Jenkins over Kubernetes

task 3

Perform second task on top of Kubernetes where we use Kubernetes resources.

1. Create container image that’s has Jenkins installed using dockerfile Or You can use the Jenkins Server on RHEL 8/7

2. When we launch this image, it should automatically starts Jenkins service in the container.

3. Create a job chain of job1, job2, job3 and job4 using build pipeline plugin in Jenkins

4. Job1 : Pull the Github repo automatically when some developers push repo to Github.

5. Job2 :
1. By looking at the code or program file, Jenkins should automatically start the respective language interpreter installed image container to deploy code on top of Kubernetes (eg. If code is of PHP, then Jenkins should start the container that has PHP already installed)

2. Expose your pod so that testing team could perform the testing on the pod

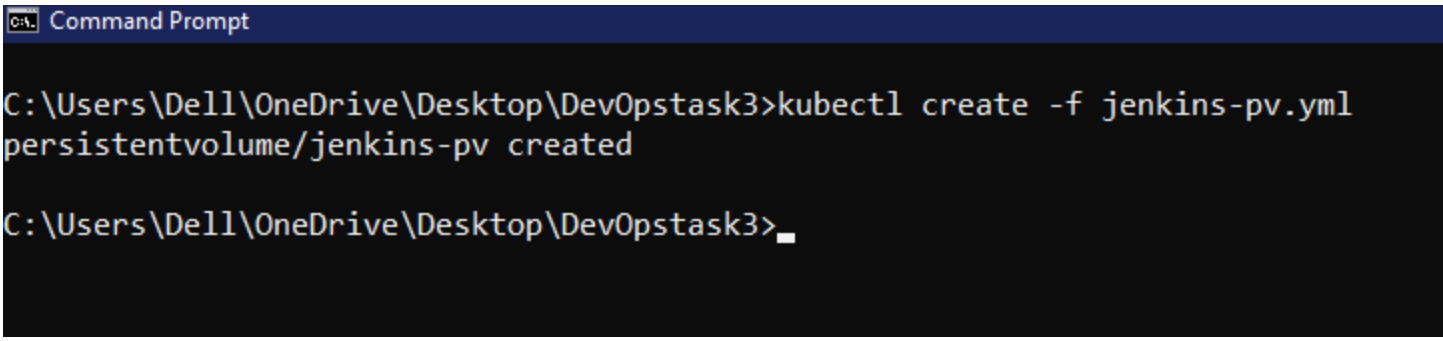
3. Make the data to remain persistent (If server collects some data like logs, other user information)

6. Job3 : Test your app if it is working or not.

7. Job4 : if app is not working , then send email to developer with error messages and redeploy the application after code is being edited by the developer

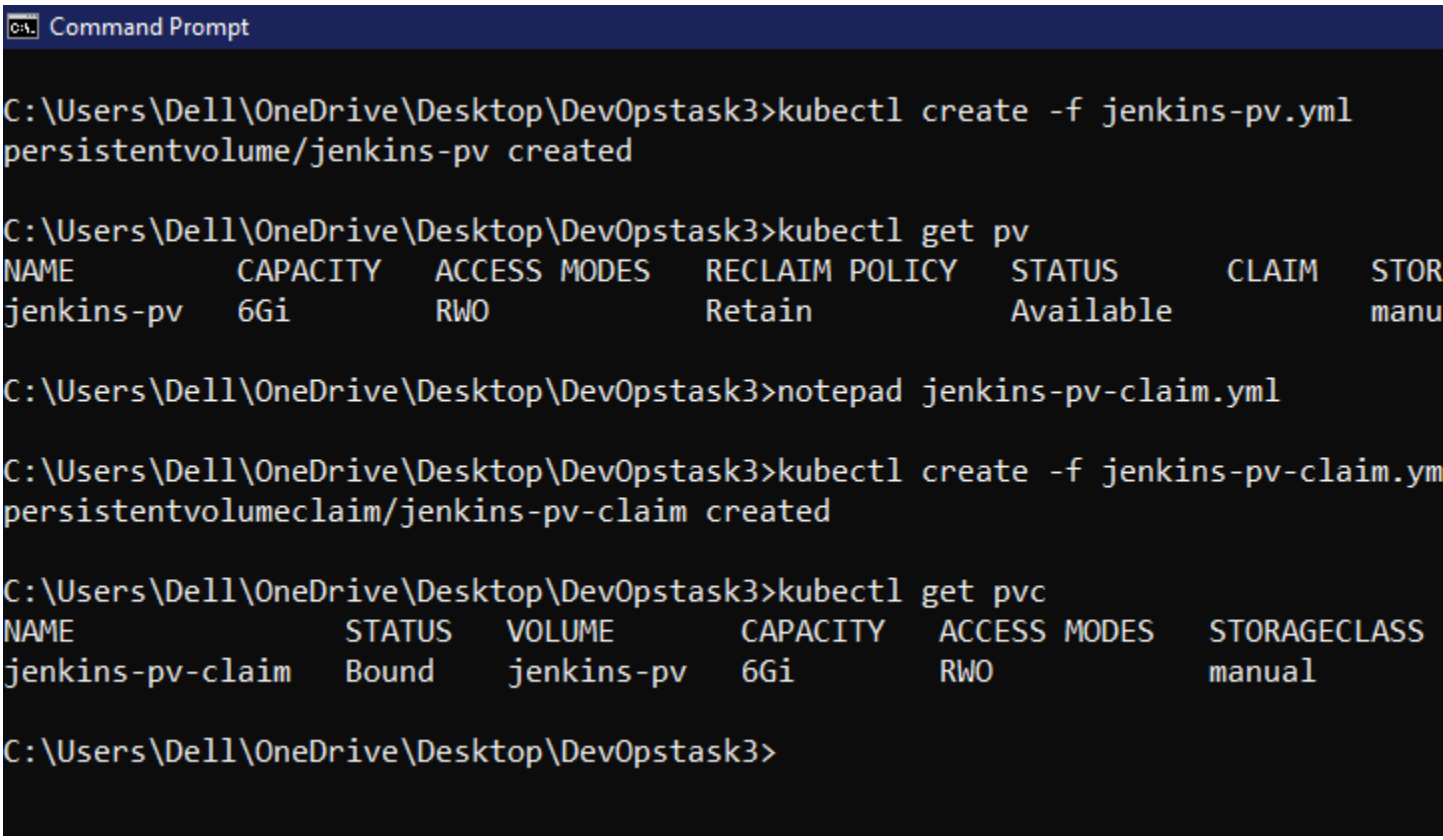
Step 1

Create Persistent Volume (PV) for jenkins



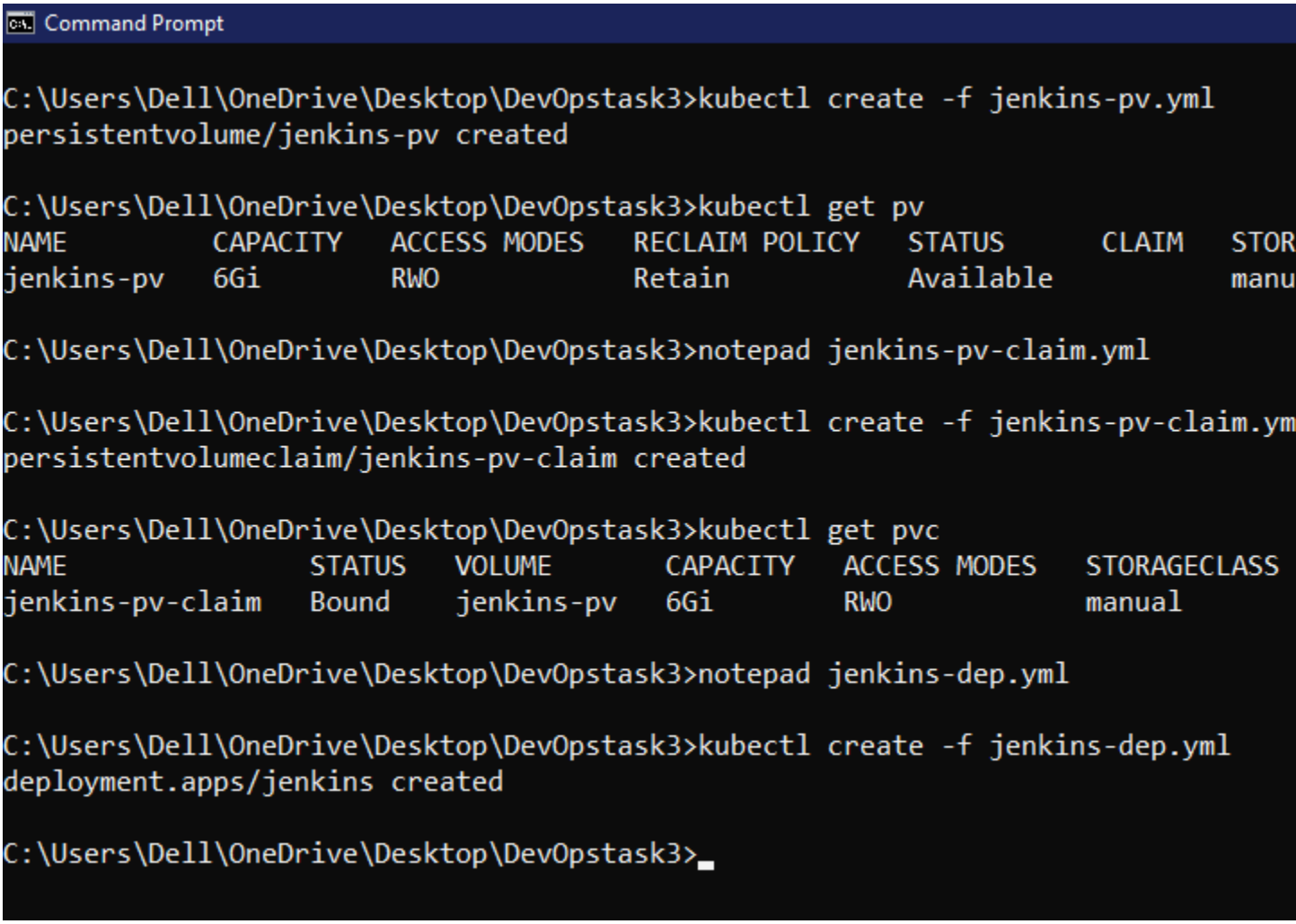
Step 2

Create Persistent Volume Claim(PVC) for jenkins



Step 3

Create deployment for jenkins



This whole process of PV, PVC, Deployment is done so that we create a layer of jenkins over K8s also the data is made persistent by this so that even if by chance the pod gets crashed or shutdown the deployment over it will launch new pod with data coming from PV. This way 2 problems are solved by K8s against docker.

1 if the pod is crashed the deployment will take care of it , we need not worry about the relaunch.

2 the data is permanent so that there is no loss in terms of data making the system more reliable.

Step 4

we need to expose the deployment for external use especially the 8080 port.

```
C:\Users\Dell\OneDrive\Desktop\DevOpstask3>kubectl expose deployment jenkins --p
service/jenkins exposed

C:\Users\Dell\OneDrive\Desktop\DevOpstask3>
```

Step 5

Get the ip of cluster from Minikube , Here 192.168.99.104

```
C:\Users\Dell\OneDrive\Desktop\DevOpstask3>minikube ip
192.168.99.104

C:\Users\Dell\OneDrive\Desktop\DevOpstask3>
```

Step 6

from the NodePort service we get the port no. here 30594

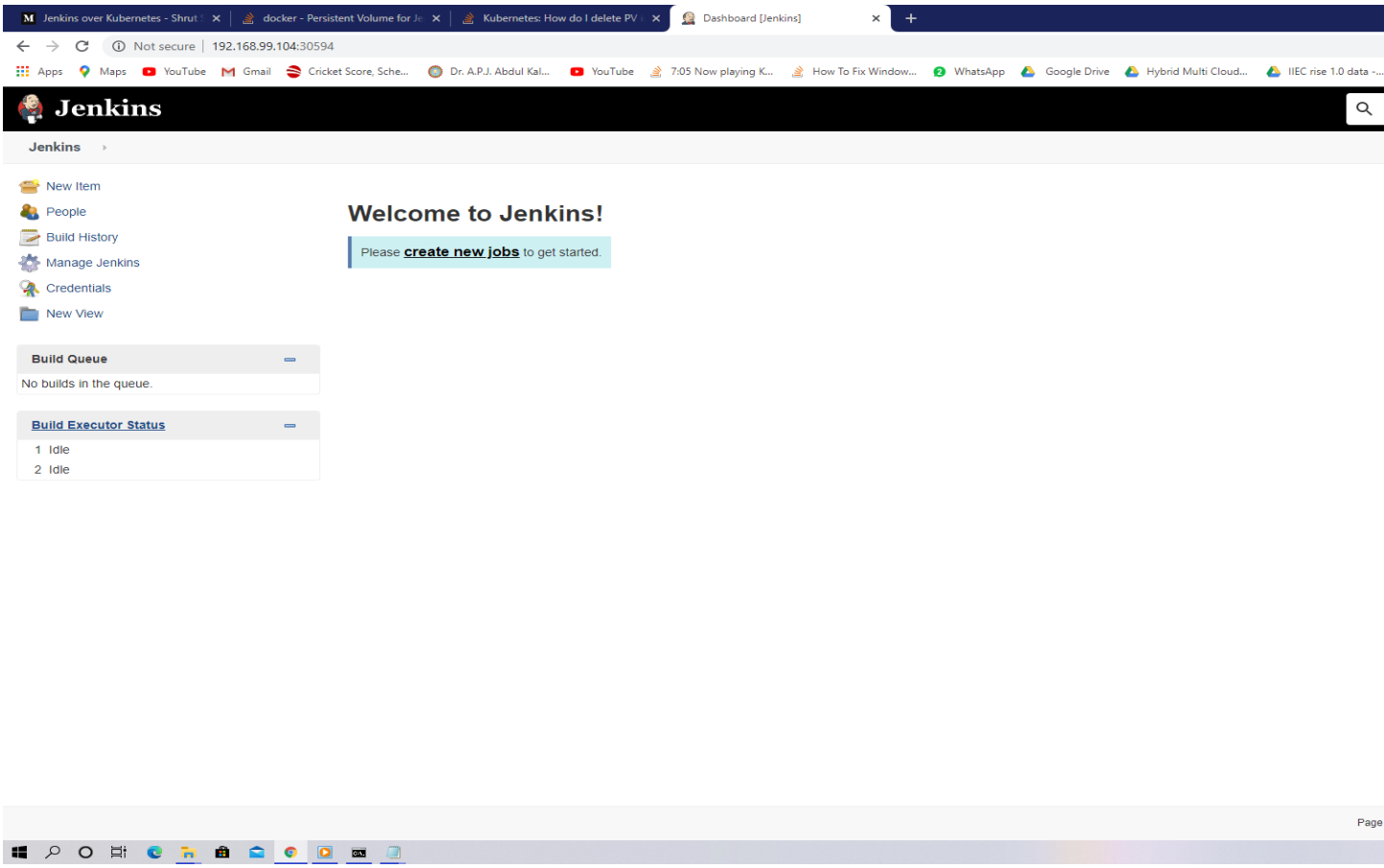
```
C:\Users\Dell\OneDrive\Desktop\DevOpstask3>kubectl get svc
NAME         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
jenkins      NodePort    10.100.127.67   <none>        8080:30594/TCP   4m2s
kubernetes   ClusterIP   10.96.0.1       <none>        443/TCP          53m

C:\Users\Dell\OneDrive\Desktop\DevOpstask3>
```

Combining both 192.168.99.104:30594

Step 7

access this 192.168.99.104:30954 to get the page



Step 8

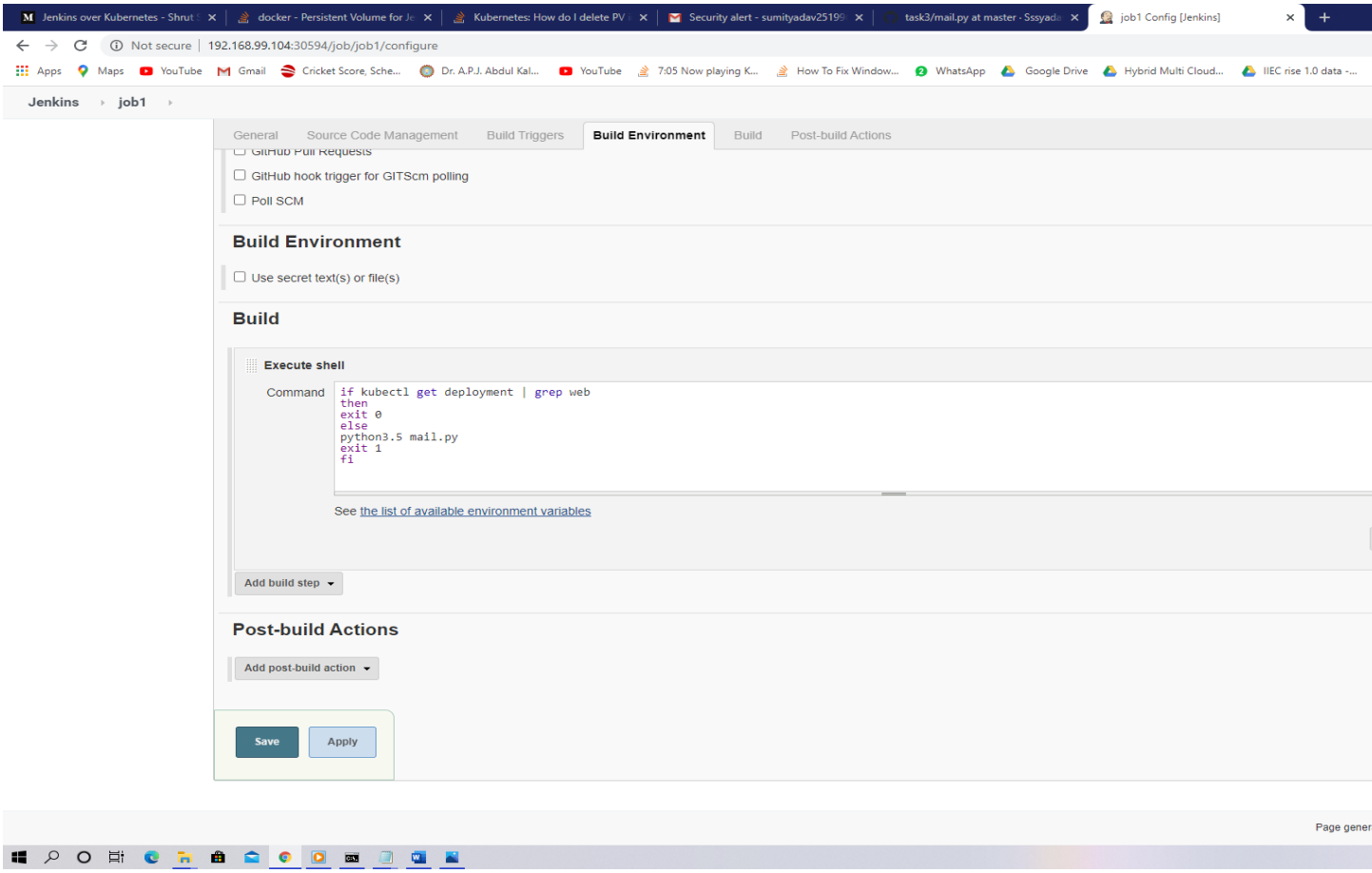
We make one more job here for email notification

if the code runs perfect that’s good but if the code has some error or fault thn we need to send a mail as notification to developer for correction.

Now here the same production and testing environment is created but instead of docker we use K8s .

Now on top of this create deployment of pod for testing and production

if the code is not proper 1 job is run for email notification.



mail.py code

```
# Python code to illustrate Sending mail from
# your Gmail account
import smtplib

# creates SMTP session
s = smtplib.SMTP('smtp.gmail.com', 587)
```

```
# start TLS for security
s.starttls()

# Authentication
s.login("sumityadav25199@gmail.com", "ss********")

# message to be sent
message = "This mail is notify the developer hat your latest code been failed ,please take action on it as soon as possibe"

# sending the mail
s.sendmail("sumityadav25199@gmail.com", "sumityadav25199@gmail.com", message1)

# terminating the session
s.quit()
```