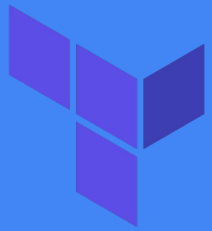


HashiCorp

Terraform

Infrastructure-as-Code

Terraform on AWS

What is IaC?

- A way to help infrastructure provisioning and configuration through code without the need to do that physically or manually
- Easier infrastructure automation with the pipelines
- ***It's part of your codebase/repos; tracked and version-controlled***
  - ***It's not anymore something left to the discretion of the sys-admin or devops engineer to decide informally.***
- Easier security scanning and vulnerability discovery

How to start doing it?

- Terraform (TF) by Hashicorp (AWS, Azure, GCP, K8s, …)
- Pulumi (same as TF)
- CloudFormation by AWS (only for AWS)
- Azure Resource Manager (only for Azure)
- CDKs & SDKs (e.g. AWS CDK
- …

What can you do with it?

- Creating Virtual Machines/Servers (like EC2)
- Creating Database Instances (like RDS)
- Creating a Kubernetes Cluster (like EKS)
- Creating K8s resources (deployments, ingresses, …)
- Creating anything you can think of in the cloud
- …

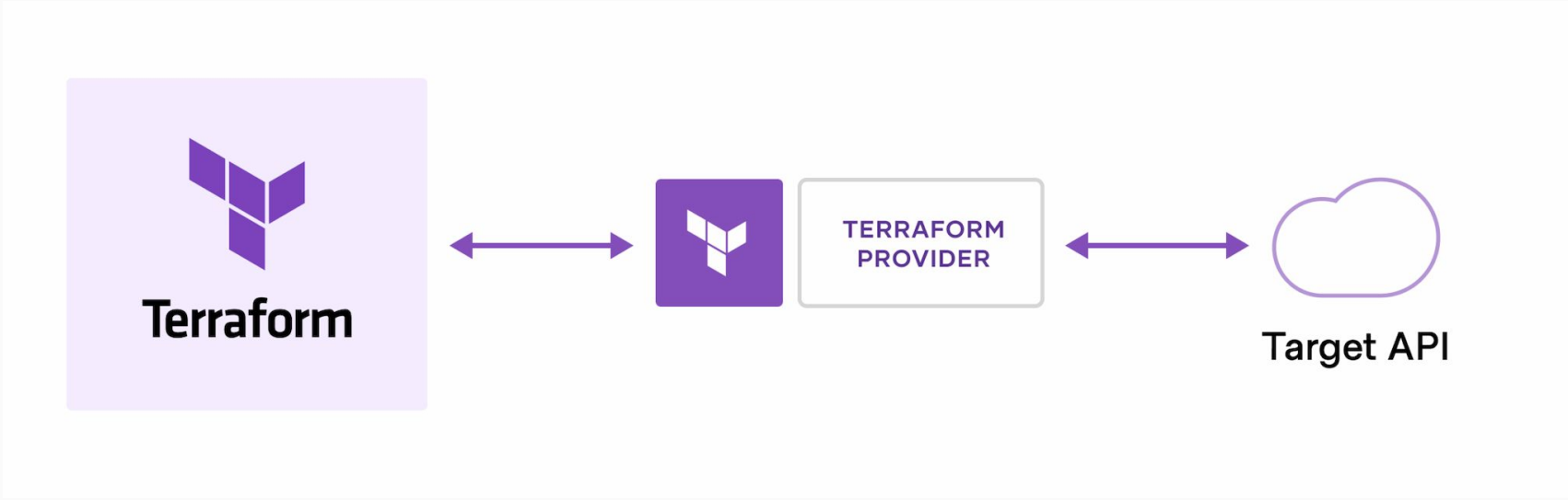
Terraform

- The most widely-used IaC tool.
- Helps manage and automate infrastructure.
- Uses a declarative language to describe infrastructure; little logic involved.

Some Terraform Terms

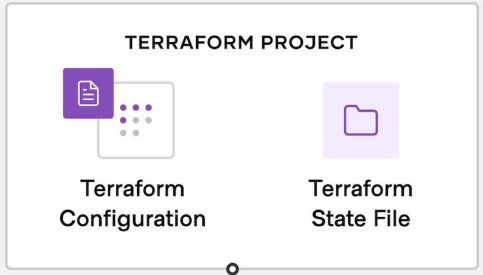
- Provider: plugins that help terraform interact with different platforms (e.g. AWS, Azure, K8s, …). Think of them as libraries/packages.
- Resource: an infrastructure piece you're defining. What can be defined? Anything a provider allows.
- State File: a file containing information about the current state of the infrastructure.
- Backend: the place where the state file is stored; can be local or in the cloud.
- Var: a terraform value/variable that the user can set
- Local: a terraform value that is constant-ish and the user can't change

Terraform



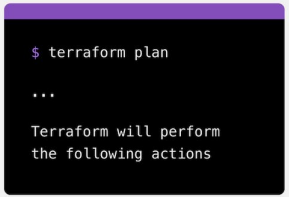
Write

Define infrastructure in configuration files



Plan

Review the changes Terraform will make to your infrastructure



Apply

Terraform provisions your infrastructure and updates the state file.