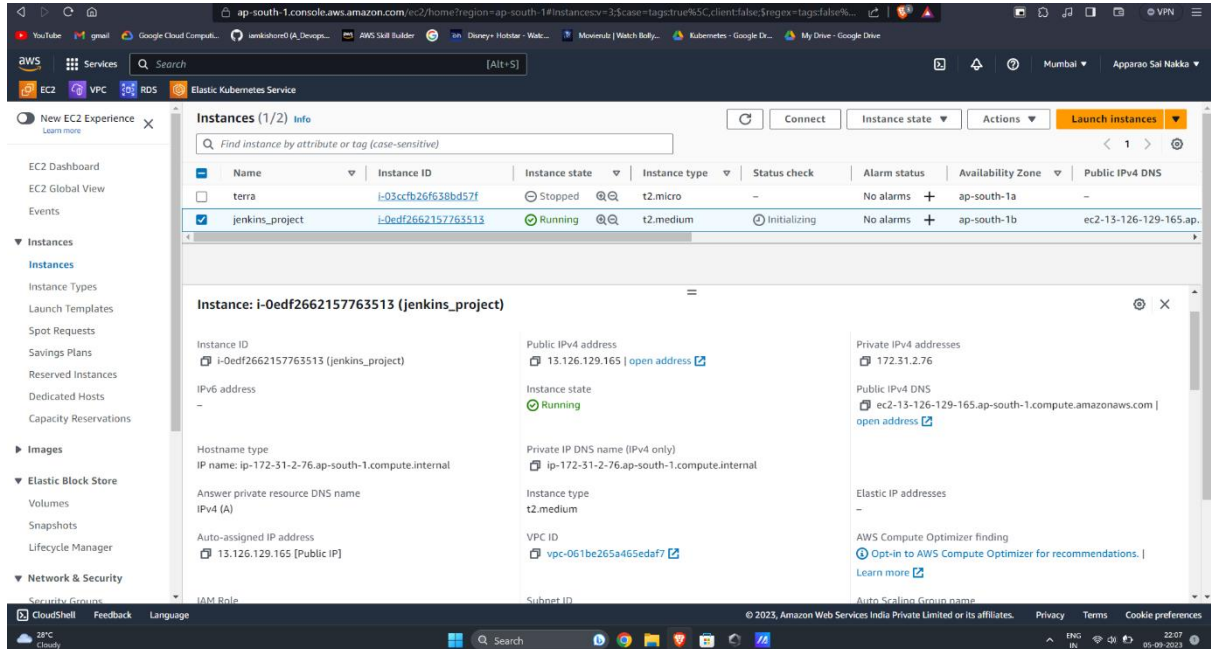


PROJECT 3: Jenkins Multibranch Pipelines and Folders Task

STEPS TO CREATE MULTI BRANCH PIPELINE WITH FOLDER:

1. Create an EC2 instance in AWS console, I create an instance with name “Jenkins_project”.



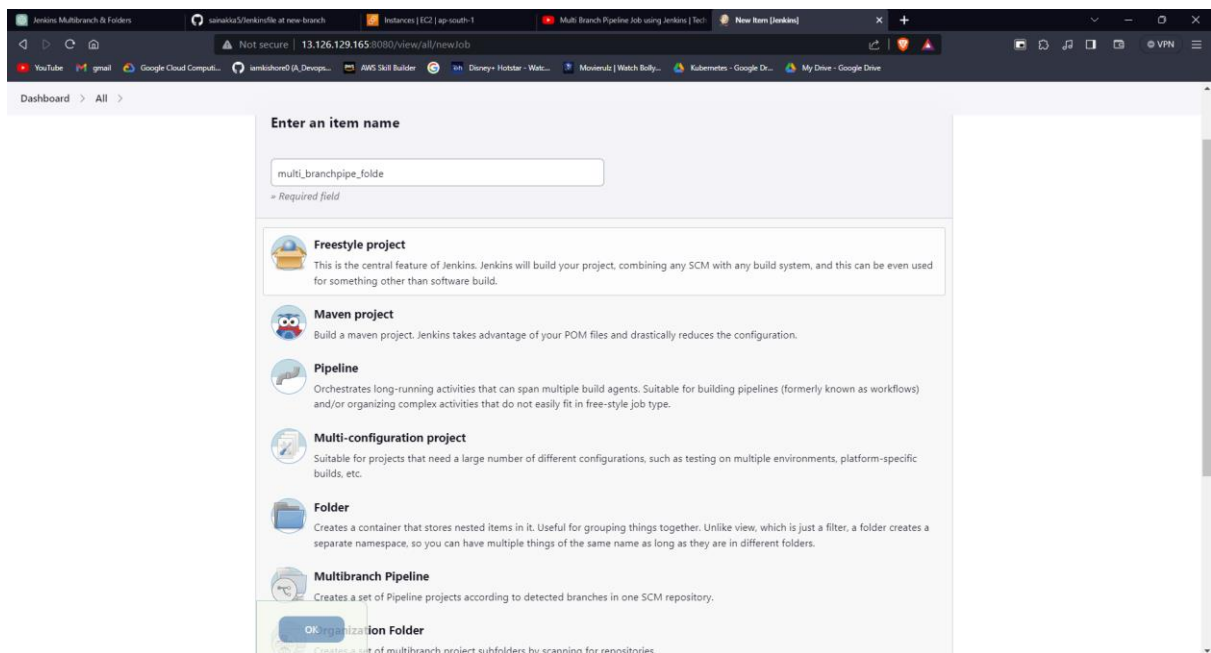
2. Install the Jenkins and its requirements in that instance using Jenkins installation code in Jenkins documentary for centos 7.

```
sudo yum install wget -y
```

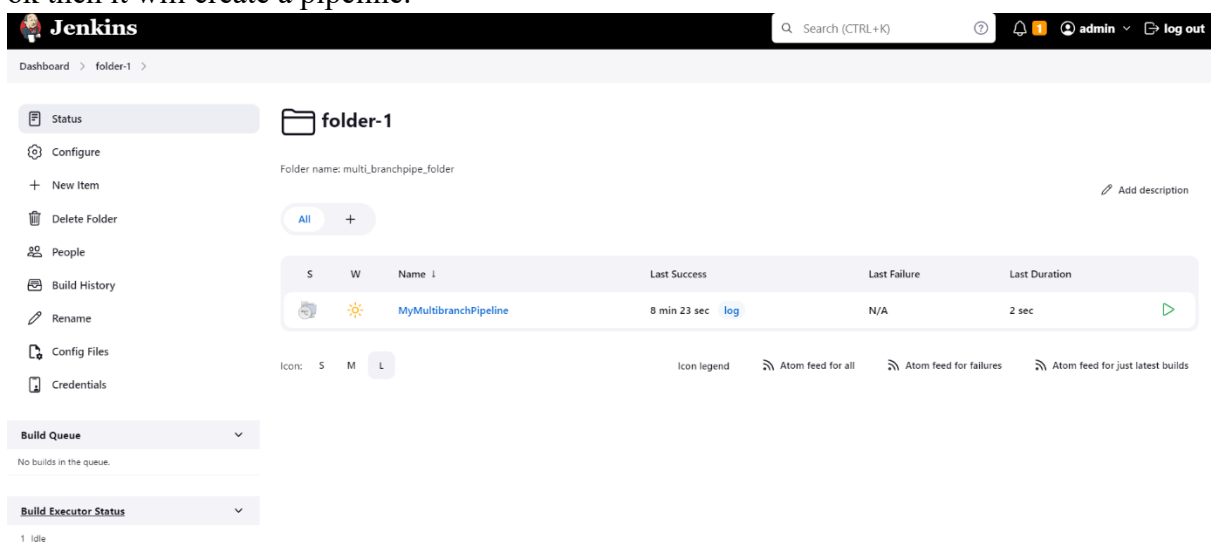
```
sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
```

```
yum install fontconfig java-11-openjdk -y
yum install jenkins -y
service jenkins start
systemctl enable jenkins
service jenkins status
```

3. After execution of the above script, enable the http (8080) port in your Jenkins server security group then login into Jenkins with your ip address like, give your <public_ip:8080> on your browser and give the authentication password, which you can find it in path < /var/jenkins_home/secrets/initialAdminPassword > of Jenkins server give that password and login to the Jenkins webserver.
4. Install the required plugins for our project and create a new folder by clicking on new item and give a name to that folder
click on the folder option as shown in the below fig, then it will create a folder to create multibranch pipeline.

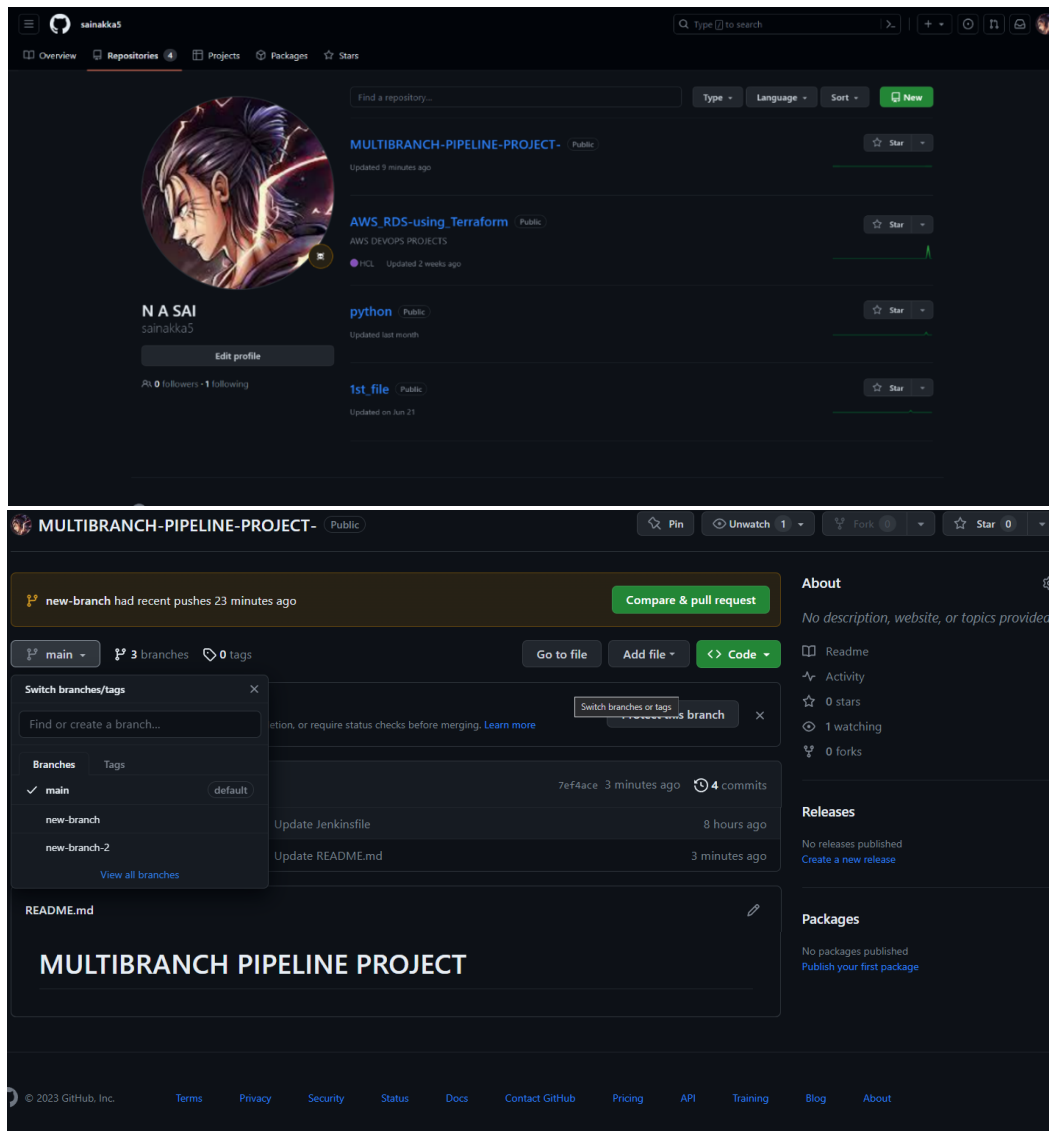


- Click on the folder you created and, in that folder, create a multibranch pipeline, again click on new item and give the name as “Mymultibranchpipeline” click on multibranch pipeline and click ok then it will create a pipeline.

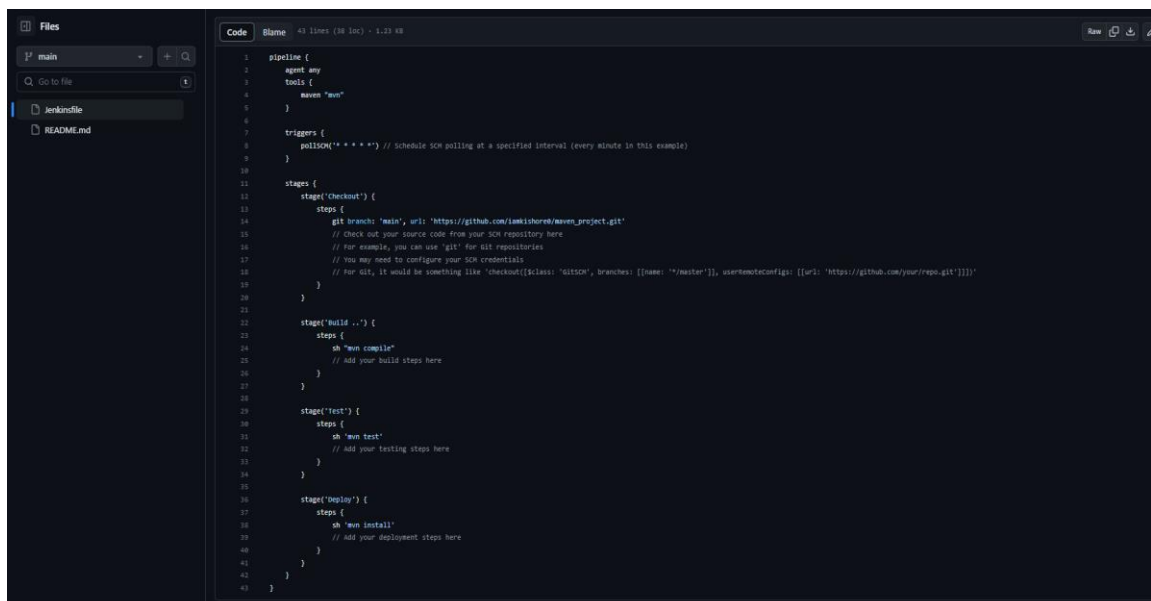


- In that multibranch pipeline go to configuration> branch source > add the git and give the URL of your git repository which includes a code with different branches, to build different pipelines for these branches.
- The code and repository are mentioned in the below figures if it was a private repo give credentials.
- Here I had created 3 branches but 2 branches named main, new-branch are allowed for build pipeline for this task I had add Discover branches, Filter by name (with wildcards) and include your branches there.
- Enable scan webhook option in Scan Multibranch Pipeline Triggers, give the URL of webhook in GitHub repository setting. This option helps to build the pipeline when ever we changed and commit the repository file. Instead of clicking scan option every time just give this option for automatic build of multibranch pipeline. These are done as shown in below figures.

GITHUB : <https://github.com/sainakka5>



10. In the below fig there is a code to replace the webhook tool called poll scm it takes time intervals to build the change in branch.



Dashboard > folder-1 > MyMultibranchPipeline > Configuration

Configuration

- General
- Branch Sources
- Build Configuration
- Scan Multibranch Pipeline Triggers
- Orphaned Item Strategy
- Appearance
- Health metrics
- Properties

Branch Sources

Gilt

Project Repository ?

Credentials ?
- none -
[Add](#)

Behaviors

Discover branches ?

Filter by name (with wildcards)

Include ?

Exclude ?

[Save](#) [Apply](#)

by Jenkinsfile

Script Path ?

Scan Multibranch Pipeline Triggers

☐ Enable Artifactory trigger

☐ Periodically if not otherwise run ?

☒ Scan by webhook ?

Trigger tool: (web hook, webhooks, hook)

Orphaned Item Strategy

Jobs for removed SCM heads (i.e. deleted branches) can be removed immediately or kept based on a desired retention strategy. By default, jobs will be removed as soon as Jenkins determines their associated SCM head no longer exists. As an example, it may be useful to configure a different retention strategy to be able to examine build results of a branch after it has been removed.

☐ Abort builds ?

☒ Discard old items

[Save](#) [Apply](#)

Behaviors

Within Repository

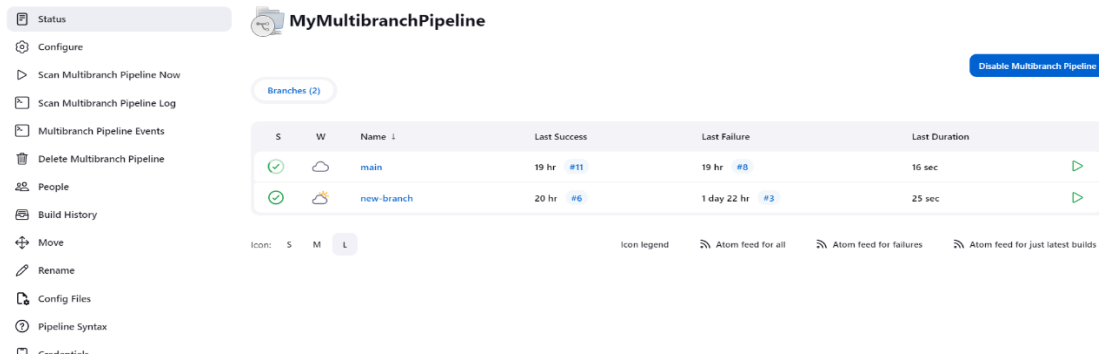
Discover branches ?

Filter by name (with wildcards)

Include ?

Exclude ?

11. After saving all the options in configuration of multibranch pipeline it will scan and build the different pipelines for allowed branches of repository if we change the content and commit the changes and come to the Jenkins server it will build a pipeline for the changed branch. Like the below figure.



- Here I had changed the code in main branch so, the main branch pipeline only get build. It will not touch the remaining branches and pipelines

12. This is the code which builds the pipelines whenever we change the changes in branches, we have to paste the pollSCM code in your Jenkins file.

```
pipeline {
    agent any

    tools {
        maven "mvn"
    }

    triggers {
        pollSCM('* * * * *') // Schedule SCM polling at a specified interval (every minute in this
example)
    }

    stages {
        stage('Checkout') {
            steps {
                git branch: 'main', url: 'https://github.com/iamkishore0/maven_project.git'
                // Check out your source code from your SCM repository here
                // For example, you can use 'git' for Git repositories
                // You may need to configure your SCM credentials

                // For Git, it would be something like 'checkout([$class: 'GitSCM', branches: [[name:
                '*/master']], userRemoteConfigs: [[url: 'https://github.com/your/repo.git']]])'
            }
        }

        stage('Build ') {
```

```
    steps {
        sh "mvn compile"
        // Add your build steps here
    }
}

stage('Test') {
    steps {
        sh 'mvn test'
        // Add your testing steps here
    }
}

stage('Deploy') {
    steps {
        sh 'mvn install'
        // Add your deployment steps here
    }
}
}
```

13. Multibranch pipelines in Jenkins are an efficient way to handle multiple branches within a single pipeline project. They automatically discover branches in a repository and create corresponding Jenkins pipelines. This feature simplifies the management of continuous integration and delivery for various branches.

Advantages of Multibranch Pipelines:

1. **Automated Branch Detection:** Multibranch pipelines automatically discover branches, reducing the need for manual job creation and maintenance.
2. **Isolation of Builds:** Each branch has its own pipeline, ensuring that changes in one branch do not interfere with other branches.
3. **Improved Code Quality:** By running builds and tests on each branch, developers can identify and fix issues early in the development process.
4. **Streamlined Pipeline Configuration:** Multibranch pipelines use a Jenkinsfile in the repository, allowing version-controlled pipeline definitions and making it easier to modify the pipeline behavior.
5. **Regarding folders,** they offer a way to organize jobs in Jenkins, reducing clutter and making it easier to manage multiple jobs or projects. Folders help maintain a more structured and logical view of your Jenkins jobs, especially when dealing with a large number of projects.

GITHUB : <https://github.com/sainakka5>

6. To complete this assignment, students should demonstrate their ability to set up Jenkins, create multibranch pipelines, use Jenkinsfiles to define build stages, and understand the benefits of organizing jobs into folders.