

Task-22

SCP and Jenkins CI/CD for Ec2

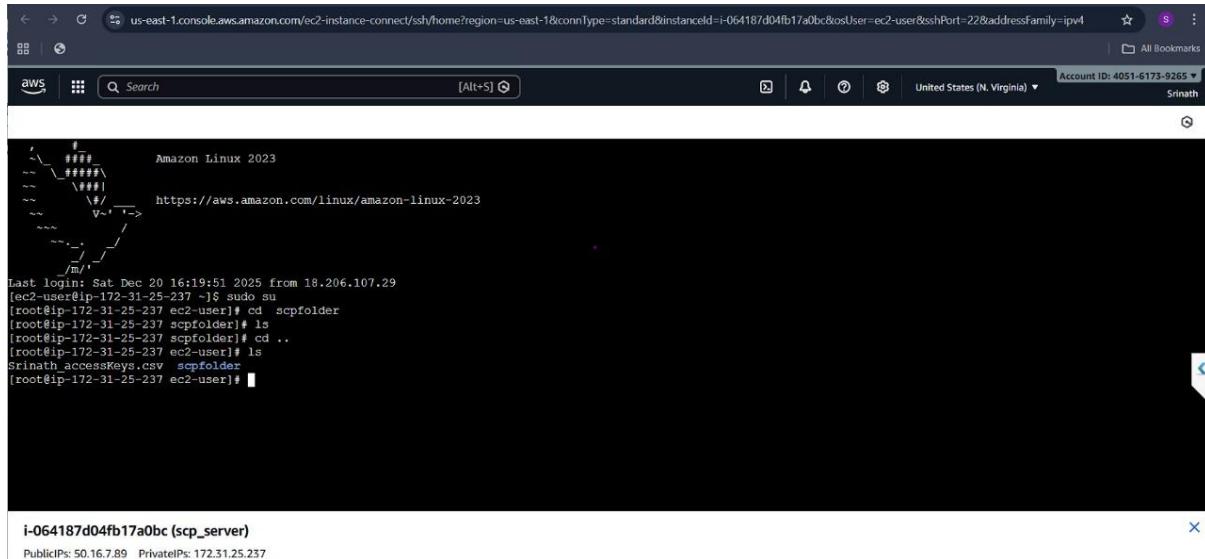
SCP:

The scp command (secure copy) is a command-line utility used to securely transfer files and directories between local and remote systems, or between two remote systems. It uses the SSH (Secure Shell) protocol for both authentication and encryption, ensuring data confidentiality during transfer.

```
SRINATH@Leo MINGW64 ~/downloads
$ scp -i mykey.pem Srinath_accessKeys.csv ec2-user@50.16.7.89:/home/ec2-user/scpfolder
scp: dest open '/home/ec2-user/scpfolder/Srinath_accessKeys.csv': Permission denied
scp: failed to upload file Srinath_accessKeys.csv to /home/ec2-user/scpfolder

SRINATH@Leo MINGW64 ~/downloads
$ scp -i mykey.pem Srinath_accessKeys.csv ec2-user@50.16.7.89:/home/ec2-user/
Srinath_accessKeys.csv

SRINATH@Leo MINGW64 ~/downloads
$
```



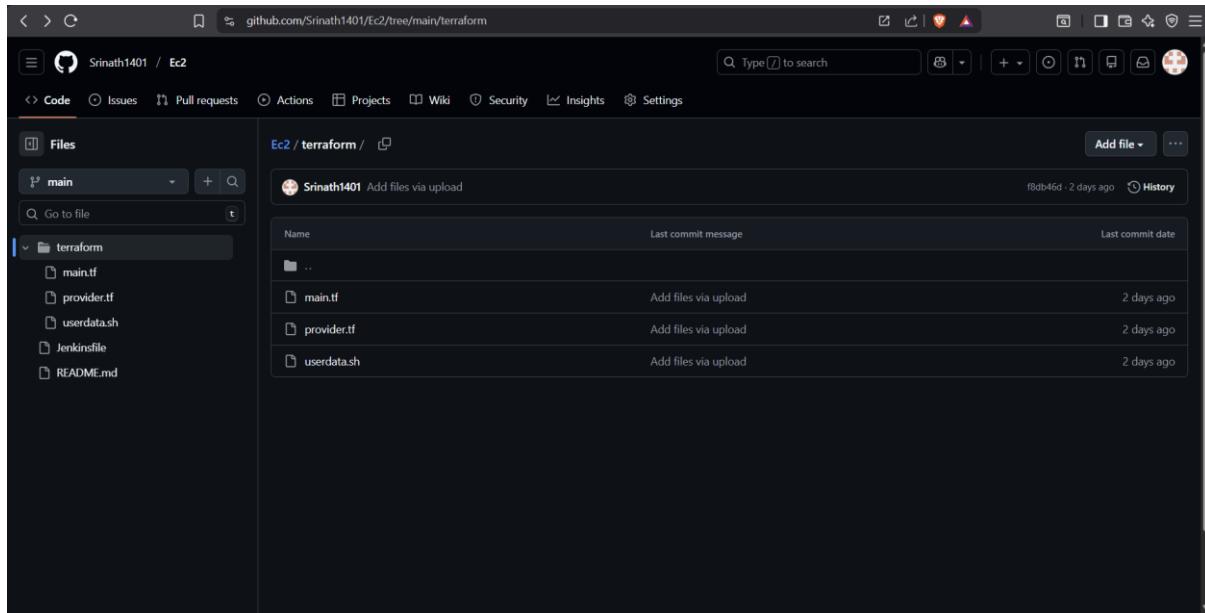
```
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

Last login: Sat Dec 20 16:19:51 2025 from 18.206.107.29
[ec2-user@ip-172-31-25-237 ~]$ sudo su
[root@ip-172-31-25-237 ec2-user]# cd scpfolder
[root@ip-172-31-25-237 scpfolder]# ls
[root@ip-172-31-25-237 scpfolder]# cd ..
[root@ip-172-31-25-237 ec2-user]# ls
Srinath_accessKeys.csv  scpfolder
[root@ip-172-31-25-237 ec2-user]#
```

i-064187d04fb17a0bc (scp_server)
PublicIPs: 50.16.7.89 PrivateIPs: 172.31.25.237

Jenkins CI/CD for Ec2

Create Terraform folder in github and put terraform files:



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"]
}
```

```
egress {
```

```

from_port = 0
to_port = 0
protocol = "-1"
cidr_blocks = ["0.0.0.0/0"]
}

}

resource "aws_instance" "web" {
ami           = "ami-0ecb62995f68bb549" # 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-Web-Server"
}
}

```

Provider.tf:

```

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

```

Userdata.tf:

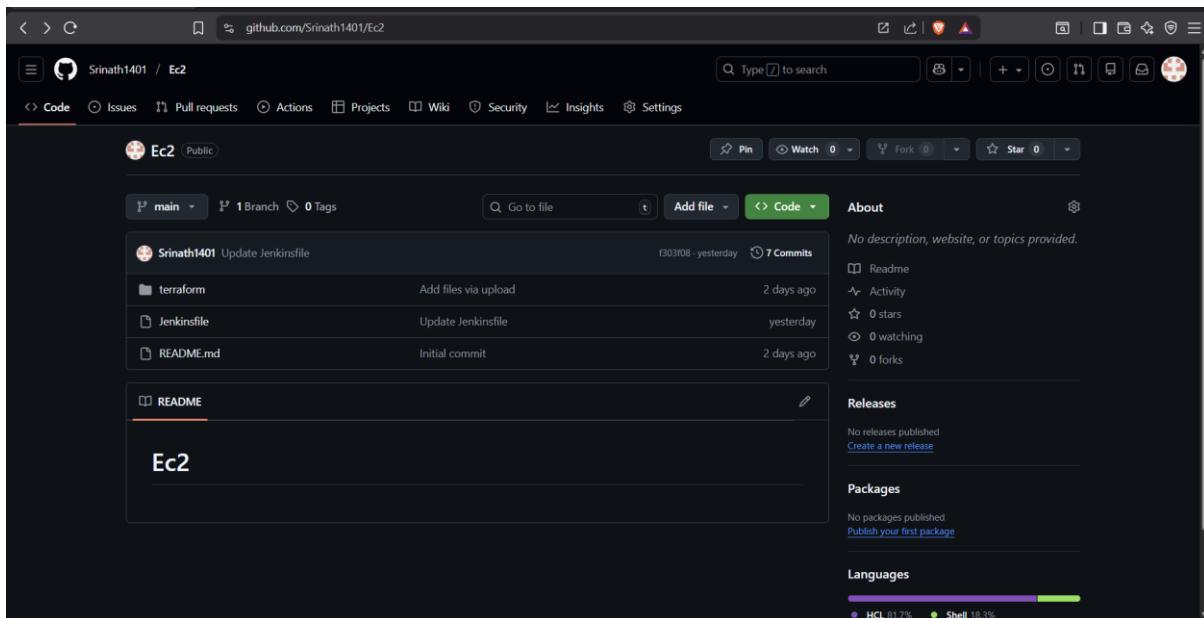
```

#!/bin/bash
apt update -y
apt install nginx -y
systemctl start nginx
systemctl enable nginx

echo "<h1>Deployed using Jenkins + Terraform 🚀 </h1>" >
/var/www/html/index.htm

```

Jenkinsfile:



Code:

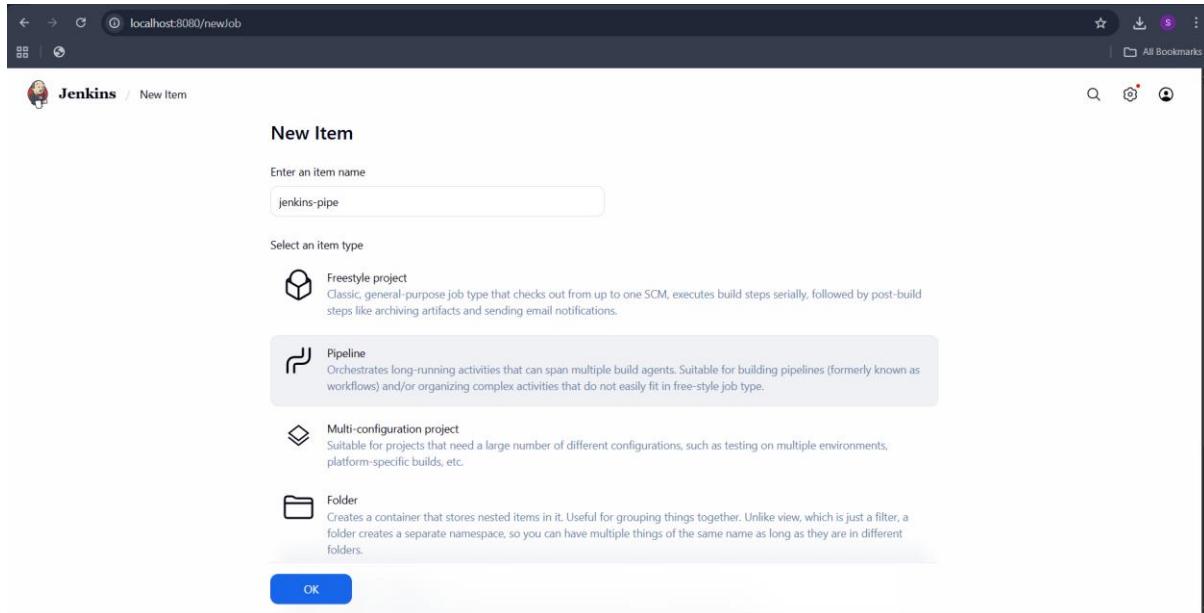
```
pipeline {
    agent any

    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/Ec2.git'
            }
        }
        stage('Terraform Steps') {
            steps {
                withCredentials([
                    $class: 'AmazonWebServicesCredentialsBinding',

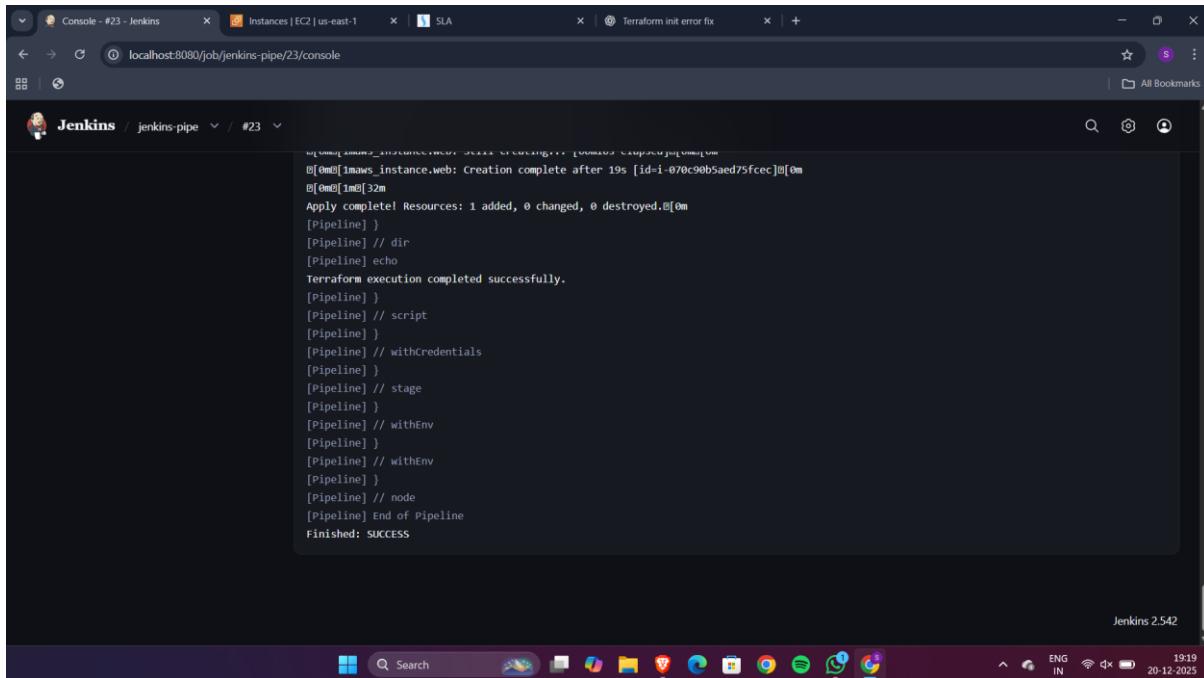
```

```
    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:



Output:



The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed. The main content area displays a table titled "Instances (1) Info". The table has columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4. One row is visible, showing "Jenkins-Web-..." as the name, "i-070c90b5aed75fc0c" as the Instance ID, "Running" as the Instance state, "t2.micro" as the Instance type, "Initializing" as the Status check, "View alarms +" as the Alarm status, "us-east-1b" as the Availability Zone, and "ec2-54-197-24-135" as the Public IPv4. There are buttons for "Connect", "Actions", and "Launch instances". A search bar at the top says "Find Instance by attribute or tag (case-sensitive)" with a filter set to "Instance state = running".

The screenshot shows a web browser window with the URL "54.197.32.135". The title bar says "Not secure". The page content includes the text "Deployed using Jenkins + Terraform 🎉".

