

TERRAFORM

JenkinsFile:

```
    pipeline {
    agent any

    environment {
        // Ensure that AWS credentials are available as environment variables
        AWS_ACCESS_KEY_ID    = credentials('aws-terraform-credentials') // Reference
your AWS credentials ID in Jenkins
        AWS_SECRET_ACCESS_KEY = credentials('aws-terraform-credentials')
        AWS_DEFAULT_REGION    = 'us-east-1' // Specify your AWS region
    }

    stages {
        stage('Clone Repository') {
            steps {
                script {
                    // Clone the repository
                    // Using Git credentials if the repository is private
                    git url: 'https://github.com/Krupa2003/terraformassign.git', branch: 'main'

                    // Print directory listing to verify files
                    sh 'ls -la'
                }
            }
        }

        stage('Terraform Init') {
            steps {
                script {
                    // Ensure Terraform init is run in the correct directory
                    // If Terraform files are in the root of the repository, this is sufficient
                    sh 'terraform init'
                }
            }
        }

        stage('Terraform Plan') {
            steps {
```

```

        script {
            // Plan the infrastructure
            sh 'terraform plan'
        }
    }
}

stage('Terraform Apply') {
    steps {
        script {
            // Apply the Terraform configuration
            // Use the `-auto-approve` flag to skip the confirmation prompt
            sh 'terraform apply -auto-approve'
        }
    }
}

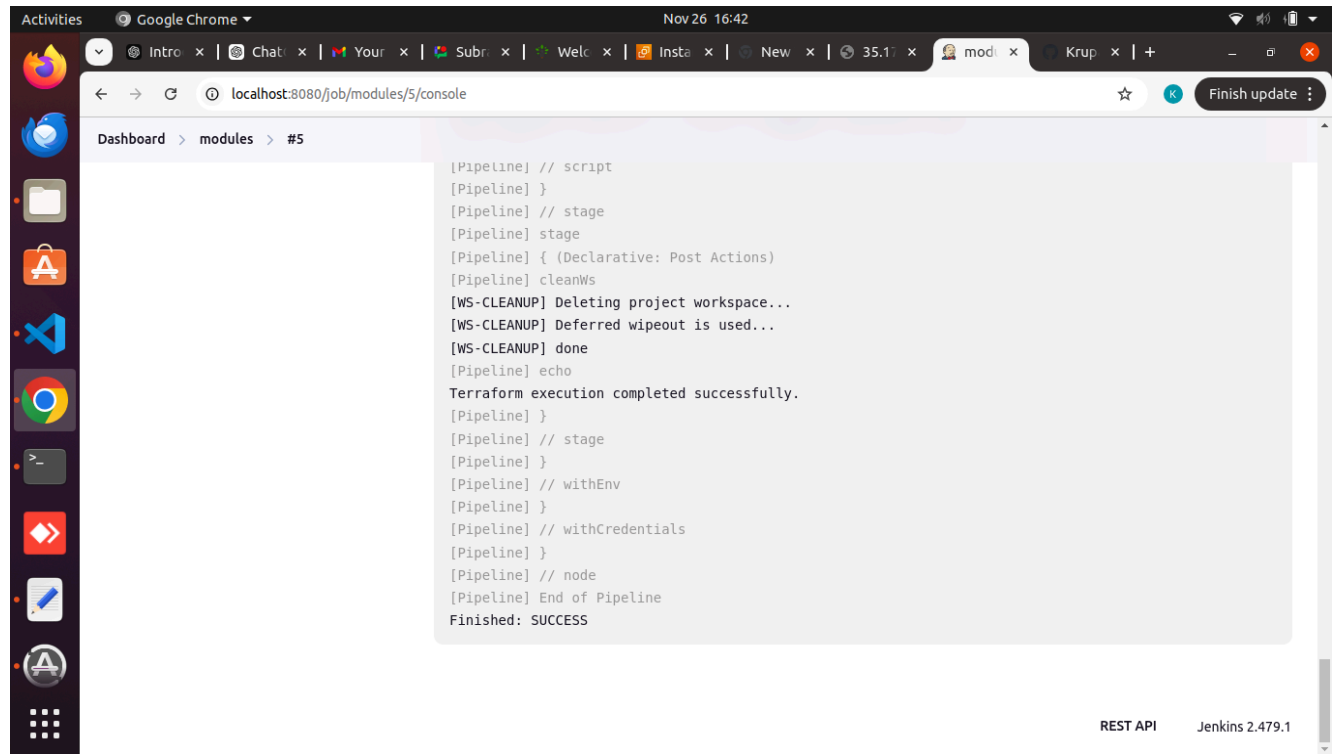
post {
    always {
        // Always clean the workspace after execution to avoid conflicts in future builds
        cleanWs()
    }

    success {
        // Notify on success, if necessary
        echo "Terraform execution completed successfully."
    }

    failure {
        // Handle failure, possibly notify through Slack, email, etc.
        echo "Terraform execution failed."
    }
}
}

```

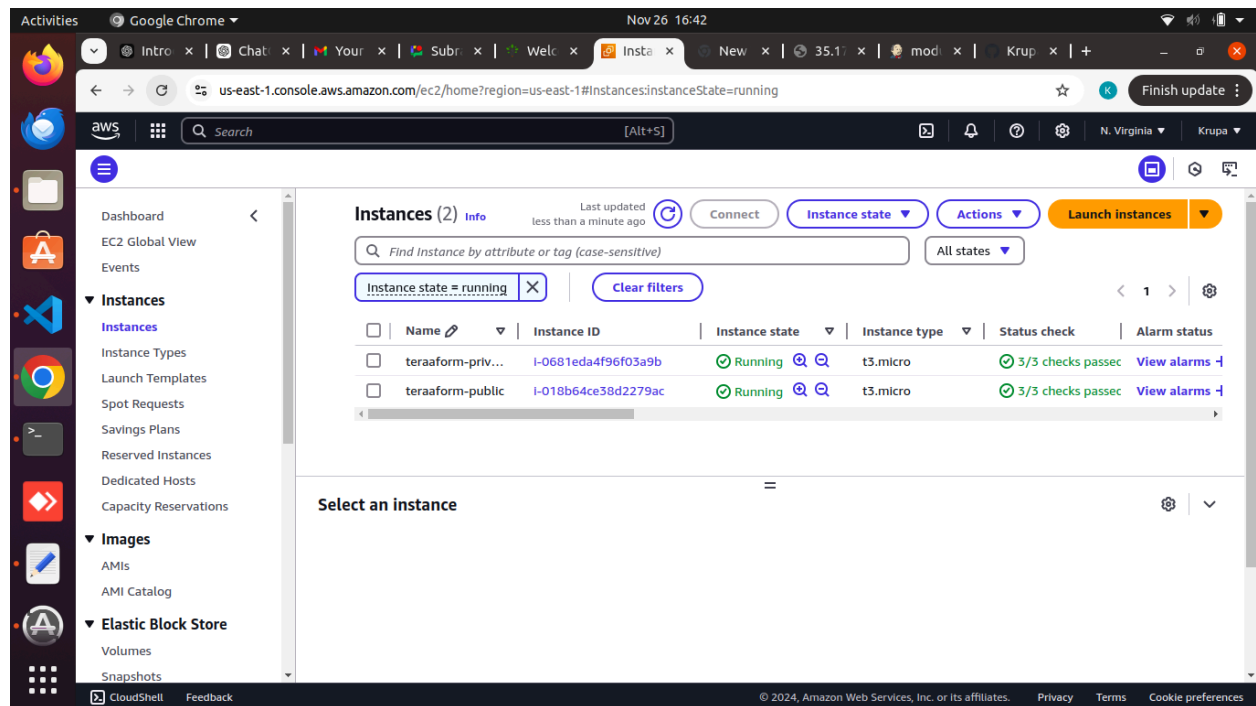
Output:



The screenshot shows a web browser window with the URL `localhost:8080/job/modules/5/console`. The page displays the output of a Jenkins pipeline job. The output text is as follows:

```
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] cleanWs
[WS-CLEANUP] Deleting project workspace...
[WS-CLEANUP] Deferred wipeout is used...
[WS-CLEANUP] done
[Pipeline] echo
Terraform execution completed successfully.
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

At the bottom right of the console output area, the text "REST API" and "Jenkins 2.479.1" is visible.



The screenshot shows the AWS Management Console for the `us-east-1` region. The "Instances" page is displayed, showing a list of two EC2 instances. The left sidebar contains navigation links for Dashboard, EC2 Global View, Events, and various instance types and templates. The main content area shows the "Instances (2)" list with the following details:

| | Name | Instance ID | Instance state | Instance type | Status check | Alarm status |
|--------------------------|-------------------|---------------------|----------------|---------------|-----------------|--------------|
| <input type="checkbox"/> | teraaform-priv... | i-0681eda4f96f03a9b | Running | t3.micro | 3/3 checks pass | View alarms |
| <input type="checkbox"/> | teraaform-public | i-018b64ce38d2279ac | Running | t3.micro | 3/3 checks pass | View alarms |

Below the table, there is a "Select an instance" section. The footer of the console shows the copyright notice: "© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences".