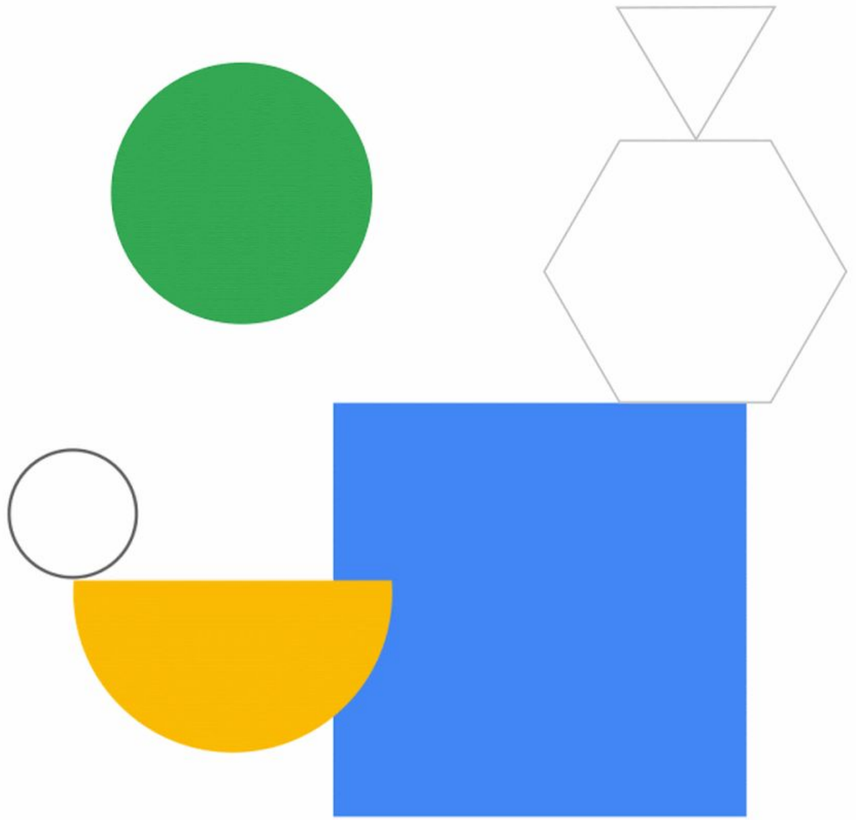
Google Cloud

Introduction to Terraform for Google Cloud



Objectives

Upon completion of this module, you will be able to:

01 Define infrastructure as code.

02 Explain the features and benefits of using Terraform.

03 Explain use cases of Terraform for Google Cloud.

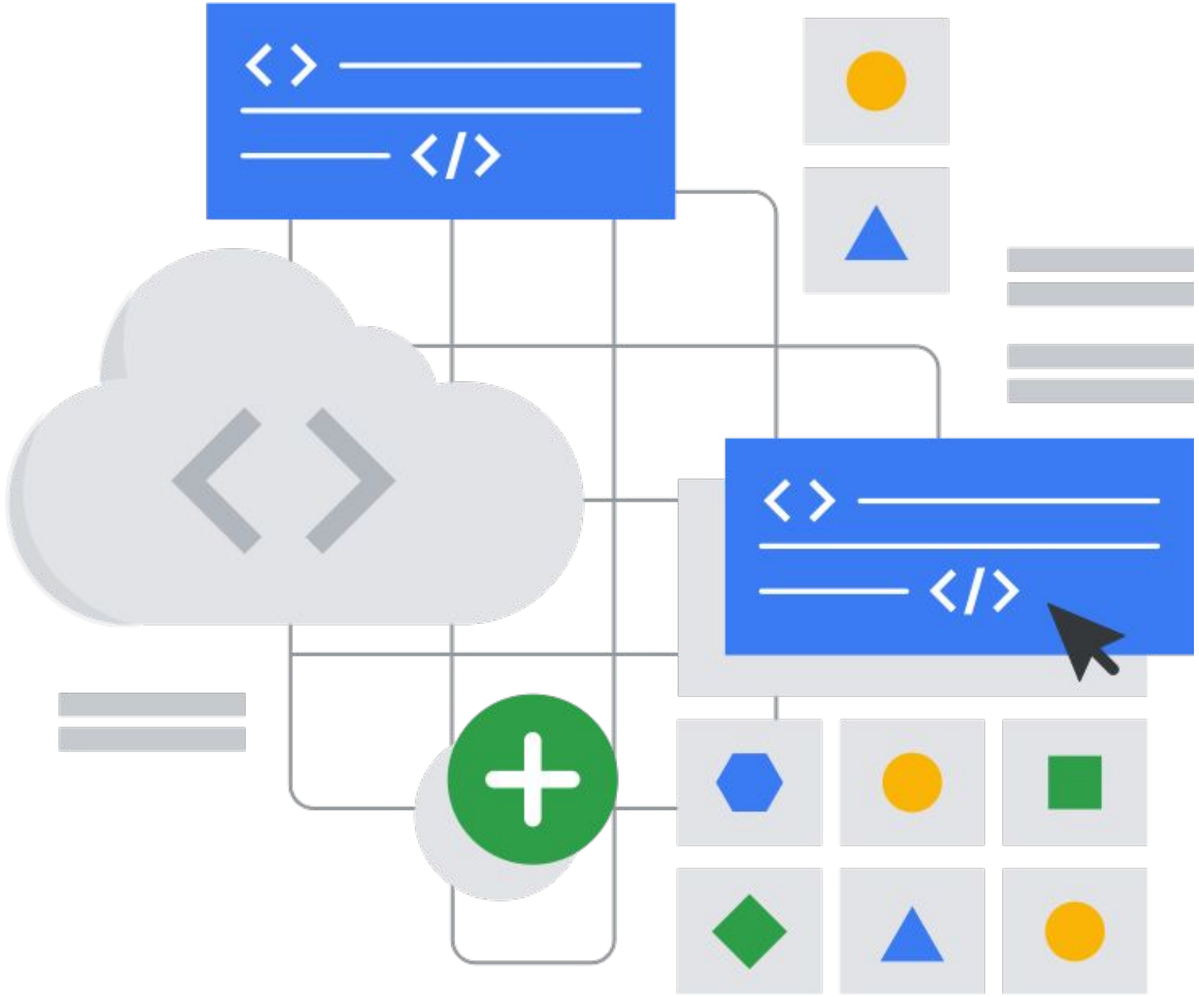
04 Describe how to use Terraform for Google Cloud.

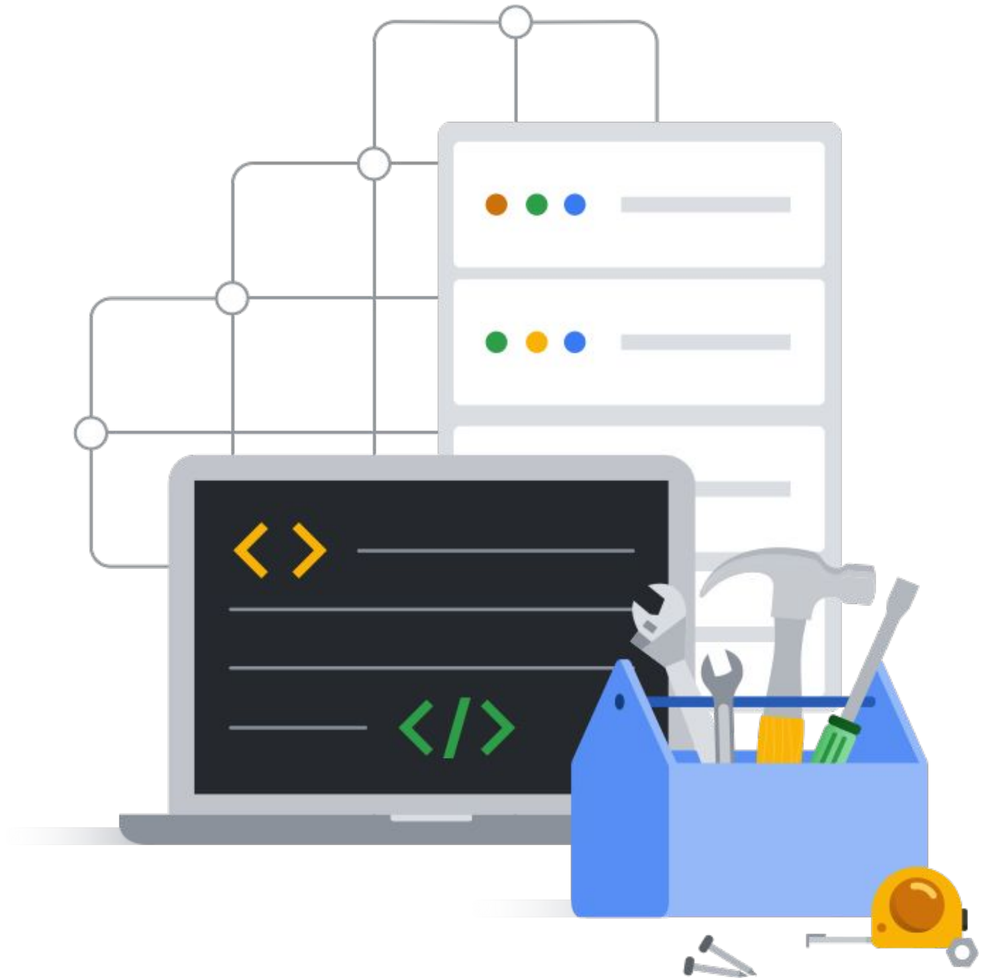
Topics

01	Infrastructure as code
02	Terraform overview
03	Using Terraform

What is infrastructure as code (IaC)?

Instead of clicking around a web UI or using SSH to connect to a server and manually executing commands, **with IaC you can write code in files to define, provision, and manage your infrastructure.**





DevOps

Emphasizes the collaboration and communication of both software developers and IT operations teams

Automates the process of software delivery and infrastructure changes

Problems that IaC can solve

Inability to scale rapidly

Requires rapid scaling of IT infrastructure

Operational bottlenecks

Challenge of managing infrastructure consistently in scale

Disconnected feedback loops

Communication gap between software and IT teams

High manual errors

Increased scale leads to increased human errors

Benefits of IaC

Declarative	Specify the desired state of infrastructure, not updates.
Code management	Commit, version, trace, and collaborate, just like source code.
Auditable	Compare infrastructure between desired state and current state.
Portable	Build reusable modules across an organization.

Provisioning versus configuration

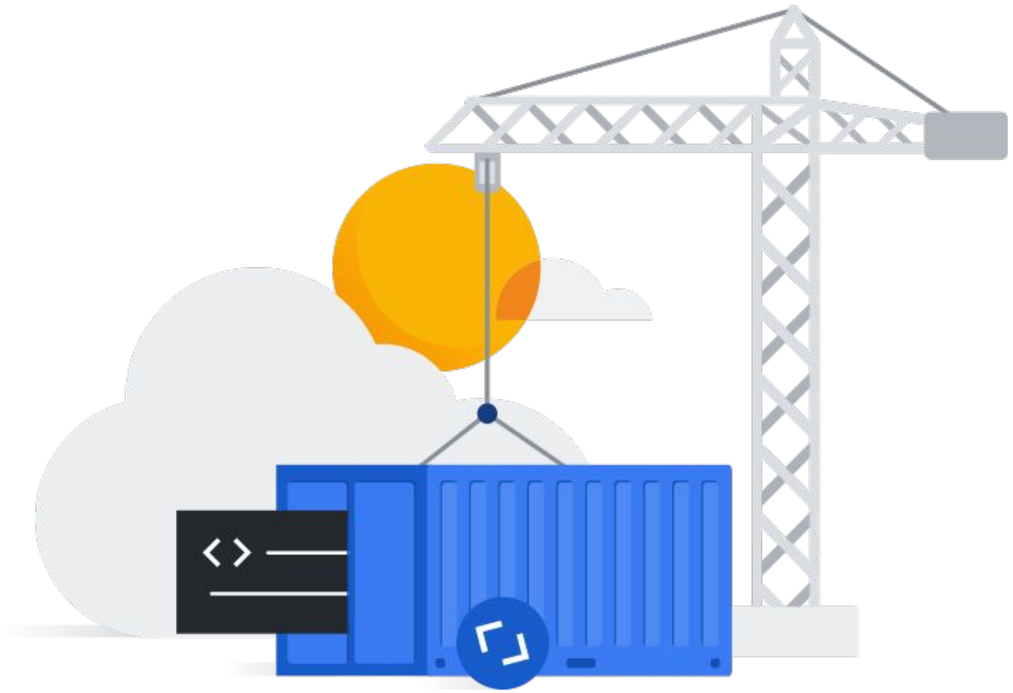
Infrastructure as code

- Used for provisioning and managing cloud resources.
- Example: Creating and provisioning a VM instance.
- Referring to frameworks that manipulate Google Cloud APIs to deploy the infrastructure.

Configuration Management

- Used for virtual machine OS-level configuration.
- Example: Configuring the internals of the VMs.
- Referring to package configurations and software maintenance.

Provisioning versus configuration



Infrastructure as a code

- Launch a GKE cluster

Configuration management

- Deploy containers into the GKE cluster

IaC takes the declarative approach to infrastructure

Imperative (command)

Command line

"Give me five servers"

How to create?

VS.

Declarative (statement)

YAML

"I should have five servers"

What to create?

Topics

01	Infrastructure as code
02	Terraform overview
03	Using Terraform

Terraform is an infrastructure
as code tool created by HashiCorp that
lets you provision Google Cloud resources
with declarative configuration files

Terraform features

Multi-cloud and multi-API

Enterprise support

Large community

Infrastructure provisioning

Terraform for Google Cloud

- Provision resources
- Create resource dependencies
- Standardize configurations
- Validate inputs to resource arguments

IaC configuration workflow

Scope	Author	Initialize	Plan	Apply
Confirm the resources required for a project.	Author the configuration files based on the scope.	Download the provider plugins and initialize directory	View execution plan for resources created, modified, or destroyed.	Create actual infrastructure resources.

Terraform Workflow

Terraform use cases

Manage infrastructure

Terraform takes an immutable approach to building and managing infrastructure.

Track changes

Terraform enables you to review the changes before they are applied to the configuration setup.

Automate changes

Terraform defines the end state of the infrastructure instead of a series of steps to achieve it.

Standardize configurations

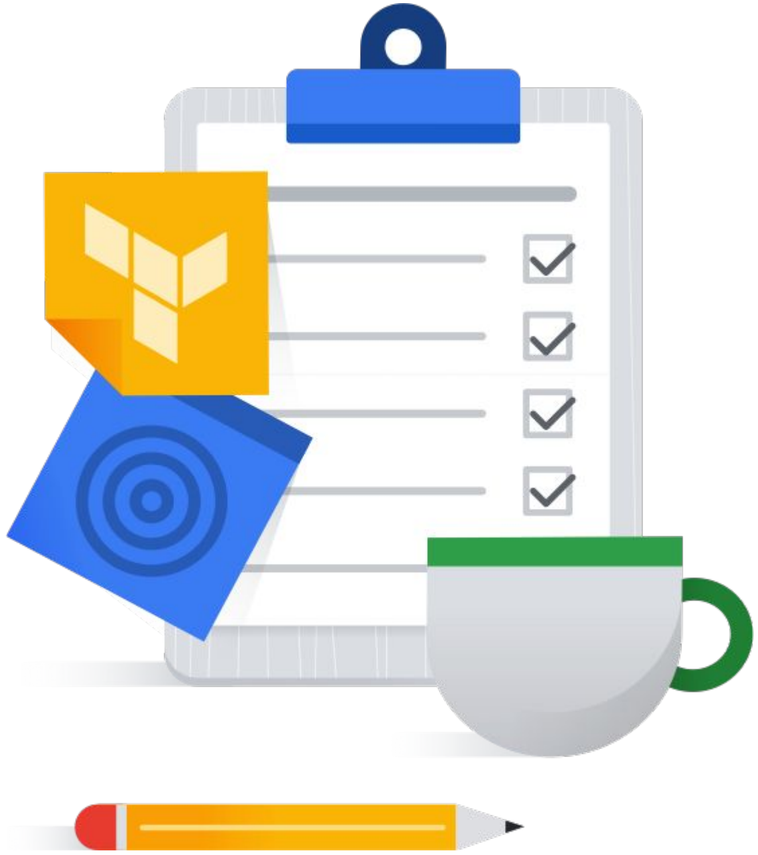
Terraform uses modules to implement best practices and improve efficiency.

Topics

01	Infrastructure as code
02	Terraform overview
03	Using Terraform

Using Terraform

1. Terraform recognizes configuration files written in .tf file.
2. Terraform generates an execution plan.
3. Terraform uses this plan to create infrastructure.
4. Terraform determines the changes and creates incremental execution plans.

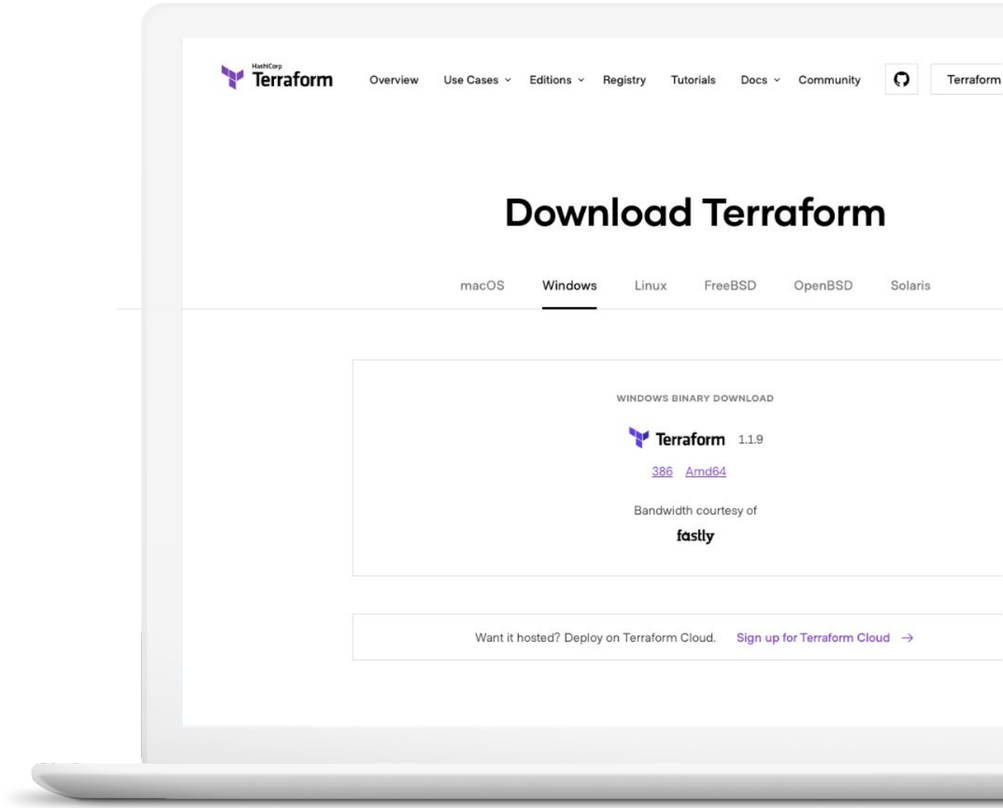


Running Terraform in production

	Managed	Pros	Cons
Terraform Community Edition		• Deployed on a local machine or compute resource in cloud • No license cost • Use public registry within your code	• Does not support concurrent deployments • Only interfaced through CLI
Terraform Cloud		• SaaS based version • Small operational overhead • Comes in three plans • Supports concurrent deployments • Can be accessed through GUI and CLI	• License cost for advanced features
Terraform Enterprise		• Private implementation • Supports concurrent deployments • Secure deployment • Can be accessed through GUI and CLI	• Infrastructure and license costs • Large operational overhead

Installing Terraform on local machine

1. Download the appropriate package based on your system.
2. Unzip the package. Terraform includes a single binary called `terraform`.
3. Edit the PATH variable to include `terraform`.
4. Verify the installation, enter `terraform -help` in a new terminal.



Authentication for Google Cloud

On your workstation

Authenticate Terraform using Google Cloud SDK. On Cloud Shell, it is pre-authenticated for you.

In a VM on Google Cloud

Configure the VM to use a Google Service Account.

Outside Google Cloud

Use workload identity federation, generate a service account key and set env variables.

Example: Creating a VPC network

Create .tf files

Define provider

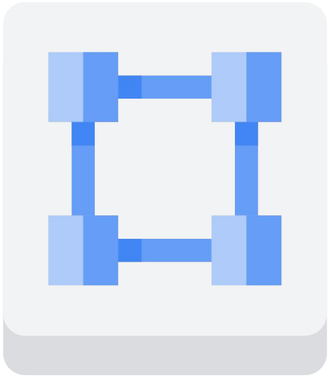
`terraform init`

```
resource "google_compute_network" "mynetwork" {
  name = "vpc-network"
}
```

`terraform plan`

`terraform apply`

VPC



Quiz

Quiz | Question 1

Question

Select the three Terraform editions available in production.

A. Terraform Cloud

B. Terraform Analytics

C. Terraform Community Edition

D Terraform Cyber

E Terraform Enterprise

Quiz | Question 1

Answer

Select the three Terraform editions available in production.

A. Terraform Cloud

B. Terraform Analytics

C. Terraform Community Edition

D Terraform Cyber

E Terraform Enterprise

Quiz | Question 2

Question

Select the two use-cases for Terraform.

A. Automate changes

B. Provision an application

C. Standardize configurations

D Provide financial analytics

E Run OS level customization

Quiz | Question 2

Answer

Select the two use cases for Terraform.

A. Automate changes

B. Provision an application

C. Standardize configurations

D Provide financial analytics

E Run OS level customization

Quiz | Question 3

Question

Which one of the following statements is true regarding Terraform?

A. Terraform can be used for multi-cloud deployments.

B. Terraform can only be used for on-premises deployments.

C. Terraform is used to configure applications on Google Cloud.

D Terraform uses the imperative approach to define infrastructure components.

Quiz | Question 3

Answer

Which one of the following statements is true regarding Terraform?

A. Terraform can be used for multi-cloud deployments.

B. Terraform can only be used for on-premises deployments.

C. Terraform is used to configure applications on Google Cloud.

D Terraform uses the imperative approach to define infrastructure components.

Quiz | Question 4

Question

What is infrastructure as code (IaC)?

A. IaC is a cloud computing model that offers resources on demand to businesses and individuals by using the cloud.

B. IaC is a tool to maintain consistency in an application deployment environment.

C. IaC is a process to define, provision, and manage cloud infrastructure by writing code in files.

D IaC is a data warehouse running on serverless infrastructure.

Quiz | Question 4

Answer

What is infrastructure as code (IaC)?

A. IaC is a cloud computing model that offers resources on demand to businesses and individuals by using the cloud.

B. IaC is a tool to maintain consistency in an application deployment environment.

C. IaC is a process to define, provision, and manage cloud infrastructure by writing code in files.

D IaC is a data warehouse running on serverless infrastructure.

Module review

01 Define infrastructure as code.

02 Explain the features and benefits of using Terraform.

03 Explain use cases of Terraform for Google Cloud.

04 Describe how to use Terraform for Google Cloud.


Google Cloud