Lab 18: CI/CD with Terraform using Jenkins (Basic Level)

Author: Dr. Sandeep Kumar Sharma
Level: Beginner
Platform: Ubuntu Linux + Microsoft Azure + Jenkins
Prerequisite: Lab 1 to Lab 17

Learning Objective

Participants will learn:

- What CI/CD means for Terraform
- Why Jenkins is used
- Basic Jenkins architecture
- How Jenkins works with Terraform
- How to build a simple Terraform pipeline
- How to automate Terraform using Jenkins

Learning Outcome

After completing this lab, participants will:

- Understand CI/CD basics
- Understand Jenkins + Terraform integration
- Create a simple pipeline
- Automate infrastructure deployment

Concept Explanation

What is CI/CD with Terraform?

CI/CD with Terraform means:

- Code is stored in Git
- Jenkins pulls the code
- Jenkins runs Terraform
- Infrastructure is created automatically

Why Jenkins?

- Open source
- Widely used
- Industry standard
- Easy to learn
- Pipeline based automation

Architecture

```
Developer → GitHub → Jenkins → Terraform → Azure
```

Hands-On Lab

Step 1: Install Jenkins on Ubuntu

```
sudo apt update
sudo apt install openjdk-11-jdk -y
wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -
sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ > /etc/apt/
sources.list.d/jenkins.list'
sudo apt update
sudo apt install jenkins -y
```

Start Jenkins:

```
sudo systemctl start jenkins
sudo systemctl enable jenkins
```

Step 2: Access Jenkins

Browser:

```
http://<server-ip>:8080
```

Unlock Jenkins:

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

Install suggested plugins.

Step 3: Install Terraform on Jenkins Server

```
wget https://releases.hashicorp.com/terraform/1.6.0/
terraform_1.6.0_linux_amd64.zip
unzip terraform_1.6.0_linux_amd64.zip
sudo mv terraform /usr/local/bin/
terraform -v
```

GitHub Repository Setup

Step 4: Create GitHub Repo

Name:

```
terraform-jenkins-lab
```

Step 5: Add Terraform Code

`main.tf`

```
provider "azurerm" {
  features {}
}

resource "azurerm_resource_group" "rg" {
  name     = "rg-jenkins-demo"
  location = "East US"
}
```

Push to GitHub.

Azure Authentication

Step 6: Create Service Principal

```
az ad sp create-for-rbac --name terraform-jenkins-sp --role Contributor --
scopes /subscriptions/<SUBSCRIPTION_ID>
```

Save:

- clientId
- clientSecret
- tenantId
- subscriptionId

Jenkins Configuration

Step 7: Add Azure Credentials in Jenkins

Jenkins Dashboard → Manage Jenkins → Credentials → Global → Add Credentials

Type: Secret text / Username & Password

Add:

```
ARM_CLIENT_ID
ARM_CLIENT_SECRET
ARM_TENANT_ID
ARM_SUBSCRIPTION_ID
```

Create Jenkins Pipeline

Step 8: Create New Pipeline Job

Jenkins → New Item → Name: `terraform-pipeline` → Pipeline

Step 9: Pipeline Script (Basic)

```
pipeline {
    agent any

    environment {
        ARM_CLIENT_ID = credentials('ARM_CLIENT_ID')
        ARM_CLIENT_SECRET = credentials('ARM_CLIENT_SECRET')
        ARM_TENANT_ID = credentials('ARM_TENANT_ID')
        ARM_SUBSCRIPTION_ID = credentials('ARM_SUBSCRIPTION_ID')
    }

    stages {
        stage('Checkout') {
            steps {
                git 'https://github.com/<your-username>/terraform-jenkins-
lab.git'
            }
        }

        stage('Terraform Init') {
            steps {
                sh 'terraform init'
            }
        }

        stage('Terraform Plan') {
            steps {
                sh 'terraform plan'
            }
        }

        stage('Terraform Apply') {
            steps {
                sh 'terraform apply -auto-approve'
            }
        }
    }
}
```

Run Pipeline

Step 10: Build Pipeline

Click **Build Now**

Verification

- Jenkins Console Output
- Azure Portal → Resource Group `rg-jenkins-demo`

CI/CD Flow

```
Git Commit → Jenkins Pull → Terraform Init → Plan → Apply → Azure Infra
```

Cleanup

Destroy infra by modifying pipeline:

```
sh 'terraform destroy -auto-approve'
```

or manual destroy.

Basic Understanding Summary

- Jenkins = Automation engine
- Terraform = Infrastructure engine
- Git = Source control
- Azure = Cloud platform

Together:

CI/CD for Infrastructure