

Infrastructure as Code:

Terraform vs Ansible

Infrastructure as Code (IaC) is the practice of managing and provisioning infrastructure through code instead of manual processes. This approach brings automation, repeatability, and version control to infrastructure provisioning, aligning it closely with modern DevOps practices.

Two of the most popular tools in the IaC ecosystem are **Terraform** and **Ansible**. While both are used to automate infrastructure, they differ in purpose, approach, and underlying architecture.

Terraform: Declarative Provisioning

Terraform, developed by HashiCorp, is a tool focused on **provisioning infrastructure**.

Key Features:

- **Declarative Syntax** (HCL): You describe what the infrastructure should look like, not how to create it.
- **Immutable Infrastructure**: Rather than modifying existing infrastructure, Terraform often destroys and recreates resources to reach the desired state.
- **State Management**: Maintains a `.tfstate` file that tracks real infrastructure vs. desired state.
- **Provider Ecosystem**: Supports AWS, Azure, GCP, Kubernetes, and more through plugins called providers.

Use Cases:

- Provisioning EC2 instances, load balancers, databases, VPCs.
- Setting up cloud resources across multi-cloud environments.

- Managing Kubernetes infrastructure.

Terraform vs Ansible: Key Differences

Feature	Terraform	Ansible
Type	Infrastructure Provisioning	Configuration Management
Language	Declarative (HCL)	Procedural (YAML)
State Management	Yes (tfstate)	No (stateless)
Execution Mode	Plan → Apply	Ad-hoc or Playbook-driven
Agent Requirement	No	No (uses SSH)
Best For	Provisioning infrastructure	Configuring OS and software
Cloud Support	Excellent (AWS, Azure, GCP, etc.)	Good (through modules)
Orchestration	Limited (via <code>depends_on</code>)	Strong (task sequencing, roles)

Summary

- **Terraform** is best for infrastructure provisioning in a consistent, version-controlled manner.
- **Ansible** excels at post-provisioning configuration and ongoing system management.
- They solve different problems but together enable robust, scalable, and automated infrastructure.