Devops Project 1

Pre-requisite for CI/CD pipeline project

1.Github account

2.Simple java project

3.AWS account

4.Jenkins Server(linux)

5.Maven and Git installation

6.Tomcat Server (Linux)

Objective:

To deploy war file on Amazon EC2 instance (tomcat server) using git and Jenkins.

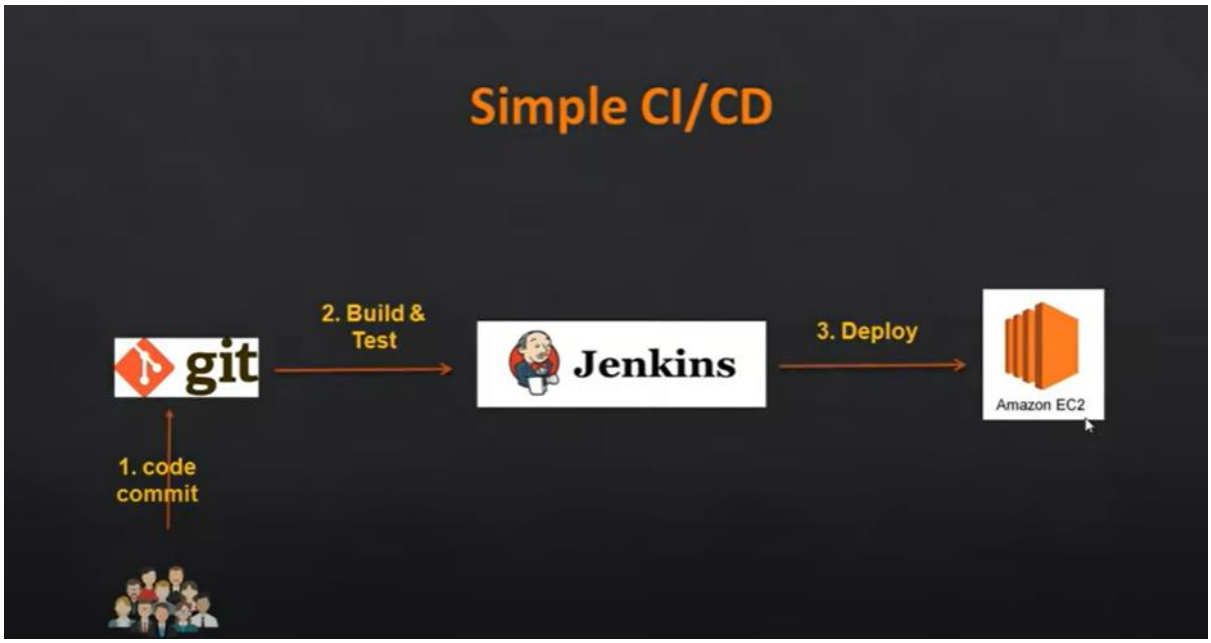
Steps:

1. Developers will commit the changes to the code using git on github account

2. Jenkins will build and test the project for every commit made by developers and create distributable jar or war file.

3. Finally Jenkins will deploy war file on tomcat server residing on ec2 instance.

Project Workflow:



Part-01 : Adding steps for Integration

Steps to create Jenkin job

1. Login to Jenkins console
2. Create *Jenkins job*, Fill the following details,
    - *Source Code Management:*
        - Repository: https://github.com/ValaxyTech/hello-world.git
        - Branches to build : */master
    - *Build:*
        - Root POM:pom.xml
        - Goals and options : clean install package

Part-02: Adding Deployment Steps

in this port we are going to install 'deploy to container' plugin. this is need to deploy on tomcat server which we are using.

- Install maven plugin without restart
    - Manage Jenkins > Jenkins Plugins > available > deploy to container

To deploy our build artifacts on tomcat server our Jenkins server need access. For this we should setup credentials. This option is available in Jenkins home page

- setup credentials
    - credentials > jenkins > Global credentials > add credentials
        - Username : deployer
        - Password : XXXXXXX
        - id : Tomcat_user
        - Description: Tomcat user to deploy on tomcat server

Modify the same job which created in part-01 and add deployment steps.

- Post Steps
    - Deploy war/ear to container
        - WAR/EAR files : **/*.war
        - Containers : Tomcat 8.x
            - Credentials: Tomcat_user (which created in above step)
            - Tomcat URL : http://<PUBLIC_IP>:<PORT_NO>

Save and run the job now.

Part-03 : Continuous Integration & Continuous Deployment (CI/CD)

Now job is running fine but to make this as Continuous Integration and Continuous Deployment Tod do that go back and modify job as below.

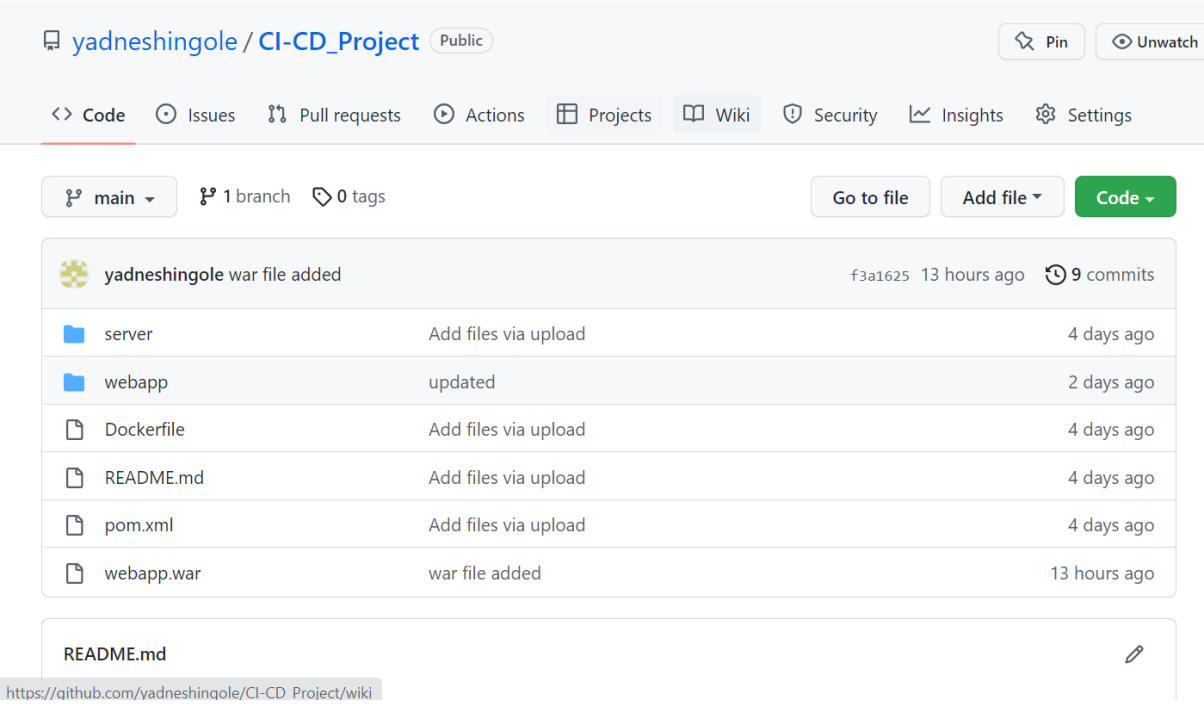
- Build Triggers
    - Poll SCM
        - schedule */2 * * * *

Save the job and modify the code in GitHub. Then you could see your job get trigger a build without any manual intervention.

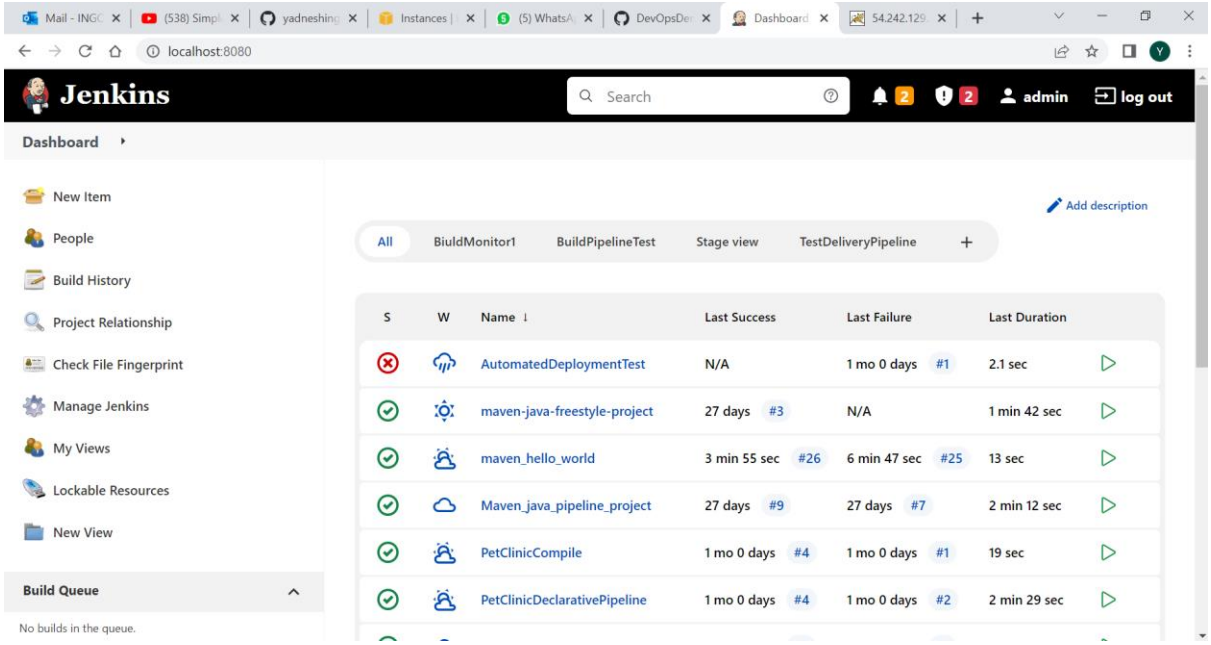
Flow :

Step1: Make a simple maven project and push it on github account

- Pom.xml file is used to build maven project

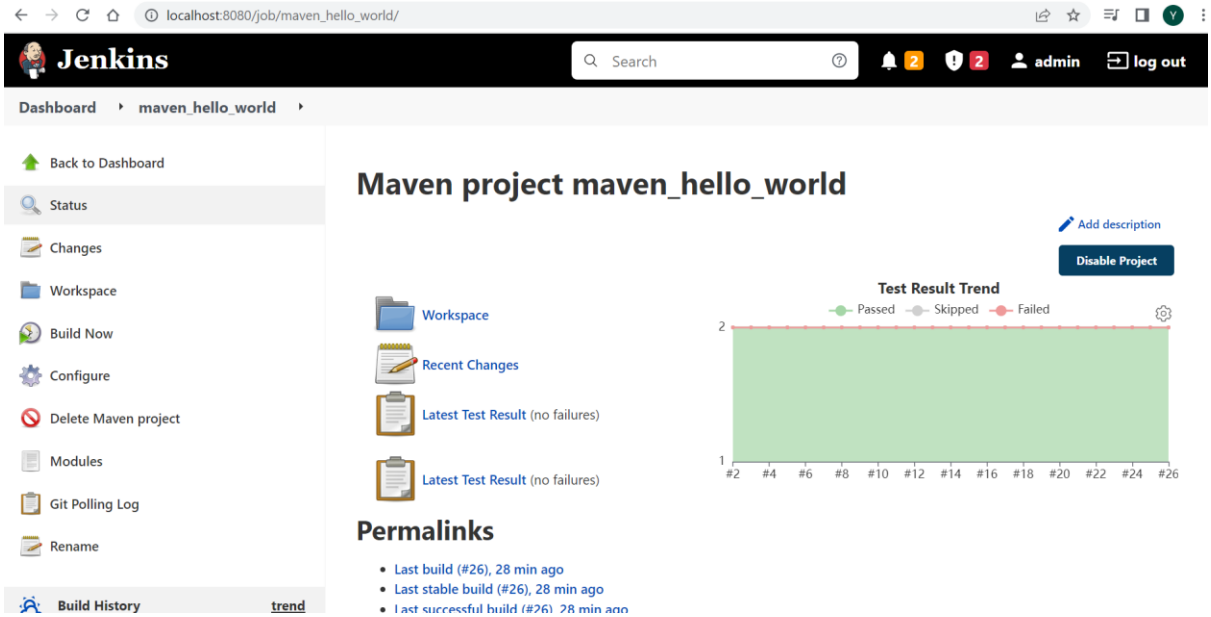


Step 2: Start Jenkins Server and access it on localhost port 8080

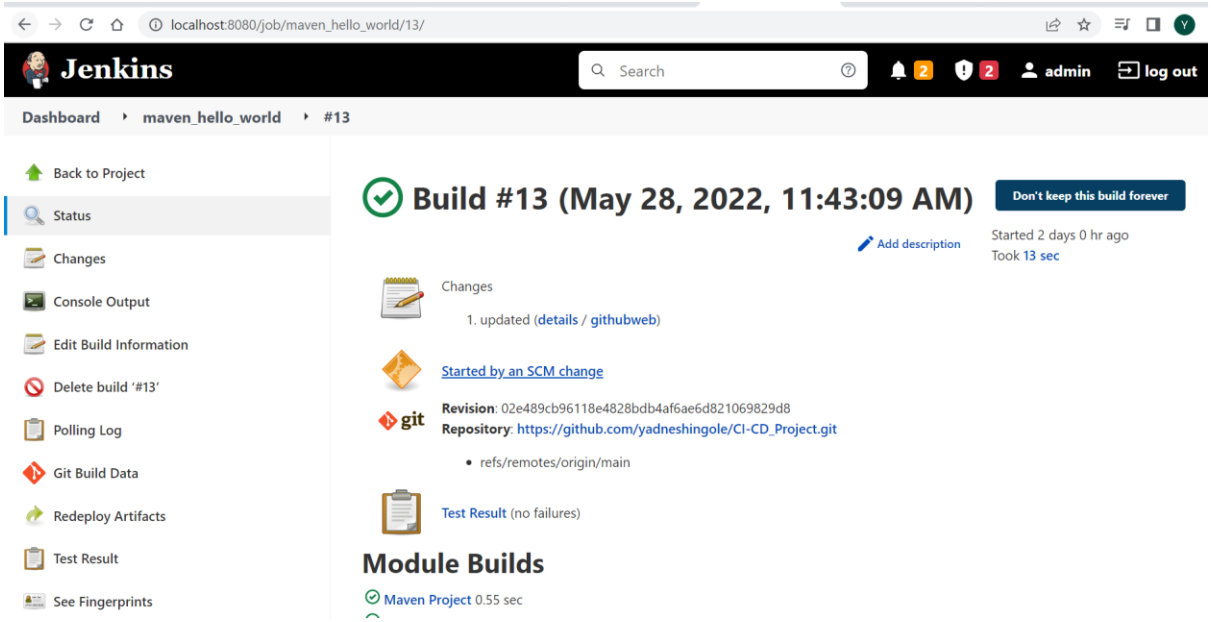


Step 3: Make a simple maven Project on Jenkins

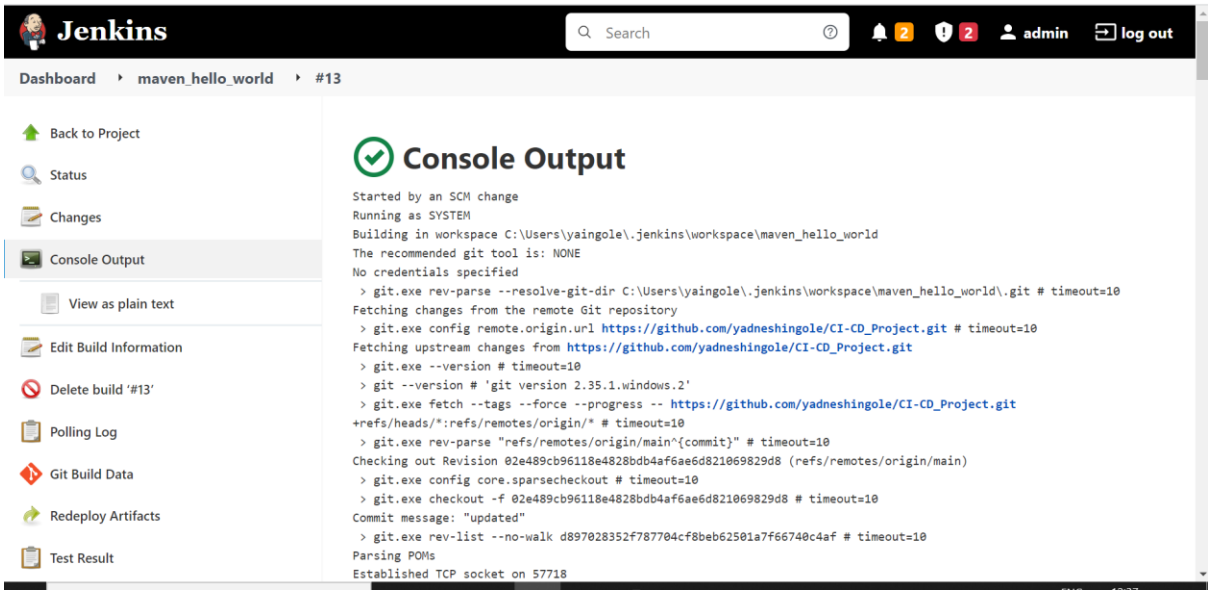
- Go to configure project and add scm of git repository
- Go to build options and add goals such as
- Clean – For Cleaning previous maven project
- install – for installing necessary packages and dependency
- Package – for building jar, war file from java project



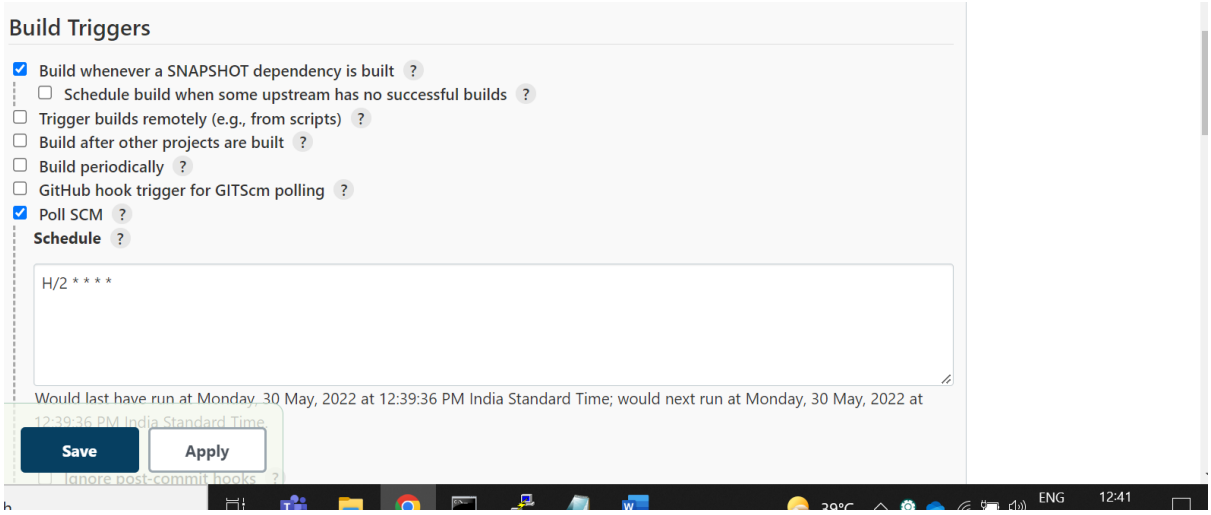
Step 4: Click on save and build the project



Step 5: Click on Console Ouput to see the Output



Step 6: In build trigger section checkbox on poll scm and add H/2 * * * * means it check git repository every 2 minutes if any change is there it automatically build the project again.



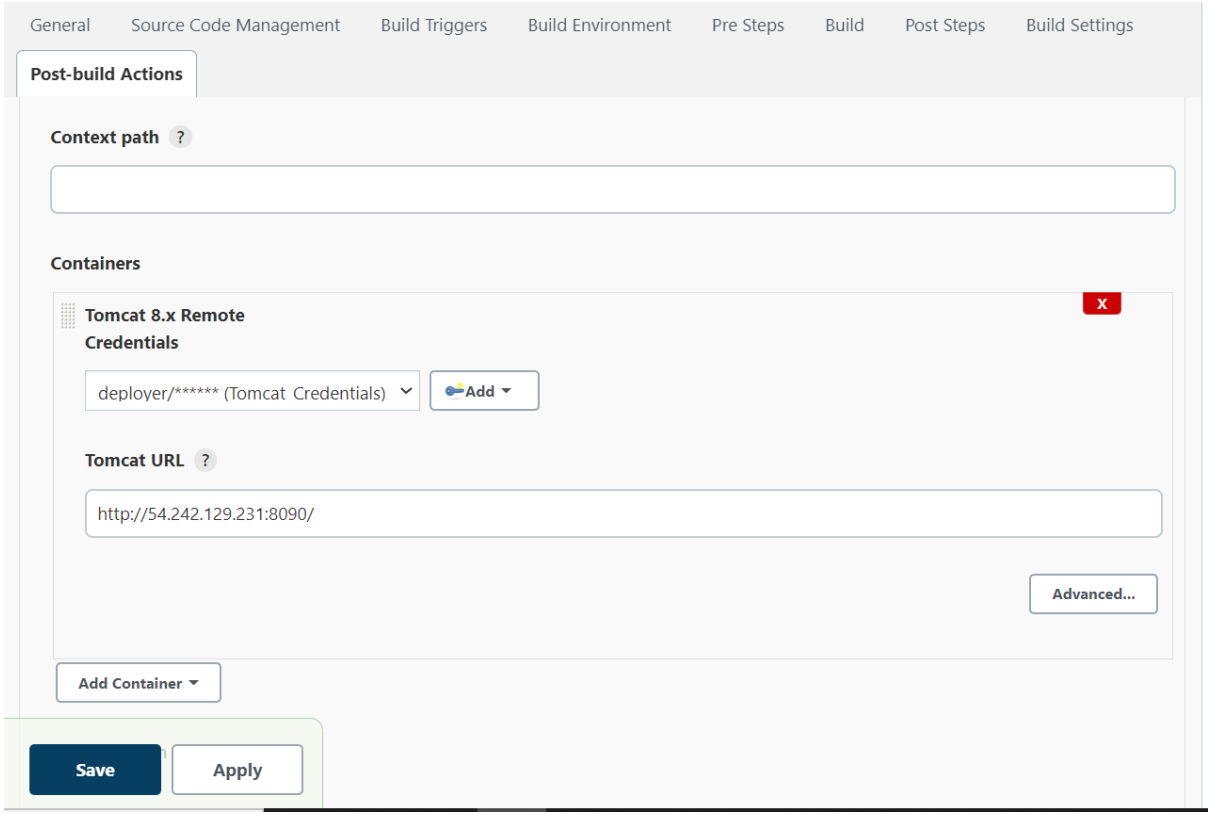
Now Continuous Integration is complete

Step 7: Now Make ec2 instance on aws and install tomcat8 onto it

- Launch as ec2-user with the help of putty

Step 8: Go back to configure project in Jenkins and add post build action

- Enter the url of tomcat server and credentials
- Apply and save the project
- Build the project again



You can access it in web browser

http://54.242.129.231:8090/webapp/

Now we have successfully completed the continuous deployment step.

so whenever push is done or file is updated on github, jenkins uses poll scm to trigger the build and deploy it on tomcat server so that client can see the changes made to application for any new features or bugs simultaneously and make changes accordingly.

we don't need to worry about compiling the code, creating distributable jar and war file, managing the project structure and deploying it on tomcat server.