

Submitted by

Vamsikrishna Annadanam

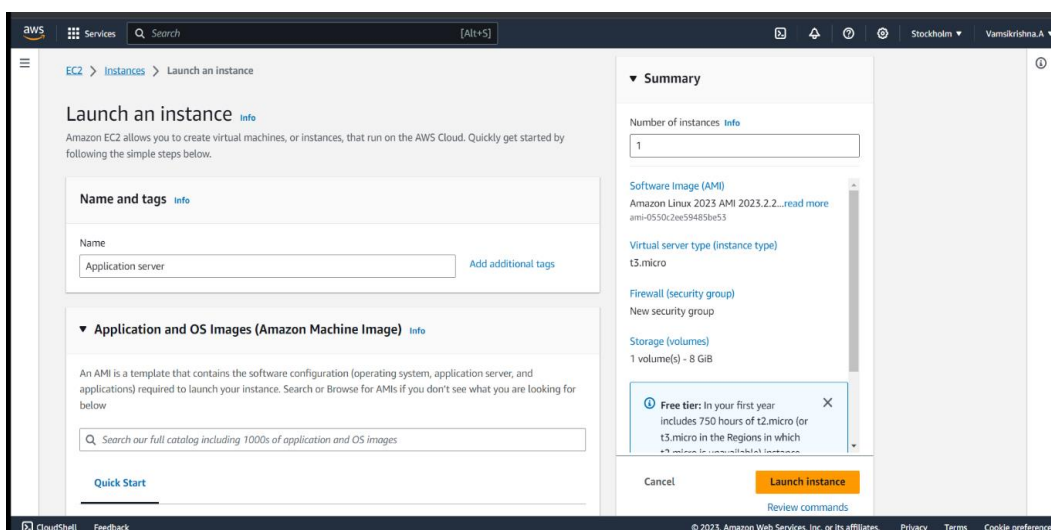
[Vamsiannadanam918@gmail.com](mailto:Vamsiannadanam918@gmail.com)

Ph – 8897672249

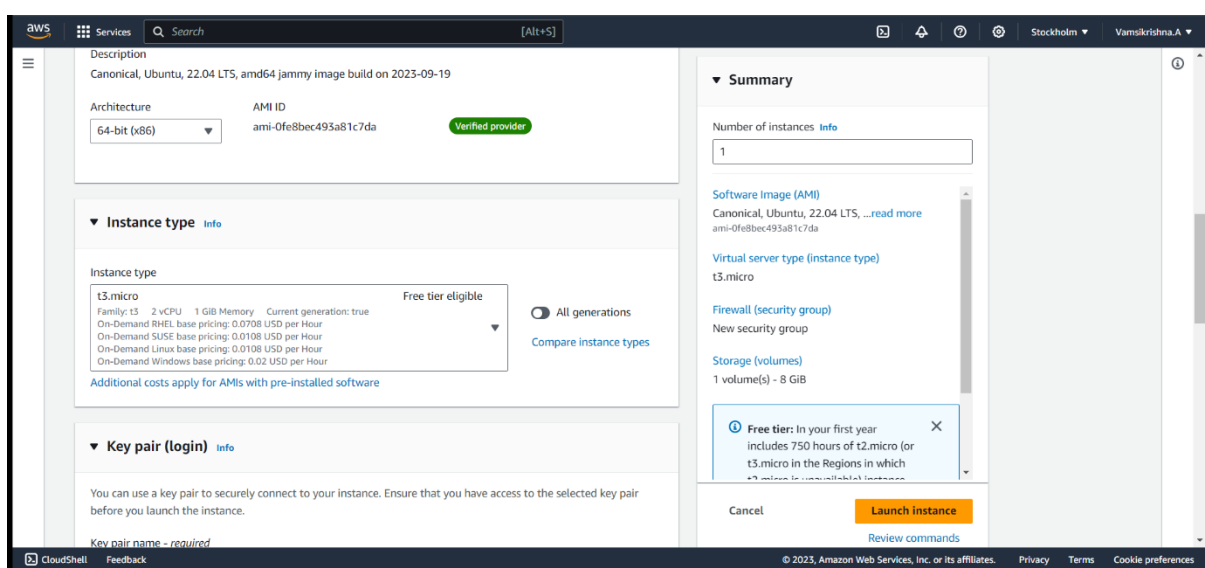
Linked in- <https://www.linkedin.com/in/vamsikrishna-annadanam-0b315a268/>

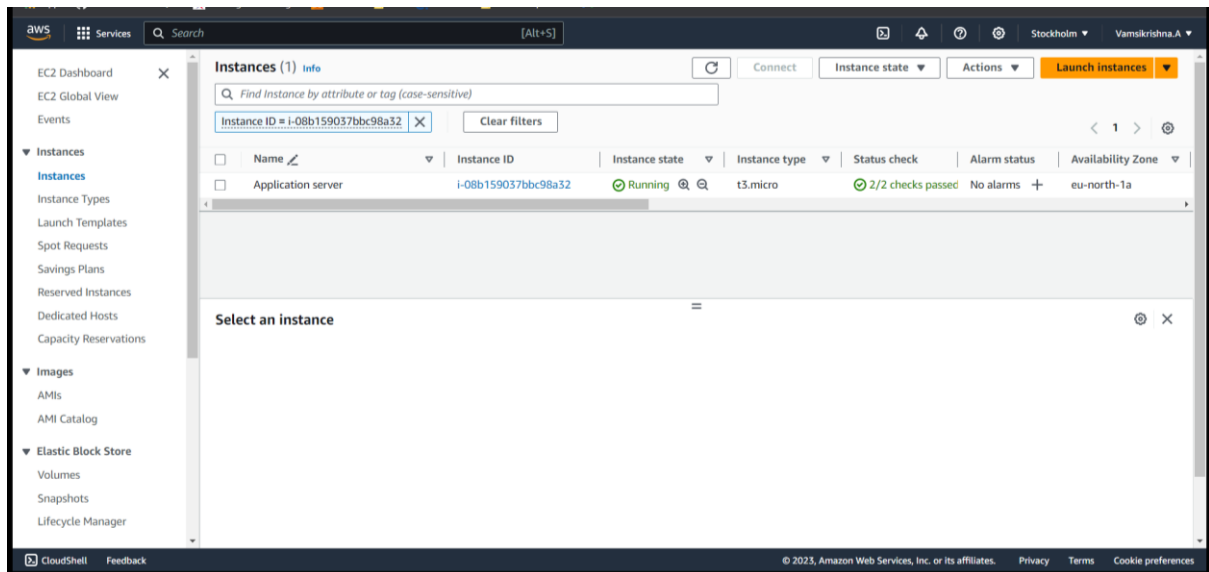
## Python application deployment CICD.

Git repository- <https://github.com/vamsikrishna918/PythonWebApplication>

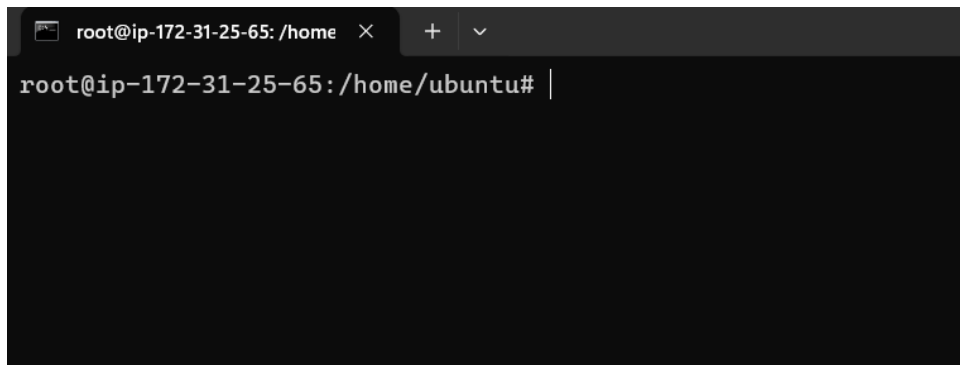


Have gone with free-tier **T3.micro**



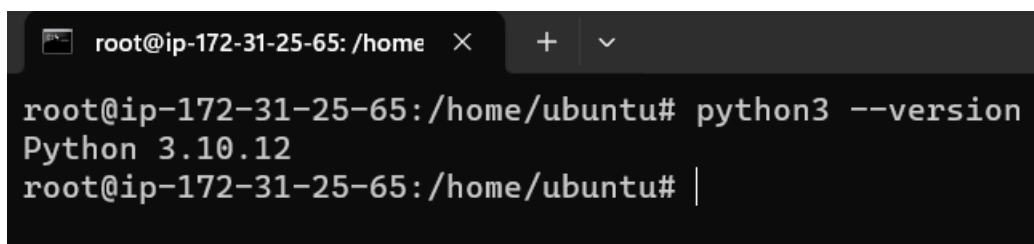


Connected to the instance with SSH via .pem file



**Command** –

- apt update
- Apt install python3



- Creating and updating the python file as below

////

from flask import Flask

app = Flask(\_\_name\_\_)

```

@app.route('/')

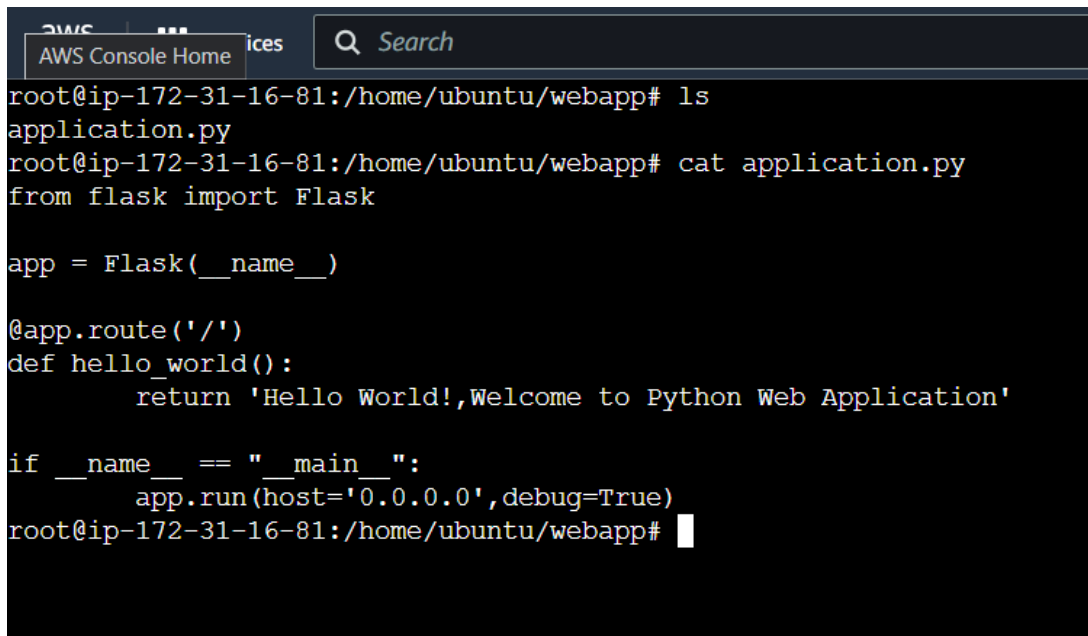
def hello_world():

    return 'Hello World!,Welcome to Python Web Application'

if __name__ == "__main__":

    app.run(host='0.0.0.0',debug=True)

```



The screenshot shows the AWS Console Home interface with a terminal window open. The terminal displays the following commands and output:

```

root@ip-172-31-16-81:/home/ubuntu/webapp# ls
application.py
root@ip-172-31-16-81:/home/ubuntu/webapp# cat application.py
from flask import Flask

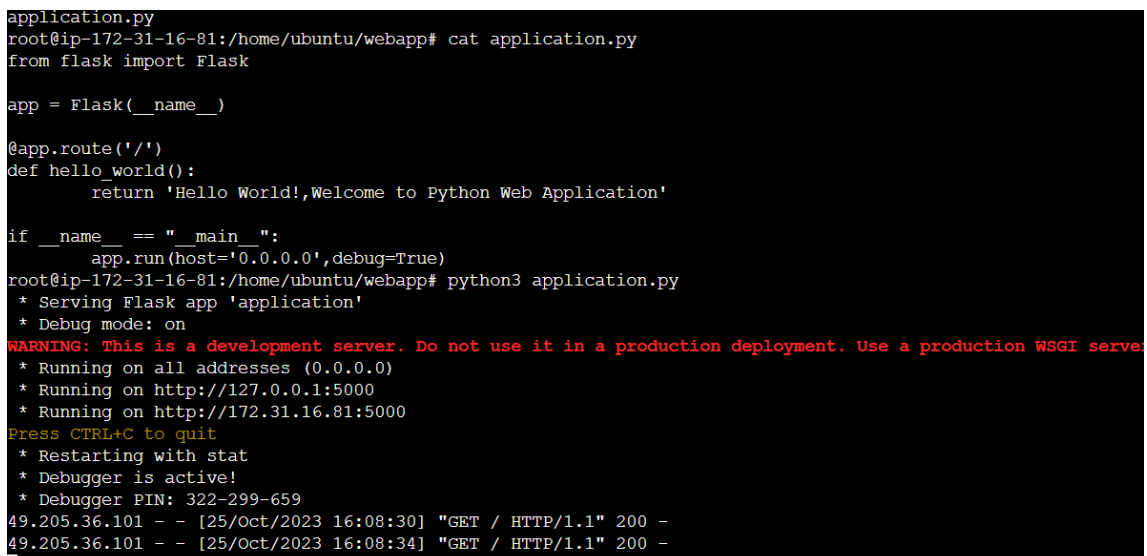
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello World!,Welcome to Python Web Application'

if __name__ == "__main__":
    app.run(host='0.0.0.0',debug=True)
root@ip-172-31-16-81:/home/ubuntu/webapp#

```

- Running the application locally on the instance:



The screenshot shows a terminal window with the following commands and output:

```

application.py
root@ip-172-31-16-81:/home/ubuntu/webapp# cat application.py
from flask import Flask

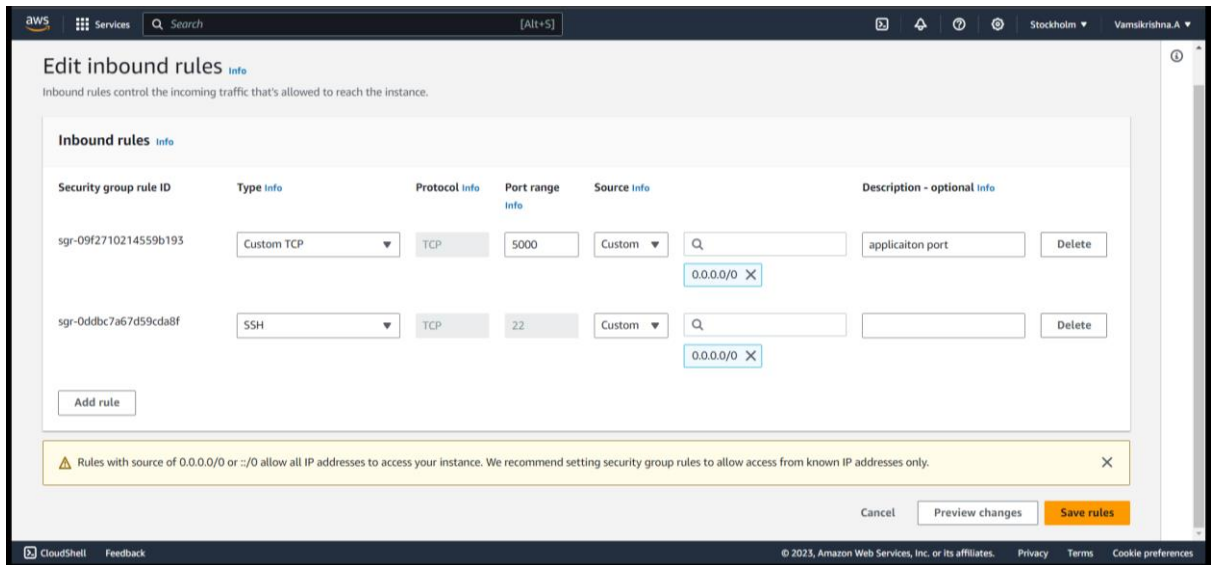
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello World!,Welcome to Python Web Application'

if __name__ == "__main__":
    app.run(host='0.0.0.0',debug=True)
root@ip-172-31-16-81:/home/ubuntu/webapp# python3 application.py
* Serving Flask app 'application'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.31.16.81:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 322-299-659
49.205.36.101 - - [25/Oct/2023 16:08:30] "GET / HTTP/1.1" 200 -
49.205.36.101 - - [25/Oct/2023 16:08:34] "GET / HTTP/1.1" 200 -

```

- Open port of security group port **5000** in Inbound rules



## • Jenkins

- Follow the steps In the below url of official Jenkins doc

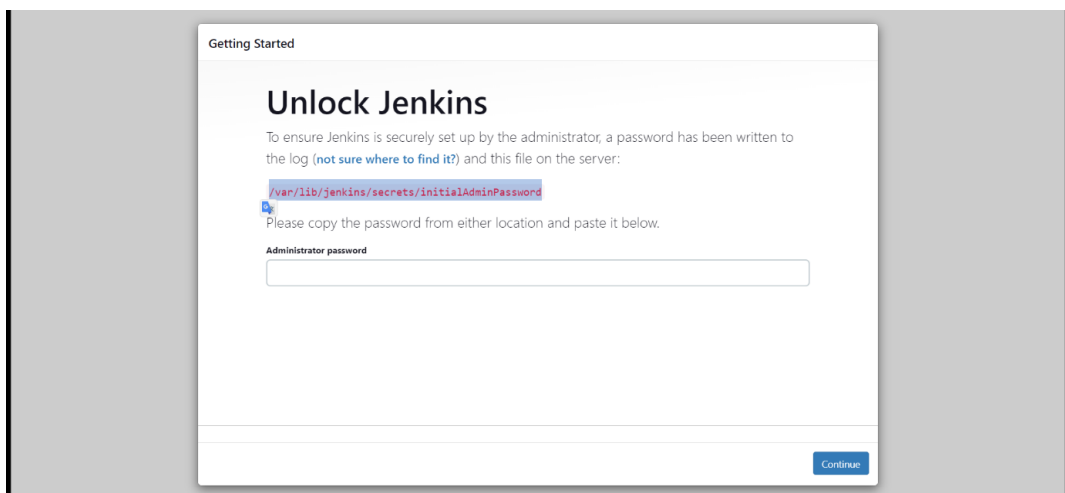
<https://www.jenkins.io/doc/book/installing/linux/#debianubuntu>

```

AWS Console Home 16-81:/home/ubuntu/webapp# java --version
openjdk version "17.0.8.1" 2023-08-24
OpenJDK Runtime Environment (build 17.0.8.1+1-Ubuntu-0ubuntu122.04)
OpenJDK 64-Bit Server VM (build 17.0.8.1+1-Ubuntu-0ubuntu122.04, mixed mode, sharing)
root@ip-172-31-16-81:/home/ubuntu/webapp# jenkins --version
2.414.3
root@ip-172-31-16-81:/home/ubuntu/webapp# ps -ef |grep jenkins
jenkins 7667 1 32 16:17 ? 00:01:13 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080
root 7841 1670 0 16:21 pts/1 00:00:00 grep --color=auto jenkins
root@ip-172-31-16-81:/home/ubuntu/webapp#

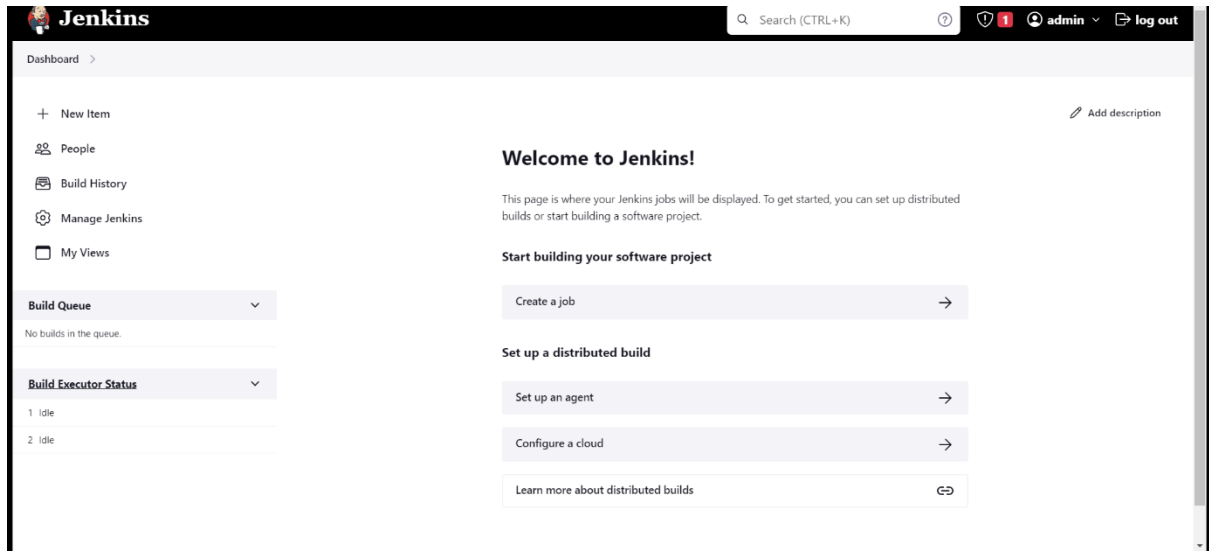
```

- Open port of security group port **8080** in Inbound rules

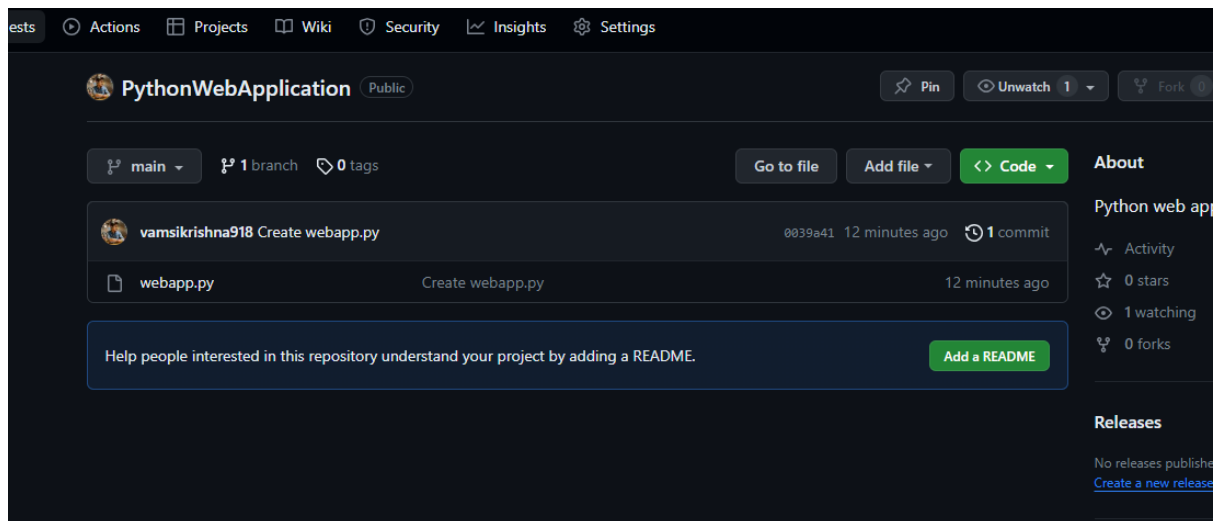


To get initial password:

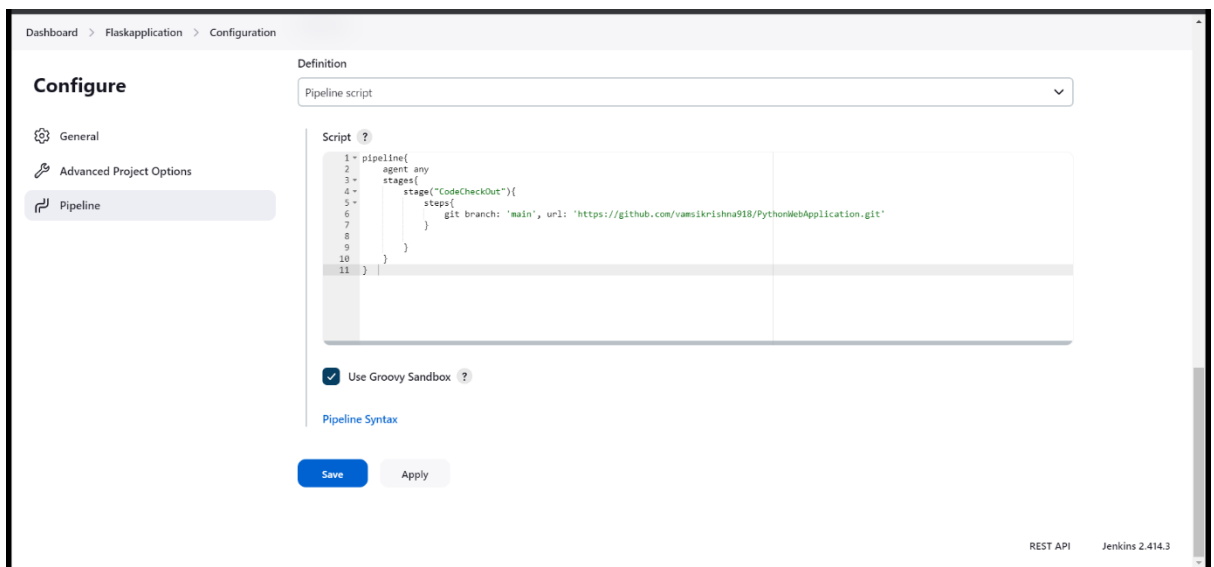
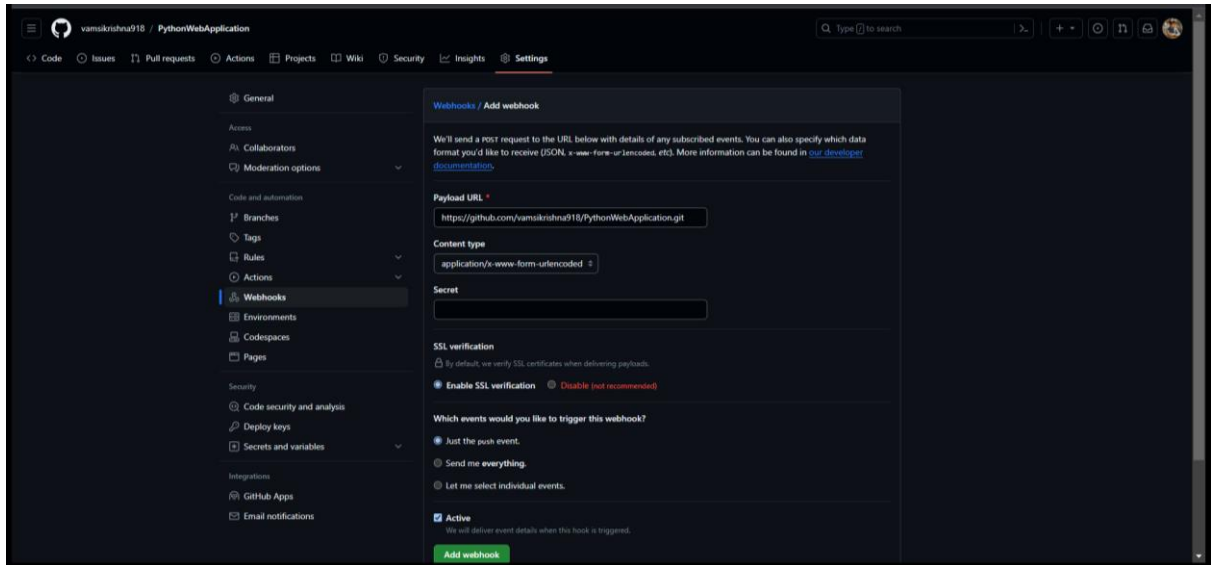
```
cat /var/lib/jenkins/secrets/initialAdminPassword
```



- Commit the application to GitHub



## Webhook:



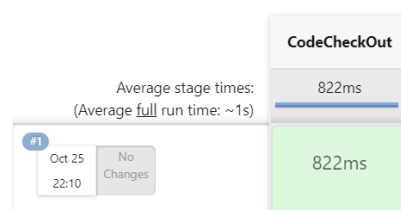
- ▶ Build Now
- ⚙️ Configure
- 🗑️ Delete Pipeline
- 🔍 Full Stage View
- ✎️ Rename
- ❓ Pipeline Syntax

🌞 **Build History** trend ▾

🔍 Filter builds... /

📄 #1  
Oct 25, 2023, 4:40 PM

### Stage View



### Permalinks

- [Last build \(#1\), 51 sec ago](#)
- [Last stable build \(#1\), 51 sec ago](#)

Install docker on ec2 instance:

Follow below url

Docker - <https://docs.docker.com/engine/install/ubuntu/>

```
root@ip-172-31-16-81:/home/ubuntu/webapp# docker --version
Docker version 24.0.6, build ed223bc
root@ip-172-31-16-81:/home/ubuntu/webapp#
```

Create a **Docker file**

```
root@ip-172-31-16-81:/home/ubuntu/webapp# cat Dockerfile
# Use an official Python runtime as the base image
FROM python:3.9-slim

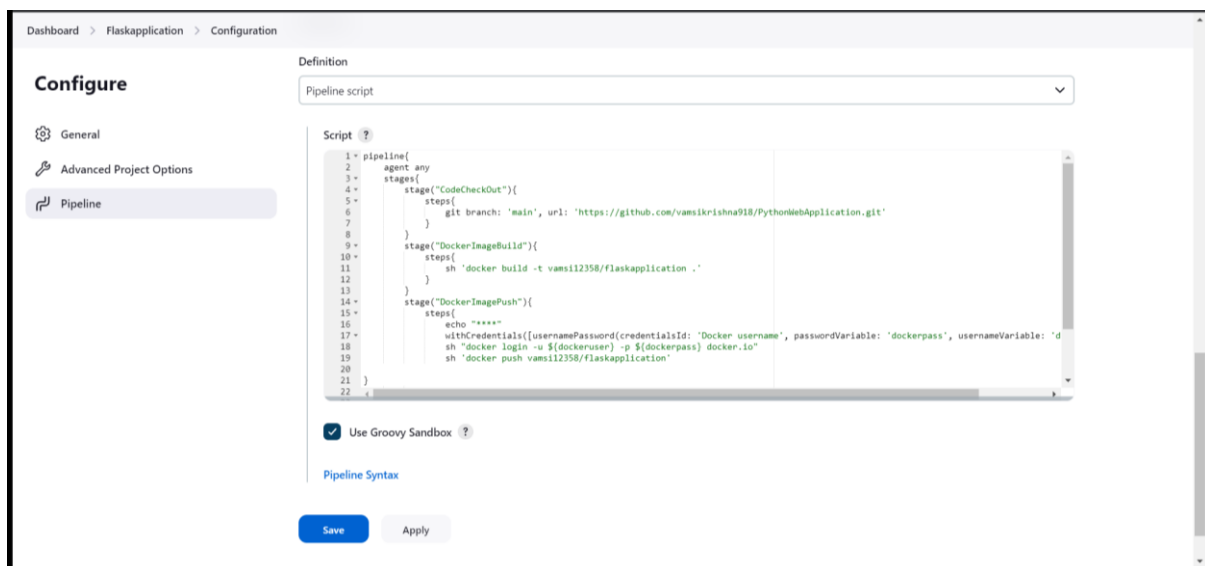
# Set the working directory in the container
WORKDIR /app

# install required packages for system
RUN apt-get update \
    && apt-get upgrade -y \
    && rm -rf /var/lib/apt/lists/* \
    && pip install flask

COPY . .

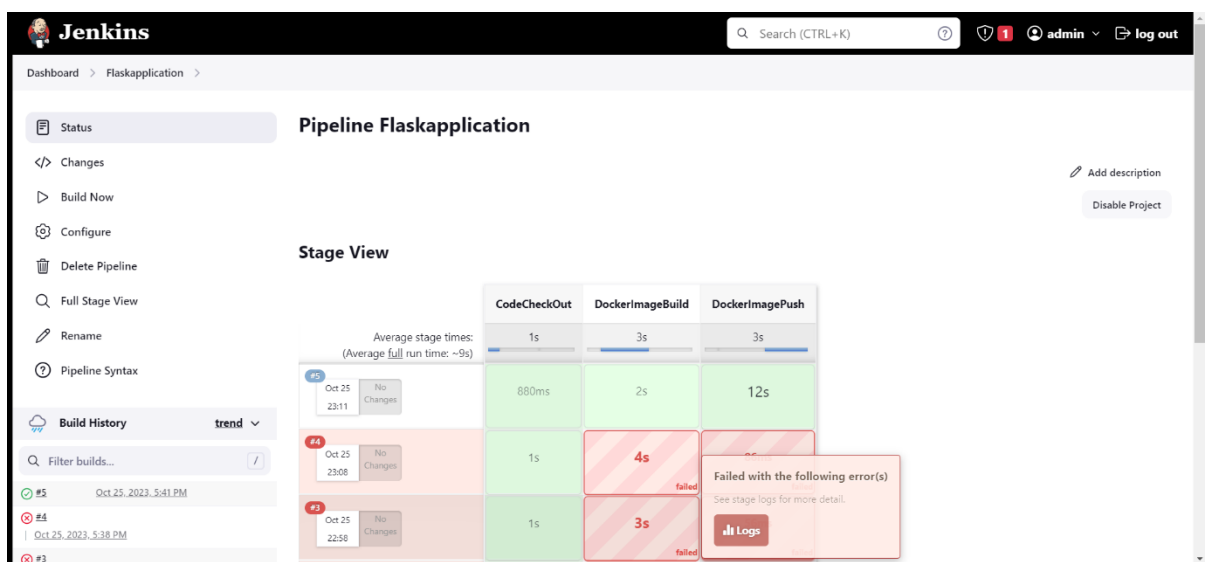
CMD ["python", "webapp.py"]
root@ip-172-31-16-81:/home/ubuntu/webapp#
```

## Building the pipeline and push the image to **Docker Hub**:



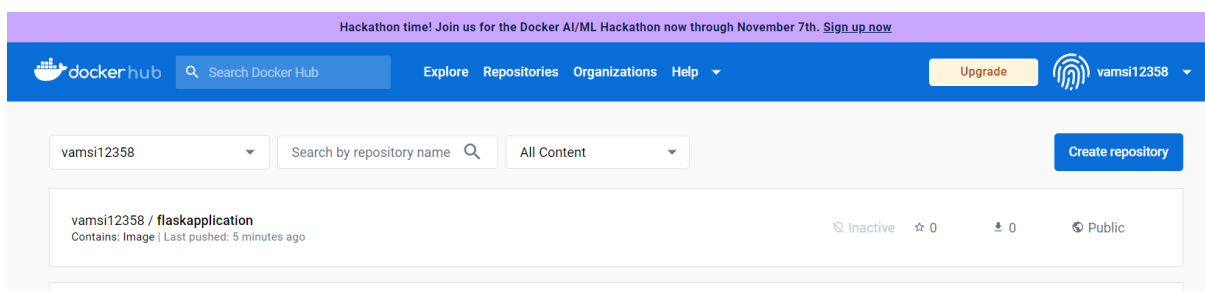
The screenshot shows the Jenkins 'Configure' page for a pipeline named 'Flaskapplication'. The 'Definition' dropdown is set to 'Pipeline script'. The 'Script' section contains a Groovy pipeline script with three stages: 'CodeCheckOut', 'DockerImageBuild', and 'DockerImagePush'. The 'DockerImageBuild' stage runs 'docker build -t vamsi12358/flaskapplication .'. The 'DockerImagePush' stage runs 'docker login' and 'docker push vamsi12358/flaskapplication'. The 'Use Groovy Sandbox' checkbox is checked. The 'Save' button is highlighted.

```
1 pipeline{
2   agent any
3   stages{
4     stage("CodeCheckOut"){
5       steps{
6         git branch: 'main', url: 'https://github.com/vamsikrishna918/PythonWebApplication.git'
7       }
8     }
9     stage("DockerImageBuild"){
10      steps{
11        sh 'docker build -t vamsi12358/flaskapplication .'
12      }
13    }
14    stage("DockerImagePush"){
15      steps{
16        echo "*****"
17        withCredentials([usernamePassword(credentialId: 'Docker username', passwordVariable: 'dockerpass', usernameVariable: 'd
18        sh 'docker login -u ${dockeruser} -p ${dockerpass} docker.io'
19        sh 'docker push vamsi12358/flaskapplication'
20      }
21    }
22  }
```



The screenshot shows the Jenkins 'Pipeline Flaskapplication' view. The 'Stage View' tab is active, displaying a table of stage execution times. The 'CodeCheckOut' stage has an average time of 1s. The 'DockerImageBuild' stage has an average time of 3s. The 'DockerImagePush' stage has an average time of 3s. The table shows three builds: #5 (Oct 25, 23:11), #4 (Oct 25, 23:08), and #3 (Oct 25, 22:58). Build #4 failed with the error 'Failed with the following error(s)'. The 'Build History' section on the left shows the build status and time for each build.

Build	CodeCheckOut	DockerImageBuild	DockerImagePush
#5 (Oct 25, 23:11)	880ms	2s	12s
#4 (Oct 25, 23:08)	1s	4s	Failed
#3 (Oct 25, 22:58)	1s	3s	Failed



The screenshot shows the Docker Hub repository page for 'vamsi12358 / flaskapplication'. The repository is listed as 'Inactive' with 0 stars and 0 forks. The 'Contains: Image' field is visible. The 'Create repository' button is highlighted.

Hackathon time! Join us for the Docker AI/ML Hackathon now through November 7th. [Sign up now](#)

dockerhub Search Docker Hub Explore Repositories Organizations Help Upgrade vamsi12358

vamsi12358 Search by repository name All Content Create repository

vamsi12358 / flaskapplication Contains: Image | Last pushed: 5 minutes ago Inactive 0 0 Public



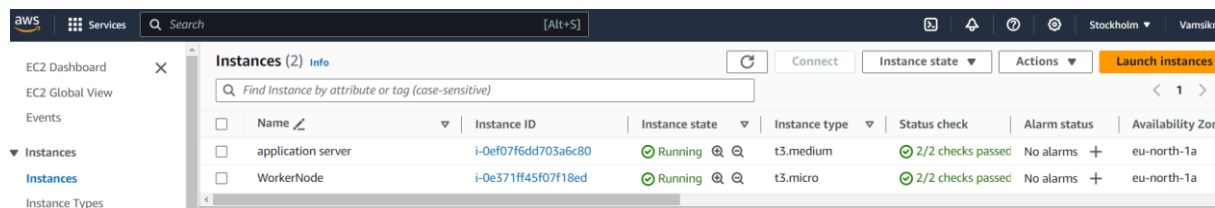
## Kubernetes:

Installation:

kubeadm

[https://github.com/LondheShubham153/kubestarter/blob/main/kubeadm installation.md](https://github.com/LondheShubham153/kubestarter/blob/main/kubeadm%20installation.md)

Create a ec2 instance as worker node:



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
application server	i-0ef07f6dd703a6c80	Running	t3.medium	2/2 checks passed	No alarms	eu-north-1a
WorkerNode	i-0e371ff45f07f18ed	Running	t3.micro	2/2 checks passed	No alarms	eu-north-1a

Setup the master and worker nodes

```
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...
I1025 18:21:16.252164      4903 cert_rotation.go:137] Starting client certificate rota
I1025 18:21:16.255089      4903 kubelet.go:188] [kubelet-start] preserving the crisoc
I1025 18:21:16.255111      4903 patchnode.go:30] [patchnode] Uploading the CRI Socket
annotation
```

This node has joined the cluster:

- \* Certificate signing request was sent to apiserver and a response was received.
- \* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

```
root@ip-172-31-23-127:/home/ubuntu#
```

i-0e371ff45f07f18ed (WorkerNode)

PublicIPs: 13.48.84.250 PrivateIPs: 172.31.23.127

```
p-172-31-16-81 Ready control-plane,master 7m15s v1.20.0
root@ip-172-31-16-81:/home/ubuntu# kubectl get nodes
NAME                STATUS    ROLES                  AGE     VERSION
p-172-31-16-81      Ready    control-plane,master   13m     v1.20.0
p-172-31-23-127     Ready    <none>                 3m3s    v1.20.0
root@ip-172-31-16-81:/home/ubuntu#
```

i-0ef07f6dd703a6c80 (application server)

PublicIPs: 13.49.76.94 PrivateIPs: 172.31.16.81

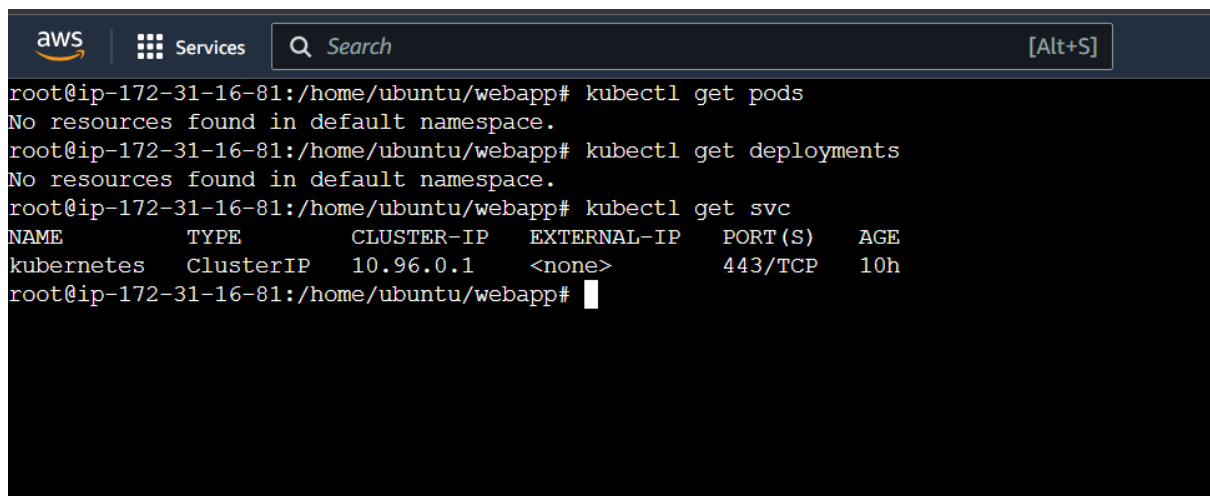
**Created a deployment and service manifests.**

With commands on master – **kubectl apply -f deployment.yml**

And post that

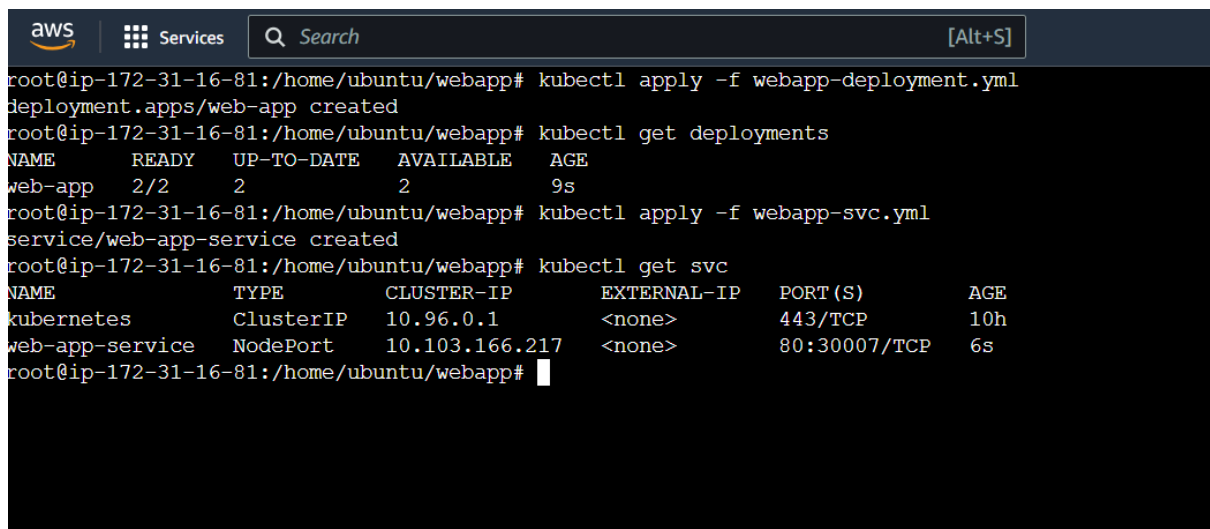
**Kubectl apply -f service.yml** To up our cluster

Before applying-



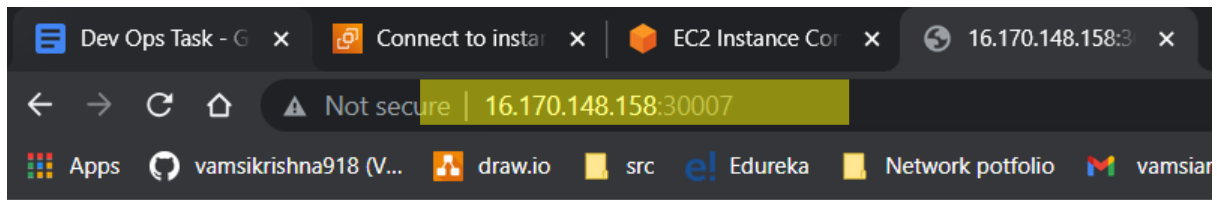
```
aws | Services | Search [Alt+S]
root@ip-172-31-16-81:/home/ubuntu/webapp# kubectl get pods
No resources found in default namespace.
root@ip-172-31-16-81:/home/ubuntu/webapp# kubectl get deployments
No resources found in default namespace.
root@ip-172-31-16-81:/home/ubuntu/webapp# kubectl get svc
NAME          TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
kubernetes    ClusterIP     10.96.0.1     <none>         443/TCP    10h
root@ip-172-31-16-81:/home/ubuntu/webapp#
```

After:



```
aws | Services | Search [Alt+S]
root@ip-172-31-16-81:/home/ubuntu/webapp# kubectl apply -f webapp-deployment.yml
deployment.apps/web-app created
root@ip-172-31-16-81:/home/ubuntu/webapp# kubectl get deployments
NAME    READY    UP-TO-DATE    AVAILABLE    AGE
web-app  2/2      2             2            9s
root@ip-172-31-16-81:/home/ubuntu/webapp# kubectl apply -f webapp-svc.yml
service/web-app-service created
root@ip-172-31-16-81:/home/ubuntu/webapp# kubectl get svc
NAME             TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
kubernetes       ClusterIP     10.96.0.1     <none>         443/TCP          10h
web-app-service   NodePort      10.103.166.217 <none>         80:30007/TCP     6s
root@ip-172-31-16-81:/home/ubuntu/webapp#
```

Open port **30007** on worker node in security Inbound rules to access our application.



Hello World!, Welcome to Python Web Application

## Running on worker node

aws Services Search [Alt+S]

```
ubuntu@ip-172-31-23-127:~$ docker images
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: G
/var/run/docker.sock: connect: permission denied
ubuntu@ip-172-31-23-127:~$ docker images
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: G
/var/run/docker.sock: connect: permission denied
ubuntu@ip-172-31-23-127:~$ sudo usermod -aG docker $USER
ubuntu@ip-172-31-23-127:~$ docker images
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: G
ubuntu@ip-172-31-23-127:~$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
vamsi12358/flaskapplication	latest	e66b2e44f5b3	11 hours ago	146MB
k8s.gcr.io/kube-proxy	v1.20.15	46e2cd1b2594	21 months ago	99.7MB
weaveworks/weave-npc	latest	690c3345cc9c	2 years ago	39.3MB
weaveworks/weave-kube	latest	62fea85d6052	2 years ago	89MB
k8s.gcr.io/pause	3.2	80d28bedfe5d	3 years ago	683kB

```
ubuntu@ip-172-31-23-127:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
f36fe3534e3e	vamsi12358/flaskapplication	"python webapp.py"	9 minutes ago	Up 9 minutes
1-bd03-413d-9255-120528df6336_0				
4a4a8bf73de7	vamsi12358/flaskapplication	"python webapp.py"	9 minutes ago	Up 9 minutes
2-af05-46ad-b30b-eb2065e9152c_0				
dc38efaa6de8	k8s.gcr.io/pause:3.2	"/pause"	9 minutes ago	Up 9 minutes
d-9255-120528df6336_0				
26aa36e5b4e4	k8s.gcr.io/pause:3.2	"/pause"	9 minutes ago	Up 9 minutes
d-b30b-eb2065e9152c_0				

i-0e371ff45f07f18ed (WorkerNode)

PublicIPs: 13.51.237.3 PrivateIPs: 172.31.23.127

## Scaling manually.

Before scaling: running 2 pods

```
service/web-app-service unchanged
root@ip-172-31-16-81:/home/ubuntu/webapp# kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
web-app-76d4f9ff6-kkgx9             1/1     Running   0           8s
web-app-76d4f9ff6-nhm6k             1/1     Running   0           8s
root@ip-172-31-16-81:/home/ubuntu/webapp#
```

## Increasing replicas to 4:

kubectl scale deployment web-app --replicas=4

```
root@ip-172-31-16-81:/home/ubuntu/webapp# kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
web-app-76d4f9ff6-kkgx9             1/1     Running   0           3m23s
web-app-76d4f9ff6-nhm6k             1/1     Running   0           3m23s
root@ip-172-31-16-81:/home/ubuntu/webapp# kubectl get deployment
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
web-app   2/2     2             2           3m47s
root@ip-172-31-16-81:/home/ubuntu/webapp# kubectl scale deployment web-app --replicas=4
deployment.apps/web-app scaled
root@ip-172-31-16-81:/home/ubuntu/webapp# kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
web-app-76d4f9ff6-kkgx9             1/1     Running   0           4m
web-app-76d4f9ff6-nhm6k             1/1     Running   0           4m
web-app-76d4f9ff6-qhhnd             1/1     Running   0           5s
web-app-76d4f9ff6-svk4d             1/1     Running   0           5s
root@ip-172-31-16-81:/home/ubuntu/webapp# kubectl get deployment
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
web-app   4/4     4             4           4m6s
root@ip-172-31-16-81:/home/ubuntu/webapp#
```

## Creating HPA:

**If cpu utilization is more than 50 % then the pods will be scaled up automatically.**

apiVersion: autoscaling/v1

kind: HorizontalPodAutoscaler

metadata:

name: hpa-demo-deployment

spec:

scaleTargetRef:

apiVersion: apps/v1

kind: Deployment

```
name: webapp-deployment
```

minReplicas: 2

maxReplicas: 4

targetCPUUtilizationPercentage: 50

```

AWS Console Home Search [Alt+S]
root@ip-172-31-16-81:/home/ubuntu/webapp# kubectl get hpa
NAME                REFERENCE                TARGETS      MINPODS   MAXPODS   REPLICAS   AGE
hpa-demo-deployment Deployment/webapp-deployment <unknown>/50% 2          4         0          8s
root@ip-172-31-16-81:/home/ubuntu/webapp#

```

# CD with Jenkins.

The screenshot shows the Jenkins Pipeline Flaskapplication dashboard. The left sidebar contains navigation links: Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, Rename, and Pipeline Syntax. The main area displays the Stage View for the Pipeline Flaskapplication. It shows a table of stages and their durations. The stages are CodeCheckOut, ImagePrune, DockerImageBuild, DockerImagePush, and Deployment and Service Apply. The table shows the average stage times and the average full run time. The build history shows two builds: #9 (Oct 26, 2023, 4:36 PM) and #8 (Oct 26, 2023, 4:31 PM). The build #8 is marked as failed.

Stage	CodeCheckOut	ImagePrune	DockerImageBuild	DockerImagePush	Deployment and Service Apply
Average stage times:	944ms	609ms	2s	9s	1s
Average full run time: ~15s					
#9	734ms	439ms	1s	9s	1s
#8	1s	779ms	3s	10s	1s

Permalinks

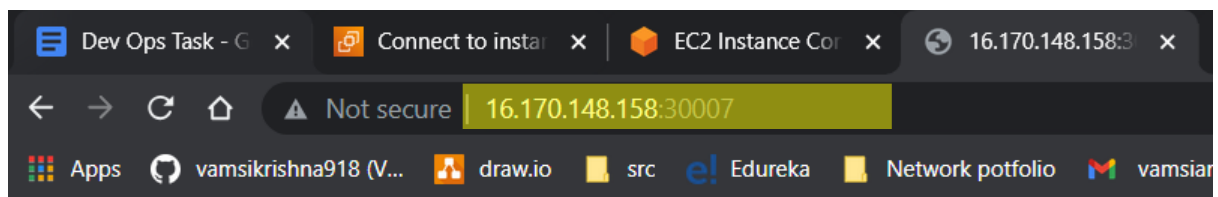
The screenshot shows the Jenkins Pipeline Flaskapplication console output for build #9. The output displays the pipeline script and the execution steps. The pipeline script is as follows:

```
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deployment and Service Apply)
[Pipeline] withKubeCredentials
[Pipeline] {
[Pipeline] sh
+ kubectl apply -f webapp-deployment.yml
deployment.apps/web-app created
[Pipeline] echo
**Deployment created**
[Pipeline] sh
+ kubectl apply -f webapp-svc.yml
service/web-app-service created
[Pipeline] echo
**Service created**
[Pipeline] }
[kubernetes-cli] kubectl configuration cleaned up
[Pipeline] // withKubeCredentials
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

REST API Jenkins 2.414.3

```
aws | Services | Search [Alt+S]
root@ip-172-31-16-81:/home/ubuntu/webapp# kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
web-app-76d4f9ff6-6fwkm            1/1     Running   0           4m25s
web-app-76d4f9ff6-s7cgr            1/1     Running   0           4m25s
root@ip-172-31-16-81:/home/ubuntu/webapp# kubectl get deploy
NAME    READY   UP-TO-DATE   AVAILABLE   AGE
web-app 2/2      2             2           4m32s
root@ip-172-31-16-81:/home/ubuntu/webapp# kubectl get svc
NAME                TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
kubernetes          ClusterIP   10.96.0.1    <none>        443/TCP          22h
web-app-service     NodePort    10.97.35.228 <none>        80:30007/TCP     4m40s
root@ip-172-31-16-81:/home/ubuntu/webapp#
```

## Technical Checkout-



Hello World!, Welcome to Python Web Application

