:Open Shift Cluster Build:

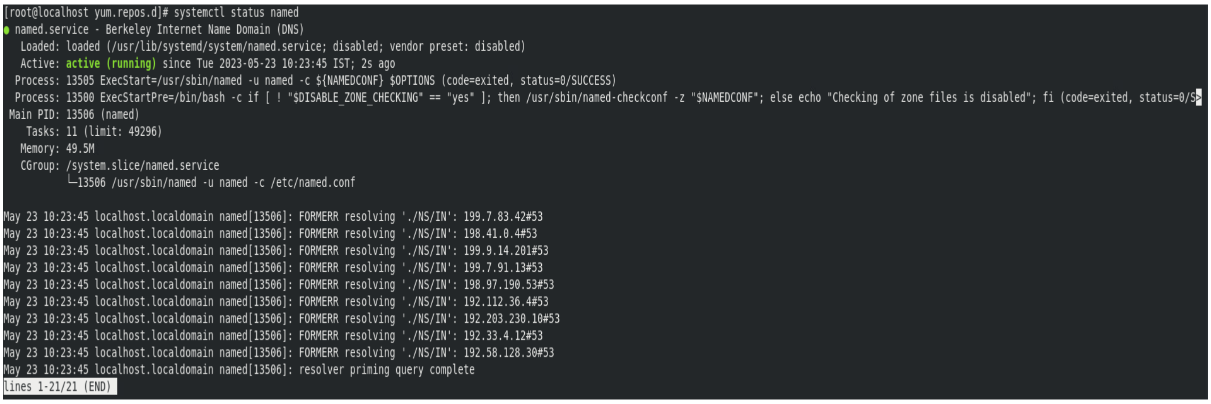
Building openshift cluster on a bare metal server with user-provisioned-infrastructure method (UPI).

1. Install the following on the helper:

- Git
- Ansible
- Bind Package

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

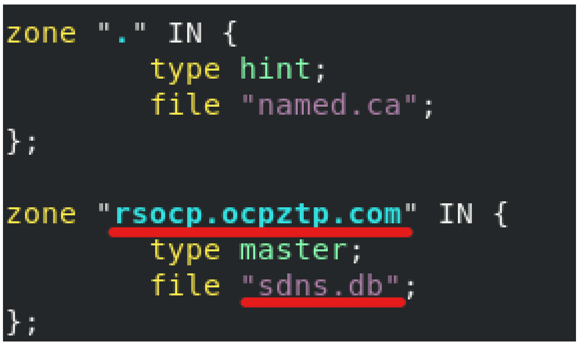
- #systemctl status named



```
[root@localhost yum.repos.d]# systemctl status named
● named.service - Berkeley Internet Name Domain (DNS)
   Loaded: loaded (/usr/lib/systemd/system/named.service; disabled; vendor preset: disabled)
   Active: active (running) since Tue 2023-05-23 10:23:45 IST; 2s ago
  Process: 13505 ExecStart=/usr/sbin/named -u named -c ${NAMEDCONF} $OPTIONS (code=exited, status=0/SUCCESS)
  Process: 13500 ExecStartPre=/bin/bash -c if [ ! "$DISABLE_ZONE_CHECKING" == "yes" ]; then /usr/sbin/named-checkconf -z "$NAMEDCONF"; else echo "Checking of zone files is disabled"; fi (code=exited, status=0/S>
 Main PID: 13506 (named)
    Tasks: 11 (limit: 49296)
   Memory: 49.5M
   CGroup: /system.slice/named.service
           └─13506 /usr/sbin/named -u named -c /etc/named.conf

May 23 10:23:45 localhost.localdomain named[13506]: FORMERR resolving './NS/IN': 199.7.83.42#53
May 23 10:23:45 localhost.localdomain named[13506]: FORMERR resolving './NS/IN': 198.41.0.4#53
May 23 10:23:45 localhost.localdomain named[13506]: FORMERR resolving './NS/IN': 199.9.14.201#53
May 23 10:23:45 localhost.localdomain named[13506]: FORMERR resolving './NS/IN': 199.7.91.13#53
May 23 10:23:45 localhost.localdomain named[13506]: FORMERR resolving './NS/IN': 198.97.190.53#53
May 23 10:23:45 localhost.localdomain named[13506]: FORMERR resolving './NS/IN': 192.112.36.4#53
May 23 10:23:45 localhost.localdomain named[13506]: FORMERR resolving './NS/IN': 192.203.230.10#53
May 23 10:23:45 localhost.localdomain named[13506]: FORMERR resolving './NS/IN': 192.33.4.12#53
May 23 10:23:45 localhost.localdomain named[13506]: FORMERR resolving './NS/IN': 192.58.128.30#53
May 23 10:23:45 localhost.localdomain named[13506]: resolver priming query complete
lines 1-21/21 (END)
```

0

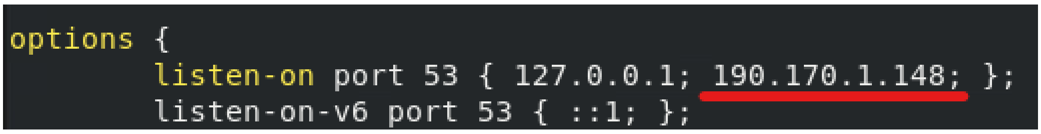
- #vim /etc/named.conf



```
zone "." IN {
        type hint;
        file "named.ca";
};

zone "rsocp.ocpztp.com" IN {
        type master;
        file "sdns.db";
};
```

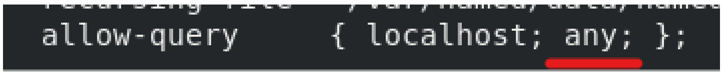
Here, “rsocp.ocpztp.com” is the name of the cluster, and “sdns.db” is the name of the file we make the DNS entry.

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



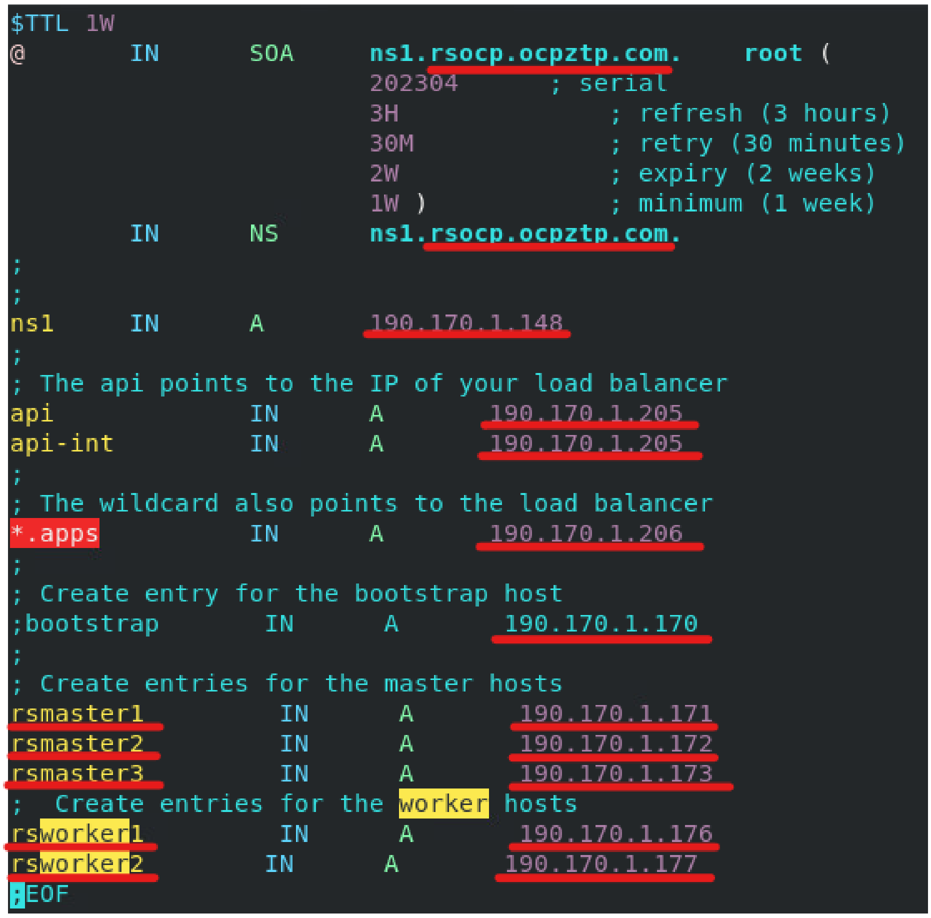
```
options {
        listen-on port 53 { 127.0.0.1; 190.170.1.148; };
        listen-on-v6 port 53 { ::1; };
```

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



```
recursing-file  "/var/named/data/named
allow-query     { localhost; any; };
```

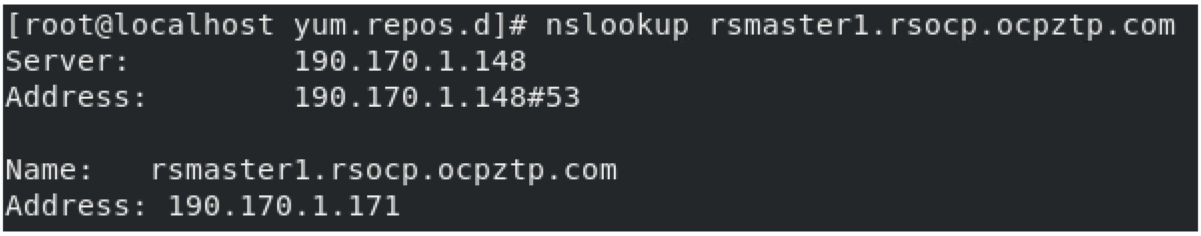
3. Now create the DNS file and make the entries.
   - #cd /var/named/
   - #vim filename.db (sdns.db in our case)



```
$TTL 1W
@       IN      SOA     ns1.rsocp.ocpztp.com.    root (
                        202304      ; serial
                        3H              ; refresh (3 hours)
                        30M             ; retry (30 minutes)
                        2W              ; expiry (2 weeks)
                        1W )            ; minimum (1 week)
        IN      NS      ns1.rsocp.ocpztp.com.
;
;
ns1     IN      A       190.170.1.148
;
; The api points to the IP of your load balancer
api             IN      A       190.170.1.205
api-int         IN      A       190.170.1.205
;
; The wildcard also points to the load balancer
*.apps          IN      A       190.170.1.206
;
; Create entry for the bootstrap host
;bootstrap       IN      A       190.170.1.170
;
; Create entries for the master hosts
rsmaster1         IN      A       190.170.1.171
rsmaster2         IN      A       190.170.1.172
rsmaster3         IN      A       190.170.1.173
;  Create entries for the worker hosts
rsworker1         IN      A       190.170.1.176
rsworker2        IN      A       190.170.1.177
;EOF
```

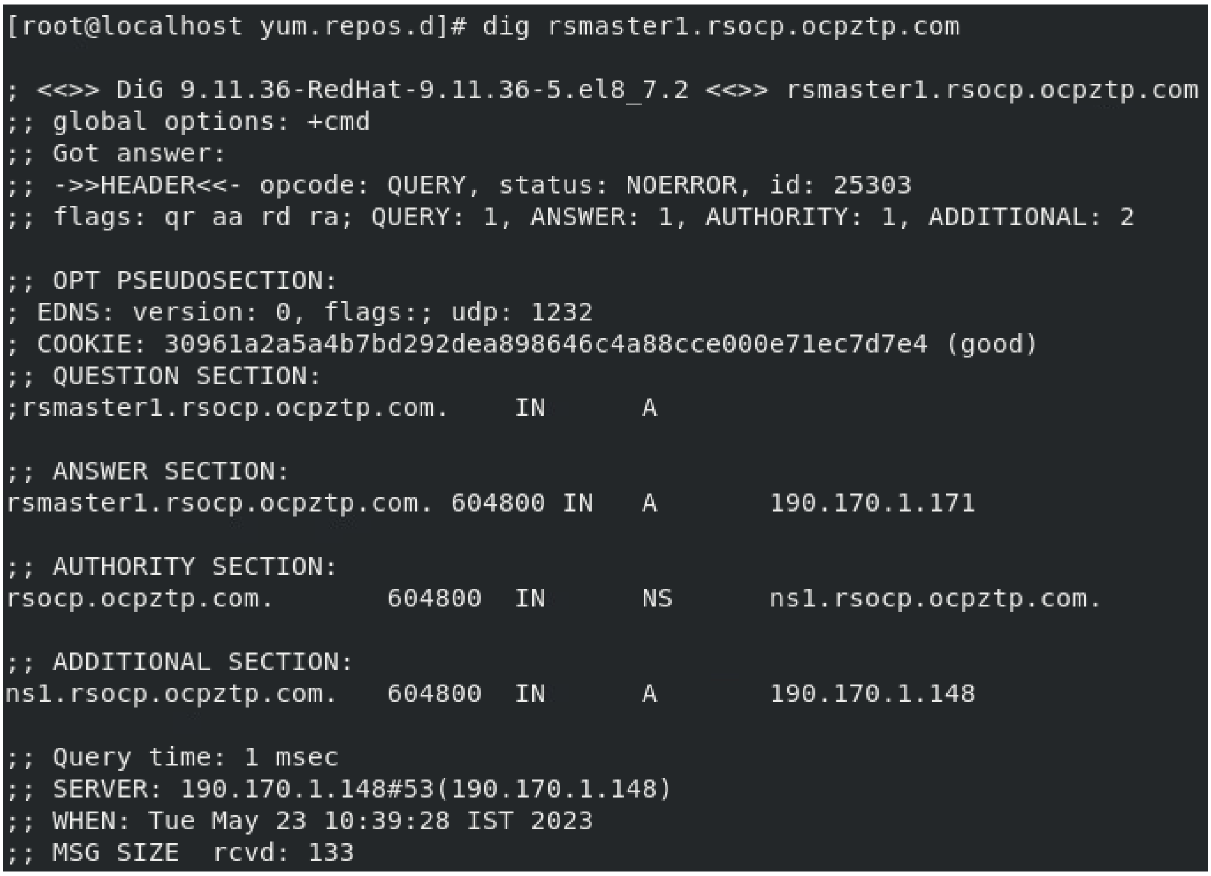
Make the DNS entries like cluster name, api ip, api-init ip, apps ip, master and worker node's name and ip as shown above.

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"



```
[root@localhost yum.repos.d]# nslookup rsmaster1.rsocp.ocpztp.com
Server:         190.170.1.148
Address:        190.170.1.148#53

Name:   rsmaster1.rsocp.ocpztp.com
Address: 190.170.1.171
```

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



```
[root@localhost yum.repos.d]# dig rsmaster1.rsocp.ocpztp.com

; <<>> DiG 9.11.36-RedHat-9.11.36-5.el8_7.2 <<>> rsmaster1.rsocp.ocpztp.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 25303
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 30961a2a5a4b7bd292dea898646c4a88cce000e71ec7d7e4 (good)
;; QUESTION SECTION:
;rsmaster1.rsocp.ocpztp.com.    IN      A

;; ANSWER SECTION:
rsmaster1.rsocp.ocpztp.com. 604800 IN   A       190.170.1.171

;; AUTHORITY SECTION:
rsocp.ocpztp.com.       604800  IN      NS      ns1.rsocp.ocpztp.com.

;; ADDITIONAL SECTION:
ns1.rsocp.ocpztp.com.   604800  IN      A       190.170.1.148

;; Query time: 1 msec
;; SERVER: 190.170.1.148#53(190.170.1.148)
;; WHEN: Tue May 23 10:39:28 IST 2023
;; MSG SIZE  rcvd: 133
```

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 down any version based on our requirements. Here we have used 4.10.37:

- openshift-client-linux-4.10.37 link: https://mirror.openshift.com/pub/openshift-v4/x86_64/clients/ocp/4.10.37/openshift-client-linux-4.10.37.tar.gz
- openshift-install-linux-4.10.37 link: https://mirror.openshift.com/pub/openshift-v4/x86_64/clients/ocp/4.10.37/openshift-install-linux-4.10.37.tar.gz

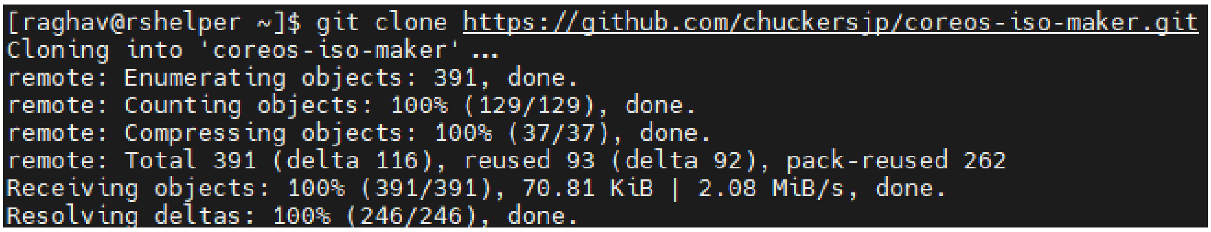
Now, we will have the openshift-client-linux-4.10.37, openshift-install-linux-4.10.37 which we have to unzip using the following commands:

- #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.

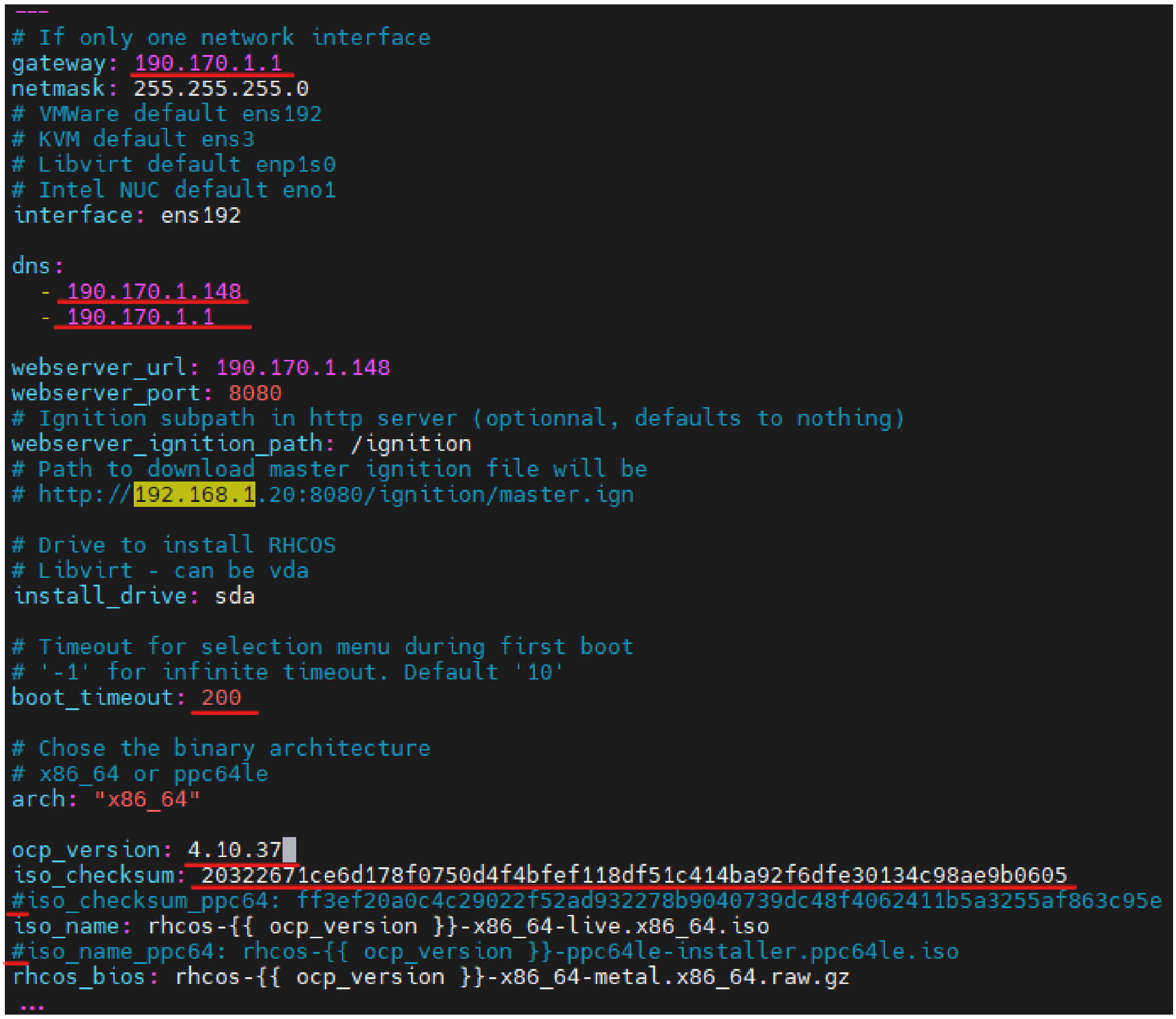
6. Now we get a core-iso-maker to generate an iso image. Here we are getting the core-iso-maker from a git repository.

- #git clone https://github.com/chuckersjp/coreos-iso-maker.g



```
[raghav@rshelper ~]$ git clone https://github.com/chuckersjp/coreos-iso-maker.git
Cloning into 'coreos-iso-maker'...
remote: Enumerating objects: 391, done.
remote: Counting objects: 100% (129/129), done.
remote: Compressing objects: 100% (37/37), done.
remote: Total 391 (delta 116), reused 93 (delta 92), pack-reused 262
Receiving objects: 100% (391/391), 70.81 KiB | 2.08 MiB/s, done.
Resolving deltas: 100% (246/246), done.
```

- #cd coreos-iso-maker/
- #cd group_vars/ (Here, we have the “all.yml” file)
- #vim all.yam



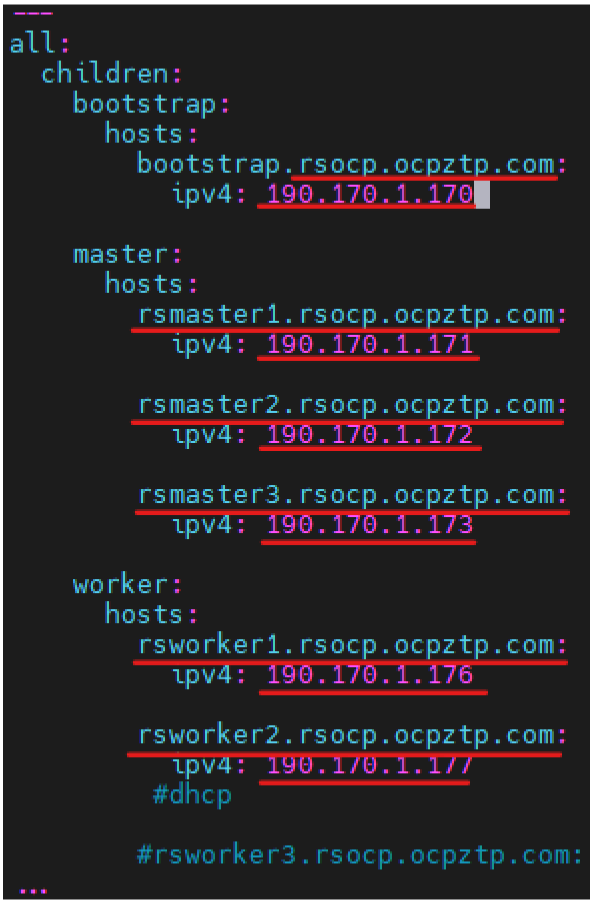
```
---
# If only one network interface
gateway: 190.170.1.1
netmask: 255.255.255.0
# VMWare default ens192
# KVM default ens3
# Libvirt default enp1s0
# Intel NUC default eno1
interface: ens192

dns:
  - 190.170.1.148
  - 190.170.1.1

webserver_url: 190.170.1.148
webserver_port: 8080
# Ignition subpath in http server (optionnal, defaults to nothing)
webserver_ignition_path: /ignition
# Path to download master ignition file will be
# http://192.168.1.20:8080/ignition/master.ign

# Drive to install RHCOS
# Libvirt - can be vda
install_drive: sda

# Timeout for selection menu during first boot
# '-1' for infinite timeout. Default '10'
boot_timeout: 200

# Chose the binary architecture
# x86_64 or ppc64le
arch: "x86_64"

ocp_version: 4.10.37
iso_checksum: 20322671ce6d178f0750d4f4bfef118df51c414ba92f6dfe30134c98ae9b0605
#iso_checksum_ppc64: ff3ef20a0c4c29022f52ad932278b9040739dc48f4062411b5a3255af863c95e
iso_name: rhcos-{{ ocp_version }}-x86_64-live.x86_64.iso
#iso_name_ppc64: rhcos-{{ ocp_version }}-ppc64le-installer.ppc64le.iso
rhcos_bios: rhcos-{{ ocp_version }}-x86_64-metal.x86_64.raw.gz
...
```

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.



```
---
all:
  children:
    bootstrap:
      hosts:
        bootstrap.rsocp.ocpztp.com:
          ipv4: 190.170.1.170

    master:
      hosts:
        rsmaster1.rsocp.ocpztp.com:
          ipv4: 190.170.1.171

        rsmaster2.rsocp.ocpztp.com:
          ipv4: 190.170.1.172

        rsmaster3.rsocp.ocpztp.com:
          ipv4: 190.170.1.173

    worker:
      hosts:
        rsworker1.rsocp.ocpztp.com:
          ipv4: 190.170.1.176

        rsworker2.rsocp.ocpztp.com:
          ipv4: 190.170.1.177
         #dhcp

        #rsworker3.rsocp.ocpztp.com:
...
```

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” 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

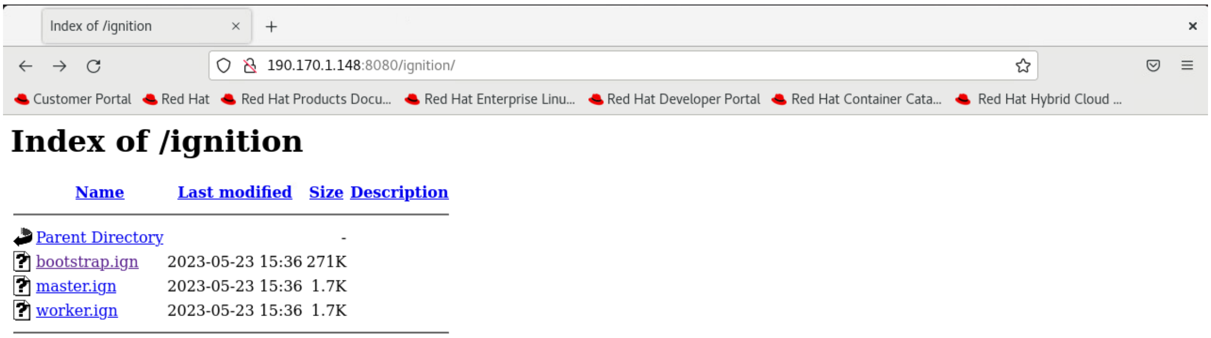
Go to “console.redhat.com” and select Red Hat Openshift on the top left select “cluster”, click on create cluster and select “Datacenter” and select the Infrastructure provider in our case it is “Bare Metal (x86_64)”, now select “Full Control” now download/copy the pull secrete.

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.

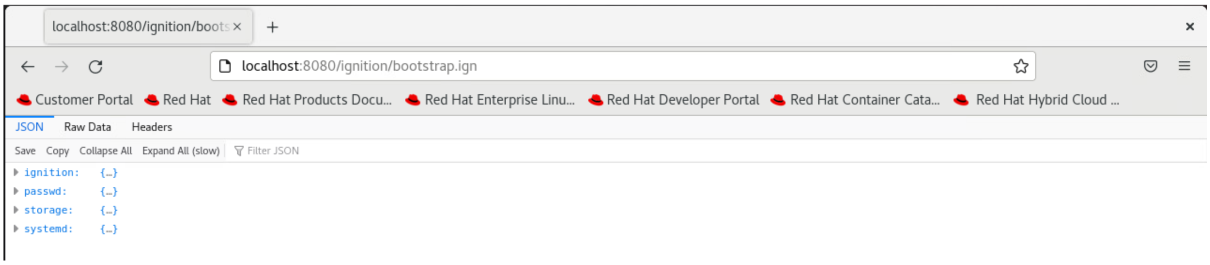
- #openshift-install create manifests --dir <installation_directory>
- Remove the Kubernetes manifest file if there is any in the manifest or openshift directories that we created by the above command.:
  #rm -f OpenShift/99_openshift-cluster-api_master-machines-*.yaml
  #rm -f OpenShift/99_openshift-cluster-api_worker-machineset-*.yaml
- #openshift-install create ignition-configs --dir <installation_directory>
- #yum install -y httpd
- #systemctl enable –now httpd
- #mkdir 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” we should be able to see the page below:



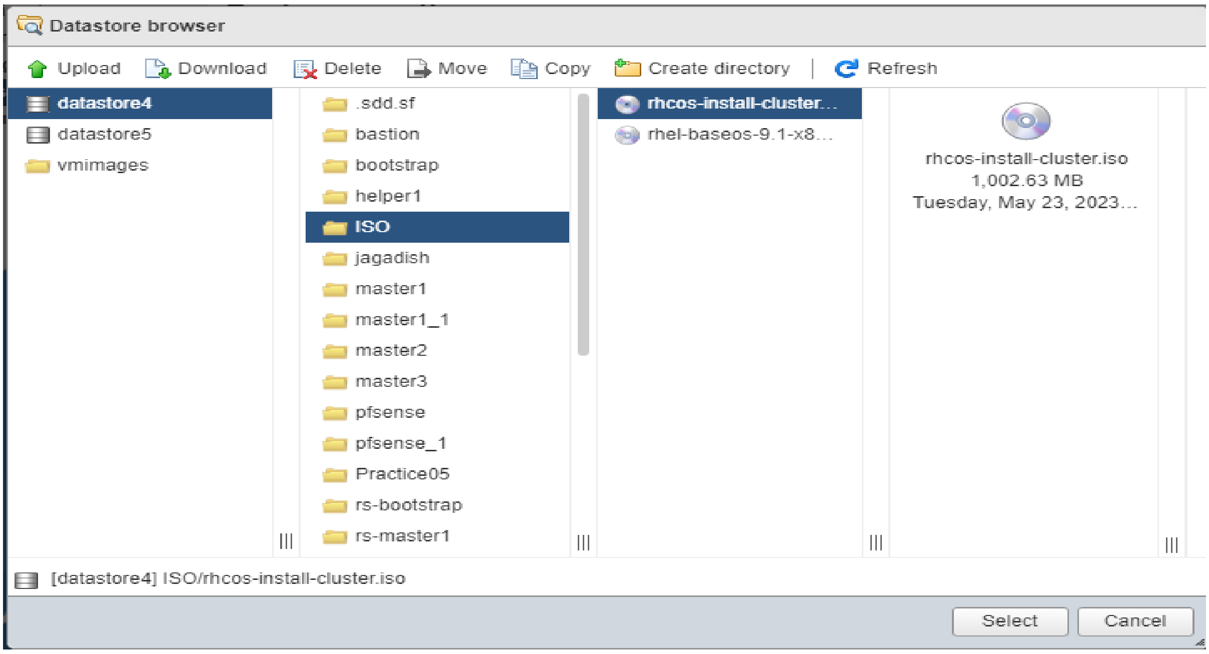
Index of /ignition

Name	Last modified	Size	Description
Parent Directory		-	
bootstrap.ign	2023-05-23 15:36	271K	
master.ign	2023-05-23 15:36	1.7K	
worker.ign	2023-05-23 15:36	1.7K	



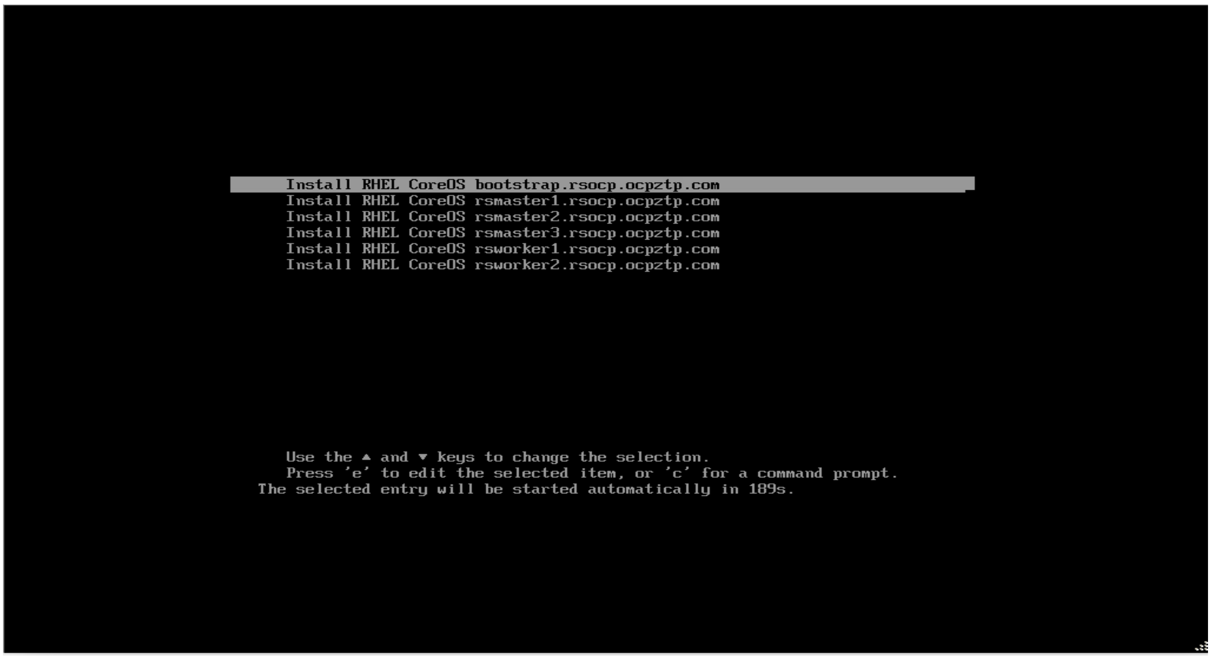
```
ignition:   {…}
passwd:     {…}
storage:    {…}
systemd:    {…}
```

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.



rhcos-install-cluster.iso
1,002.63 MB
Tuesday, May 23, 2023...

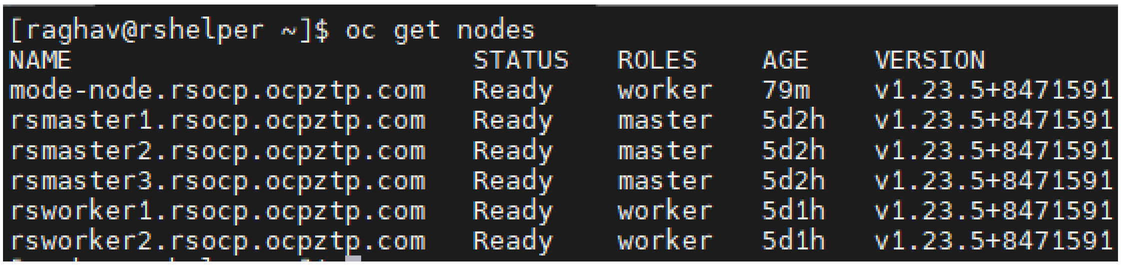
[datastore4] ISO/rhcos-install-cluster.iso



```
Install RHEL CoreOS bootstrap.rsocp.ocpztp.com
Install RHEL CoreOS rsmaster1.rsocp.ocpztp.com
Install RHEL CoreOS rsmaster2.rsocp.ocpztp.com
Install RHEL CoreOS rsmaster3.rsocp.ocpztp.com
Install RHEL CoreOS rsworker1.rsocp.ocpztp.com
Install RHEL CoreOS rsworker2.rsocp.ocpztp.com

  Use the ▲ and ▼ keys to change the selection.
  Press 'e' to edit the selected item, or 'c' for a command prompt.
The selected entry will be started automatically in 189s.
```

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



```
[raghav@rshelper ~]$ oc get nodes
NAME                          STATUS   ROLES    AGE    VERSION
mode-node.rsocp.ocpztp.com    Ready    worker   79m    v1.23.5+8471591
rsmaster1.rsocp.ocpztp.com    Ready    master   5d2h   v1.23.5+8471591
rsmaster2.rsocp.ocpztp.com    Ready    master   5d2h   v1.23.5+8471591
rsmaster3.rsocp.ocpztp.com    Ready    master   5d2h   v1.23.5+8471591
rsworker1.rsocp.ocpztp.com    Ready    worker   5d1h   v1.23.5+8471591
rsworker2.rsocp.ocpztp.com    Ready    worker   5d1h   v1.23.5+8471591
```

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)

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