Helm Assignment with Chartmuseum

Question: -

1. Deploying chart museum in local machine (Reference: https://chartmuseum.com/)

Answer: -

To deploy our project, initially I have to follow below steps as -

1. Before doing this project, I have to start minikube by

   minikube start

```
(base) souvik@it002108:~$ minikube start
😄  minikube v1.24.0 on Ubuntu 20.04
✨  Using the docker driver based on existing profile
👍  Starting control plane node minikube in cluster minikube
🚜  Pulling base image ...
🔄  Restarting existing docker container for "minikube" ...
🐳  Preparing Kubernetes v1.22.3 on Docker 20.10.8 ...
🔎  Verifying Kubernetes components...
    ▪ Using image kubernetesui/dashboard:v2.3.1
    ▪ Using image kubernetesui/metrics-scraper:v1.0.7
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: storage-provisioner, default-storageclass, dashboard
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

2. Now, I have to download and install all dependencies for installing helm with below command -

   a) curl https://baltocdn.com/helm/signing.asc | sudo apt-key add -

   b) sudo apt-get install apt-transport-https --yes

   c) echo "deb https://baltocdn.com/helm/stable/debian/ all main" | sudo tee /etc/apt/sources.list.d/helm-stable-debian.list

```
(base) souvik@it002108:~$ curl https://baltocdn.com/helm/signing.asc | sudo apt-key add -
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1700  100  1700    0     0    224      0  0:00:07  0:00:07 --:--:--   425
OK
(base) souvik@it002108:~$ sudo apt-get install apt-transport-https --yes
Reading package lists... Done
Building dependency tree
Reading state information... Done
apt-transport-https is already the newest version (2.0.6).
The following package was automatically installed and is no longer required:
  mongo-tools
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
(base) souvik@it002108:~$ echo "deb https://baltocdn.com/helm/stable/debian/ all main" | sudo tee /etc/apt/sources.list.d/helm-stable-debian.l
ist
deb https://baltocdn.com/helm/stable/debian/ all main
```

3. Now we have to install helm by "**sudo apt-get install helm".**

```
(base) souvik@it002108:~$ sudo apt-get install helm
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  mongo-tools
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  helm
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 13.8 MB of archives.
After this operation, 45.7 MB of additional disk space will be used.
Get:1 https://baltocdn.com/helm/stable/debian all/main amd64 helm amd64 3.7.2-1 [13.8 MB]
Fetched 13.8 MB in 7s (1,900 kB/s)
Selecting previously unselected package helm.
(Reading database ... 214834 files and directories currently installed.)
Preparing to unpack .../helm_3.7.2-1_amd64.deb ...
Unpacking helm (3.7.2-1) ...
Setting up helm (3.7.2-1) ...
Processing triggers for man-db (2.9.1-1) ...
```

4. Now to check the helm installation, I have to run **"helm version"** which helps me to understand that helm is installed successfully into my system.

```
(base) souvik@it002108:~$ helm version
version.BuildInfo{Version:"v3.7.2", GitCommit:"663a896f4a815053445eec4153677d
dc24a0a361", GitTreeState:"clean", GoVersion:"go1.16.10"}
(base) souvik@it002108:~$
```

5. Now, I shall check minikube status by using **"minikube status"** to check it's up and running or not and if it's running properly then go for my assignment.

```
(base) souvik@it002108:~$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
```

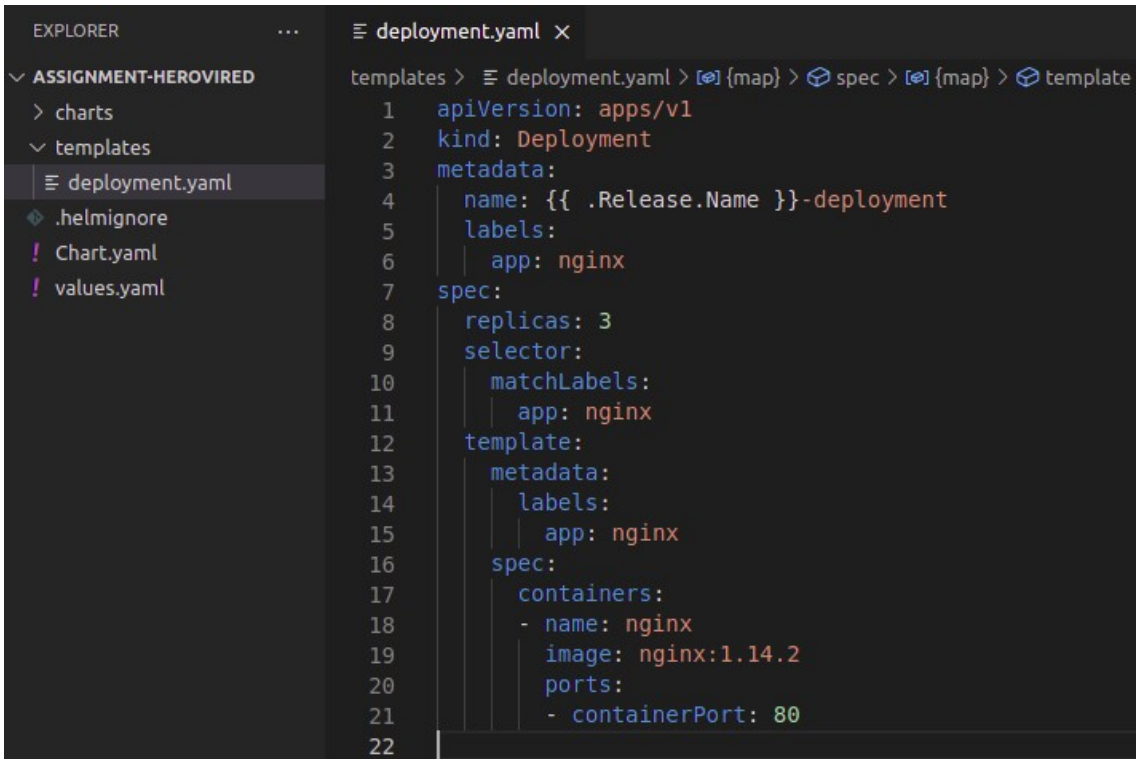
For this assessment, I have followed below steps to deploy chartmuseum in my local system. Steps are -

1. Initially, I have to create helm chart by using **“helm create <Chart_Name>”** command. Here,

   I used: **helm create Assignment-Herovired**

```
(base) souvik@it002108:~/Desktop/Assessment/kubernetes/helm$ helm create Assi
gnment-Herovired
Creating Assignment-Herovired
```

2. After that, I have deleted all the files & folders from “**templates**” folder and written a “**deployment.yaml**” within that “**templates**” folder. I have used **Nignx:1.14.2** image to do so. Code is shared in below -



3. After that, we will install a helm chart by using “**helm install <release_name> <chart_dir_name>**” command. Here -

   I used: **helm install my-assignment Assignment-herovired**

```
(base) souvik@it002108:~/Desktop/Assessment/kubernetes/helm$ helm install my-assignment Assignment-
Herovired
NAME: my-assignment
LAST DEPLOYED: Sun Dec 26 01:30:59 2021
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
(base) souvik@it002108:~/Desktop/Assessment/kubernetes/helm$
```

4. Now, we have chacked that our "**deployment.yaml**" file is being deployed or not. To do so, I have used –

   i) kubectl get deploy – for checking our deployment.yaml file is deployed successfully or not.

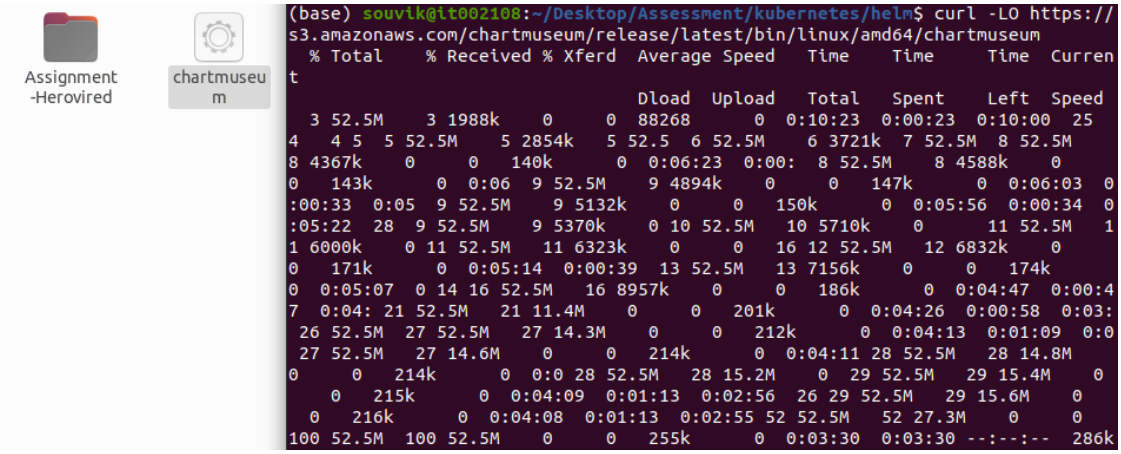
   ii) kubectl get rs – for checking our three replicaset is deployed successfully and all three are ready or not.

```
(base) souvik@it002108:~/Desktop/Assessment/kubernetes/helm$ kubectl get deployments
NAME                    READY   UP-TO-DATE   AVAILABLE   AGE
hello-minikube          1/1     1            1           8d
my-assignment-deployment   3/3     3            3           2m21s
(base) souvik@it002108:~/Desktop/Assessment/kubernetes/helm$ kubectl get rs
NAME                                 DESIRED   CURRENT   READY   AGE
hello-minikube-6ddfcc9757            1         1         1       8d
my-assignment-deployment-66b6c48dd5   3         3         3       4m10s
(base) souvik@it002108:~/Desktop/Assessment/kubernetes/helm$
```

5. Now, we have to set-up our chartmuseum to deploy our helm chart in our local mechine. The following steps should be followed -

   a) Initially, we have to install **chartmuseum** by -

   curl -LO https://s3.amazonaws.com/chartmuseum/release/latest/bin/linux/amd64/chartmuseum



   b) After that, we have to give permission to that **chartmuseum** file by – "**chmod +x chartmuseum**" and after that we have to send that file to the helm repository. Here, for me, it's usr/sbin/. So, I move my **chartmuseum** file to that directory.

```
(base) souvik@it002108:~/Desktop/Assessment/kubernetes/helm$ chmod +x ./chartm
useum
(base) souvik@it002108:~/Desktop/Assessment/kubernetes/helm$ which heml
(base) souvik@it002108:~/Desktop/Assessment/kubernetes/helm$ which helm
/usr/sbin/helm
(base) souvik@it002108:~/Desktop/Assessment/kubernetes/helm$ mv ./chartmuseum
/usr/sbin/
mv: cannot move './chartmuseum' to '/usr/sbin/chartmuseum': Permission denied
(base) souvik@it002108:~/Desktop/Assessment/kubernetes/helm$ sudo mv ./chartmu
seum /usr/sbin/
[sudo] password for souvik:
(base) souvik@it002108:~/Desktop/Assessment/kubernetes/helm$ chartmuseum --ver
sion
ChartMuseum version 0.12.0 (build 101e26a)
(base) souvik@it002108:~/Desktop/Assessment/kubernetes/helm$
```

c) Now, in another terminal, I have run my chartmuseum in listening mode for a specific port number by – "**chartmuseum --port=<Port_No> --storage="local" --storage-local-rootdir= '<Chart_Dir>'**". Here,

I used: **chartmuseum --port=8080 --storage="local" --storage-local-rootdir= "./Assignment_Herovired".**

```
(base) souvik@it002108:~/Desktop/Assessment/kubernetes/helm$ chartmuseum --port=
8080 --storage="local" --storage-local-rootdir="./Assignment-Herovired"
2021-12-26T03:11:57.063+0530    INFO    Starting ChartMuseum    {"port": 8080}
```

d) And then, I have added my chartmuseum to my helm repository. And updated by "**helm repo update**".

```
(base) souvik@it002108:~/Desktop/Assessment/kubernetes/helm$ helm repo add chartmuseum
 http://localhost:8080
"chartmuseum" has been added to your repositories
(base) souvik@it002108:~/Desktop/Assessment/kubernetes/helm$
```

6. After setting up our chartmuseum, We have to create a package of our helm chart by "**helm package .**" after entering into that particular helm-chart directory using "**cd Assignment-Herovired**".

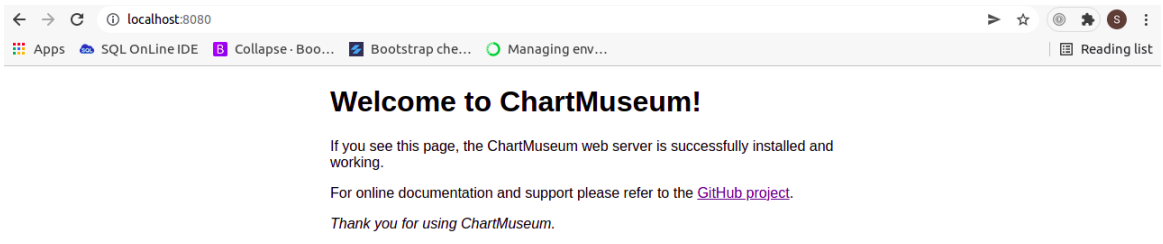
```
(base) souvik@it002108:~/Desktop/Assessment/kubernetes/helm/Assignment-Herovired$ helm package .
Successfully packaged chart and saved it to: /home/souvik/Desktop/Assessment/kubernetes/helm/Assignment-Herovi
red/Assignment-Herovired-0.1.0.tgz
(base) souvik@it002108:~/Desktop/Assessment/kubernetes/helm/Assignment-Herovired$
```

7. Now, I have uploaded my chart package which is created by above step by - "**curl --data-binary "@<Chart_Package_Name>" http://localhost:8080/api/charts**". Here,

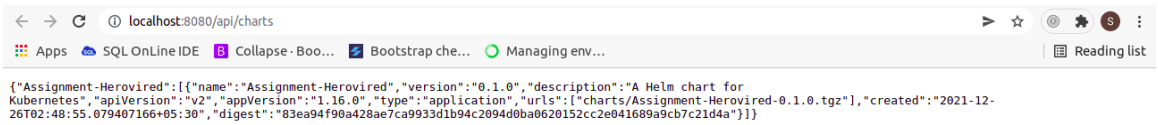
I used: **curl --data-binary "@Assignment-Herovired-0.1.0.tgz" http://localhost:8080/api/charts**

```
(base) souvik@it002108:~/Desktop/Assessment/kubernetes/helm/Assignment-Herovired$ curl
 --data-binary "@Assignment-Herovired-0.1.0.tgz" http://localhost:8080/api/charts
{"error":"file already exists"}
(base) souvik@it002108:~/Desktop/Assessment/kubernetes/helm/Assignment-Herovired$
```

8. After that, we can easily access our localhost with 8080 port as – **http://localhost:8080**



9. Also, we can check that our uploaded chart is working or not with the specific URL. As – **http://localhost:8080/api/charts**



10. We can also do the same from helm itself by -

```
(base) souvik@it002108:~/Desktop/Assessment/kubernetes/helm$ helm install locally-deployed
 chartmuseum/Assignment-Herovired
NAME: locally-deployed
LAST DEPLOYED: Sun Dec 26 03:26:11 2021
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
(base) souvik@it002108:~/Desktop/Assessment/kubernetes/helm$ kubectl get deploy
NAME                         READY   UP-TO-DATE   AVAILABLE   AGE
hello-minikube               1/1     1            1           8d
locally-deployed-deployment  3/3     3            3           56s
my-assignment-deployment     3/3     3            3           41m
(base) souvik@it002108:~/Desktop/Assessment/kubernetes/helm$ kubectl get rs
NAME                                   DESIRED   CURRENT   READY   AGE
hello-minikube-6ddfcc9757              1         1         1       8d
locally-deployed-deployment-66b6c48dd5 3         3         3       68s
my-assignment-deployment-66b6c48dd5    3         3         3       42m
(base) souvik@it002108:~/Desktop/Assessment/kubernetes/helm$
```

- these are the steps need to be followed to deploy chartmuseum into our local machine.