# **Complete Terraform Notes for DevOps**

## **1. Introduction to Terraform**

Terraform is an open-source Infrastructure as Code (IaC) tool developed by HashiCorp. It allows managing and provisioning infrastructure using declarative configuration files.

### **Why Terraform for DevOps?**

✅ Infrastructure as Code (IaC)  
✅ Supports multiple cloud providers (AWS, Azure, GCP)  
✅ Declarative syntax for easy infrastructure management  
✅ Efficient state management  
✅ Modular and scalable

## **2. Installing Terraform**

### **On Linux/macOS**

wget https://releases.hashicorp.com/terraform/X.Y.Z/terraform\_X.Y.Z\_linux\_amd64.zip

unzip terraform\_X.Y.Z\_linux\_amd64.zip

sudo mv terraform /usr/local/bin/

terraform -v # Verify installation

### **On Windows**

1. Download Terraform from [Terraform Official Website](https://developer.hashicorp.com/terraform/downloads)
2. Extract and add it to the system PATH
3. Verify using:

terraform -v

## **3. Terraform Workflow**

1. **Write** - Define infrastructure in .tf files.
2. **Init** - Initialize the Terraform working directory.
3. **Plan** - Preview changes before applying.
4. **Apply** - Deploy infrastructure.
5. **Destroy** - Tear down infrastructure.

### **Basic Terraform Commands**

terraform init # Initialize Terraform working directory

terraform plan # Show execution plan before applying changes

terraform apply # Apply changes and provision infrastructure

terraform destroy # Destroy the created infrastructure

## **4. Basic Terraform Configuration Example**

### **Deploying an AWS EC2 Instance**

provider "aws" {

region = "us-east-1"

}

resource "aws\_instance" "example" {

ami = "ami-0abcdef1234567890"

instance\_type = "t2.micro"

tags = {

Name = "TerraformInstance"

}

}

### **Executing Terraform**

terraform init

terraform plan

terraform apply

## **5. Variables in Terraform**

### **Using Variables in Terraform**

variable "region" {

default = "us-east-1"

}

provider "aws" {

region = var.region

}

### **Defining Variables in a Separate File (**variables.tf**)**

variable "instance\_type" {

type = string

default = "t2.micro"

}

## **6. Outputs in Terraform**

### **Defining an Output**

output "instance\_ip" {

value = aws\_instance.example.public\_ip

}

### **Getting Outputs**

terraform output instance\_ip

## **7. Terraform State Management**

### **State File (**terraform.tfstate**)**

* Keeps track of infrastructure resources.
* Should be stored securely (e.g., remote backend in S3).

### **State Management Commands**

terraform state list # List resources in state

terraform state show aws\_instance.example # Show details of a resource

terraform refresh # Sync real-world infrastructure with state file

## **8. Terraform Modules**

Modules help reuse and organize Terraform configurations.

### **Creating a Module (**modules/aws\_instance/main.tf**)**

variable "instance\_type" {}

resource "aws\_instance" "web" {

ami = "ami-0abcdef1234567890"

instance\_type = var.instance\_type

}

### **Using a Module**

module "web" {

source = "./modules/aws\_instance"

instance\_type = "t2.micro"

}

## **9. Terraform Provisioners**

Provisioners run scripts on a resource.

resource "aws\_instance" "example" {

ami = "ami-0abcdef1234567890"

instance\_type = "t2.micro"

provisioner "local-exec" {

command = "echo Instance created!"

}

}

## **10. Terraform with Kubernetes**

### **Deploying a Kubernetes Pod**

provider "kubernetes" {

config\_path = "~/.kube/config"

}

resource "kubernetes\_pod" "nginx" {

metadata {

name = "nginx-pod"

}

spec {

container {

image = "nginx:latest"

name = "nginx-container"

}

}

}

## **11. Terraform Best Practices**

✅ Use remote state management (S3, GCS, etc.)  
✅ Modularize configurations for reusability  
✅ Use variables and outputs efficiently  
✅ Implement proper IAM permissions for security  
✅ Enable Terraform locking mechanisms to avoid conflicts

## **Conclusion**

Terraform simplifies infrastructure provisioning for DevOps. Mastering Terraform helps automate cloud deployments efficiently. 🚀 Let me know if you need additional topics! 😊