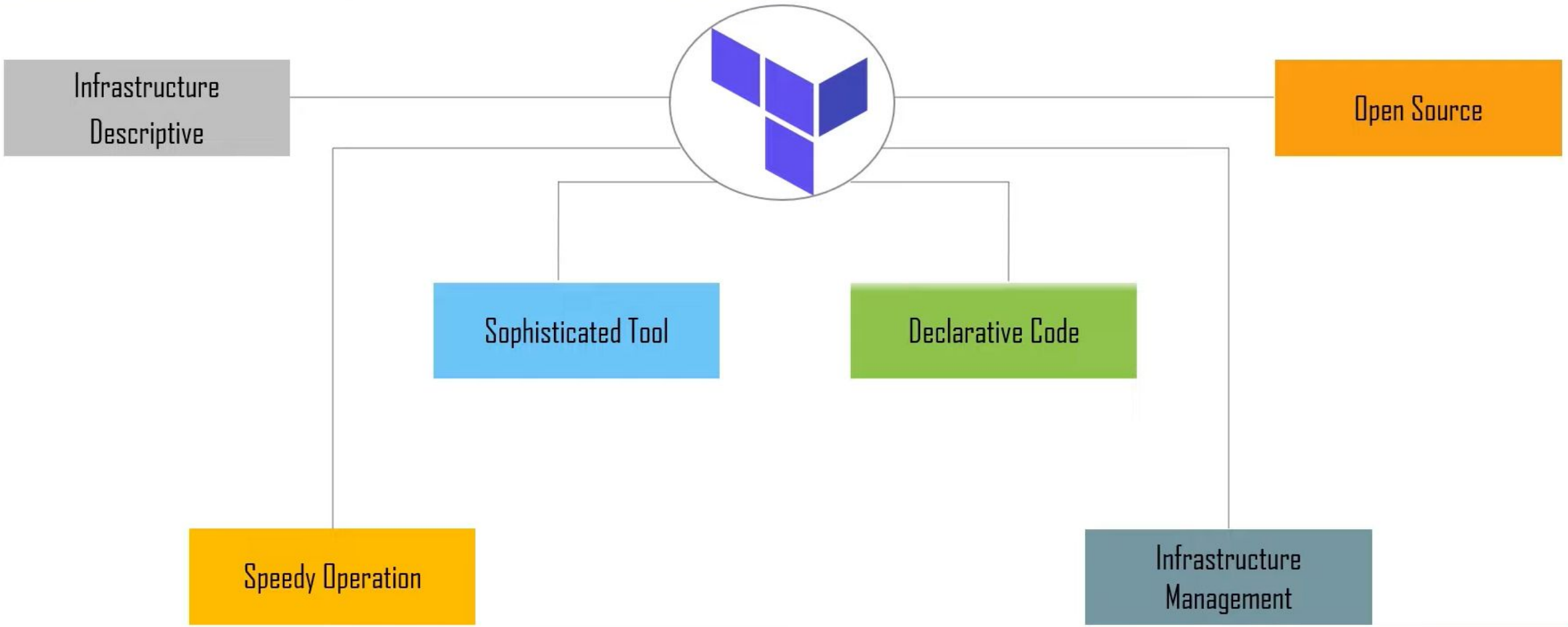

HashiCorp
Terraform

Why Terraform?



What is Terraform?

Terraform

Terraform is an open-source tool that lets you provision Google Cloud resources with declarative configuration files—resources such as virtual machines, containers, storage, and networking

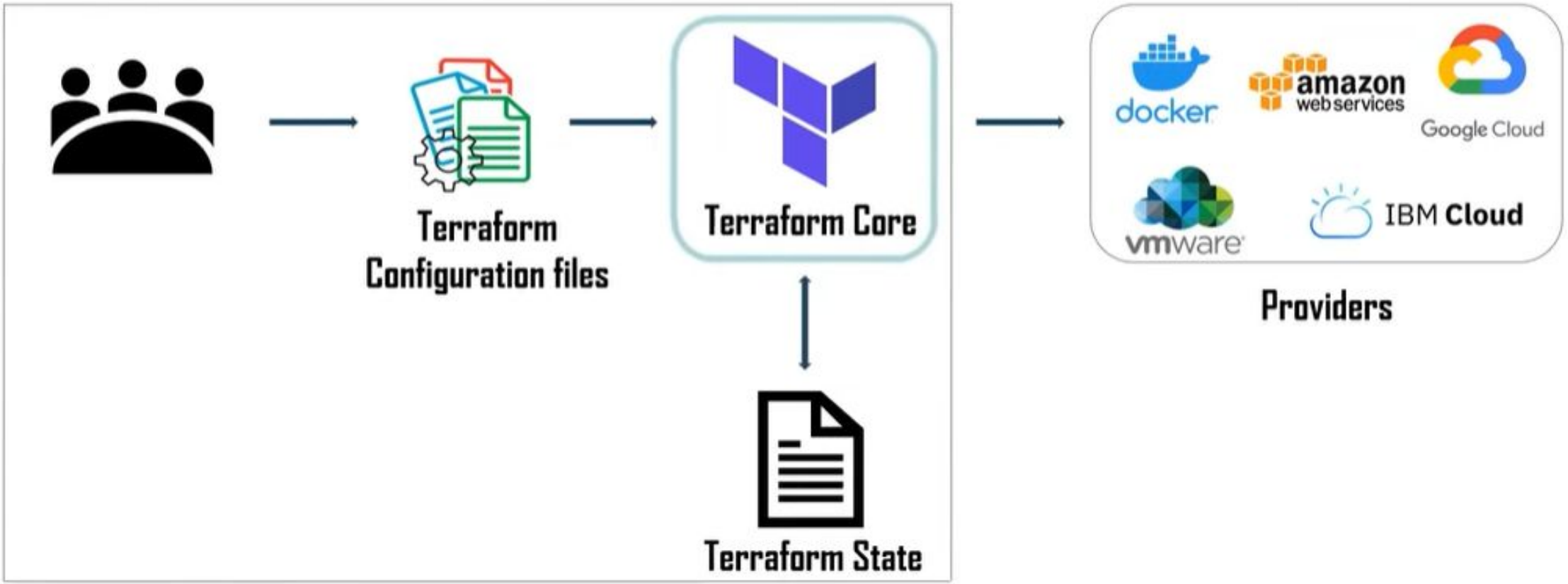
- automate and manage your infrastructure
- your platform
- and services that run on that platform

- ✓ open source
- ✓ declarative

Declarative = define **WHAT** end result you want

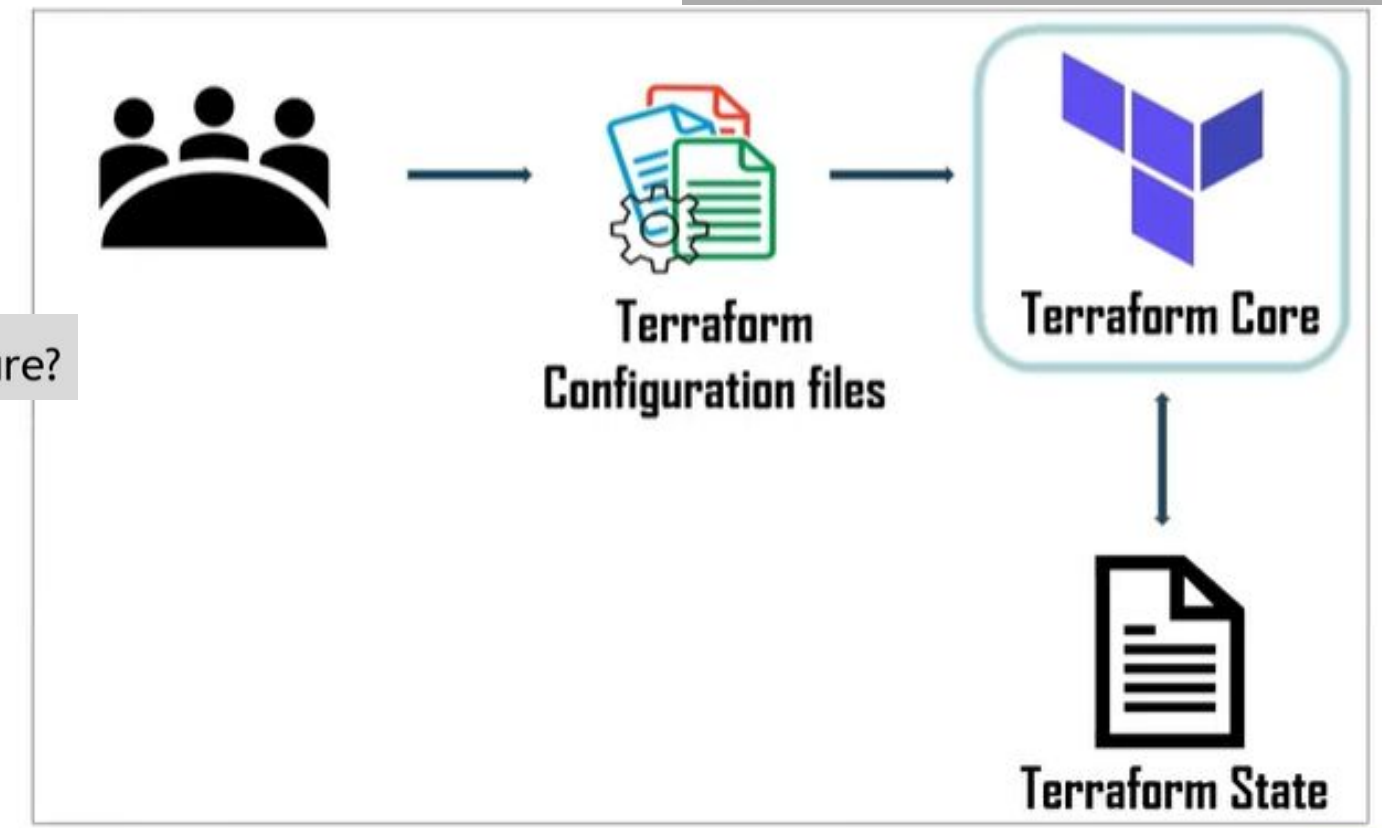
Imperative = define exact steps - **HOW**

How Terraform Works?



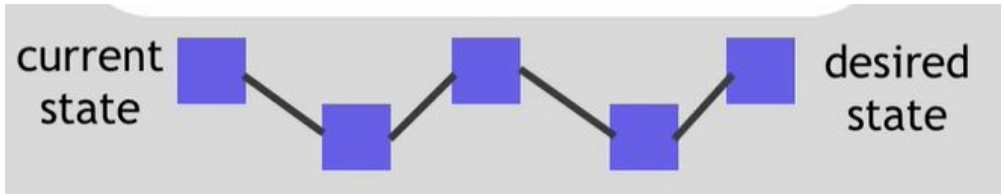
Terraform Core

Plan: What needs to be created/updated/destroyed?



Input Sources to Terraform core:

- Terraform configuration — What to create/configure?
- Terraform state — current state of setup



Example Configuration Files

```
# Configure the AWS Provider
provider "aws" {
  version = "~> 2.0"
  region  = "us-east-1"
}

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

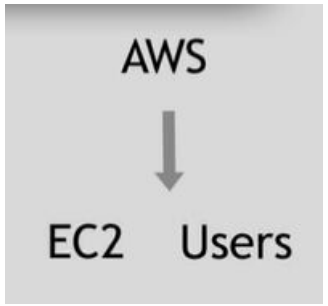
Example Configuration Files

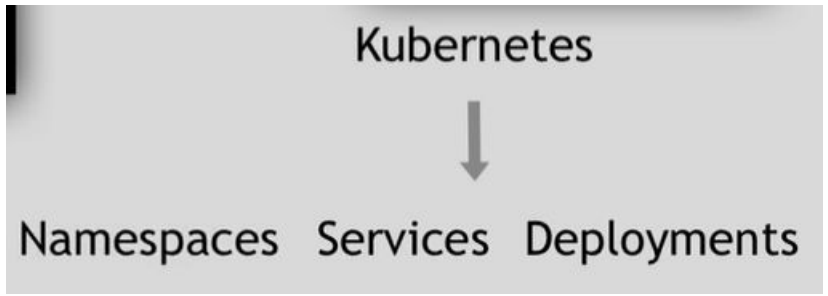
```
# Configure the Kubernetes Provider
provider "kubernetes" {
  config_context_auth_info = "ops"
  config_context_cluster   = "mycluster"
}

resource "kubernetes_namespace" "example" {
  metadata {
    name = "my-first-namespace"
  }
}
```

Providers

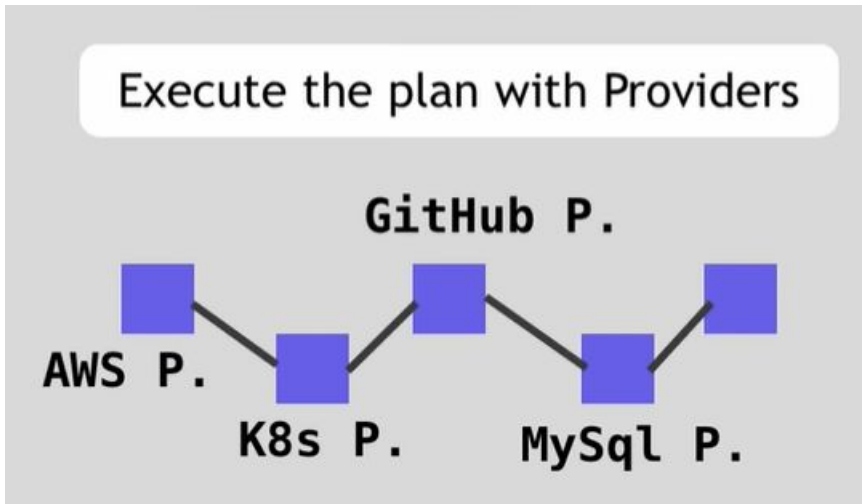
Google Cloud


AWS
K8s
Service


AWS
EC2
Users


Kubernetes
Namespaces
Services
Deployments


Complete Application Setup


Execute the plan with Providers
GitHub P.
AWS P.
K8s P.
MySql P.

Terraform Lifecycle

terraform init	It initializes the working directory which consists of all the configuration files
terraform validate	It validates the configuration files in a directory, referring only to the configuration and not accessing any remote services such as remote state, provider APIs, etc.
terraform plan	It is used to create an execution plan to reach a desired state of the infrastructure. Changes in the configuration files are done in order to achieve the desired state.
terraform apply	It makes the changes in the infrastructure as defined in the plan, and the infrastructure comes to the desired state.
terraform destroy	It is used to delete all the old infrastructure resources, which are marked tainted after the apply phase.

Terraform Lifecycle

terraform init

Usage:

- This command performs several different initialization steps in order to prepare the current working directory for use with terraform.
- This command is always safe to run multiple times, to bring the working directory up to date with changes in the configuration.

Terraform Lifecycle

terraform validate

Usage:

This command accepts the following options:

- json - Produce output in a machine-readable JSON format, suitable for use in text editor integrations and other automated systems.
- no-color - If specified, output won't contain any color.

Terraform Lifecycle

terraform plan

Usage:

The plan subcommand looks in the current working directory for the root module configuration.
Because the plan command is one of the main commands of Terraform, it has a variety of different options, described in the following sections.

- Planning Modes
- Planning Options
- Resource Targeting
- Other Options

Terraform Lifecycle

terraform apply

Usage:

The behavior of terraform apply differs significantly depending on whether you pass it the filename of a previously-saved plan file.

Modes:

- Automatic plan mode
- Saved Plan mode

Terraform Lifecycle

terraform destroy

Usage:

This command is just a convenience alias for the following command:

terraform apply -destroy

You can also create a speculative destroy plan, to see what the effect of destroying would be, by running the following command:

terraform plan -destroy

- What is Terraform | Terraform by HashiCorp
- Destroy Infrastructure | Terraform - HashiCorp Learn