**Terraform Notes for DevOps Beginners**

**1. Introduction**

Terraform is an **open-source Infrastructure as Code (IaC) tool** developed by **HashiCorp**.  
It helps DevOps engineers and cloud administrators **provision, manage, and automate infrastructure** in a safe, consistent, and repeatable way.

* Written in **HashiCorp Configuration Language (HCL)** (also supports JSON).
* Works with **multiple cloud providers** (AWS, Azure, GCP, etc.) and on-premises solutions.
* Focuses on **declarative configuration** → you describe the desired infrastructure, and Terraform builds it for you.

**2. Infrastructure as Code (IaC)**

**Definition:** IaC is the practice of managing and provisioning computing infrastructure (servers, networks, databases, etc.) using **machine-readable configuration files** instead of manual processes.

**Benefits of IaC**

* **Automation** → faster deployments.
* **Consistency** → avoids manual errors.
* **Scalability** → repeat configurations across environments.
* **Version Control** → configs can be tracked in Git.

**3. IaC with Terraform**

Terraform applies IaC principles by:

* Allowing you to **write infrastructure in HCL files**.
* Managing both **cloud resources** (e.g., AWS EC2, S3) and **on-prem systems**.
* Using a **state file** to track existing resources.
* Ensuring **idempotency** → applying the same config multiple times gives the same result.

**4. Terraform Workflow**

The standard Terraform workflow has **three main steps**:

1. **Write Configuration**
   * Create .tf files with resource definitions.
   * Example: defining an EC2 instance or VPC.
2. **Plan**
   * Run terraform plan to preview what will be created, changed, or destroyed.
   * Ensures safety before making changes.
3. **Apply (Deploy)**
   * Run terraform apply to execute the plan.
   * Terraform provisions infrastructure exactly as described in configs.

**5. Terraform Key Concepts**

**a. Terraform Initialization (terraform init)**

* Initializes the working directory.
* Downloads required **providers** (e.g., AWS, Azure).
* Prepares the environment for use.
* Must be run **once per project** before other commands.

**b. Terraform Plan (terraform plan)**

* Creates an **execution plan** without making changes.
* Shows **actions Terraform will perform** (Add, Change, Destroy).
* Safe step before applying.

**c. Terraform Apply (terraform apply)**

* Executes the changes defined in your .tf files.
* Applies the plan to real infrastructure.
* After execution, Terraform updates the **state file**.

**d. Terraform Destroy (terraform destroy)**

* Removes all resources defined in the configuration.
* Useful for cleaning up test environments.
* Always review before destroying to avoid accidental deletions.