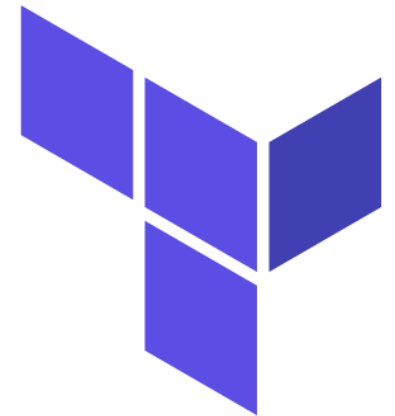Jitendra Singh Tomar

Index

- Introduction

- What is Terraform?

- Why Terraform?

- Terraform Architecture

- How Terraform Works

- Terraform Basics, Extensions

- Terraform Config files

- Terraform Commands

- Demo

What is Terraform?

- Terraform is an open-source infrastructure as code (IAC) tool that allows you to define and manage your cloud resources declaratively, using a high-level configuration language.

- Terraform is a powerful and flexible tool for managing cloud infrastructure that can help you streamline your workflows, reduce costs, and increase the agility and reliability of your systems.
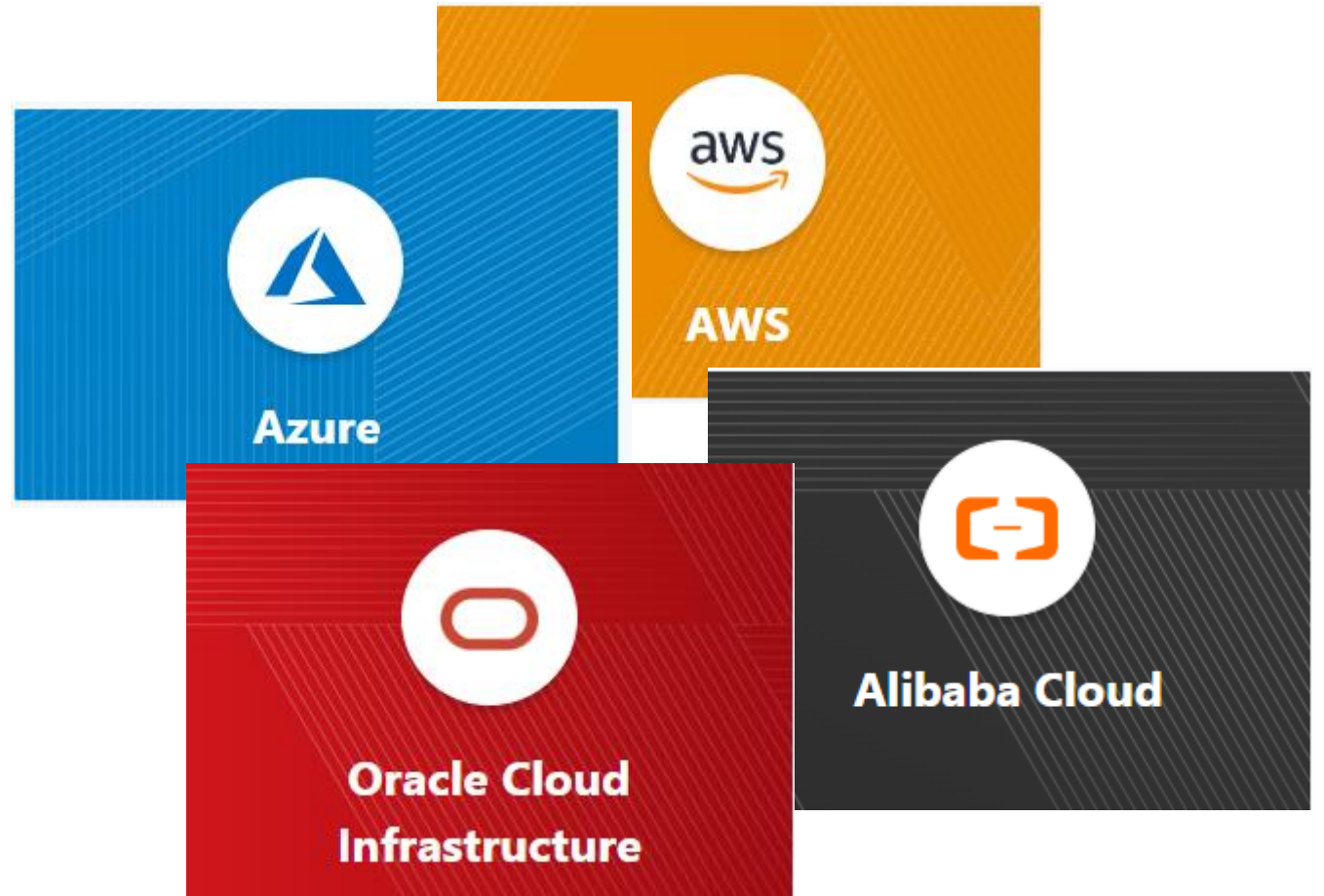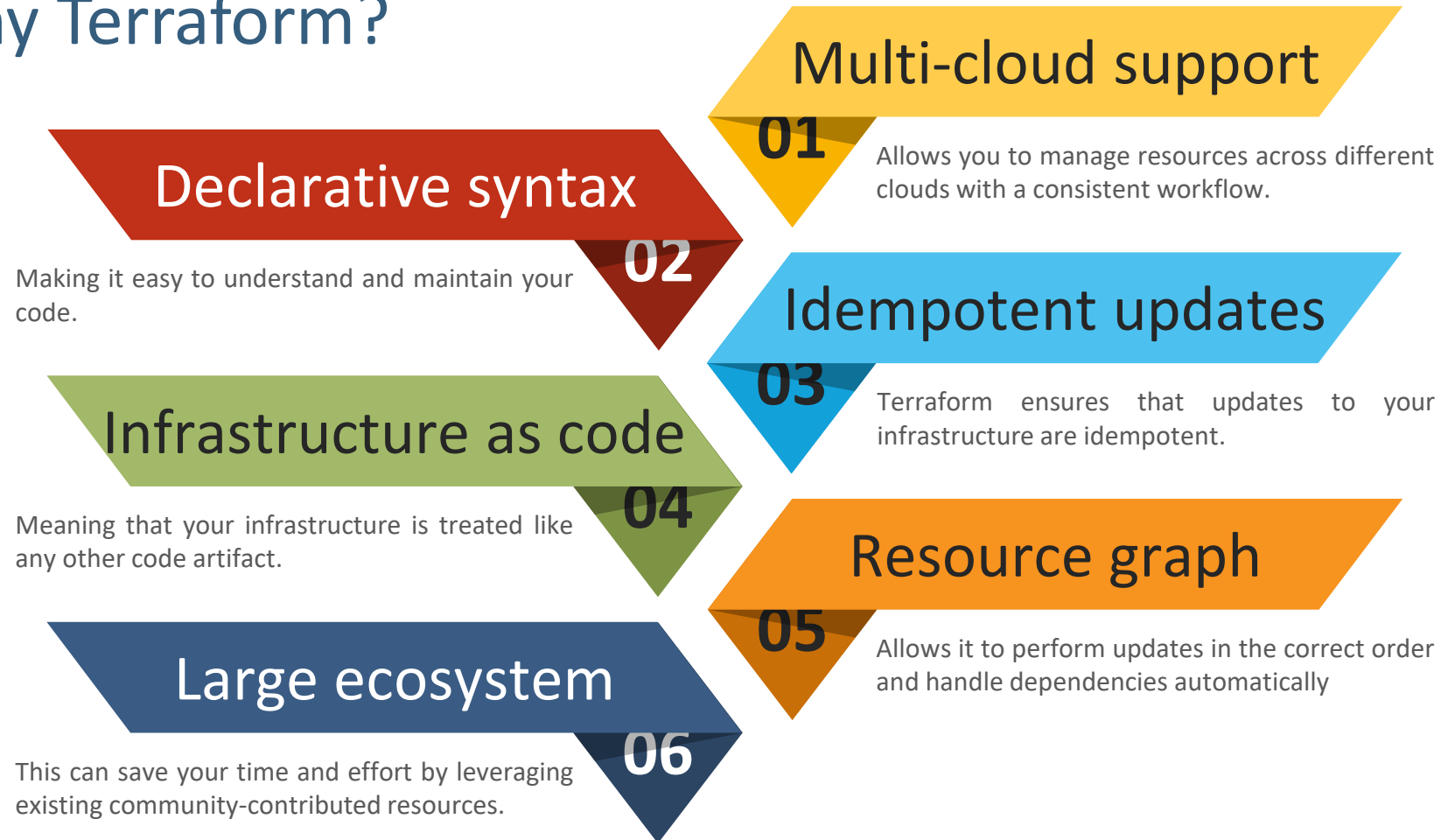
With Terraform

You can
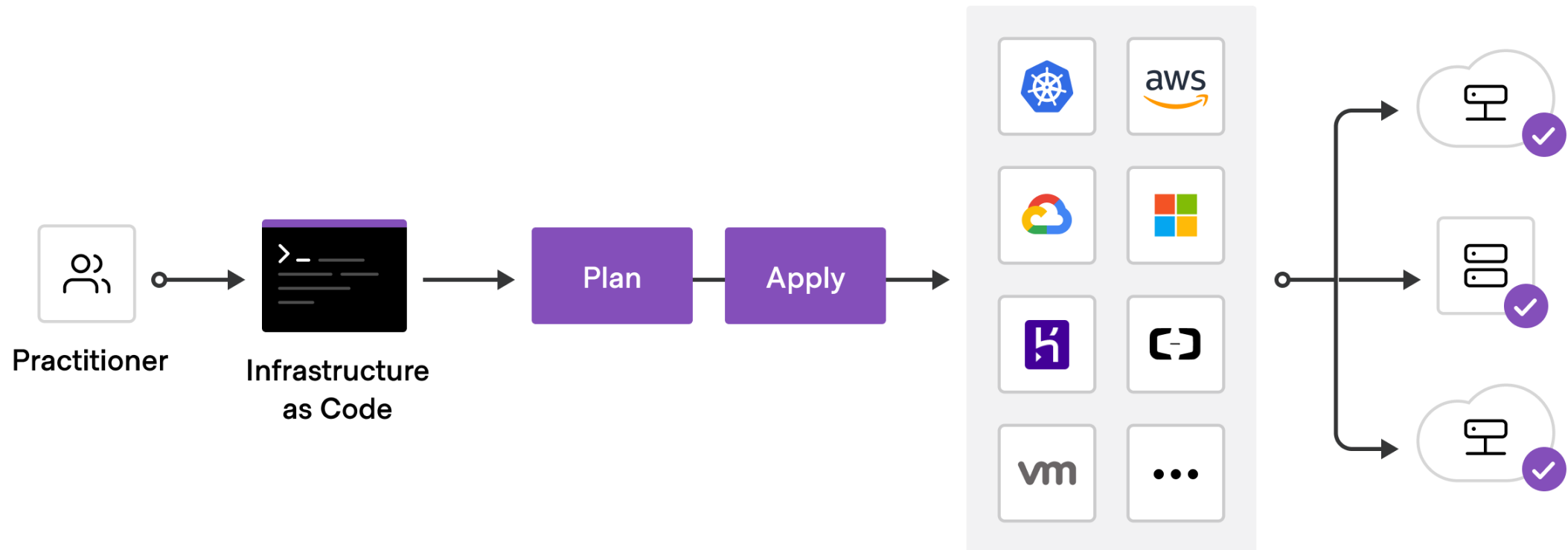- Create
- Modify
- Destroy

Cloud resources like
- Public cloud providers
- Private cloud environments
- Hybrid cloud environments
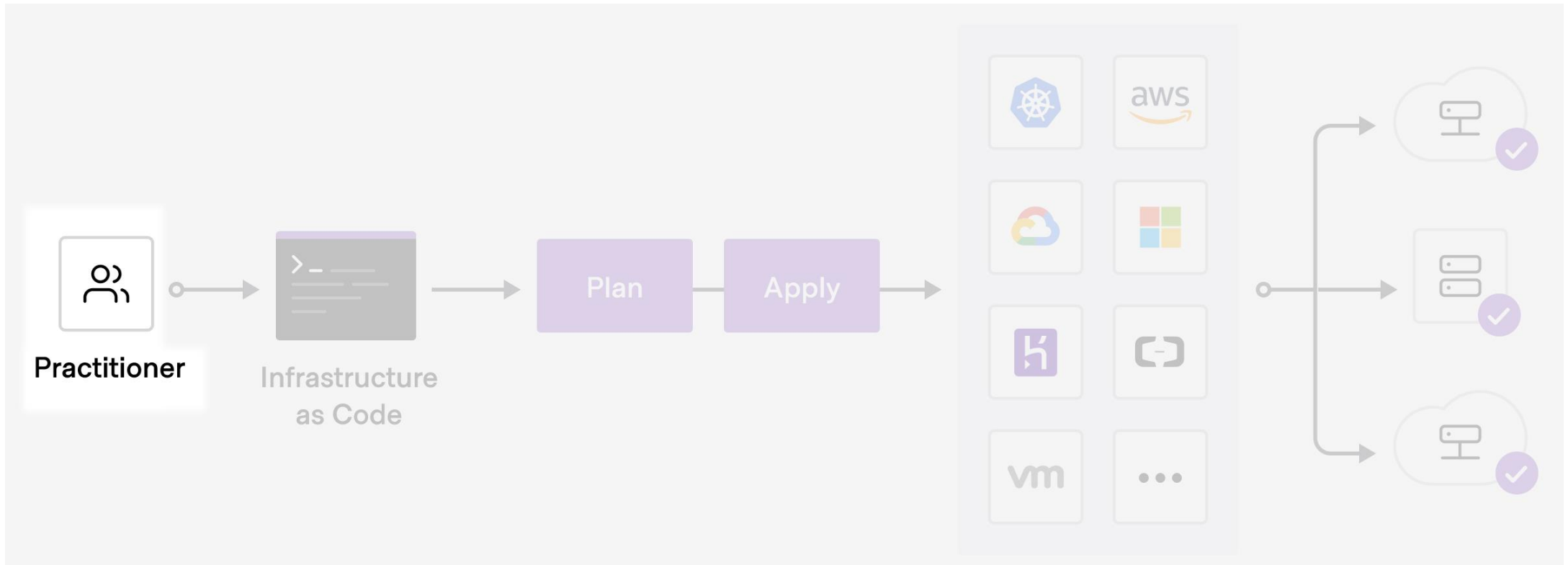- Networking infrastructure
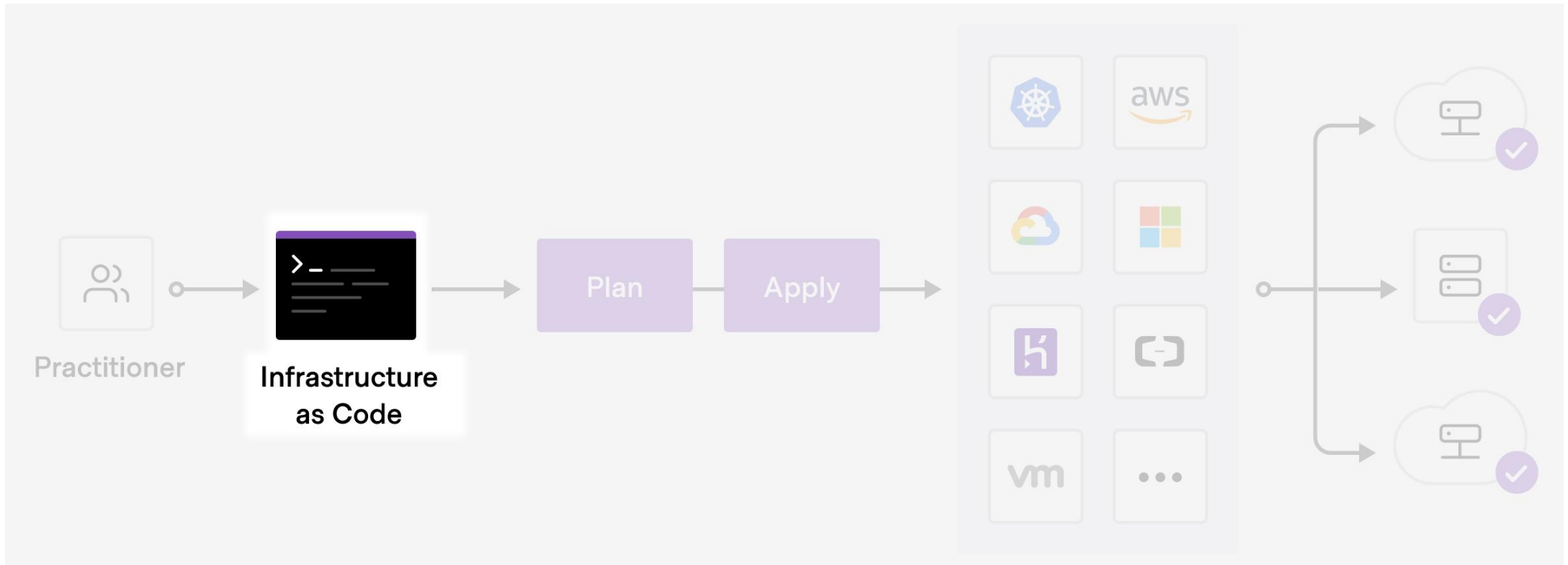- Databases, Containers etc…

Why Terraform?

Multi-cloud support

01 Allows you to manage resources across different clouds with a consistent workflow.

Declarative syntax

02 Making it easy to understand and maintain your code.

Idempotent updates

03 Terraform ensures that updates to your infrastructure are idempotent.

Infrastructure as code

04 Meaning that your infrastructure is treated like any other code artifact.

Resource graph

05 Allows it to perform updates in the correct order and handle dependencies automatically

Large ecosystem

06 This can save your time and effort by leveraging existing community-contributed resources.
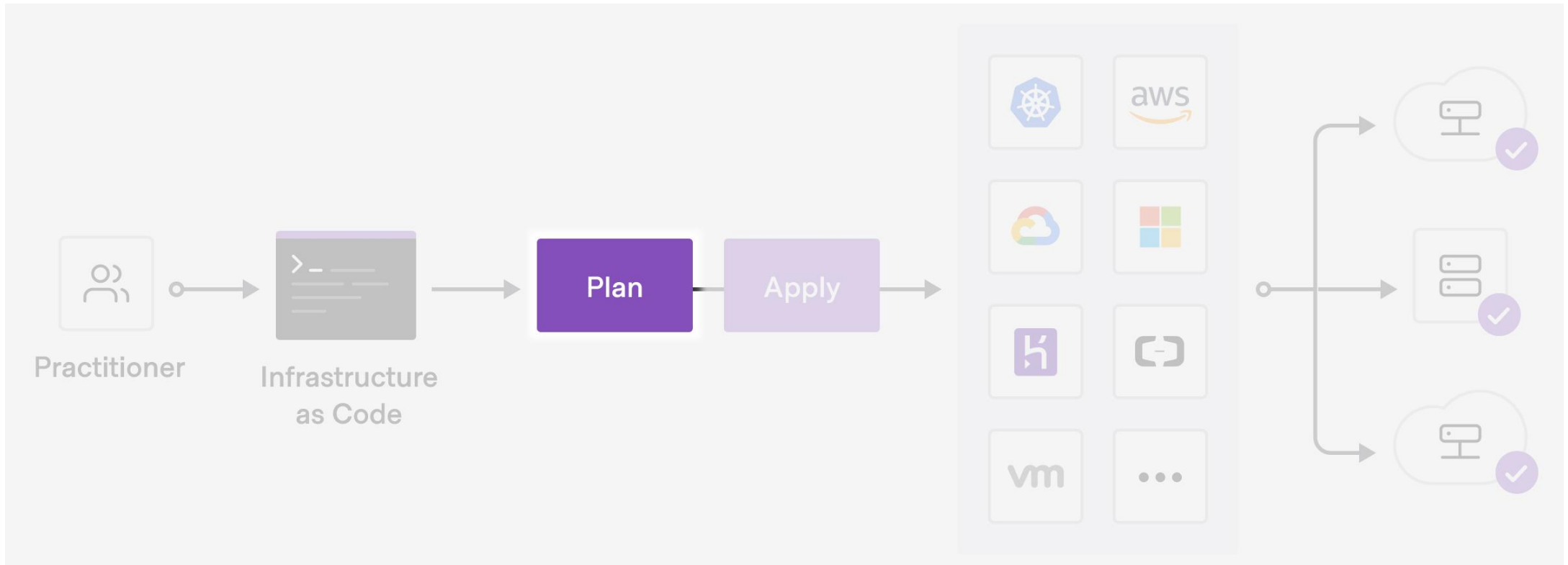
Terraform Architecture
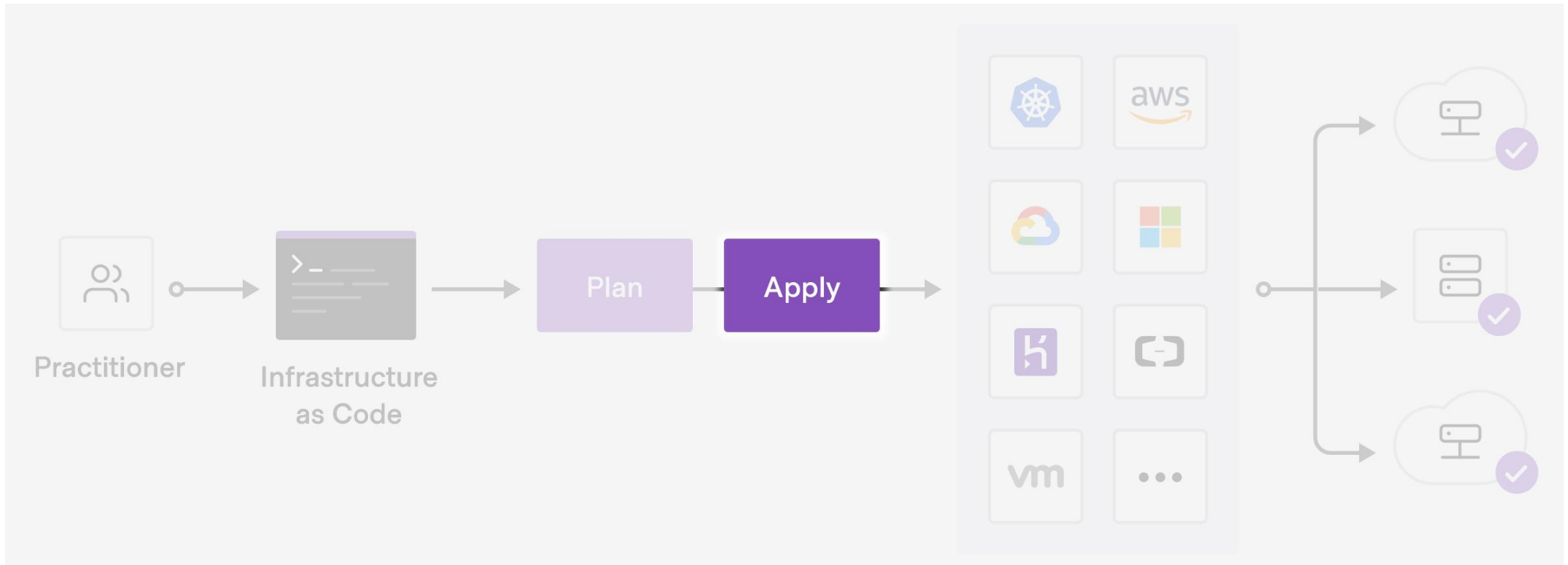
Terraform Architecture

Terraform Architecture

Terraform Architecture
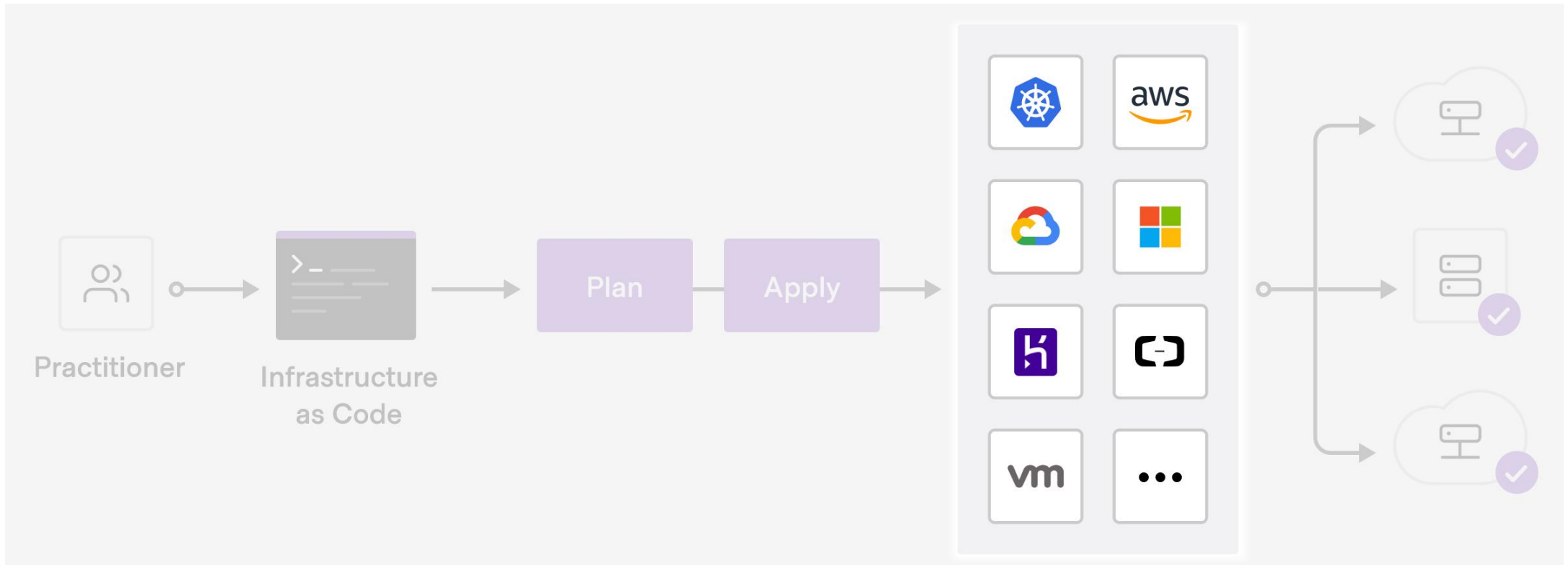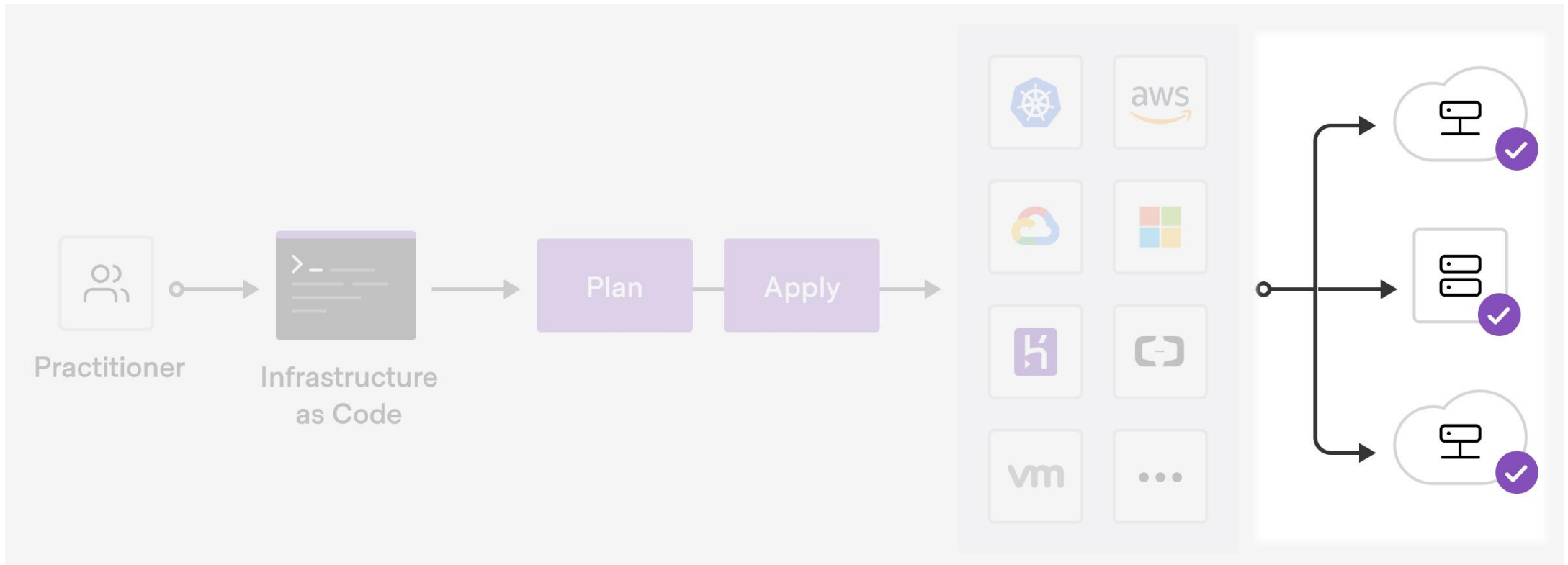
Terraform Architecture

Terraform Architecture

Terraform Architecture

How Terraform Works?
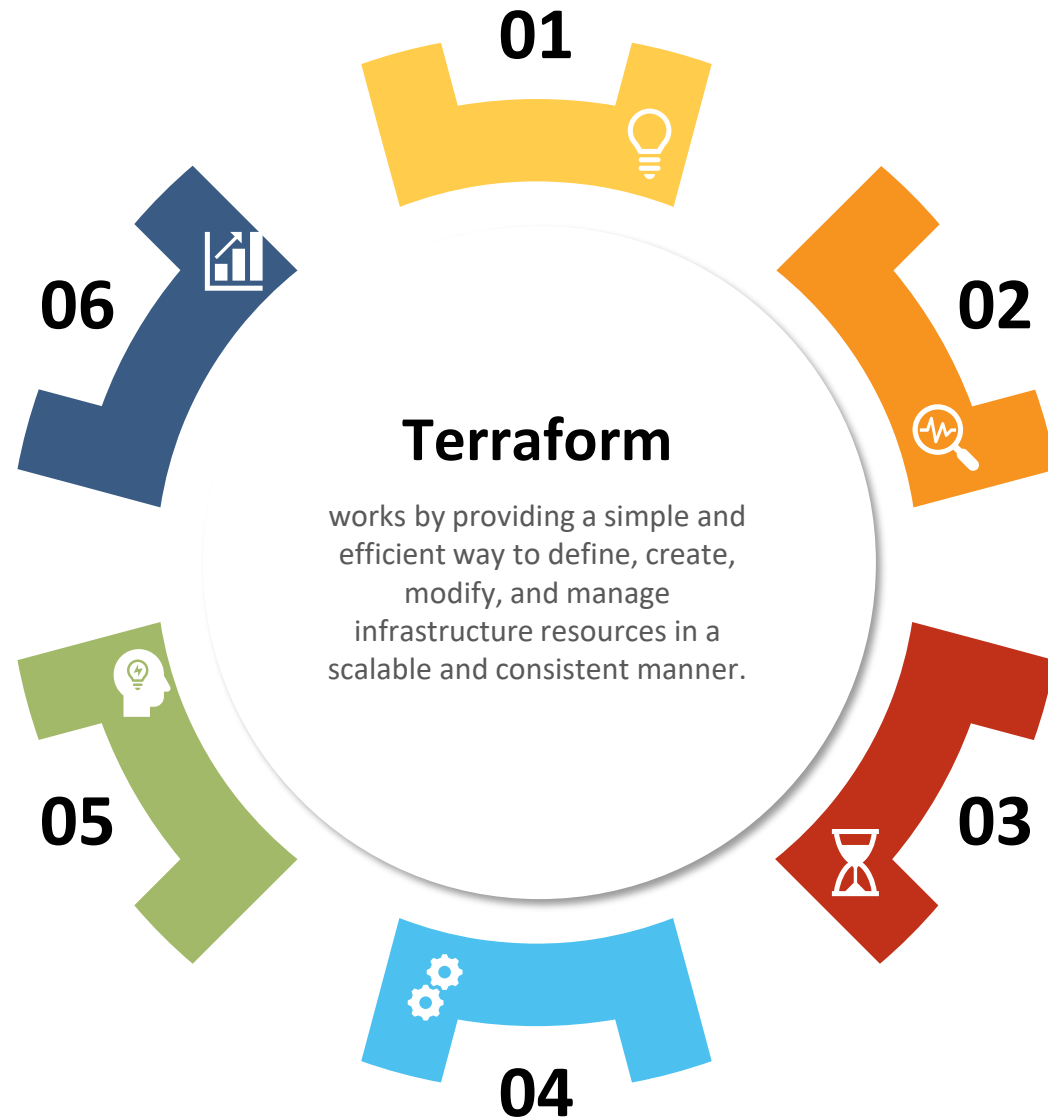
Continuous Management

Terraform continuously monitors the state of the infrastructure resources and updates them as necessary to maintain the desired state.

State Management

The state file is also used to track changes over time and to identify differences between the desired and actual state.

Execution

Terraform manages the dependencies between resources and ensures that they are created in the correct order.

01

06

Terraform

works by providing a simple and efficient way to define, create, modify, and manage infrastructure resources in a scalable and consistent manner.

05

04

02

03

Configuration File

The configuration file specifies the provider to use, the resources to create, and their desired state.

Initialization

This is done using the "**terraform init**" command, which downloads the necessary provider plugins and modules.

Planning

The plan includes the actions that will be taken, such as creating new resources, modifying existing resources, or destroying resources.

Terraform Basics

Configuration **1**

Providers **2**

Resources **3**

Modules **4**

Terraform Basics

5 State Management

6 Command-line Interface

7 Plan

8 Apply

Terraform Config File extensions

.tf
.TF is the primary extension used in Terraform for configuration files.

.tfvars
These files contain variable definitions for your Terraform configuration.

.tfstate
These files store the current state of the infrastructure resources managed by Terraform.

.tfplan
They contain a summary of the changes that Terraform plans to make to your infrastructure resources.

.tfignore
These files are used to specify files or directories that should be ignored by Terraform

Terraform Commands

1 terraform init

2 terraform plan

3 terraform apply

4 terraform destroy

Terraform cmds

5 terraform validate

6 terraform state

7 terraform output

8 terraform refresh