## Resources and Modules

The main purpose of the Terraform language is declaring [resources](https://www.terraform.io/docs/configuration/resources.html). A group of resources can be gathered into a [module](https://www.terraform.io/docs/configuration/modules.html), which creates a larger unit of configuration. A resource describes a single infrastructure object, while a module might describe a set of objects and the necessary relationships between them in order to create a higher-level system.

A Terraform configuration consists of a root module, where evaluation begins, along with a tree of child modules created when one module calls another.

# Backends

A "backend" in Terraform determines how state is loaded and how an operation such as apply is executed. This abstraction enables non-local file state storage, remote execution, etc.By default, Terraform uses the "local" backend, which is the normal behavior of Terraform you're used to.

# Modules

A module is a container for multiple resources that are used together. Modules can be used to create lightweight abstractions, so that you can describe your infrastructure in terms of its architecture, rather than directly in terms of physical objects.

The .tf files in your working directory when you run [terraform plan](https://www.terraform.io/docs/commands/plan.html) or [terraform apply](https://www.terraform.io/docs/commands/apply.html) together form the root module. That module may [call other modules](https://www.terraform.io/docs/configuration/modules.html#calling-a-child-module) and connect them together by passing output values from one to input values of another.

# State

Terraform must store state about your managed infrastructure and configuration. This state is used by Terraform to map real world resources to your configuration, keep track of metadata, and to improve performance for large infrastructures.

This state is stored by default in a local file named "terraform.tfstate", but it can also be stored remotely, which works better in a team environment.

### components of Terraform

The logical division of Terraform into distinct structures refers to two distinct components. The two components are the Terraform Core and Terraform Plugins. In addition, Terraform Core also offers diverse ways of discovering and loading plugins according to requirements. The Terraform Plugins represent an implementation for a specific service such as bash or AWS or provisioner.