

```

pipeline {
  agent { label 'build-server-1' } // Ensure job runs on the right node

  environment {
    TF_WORKING_DIR = "AWS/terraform" // Set Terraform working directory inside
    AWS/terraform
    TF_VERSION = ">=5.32.0" // Adjust to match your Terraform version
    AWS_REGION = "ca-central-1" // Ensure AWS Region is correctly set
  }

  parameters {
    choice(name: 'ACTION', choices: ['plan', 'apply', 'destroy'], description: 'Choose Terraform
    Action')
    string(name: 'FME_CORE_IMAGE_ID', description: 'Enter the FME Core AMI ID')
    string(name: 'FME_ENGINE_IMAGE_ID', description: 'Enter the FME Engine AMI ID')
  }

  stages {
    stage('Checkout') {
      steps {
        git branch: 'main',
        credentialsId: 'github-cred', // Use the Jenkins credential ID
        url: 'https://github.com/<GitHub Repo>/fme-server-iac-templates.git'
      }
    }

    stage('Initialize Terraform') {
      steps {
        dir("${TF_WORKING_DIR}") { // Run inside AWS/terraform
          sh 'terraform init -upgrade'
        }
      }
    }

    stage('Validate Terraform') {
      steps {
        dir("${TF_WORKING_DIR}") {
          sh 'terraform validate'
        }
      }
    }

    stage('Terraform Plan') {
      when {

```

```

    expression { params.ACTION == 'plan' || params.ACTION == 'apply' }
  }
  steps {
    withCredentials([
      string(credentialsId: 'aws-access-key', variable: 'AWS_ACCESS_KEY_ID'),
      string(credentialsId: 'aws-secret-key', variable: 'AWS_SECRET_ACCESS_KEY')
    ]) {
      dir("${TF_WORKING_DIR}") {
        sh """
          export AWS_ACCESS_KEY_ID=$AWS_ACCESS_KEY_ID
          export AWS_SECRET_ACCESS_KEY=$AWS_SECRET_ACCESS_KEY
          export AWS_REGION="ca-central-1"

          terraform plan -var="owner=MerlineGeorge" \
            -var="public_access=0.0.0.0/0" \
            -var="region=ca-central-1" \
            -var="fme_core_image_id=${params.FME_CORE_IMAGE_ID}" \
            -var="fme_engine_image_id=${params.FME_ENGINE_IMAGE_ID}" \
            -var="ad_admin_pw=<inputHere>" \
            -var="db_admin_user=<inputHere>" \
            -var="db_admin_pw=<inputHere>" \
            -out=tfplan
          """ // Please DO NOT use FMEFLOW or POSTGRES as db_admin_user
        }
      }
    }
  }
}

stage('Terraform Apply') {
  when {
    expression { params.ACTION == 'apply' }
  }
  steps {
    withCredentials([
      string(credentialsId: 'aws-access-key', variable: 'AWS_ACCESS_KEY_ID'),
      string(credentialsId: 'aws-secret-key', variable: 'AWS_SECRET_ACCESS_KEY')
    ]) {
      dir("${TF_WORKING_DIR}") {
        sh """
          export AWS_ACCESS_KEY_ID=$AWS_ACCESS_KEY_ID
          export AWS_SECRET_ACCESS_KEY=$AWS_SECRET_ACCESS_KEY
          export AWS_REGION="ca-central-1"

          terraform apply -auto-approve tfplan
        """
      }
    }
  }
}

```

```

        """
    }
}
}
}

stage('Terraform Destroy') {
    when {
        expression { params.ACTION == 'destroy' }
    }
    steps {
        withCredentials([
            string(credentialsId: 'aws-access-key', variable: 'AWS_ACCESS_KEY_ID'),
            string(credentialsId: 'aws-secret-key', variable: 'AWS_SECRET_ACCESS_KEY')
        ]) {
            dir("${TF_WORKING_DIR}") {
                sh """
                    export AWS_ACCESS_KEY_ID=${AWS_ACCESS_KEY_ID}
                    export AWS_SECRET_ACCESS_KEY=${AWS_SECRET_ACCESS_KEY}
                    export AWS_REGION="ca-central-1"

                    terraform destroy -auto-approve \
                        -var="owner=MerlineGeorge" \
                        -var="public_access=0.0.0.0/0" \
                        -var="region=ca-central-1" \
                        -var="fme_core_image_id=${params.FME_CORE_IMAGE_ID}" \
                        -var="fme_engine_image_id=${params.FME_ENGINE_IMAGE_ID}" \
                        -var="ad_admin_pw=<inputHere>" \
                        -var="db_admin_user=<inputHere>" \
                        -var="db_admin_pw=<inputHere>" \
                    """ // Please DO NOT use FMEFLOW or POSTGRES as db_admin_user
            }
        }
    }
}

post {
    always {
        echo "Pipeline execution completed"
    }
}
}

```