

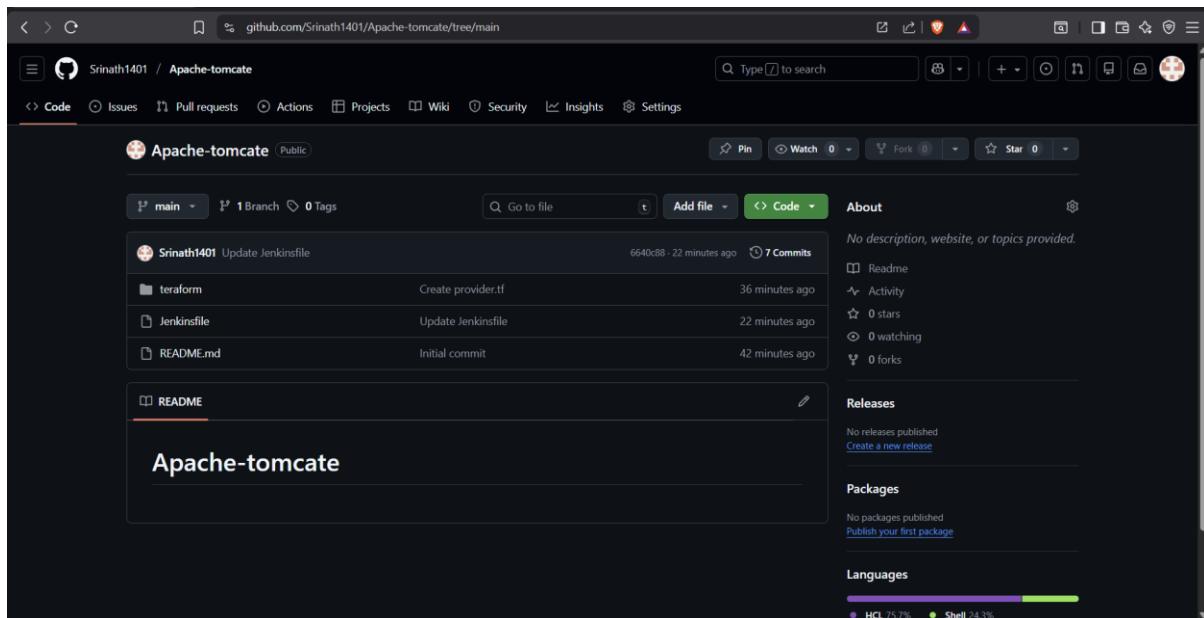
## Task-23

### Java - Tomcat Server Automation

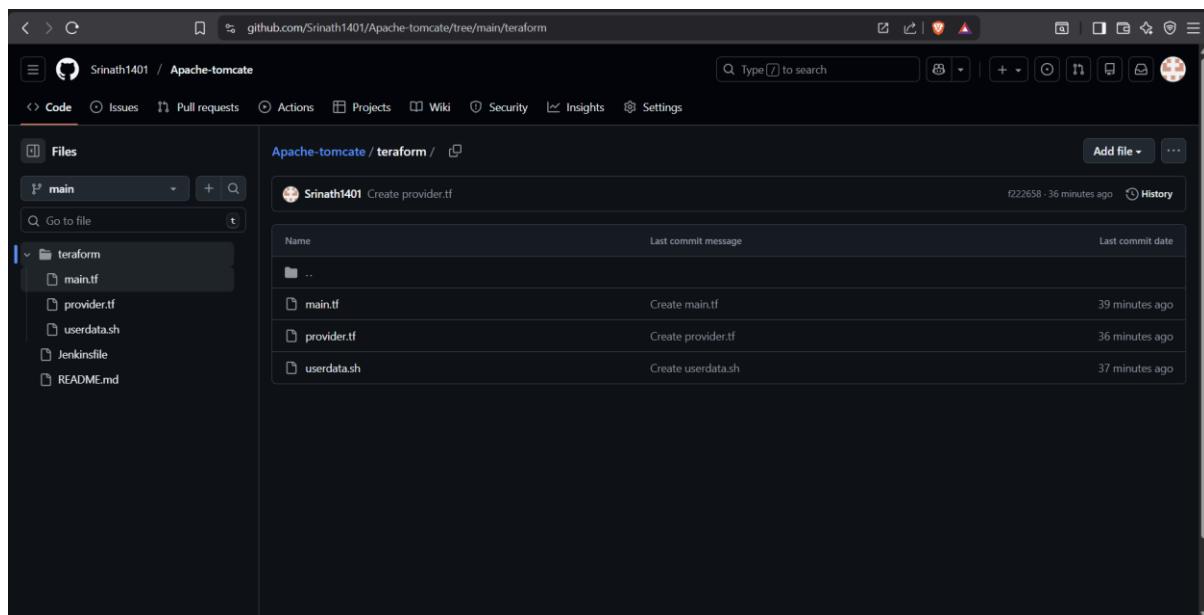
#### Tomcat:

A Tomcat primarily refers to Apache Tomcat, a popular, open-source Java servlet container and web server that runs Java-based web applications, implementing Jakarta EE specs like Servlets, JSP, and WebSockets, allowing developers to build and deploy dynamic web apps

#### Create Terraform folder and file:



The screenshot shows a GitHub repository named 'Apache-tomcate'. The repository has 1 branch and 0 tags. It contains several files: 'Create provider.tf' (36 minutes ago), 'Update Jenkinsfile' (22 minutes ago), 'Jenkinsfile' (22 minutes ago), and 'README.md' (42 minutes ago). The 'About' section indicates there is no description, website, or topics provided. The 'Languages' section shows HCL at 75.7% and Shell at 24.3%. The 'Releases' and 'Packages' sections are currently empty.



The screenshot shows the 'terraform' directory within the 'Apache-tomcate' repository. The 'Files' sidebar shows the structure: 'main' (selected) and 'terraform' (expanded). Inside 'terraform', there are four files: 'main.tf', 'provider.tf', 'userdata.sh', and 'Jenkinsfile'. The 'main.tf' file is selected. The main content area shows the code for 'Create provider.tf':

```
provider "aws" { }
```

## Code:

### Main.tf:

```
resource "aws_security_group" "web_sg" {
  name = "web-sg"

  ingress {
    from_port  = 22
    to_port   = 22
    protocol  = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  ingress {
    from_port  = 80
    to_port   = 80
    protocol  = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  ingress {
    from_port  = 8080
    to_port   = 8080
    protocol  = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  egress {
    from_port  = 0
    to_port   = 0
    protocol  = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
}
resource "aws_instance" "web" {
  ami           = "ami-068c0051b15cdb816" # Ubuntu 22.04 ap-south-1
  instance_type = "t2.micro"
  key_name      = "mykey"
  vpc_security_group_ids = [aws_security_group.web_sg.id]
  user_data = file("userdata.sh")
  tags = {
    Name = "Jenkins-Tomcate-Server"
  }
}
```

### Provider.tf:

```
provider "aws" {
  region = "us-east-1"
```

```
}
```

### Userdata.sh:

```
#!/bin/bash
yum update -y
yum install docker -y
systemctl start docker
systemctl enable docker
usermod -aG docker ec2-user
docker pull tomcat:9.0
docker run -d \
--name tomcat-server \
-p 8080:8080 \
tomcat:9.0
```

### Jenkinsfile:

```
pipeline {
    agent any

    environment {
        AWS_DEFAULT_REGION = "us-east-1"
    }

    stages {
        stage('Checkout Code') {
            steps {
                echo "Checking out code from GitHub..."
                git branch: 'main',
                    url: 'https://github.com/Srinath1401/Apache-tomcate.git'
            }
        }

        stage('Terraform Steps') {
            steps {
                withCredentials([
                    $class: 'AmazonWebServicesCredentialsBinding',
                    credentialsId: 'aws-creds'
                ]) {
                    script {
                        echo "Checking if Terraform is installed..."

                        //  Absolute path check
                        def terraformCheck = bat(
                            script: 'C:\\terraform\\terraform.exe -version',

```

```
        returnStatus: true
    )

    if (terraformCheck != 0) {
        error "Terraform executable not found at C:\\terraform\\terraform.exe"
    }

    echo "Terraform found. Running Init, Plan, Apply..."

    dir('terraform') {
        bat 'C:\\terraform\\terraform.exe init'
        bat 'C:\\terraform\\terraform.exe plan'
        bat 'C:\\terraform\\terraform.exe apply -auto-approve'
    }

    echo "Terraform execution completed successfully."
}

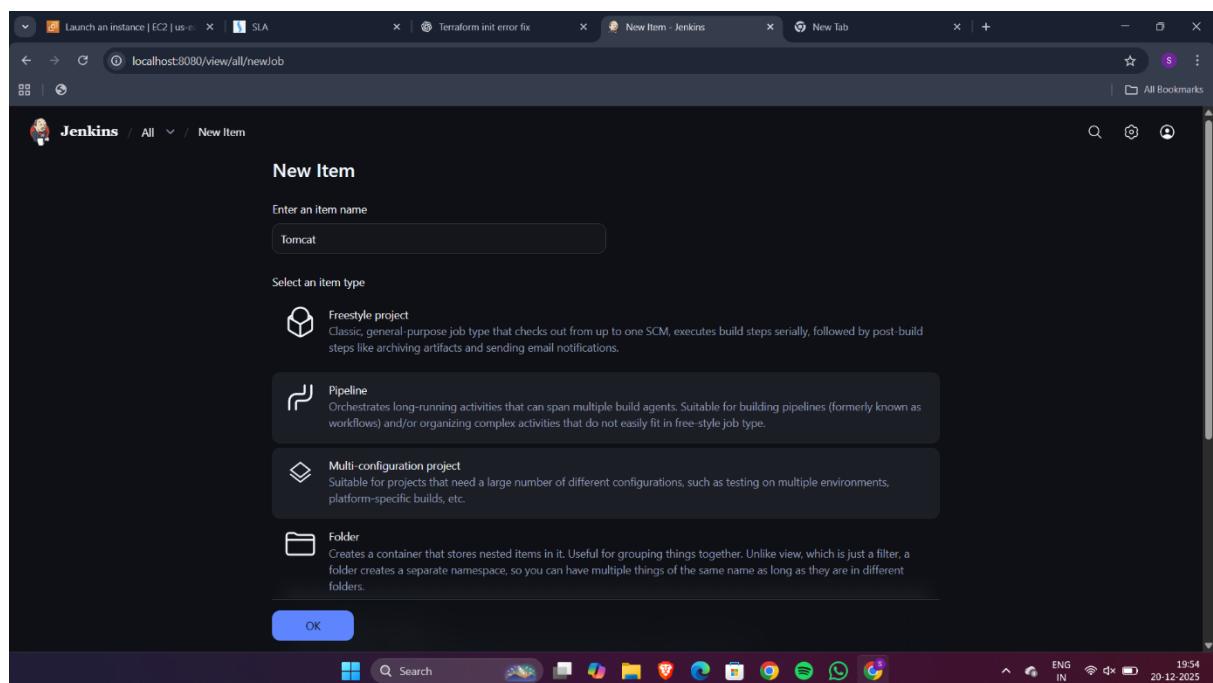
}

}

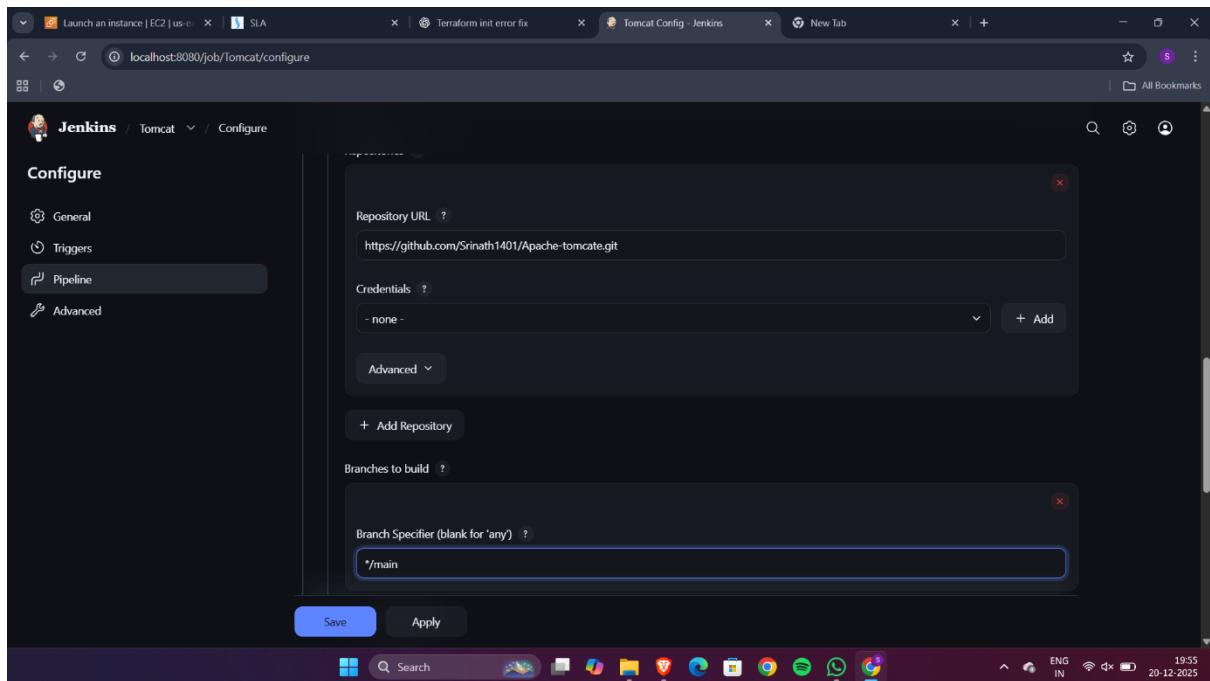
}

}
```

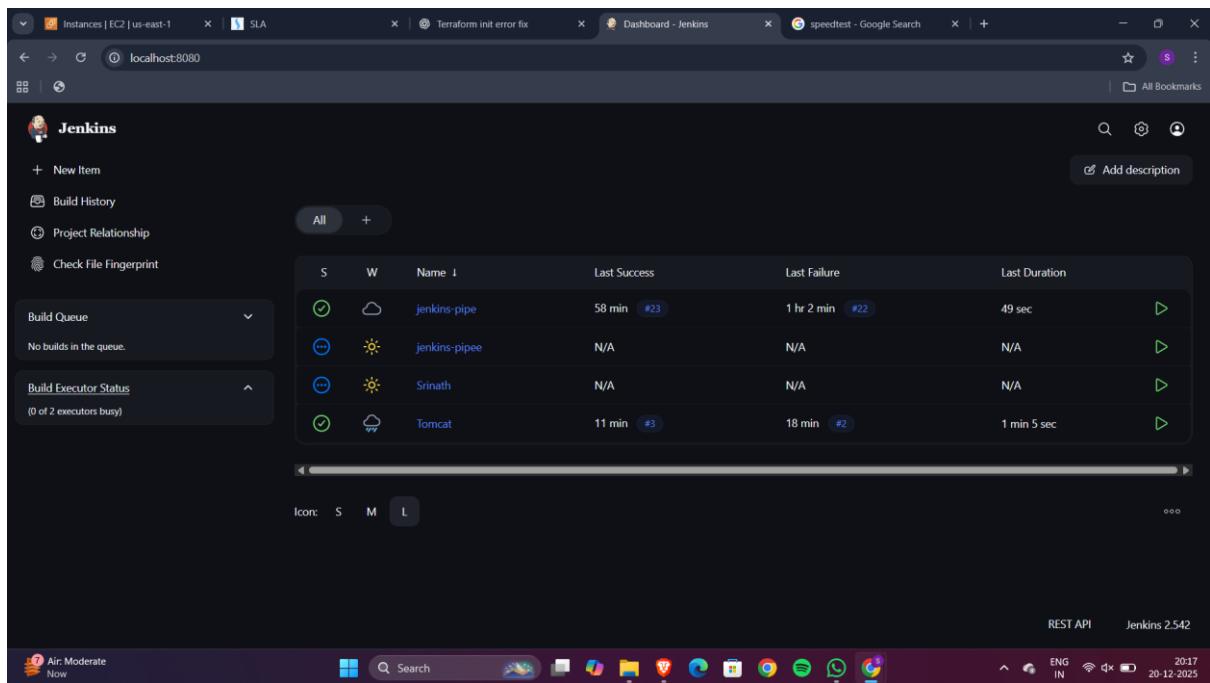
## Create Jenkins Pipeline:



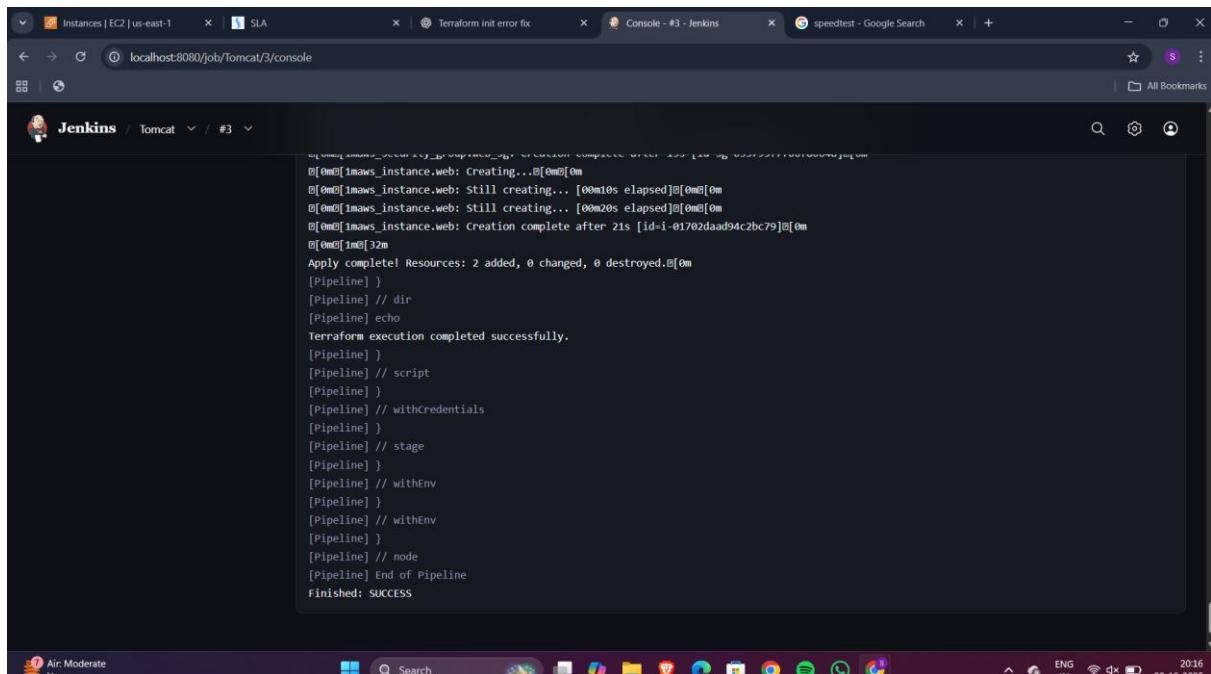
## Configure pipeline:



## Build the Jenkins:

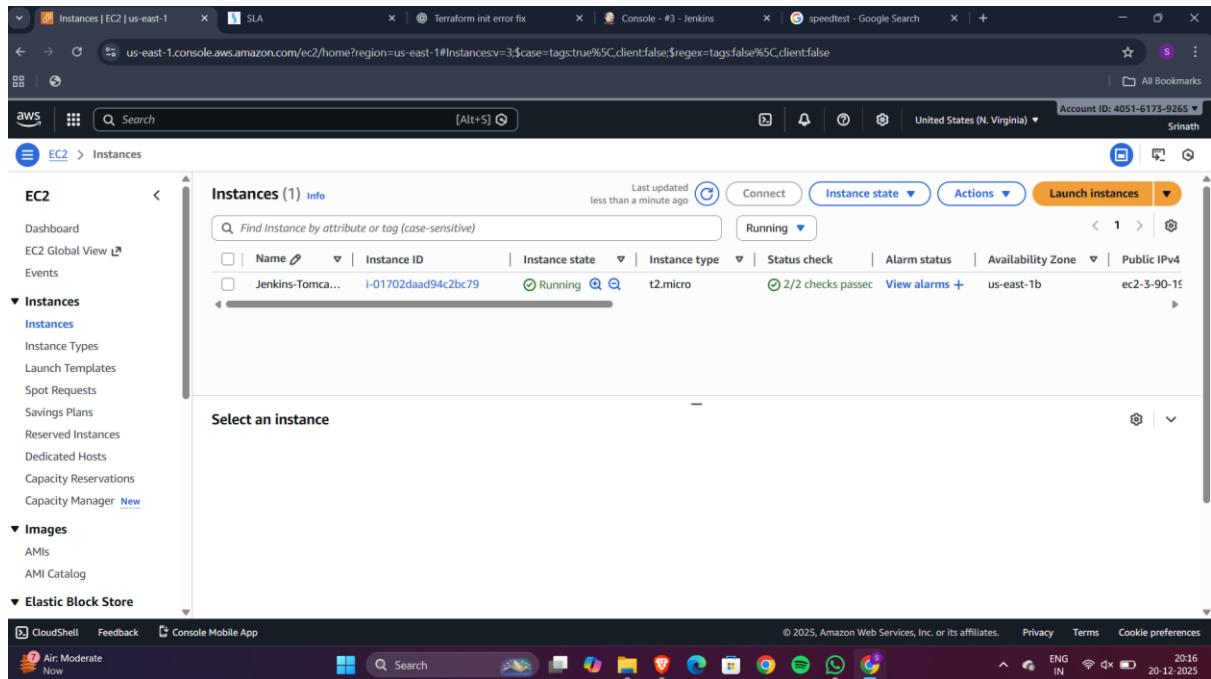


## Build Success:



```
[0m[1maws_instance.web: creating...[0m[0m
[0m[1maws_instance.web: Still creating... [00m10s elapsed][0m[0m
[0m[1maws_instance.web: Still creating... [00m20s elapsed][0m[0m
[0m[1maws_instance.web: creation complete after 21s [id=1-01702daad94c2bc79][0m
[0m[1m[32m
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.[0m
[Pipeline]
[Pipeline] // dir
[Pipeline] echo
Terraform execution completed successfully.
[Pipeline]
[Pipeline] // script
[Pipeline]
[Pipeline] // withCredentials
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // withEnv
[Pipeline]
[Pipeline] // withEnv
[Pipeline]
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

## OUTPUT:



The screenshot shows the AWS EC2 Instances page. The left sidebar has a tree view with 'EC2' selected, showing 'Instances (1)'. The main table displays one instance:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
Jenkins-Tomcat...	i-01702daad94c2bc79	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	ec2-3-90-15

## **Output of Tomcat:**

