Get the best out of Live Sessions

HOW?

Check your Internet Connection

Log in 10 mins before, and check your internet connection to avoid any network issues during the LIVE session.

Speak with the Instructor

By default, you will be on mute to avoid any background noise. However, if required you will be **unmuted by instructor**.

Clear Your Doubts

Feel free to clear your doubts. Use the "**Questions**" tab on your webinar tool to interact with the instructor at any point during the class.

Let us know if you liked our content

Please share feedback after each class. It will help us to enhance your learning experience.

edureka!

edureka!

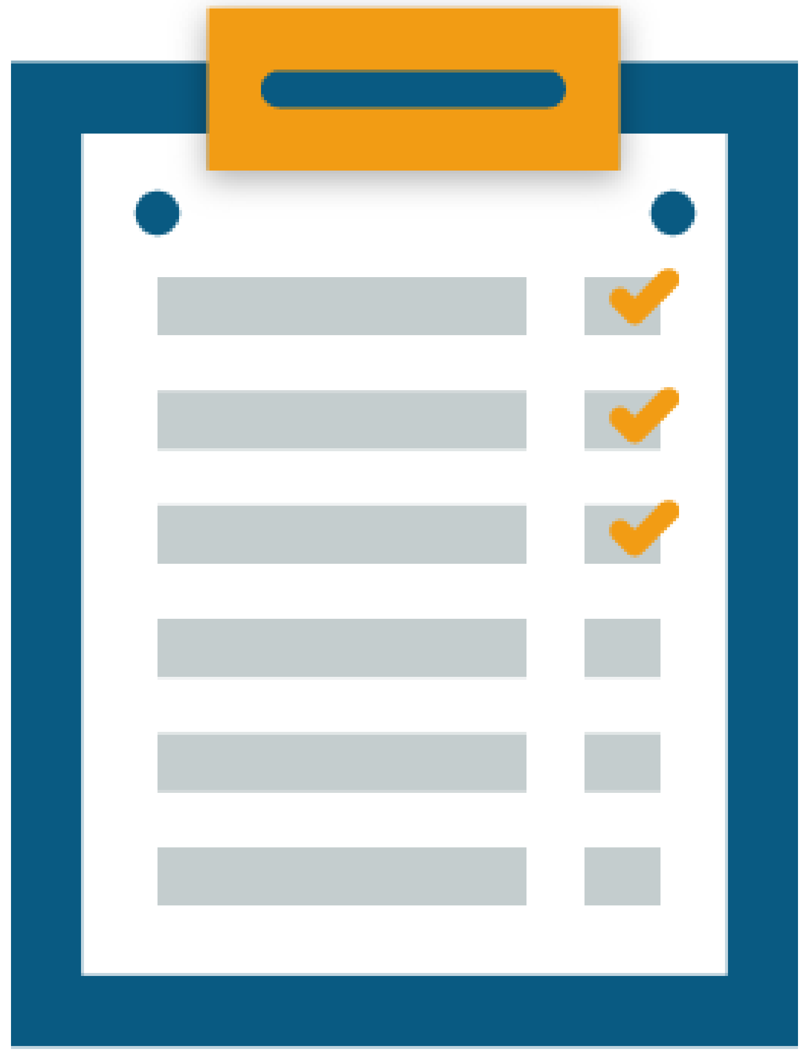
DevOps Certification Training

Module 11 – Provisioning Infrastructure using Terraform Part - I

Topics

Following are the topics covered in this module:

- Introduction to Terraform
- Terraform vs Ansible
- Terraform Architecture
- Terraform Configuration
- Terraform Common Commands
- Managing Terraform Resources



edureka!

Objectives

After completing this module, you should be able to:

- Understand Provisioning using Terraform
- Learn the Difference between Terraform vs Ansible
- Understand Terraform Architecture
- Deploy a Terraform Configuration File
- Use Basic Terraform Commands
- Manage Terraform Resources

Rapyder’s Approach to IaC using Terraform

edureka!

Rapyder

Rapyder is a Cloud Consultancy firm providing cloud-based infrastructure solutions

1

The company started as a consultancy helping run school competitions back in 1991

2

Their clientele includes the likes of Reliance, sify, lynk, Neogrowth, mapmyindia etc.

3

edureka!

Yelp: Monolithic Architecture

- Some of the clients required their infrastructure be spread across multiple cloud providers
- The infrastructure needs should be met in minutes
- Setting up database backups along with the infrastructure was a part of the setup

rapyder

Terraform to the Rescue

1 Terraform provided the solution to their problem by being cloud agnostic

2 Infrastructure could be setup from different vendors like aws and azure within minutes using Terraform

3 Managing Database backups also became quite painless with Terraform

4 Terraform enabled clients to test applications in production without any downtime

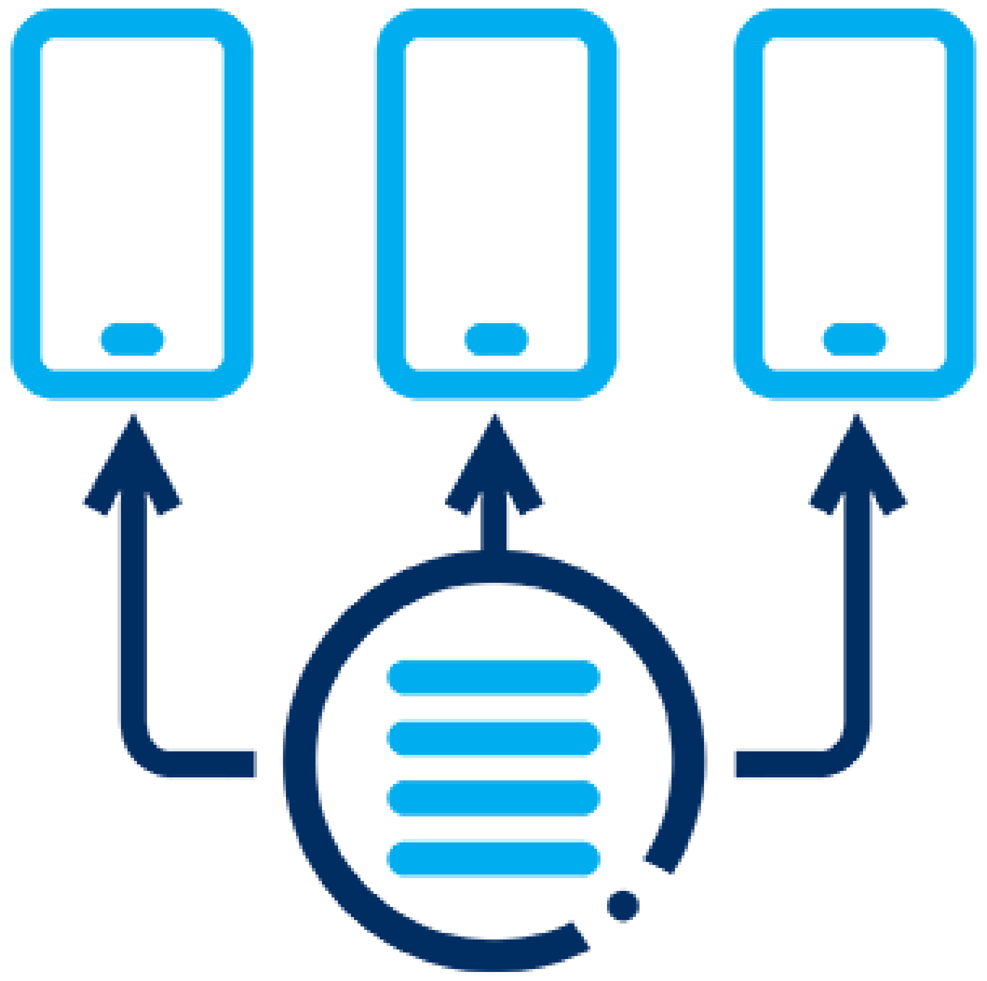
edureka!

Introduction to Terraform

edureka!

Provisioning

Provisioning involves providing the IT infrastructure to host services for your organization's business, users, employees and customers



Introduction to Terraform

Terraform is an Infrastructure as Code Software used for provisioning infrastructure safely and efficiently

HashiCorp
Terraform

Terraform: Features

IaC

Infrastructure as Code allows for users to describe their desired infrastructure as a high-level configuration syntax

Execution Plans

Execution plans allows you to preview the changes your configuration will make

Resource Graph

Terraform creates graphs to ensure that all the non-dependent resources are created parallelly

Change Automation

With the help of execution plans and resource graphs users know exactly what will change avoiding many human errors

Terraform vs Ansible

edureka!

Terraform vs Ansible

Mainly a infrastructure provisioning tool	Mainly a infrastructure provisioning tool
Advanced orchestration options	Relatively poor orchestration performance
Less mature because relatively new but has good community support	Much more mature with huge community support
Chances of configuration drift less	Higher chances of configuration drift

edureka!

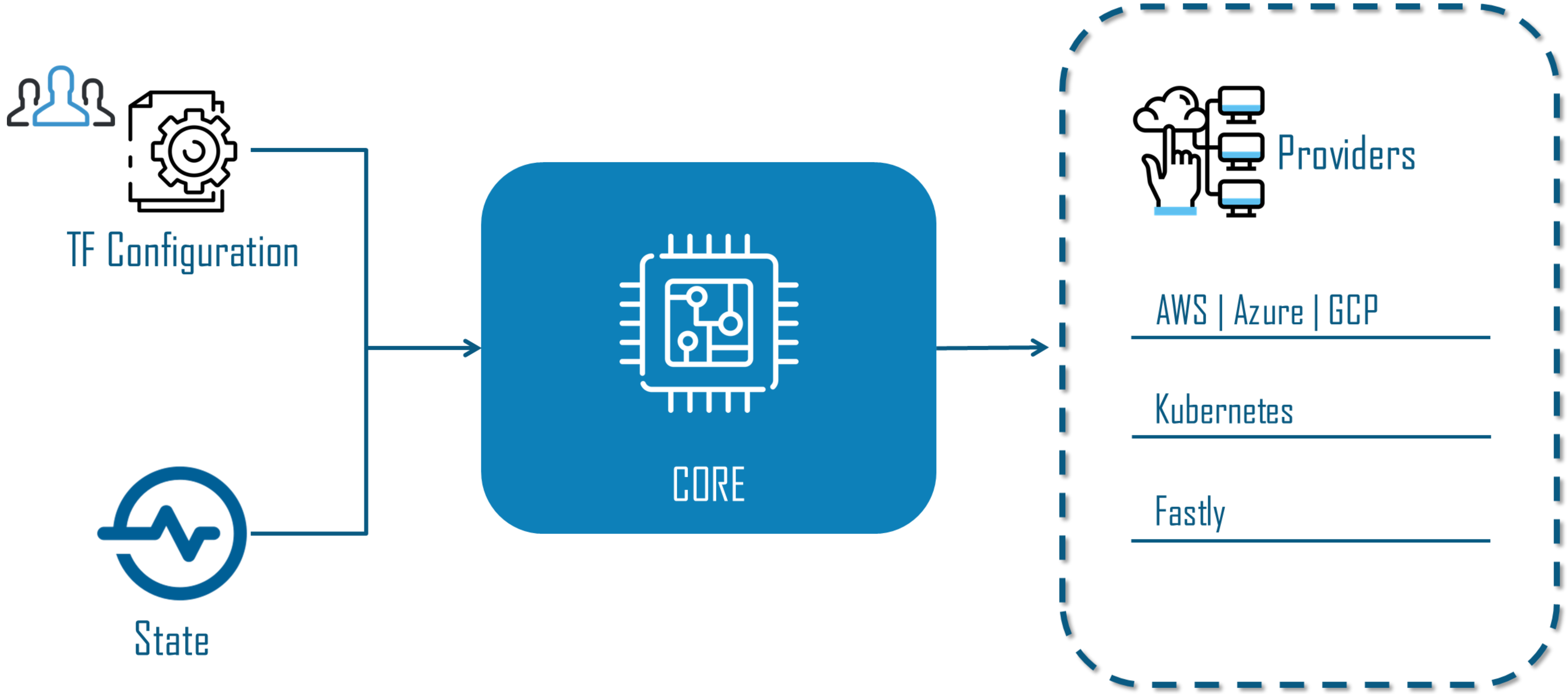
Terraform Architecture

edureka!

Terraform Architecture

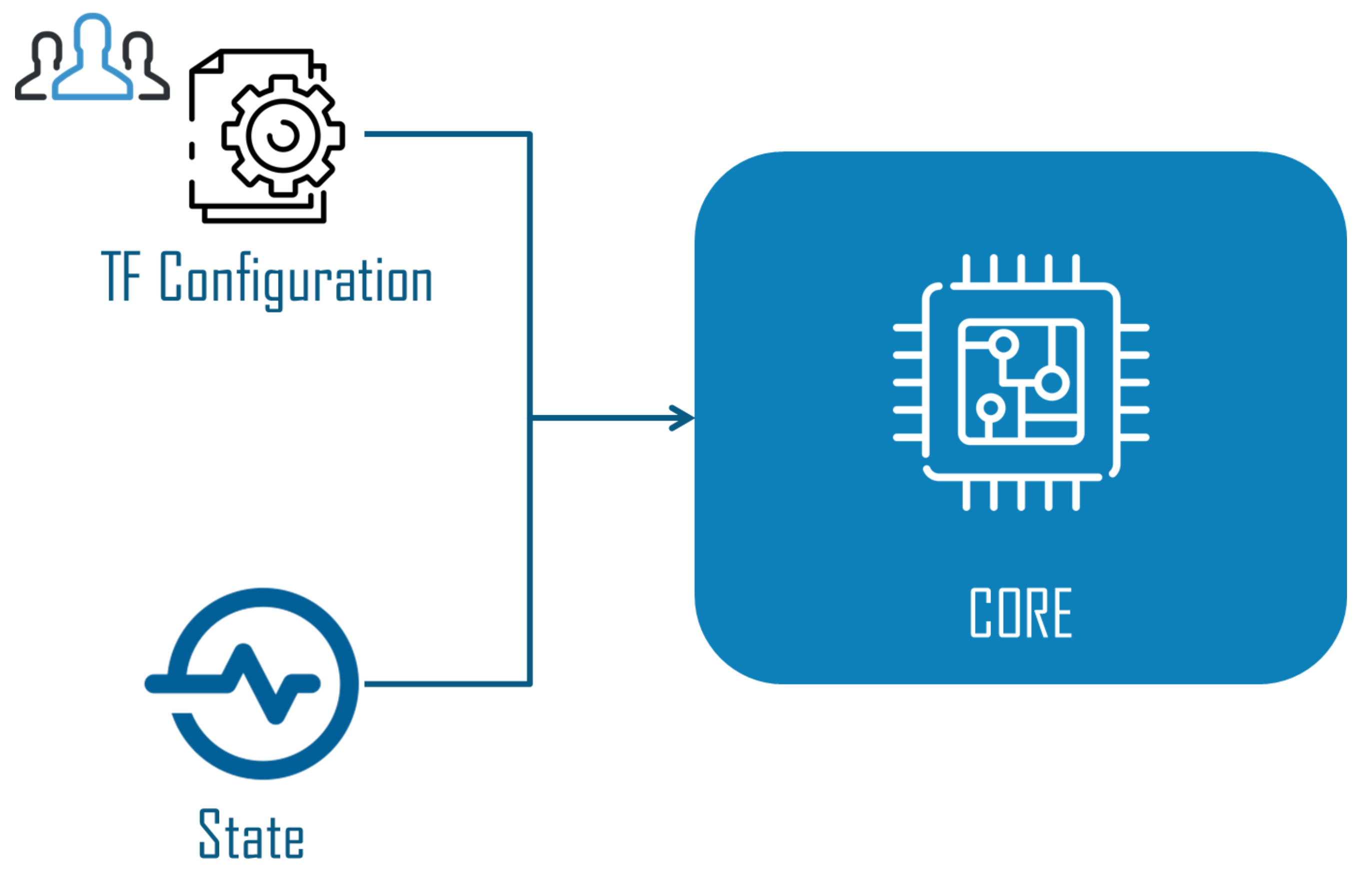
Terraform architecture constitutes of two main components

- Core
- Providers



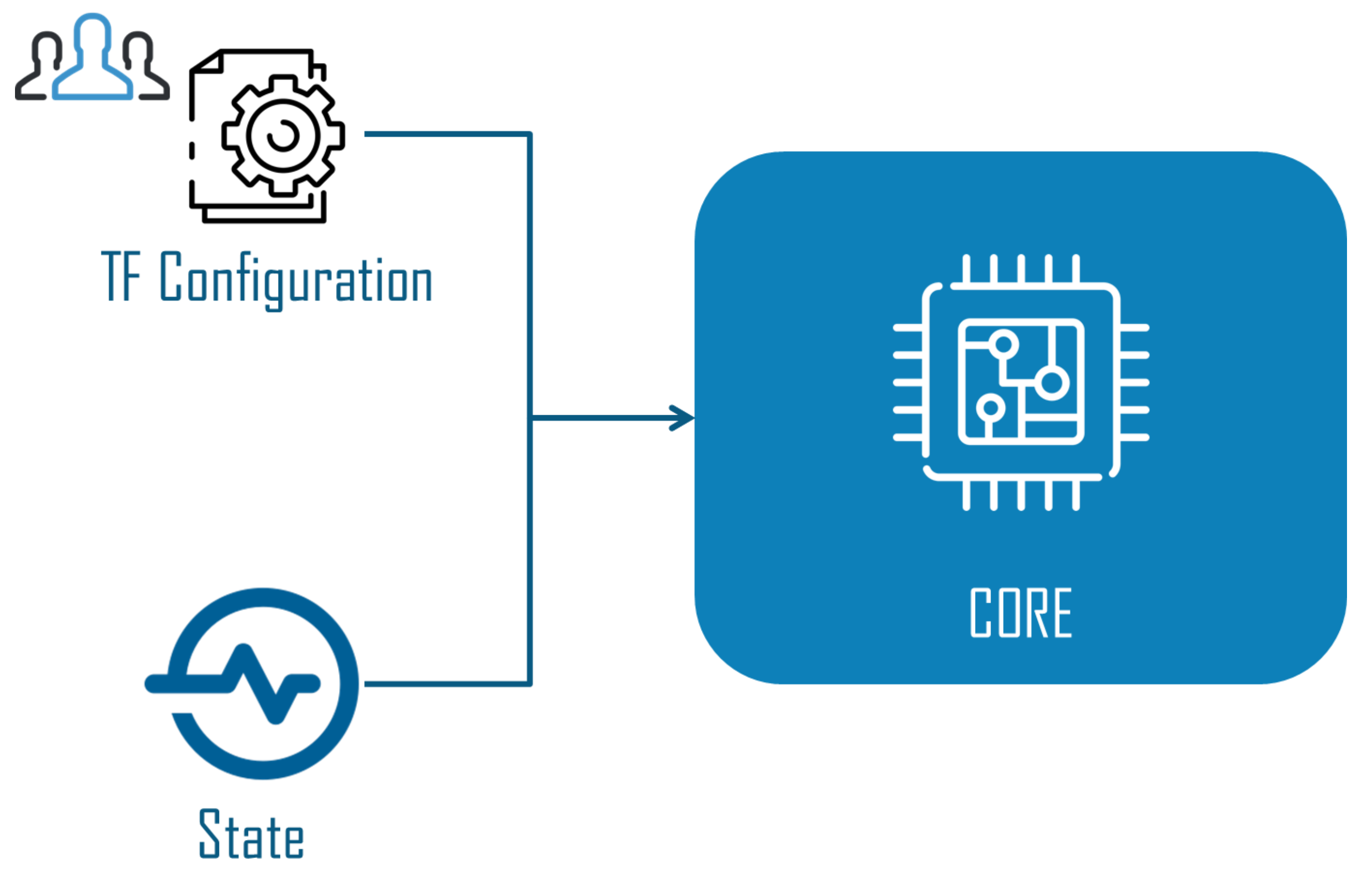
Terraform Architecture: Core

- Terraform's core is a statically-compiled binary written in golang
- Core takes input in the form of terraform configuration from the user and information from state
- Using the configuration file and the state information, core creates a plan to bring the system to the desired state



Terraform Architecture: Core

- The Terraform configuration is written by the user to define what needs to be done
- Terraform configuration declares resources, which represent the infrastructure objects
- State is a file that up-to-date information about the current setup of the infrastructure
- Before any new operation, terraform updates the state with the real infrastructure



Terraform Architecture: Providers

- The second main component of Terraform is the Provider Plugin
- The configuration defines which provider(s) to use for a particular setup
- Providers add resources which can be managed using terraform
- Most of the major providers can be found on the terraform registry

Providers

AWS | Azure | GCP

Kubernetes

Fastly

Demo: Setting up AWS for Terraform

Demo: Setting up Terraform on AWS

edureka!

Terraform Configuration

edureka!

Terraform Configuration

- Terraform configuration is the primary way of provisioning infrastructure using Terraform
- The configuration is written in the terraform language
- Terraform language is declarative in nature
- The configuration tells terraform how to manage and provision the infrastructure

edureka!

Terraform Configuration: Syntax elements

```
resource “aws_vpc” “main”{
   cidr_block = var.base_cidr_block
}

<BLOCK TYPE> “<BLOCK LABEL>” “<BLOCK LABEL>”{
   # Block body
   <IDENTIFIER> = <EXPRESSION> # Argument
}
```

Resource

- Resource blocks are used to define infrastructure objects
- Resource behavior defines how terraform handles resource declarations
- Meta-arguments provides special arguments that can be used with each resource type

edureka!

Terraform Configuration: Syntax elements

```
resource “aws_vpc” “main”{
   cidr_block = var.base_cidr_block
}

<BLOCK TYPE> “<BLOCK LABEL>” “<BLOCK LABEL>”{
   # Block body
   <IDENTIFIER> = <EXPRESSION> # Argument
}
```

Blocks

Blocks acts as containers for configuration of objects such as resources. It can have a block type, 0 or more labels and any number of arguments

Arguments

Arguments are inputs given to a name. They are used in blocks to

Expressions

These represent values. They can be direct values, referenced, or a set of values

edureka!

Demo: Writing a Terraform Configuration

edureka!

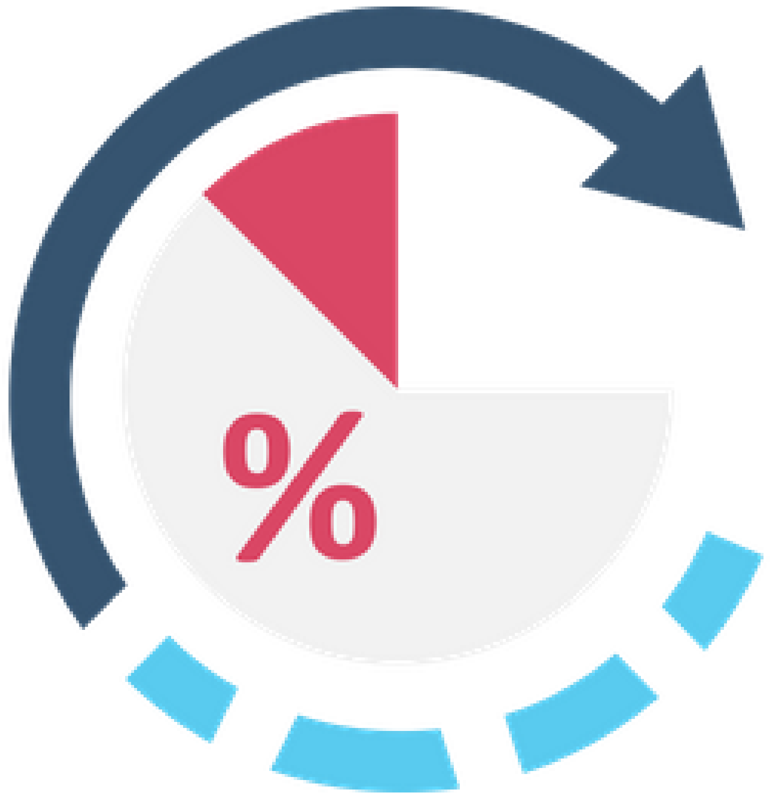
Terraform Basic Commands

Terraform Commands: Init

Init

- The `init` command is used to prepare the working directory for follow up commands
- This is the first command that is run after writing a new configuration
- The working directory must contain a configuration to run this command

]init[

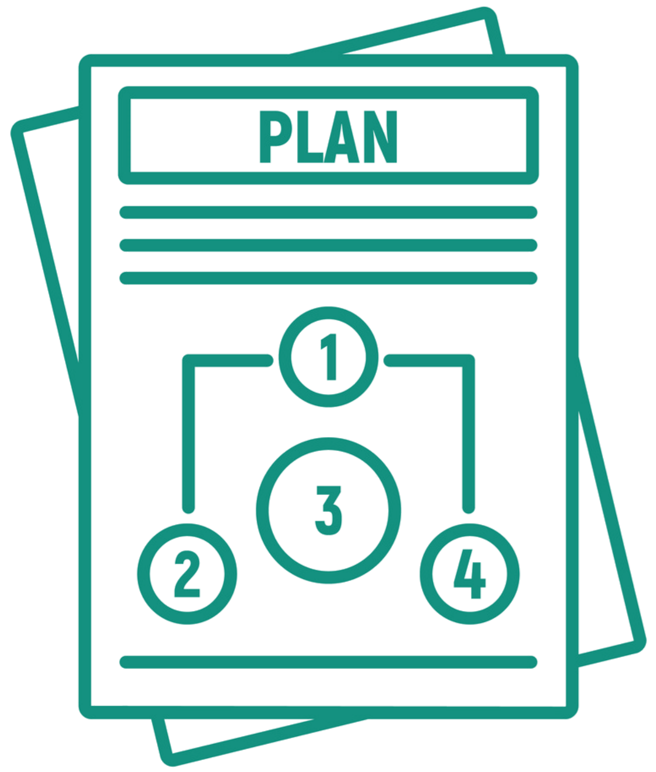


edureka!

Terraform Commands: Plan

Plan

- It displays the changes that will be made by the configuration
- It creates an execution plan by refreshing the state and comparing it with the required configuration
- It is equivalent to the dry run feature available in some of build softwares

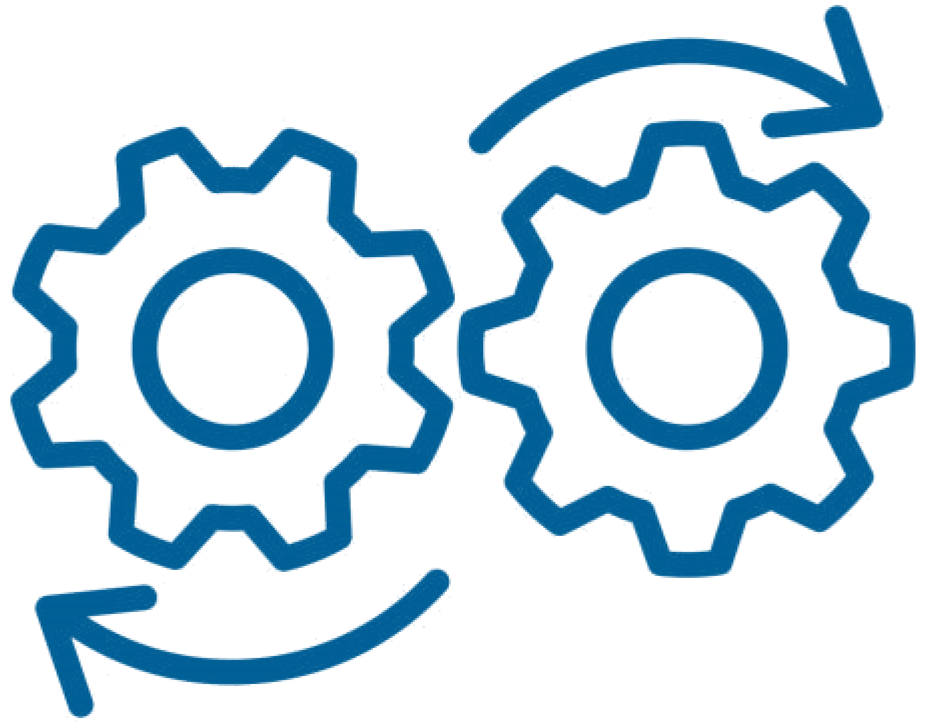


edureka!

Terraform Commands: Apply

Apply

- Apply as the name suggests applies the newly created configuration
- It can also be used to deploy the pre-determined set of actions generated in the execution plan



edureka!

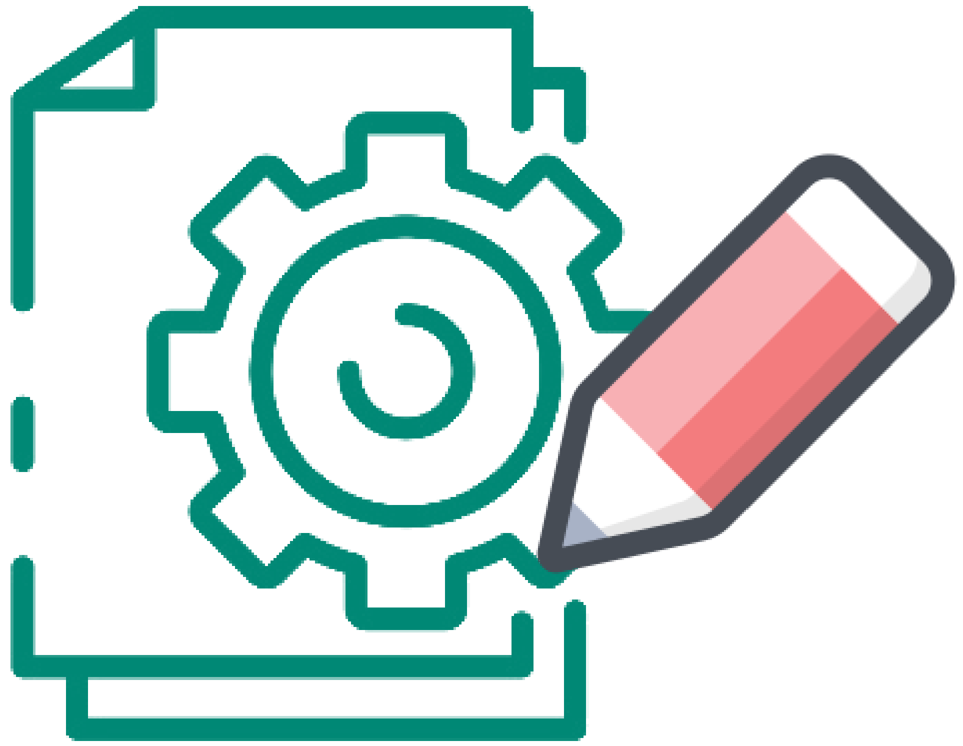
Demo: Creating and Running a new Terraform configuration

Managing Resources in Terraform

edureka!

Modifying Resources

- Modifying an existing resource can be done by simply making the changes to the existing terraform configuration and executing the apply command
- The changes can also be seen marked with a yellow tilde '~' sign in execution plan



Deleting Resources

- Deleting resources can be done by using the destroy command
- Simply running the destroy command terminates all the resources created by terraform
- In-order to delete a specific resource, parameters can be passed

edureka!

Demo: Managing Resources in Terraform

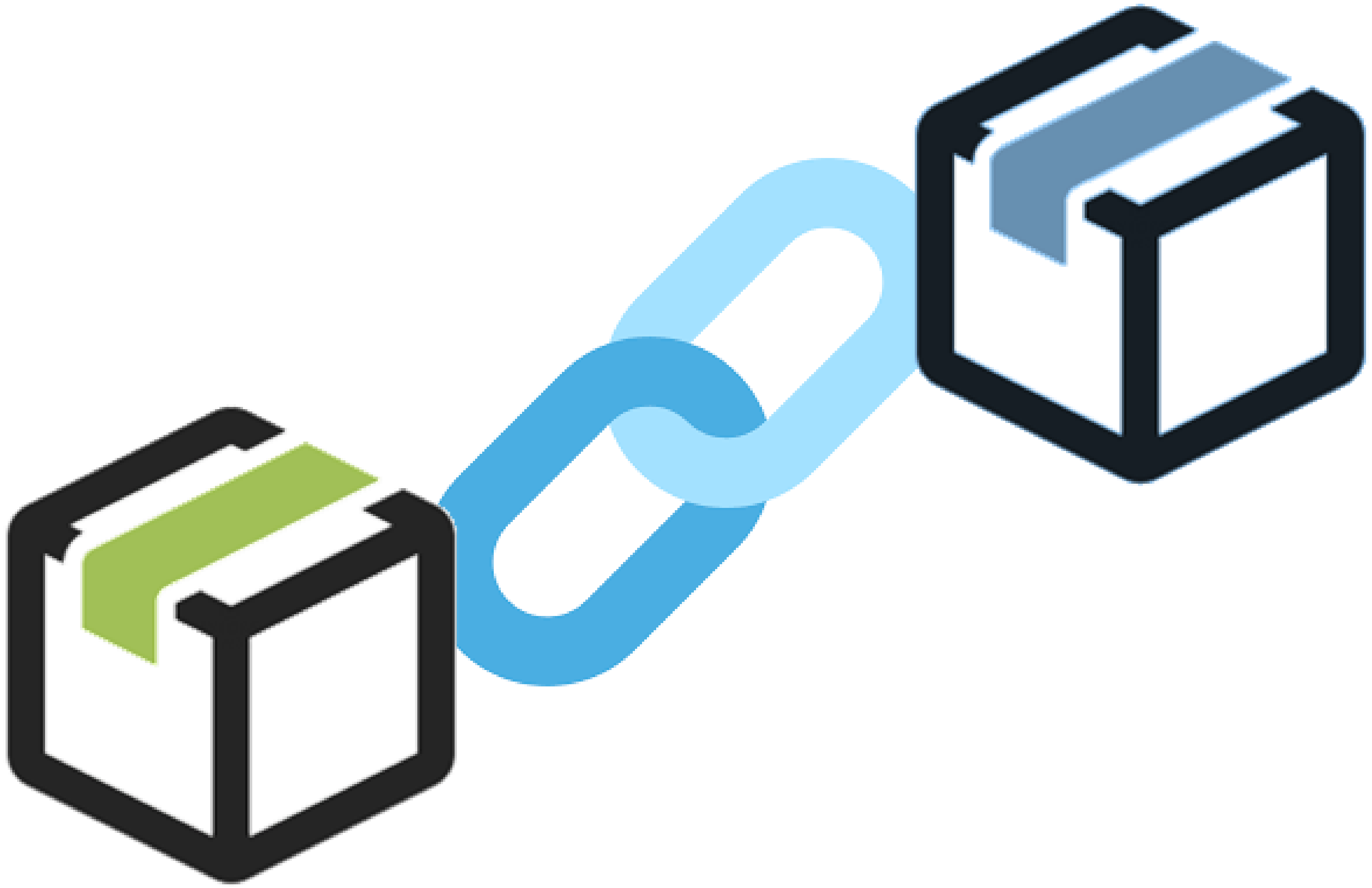
edureka!

Referencing Resources in Terraform

edureka!

Referencing Resources

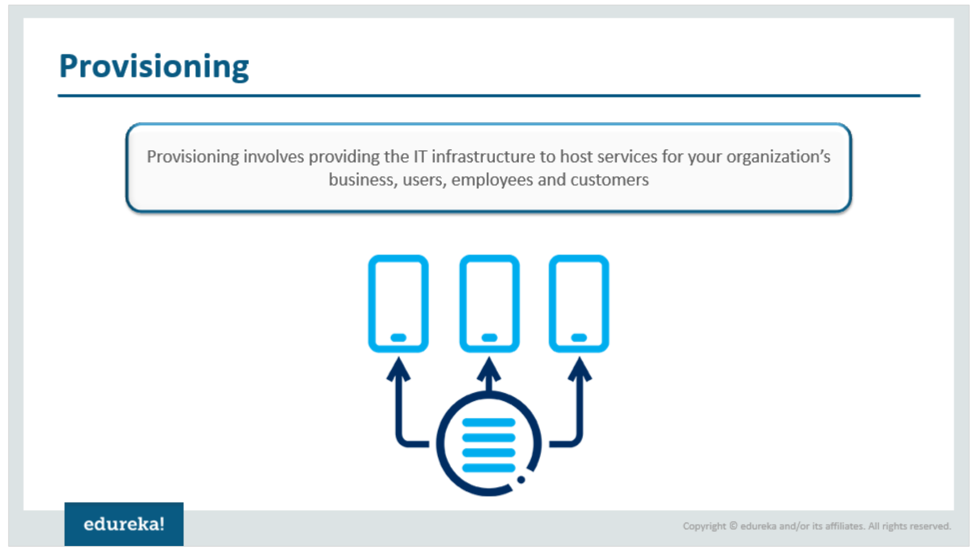
Terraform gives you the option to replicate infrastructure by referencing existing resources



edureka!

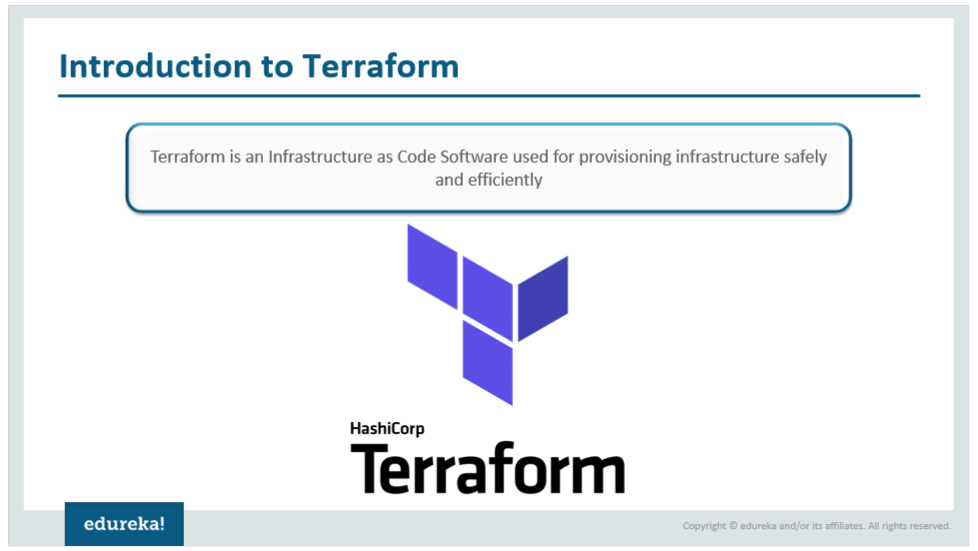
Demo: Managing Resources in Terraform

Summary



Provisioning

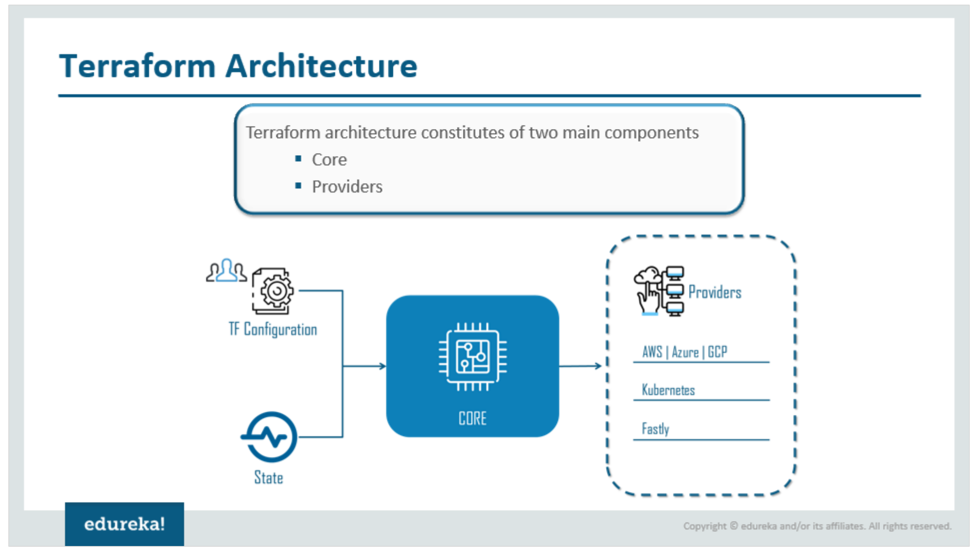
Provisioning involves providing the IT infrastructure to host services for your organization's business, users, employees and customers



Introduction to Terraform

Terraform is an Infrastructure as Code Software used for provisioning infrastructure safely and efficiently

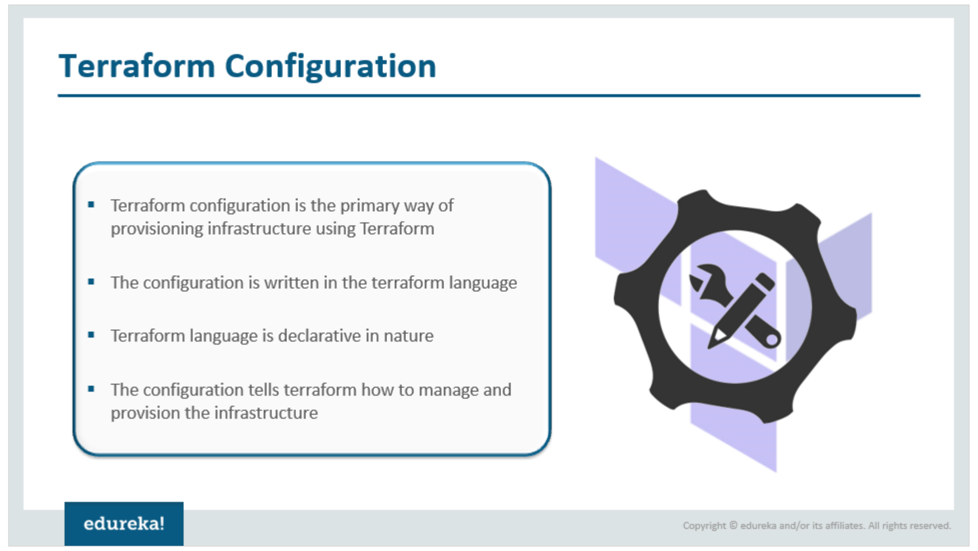
HashiCorp Terraform



Terraform Architecture

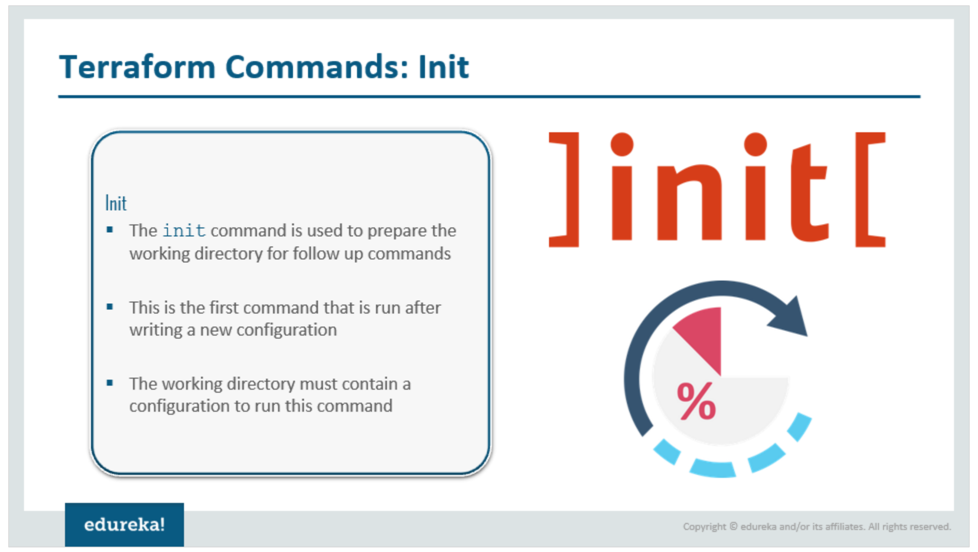
Terraform architecture constitutes of two main components
- Core
- Providers

TF Configuration
State
CORE
Providers
AWS | Azure | GCP
Kubernetes
Fastly



Terraform Configuration

- Terraform configuration is the primary way of provisioning infrastructure using Terraform
- The configuration is written in the terraform language
- Terraform language is declarative in nature
- The configuration tells terraform how to manage and provision the infrastructure

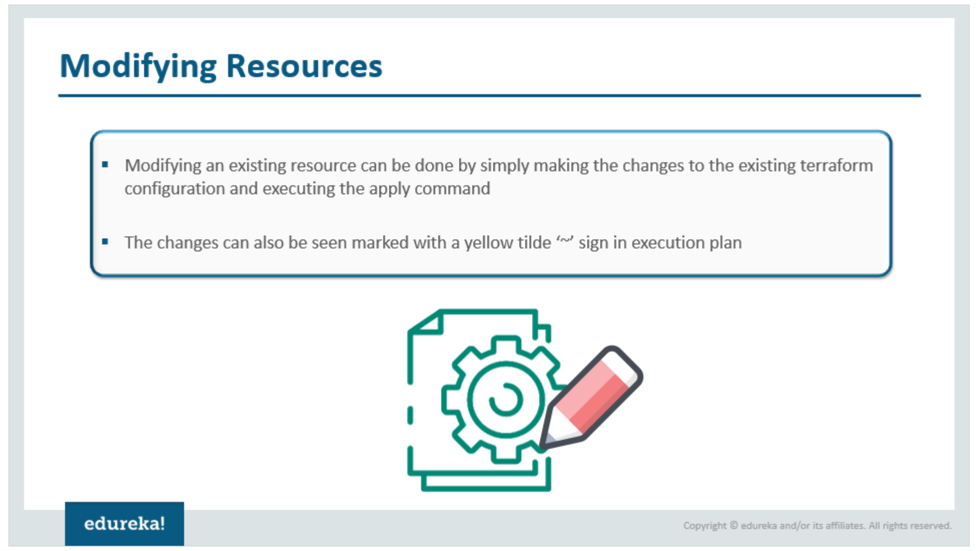


Terraform Commands: Init

Init
- The `init` command is used to prepare the working directory for follow up commands
- This is the first command that is run after writing a new configuration
- The working directory must contain a configuration to run this command

]init[

%



Modifying Resources

- Modifying an existing resource can be done by simply making the changes to the existing terraform configuration and executing the apply command
- The changes can also be seen marked with a yellow tilde '~' sign in execution plan

Questions

edureka!

Comments

Ratings

Suggestions

FEEDBACK

Survey

Likes

Ideas

edureka!

Thank You

For more information please visit our website
www.edureka.co