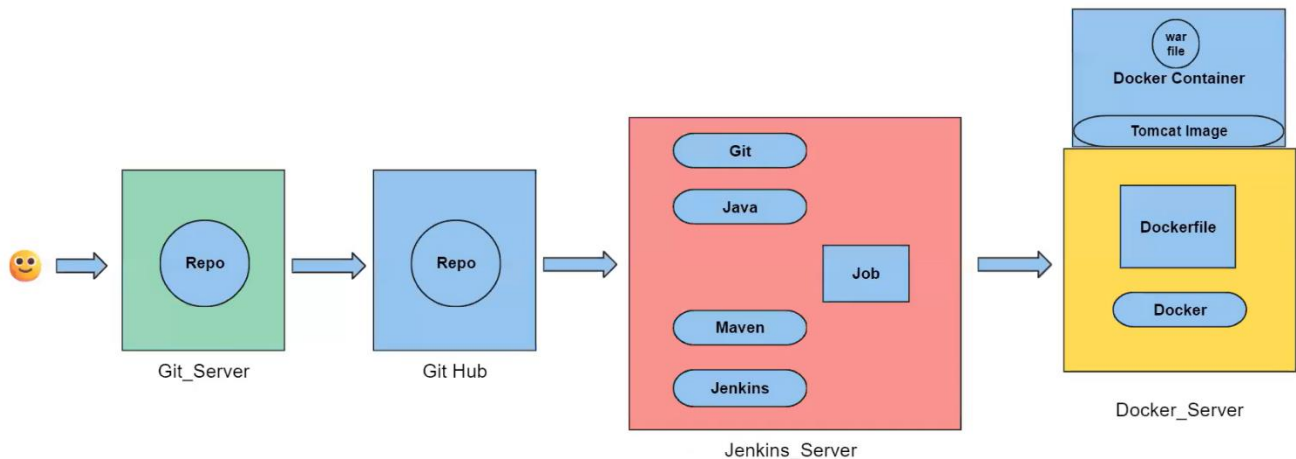


DOCKER DEPLOYMENT

CI

Project Overview

The document outlines a CI/CD pipeline using Git, Jenkins, Docker, and AWS. It begins with setting up a Git server to host the code, creating a Dockerfile, and configuring a Jenkins pipeline to build, test, and push Docker images to Docker Hub. An EC2 instance is set up as a Docker server to pull and run the images. Automation is achieved through SCM polling and linking CI and CD jobs, ensuring seamless integration and deployment. The final application is accessible via the Docker server's public IP.



Technologies Used

- Git - For version control for tracking changes in the code files
- Maven – For Continuous Build
- Jenkins - For continuous integration and continuous deployment
- Docker - containerization

Git-server

Launch git server for developer

Install git

Create passwordless ssh connection

Clone the Forfillapp repository

Inside the Dockerfile

Paste the following commands

```
FROM tomcat:latest
```

```
RUN cp -R /usr/local/tomcat/webapps.dist/* /usr/local/tomcat/webapps/
```

```
COPY webapp/target/*.war /usr/local/tomcat/webapps/
```

Push the code to the central repository (github)

Jenkins server

Install git

Download and Install maven

Install java-17*

Install docker → `yum install docker -y`

Configure java and maven path in `.bash_profile`

Start the Jenkins and docker service

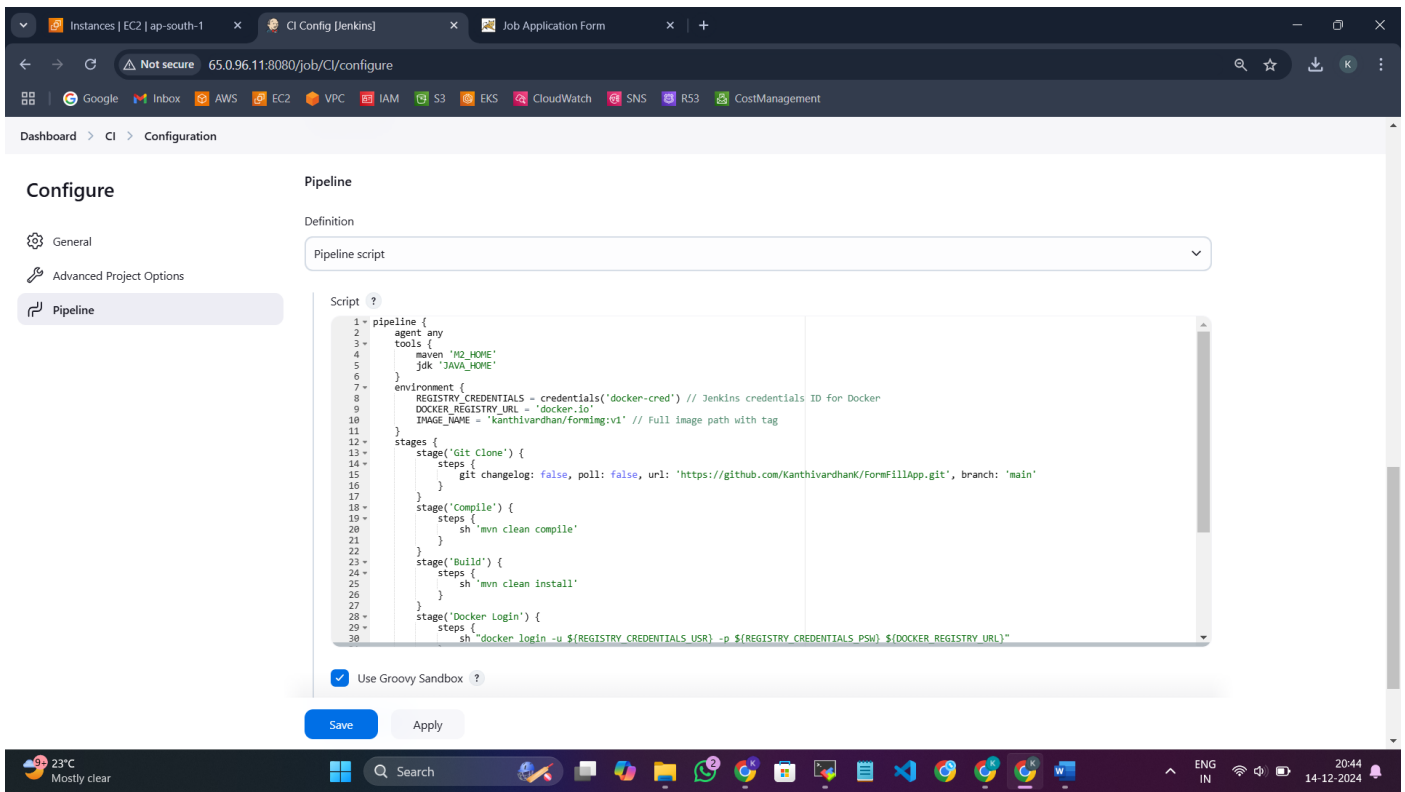
Connect to Jenkins dashboard (Jenkins public ip : 8080)

Install plugins maven invoker, maven integration, pipeline, pipeline:stageview, docker in Manage Jenkins > Plugins

Create a pipeline job name CI

Inside pipeline script add the following script

```
pipeline {
  agent any
  tools {
    maven 'M2_HOME'
    jdk 'JAVA_HOME'
  }
  environment {
    REGISTRY_CREDENTIALS = credentials('docker-cred') // Jenkins credentials ID for Docker
    DOCKER_REGISTRY_URL = 'docker.io'
    IMAGE_NAME = 'kanthivardhan/formimg:v1' // Full image path with tag
  }
  stages {
    stage('Git Clone') {
      steps {
        git changelog: false, poll: false, url: 'https://github.com/KanthivardhanK/FormFillApp.git', branch: 'main'
      }
    }
    stage('Compile') {
      steps {
        sh 'mvn clean compile'
      }
    }
    stage('Build') {
      steps {
        sh 'mvn clean install'
      }
    }
    stage('Docker Login') {
      steps {
        sh "docker login -u ${REGISTRY_CREDENTIALS_USR} -p ${REGISTRY_CREDENTIALS_PSW}
${DOCKER_REGISTRY_URL}"
      }
    }
    stage('Build Docker Image') {
      steps {
        sh "docker build -t ${IMAGE_NAME} ."
      }
    }
    stage('Push Docker Image') {
      steps {
        sh "docker push ${IMAGE_NAME}"
      }
    }
  }
}
```

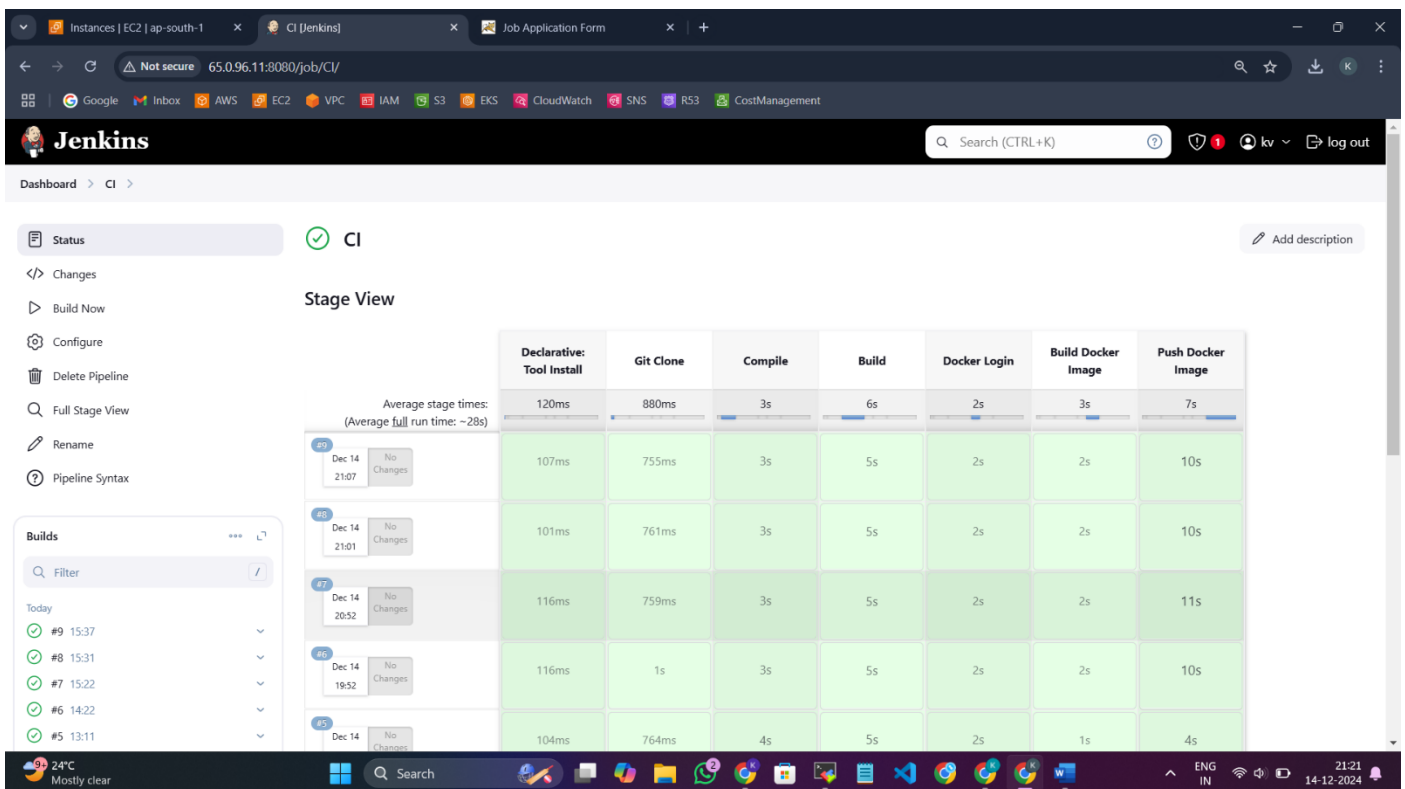


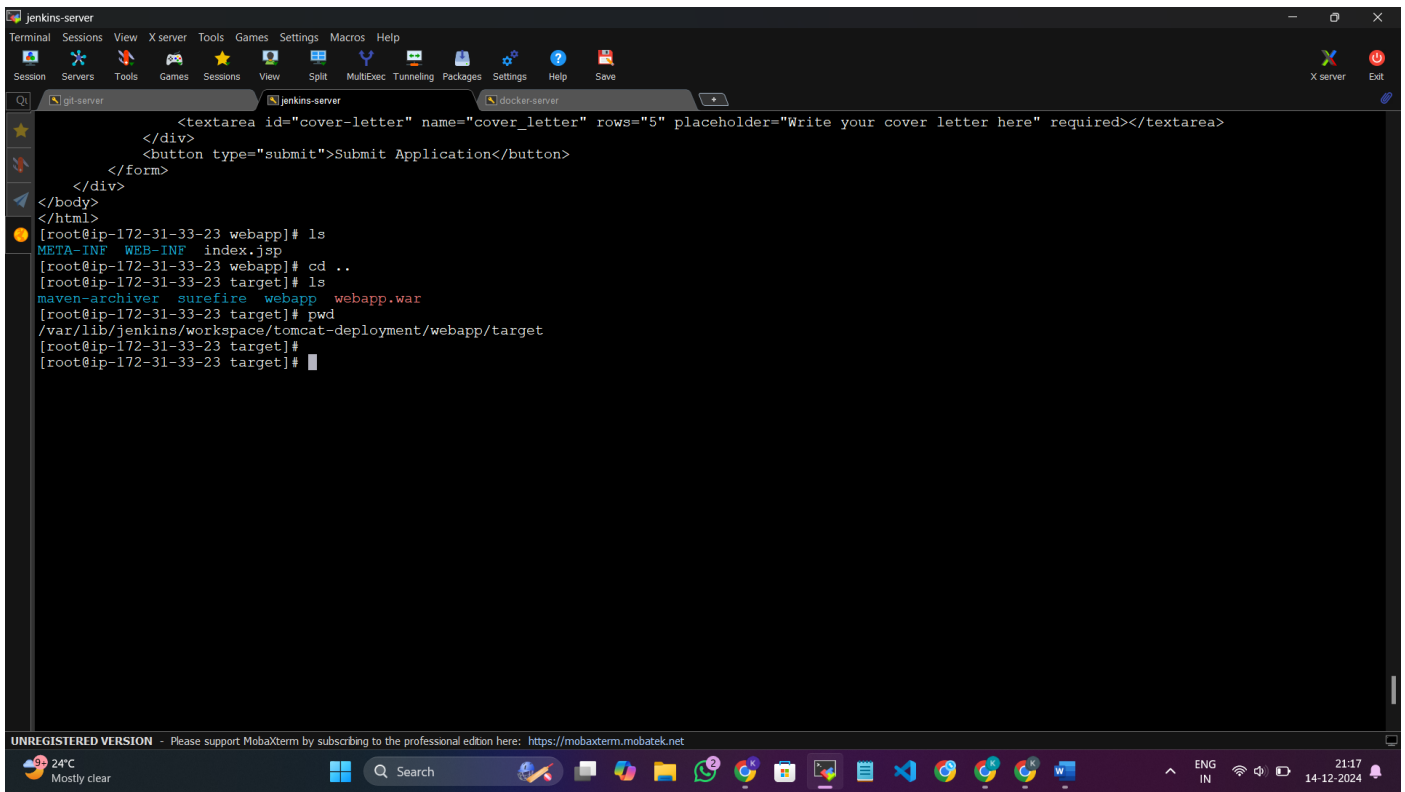
Before configure the java, maven, git paths in Jenkins>manage Jenkins>tools

Add credentials in manage Jenkins>security>global> add username with password, give username and password and id as docker-cred

“sudo usermod -aG docker Jenkins” in jenkins

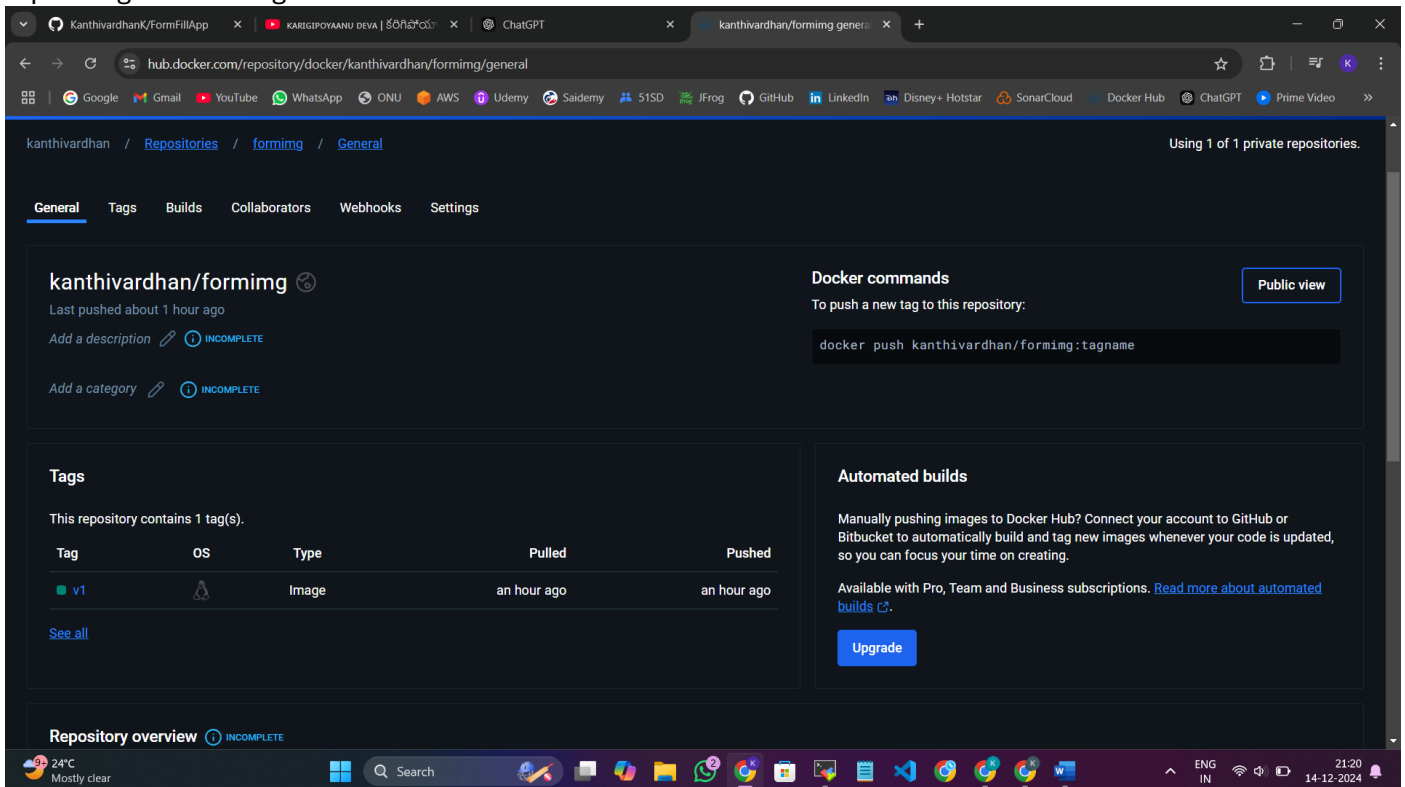
Now build the job





After building the job

- 1-github repository will be cloned
- 2-code will be compiled
- 3-warfile is created
- 4-logging to dockerhub
- 5-building dockerimage
- 6-pushing dockerimage to dockerhub



CD

Launch an ec2 server with following configuration

Name: docker server
Instance type : t2.micro
Security group ports: 80, 22, 8080
Volume: 8 Gi

Connect to instance

Install docker

Yum install docker -y

Service docker start

Vi /etc/ssh/sshd_config

passwordAuthentication yes

permitrootlogin yes

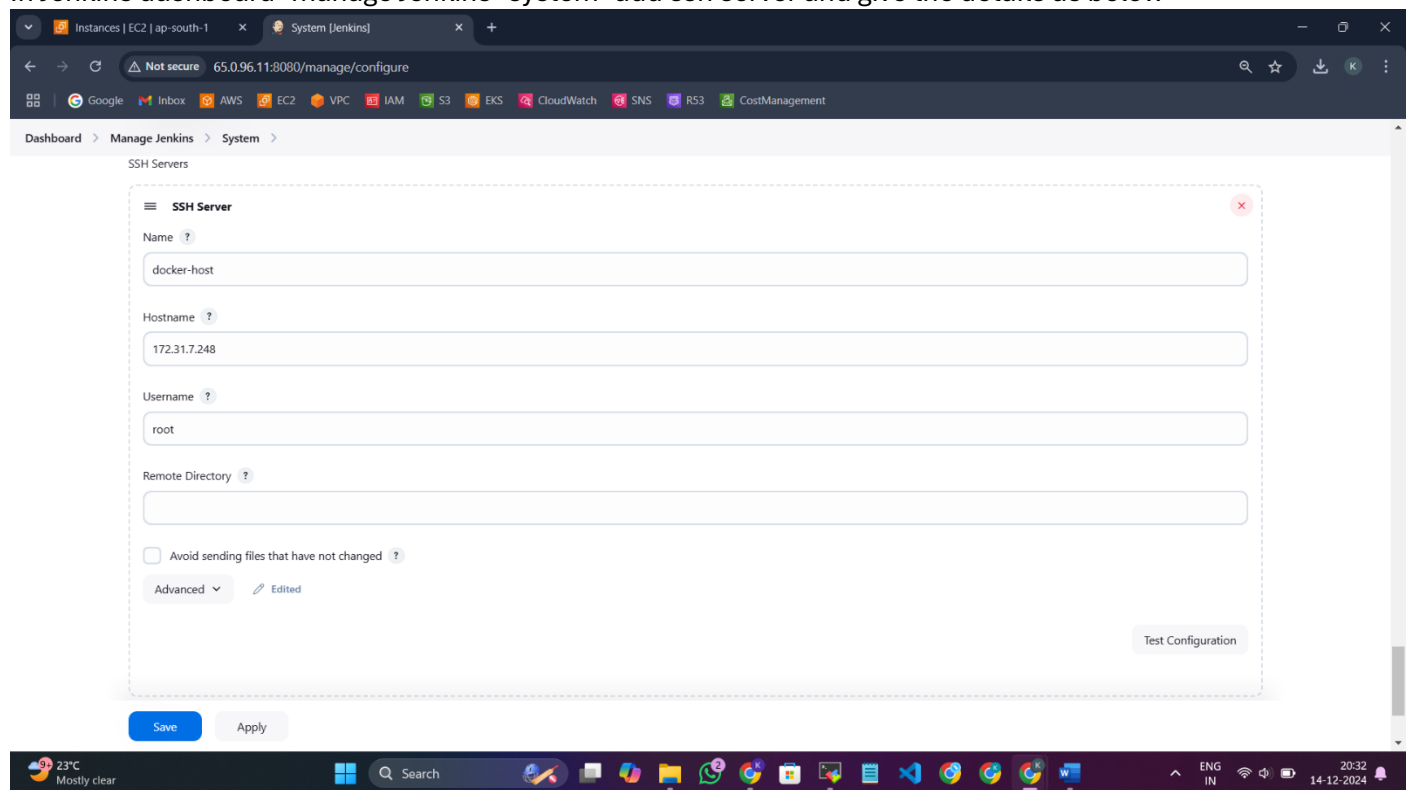
service sshd reload

passwd root

.

Connecting docker server with Jenkins server

In Jenkins dashboard>manage Jenkins>system>add ssh server and give the details as below



The screenshot shows the Jenkins 'SSH Servers' configuration page. The 'Name' field is set to 'docker-host', 'Hostname' is '172.31.7.248', and 'Username' is 'root'. The 'Remote Directory' field is empty. There is a checkbox for 'Avoid sending files that have not changed' which is unchecked. At the bottom, there are 'Save' and 'Apply' buttons, and a 'Test Configuration' button.

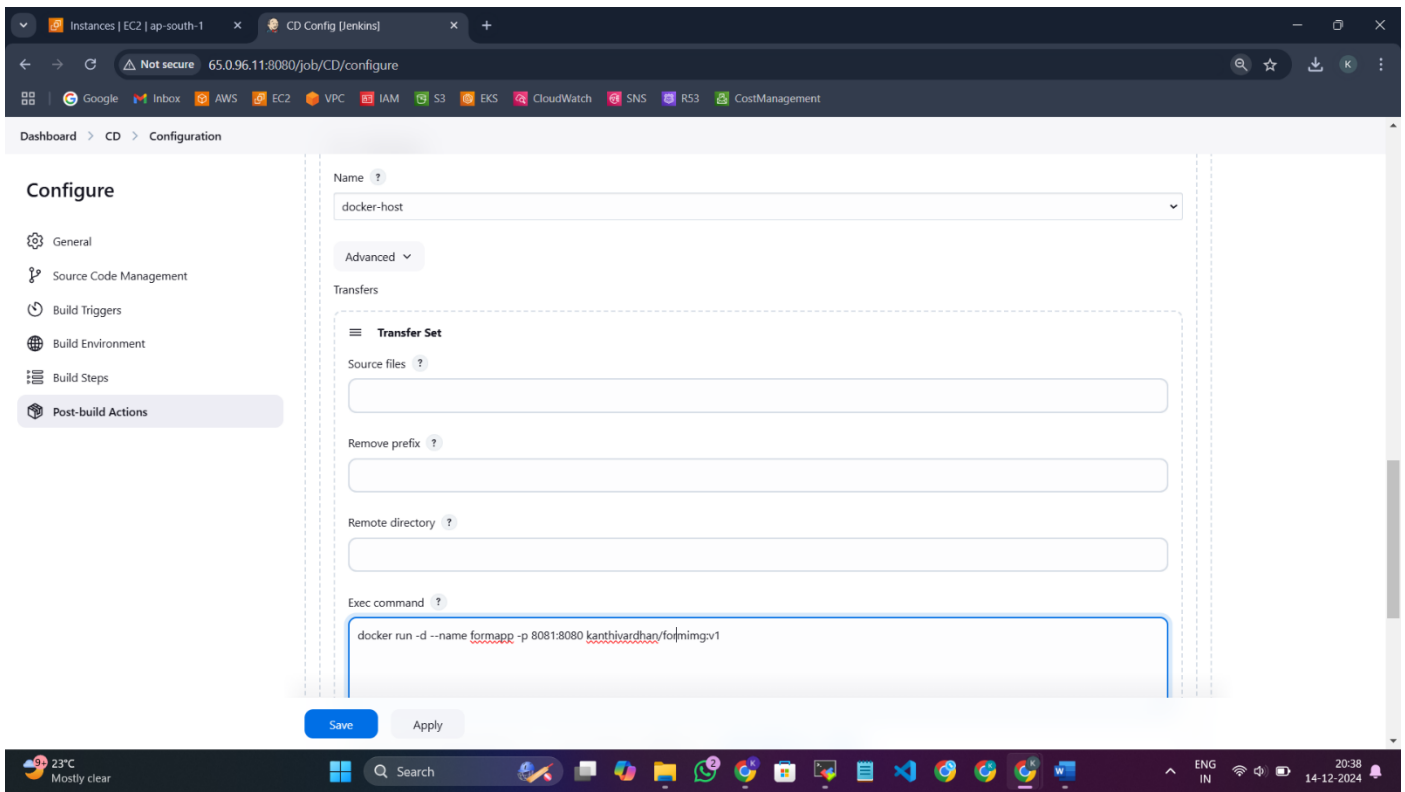
And test configuration

Create a job CD

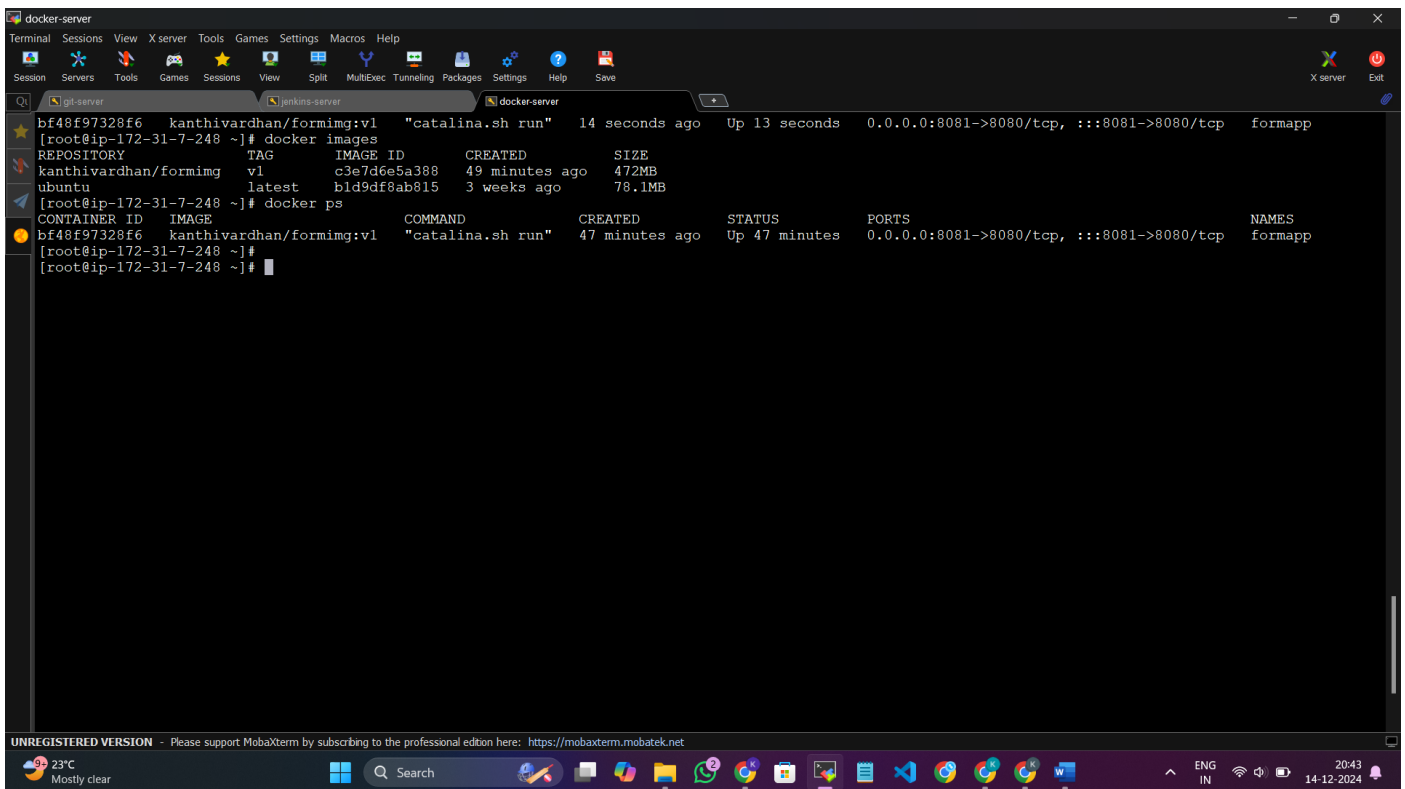
Post build actions> build artifacts over ssh> Name> select docker-host> in Exec command

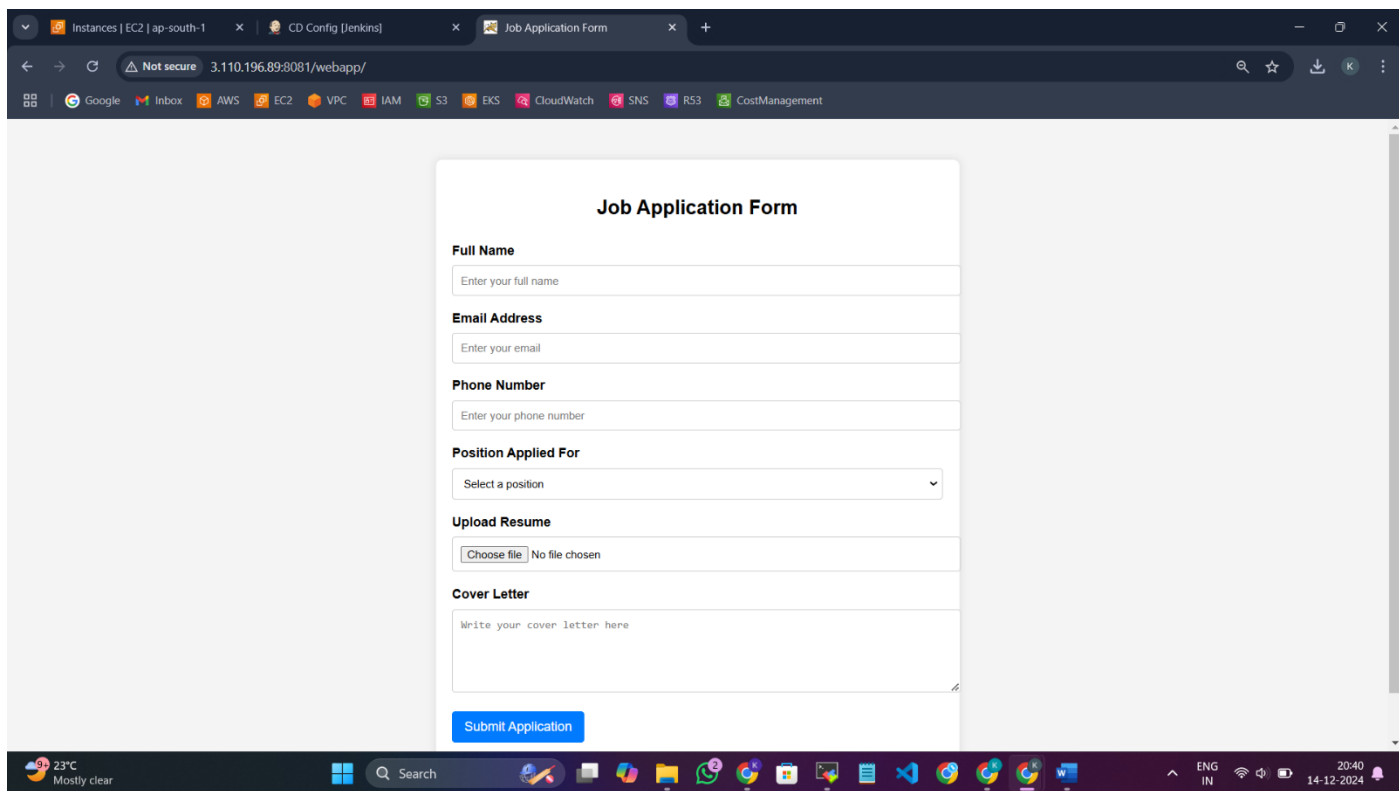
docker run -d --name formapp -p 8081:8080 kanthivardhan/formimg:v1

and save



run the job
when job is triggered image will be pulled from docker hub
container is created



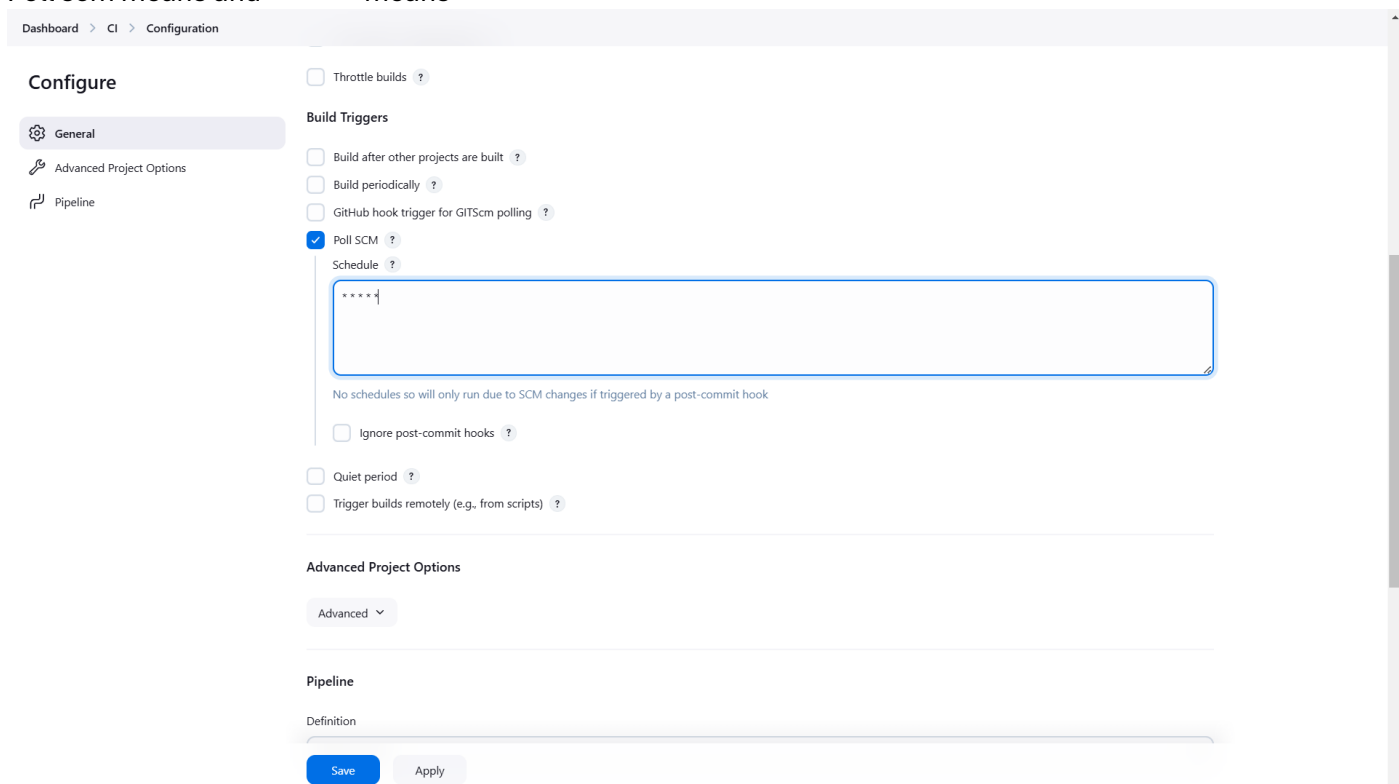


Automating the CI/CD In CI job

Select poll scm

In the box provided give * * * * *

Poll scm means and * * * * * means



The screenshot shows the Jenkins Dashboard in a web browser. The top navigation bar includes links to 'Instances | EC2 | ap-south-1', 'Dashboard [Jenkins]', and 'Job Application Form'. The dashboard itself has a sidebar with links like 'New Item', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', and 'My Views'. The main area displays a table of build history with columns for status (S), warning (W), name (Name), last success, last failure, and last duration. Below the table, there are sections for 'Build Queue' (showing 'No builds in the queue') and 'Build Executor Status' (showing '0/2'). The bottom status bar indicates 'REST API' and 'Jenkins 2.479.2'.

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	CD	49 min #2	N/A	15 sec
✓	☁	CI	52 min #6	2 hr 8 min #3	26 sec

Linking the two jobs

In CD> configure> build after projects are built>Projects to watch> CI

The screenshot shows the 'Configure' page for a Jenkins job named 'CD'. The left sidebar has tabs for 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', 'Build Steps', and 'Post-build Actions'. The 'Build Triggers' tab is selected, showing options for 'Trigger builds remotely', 'Build after other projects are built' (checked), and 'Projects to watch' (set to 'CI'). There are also radio buttons for 'Trigger only if build is stable' (selected), 'Trigger even if the build is unstable', 'Trigger even if the build fails', and 'Always trigger, even if the build is aborted'. Below these are checkboxes for 'Build periodically', 'GitHub hook trigger for GITScm polling', and 'Poll SCM'. The 'Build Environment' section at the bottom has checkboxes for 'Use secret text(s) or file(s)' and 'Send files or execute commands over SSH before the build starts'. 'Save' and 'Apply' buttons are at the bottom.

Whenever new code or code into GitHub is added/modified,

1-automatically CI job will be triggered

1a-code will be cloned

1b-build the war file

1c-creating image including the war file

1d-login to docker hub

1e-push the image to docker hub

2-after the CI job is completed immediately CD job will be triggered

2a-docker image will be pulled from docker hub

2b-docker container will be created

3-finally we can access the website at docker-server public ip :8081/webapp

The screenshot shows the Jenkins Dashboard interface. The top navigation bar includes the Jenkins logo, a search bar, and a 'log out' button. The left sidebar contains links to 'New Item', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', and 'My Views'. The main content area displays a table of job execution history with columns for 'S' (Status), 'W' (Webhook), 'Name', 'Last Success', 'Last Failure', and 'Last Duration'. Below the table, there is a 'Build Queue' section showing 'No builds in the queue' and a 'Build Executor Status' section showing '1/2' (Push Docker image).

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	CD	5 min 44 sec #5	N/A	16 sec
✓	☀	CI	6 min 19 sec #8	2 hr 31 min #3	25 sec

This screenshot is similar to the one above, showing the Jenkins Dashboard. The 'Build Queue' section now shows 'Build Queue (1)' with 'CD' listed. The 'Build Executor Status' section shows '0/2'. The Windows taskbar at the bottom indicates the system time as 21:07 on 14-12-2024.

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	CD	5 min 52 sec #5	N/A	16 sec
✓	☀	CI	26 sec #9	2 hr 31 min #3	25 sec

Instances | EC2 | ap-south-1Dashboard [Jenkins]Job Application Form

Not secure65.0.96.11:8080

GoogleInboxAWS EC2VPCIAM S3EKSCloudWatchSNSR53CostManagement

Jenkins

Search (CTRL+K)

kvlog out

Dashboard

+ New Item

Build History

Project Relationship

Check File Fingerprint

Manage Jenkins

My Views

All

+

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	CD	44 sec #6	N/A	16 sec
✓	☀	CI	1 min 18 sec #9	2 hr 32 min #3	25 sec

Build Queue

No builds in the queue.

Build Executor Status

CD

#7

REST APIJenkins 2.479.2

24°C Mostly clear

Search

Instances | EC2 | ap-south-1Dashboard [Jenkins]Job Application Form

Not secure3.110.196.89:8081/webapp/

GoogleInboxAWS EC2VPCIAM S3EKSCloudWatchSNSR53CostManagement

Job Application Form

Full Name

Enter your full name

Email Address

Enter your email

Phone Number

Enter your phone number

Position Applied For

Select a position

Upload Resume

Choose fileNo file chosen

Cover Letter

Write your cover letter here

Submit Application