**Cloud Native Installation Document**

1. Kubernetes supported version is 1.21.x for PS6. Verify this.

Graphical user interface, text

Description automatically generated

1. Install a Docker Engine and container runtime supported by Kubernetes. (This is already present/preinstalled with Kubernetes cluster). Please confirm in the Kubernetes cluster ( Master and worker nodes), container engine docker is installed.

Graphical user interface, text

Description automatically generated

1. Check docker version on jumphost.

brm@ssgc-jumphost-01:~$ docker --version

Docker version 20.10.7, build 20.10.7-0ubuntu5~18.04.3

1. Install Helm. Follow below steps:-

a) Download the Helm software from https://github.com/helm/helm/releases.

For the list of supported Helm versions, see "BRM Cloud Native Deployment

Software Compatibility" in BRM Compatibility Matrix.

wget <https://get.helm.sh/helm-v3.6.1-linux-amd64.tar.gz>

b) Extract the Helm files from the archive:

tar -zxvf helm-version-linux-amd64.tar.gz

where version is the Helm version number.

c) Find the helm binary in the unpacked directory and then move it to your desired

directory. For example:

mv linux-amd64/helm /usr/local/bin/helm

d) Check Helm Version

helm version

1. Download the BRM Cloud Native Deployment Package for Patch Set 6 from Oracle support (https://support.oracle.com).

* Patch 33888870: For BRM cloud native packages such as BRM server, BRM REST Services Manager, Billing Care, Business Operations Center, and Pipeline Configuration Center
* Patch 33888904: For PDC cloud native packages.
* Patch 33888886: For ECE cloud native packages

1. Extracting the Helm Charts

tar xvzf oc-cn-helm-chart-12.0.0.x.0.tgz

tar xvzf oc-cn-ece-helm-chart-12.0.0.x.0.tgz

tar xvzf oc-cn-op-job-helm-chart-12.0.0.x.0.tgz

tar xvzf oc-cn-init-db-helm-chart-12.0.0.x.0.tgz

1. Upload all required images to the registry.
2. Upload images in shared repository/registry as below :-

docker load --input oc-cn-brm-cm-12.0.0.x.0.tar

1. Verify that image is loaded correctly by entering this command:

docker images cm:12.0.0.x.0

1. Tag the image with the registry server by entering this command:

docker tag cm:12.0.0.x.0 RepoHost:RepoPort/cm:12.0.0.x.0

1. Push the image to the registry server by entering this command:

docker push RepoHost:RepoPort/cm:12.0.0.x.0

Below is the sample for pushing one image

**Oc-cn-brm-cm-12.0.0.6.0.tar:**

1. sudo docker load --input oc-cn-brm-cm-12.0.0.6.0.tar  
   A picture containing table

   Description automatically generated
2. sudo docker images cm:12.0.0.6.  
    **Text

   Description automatically generated**
3. Tag  
   - docker tag cm:12.0.0.6.0 asia-southeast1-docker.pkg.dev/project-launchpad-337407/brm-cloud/cm:12.0.0.6.0

4. Push -   
docker push asia-southeast1-docker.pkg.dev/project-launchpad-337407/brm-cloud/cm:12.0.0.6.0  
  
Text

Description automatically generated

**This is for docker brm init db:-**

docker load --input oc-cn-brm-init-db-12.0.0.6.0.tar

docker images init\_db:12.0.0.6.0

docker tag init\_db:12.0.0.6.0 asia-southeast1-docker.pkg.dev/project-launchpad-337407/brm-cloud/init\_db:12.0.0.6.0

docker push asia-southeast1-docker.pkg.dev/project-launchpad-337407/brm-cloud/init\_db:12.0.0.6.0

**For oc-cn-brm-eai-js-12.0.0.6.0.tar:-**

sudo docker load --input oc-cn-brm-eai-js-12.0.0.6.0.tar

sudo docker images eai\_js:12.0.0.6.0

sudo docker tag eai\_js:12.0.0.6.0 asia-southeast1-docker.pkg.dev/project-launchpad-337407/brm-cloud/eai\_js:12.0.0.6.0

sudo docker push asia-southeast1-docker.pkg.dev/project-launchpad-337407/brm-cloud/eai\_js:12.0.0.6.0

**For oc-cn-brm-dm-oracle-12.0.0.6.0.tar:-**

sudo docker load --input oc-cn-brm-dm-oracle-12.0.0.6.0.tar

sudo docker images dm\_oracle:12.0.0.6.0

sudo docker tag dm\_oracle:12.0.0.6.0 asia-southeast1-docker.pkg.dev/project-launchpad-337407/brm-cloud/dm\_oracle:12.0.0.6.0

sudo docker push asia-southeast1-docker.pkg.dev/project-launchpad-337407/brm-cloud/dm\_oracle:12.0.0.6.0

**Other images loaded are :-**

**oc-cn-brm-dm-ifw-sync-12.0.0.6.0.tar**

**oc-cn-brm-dm-aq-12.0.0.6.0.tar**

**oc-cn-brm-config-jobs-12.0.0.6.0.tar**

**oc-cn-brm-upgrade-12.0.0.6.0.tar**

**oc-cn-brm-rem-12.0.0.6.0.tar**

**oc-cn-brm-apps-12.0.0.6.0.tar**

**oc-cn-brm-dm-kafka-12.0.0.6.0.tar**

**oc-cn-brm-rel-12.0.0.6.0.tar**

**Load PDC Images :-**

**For oc-cn-pdcapp-12.0.0.6.0.tar :-**

sudo docker load --input oc-cn-pdcapp-12.0.0.6.0.tar

sudo docker images oracle/pdcapp:12.0.0.6.0

sudo docker tag oracle/pdcapp:12.0.0.6.0 asia-southeast1-docker.pkg.dev/project-launchpad-337407/brm-cloud/oracle/pdcapp:12.0.0.6.0

sudo docker push asia-southeast1-docker.pkg.dev/project-launchpad-337407/brm-cloud/oracle/pdcapp:12.0.0.6.0 ****

**Other images loaded are :-**

**oc-cn-pdcbrminteg-12.0.0.6.0.tar**

and **oc-cn-pdcie-12.0.0.6.0.tar**.

**Load ECE Images:-**

sudo docker load --input oc-cn-ece-12.0.0.6.0.tar

sudo docker images oc-cn-ece:12.0.0.6.0

sudo docker tag oc-cn-ece:12.0.0.6.0 asia-southeast1-docker.pkg.dev/project-launchpad-337407/brm-cloud/oc-cn-ece:12.0.0.6.0

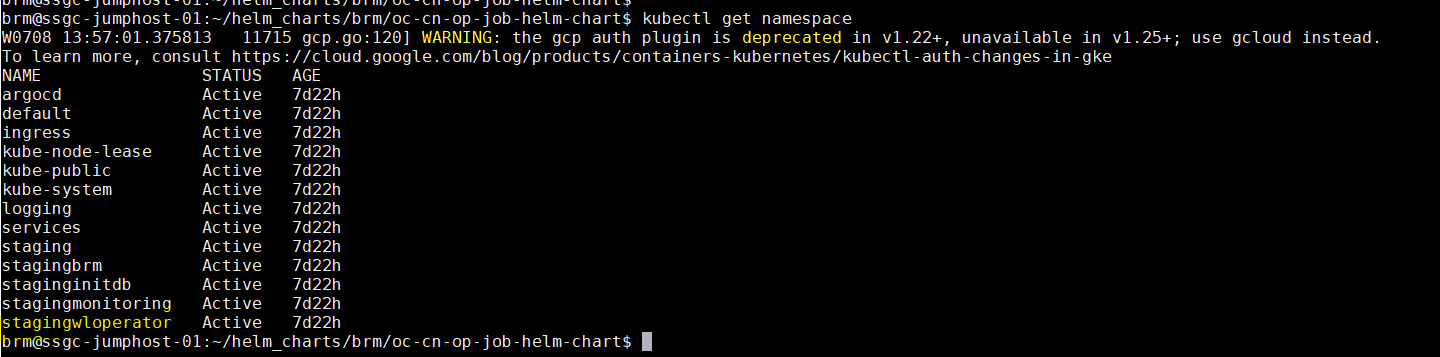
sudo docker push asia-southeast1-docker.pkg.dev/project-launchpad-337407/brm-cloud/oc-cn-ece:12.0.0.6.0

**Prerequisites Software :-**

1. **Kubernetes Weblogic operator :-**

**Note :- Better to download the helm of weblogic-operator and install it (Steps defined in 2nd method). Update the domainNamespace in its helm to the namespace used for brm,pdc helm before installing this. Here we did by referring the github.**

Check one namespace is created/present to install this. Here it is “stagingwloperator”.



1. Download weblogic operator using below command:-

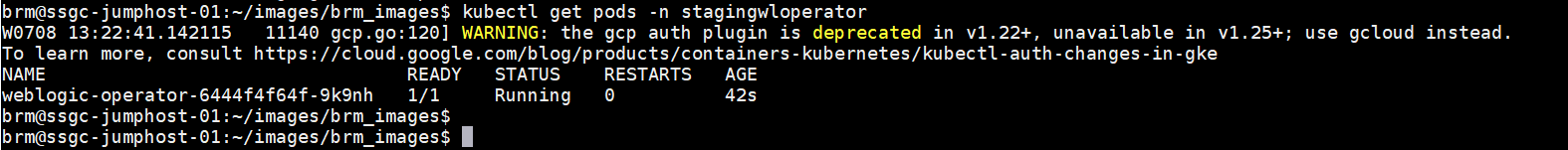
helm repo add weblogic-operator https://oracle.github.io/weblogic-kubernetes-operator/charts

1. Install weblogic operator helm using below command :-

helm install weblogic-operator weblogic-operator/weblogic-operator --namespace stagingwloperator --version 3.3.7

where, 3.3.7 is version of weblogic-operator

stagingwloperator is namespace for weblogic operator

****

**2nd way to install weblogic-operator (Preferred way)**

1) To pull the image to docker from link

sudo docker pull ghcr.io/oracle/weblogic-kubernetes-operator:3.4.3

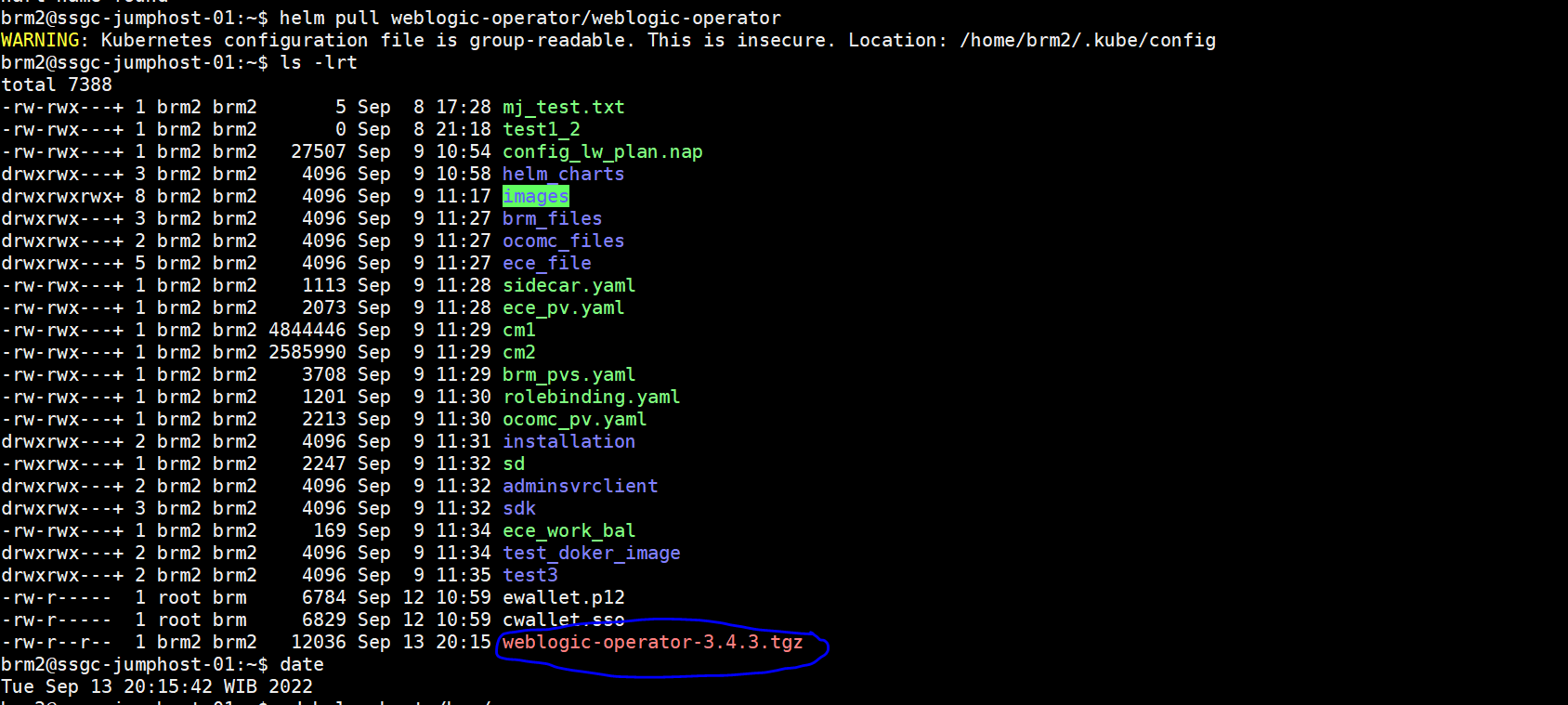
2) Add helm repo for weblogic-operator as given in Oracle documentation.

helm repo add weblogic-operator https://oracle.github.io/weblogic-kubernetes-operator/charts

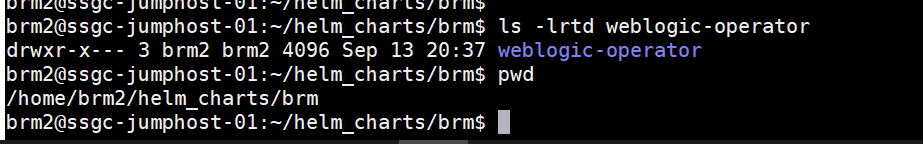
3) To pull helm chart for weblogic-operator by below command

helm pull weblogic-operator/weblogic-operator

helm chart will be pulled in jumphost as shown below .

****

4) Untar it in as per your directory structure



5) Now before running helm install, we need to upload the image to repository. Use below commands :-

1. Verify the image is present in docker

sudo docker images ghcr.io/oracle/weblogic-kubernetes-operator:3.4.3

1. Tag the image with repository server as below

sudo docker tag ghcr.io/oracle/weblogic-kubernetes-operator:3.4.3 asia-southeast1-docker.pkg.dev/project-launchpad-337407/brm-cloud/weblogic-operator:3.4.3

1. Push the image to repository

sudo docker push asia-southeast1-docker.pkg.dev/project-launchpad-337407/brm-cloud/weblogic-operator:3.4.3

**Note : docker push was not working from “brm2” user. Need to switch the user to “brm” and run docker push.**

6) Copy values.yaml to override-values.yaml

cp values.yaml override-values.yaml

7) Update below fields in override-values.yaml:-

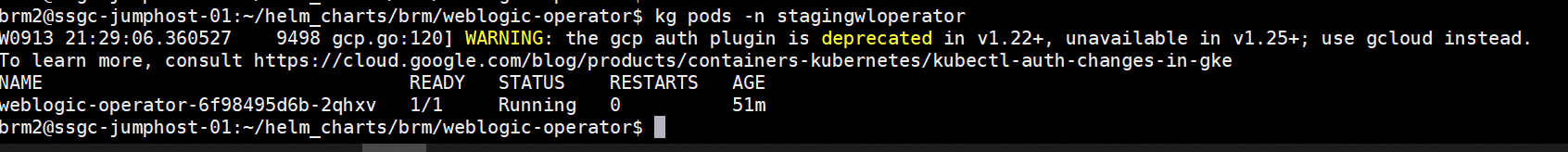
“domainNamespaces” : with the namespace on which we are going to install “oc-cn-helm-chart” ( in our case it is “stagingbrm”)

“image” : with repository detail which in our case is “asia-southeast1-docker.pkg.dev/project-launchpad-337407/brm-cloud/weblogic-operator:3.4.3”

8) Run helm install

helm install weblogic-operator ../weblogic-operator --namespace stagingwloperator --values override-values.yaml --debug

9) Verify the pods



**Installing init-DB Helm Chart :-**

We are deploying this with new database schema.

1. Update the override-values.yaml for initdb helm charts.

Password used in override-values.yaml :-

Brm\_root\_pass : **password**

Wallet password : **LWCN1234**

Encode these on base 64 and put it in override-values.yaml.

brm@ssgc-jumphost-01:~$ echo -n 'LWCN1234' | base64

TFdDTjEyMzQ=

Other details updated in override-values.yaml are :-

imageRepository: "asia-southeast1-docker.pkg.dev/project-launchpad-337407/brm-cloud/"

imagePullSecrets: ""

uniPass: ""

ocbrm:

brm\_root\_pass: cm9vdEAxMjMj

wallet:

client: TFdDTjEyMzQ=

server: TFdDTjEyMzQ=

root: TFdDTjEyMzQ=

db:

deployment:

imageName: init\_db

imageTag: 12.0.0.6.0

host: 10.170.15.18

port: 1521

service: LW5G19C

enable\_partition: 'Yes'

schemauser: pin

schemapass: ZlZRdTY0Y3I3dDg1

schematablespace: PIN00

indextablespace: PINX00

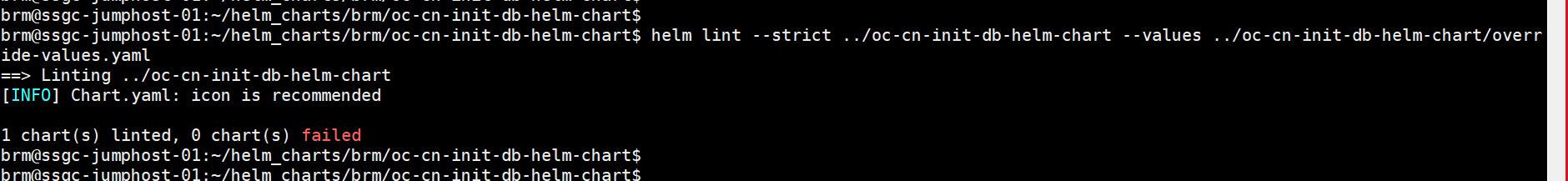
Note: All passwords are encoded on base 64.

1. Set context to initdb namespace .

kubectl config set-context --current --namespace=staginginitdb

1. Check the helm chart

helm lint --strict ../oc-cn-init-db-helm-chart --values ../oc-cn-init-db-helm-chart/override-values.yaml



1. Install/Deploy initdb helm

For ex:- helm install InitDbReleaseName oc-cn-init-db-helm-chart --namespace InitDbNameSpace --values OverrideValuesFile

Here, InitDbReleaseName :- this name should be lower case

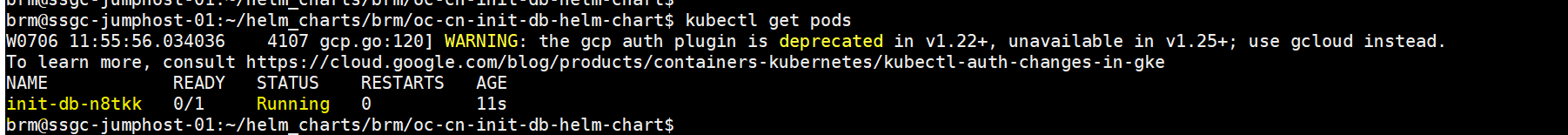
helm install initdbreleasename ../oc-cn-init-db-helm-chart --namespace staginginitdb --values override-values.yaml

Text

Description automatically generated

1. Verify pods using below command:-

Kubectl get pods



1. Login to pods using below commad:-

kubectl exec -it pod\_name – bash

Ex:- kubectl exec -it init-db-gwtnv -- bash

1. Check DB connectivity from pod

curl -v <telnet://10.170.15.25:1521>

A screenshot of a computer

Description automatically generated with medium confidence

1. Check pod logs

Command : kubectl logs -f pod\_name

Ex:- kubectl logs -f init-db-gwtnv

Note : It took around 20 mins for completion.

**Installing oc-op-job-helm-chart :-**

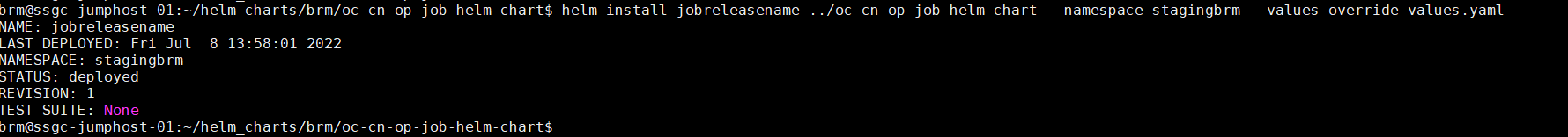
1. Untar oc-cn-op-job-helm-chart-12.0.0.6.0.tgz .

tar -zxvf oc-cn-op-job-helm-chart-12.0.0.6.0.tgz

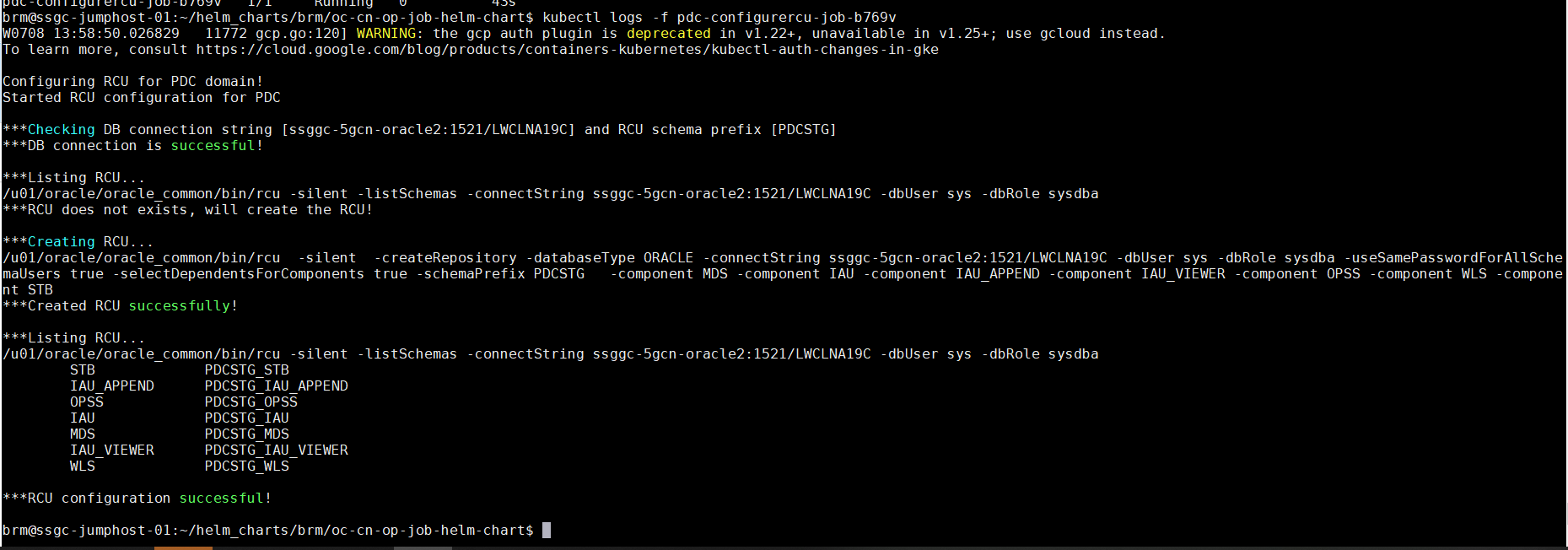
1. Copy values.yaml to override-values.yaml and update all the required fields in override-values.yaml.
2. Install oc-op-job helm chart using below command.

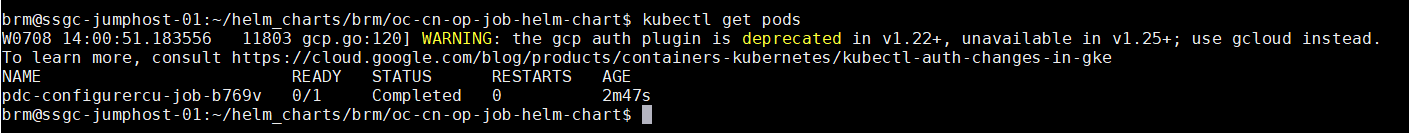
helm install jobreleasename ../oc-cn-op-job-helm-chart --namespace stagingbrm --values override-values.yaml

where, namespace “stagingbrm” is same for oc-cn-helm-chart.



Job successfully completed





**Installing oc-cn Helm Chart :-**

1. Update the override-values.yaml for oc-cn helm charts.

Note : We are deploying only brm and pdc only as of now. Skipping BOC,BC, PCC etc.

**Passwords/Other details :-**

**rcuPrefix:** PDCSTG

**PDC :-**

**AdminPassword :** weblogic123

**rcuSchemaPassword :** lwcn5gbrm

**wallet :** LWCN1234

**PDC Admin user password :** Pricing@123

1. Install the helm by using below command.

helm install brmreleasename ../oc-cn-helm-chart --namespace stagingbrm --values override-values.yaml

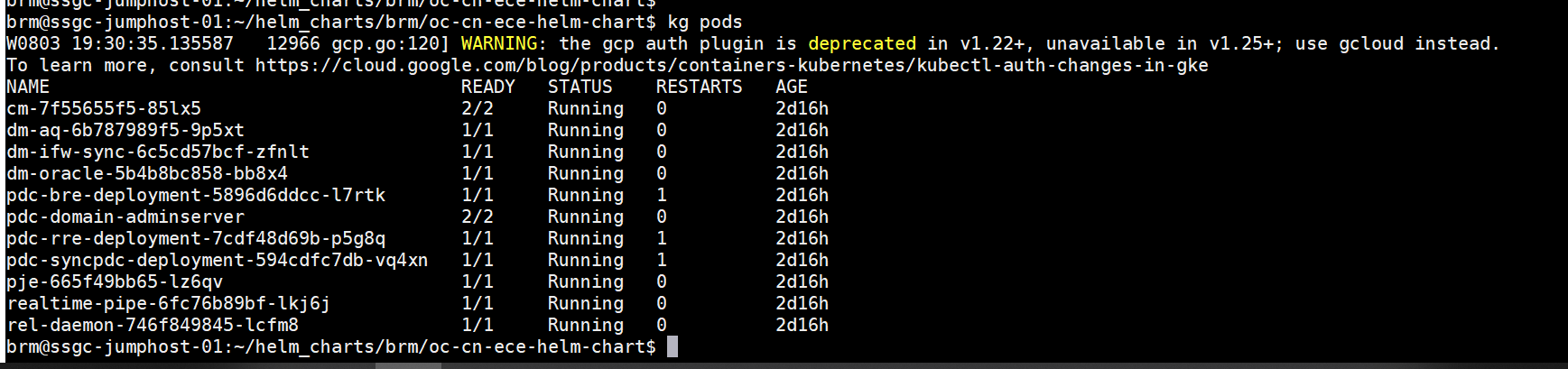
Or

helm install brmreleasename ../oc-cn-helm-chart --namespace stagingbrm --timeout 25m --values override-values.yaml --debug

where, brmreleasename :- release name (random name)

stagingbrm :- namespace in which we want to install this helm.

After it is installed successfully, you can see all the below pods.



**Installing oc-cn-ece Helm Chart :-**

1. Update the override-values.yaml for oc-cn helm charts.

walletPassword :- LWCN1234

JMSQUEUEPASSWORD:- Password123

RADIUSSHAREDSECRET:- radius123

NOTIFYEVENTKEYPASS:- Password123

BRMGATEWAYPASSWORD:- Password123

PDCPASSWORD:- Pricing@123

PDCKEYSTOREPASSWORD:- PDCSSLPass123#

PERSISTENCEDATABASEKEYPASS :- passw0rd

ECEHTTPGATEWAYSERVERSSLKEYSTOREPASSWORD:- Password123

BRM\_SERVER\_WALLET\_PASSWD :- LWCN1234

BRM\_ROOT\_WALLET\_PASSWD :- LWCN1234

BRMDATABASEKEYPASS :- passw0rd

1. Install the helm by using below command.

helm install ecereleasename ../oc-cn-ece-helm-chart --namespace stagingbrm --values override-values.yaml

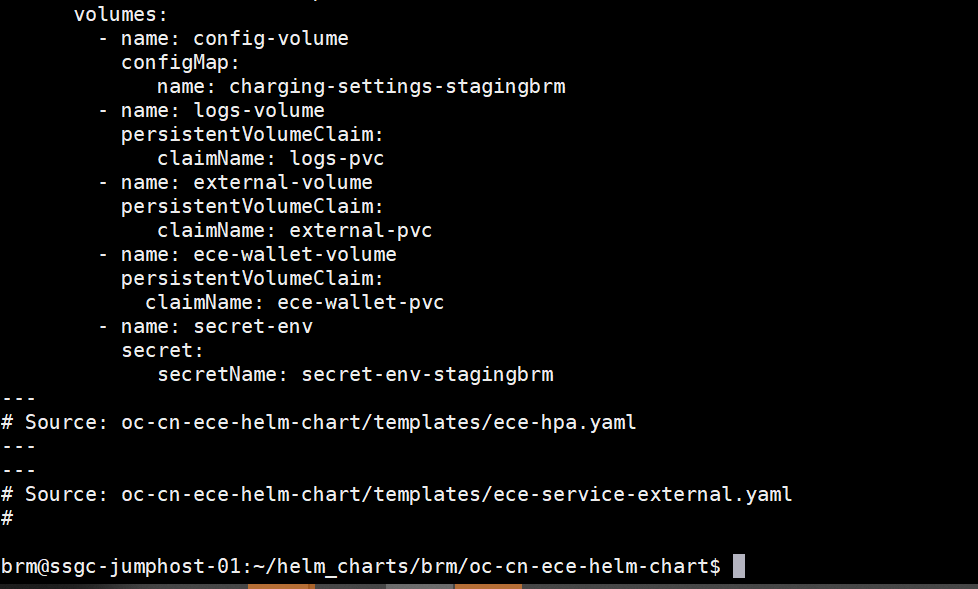
Or

helm install ecereleasename ../oc-cn-ece-helm-chart --namespace stagingbrm --values override-values.yaml --debug

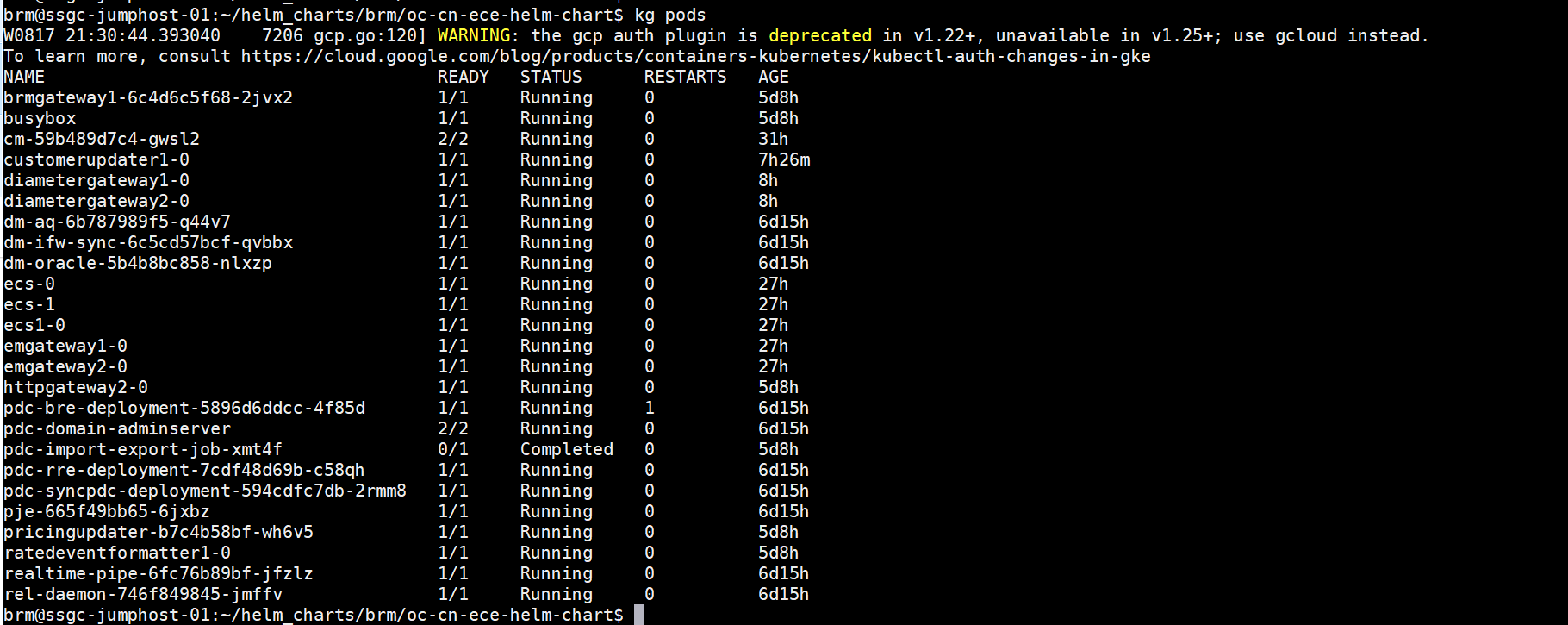
where, ecereleasename :- release name (random name)

stagingbrm :- namespace in which we want to install this helm.

It will take 2-3 mins to execute above command. But it will take some time for every pods to comeup. Keep monitoring the pods and respective logs.

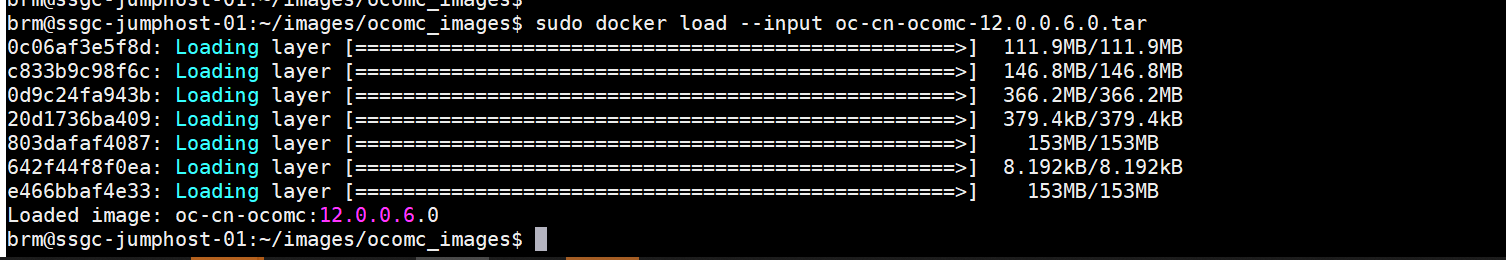


After it is installed successfully, you can see all the below pods (It includes pods of BRM,ECE and PDC).

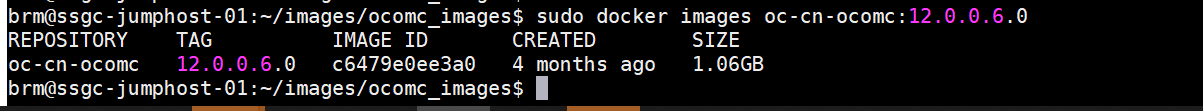


**Installing OCOMC Helm Chart :-**

1. Upload the images to repository .
2. sudo docker load --input oc-cn-ocomc-12.0.0.6.0.tar



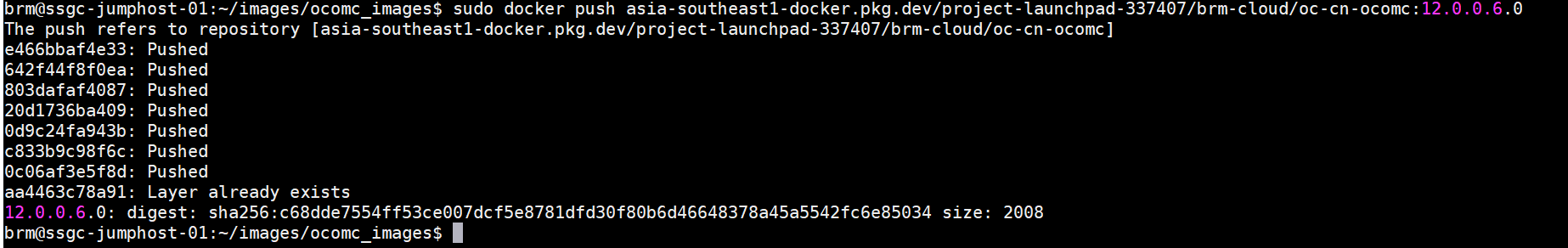
1. sudo docker images oc-cn-ocomc:12.0.0.6.0



1. sudo docker tag oc-cn-ocomc:12.0.0.6.0 asia-southeast1-docker.pkg.dev/project-launchpad-337407/brm-cloud/oc-cn-ocomc:12.0.0.6.0



1. sudo docker push asia-southeast1-docker.pkg.dev/project-launchpad-337407/brm-cloud/oc-cn-ocomc:12.0.0.6.0



1. Update the override-values.yaml for oc-cn helm charts.

nmKeypass: **Passw0rd1234#**

nmKeystorepass: **Passw0rd1234#**

adminKeypass: **Passw0rd1234#**

adminKeystorepass: **Passw0rd1234#**

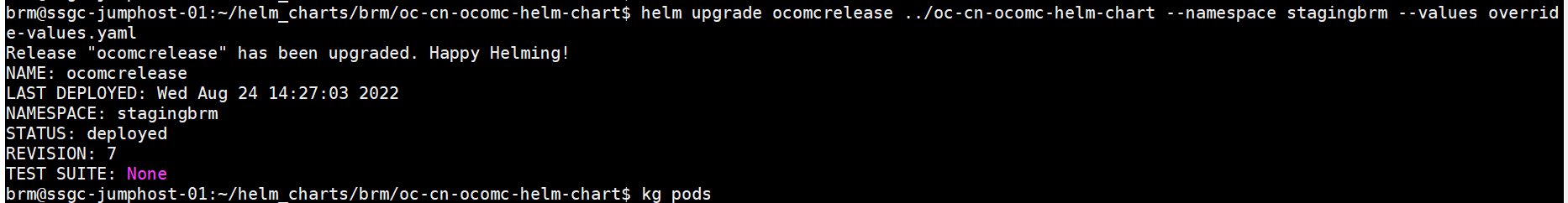
walletPassword : **Passw0rd1234#**

ocomcPassword: **Passw0rd1234#**

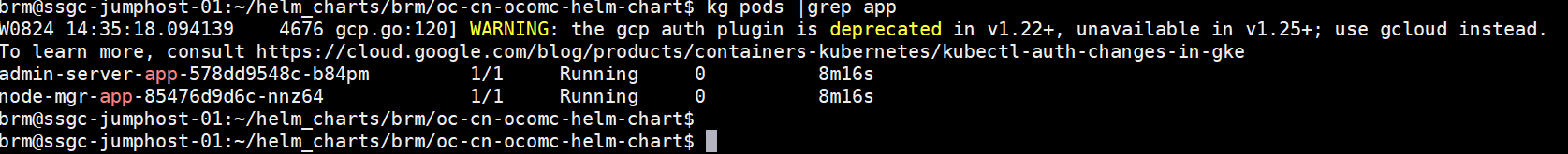
adminServerPassword: **Passw0rd1234#**

1. Run helm install after updating the yaml file.

helm install ocomcrelease ../oc-cn-ocomc-helm-chart --namespace stagingbrm --values override-values.yaml



After it is installed successfully, you can see OCOMC pods of nodemgr and adminsvr as shown below.

****

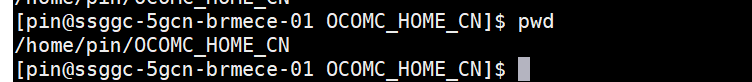
**Note : Put “ojbbc8.jar” in 3rdparty\_jars folder.**

**Installing OCOMC Adminstration Client to access GUI :-**

To access OCOMC GUI, we need to install Administration Client in jumphost/on-premise server/Windows . In our scenario, we have installed it on another on-premise server (10.170.15.47).

1. Create a directory on the server where you want to install administration client as below :-

mkdir OCOMC\_HOME\_CN/

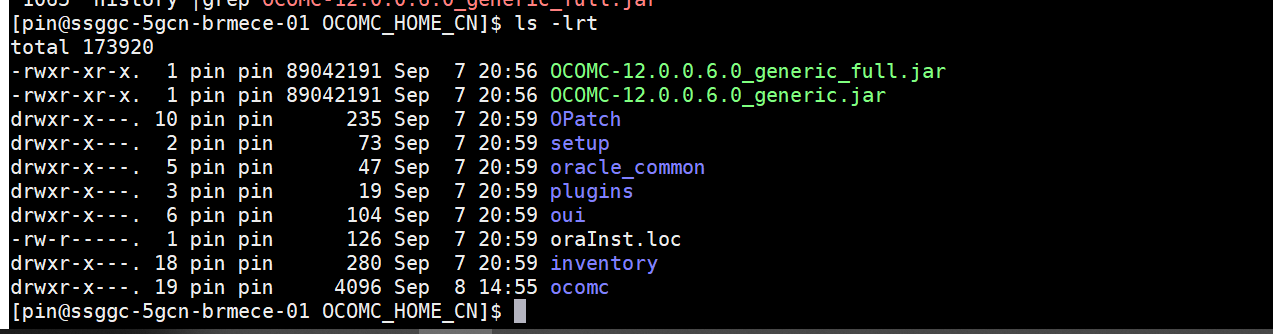


1. Download the OCOMC pack. Untar the pack in above created directory. Use below command to install it.

$JAVA\_HOME/bin/java -jar OCOMC-12.0.0.6.0\_generic\_full.jar

1. GUI will open. Follow the instructions and give the path of installation. In one of the screen it will ask, what we need to install. Select **Admin Client** and click Next.
2. Review your selections, and click **Install**.
3. Click Finish to complete the installation.

Admin client is installed now. We can see the directory structure like this.



**Post-Installation steps to access GUI:-**

1. Expose the Administration Server Pod (admin-server-app).If Administration Client is located remotely or is on a Windows system, set the Administration Server's service type to NodePort.
2. Open your override-values.yaml file for oc-cn-ocomc-helm-chart.
3. Set the ocomc.configEnv.adminsvrIp key to admin-server-app
4. If Administration Client is installed remotely or on a Windows system, set these additional keys:

• ocomc.configEnv.nmExternalPort: Set this to the external port for the Node Manager. Set this key only if your admin-server Pod and node-mgr Pod are running in different machines.

• ocomc.configEnv.adminsvrExternalPort: Set this to the external port for the Administration Server.

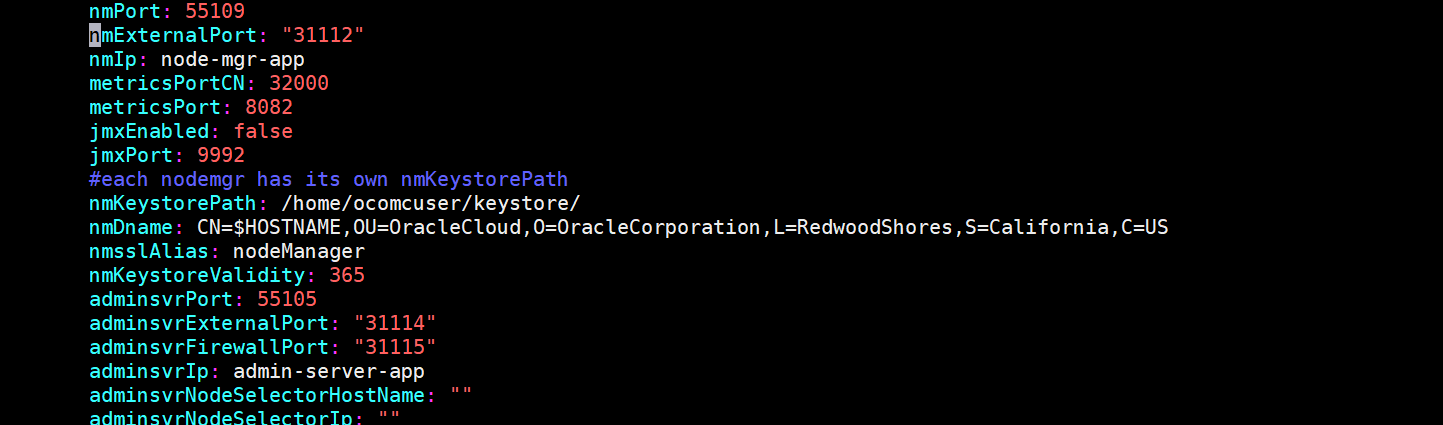
• ocomc.configEnv.adminsvrFirewallPort: Set this to the Administration Server firewall port

We have set

nmExternalPort : “31112”

adminsvrExternalPort : “31114”

adminsvrFirewallPort : “31115”



1. Save the file and run helm upgrade.

helm upgrade ocomcrelease ../oc-cn-ocomc-helm-chart --namespace stagingbrm --values override-values.yaml

1. Also in the Administration Client machine's /etc/hosts file, add the IP address of the Kubernetes node where Administration Server is running. For example:

IPAddress Hostname

10.170.15.40 admin-server-app

where, admin-server-app:- is the name of admin-server service

10.170.15.40 :- one of the worker node ip.

All the steps are completed, now to access gui, go to bin path on the server where Admin Client is installed and run “./gui” as we do in on-premise.

In hostname, we give admin-server server name “admin-server-app” and in port, we need to give adminsvrExternalPort (31114).

