

Terraform

Terraform is used for **infrastructure as code (IaC) to automate the building, changing, and versioning of cloud and on-premises infrastructure**. It allows users to define the desired state of their infrastructure in configuration files, which Terraform then uses to provision and manage resources like servers, storage, and networks across various providers, including public clouds and private data centers.

How Terraform works

Define infrastructure: You write code in a declarative language called HashiCorp Configuration Language (HCL) to describe your infrastructure needs.

Generate a plan: Terraform creates an execution plan that outlines the steps needed to reach the desired state.

Provision resources: You execute the plan, and Terraform uses APIs to create, update, or delete resources to match your configuration. **Manage infrastructure:** Terraform can manage infrastructure throughout its lifecycle, from initial provisioning to ongoing management and updates.

Key use cases

Multi-cloud deployments: Manage resources across different cloud providers (like AWS, Azure, and Google Cloud) using a single, consistent workflow.

Environment creation: Easily and consistently create identical environments for development, testing, staging, and production.

Reusable infrastructure: Use modules to create reusable components for infrastructure that can be shared across projects.

Version control: Treat your infrastructure like code by storing the configuration files in version control systems, allowing for collaboration and a history of changes.

Automated updates and rollbacks: Safely update or modify infrastructure and roll back changes if a deployment fails.

Revision #2

Created 29 October 2025 02:43:38 by AI API

Updated 7 December 2025 06:14:52 by AI Channel