

Terraform vs. Ansible: When to Use Each



When looking for an infrastructure-as-code (IaC) tool, you'll find several options, including Terraform and [Ansible](#). Both of these IaC tools provision infrastructure in the cloud, but they have distinct differences that make one better to use than the other in certain environments, which make each tool more suitable than the other for certain use cases. Terraform is more of a provisioning tool, while Ansible is used for configuration of cloud infrastructure.

What Does Terraform Do?

Instead of manually configuring cloud infrastructure (e.g., virtual machines), Terraform lets DevOps teams write configuration scripts—one or more text files written in Hashicorp Control Language—that deploy resources remotely using automation. It can be used against any infrastructure for which a provider exists.

How Does Terraform Work?

Terraform uses a configuration as the reference for the desired state of infrastructure resources and ensures that the provisioned resources match what is specified in the configuration. The use of its configurations ensures that infrastructure is deployed in a consistent manner. It can deploy virtual

machines, network components like firewalls, database servers, and any other hardware used to run the cloud environment.

What Does Ansible Do?

Ansible is used to configure infrastructure. After you deploy infrastructure, you might need to make changes to configurations to support new software or upgraded features. DevOps can write automation scripts to change configurations across multiple cloud resources.

How Does Ansible Work?

When you want to automatically configure infrastructure in the cloud, it would be better to use Ansible than Terraform. Developers create YAML files to automate configuration deployments in the cloud. Terraform also deploys configurations, but Ansible will let you update and change current infrastructure configurations.

Ansible works with “playbooks,” which are procedural scripts that let developers define what they want to execute on infrastructure. It works with DevOps operations to continually make changes to infrastructure as developers deploy applications.

Terraform vs. Ansible: 3 Similarities

Although Terraform and Ansible are used for different purposes, they have some similarities. They both have orchestration capabilities that let DevOps deploy infrastructure to the cloud. While infrastructure is deployed, they both configure it at the same time to make it functional for developers or operations people.

They both work on virtual machines without installing agents on the remote infrastructure, and both work with SSH connections. Neither of them require expensive third-party infrastructure to manage their state, meaning that Ansible and Terraform are considered masterless systems.

Terraform vs. Ansible: 3 Differences

If you use both Terraform and Ansible in your environment, you should know that they have differences, which define their uses in DevOps. Ansible is for configurations, while Terraform is for provisioning mutable infrastructure. It uses the HCL syntax, while Ansible uses the common YAML syntax.

The big difference is that Terraform is declarative, which means that code can be dispersed across multiple files, and code isn't executed in sequence. The Ansible YAML syntax is procedural, which means that every line of code is executed sequentially. Ansible playbooks are a series of executable tasks used to define configurations, which usually needs to be done in order for it to be successful.

Terraform allows for mutable functionality, which means that infrastructure can be changed to run newer applications or hardware. Ansible is immutable, which means that new infrastructure must be deployed if you change configurations.

When to Use Terraform

If you have a [DevOps team](#) that needs to rapidly deploy infrastructure along with applications, Terraform

will give developers easy access to API endpoints for fast deployments. Developers can connect to a terminal where they can write and test code before they put it into production.

It is mainly used for simple provisioning without many configurations. It has some configuration functionality during deployment, but it's not a good choice if you need to automate configurations on your public cloud hardware. For example, provisioning a virtual machine for use with software could be a good use for Terraform.

When to Use Ansible

After provisioning hardware, Ansible is a good choice for continual configuration of resources. Ansible is specific for configurations, so it's best for automating cloud resources to work with specific applications. It can also be used to help other departments as they need to make configuration changes to infrastructure.

For example, let's say that you provision a virtual machine but need to add it to a load balancer. The configurations could be automated using Ansible.

Terraform vs. Ansible: FAQs

Can you run Ansible from Terraform?

Although Ansible and Terraform are different platforms, you can invoke Ansible from Terraform. This would be done after you provision infrastructure from Terraform and then configure it using Ansible. You first need to create an Ansible playbook and then call Ansible from your Terraform HCL script.

Can Ansible replace Terraform?

Ansible can be used to deploy some infrastructure, but it can't completely replace Terraform. It deploys infrastructure, and then Ansible will execute after infrastructure deployment to configure it. They work well bundled together in a DevOps environment.

