

# Devops Project

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

## 1. Github repository

The screenshot shows a GitHub repository page for 'django-notes-app'. The repository is public and was forked from 'swathis10/django-notes-app'. The commit history shows several commits from 'swathis10' over the past two days, including changes to deployment stages and files like Dockerfile and README.md. The repository has 0 forks and 0 stars.

## 2. Launched 2 ubuntu EC2 virtual machines.

The screenshot shows the AWS EC2 Instances page. It displays two running instances: 'master' (instance ID i-0d6e8317d820ba7c) and 'slave' (instance ID i-0d6a72b8b3a6bc35). Both instances are of type t2.large and are running in the 'ap-south-1a' availability zone. They have Public IP addresses ec2-35-1! and ec2-35-1!. The status check for both instances is 2/2 checks passed.

Name	Instance ID	Instance state	Instance type	Status check	Availability Zone	Public IP
master	i-0d6e8317d820ba7c	Running	t2.large	2/2 checks passed	ap-south-1a	ec2-35-1!
slave	i-0d6a72b8b3a6bc35	Running	t2.large	2/2 checks passed	ap-south-1a	ec2-35-1!

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

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

i-0d64e8317d820ba7c (master)  
Public IPs: 35.154.26.172 Private IPs: 10.0.7.140  
CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

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

The screenshot shows the Jenkins Pipeline configuration page for a job named "demo-project". The left sidebar has tabs for General, Triggers, Pipeline (which is selected), and Advanced. The main area displays a Groovy script for the pipeline stages:

```
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(credentialsId:"Dockerhub",passwordVariable:"dockerHubPass",usernameVariable:"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~    }
}
```

At the bottom are "Save" and "Apply" buttons.

## 5.credentials for accessing docker hub and kubernetes.

The screenshot shows the Jenkins Global credentials page at [http://35.154.26.172:8080/manage/credentials/store/system/domain/\\_/](http://35.154.26.172:8080/manage/credentials/store/system/domain/_/). It lists two credentials:

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

Icons for S, M, and L are visible at the bottom left. At the bottom right, it says REST API Jenkins 2.528.1.

## 6. Installed stage view plugin and executed all the stages

The screenshot shows the Jenkins demo-project pipeline at <http://35.154.26.172:8080/job/demo-project/>. The pipeline has four stages: Clone Code, Build, Pushing to Docker Hub, and Deploy to kubernetes. Stage #11 failed with a duration of 160ms. Stage #10 succeeded with a duration of 19s. Stage #9 succeeded with a duration of 2s. Stage #8 succeeded with a duration of 36s. The pipeline summary shows average stage times: 703ms, 10s, 11s, and 2s respectively.

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

The screenshot shows the Docker Hub interface for the repository 'santhosh97401/my-note-app'. The 'General' tab is selected. The 'Tags' section displays one tag, 'latest', which was pushed 5 minutes ago. The Docker commands section includes a 'Public view' button. A sidebar on the left provides navigation links for repositories, hardened images, collaborations, settings, billing, usage, pulls, and storage.

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

The screenshot shows the AWS CloudShell terminal output. The user runs 'kubectl get svc' to list services, showing a ClusterIP service for 'kubernetes' and a NodePort service for 'todo-service'. Then, 'kubectl get pods' is run, showing three pods: 'todo-deployment-655c6b6ddc-db1tb', 'todo-deployment-655c6b6ddc-qzrgm', and 'todo-deployment-655c6b6ddc-xdcmn', all in a 'Running' state. Finally, 'kubectl get rs' is run, showing a RollingStatus for the 'todo-deployment' with 3 desired and 3 current replicas.

```
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-db1tb  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:~#
```

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

