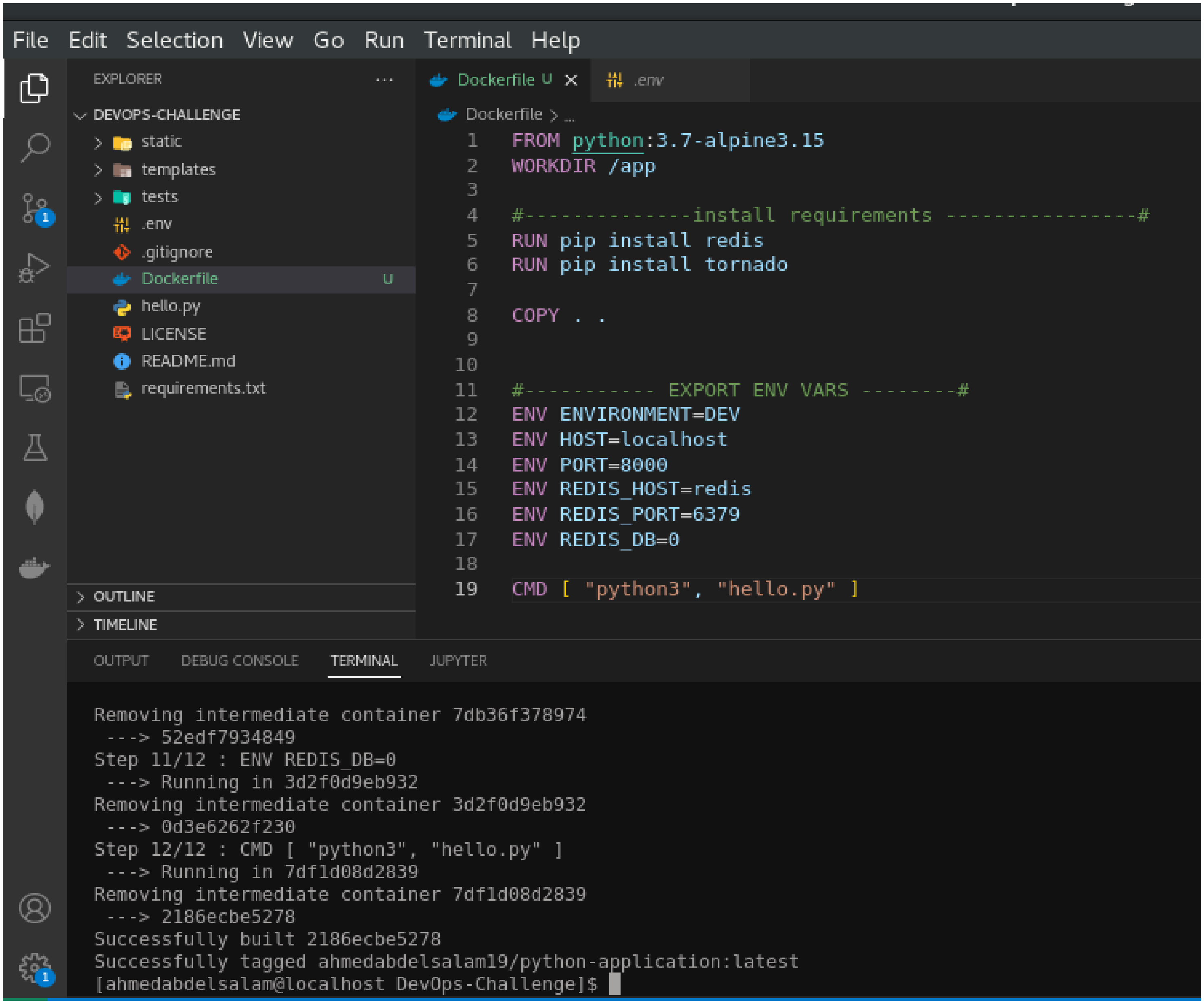
# Helm Lab

## Q1:

→ clone python-app github repo

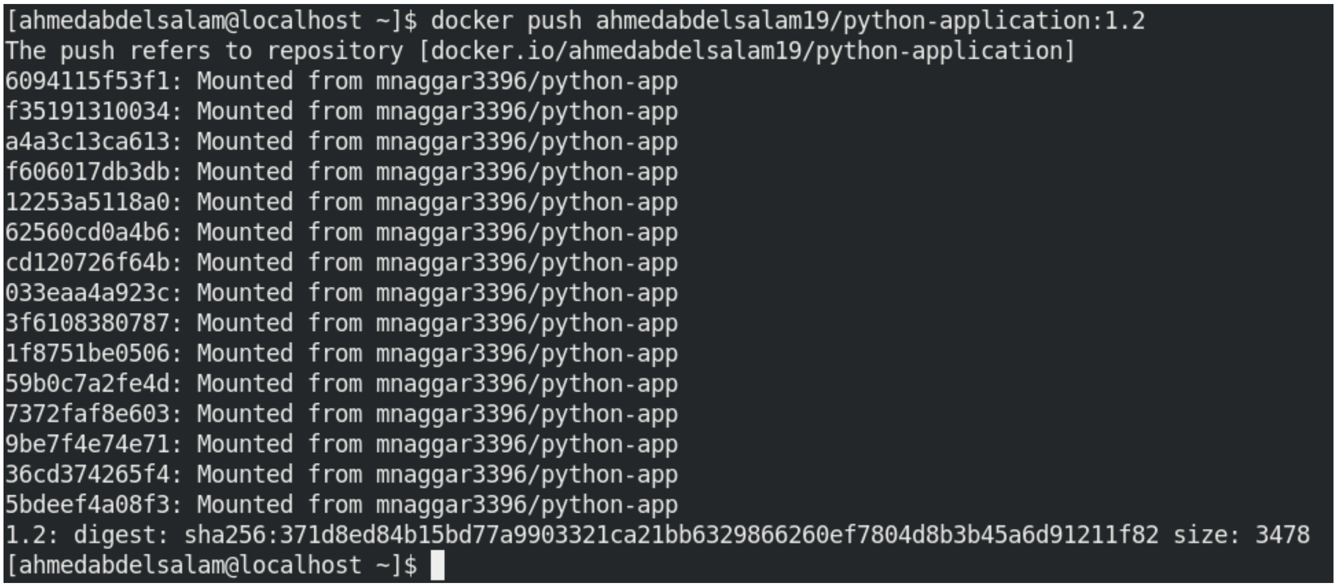
→ Create Dockerfile to build image for this app :



```
FROM python:3.7-alpine3.15
WORKDIR /app

#--------------install requirements ----------------#
RUN pip install redis
RUN pip install tornado

COPY . .


#----------- EXPORT ENV VARS --------#
ENV ENVIRONMENT=DEV
ENV HOST=localhost
ENV PORT=8000
ENV REDIS_HOST=redis
ENV REDIS_PORT=6379
ENV REDIS_DB=0

CMD [ "python3", "hello.py" ]
```

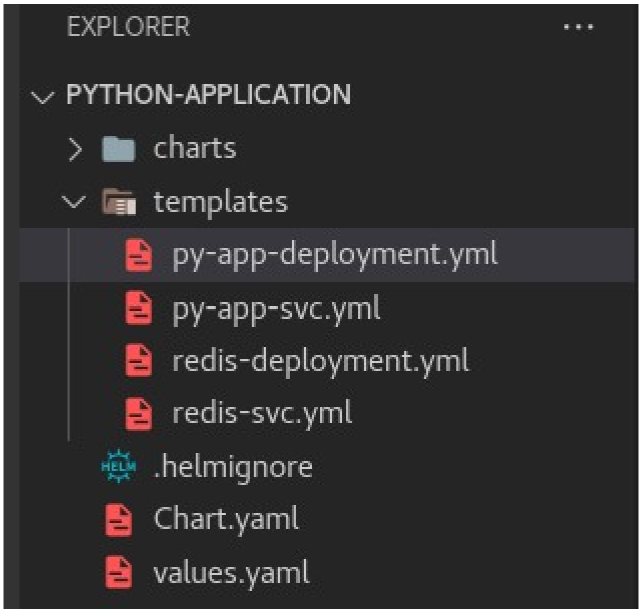
```
Removing intermediate container 7db36f378974
 ---> 52edf7934849
Step 11/12 : ENV REDIS_DB=0
 ---> Running in 3d2f0d9eb932
Removing intermediate container 3d2f0d9eb932
 ---> 0d3e6262f230
Step 12/12 : CMD [ "python3", "hello.py" ]
 ---> Running in 7df1d08d2839
Removing intermediate container 7df1d08d2839
 ---> 2186ecbe5278
Successfully built 2186ecbe5278
Successfully tagged ahmedabdelsalam19/python-application:latest
[ahmedabdelsalam@localhost DevOps-Challenge]$
```

→ build image
→ tag image
→ push image to Dockerhub

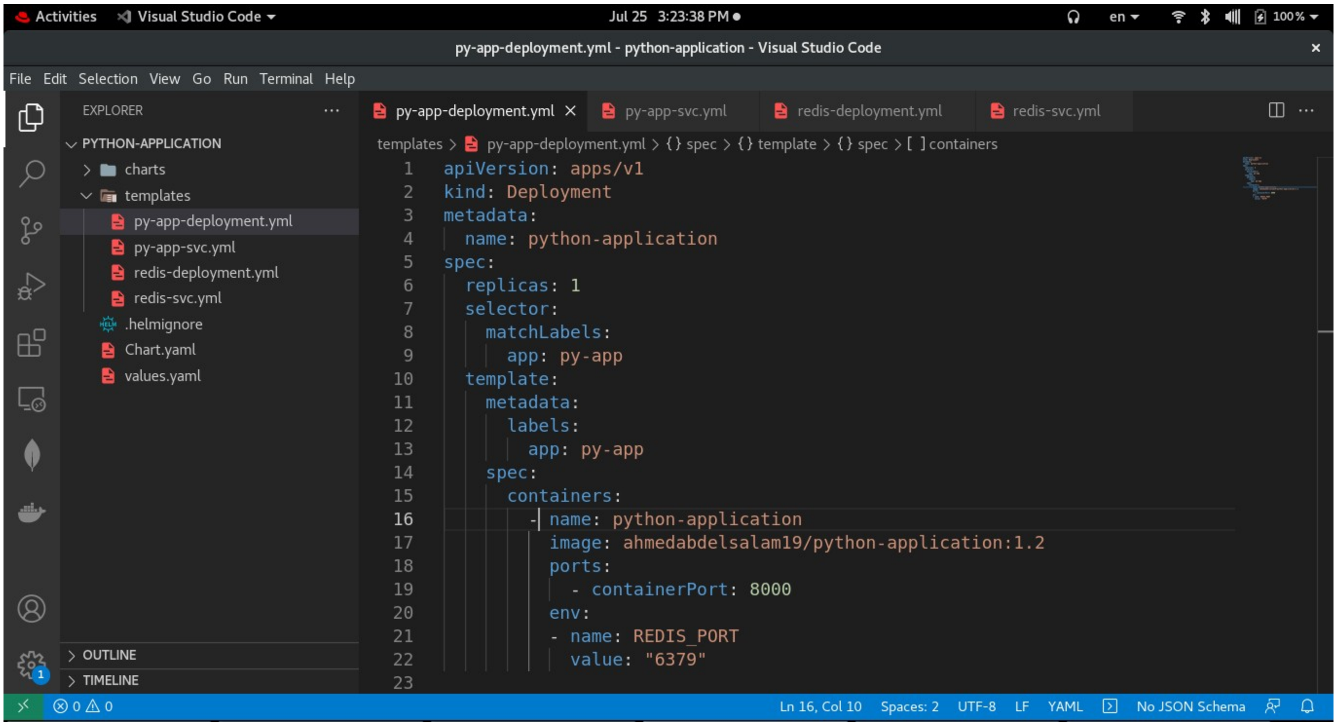


```
[ahmedabdelsalam@localhost ~]$ docker push ahmedabdelsalam19/python-application:1.2
The push refers to repository [docker.io/ahmedabdelsalam19/python-application]
6094115f53f1: Mounted from mnaggar3396/python-app
f35191310034: Mounted from mnaggar3396/python-app
a4a3c13ca613: Mounted from mnaggar3396/python-app
f606017db3db: Mounted from mnaggar3396/python-app
12253a5118a0: Mounted from mnaggar3396/python-app
62560cd0a4b6: Mounted from mnaggar3396/python-app
cd120726f64b: Mounted from mnaggar3396/python-app
033eaa4a923c: Mounted from mnaggar3396/python-app
3f6108380787: Mounted from mnaggar3396/python-app
1f8751be0506: Mounted from mnaggar3396/python-app
59b0c7a2fe4d: Mounted from mnaggar3396/python-app
7372faf8e603: Mounted from mnaggar3396/python-app
9be7f4e74e71: Mounted from mnaggar3396/python-app
36cd374265f4: Mounted from mnaggar3396/python-app
5bdeef4a08f3: Mounted from mnaggar3396/python-app
1.2: digest: sha256:371d8ed84b15bd77a9903321ca21bb6329866260ef7804d8b3b45a6d91211f82 size: 3478
[ahmedabdelsalam@localhost ~]$ 
```

→ create chart <python-application>
→ create deployment and service for both [python app and redis]

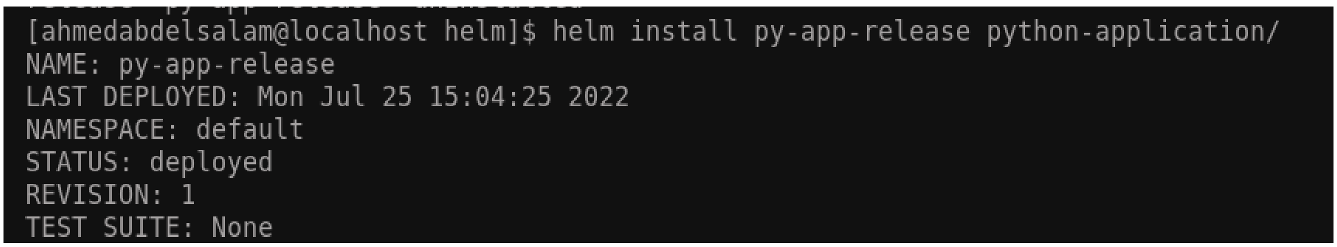


```
EXPLORER                        ···
∨ PYTHON-APPLICATION
  > charts
  ∨ templates
      py-app-deployment.yml
      py-app-svc.yml
      redis-deployment.yml
      redis-svc.yml
    .helmignore
    Chart.yaml
    values.yaml
```



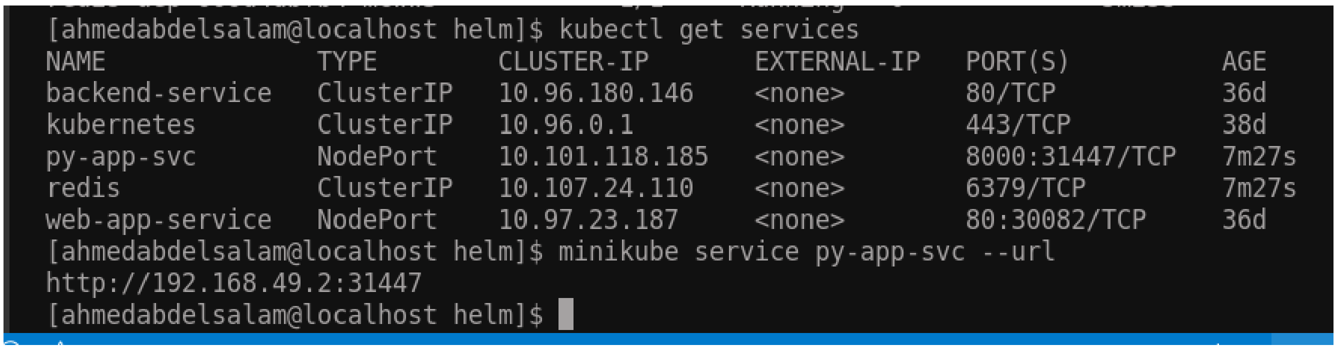
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: python-application
spec:
  replicas: 1
  selector:
    matchLabels:
      app: py-app
  template:
    metadata:
      labels:
        app: py-app
    spec:
      containers:
      - name: python-application
        image: ahmedabdelsalam19/python-application:1.2
        ports:
          - containerPort: 8000
        env:
        - name: REDIS_PORT
          value: "6379"

```

→ helm install for python-application



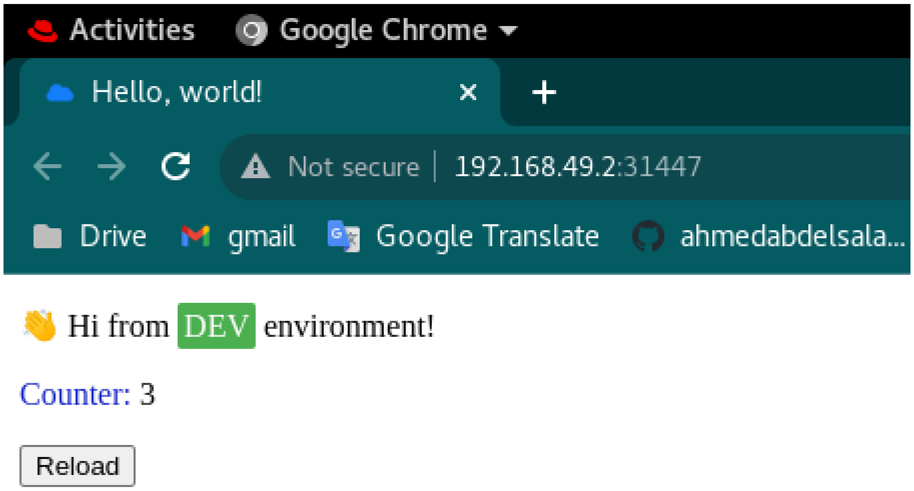
```
[ahmedabdelsalam@localhost helm]$ helm install py-app-release python-application/
NAME: py-app-release
LAST DEPLOYED: Mon Jul 25 15:04:25 2022
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
```

get ip for service:



```
[ahmedabdelsalam@localhost helm]$ kubectl get services
NAME                TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
backend-service     ClusterIP   10.96.180.146    <none>        80/TCP           36d
kubernetes          ClusterIP   10.96.0.1        <none>        443/TCP          38d
py-app-svc          NodePort    10.101.118.185   <none>        8000:31447/TCP   7m27s
redis               ClusterIP   10.107.24.110    <none>        6379/TCP         7m27s
web-app-service     NodePort    10.97.23.187     <none>        80:30082/TCP     36d
[ahmedabdelsalam@localhost helm]$ minikube service py-app-svc --url
http://192.168.49.2:31447
[ahmedabdelsalam@localhost helm]$ 
```

# completed !



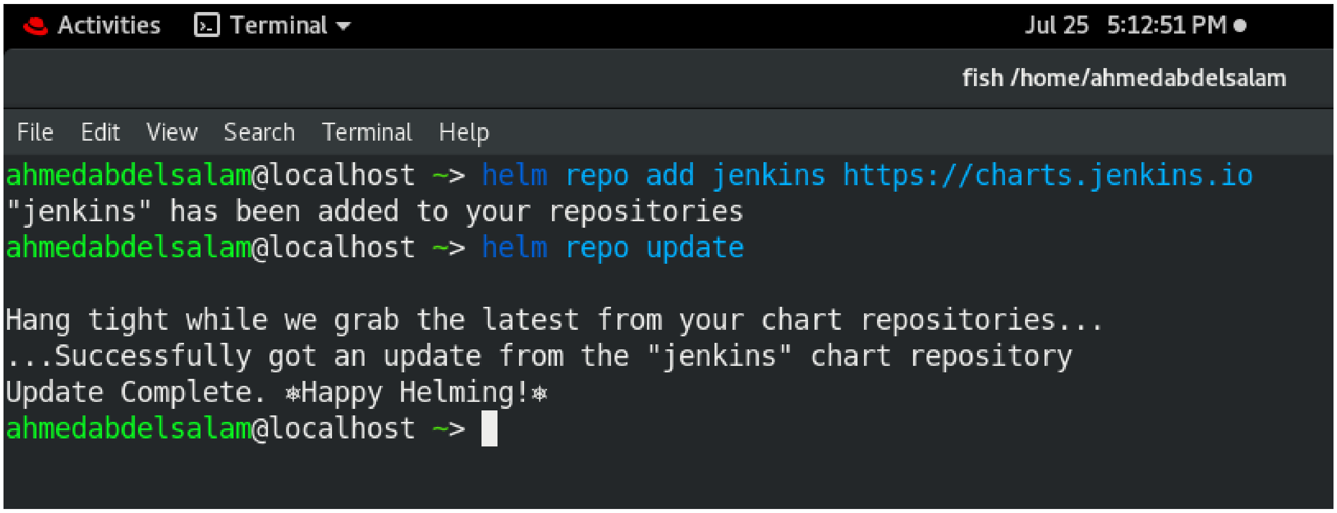
👋 Hi from DEV environment!

Counter: 3

Reload

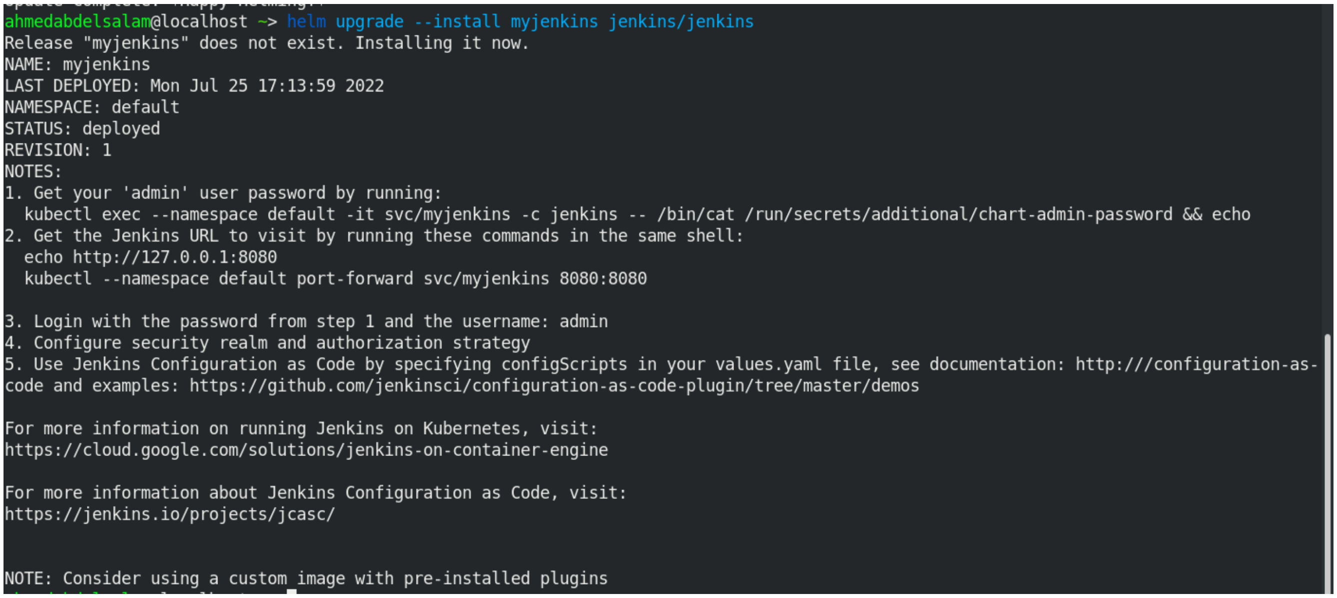
# Q2:Deploy Jenkins Chart on the cluster and login to jenkins

## → get and update jenkins repo


Activities
Terminal
Jul 25 5:12:51 PM
fish /home/ahmedabdelsalam
File Edit View Search Terminal Help

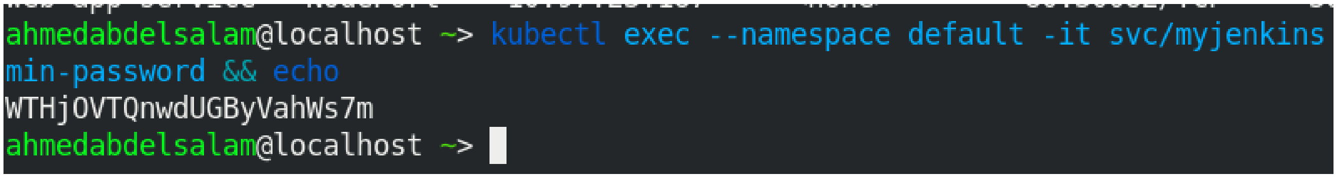
```
ahmedabdelsalam@localhost ~> helm repo add jenkins https://charts.jenkins.io
"jenkins" has been added to your repositories
ahmedabdelsalam@localhost ~> helm repo update

Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "jenkins" chart repository
Update Complete. ⎈Happy Helming!⎈
ahmedabdelsalam@localhost ~> 
```

## → create and update jenkins release

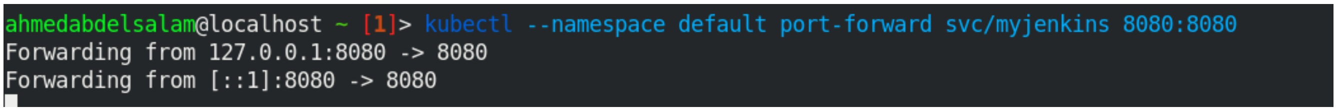


```
ahmedabdelsalam@localhost ~> helm upgrade --install myjenkins jenkins/jenkins
Release "myjenkins" does not exist. Installing it now.
NAME: myjenkins
LAST DEPLOYED: Mon Jul 25 17:13:59 2022
NAMESPACE: default
STATUS: deployed
REVISION: 1
NOTES:
1. Get your 'admin' user password by running:
  kubectl exec --namespace default -it svc/myjenkins -c jenkins -- /bin/cat /run/secrets/additional/chart-admin-password && echo
2. Get the Jenkins URL to visit by running these commands in the same shell:
  echo http://127.0.0.1:8080
  kubectl --namespace default port-forward svc/myjenkins 8080:8080

3. Login with the password from step 1 and the username: admin
4. Configure security realm and authorization strategy
5. Use Jenkins Configuration as Code by specifying configScripts in your values.yaml file, see documentation: http:///configuration-as-
code and examples: https://github.com/jenkinsci/configuration-as-code-plugin/tree/master/demos

For more information on running Jenkins on Kubernetes, visit:
https://cloud.google.com/solutions/jenkins-on-container-engine

For more information about Jenkins Configuration as Code, visit:
https://jenkins.io/projects/jcasc/


NOTE: Consider using a custom image with pre-installed plugins
```

→ execute  notes from above result
1- get admin user password
2- get myjenkins url

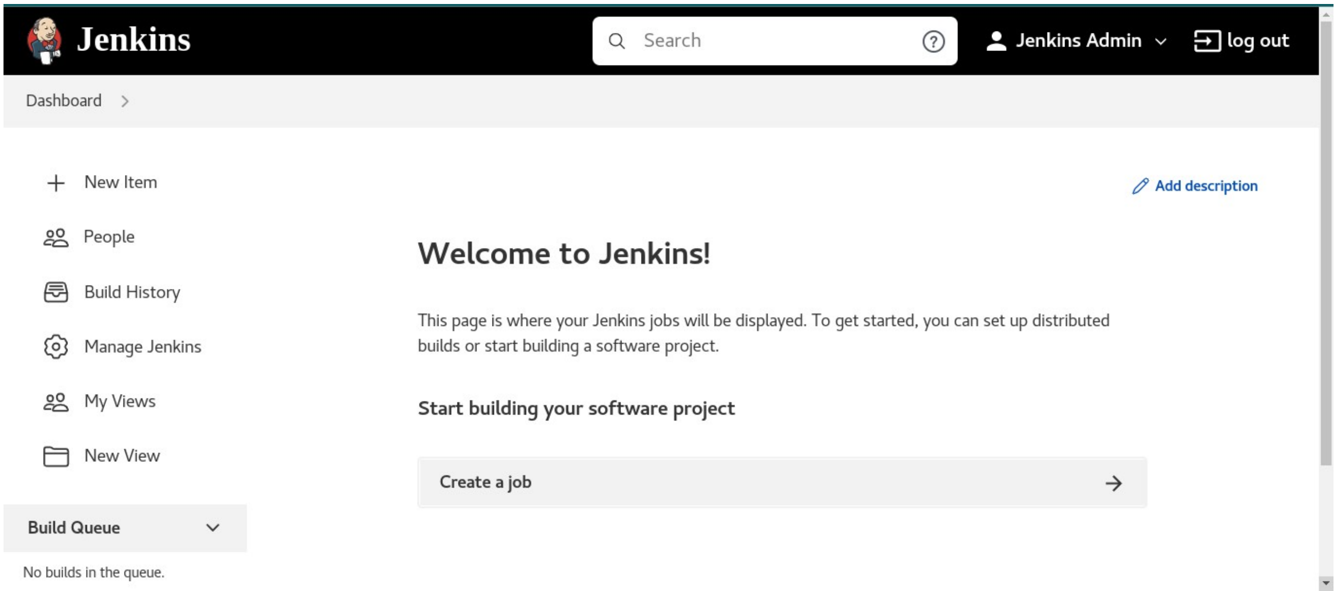


```
ahmedabdelsalam@localhost ~> kubectl exec --namespace default -it svc/myjenkins
min-password && echo
WTHjOVTQnwdUGByVahWs7m
ahmedabdelsalam@localhost ~>
```



```
ahmedabdelsalam@localhost ~ [1]> kubectl --namespace default port-forward svc/myjenkins 8080:8080
Forwarding from 127.0.0.1:8080 -> 8080
Forwarding from [::1]:8080 -> 8080
```

## → login to jenkins



Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

No builds in the queue.