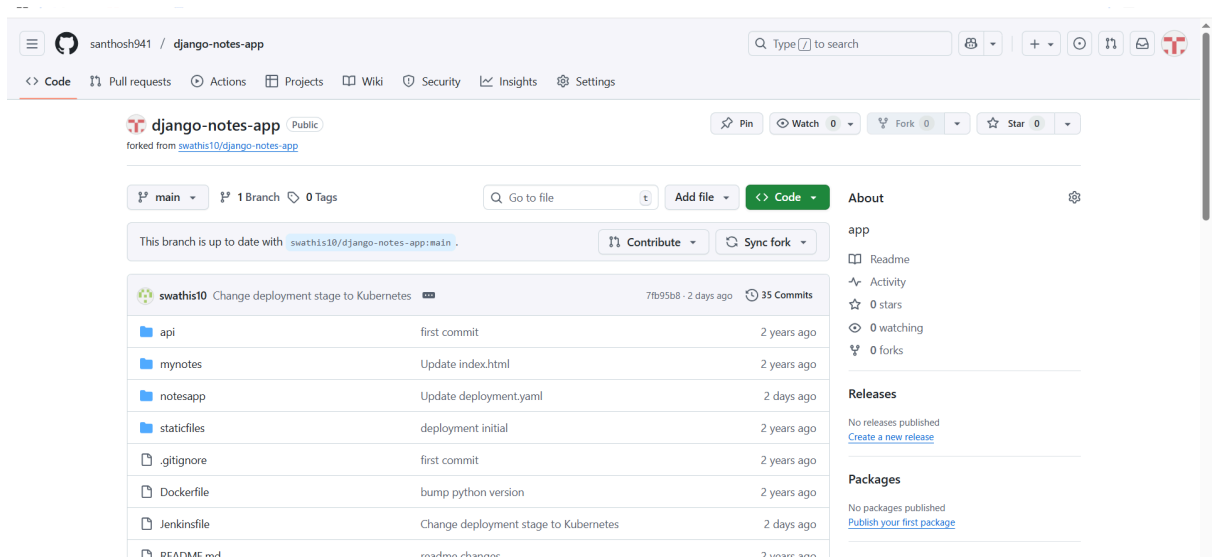


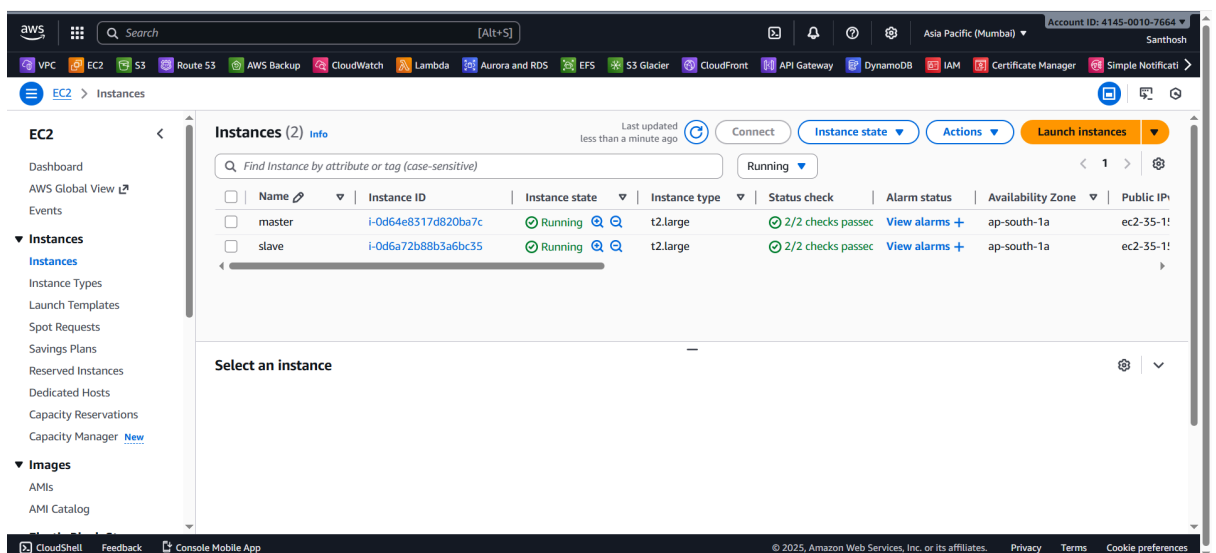
Devops Project

Automation of CI/CD pipeline for django web application using Aws,git,jenkins,docker,dockerhub,kubernetes

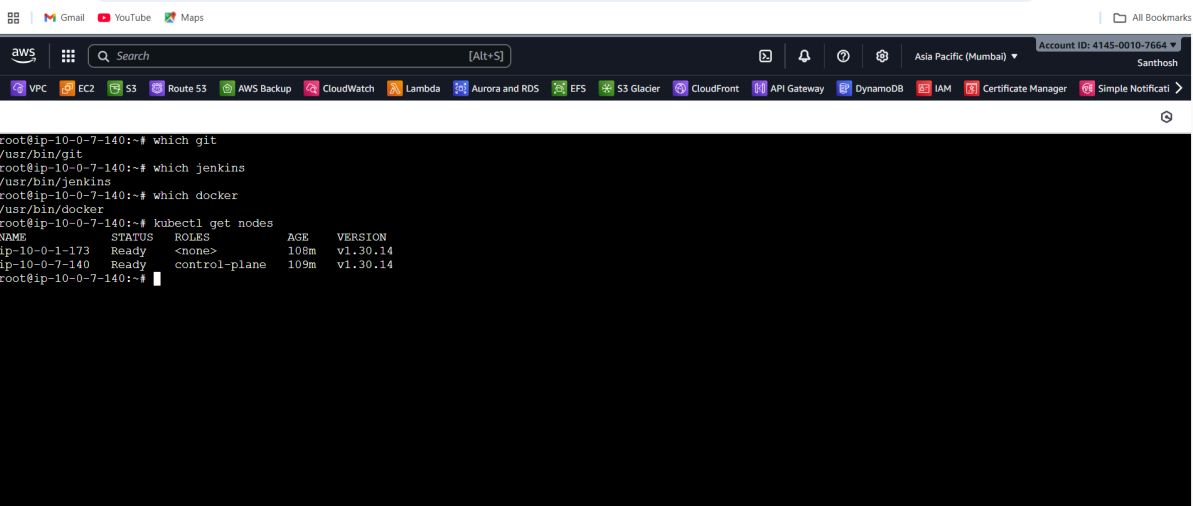
1. Github repository



2. Launched 2 ubuntu EC2 virtual machines.



3.Installed all the required tools jenkins,Docker,Kubeadm.

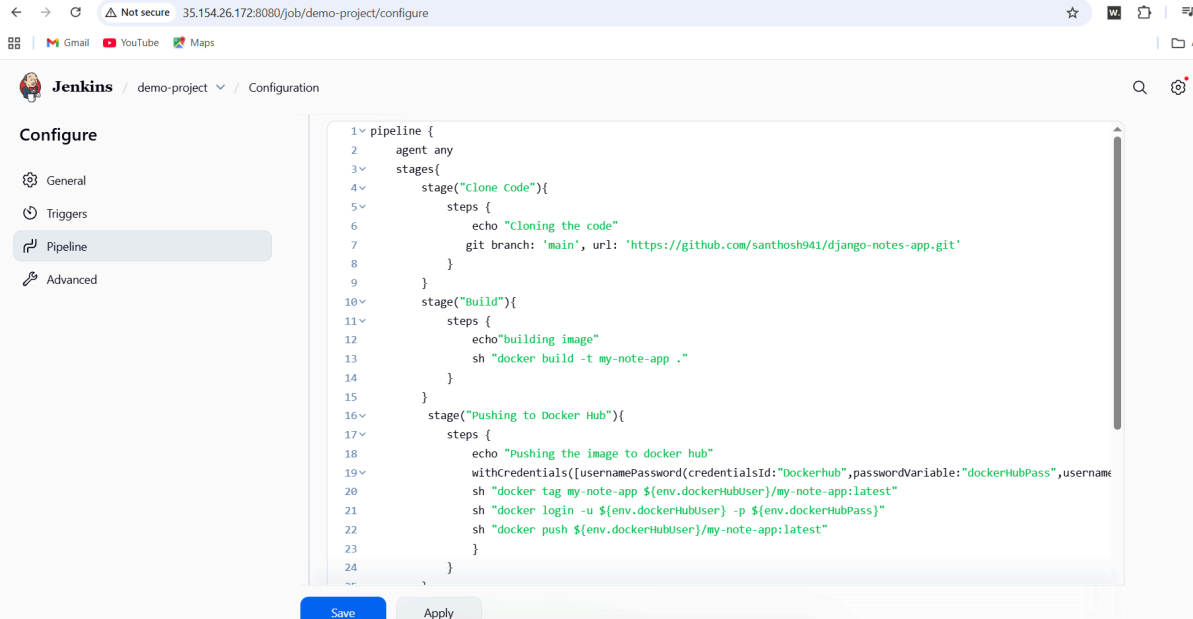


The screenshot shows an AWS CloudShell terminal window. The terminal output is as follows:

```
root@ip-10-0-7-140:~# which git
/usr/bin/git
root@ip-10-0-7-140:~# which jenkins
/usr/bin/jenkins
root@ip-10-0-7-140:~# which docker
/usr/bin/docker
root@ip-10-0-7-140:~# kubectl get nodes
NAME                STATUS    ROLES    AGE   VERSION
ip-10-0-1-173       Ready    <none>    108m  v1.30.14
ip-10-0-7-140       Ready    control-plane  109m  v1.30.14
```

Below the terminal output, the instance ID is shown: `i-0d64e8317d820ba7c (master)`. At the bottom, there is a footer with public and private IP addresses, and a copyright notice for Amazon Web Services, Inc. or its affiliates.

4.Accessed Jenkins Gui and created ci/cd pipeline.



The screenshot shows the Jenkins GUI Configuration page for a project named 'demo-project'. The 'Pipeline' tab is selected in the left sidebar. The main area displays a pipeline script in a text editor. The script is as follows:

```
1 pipeline {
2   agent any
3   stages{
4     stage("Clone Code"){
5       steps {
6         echo "Cloning the code"
7         git branch: 'main', url: 'https://github.com/santhosh941/django-notes-app.git'
8       }
9     }
10    stage("Build"){
11      steps {
12        echo "building image"
13        sh "docker build -t my-note-app ."
14      }
15    }
16    stage("Pushing to Docker Hub"){
17      steps {
18        echo "Pushing the image to docker hub"
19        withCredentials([usernamePassword(credentialId:'Dockerhub',passwordVariable:"dockerHubPass",username
20        sh "docker tag my-note-app ${env.dockerHubUser}/my-note-app:latest"
21        sh "docker login -u ${env.dockerHubUser} -p ${env.dockerHubPass}"
22        sh "docker push ${env.dockerHubUser}/my-note-app:latest"
23      }
24    }
25  }
```

At the bottom of the configuration page, there are 'Save' and 'Apply' buttons.

5.credentials for accessing docker hub and kubernetes.

← → ↺ ⚠ Not secure 35.154.26.172:8080/manage/credentials/store/system/domain/

Jenkins

Manage Jenkins

Credentials

System

Global credentials (unrestr...

🔍

⚙️

👤

Global credentials (unrestricted)

+ Add Credentials

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
Dockerhub	santhosh97401/***** (dockerhub credential)	Username with password	dockerhub credential
kubuernetes	secret.txt	Secret file	

Icon: S M L

REST API

Jenkins 2.528.1

6. Installed stage view plugin and executed all the stages

← → ↺ ⚠ Not secure 35.154.26.172:8080/job/demo-project/

Jenkins

demo-project

🔍

⚙️

👤

Build View

Configure

Delete Pipeline

Full Stage View

GitHub

Stages

Rename

Pipeline Syntax

Credentials

Builds

Filter

Today

#11 11:29 AM

#10 10:54 AM

#9 10:53 AM

#8 10:14 AM

Average stage times: (full run time: ~28s)

	Clone Code	Build	Pushing to Docker Hub	Deploy to kubernetes
#11 Nov 08 16:59 No Changes	770ms	3s	15s	2s
#10 Nov 08 16:24 No Changes	692ms	1s	19s	
#9 Nov 08 16:23 No Changes	668ms	2s	160ms failed	
#8 Nov 08 15:44 No Changes	682ms	36s		

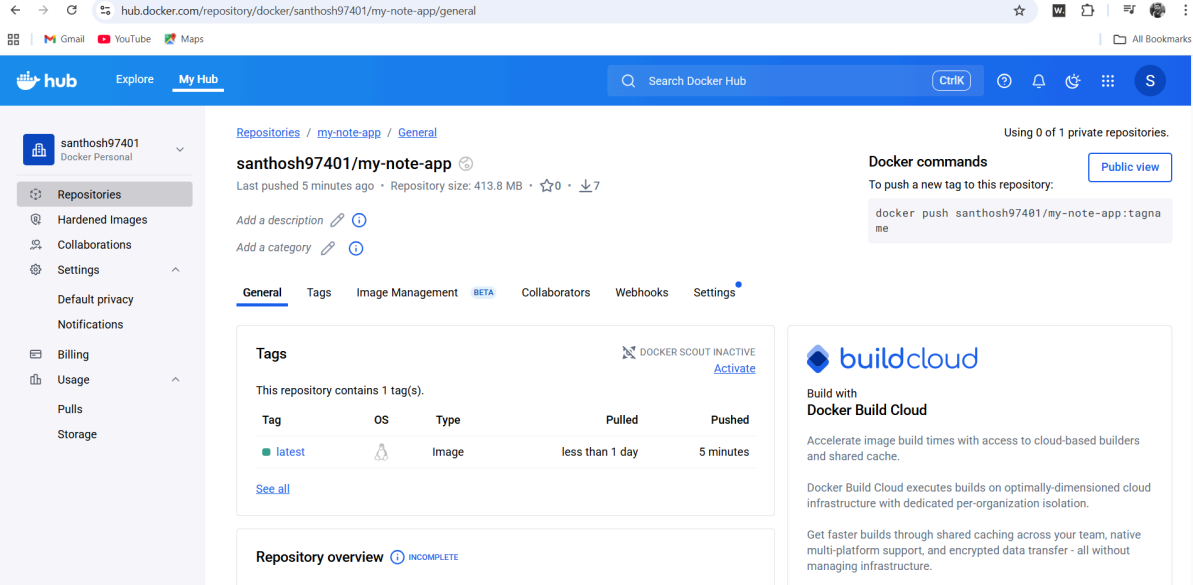
Permalinks

Last build (#11), 4 min 54 sec ago

Last stable build (#11), 4 min 54 sec ago

Last successful build (#11), 4 min 54 sec ago

7.Pushed docker image to dockerhub repository to deploy it .



The screenshot shows the Docker Hub interface for a repository named 'santhosh97401/my-note-app'. The page includes a sidebar with navigation options like Repositories, Hardened Images, Collaborations, Settings, Billing, Usage, Pulls, and Storage. The main content area displays the repository details, including the tag 'latest' and a table of tags. A 'Docker commands' section provides instructions on how to push a new tag to the repository. The page also features a 'buildcloud' advertisement and a 'Repository overview' section.

Using 0 of 1 private repositories.

Docker commands [Public view](#)

To push a new tag to this repository:

```
docker push santhosh97401/my-note-app:tagname
```

Tags [DOCKERS SCOUT INACTIVE](#) [Activate](#)

This repository contains 1 tag(s).

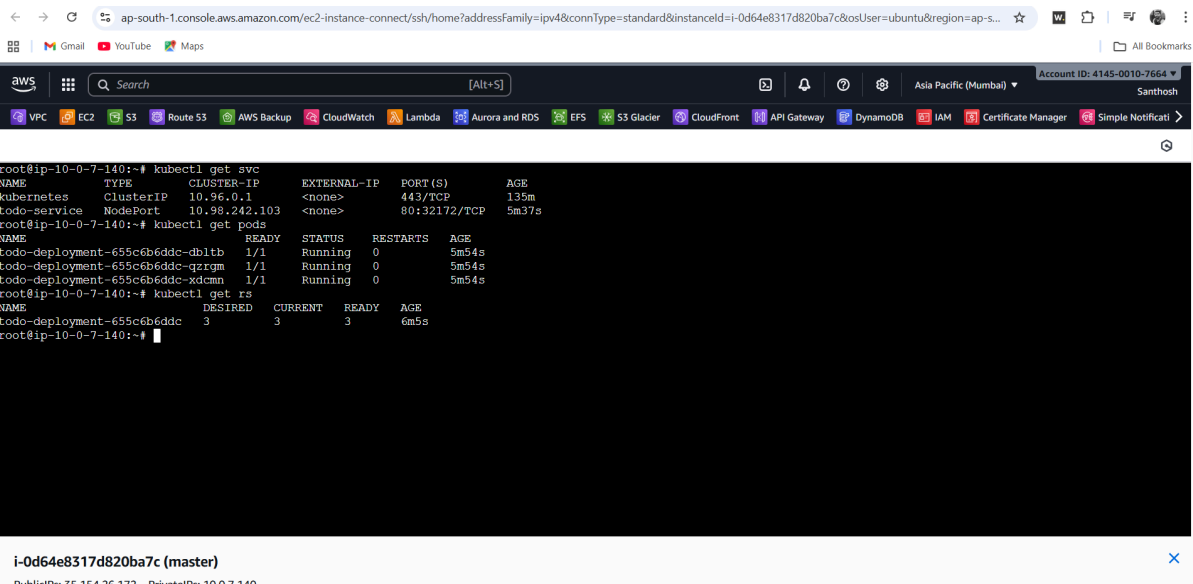
Tag	OS	Type	Pulled	Pushed
latest	linux	Image	less than 1 day	5 minutes

[See all](#)

Repository overview [INCOMPLETE](#)

An overview describes what your image does and how to run it. It displays in [the public view of your](#)

8.Created kubernetes cluster with deployment controller and service with nodeport.



The screenshot shows the AWS CloudShell terminal interface. The terminal displays the output of several Kubernetes commands executed on an EC2 instance. The commands and their outputs are as follows:

```
root@ip-10-0-7-140:~# kubectl get svc
NAME                TYPE          CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
kubernetes           ClusterIP     10.96.0.1        <none>            443/TCP          135m
todo-service         NodePort      10.98.242.103    <none>            80:32172/TCP     5m37s

root@ip-10-0-7-140:~# kubectl get pods
NAME                READY          STATUS    RESTARTS   AGE
todo-deployment-655c6b6ddc-dbltb  1/1            Running   0           5m54s
todo-deployment-655c6b6ddc-qzrgm  1/1            Running   0           5m54s
todo-deployment-655c6b6ddc-xdcmn  1/1            Running   0           5m54s

root@ip-10-0-7-140:~# kubectl get rs
NAME                DESIRED        CURRENT        READY        AGE
todo-deployment-655c6b6ddc  3              3              3            6m5s

root@ip-10-0-7-140:~#
```

The terminal output shows that the Kubernetes cluster is running successfully. The 'todo-service' is exposed on port 80, and the 'todo-deployment' is running with 3 replicas. The 'todo-service' is a NodePort service, and the 'todo-deployment' is a Deployment.

i-0d64e8317d820ba7c (master)

PublicIPs: 35.154.26.172 PrivateIPs: 10.0.7.140

CloudShell Feedback Console Mobile App

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

9.The application is live with passing the stages of pipelines.

