certificate approve <csr-certificate-1> <csr-certificate-2> <csr-certificate-3>

(The above command approves the csr certificates)the nodes are visible with the following command:

* #export KUBECONFIG=/home/user/cluster-dir/auth/kubeconfig

(This command exports the kubeconfig file so that we can use openshift command.)

* #oc get nodes



Now you should be able to see all your node details in the “oc get nodes” command output.