Infrastructure as Code - Terraform

What is terraform ?

HashiCorp Terraform is an infrastructure as code tool that lets you define both cloud and on-prem resources in human-readable configuration files that you can version, reuse, and share. You can then use a consistent workflow to provision and manage all of your infrastructure throughout its lifecycle. Terraform can manage low-level components like compute, storage, and networking resources, as well as high-level components like DNS entries and SaaS features.

What is HCL (HashiCorp Language) ?

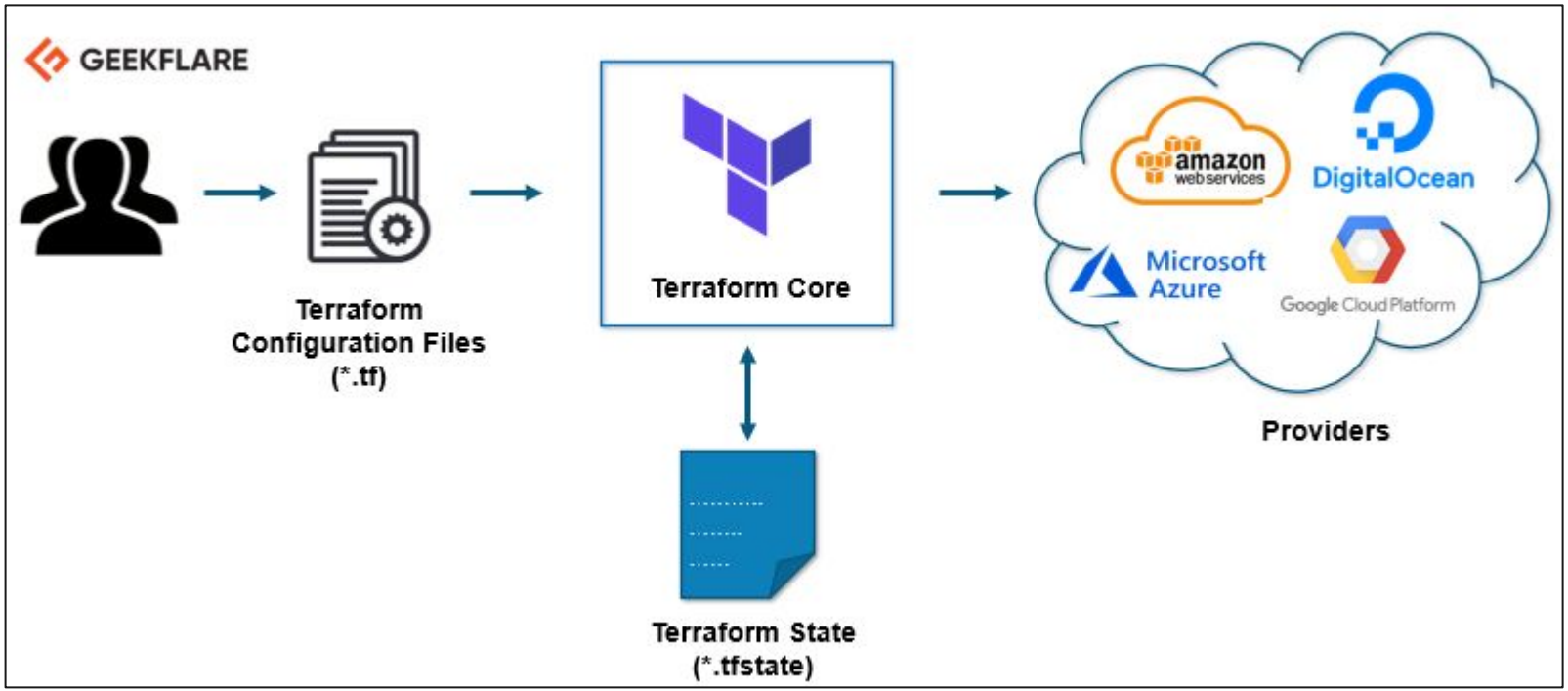
HashiCorp Configuration Language (HCL) is a unique configuration language. It was designed to be used with HashiCorp tools, notably Terraform, but HCL has expanded as a more general configuration language. It’s visually similar to JSON with additional data structures and capabilities built-in.

HCL

```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 3.0"
    }
  }
}

# Configure the AWS Provider
provider "aws" {
  region = "us-east-1"
}

# Create a VPC
resource "aws_vpc" "example" {
  cidr_block = "10.0.0.0/16"
}
```


GEEKFLARE
Terraform
Configuration Files
(*.tf)
Terraform Core
Terraform State
(*.tfstate)
amazon
web services
DigitalOcean
Microsoft
Azure
Google Cloud Platform
Providers

Terraform workflow

Assignment

Create a terraform config with any provider of choice (Docker if you prefer) and share the terraform config files (with .tf extension) and state file (with .tfstate extension)

Step 1 : Install terraform

Step 2 : Create tf config file to pull & create a docker container referring the documentation

Docs overview | kreuzwerker/docker | Terraform Registry

Step 3 : Terraform init, plan & apply