Course: DevOps Name: Billipati Sai Teja

Module: Tomcat Mail-ID: ([BILLIPATISAITEJA@GMAIL.COM](mailto:BILLIPATISAITEJA@GMAIL.COM))

Topic: Deploy in Tomcat Batch no: 115

Trainer Name: Mr. Madhukar sir Project No: 01

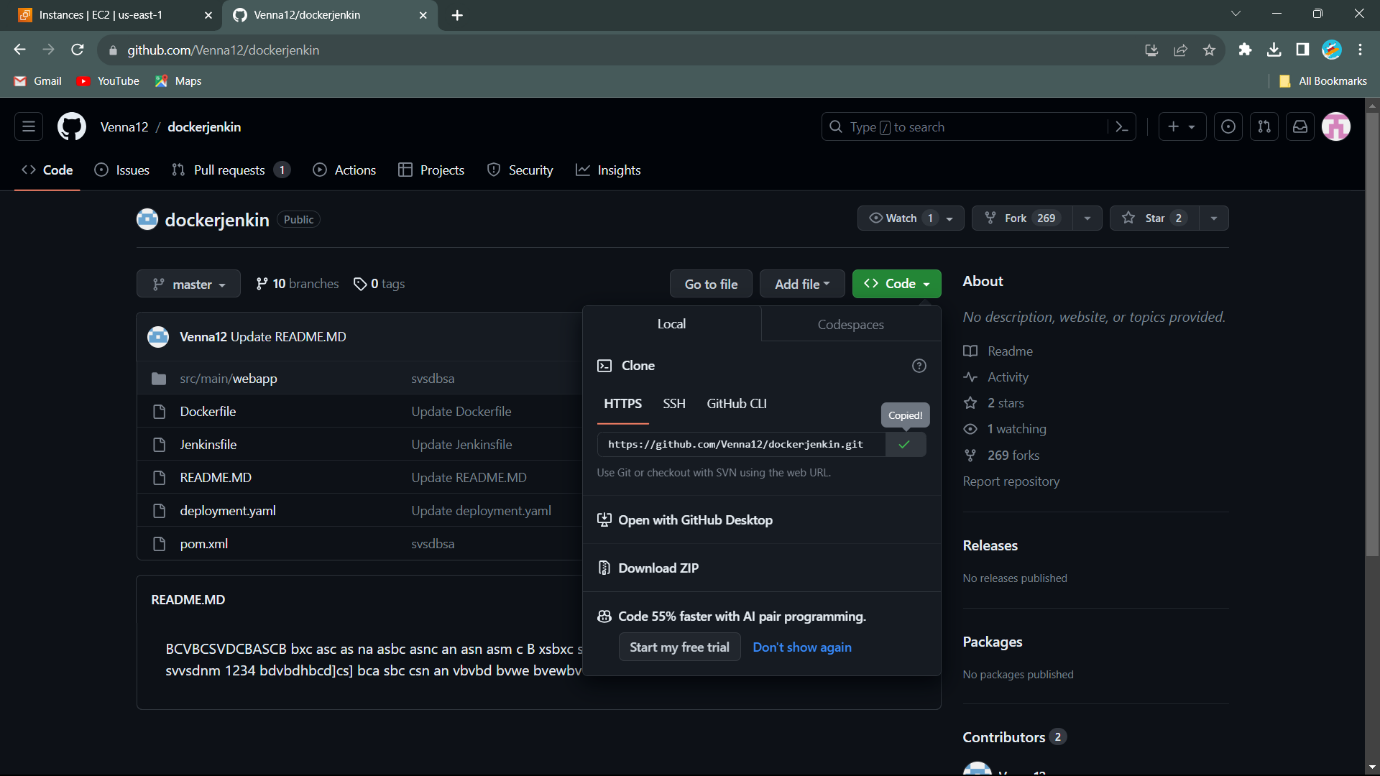
Date of submission: 29th – Nov – 2023

**Project Title:** Deploying an Application into Different Environments.

* User clone code from GitHub.
* Creating a pipeline in the Jenkins.
* User writes declarative pipeline.
* Creating AMI’s using single instance.
* Deploy application into different environments (DEV\_ENV, TEST\_ENV, PRE\_PROD\_ENV, PROD\_ENV).

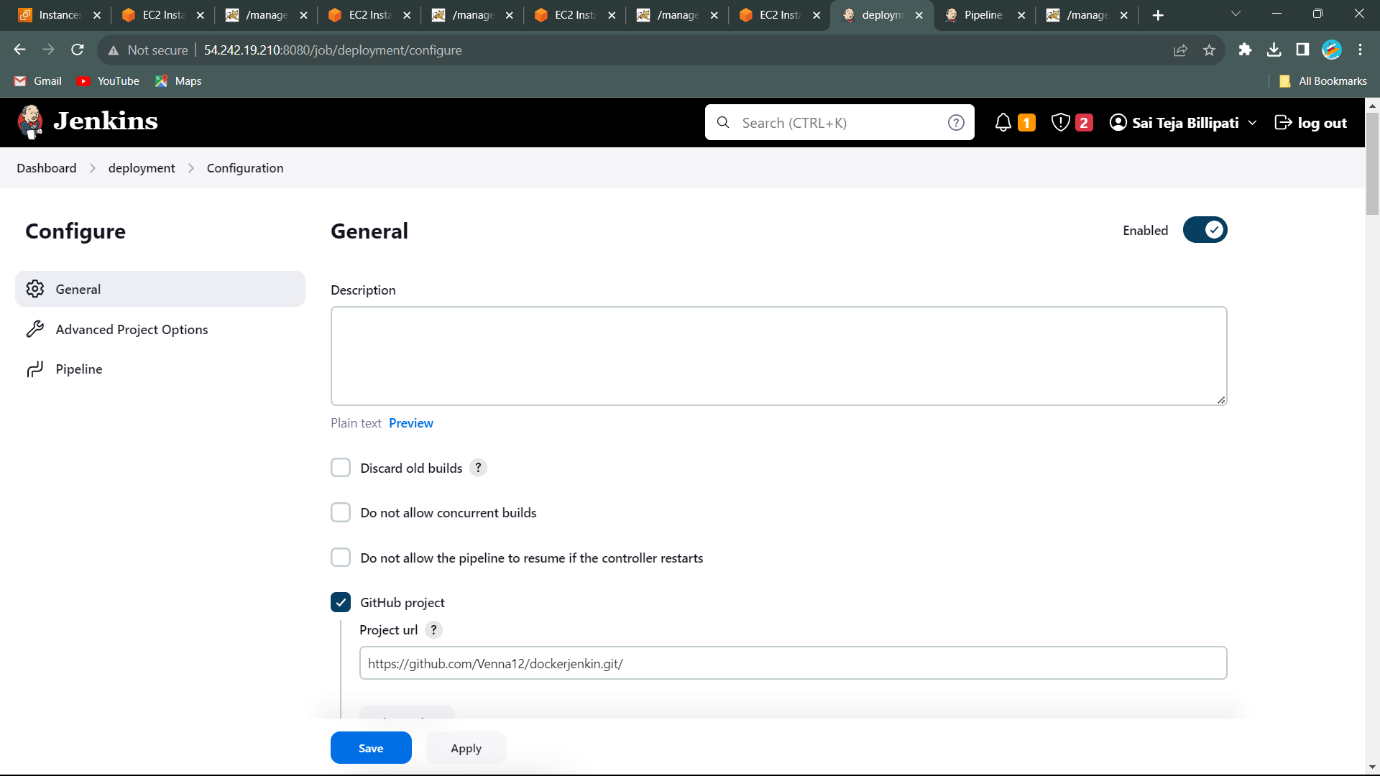
--------------------------------- **User clone code from GitHub** ---------------------------------

1. Open the AWS console and create the new instance for the project, connect the command line interface.
2. Change the user to the root user using (sudo -i).
3. Use the command to update the sever (apt update -y).
4. Install Java 11 using command (apt install default-jdk -y).
5. Install the maven using command (apt install maven).
6. Install the Jenkins sever using command line.
7. Install the Tomcat sever in the sever.
8. Clone the Application link from the GitHub.



------------------------------ **Creating a pipeline in the Jenkins** ------------------------------

* Open the Jenkins using the AWS Public IPV4 and using port no 8080.
* Create the Job using pipeline. Job name Dockerjenkins.
* We got configure dashboard, and click on the Git, add the clone link to given box.



**Project URL**: <https://github.com/Venna12/dockerjenkin.git>

------------------------ **Create the 3-AMI’s Using Single Instance** ------------------------

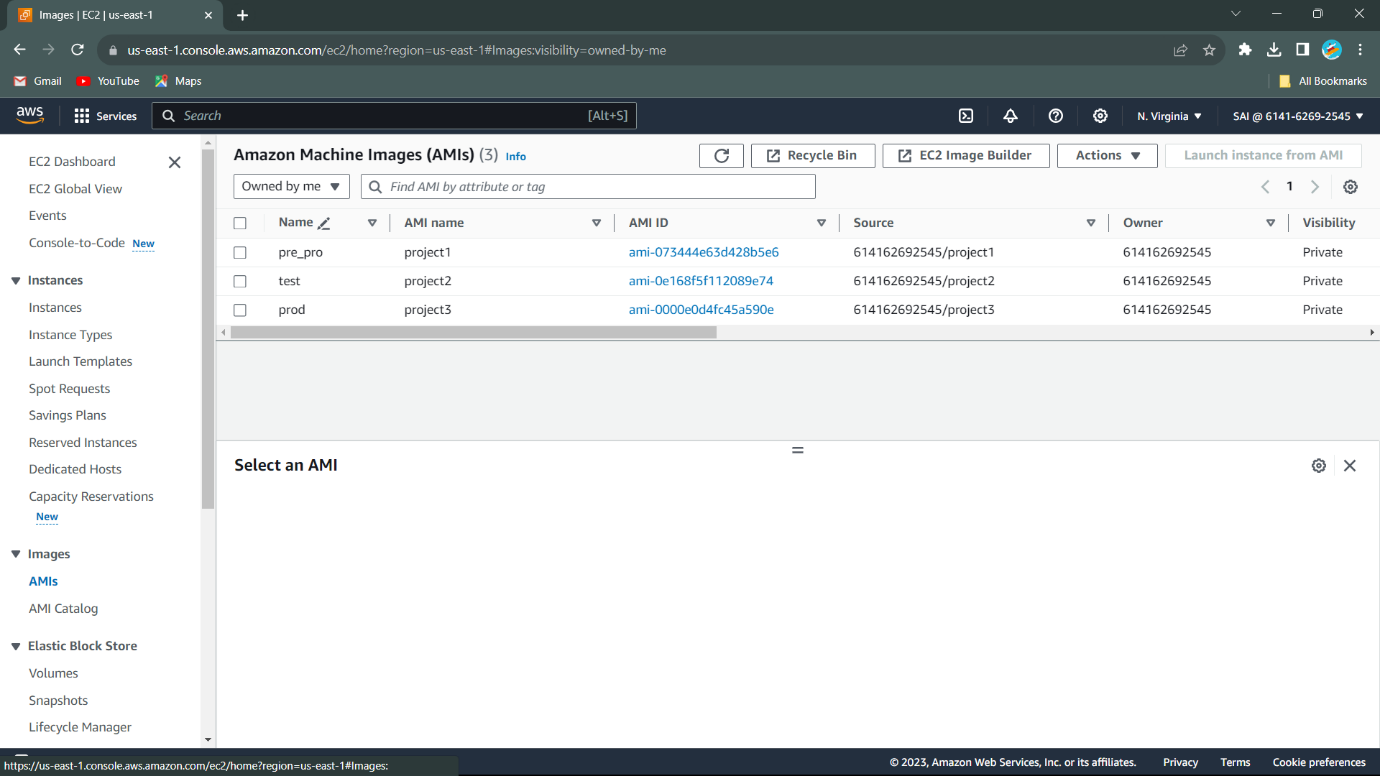
>>> Go to Main instance and go to Images and Templates.

>>> Click on the Create the Image, create the 3 Images.

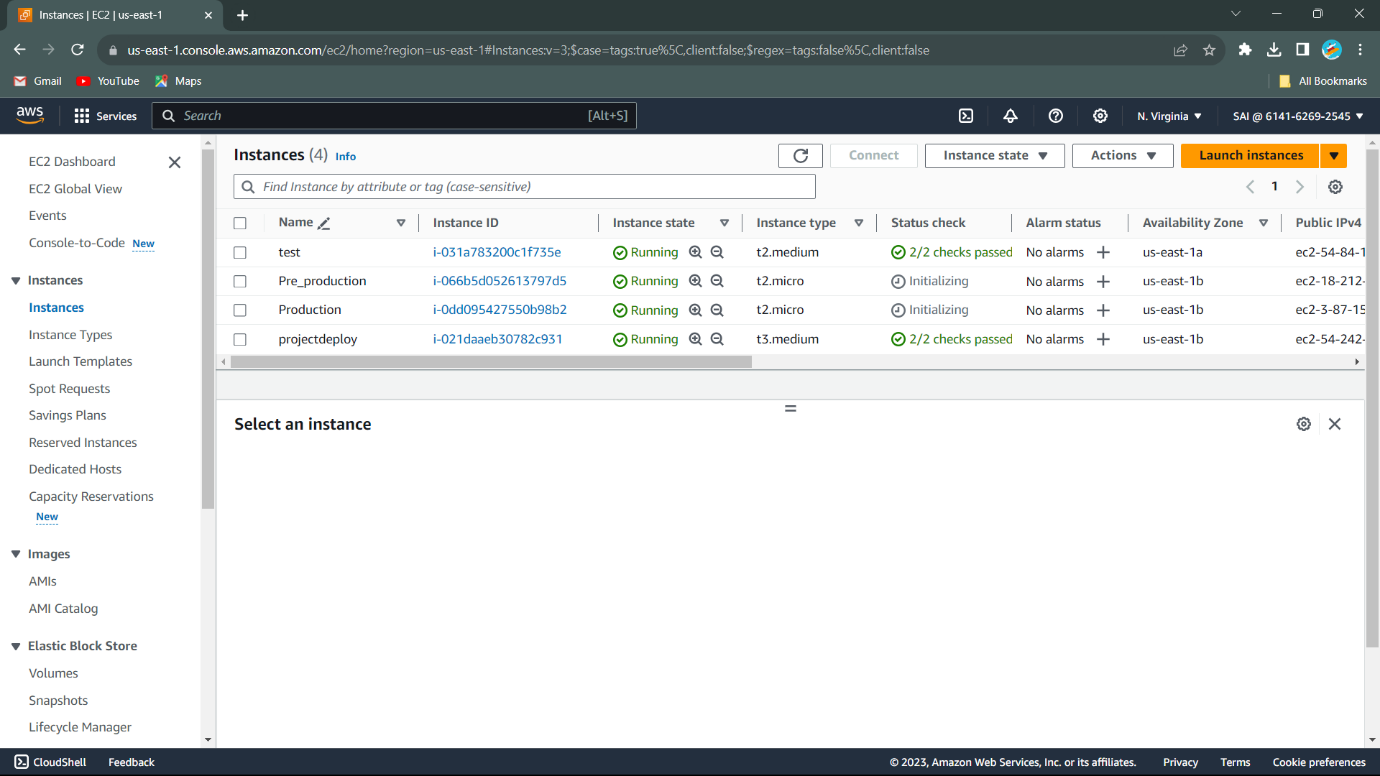
>>> Create AMI’s for the

1. TEST\_ENVIRONMENT
2. PRE\_PRODUCTION\_ENVIRONMENT
3. PRODUCTION\_ENVIRONMENT

* These are the Amazon Machine Images, related to our instances, these instances are completely images of the our main instances, so that in the all AMI’s, we get the data, software, severs same as in the main instances.
* But the change of these is IP address, we get different IP address for them.
* We should deploy the our application in these AMI’s using single pipeline.

>>> We can see the AMI’s in this figure….

* Start the AMI’s Instances. Then we get 4 running instances in the dashboard. In all instances, we have Jenkins, Java, Maven and Tomcat.



>>> Open Tomcat severs of all instances. Using their IPs with Port no.

* Go to Instances connect to the command interface.
* Go to user to root user (sudo -i) and go to Tomcat (cd tomcat).
* Go to bin and start the Tomcat using command (./startup.sh).
* Copy the IPV4 of instances and paste on google search and add port no to it.

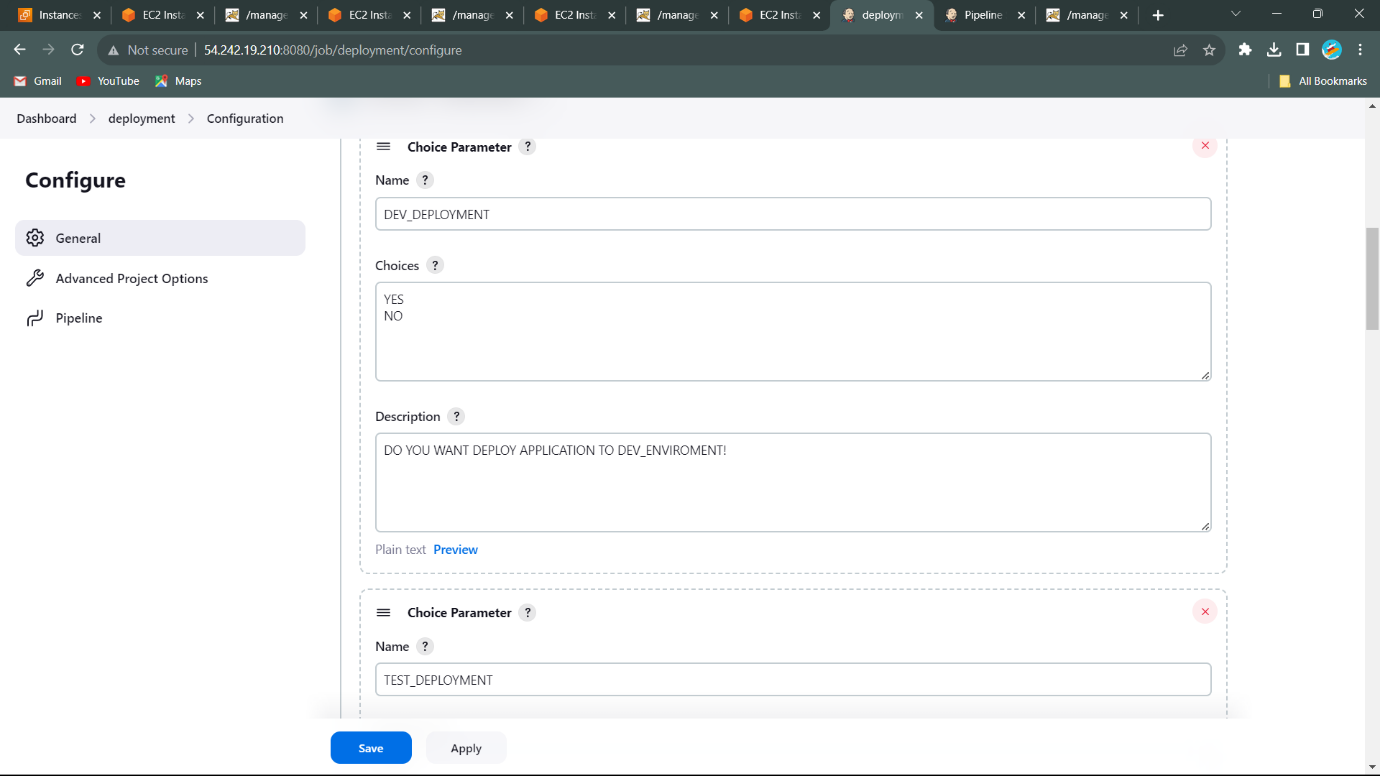
------------------------------------ **Choice Parameter** ------------------------------------

* Go to pipeline Configuration and click on **Add parameter.**

1. We get interface of the Choice Parameter.
   1. Name (Parameter name).
   2. Choices (Value either Yes or No).
   3. Description (Descripted about user need).

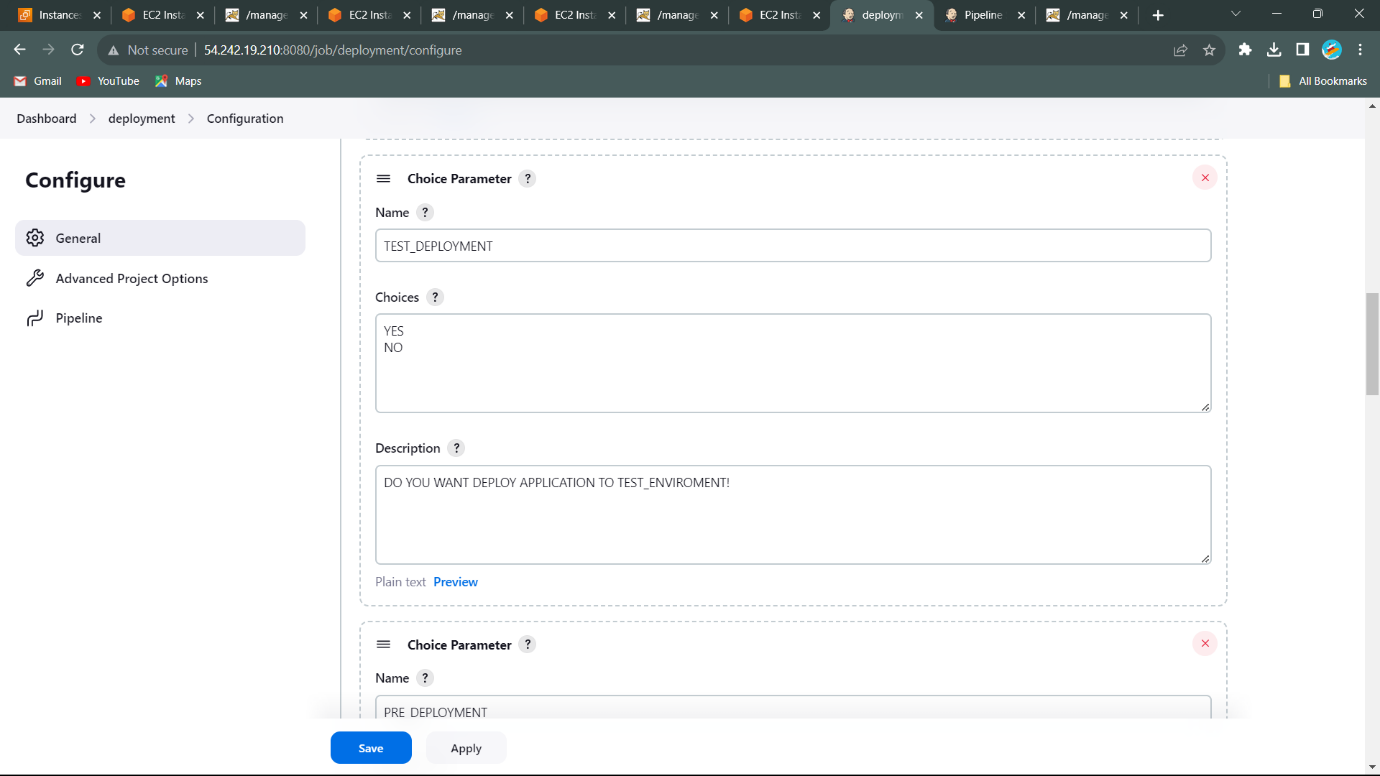
* First, we enter the Parameters for the DEV\_ENVIRONMENT. As shown in the figure…

1. Name: DEV\_DEPLOYMENT
2. Choices: YES or NO
3. Description: DO YOU WANT DEPLOY APPLICATION TO DEV\_ENVIRONMENT.



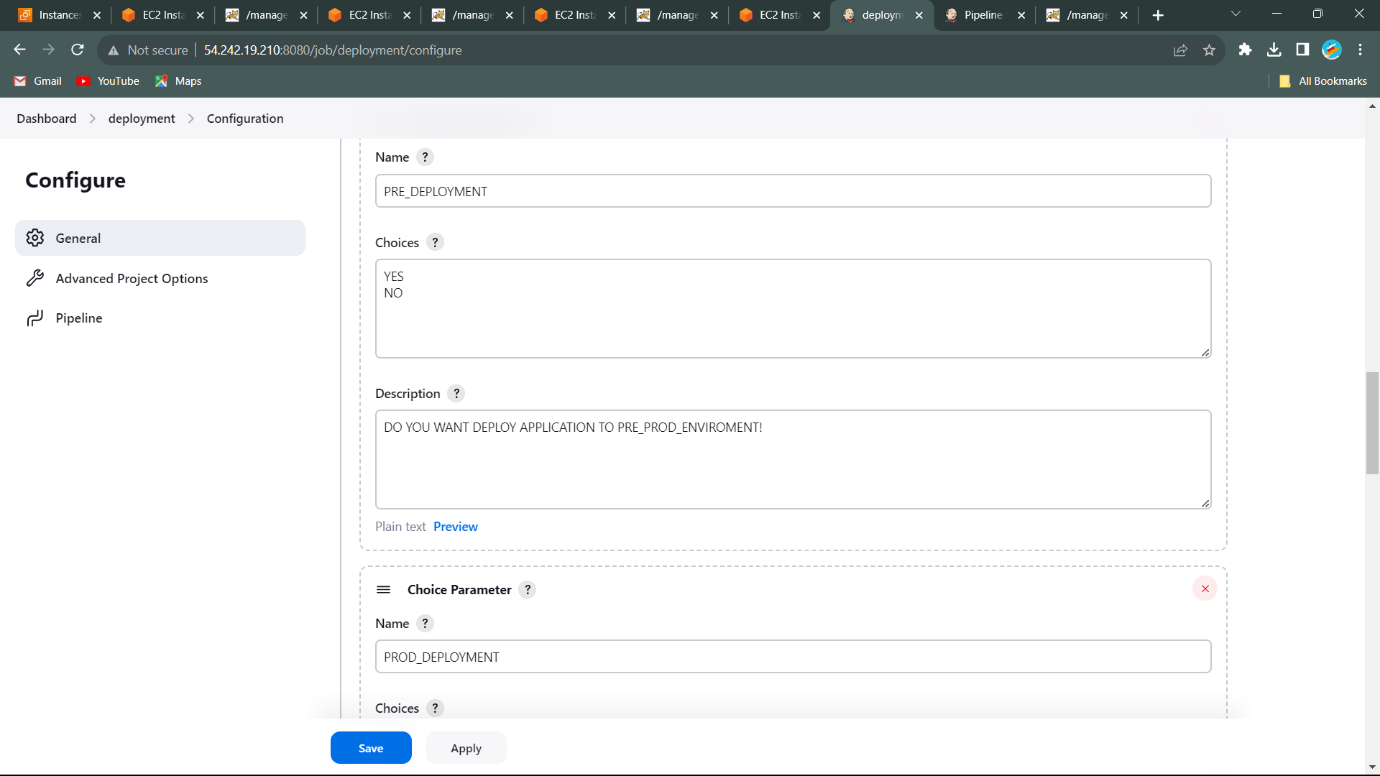
* Second, we enter the Parameters for the DEV\_ENVIRONMENT. As shown in the figure…

1. Name: TEST\_DEPLOYMENT
2. Choices: YES or NO
3. Description: DO YOU WANT DEPLOY APPLICATION TO TEST\_ENVIRONMENT.



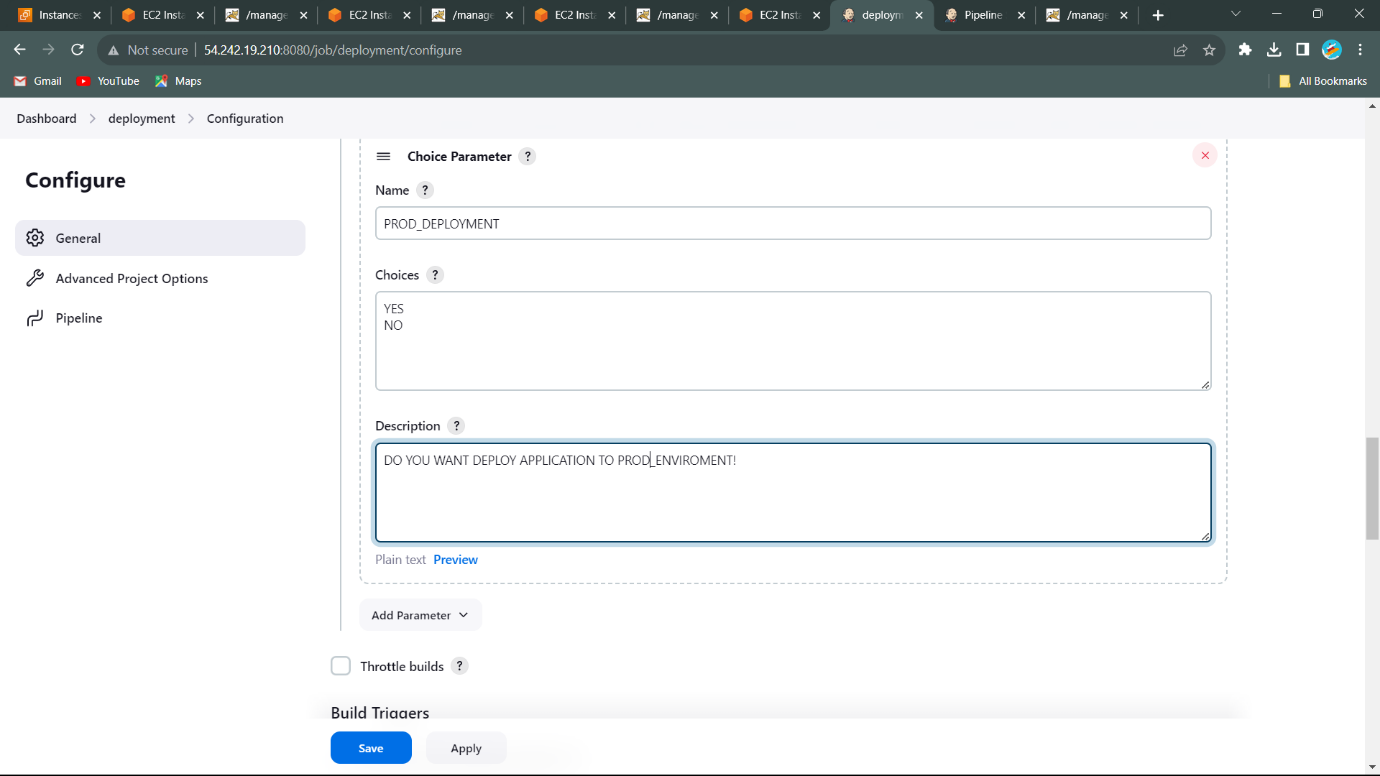
* Second, we enter the Parameters for the DEV\_ENVIRONMENT. As shown in the figure…

1. Name: PRE\_DEPLOYMENT
2. Choices: YES or NO
3. Description: DO YOU WANT DEPLOY APPLICATION TO PRE\_PROD\_ENVIRONMENT.



* Second, we enter the Parameters for the DEV\_ENVIRONMENT. As shown in the figure…

1. Name: PROD\_DEPLOYMENT
2. Choices: YES or NO
3. Description: DO YOU WANT DEPLOY APPLICATION TO PROD\_ENVIRONMENT.



>>>> Go to **Pipeline Script**

* Write the Pipeline Script for the **Choice Parameters**.
* In the pipeline script add the script for the choice parameter is

parameters {

choice (

choices: ['YES', 'NO'],

description: ‘ ’,

name: ‘ ’

)

* Add the another script to connect parameter in the Tomcat deploy code.

when{

expression{

params.Name =='YES'

}

}

>>> By using these lines, we should write the script for the pipeline to Deploy in the Different Environments using Choice Parameters.

**----------------------------- Pipeline Script to Deploy Application in Different Environments using Choice Parameters -----------------------------**

Script: pipeline{

agent any

parameters {

choice (

choices: ['YES', 'NO'],

description: 'DO YOU WANT DEPLOY APPLICATION TO DEV\_ENVIROMENT!',

name: 'DEV\_DEPLOYMENT'

)

choice (

choices: ['YES', 'NO'],

description: 'DO YOU WANT DEPLOY APPLICATION TO TEST\_ENVIROMENT!',

name: 'TEST\_DEPLOYMENT'

)

choice (

choices: ['YES', 'NO'],

description: 'DO YOU WANT DEPLOY APPLICATION TO PRE\_PROD\_ENVIROMENT!',

name: 'PRE\_DEPLOYMENT'

)

choice (

choices: ['YES', 'NO'],

description: 'DO YOU WANT DEPLOY APPLICATION TO PROD\_ENVIROMENT!',

name: 'PROD\_DEPLOYMENT'

)

}

stages{

stage('clone'){

steps{

checkout scmGit(branches: [[name: '\*/master']], extensions: [], userRemoteConfigs: [[url: 'https://github.com/Venna12/dockerjenkin.git']])

}

}

stage('package'){

steps{

sh 'mvn package'

}

}

stage('DEV\_DEPLOYMENT'){

when{

expression{

params.DEV\_DEPLOYMENT =='YES'

}

}

steps{

deploy adapters: [tomcat9(credentialsId: '048f3b84-5d84-4417-9d0f-2b0a1a2efc35', path: '', url: 'http://54.242.19.210:4040/')], contextPath: 'DEV\_App', war: '\*\*/\*.war'

}

}

stage('TEST\_ENVIROMENT'){

when{

expression{

params.TEST\_DEPLOYMENT =='YES'

}

}

steps{

deploy adapters: [tomcat9(credentialsId: '048f3b84-5d84-4417-9d0f-2b0a1a2efc35', path: '', url: 'http://54.84.101.231:4040/')], contextPath: 'TEST\_Deploy', war: '\*\*/\*.war'

}

}

stage('PRE\_PRODUCTION'){

when{

expression{

params.PRE\_DEPLOYMENT =='YES'

}

}

steps{

deploy adapters: [tomcat9(credentialsId: '048f3b84-5d84-4417-9d0f-2b0a1a2efc35', path: '', url: 'http://18.212.218.229:4040/')], contextPath: 'PRE\_PRODUCTION', war: '\*\*/\*.war'

}

}

stage('PRODUCTION\_ENIVROMENT'){

when{

expression{

params.PROD\_DEPLOYMENT =='YES'

}

}

steps{

deploy adapters: [tomcat9(credentialsId: '048f3b84-5d84-4417-9d0f-2b0a1a2efc35', path: '', url: 'http://3.87.158.160:4040/')], contextPath: 'PRODUCTION', war: '\*\*/\*.war'

}

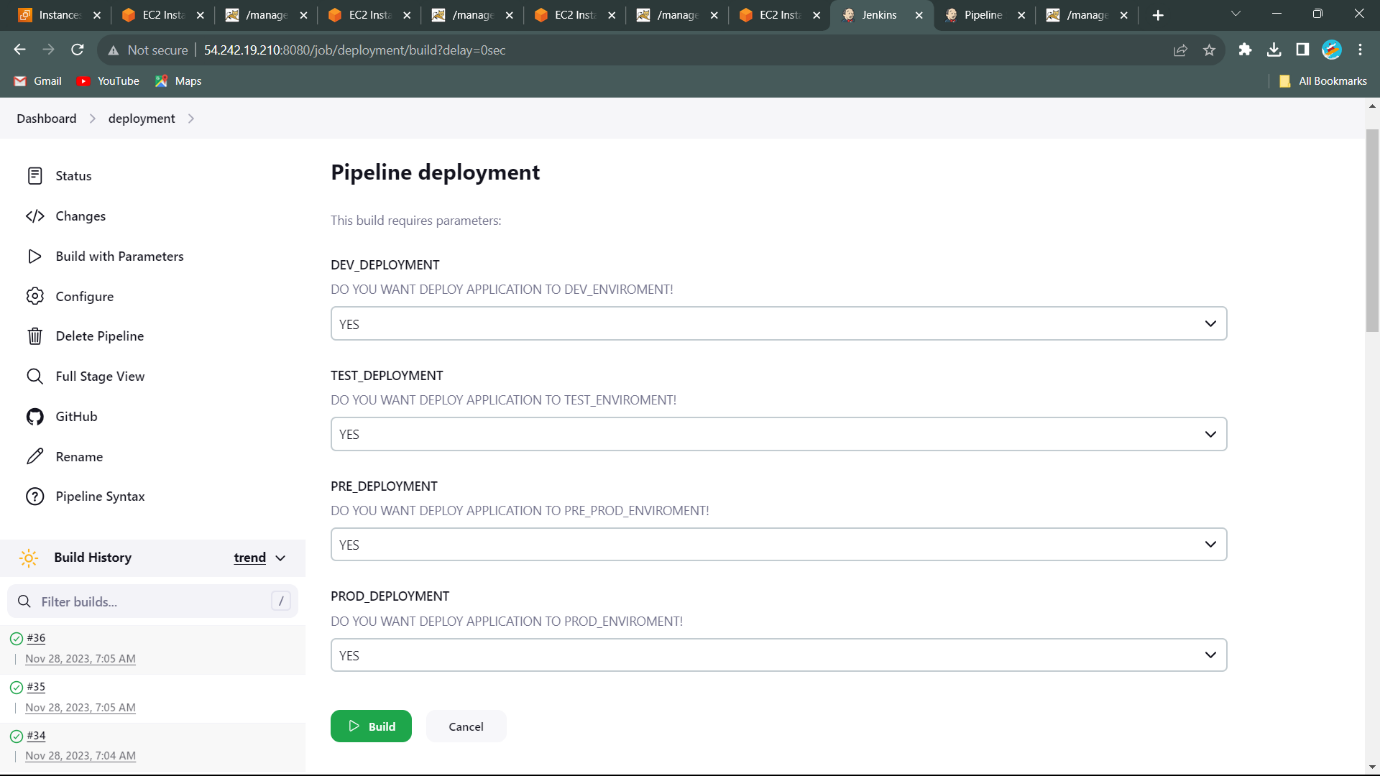
}

}

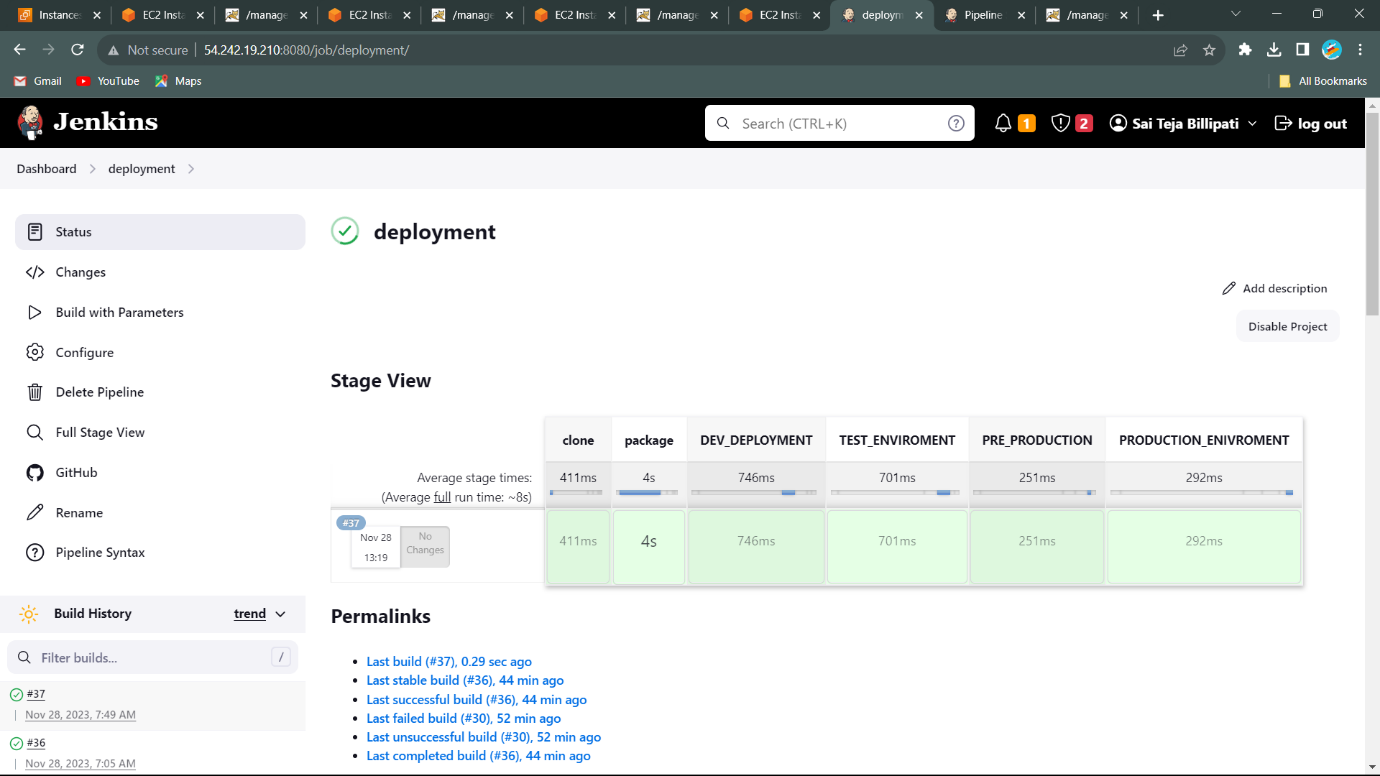
}

**Build the parameters………………………….**

* It shows the interface as shown in the figure…………..

Display DEV\_DEPLOYMENT, TEST\_DEPLOYMENT, PRE\_DEPLOYMENT, PROD\_DEPLOYMENT. With the Choice Parameters (YES OR NO).

* Application is Successfully Build and Deployed in Different Environments.
* Output displayed in the Build pipeline format……..

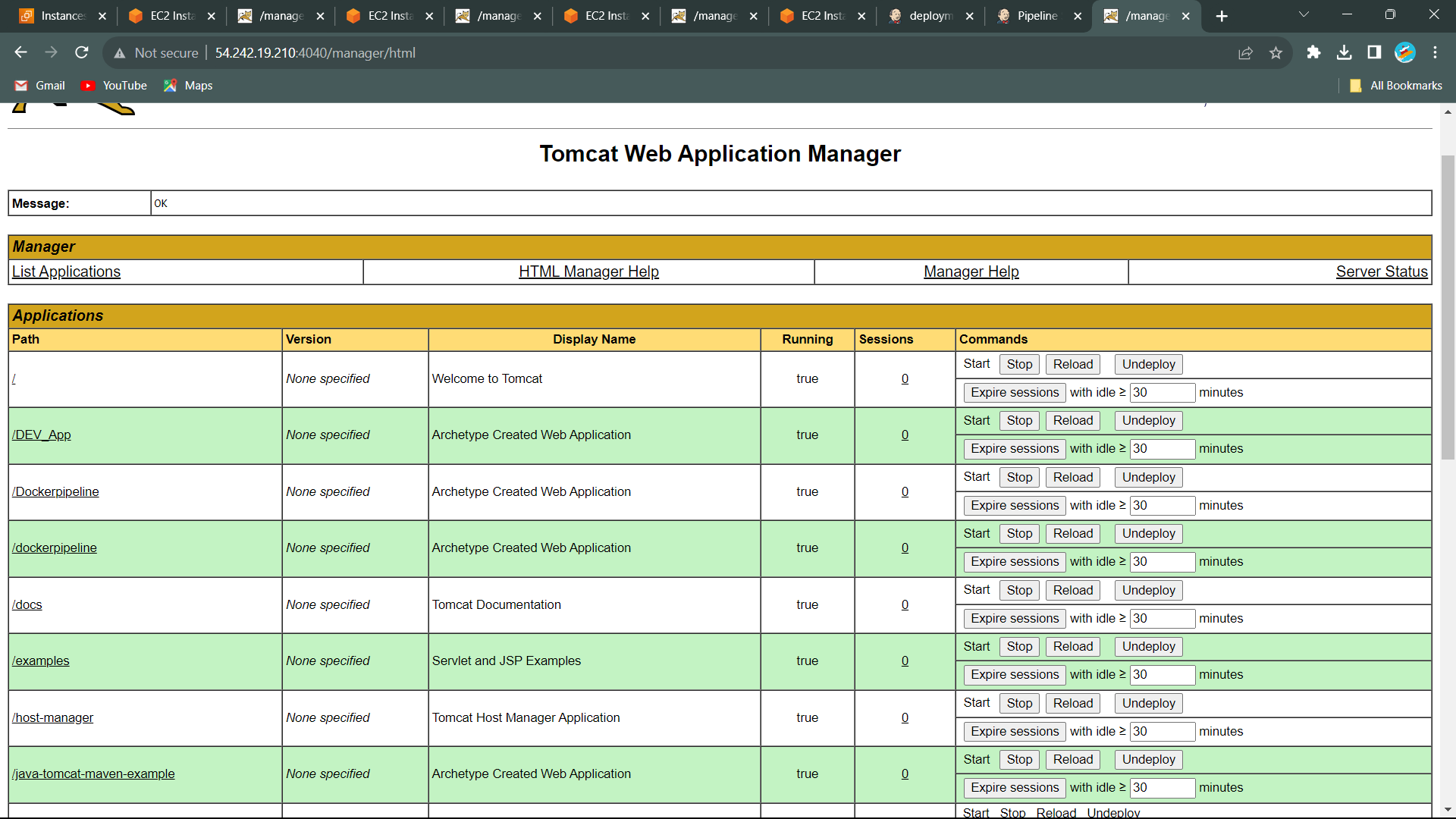


**Deployment Successful…**

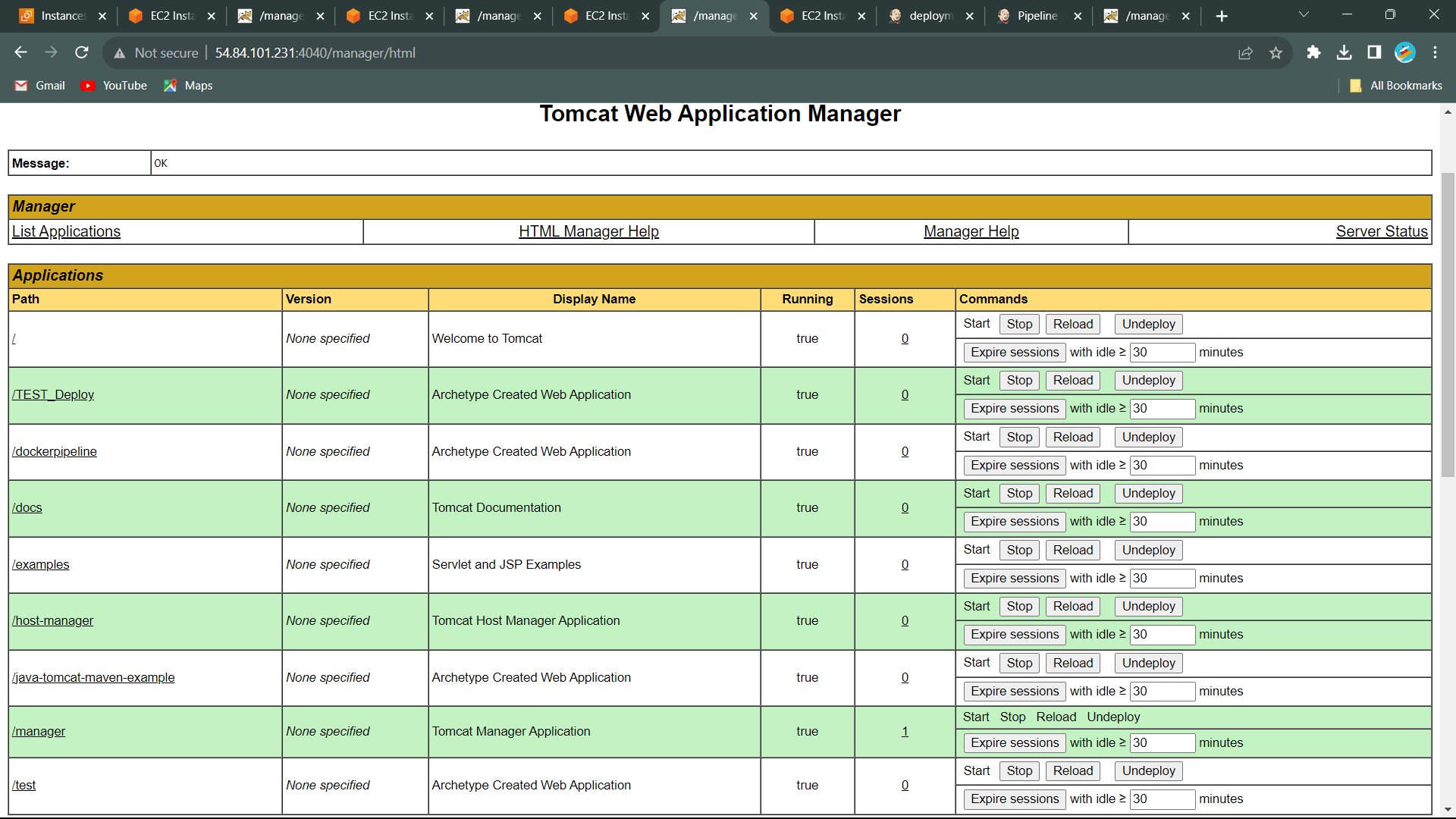
* Check the Tomcat sever whether Application is Deployed or not in it……..

1. See Application Deployed in DEV\_ENVIRONMENT.
2. See Application Deployed in TEST\_ENVIRONMENT.
3. See Application Deployed in PRE\_PROD\_ENVIRONMENT.
4. See Application Deployed in PROD\_ENVIRONMENT.
5. DEV\_ENVIRONMENT

Name: DEV\_App

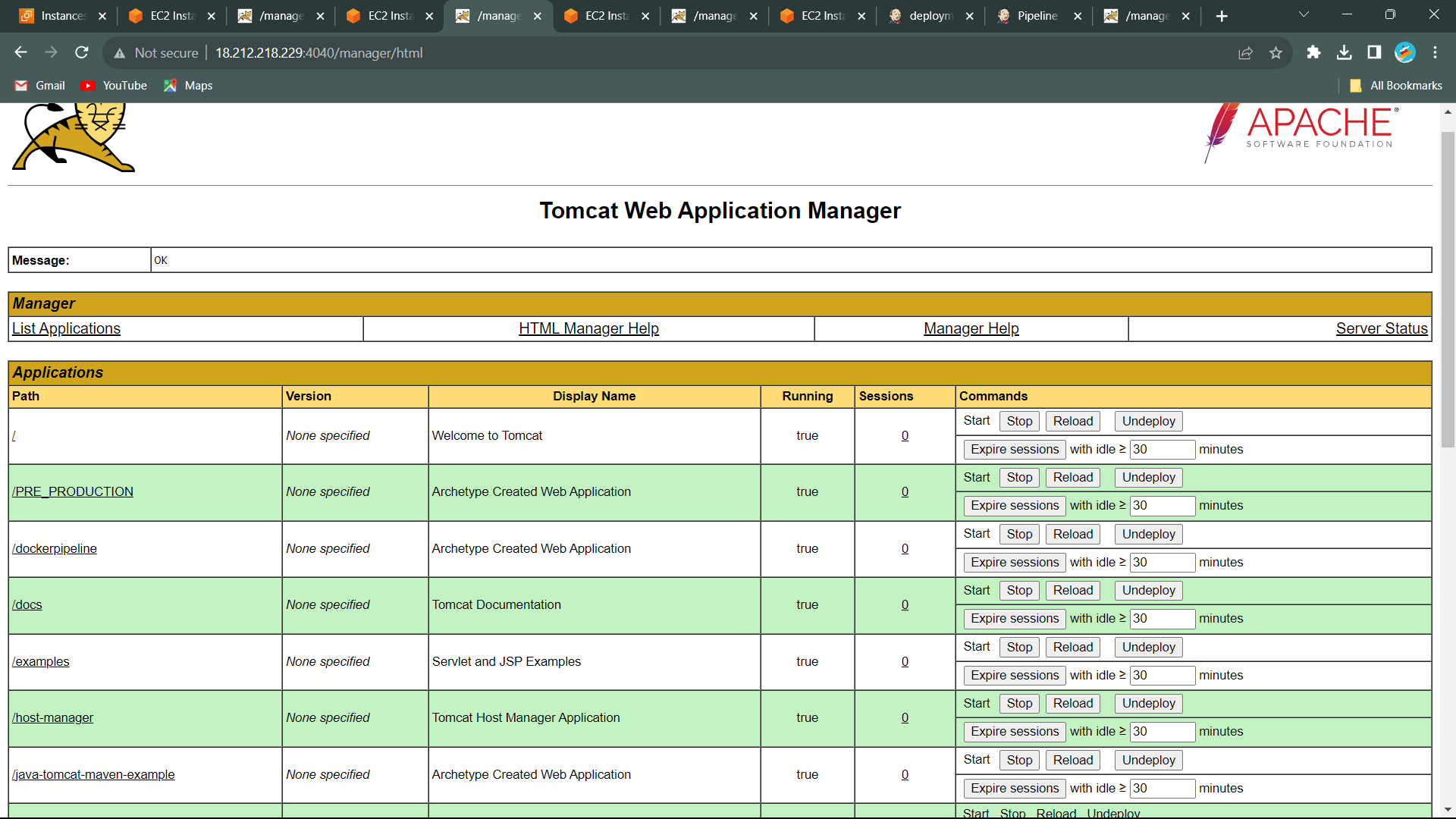


1. TEST\_ENVIRONMENT

Name: TEST\_Deploy

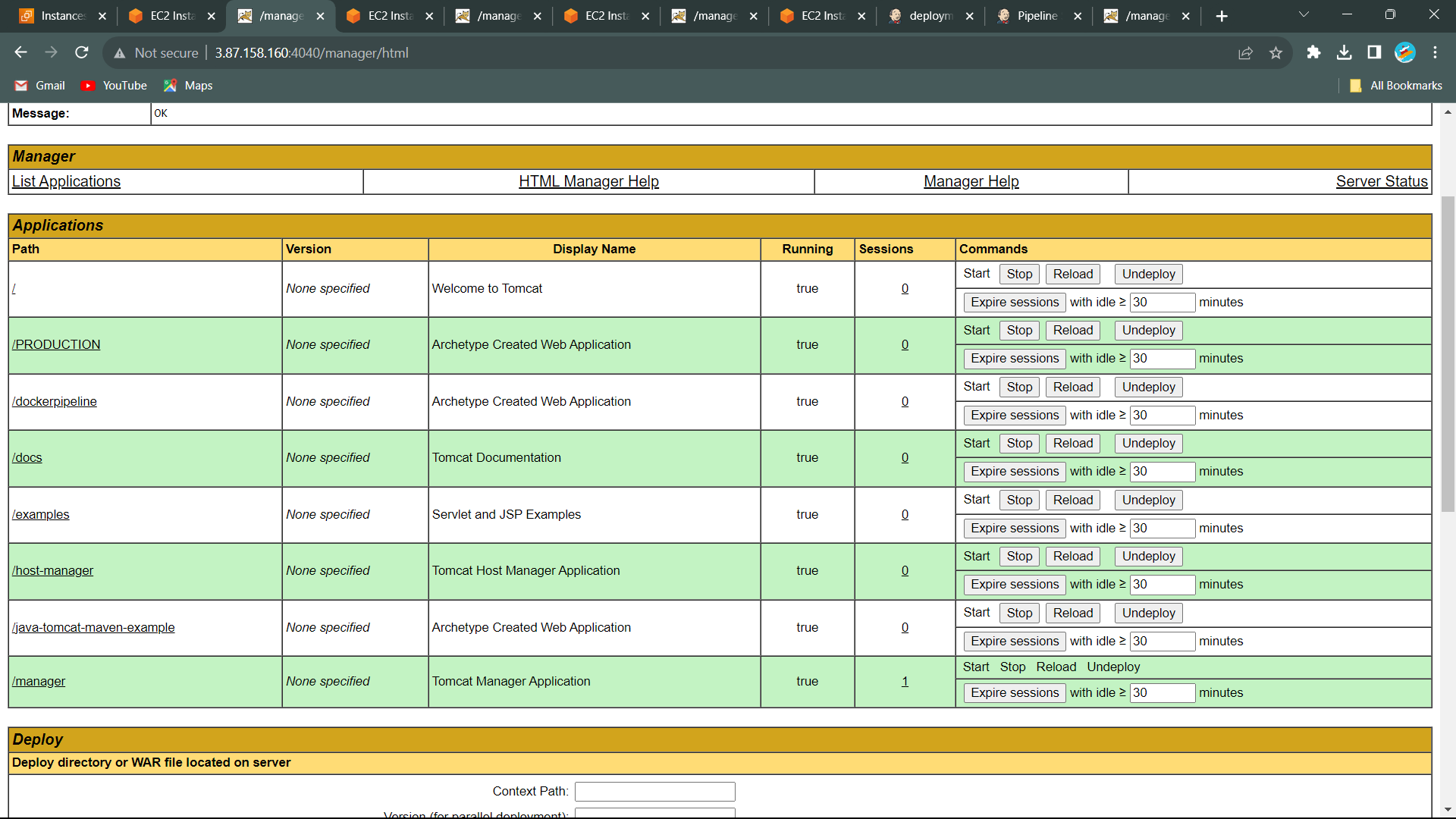
1. PRE\_PROD\_DEPLOYMENT

Name: PRE\_PRODUCTION



1. PROD\_ENVIRONMENT

Name: PRODUCTION



--------------- END ---------------