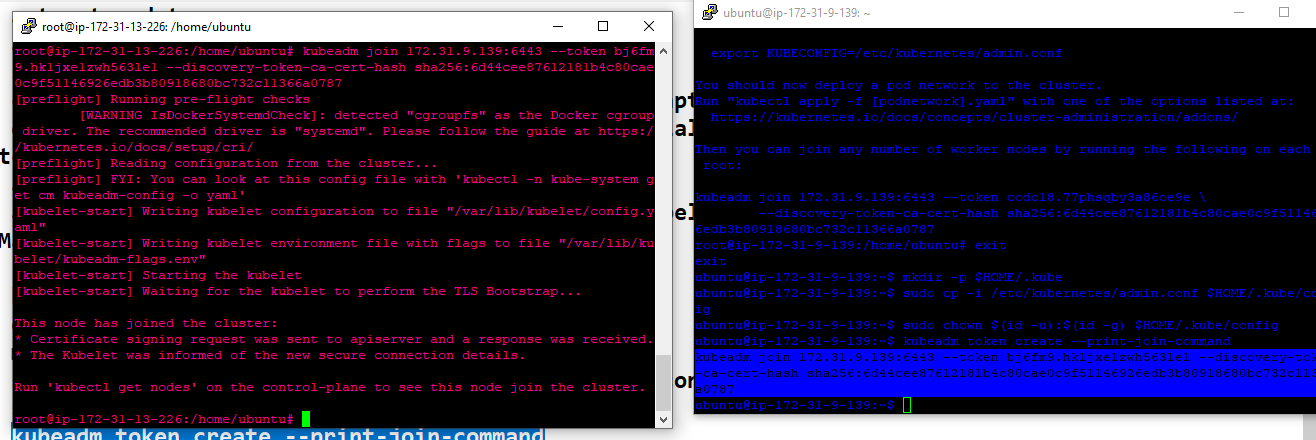
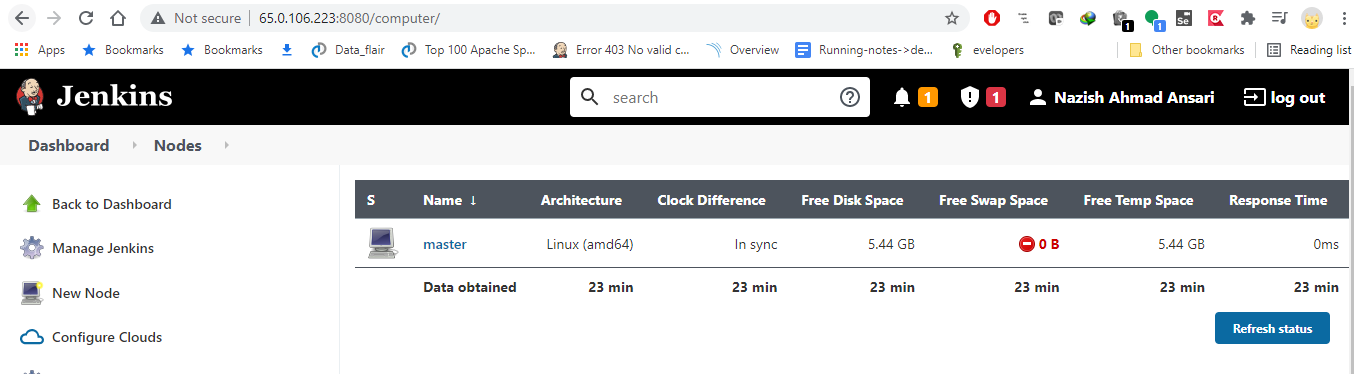
Server-1:- ORANGE ( MICRO)

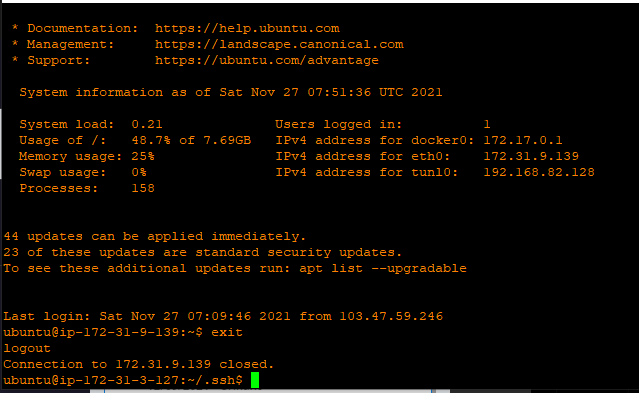
Server-2:- GREEN ( MICRO)

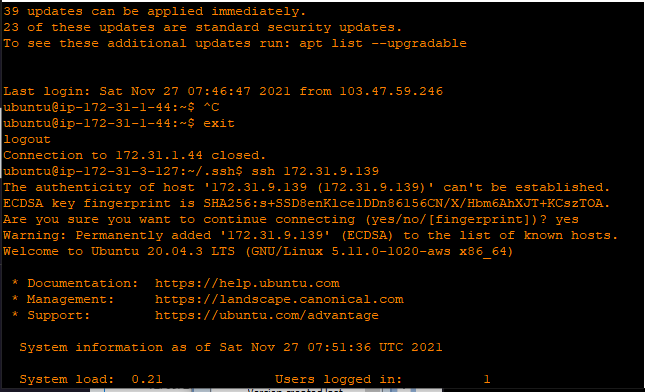
Server-3:- BLUE (MEDIUM)

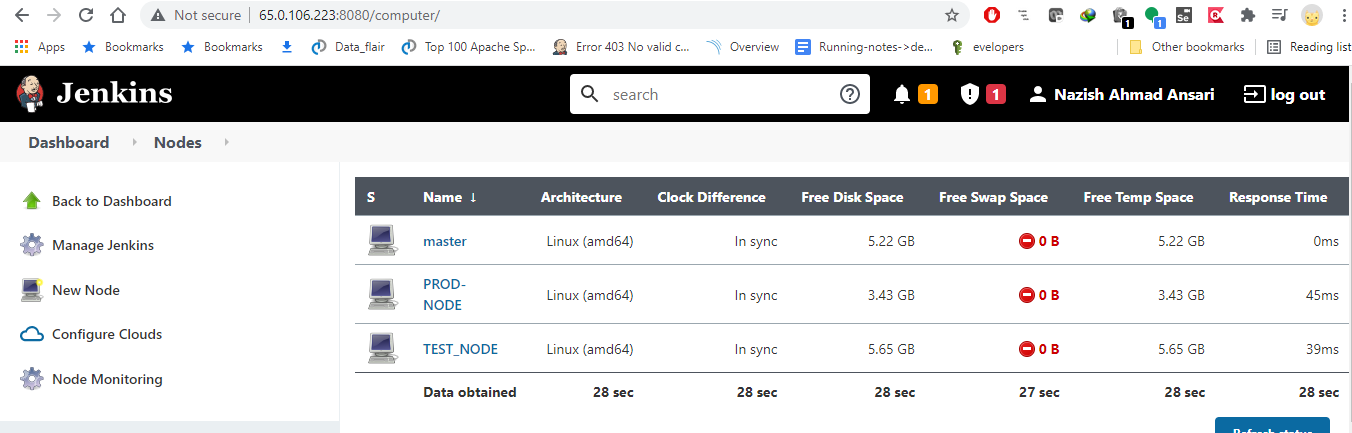
Server-4:- PINK (MEDIUM)

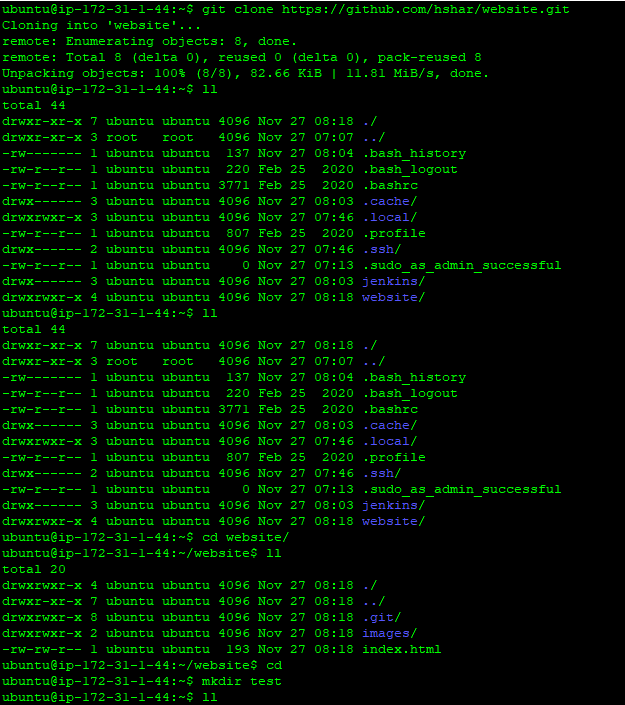


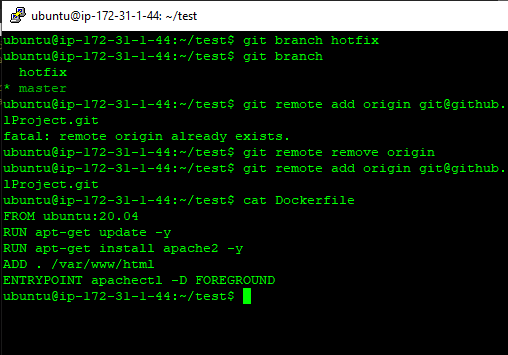


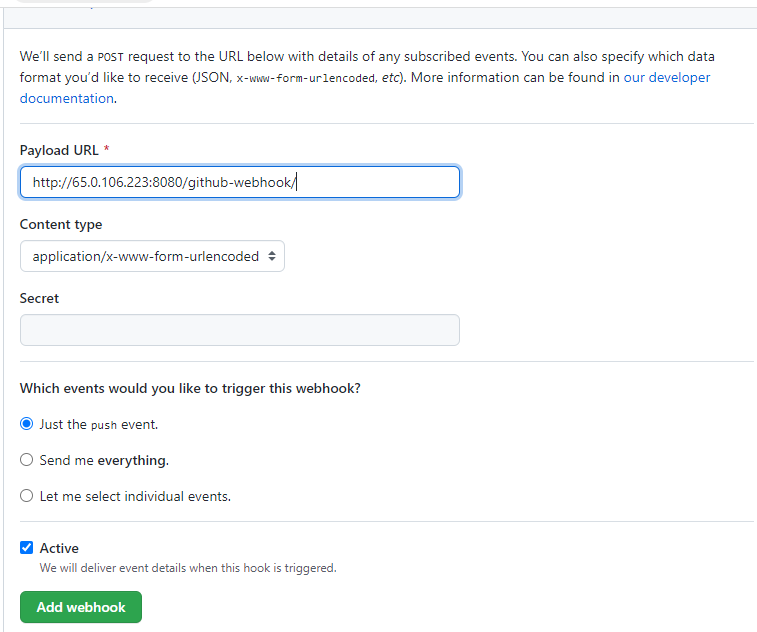


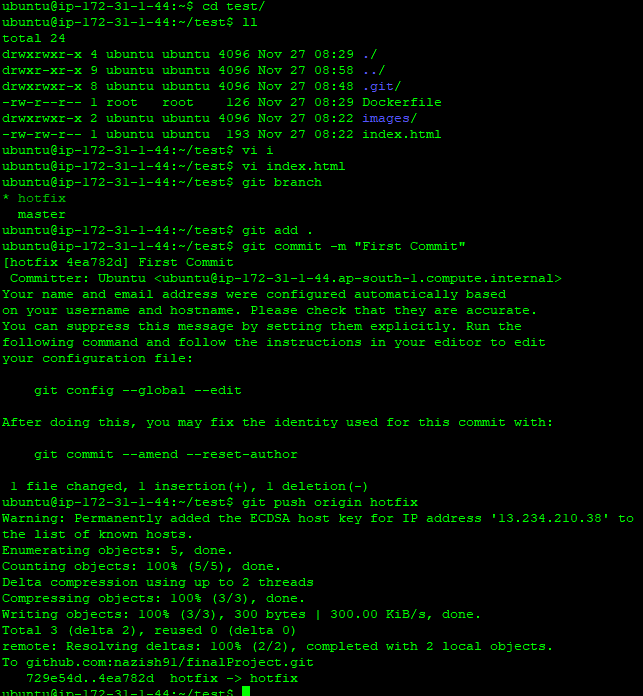


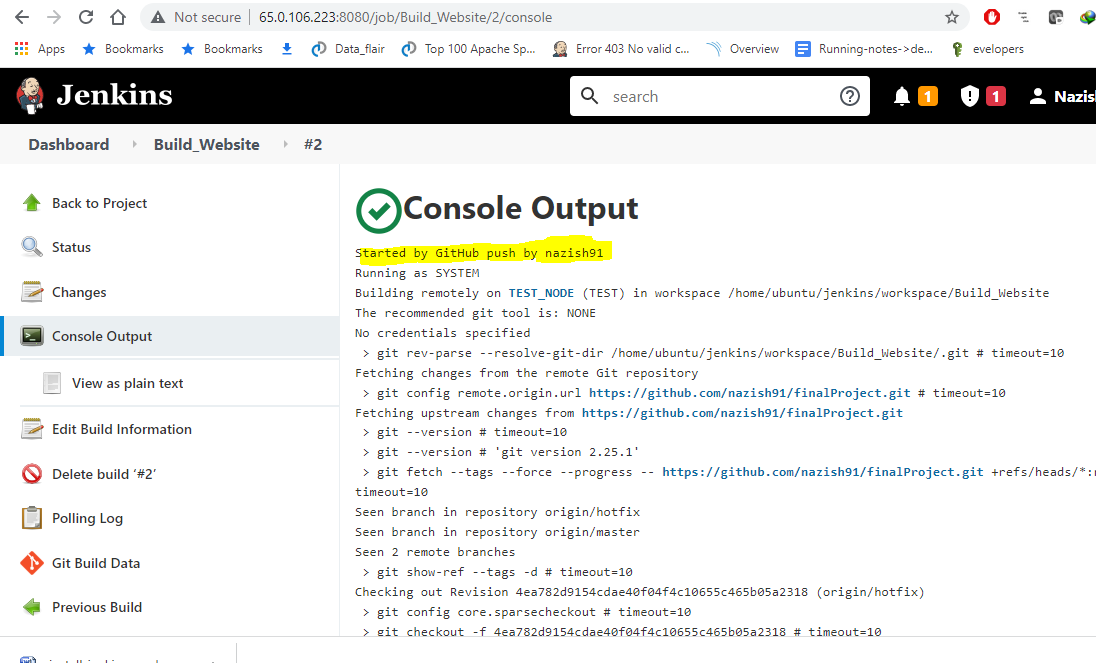


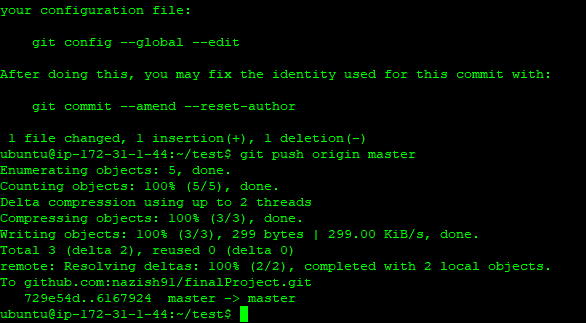




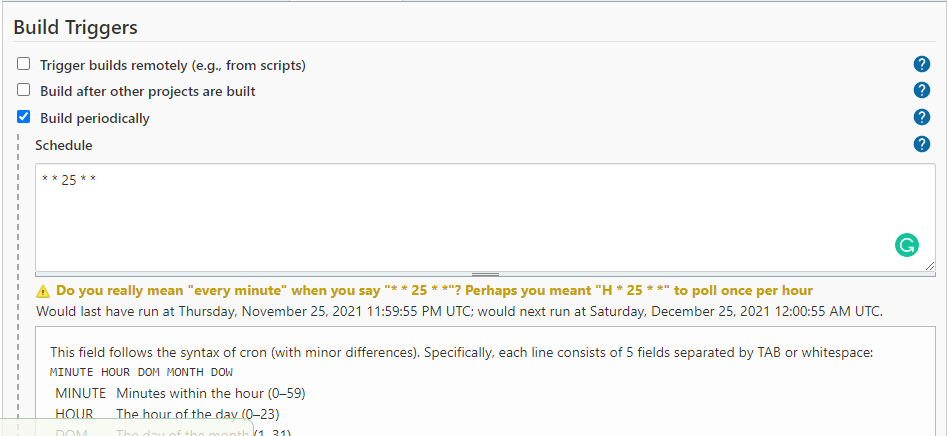


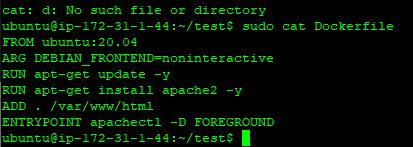












Started by user [Nazish Ahmad Ansari](http://65.0.106.223:8080/user/nazish)

[Pipeline] Start of Pipeline

[Pipeline] node

Running on [TEST\_NODE](http://65.0.106.223:8080/computer/TEST_NODE/) in /home/ubuntu/jenkins/workspace/Push\_to\_Docer\_Hub

[Pipeline] {

[Pipeline] withCredentials

Masking supported pattern matches of $DOCKERHUB\_CREDENTIALS or $DOCKERHUB\_CREDENTIALS\_PSW

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Login)

[Pipeline] sh

+ sudo docker login -u nazish91 --password-stdin

+ echo \*\*\*\*

WARNING! Your password will be stored unencrypted in /root/.docker/config.json.

Configure a credential helper to remove this warning. See

<https://docs.docker.com/engine/reference/commandline/login/#credentials-store>

Login Succeeded

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Push)

[Pipeline] sh

+ sudo docker push nazish91/app1:latest

The push refers to repository [docker.io/nazish91/app1]

43eccb4ac559: Preparing

7965d181def1: Preparing

713d8641e25e: Preparing

9f54eef41275: Preparing

9f54eef41275: Mounted from library/ubuntu

43eccb4ac559: Pushed

713d8641e25e: Pushed

7965d181def1: Pushed

latest: digest: sha256:2629fb807a406a834aa44bdf49180a568797ca313d7ab49b327fea4c28c54ab0 size: 1163

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Logout)

[Pipeline] sh

+ sudo docker logout

Removing login credentials for <https://index.docker.io/v1/>

[Pipeline] }

[Pipeline] // stage

[Pipeline] }

[Pipeline] // withCredentials

[Pipeline] }

[Pipeline] // node

[Pipeline] End of Pipeline

Finished: SUCCESS

Groovy Script to push Docker image to Docker HUb

pipeline{

agent {label 'TEST\_NODE'}

environment {

DOCKERHUB\_CREDENTIALS=credentials('docker')

}

stages {

stage('Login') {

steps {

sh 'echo $DOCKERHUB\_CREDENTIALS\_PSW | sudo docker login -u $DOCKERHUB\_CREDENTIALS\_USR --password-stdin'

}

}

stage('Push') {

steps {

sh 'sudo docker push nazish91/app1:latest'

}

}

stage('Logout') {

steps {

sh 'sudo docker logout'

}

}

}

}

Kubernetes Deploymnet Using Jenkins :--

Started by user [Nazish Ahmad Ansari](http://65.0.106.223:8080/user/nazish)

Running as SYSTEM

Building remotely on [PROD-NODE](http://65.0.106.223:8080/computer/PROD-NODE) (PROD) in workspace /home/ubuntu/jenkins/workspace/Production\_Deployment

The recommended git tool is: NONE

using credential Jenkins

> git rev-parse --resolve-git-dir /home/ubuntu/jenkins/workspace/Production\_Deployment/.git # timeout=10

Fetching changes from the remote Git repository

> git config remote.origin.url <https://github.com/nazish91/finalProject.git> # timeout=10

Fetching upstream changes from <https://github.com/nazish91/finalProject.git>

> git --version # timeout=10

> git --version # 'git version 2.25.1'

using GIT\_SSH to set credentials Jenkins

> git fetch --tags --force --progress -- <https://github.com/nazish91/finalProject.git> +refs/heads/\*:refs/remotes/origin/\* # timeout=10

> git rev-parse refs/remotes/origin/master^{commit} # timeout=10

Checking out Revision 398ded94f25be0a17b5d5bdceaafe27d85e8e1e2 (refs/remotes/origin/master)

> git config core.sparsecheckout # timeout=10

> git checkout -f 398ded94f25be0a17b5d5bdceaafe27d85e8e1e2 # timeout=10

Commit message: "second commit"

> git rev-list --no-walk 398ded94f25be0a17b5d5bdceaafe27d85e8e1e2 # timeout=10

[Production\_Deployment] $ /bin/sh -xe /tmp/jenkins7412006678267155205.sh

+ kubectl create -f apache.yaml

deployment.apps/apache-deployment created

+ kubectl get po

NAME READY STATUS RESTARTS AGE

apache-deployment-57498cdcf8-cbqhk 0/1 ContainerCreating 0 0s

apache-deployment-57498cdcf8-dv2cm 0/1 ContainerCreating 0 0s

apache-deployment-57498cdcf8-xz649 0/1 ContainerCreating 0 0s

Finished: SUCCESS

Apache.yaml File :--

apiVersion: apps/v1

kind: Deployment

metadata:

name: apache-deployment

labels:

app: apache

spec:

replicas: 3

selector:

matchLabels:

app: apache

template:

metadata:

labels:

app: apache

spec:

containers:

- name: apache

image: nazish91/app1:latest

ports:

- containerPort: 80

