Description

Demonstrate the continuous integration and delivery by building a Docker Jenkins Pipeline.

Problem Statement Scenario:

You are a DevOps consultant in AchiStar Technologies. The company decided to implement DevOps to develop and deliver their products. Since it is an Agile organization, it follows Scrum methodology to develop the projects incrementally. You are working with multiple DevOps Engineers to build a Docker Jenkins Pipeline. During the sprint planning, you agreed to take the lead on this project and plan on the requirements, system configurations, and track the efficiency. The tasks you are responsible for:

Solution

As part of the Course End Project -2

1) We have a Code repository with its commit.

   Using the code located in - https://github.com/Sonal0409/DevOpsCodeDemo.git

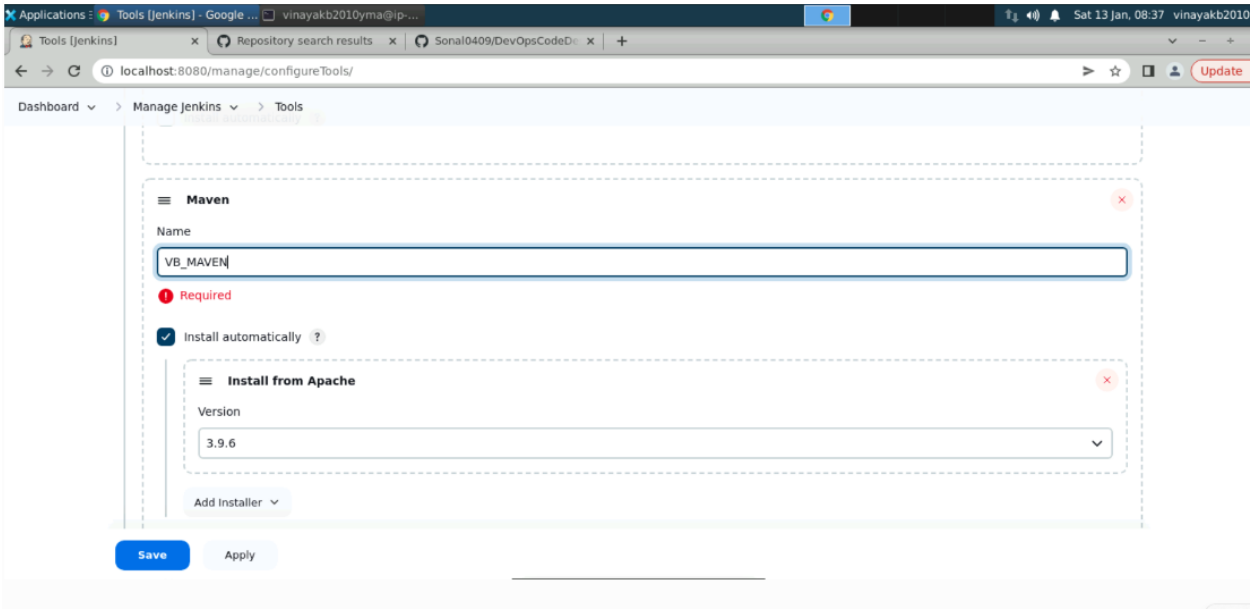
2) CI Tool used is Maven server
   a. We use Jenkins pipeline to get the code

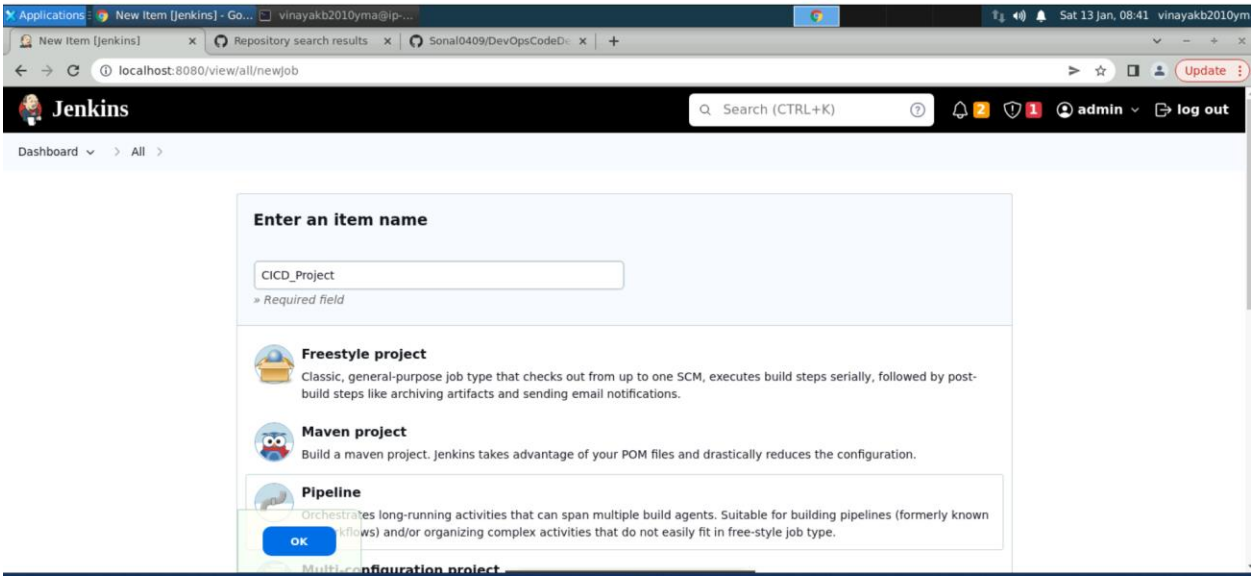
   b. Get the code Code →github → maven(maven Command to compile, test, and package the code) Artifacts/build
   c. Once the Build is available, we will have to deploy the code on a container

Continuous Integration

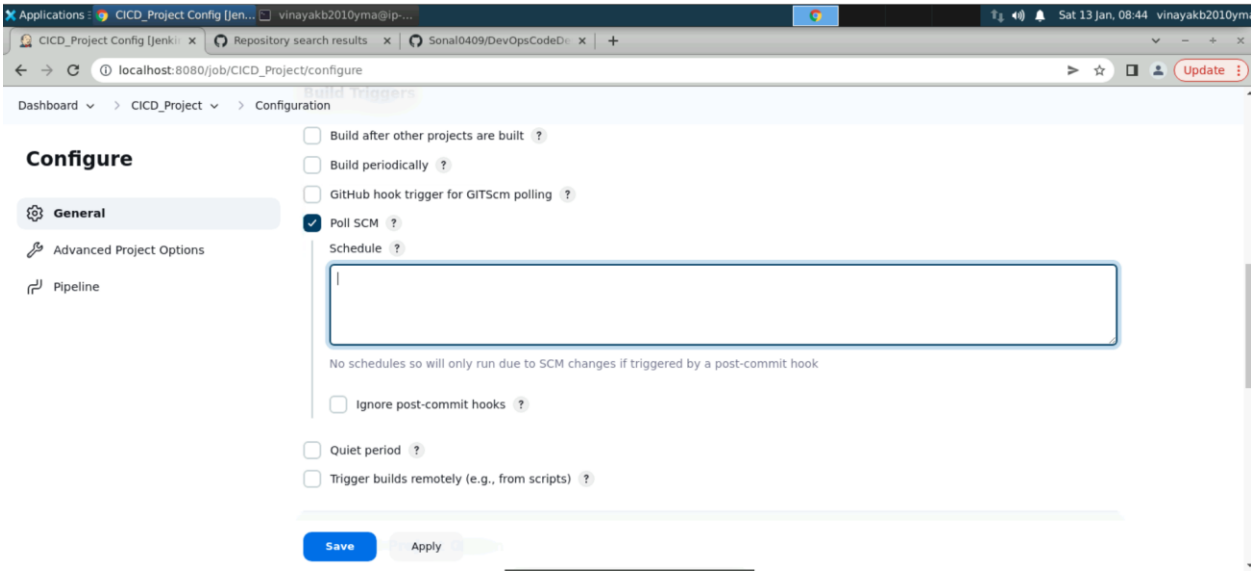
1) Launch Jenkins and Setup Maven tool configuration as we must perform the CI.

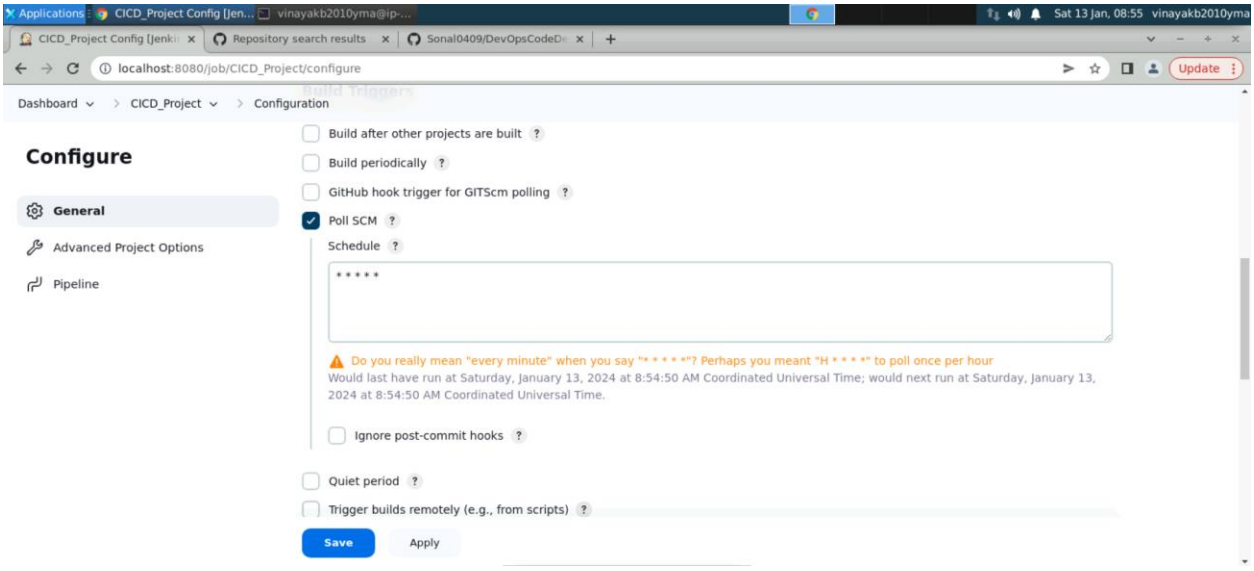


2) Create the Jenkins Pipeline
    a. Create a Pipeline a job in which we write code to clone the repository and package the code.



- The below Trigger was chosen because every time we there is a new commit, we should be able to deploy the new code.
- Jenkins will be running this job if there is a new code in the Git.
- Jenkins Pulls the Git every minute.
-
-

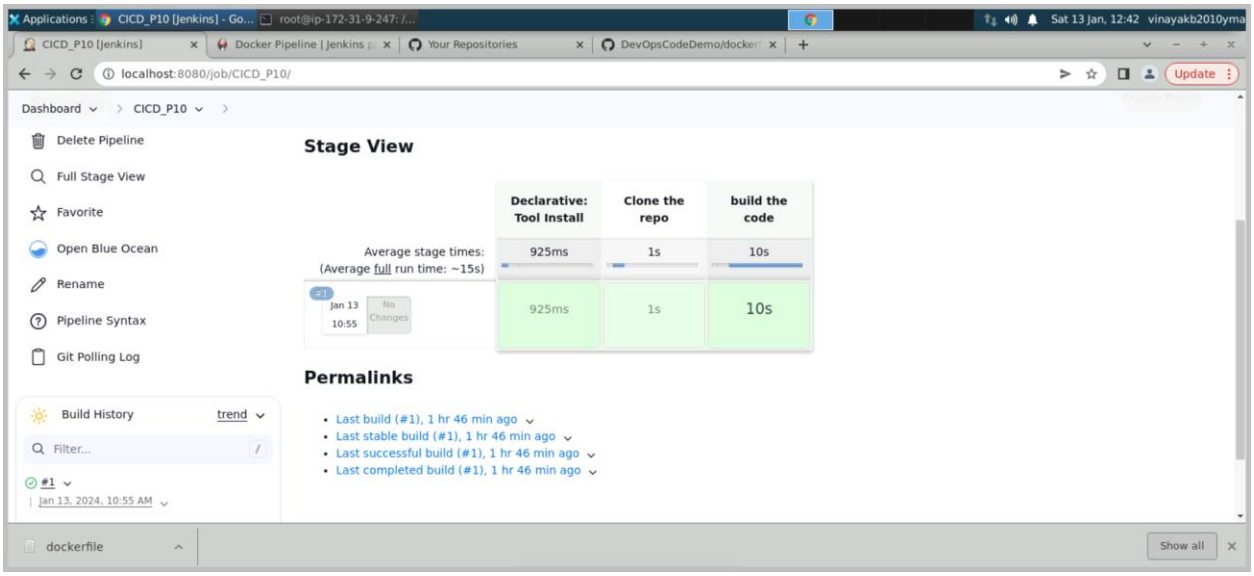


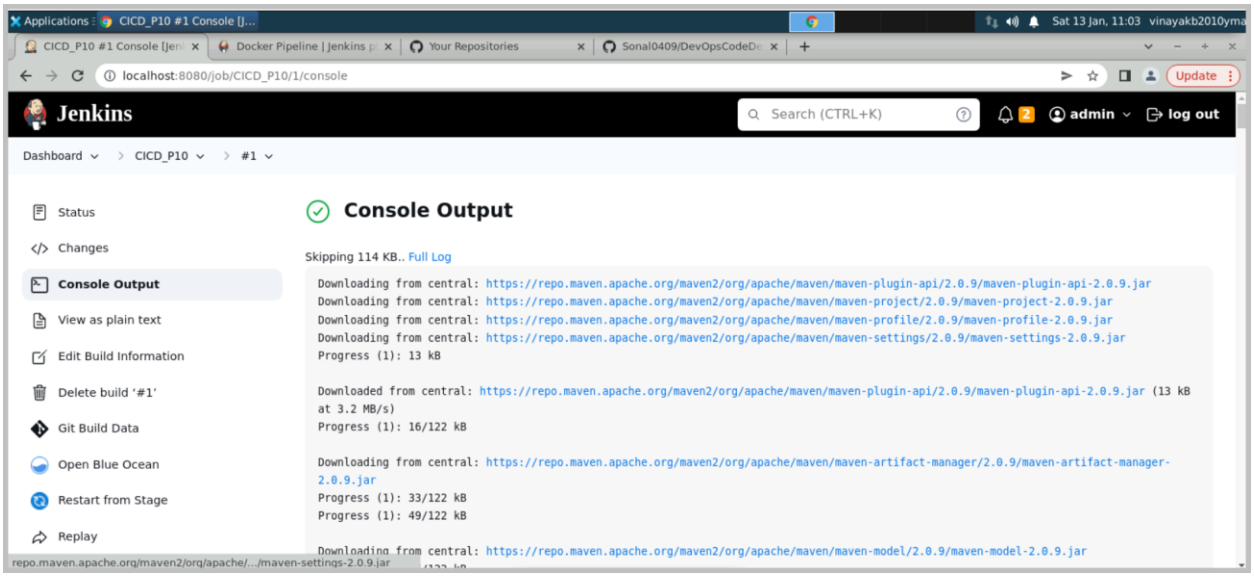


In the pipeline enter the script

```
pipeline{
   tools{ maven 'VB_MAVEN' }

   agent any

   stages{
      stage('Clone the repo')
                    {
         steps{ git 'https://github.com/Sonal0409/DevOpsCodeDemo.git' }
             }

      stage('build the code'){;
        steps{
            sh 'mvn package'
         }
      }
}
}
```

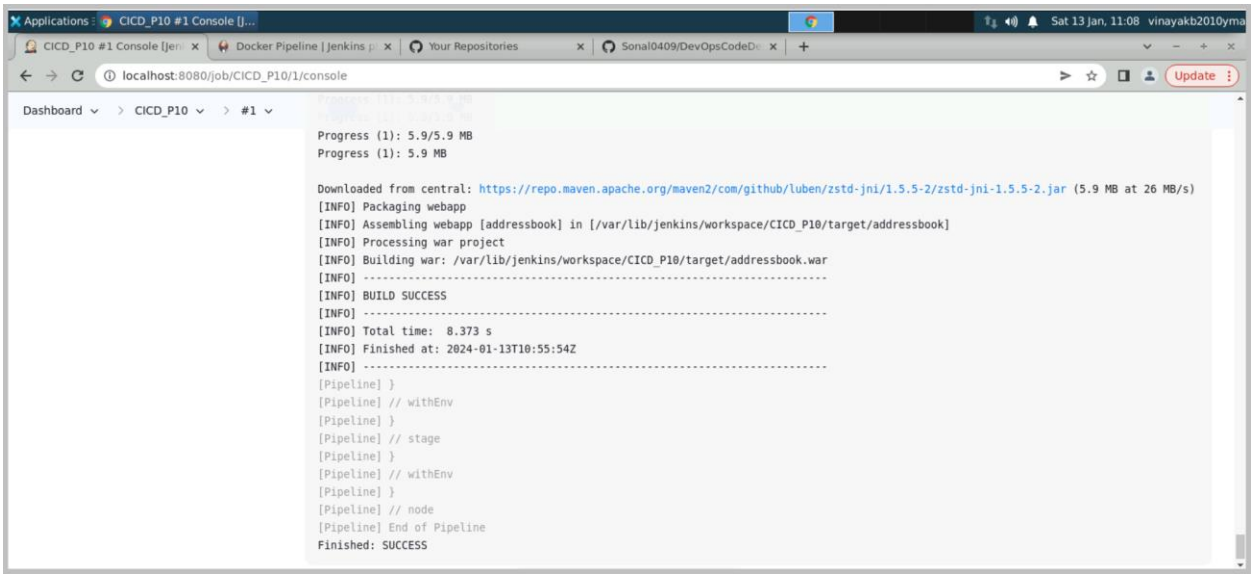
Save the above pipeline and build it, we will find the package in the Jenkins workspace Target Folder





We need to copy the path where the Application Package is present

Building war: /var/lib/jenkins/workspace/CICD_P10/target/addressbook.war



Step 3)

Continuous Delivery

1) Use the linux command to copy the **addressbook.war** into the same directory that of Docker file
2) We need to take the above file and build the DockerFile that creates a custom image.
    a. Below is the Docker File

DockerFile

FROM tomcat:9

ADD addressbook.war/usr/share/tomcat/webapps.

EXPOSE 8080

CMD [“cataline.sh”, “run”] //Cataline.sh is a home directory where a script is available in tomcat.

****

Build file must be placed where the Docker file is placed.

By Default Jenkins user can’t run Docker Commands.

Permission needs to be given to the user to run the Docker Commands

Execute the below command to allow the user to provide the commands

Chmod 777 /var/run/docker.sock

```
pipeline{
   tools{ maven 'VB_MAVEN' }

   agent any

   stages{
      stage('Clone the repo')
                    {
        steps{ git 'https://github.com/Sonal0409/DevOpsCodeDemo.git' }
            }

      stage('build the code'){
        steps{
           sh 'mvn package'
        }
                    }

          stage ('Copy the Build to workspace directory') {

                     steps {

          //Copy the file from target folder to workspace folder where Docker File exists
          // In the below command end, "." means in the current directory
                    sh 'cp /var/lib/jenkins/workspace/CICD_P10/target/addressbook.war .'
                          }
                                                          }
```

stage ('Build the Image') {

steps {

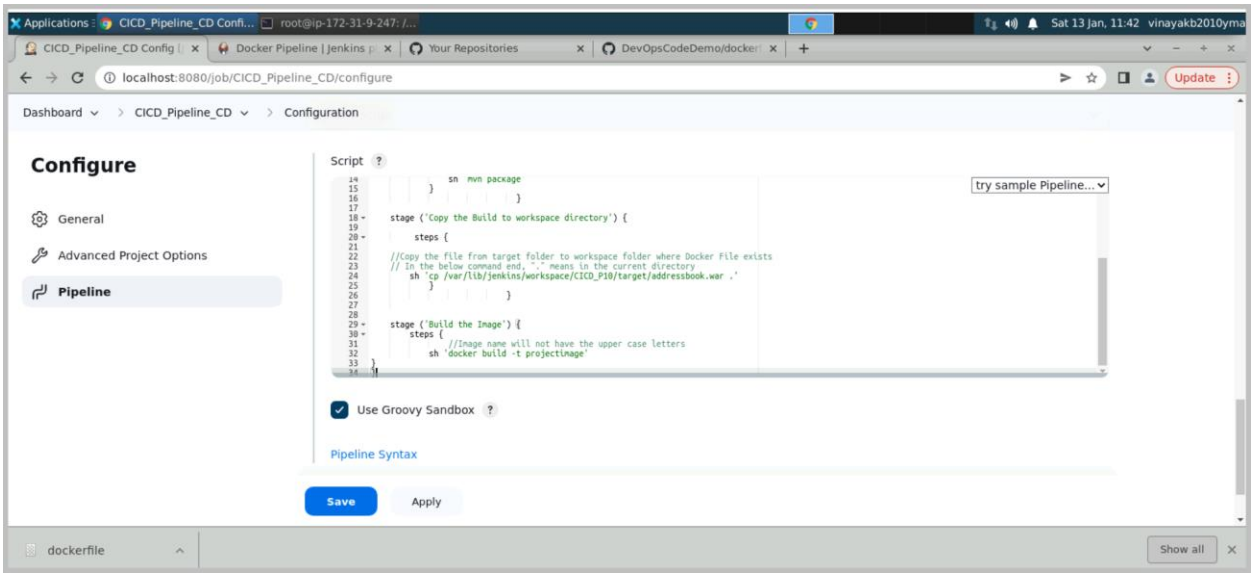
//Image name will not have the upper case letters

sh 'docker build -t projectimage'

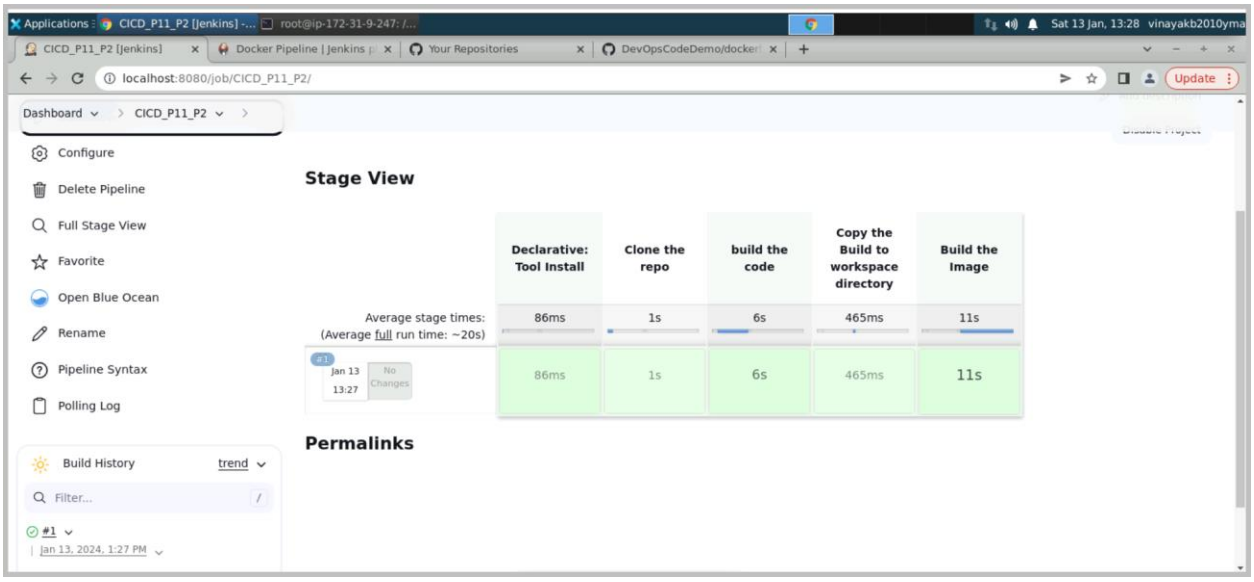
}

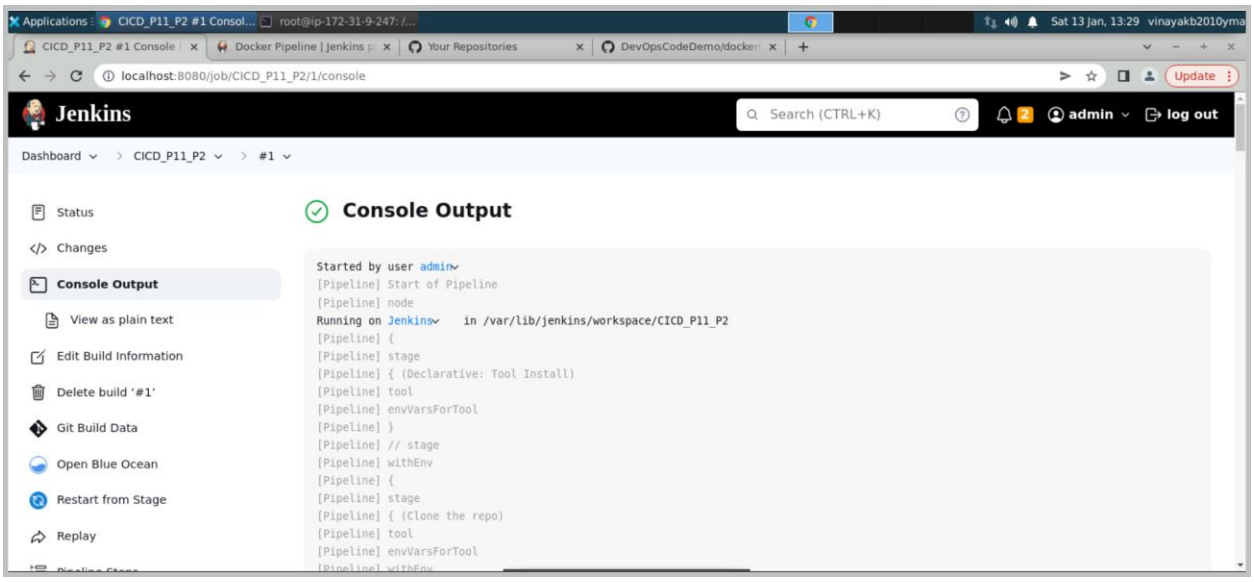
}

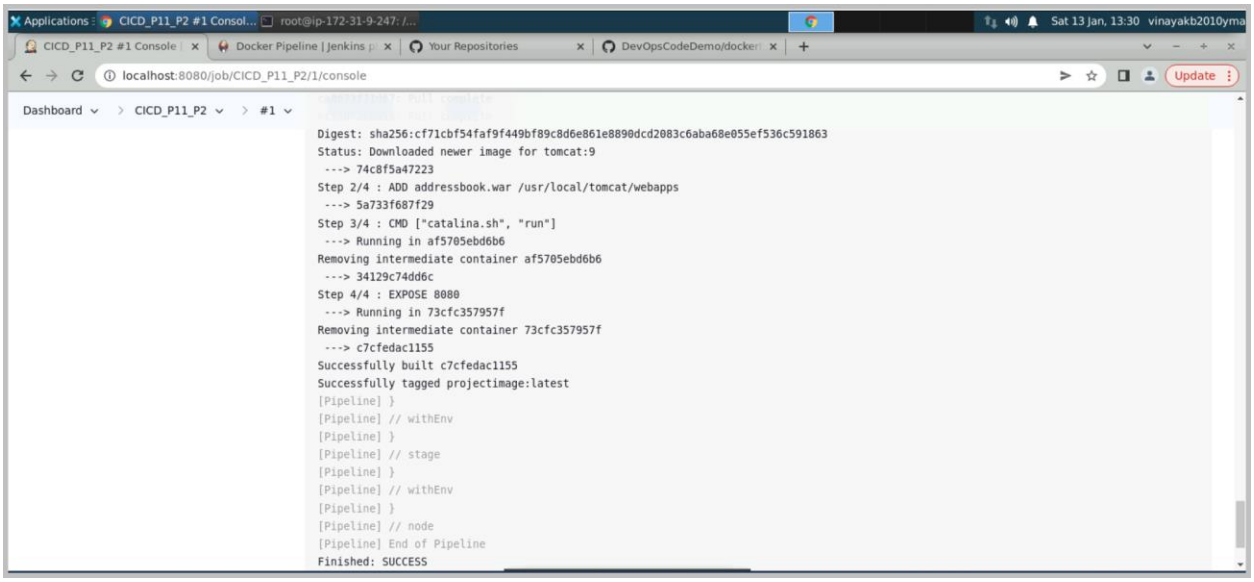
****

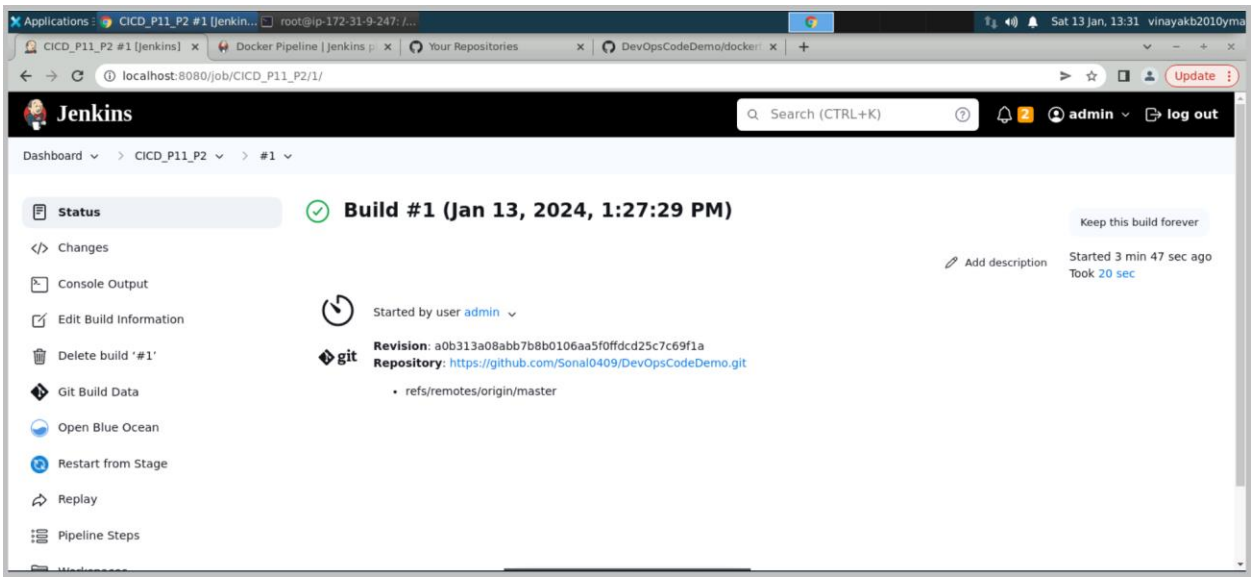


Let's Build this



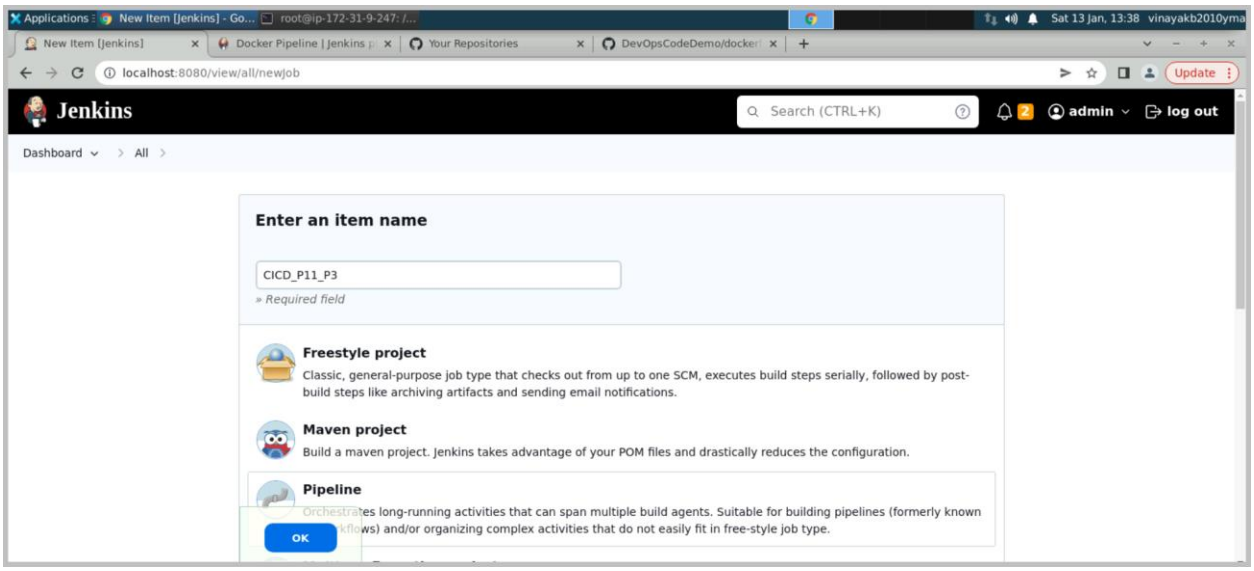

Applications
CICD_P11_P2 #1 Consol...
root@ip-172-31-9-247: /...
Sat 13 Jan, 13:29
vinayakb2010yma
CICD_P11_P2 #1 Console
Docker Pipeline | Jenkins
Your Repositories
DevOpsCodeDemo/docker
localhost:8080/job/CICD_P11_P2/1/console
Update
Jenkins
Search (CTRL+K)
admin
log out
Dashboard
CICD_P11_P2
#1
Status
Changes
Console Output
View as plain text
Edit Build Information
Delete build '#1'
Git Build Data
Open Blue Ocean
Restart from Stage
Replay
Console Output
Started by user admin
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/CICD_P11_P2
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Tool Install)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Clone the repo)
[Pipeline] tool
[Pipeline] envVarsForTool

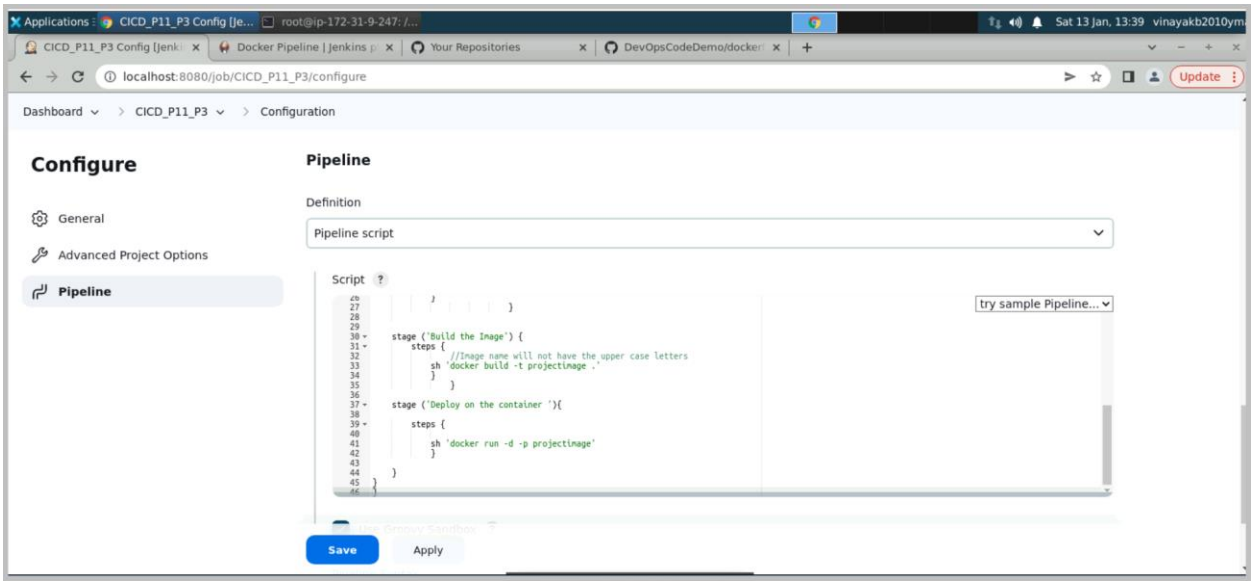

Applications
CICD_P11_P2 #1 Consol...
root@ip-172-31-9-247: /...
Sat 13 Jan, 13:30
vinayakb2010yma
CICD_P11_P2 #1 Console
Docker Pipeline | Jenkins
Your Repositories
DevOpsCodeDemo/docker
localhost:8080/job/CICD_P11_P2/1/console
Update
Dashboard
CICD_P11_P2
#1
Digest: sha256:cf71cbf54faf9f449bf89c8d6e861e8890dcd2083c6aba68e055ef536c591863
Status: Downloaded newer image for tomcat:9
---> 74c8f5a47223
Step 2/4 : ADD addressbook.war /usr/local/tomcat/webapps
---> 5a733f687f29
Step 3/4 : CMD ["catalina.sh", "run"]
---> Running in af5705ebd6b6
Removing intermediate container af5705ebd6b6
---> 34129c74dd6c
Step 4/4 : EXPOSE 8080
---> Running in 73cfc357957f
Removing intermediate container 73cfc357957f
---> c7cfedac1155
Successfully built c7cfedac1155
Successfully tagged projectimage:latest
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS



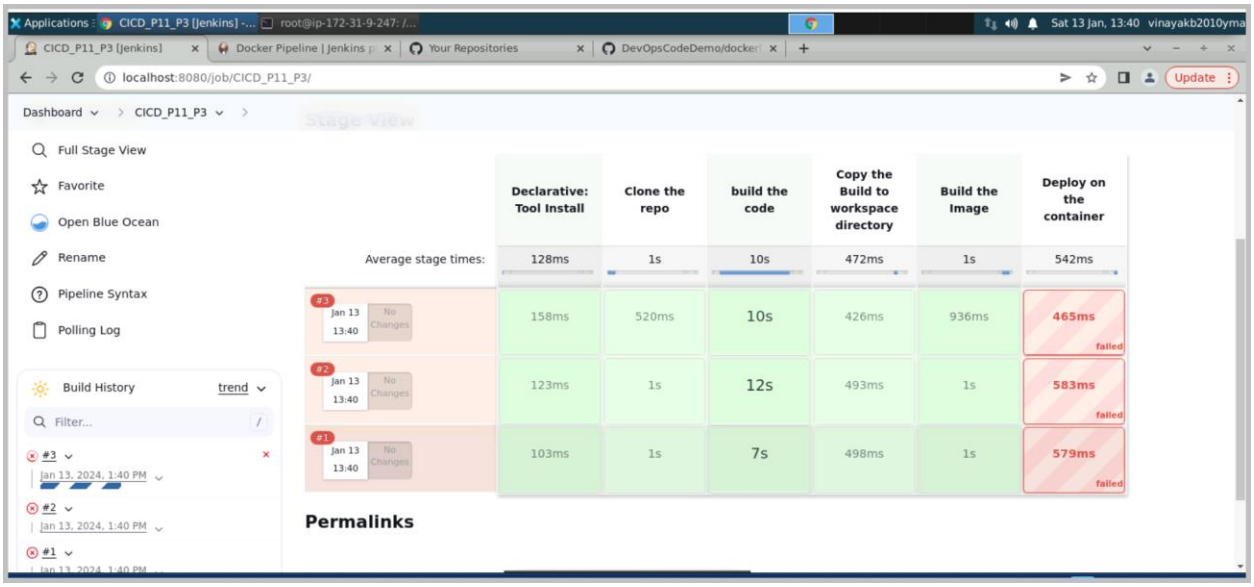
Now add the stage to run the image and deploy the application.

```
stage ('Deploy on the container '){
                        steps {
                        sh 'docker run -d -p projectimage'
                        }
        }
```



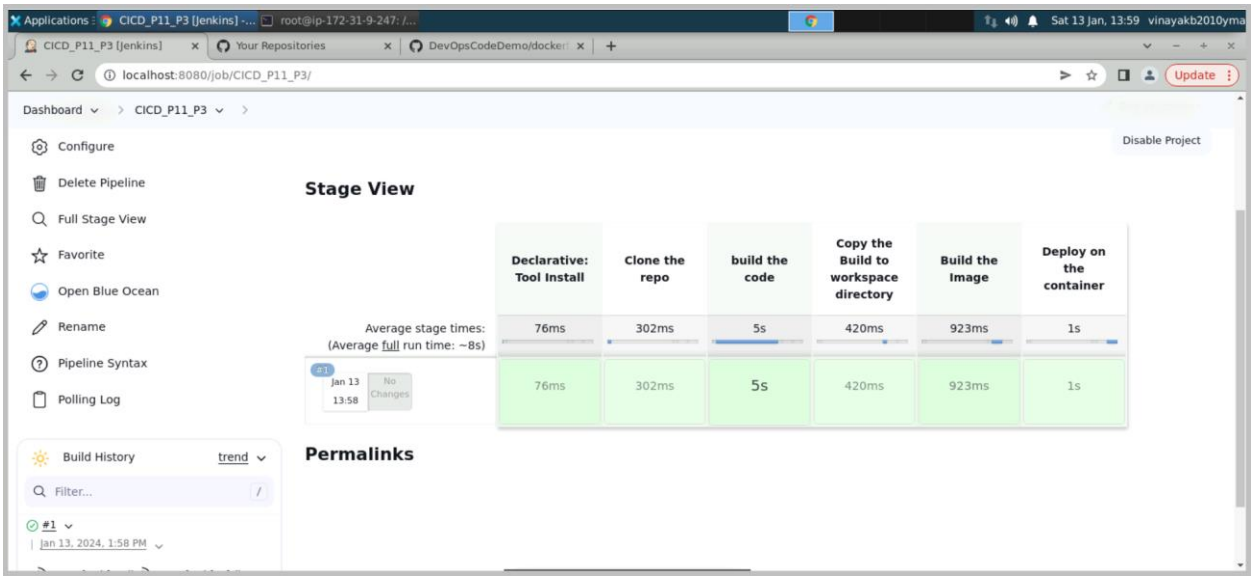


Initial Error



Later fixing the error in the command

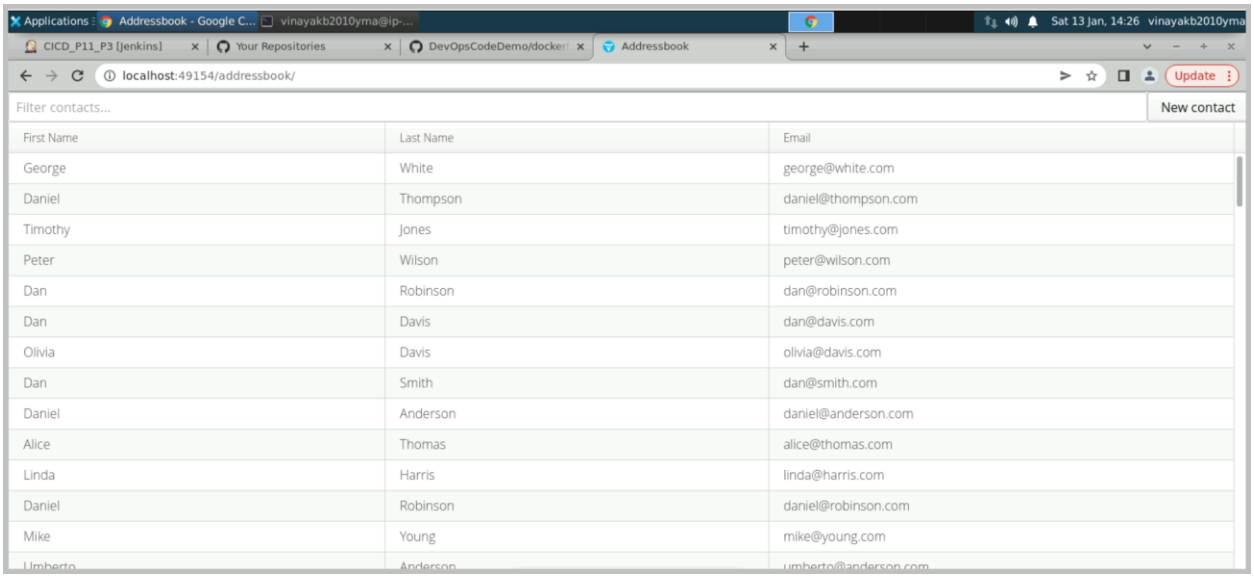
steps { sh 'docker run -d -P projectimage' }



Confirming the same from OS level

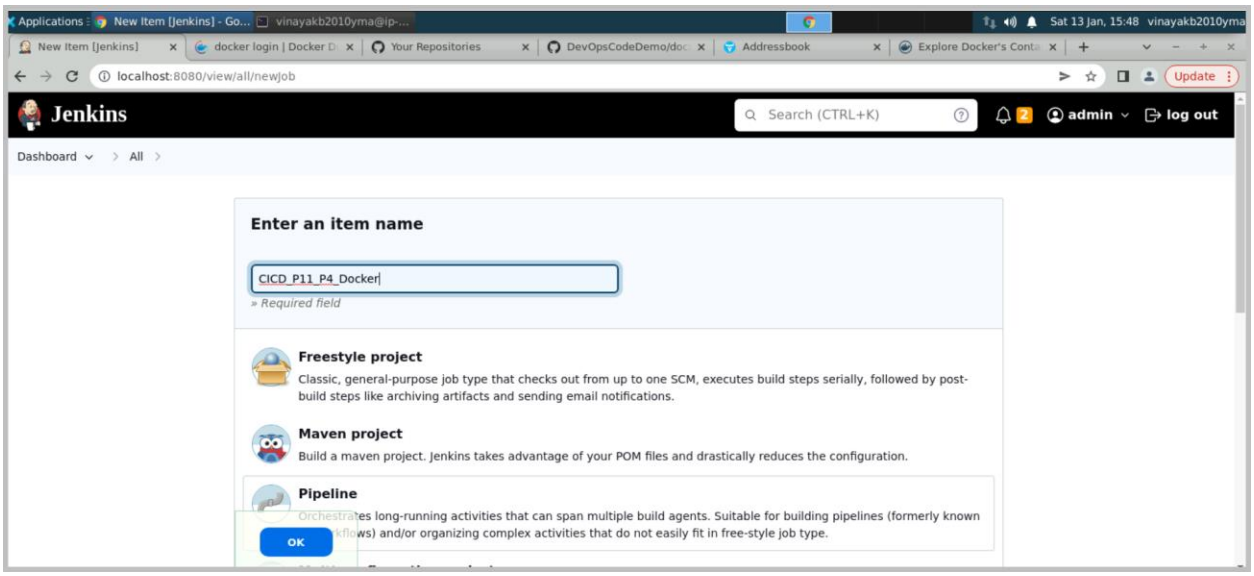
Command to confirm the same >> from the command line **docker ps --latest**

The Post mentioned in the above screen is 49154 – Using the same port we are launching the O/P

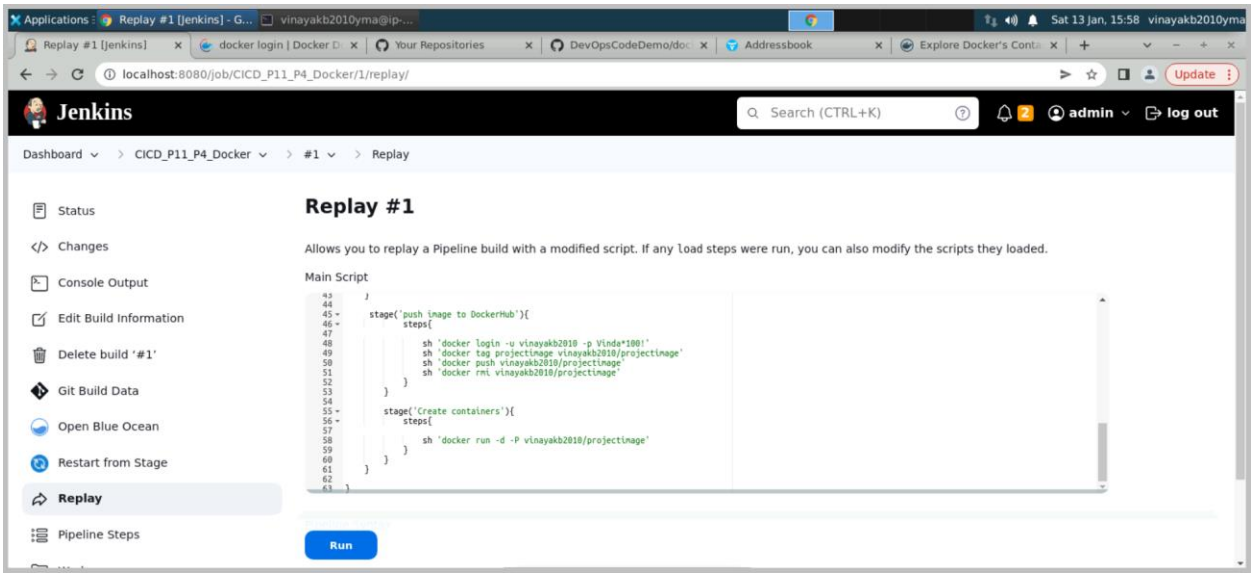


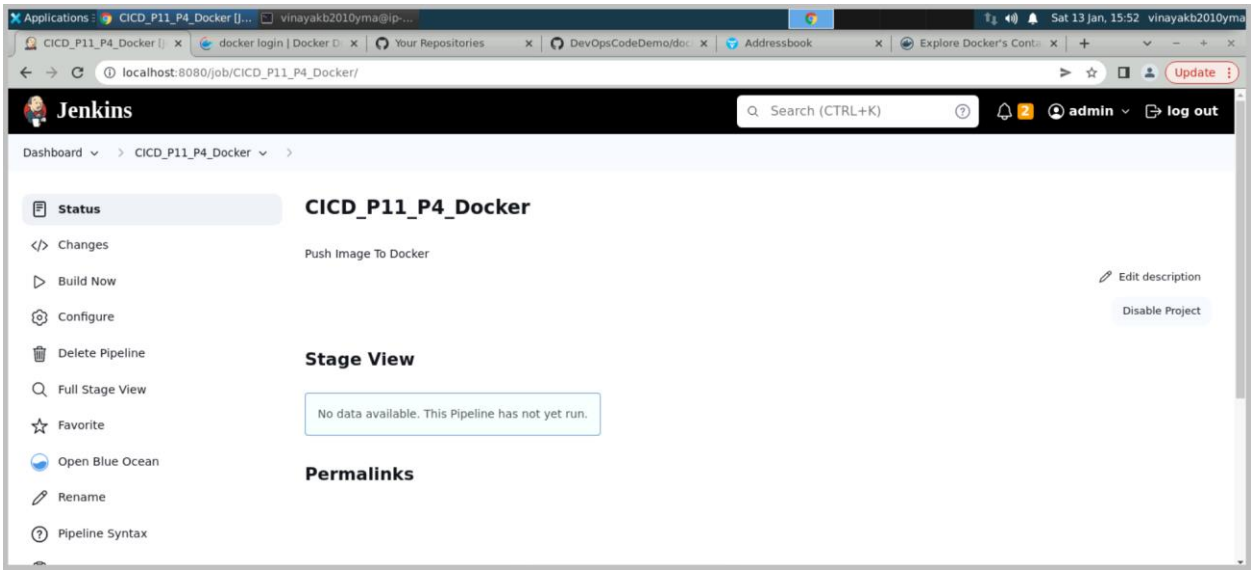
Create Image in docker.

```
            stage('push image to DockerHub'){
          steps{

             sh 'docker login -u vinayakb2010 -p Vinda*100!'
             sh 'docker tag projectimage vinayakb2010/projectimage'
             sh 'docker push vinayakb2010/projectimage'
             sh 'docker rmi vinayakb2010/projectimage'
          }
       }

       stage('Create containers'){
          steps{
             sh 'docker run -d -P vinayakb2010/myaddressbook'
          }
       }
    }

}
```

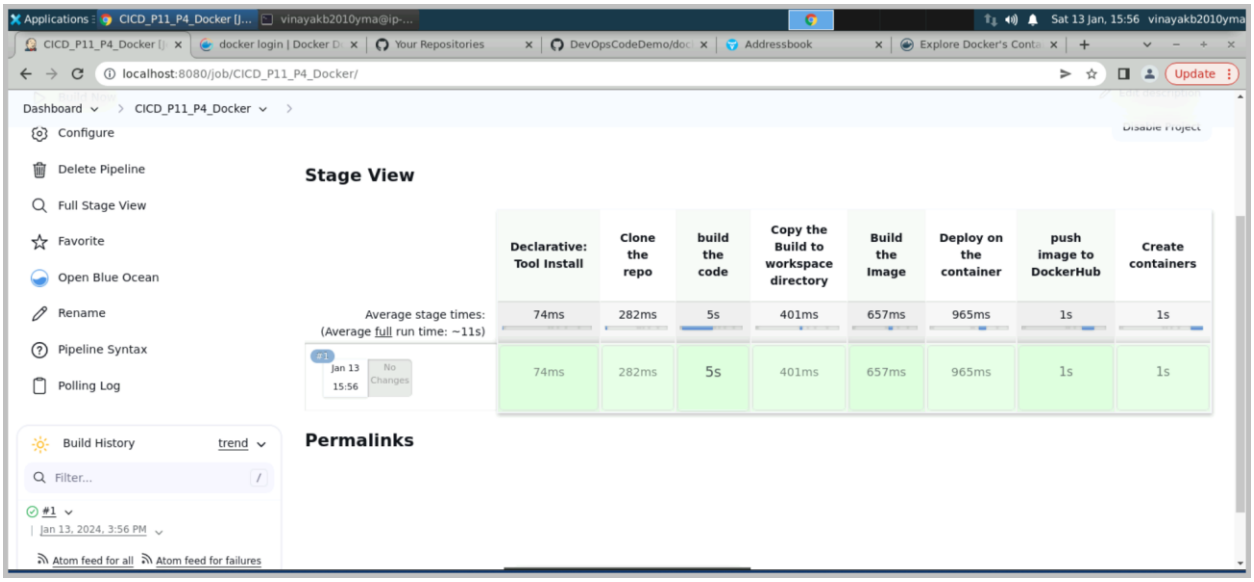


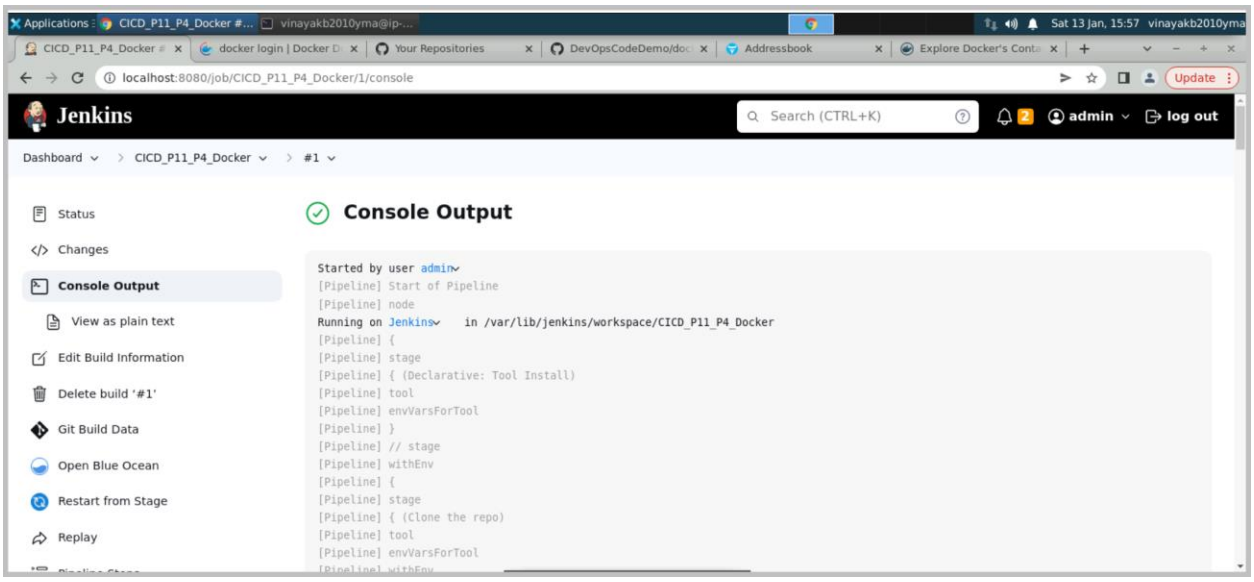
Creating a Pipeline to push the image

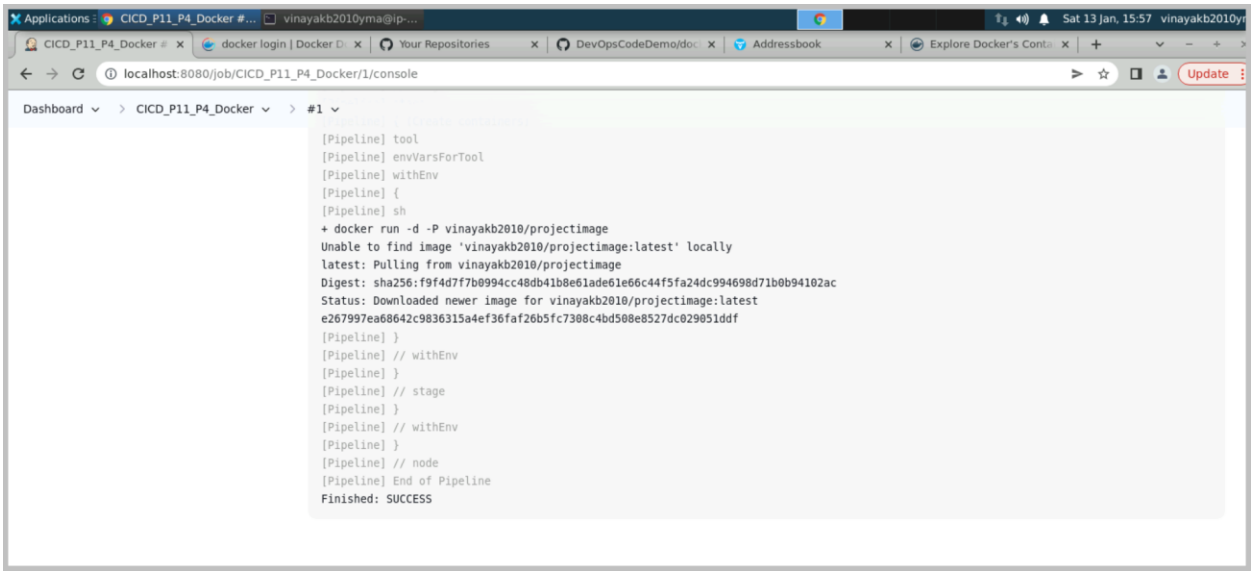


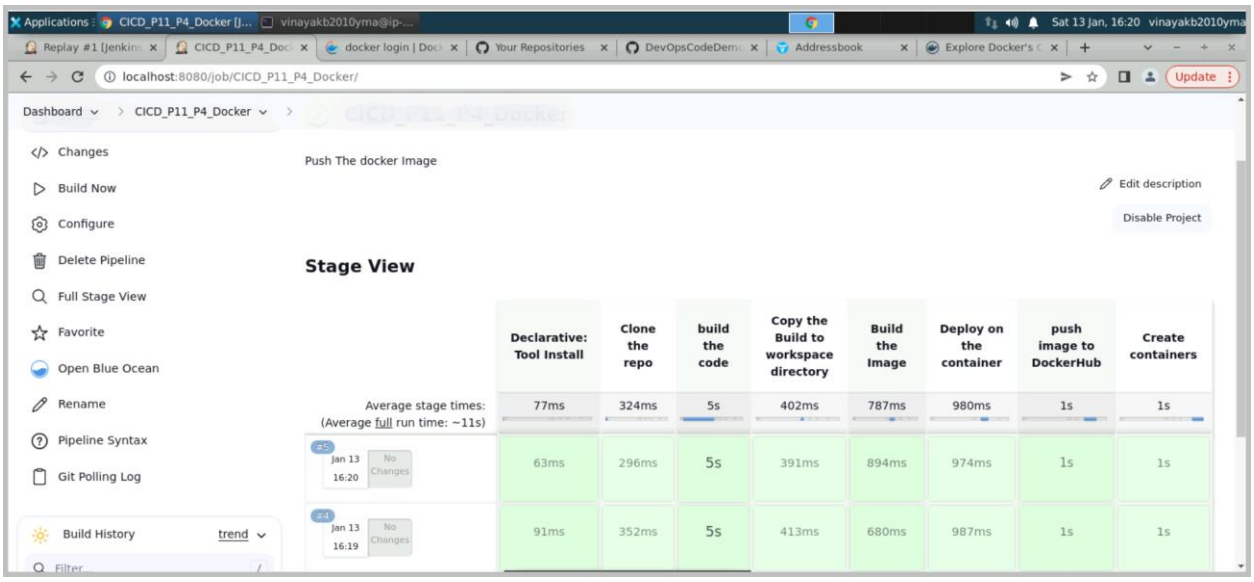


Building it now




Jenkins
Search (CTRL+K)
admin
log out
Dashboard
CICD_P11_P4_Docker
#1
Status
Changes
Console Output
View as plain text
Edit Build Information
Delete build '#1'
Git Build Data
Open Blue Ocean
Restart from Stage
Replay
Console Output
Started by user admin
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/CICD_P11_P4_Docker
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Tool Install)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Clone the repo)
[Pipeline] tool
[Pipeline] envVarsForTool


Dashboard
CICD_P11_P4_Docker
#1
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ docker run -d -P vinayakb2010/projectimage
Unable to find image 'vinayakb2010/projectimage:latest' locally
latest: Pulling from vinayakb2010/projectimage
Digest: sha256:f9f4d7f7b0994cc48db41b8e61ade61e66c44f5fa24dc994698d71b0b94102ac
Status: Downloaded newer image for vinayakb2010/projectimage:latest
e267997ea68642c9836315a4ef36faf26b5fc7308c4bd508e8527dc029051ddf
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS



Final Code That was used to build the above Pipelines and Images

```
pipeline{
   tools{ maven 'VB_MAVEN' }

   agent any

   stages{
      stage('Clone the repo')
                     {
         steps{ git 'https://github.com/Sonal0409/DevOpsCodeDemo.git' }
             }

      stage('build the code'){
        steps{
            sh 'mvn package'
         }
                     }
```

```
stage ('Copy the Build to workspace directory') {

            steps {

//Copy the file from target folder to workspace folder where Docker File exists
//In the below command end, "." means in the current directory

            sh 'cp /var/lib/jenkins/workspace/CICD_P10/target/addressbook.war .'
                        }
                                                                }



stage ('Build the Image') {
            steps {
                                    //Image name will not have the upper case letters
                        sh 'docker build -t projectimage .'
                        }
                                    }

stage ('Deploy on the container'){

            steps {
                        sh 'docker run -d -P projectimage'
                        }

}

 stage('push image to DockerHub'){
```

```
        steps{

            sh 'docker login -u vinayakb2010 -p Vinda*100!'
            sh 'docker tag projectimage vinayakb2010/projectimage'
            sh 'docker push vinayakb2010/projectimage'
            sh 'docker rmi vinayakb2010/projectimage'
        }
    }

    stage('Create containers'){
        steps{

            sh 'docker run -d -P vinayakb2010/projectimage'
        }
    }
  }

}
```