Course: DevOps **Name: CH.Lakshmi Priyank**

Module: Tomcat **Mail-ID: chlakshmipriyanka9@gmail.com**
Topic: Deploy in Tomcat **Batch no: 115**
Trainer Name: Mr. Madhukar sir **Project No: 01** **Date of submission: 6 – Dec – 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 a 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, create the new project instance, and connect the command line interface.

2. Change the user to the root user using (sudo -i).

3. Use the command to update the server (apt update -y).

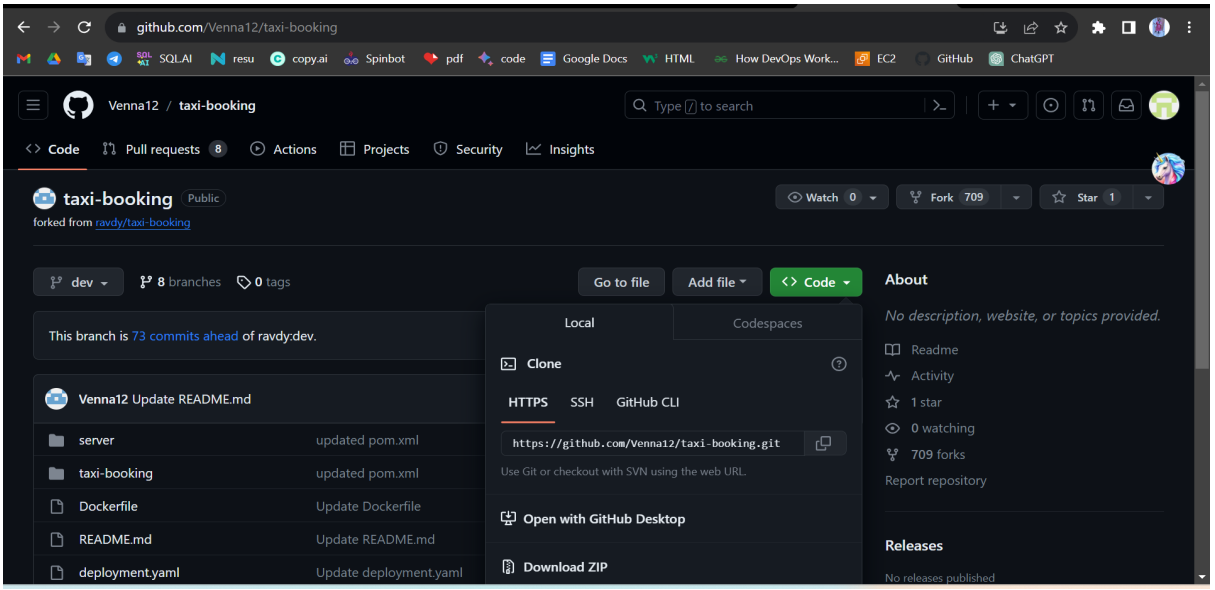
4. Install Java 11 using the command (apt install default-jdk -y).

5. Install the maven using the command (apt install maven).

6. Install the Jenkins server using the command line.

7. Install the Tomcat server in the server.

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 the pipeline. Job name Taxi-Booking.

^ We configure the dashboard and click on Git, add the clone link to the given box.

PROJECT URL:https://github.com/Venna12/taxi-booking.git

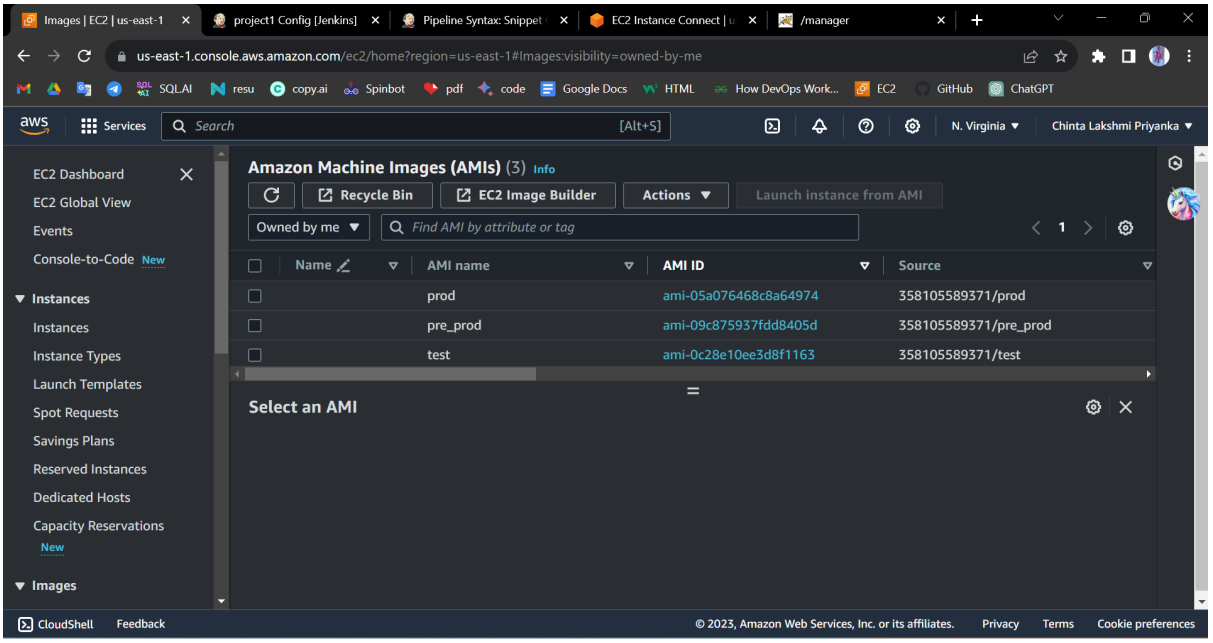
Create the 3-AMI's Using Single Instance

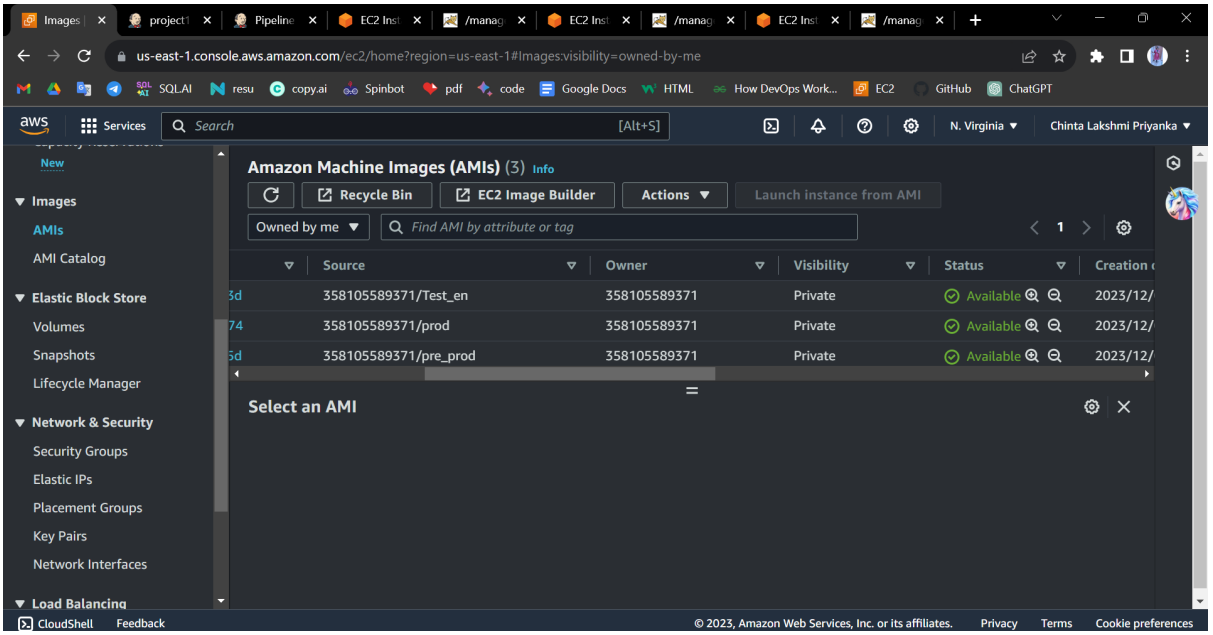
- Go to the Main instance and go to Images and Templates.
- Click on the Create the Image, create the 3 Images.
- Create AMI's for the

1. **TEST_ENVIRONMENT**
2. **PRE_PRODUCTION_ENVIRONMENT**
3. **PRODUCTION_ENVIRONMENT**

- These are the Amazon Machine Images, related to our instances, these instances are complete images of our main instances so that in all AMIs, we get the data, software, and servers the same as in the main instances.
- But the change of these is an IP address, we get a different IP address for them.

- We should deploy our application in these AMIs using a single pipeline.





- 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 servers of all instances. Using their IPs with Port no.

^ Go to Instances and connect to the command interface.

^ Go to a user to the root user (sudo -i) and go to Tomcat (cd tomcat).

^ Go to the bin and start the Tomcat using the command (./startup.sh).

^ Copy the IPV4 of instances and paste it on Google Search and add

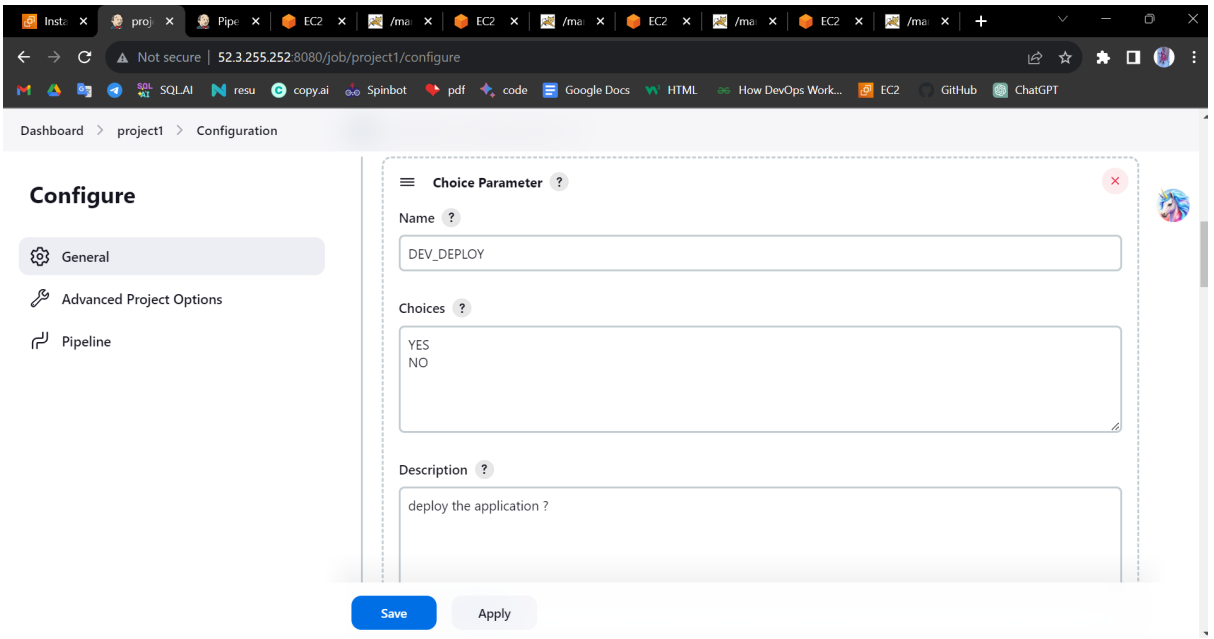
port no. Choice Parameter

- Go to pipeline Configuration and click on Add parameter.

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

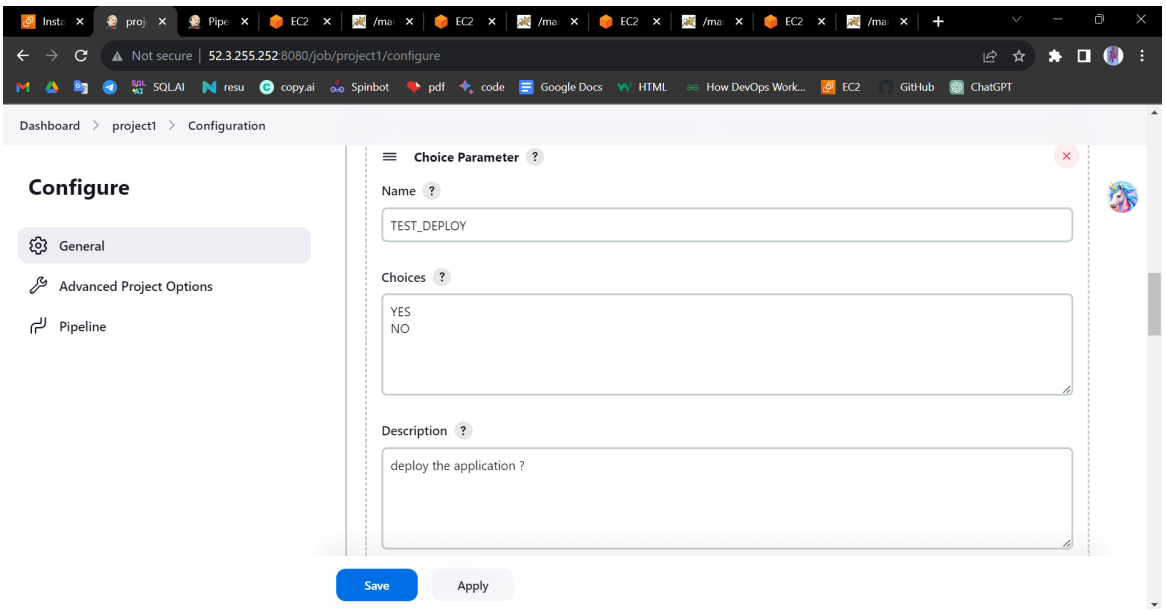
- First, we enter the Parameters for the DEV_ENVIRONMENT. As shown in the figure

    1. Name: DEV_DEPLOYMENT
    2. Choices: YES or NO
    3. Description: DO YOU WANT DEPLOY APPLICATION TO DEV_ENVIRONMENT.



And then, we enter the Parameters for the DEV_ENVIRONMENT. As shown in the figure

4. Name: TEST_DEPLOYMENT
5. Choices: YES or NO
6. Description: DO YOU WANT DEPLOY APPLICATION TO TEST_ENVIRONMENT.

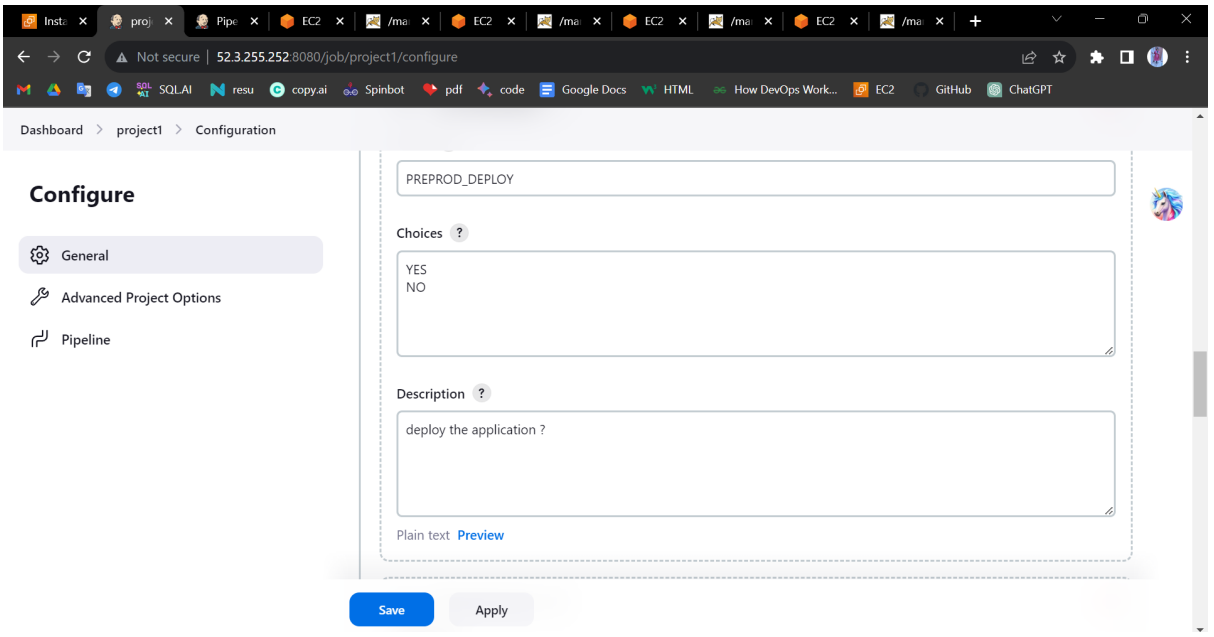


And then we enter the Parameters for the DEV_ENVIRONMENT. As shown in the figure…

7. Name: PRE_DEPLOYMENT

8. Choices: YES or NO

9. Description: DO YOU WANT DEPLOY APPLICATION TO PRE_PROD_ENVIRONMENT.

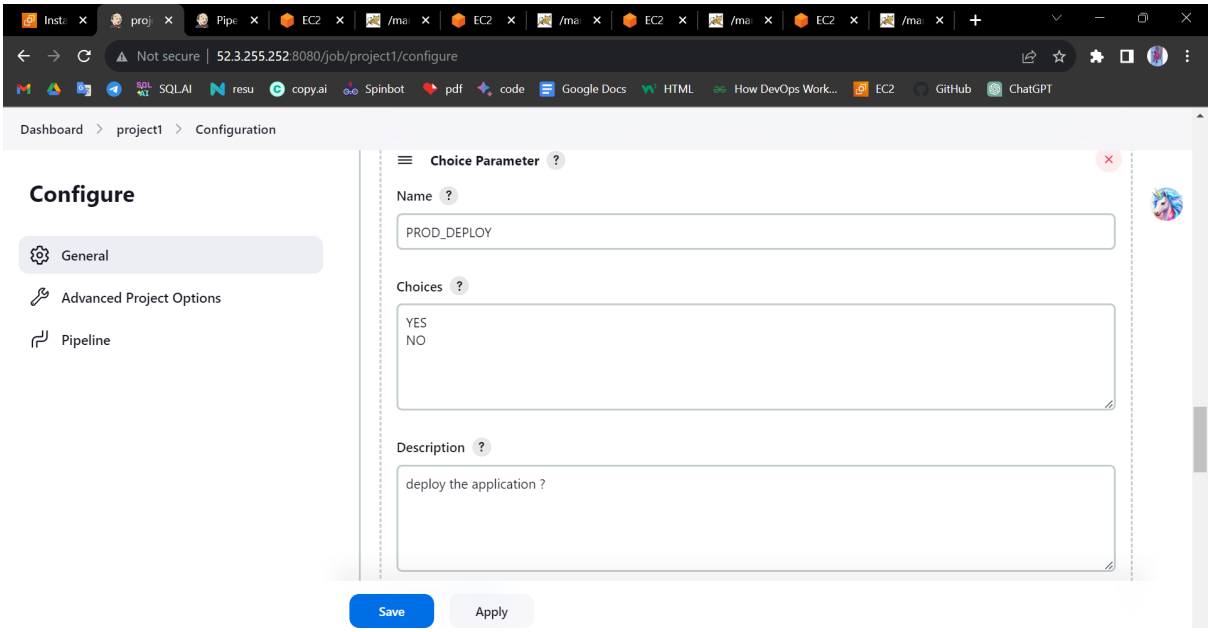


And then, we enter the Parameters for the DEV_ENVIRONMENT. As shown in the figure…

10. Name: PROD_DEPLOYMENT

11. Choices: YES or NO

12. 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 another script to connect parameters 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

```
pipeline{
    agent any
    parameters{
        choice (
            choices: ['YES', 'NO'],
            description: 'deploy the application ?',
            name: 'DEV_DEPLOY'
        )
        choice (
            choices: ['YES', 'NO'],
            description: 'deploy the application ?',
            name: 'TEST_DEPLOY'
        )
         choice (
            choices: ['YES', 'NO'],
            description: 'deploy the application ?',
            name: 'PREPROD_DEPLOY'
        )
           choice (
            choices: ['YES', 'NO'],
            description: 'deploy the application ?',
```

```
        name: 'PROD_DEPLOY'
      )
    }
    stages{
      stage('clone'){
        steps{
          checkout scmGit(branches: [[name: '*/dev']],
extensions: [], userRemoteConfigs: [[url:
'https://github.com/Venna12/dockerjenkin.git']])
        }
      }
      stage('build'){
        steps{
          sh 'mvn package'
        }
      }
      stage('deploy'){
        when{
          expression{
            params.DEV_DEPLOY =='YES'
          }
        }
        steps{
```

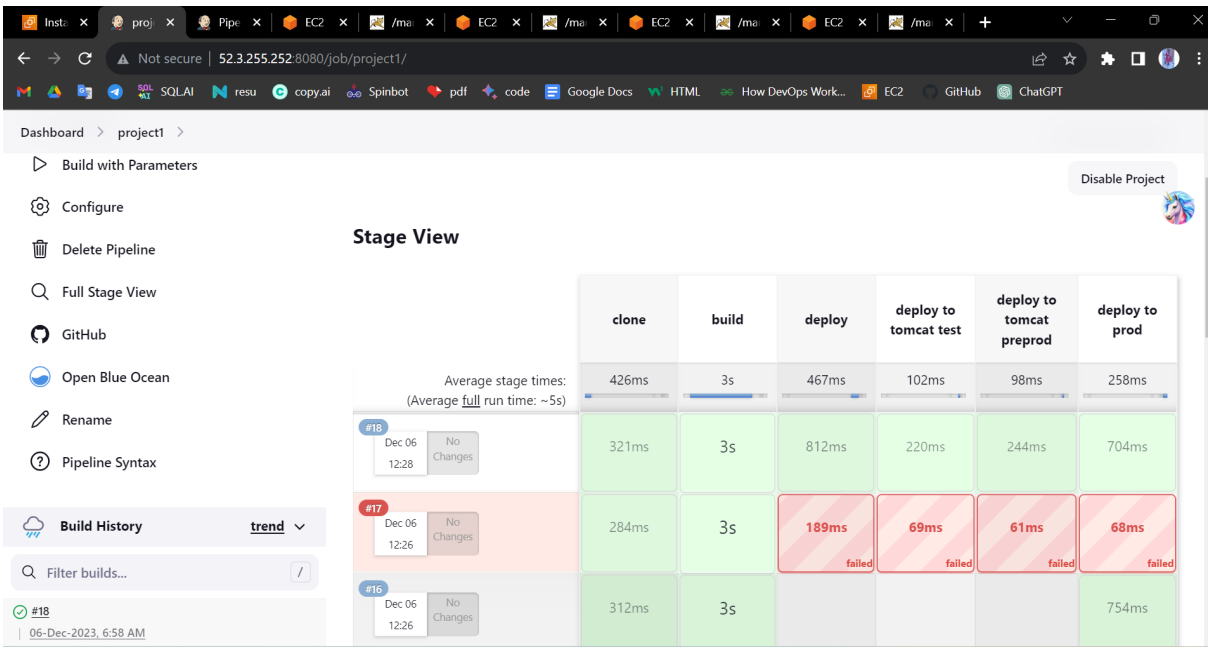
```
            deploy adapters: [tomcat9(credentialsId: 'tomcat',
path: '', url: 'http://52.3.255.252:8082/')], contextPath: 'priya',
war: '**/*.war'
         }
      }
      stage('deploy to tomcat test'){
         when{
            expression{
               params.TEST_DEPLOY =='YES'
            }
         }
         steps{
            deploy adapters: [tomcat9(credentialsId: 'tomcat',
path: '', url: 'http://54.163.214.63:8082/')], contextPath: 'test',
war: '**/*.war'
         }
      }
      stage('deploy to tomcat preprod'){
         when{
            expression{
               params.PREPROD_DEPLOY =='YES'
            }
         }
         steps{
```

```
            deploy adapters: [tomcat9(credentialsId: 'tomcat',
path: '', url: 'http://18.213.0.121:8082/')], contextPath:
'pre_prod', war: '**/*.war'
         }
      }
      stage('deploy to prod'){
         when{
            expression{
               params.PROD_DEPLOY =='YES'
            }
         }
         steps{
            deploy adapters: [tomcat9(credentialsId: 'tomcat',
path: '', url: 'http://52.90.127.93:8082/')], contextPath: 'prod',
war: '**/*.war'
         }
      }


   }
}
```

^ Application is Successfully Build and Deployed in Different Environments.

^ Output displayed in the Build pipeline format.



Deployment Successful

- Check the Tomcat server whether the Application is Deployed or not in it

  1. See Application Deployed in DEV_ENVIRONMENT.

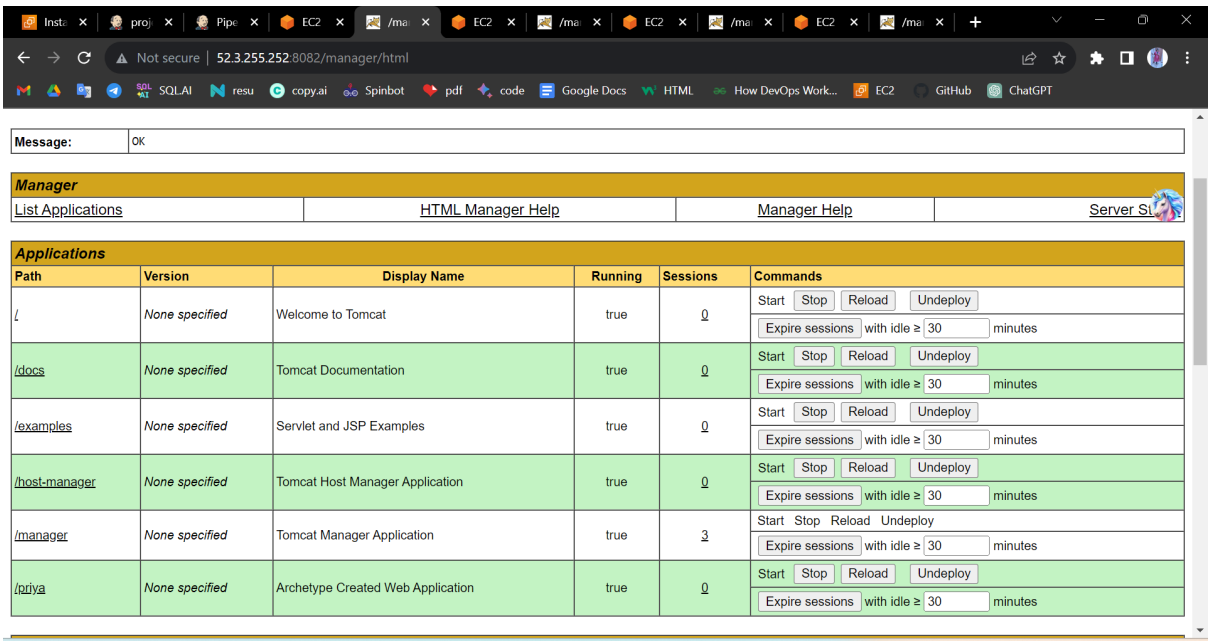
  2. See Application Deployed in TEST_ENVIRONMENT.

  3. See Application Deployed in PRE_PROD_ENVIRONMENT.

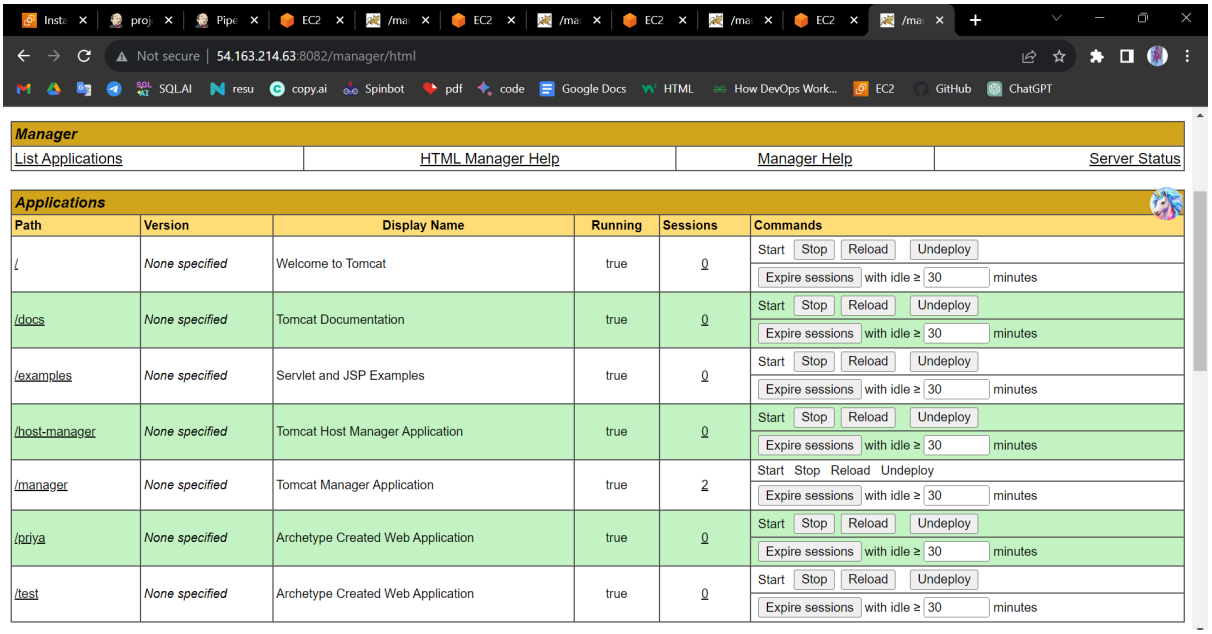
  4. See Application Deployed in Deployed in PROD_ENVIRONMENT.

1. DEV_ENVIRONME

   NT Name: DEV_App

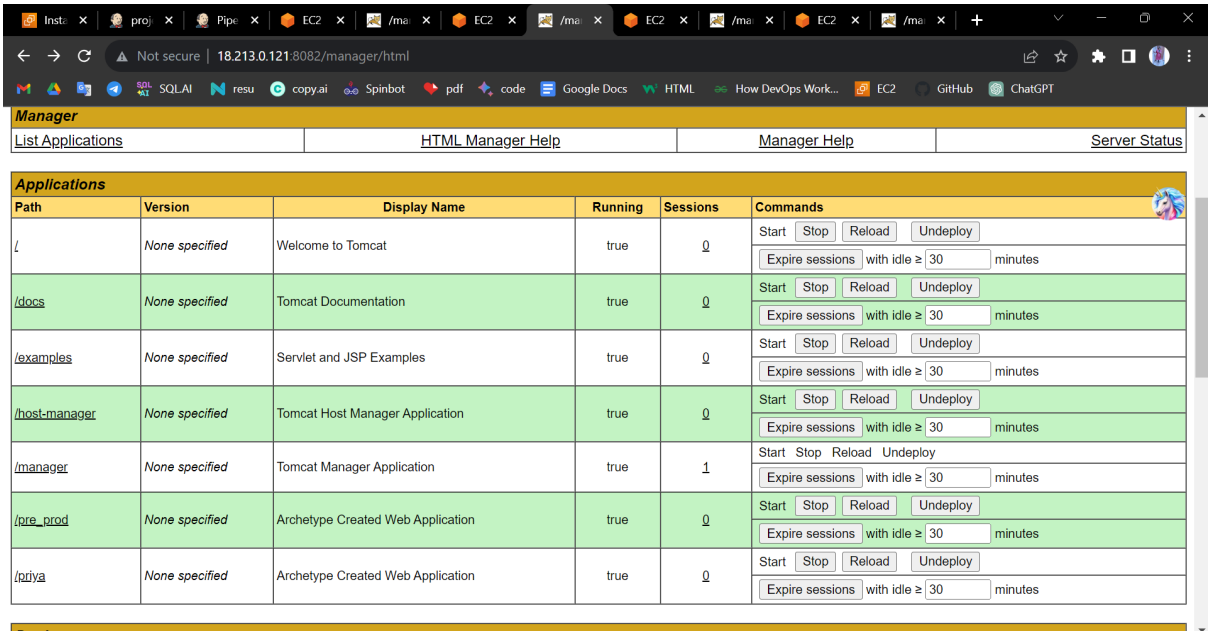


2. TEST_ENVIRONMENT Name: TEST_Deploy



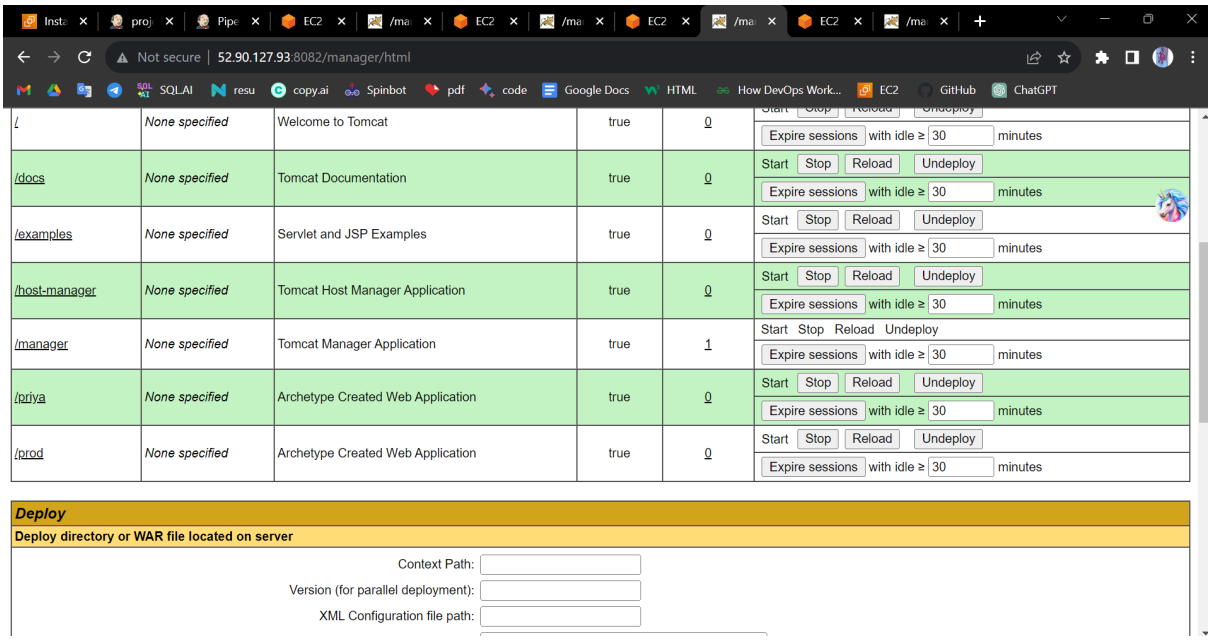
3. PRE_PROD_DEPLOYMENT

Name: PRE_PRODUCTION



4. PROD_ENVIRONMENT

Name: PRODUCTION



THE END