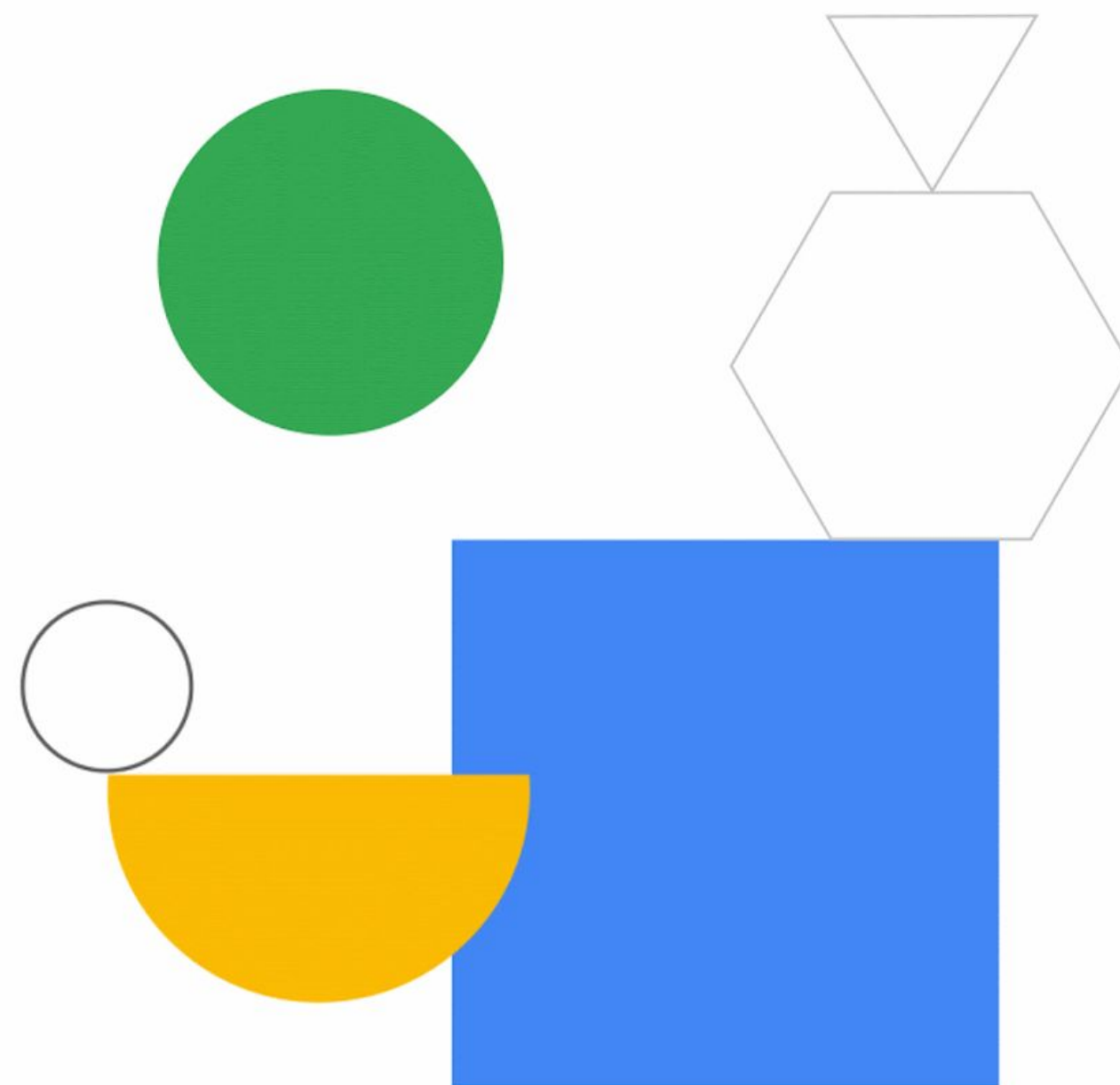


# Introduction to the Terraform State



# Objectives

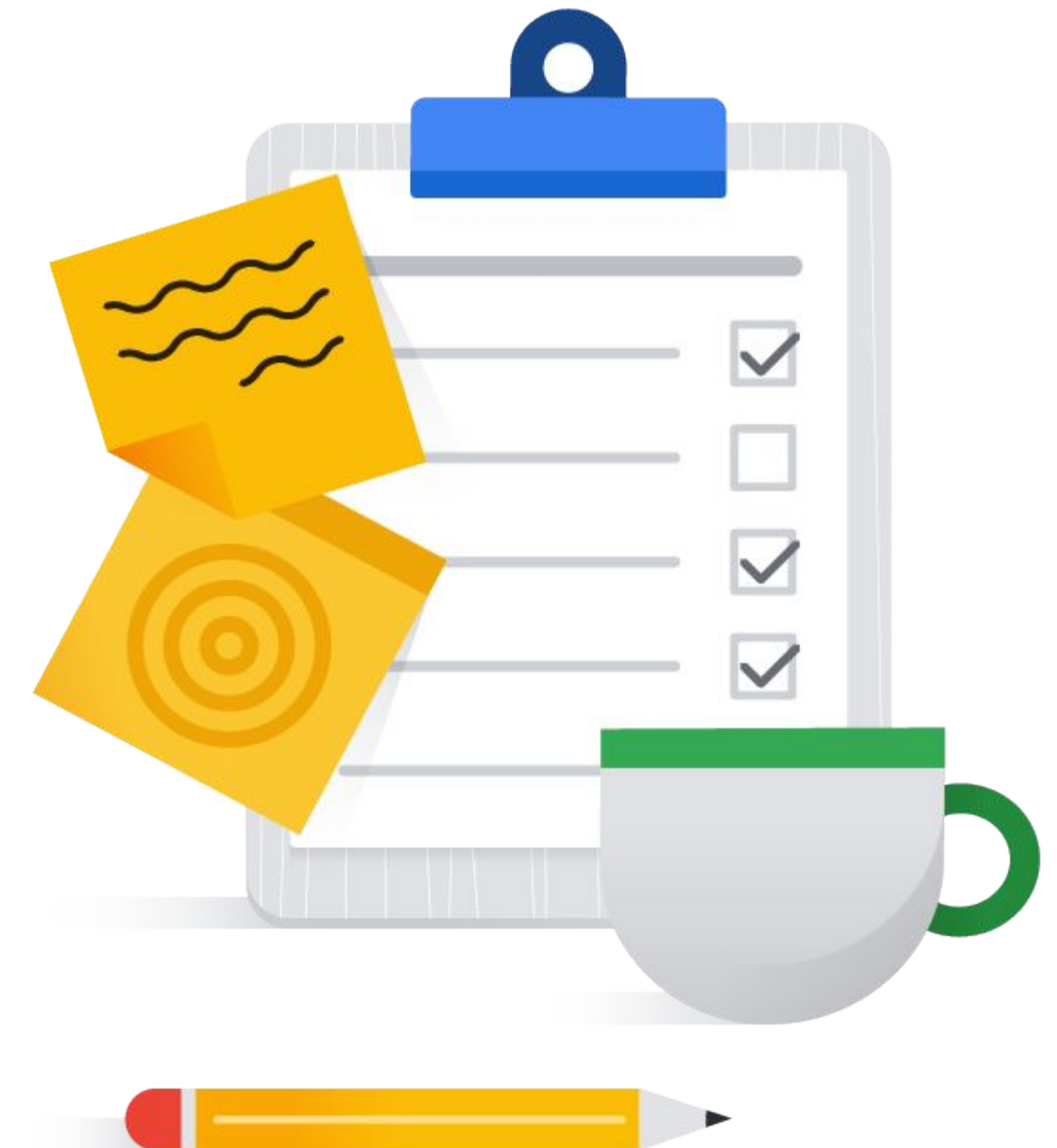
After you complete this module, you will be able to:

- 01 Define Terraform state.
- 02 List the benefits of storing state file remotely.
- 03 Explain how to store the Terraform state in a Google Cloud Storage bucket.
- 04 Explain Terraform state best practices.



# Topics

01	Terraform state overview
02	Different ways to store state files
03	Storing a state file in a remote location
04	Storing a state file in a Cloud Storage bucket
05	Terraform state best practices



# Terraform State

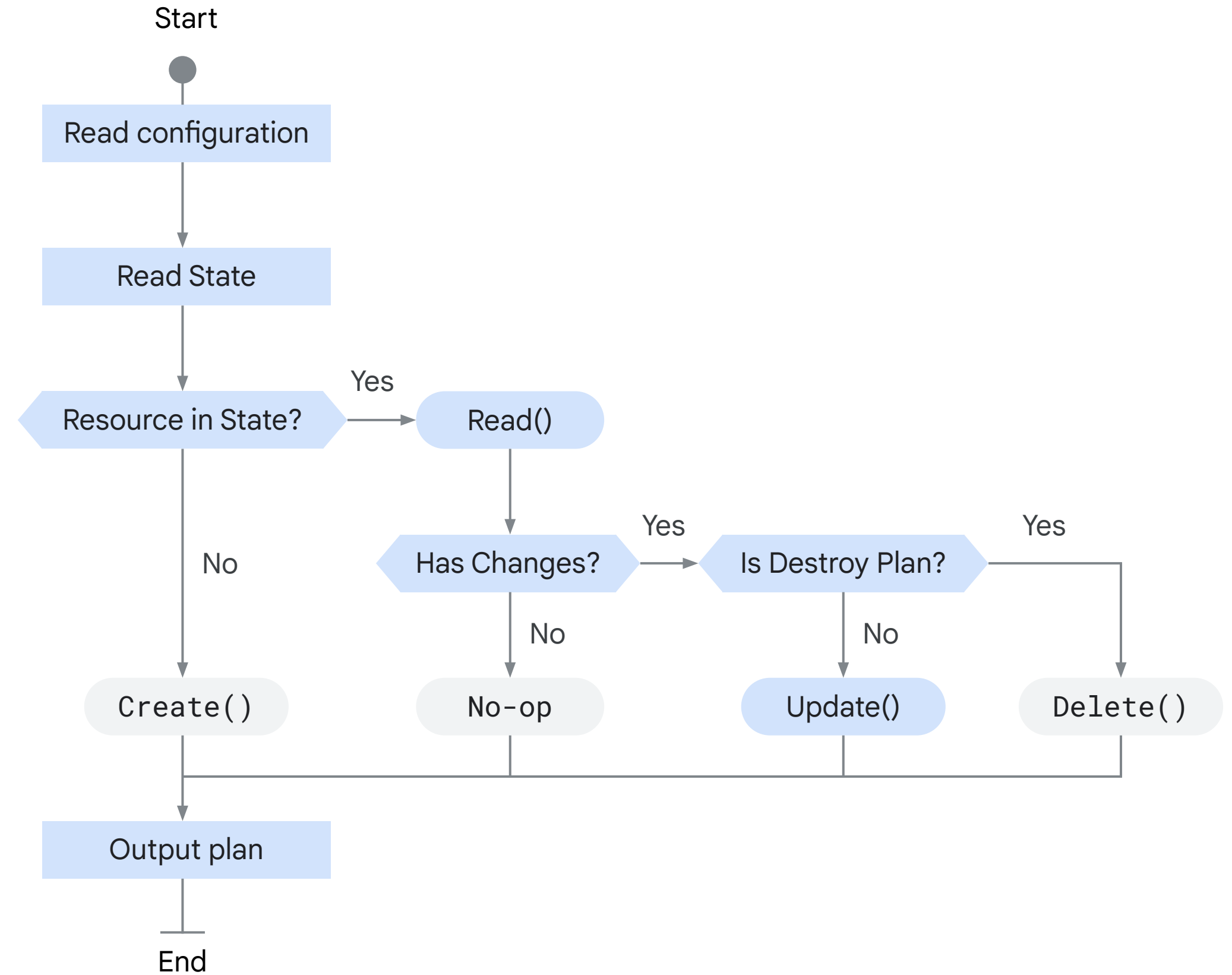
- A state is a metadata repository of your infrastructure configuration.
- A state file is stored by default in a local file named **terraform.tfstate**.
- A Terraform state stores the bindings between objects in a remote system and resource instances.
- The state file records the identity of an instance and updates or deletes in response to configuration changes.

```
-- servers/  
  -- main.tf  
  -- variables.tf  
  -- outputs.tf
```

```
-- main.tf
```

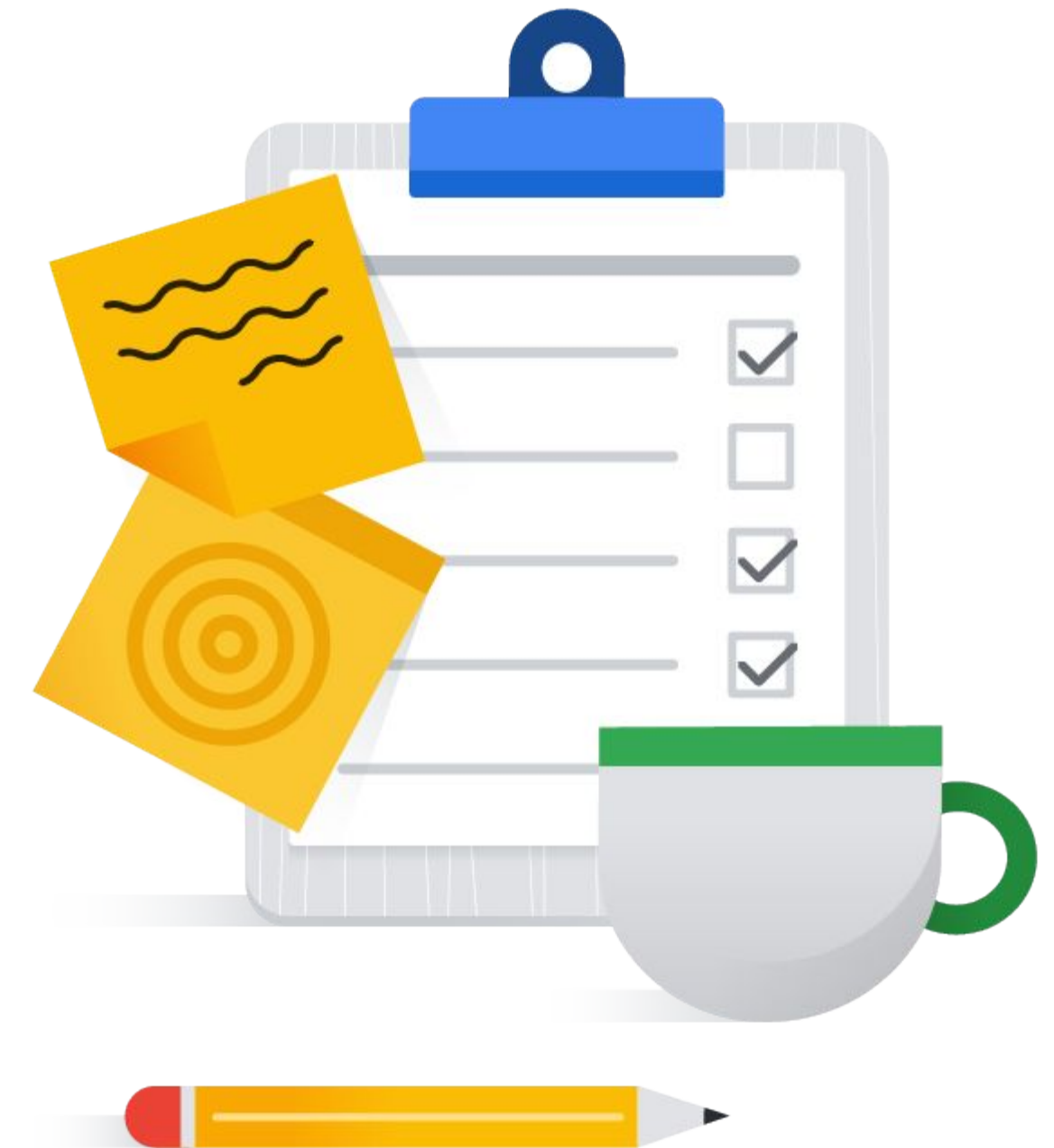
```
terraform.tfstate
```

# How information is stored in a Terraform state file



# Topics

01	Terraform state overview
02	<b>Different ways to store state files</b>
03	Storing a state file in a remote location
04	Storing a state file in a Cloud Storage bucket
05	Terraform state best practices



# Ways to save a state file

## Save locally

```
-- servers/  
  -- main.tf  
  -- variables.tf  
  -- outputs.tf  
  
-- main.tf
```

**terraform.tfstate**



The **terraform apply** command automatically generates a state file that is saved in a working directory.

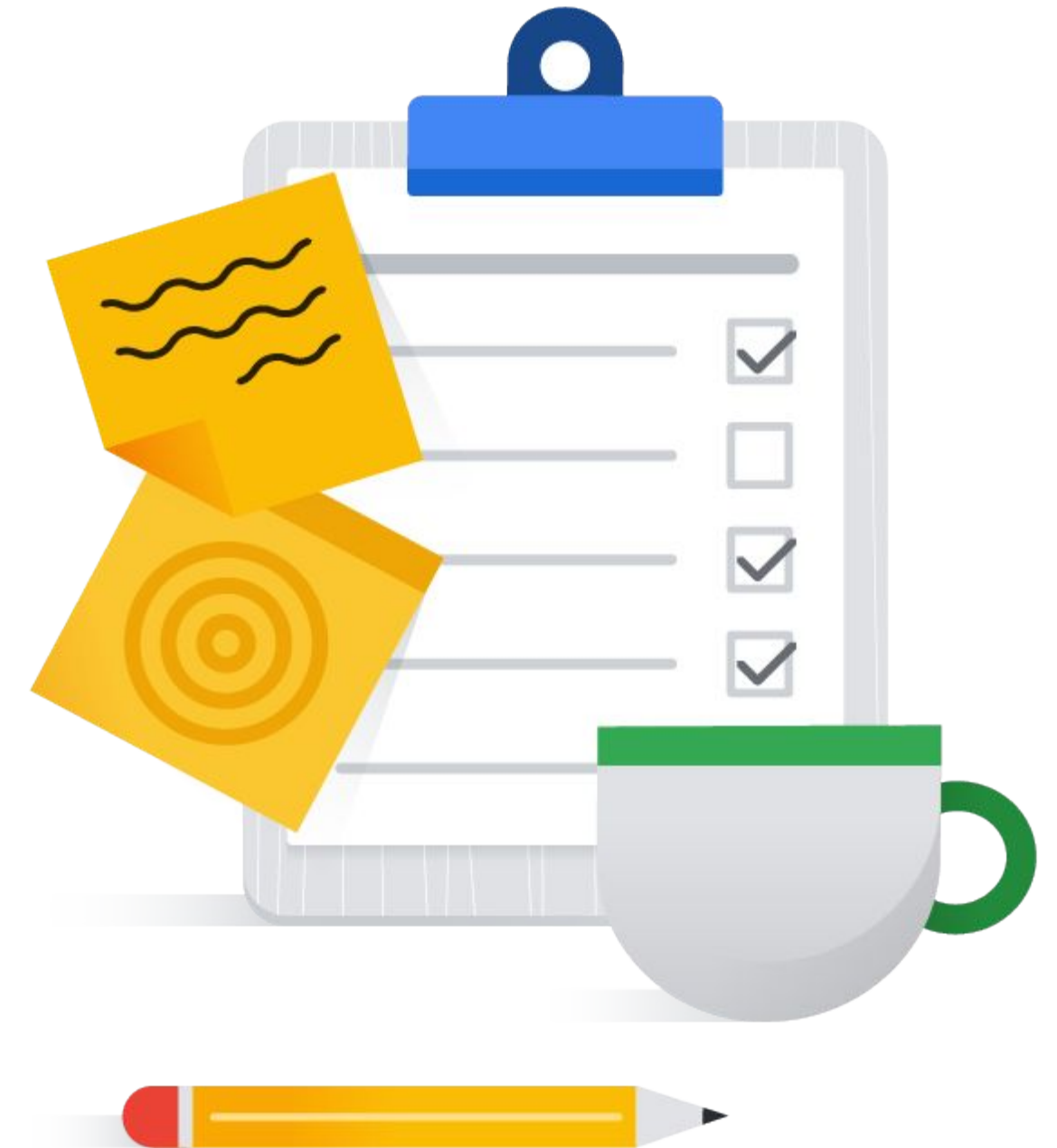
## Save remotely



The file is stored in a remote location like a Google Storage bucket or Terraform Cloud.

# Topics

01	Terraform state overview
02	Different ways to store state files
03	Storing a state file in a remote location
04	Storing a state file in a Cloud Storage bucket
05	Terraform state best practices





# Issues with storing the Terraform state locally

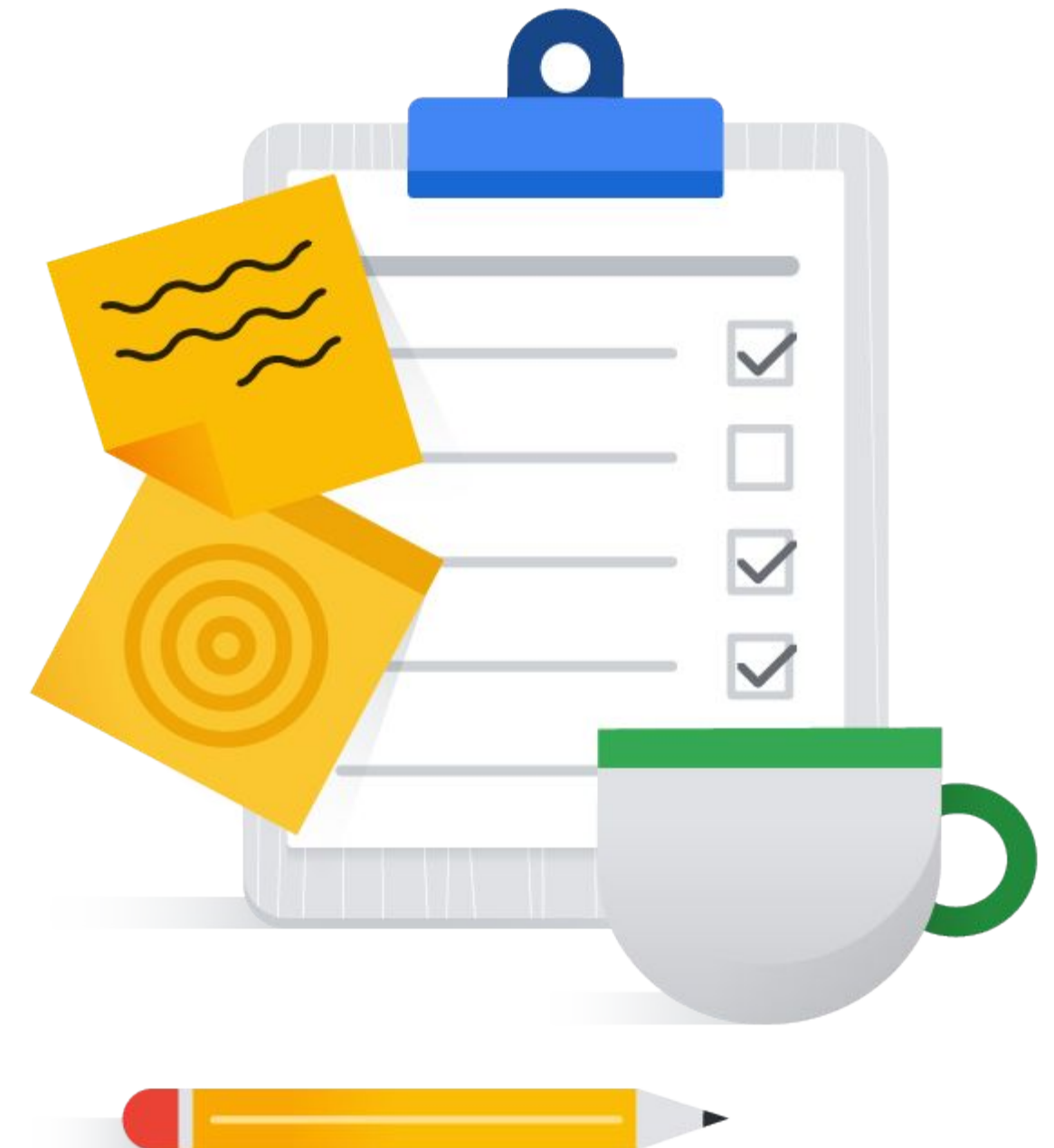
No shared access	For any update to the infrastructure, each member of your team needs access to the same state file.
No locking	When team members run Terraform at the same time, they run into conflict in access, which leads to data corruption and data loss.
No confidentiality	State file exposes all sensitive data such as username and password of the database.

# Benefits of storing a state file in a remote location

Automatic updates	Remote state supports automatic updates to the state file.
Locking	Cloud Storage buckets natively support state locking.
Secure access	Google Cloud Storage buckets support encryption and IAM policies.

# Topics

01	Terraform state overview
02	Different ways to store state files
03	Storing a state file in a remote location
04	<a href="#">Storing a state file in a Cloud Storage bucket</a>
05	Terraform state best practices



# Store Terraform state remotely in a Cloud Storage bucket

-- main.tf

M

-- backend.tf

Create the bucket.

Change the backend  
configuration

M

```
resource "google_storage_bucket" "default" {  
  name           = "bucket-tfstate"  
  force_destroy  = false  
  location       = "US"  
  storage_class  = "STANDARD"  
  versioning {  
    enabled = true  
  }  
}
```

# Store Terraform state remotely in a Cloud Storage bucket

-- main.tf

M

-- backend.tf

B

Create the bucket.

Change the backend  
configuration

M

```
resource "google_storage_bucket" "default" {  
  name          = "<my_unique_bucket_name>"  
  force_destroy = false  
  location      = "US"  
  storage_class = "STANDARD"  
  versioning {  
    enabled = true  
  }  
}
```

B

```
terraform {  
  backend "gcs" {  
    bucket = "<my_unique_bucket_name>"  
    prefix = "terraform/state"  
  }  
}
```

# Example of a state file

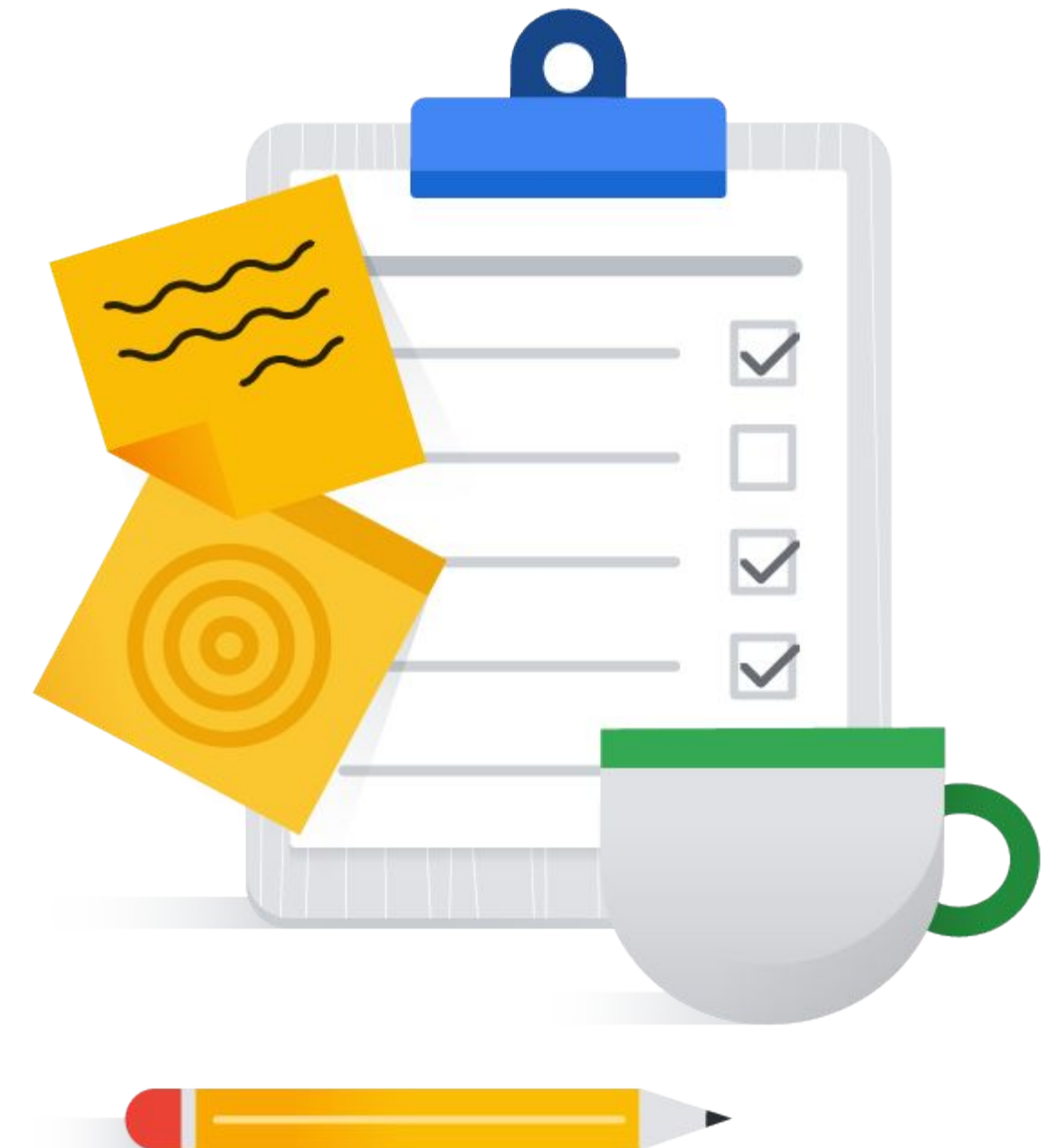


```
{
  "version": 4,
  "terraform_version": "1.2.3",
  "serial": 2,
  "lineage": "f9e0d82a-4bb4-1db2-7912-3d31117903cc",
  "outputs": {},
  "resources": [
    {
      "mode": "managed",
      "type": "google_storage_bucket",
      "name": "test-bucket-for-state",
      "provider": "provider[\"registry.terraform.io/hashicorp/google\"]",
      "instances": [
        {
          "schema_version": 0,
          ...
        }
      ]
    }
  ]
}
```

Snippet of the state file from the Cloud Storage bucket

# Topics

01	Terraform state overview
02	Different ways to store state files
03	Storing a state file in a remote location
04	Storing a state file in a Cloud Storage bucket
05	<b>Terraform state best practices</b>



# Terraform state best practices

Use remote state when working in teams

Remote state supports locking and versioning.

Encrypt state

Use customer-supplied encryption keys to add a layer of protection.

Don't store secrets in a state file

Avoid storing secrets in state because Terraform stores secret values in plaintext.

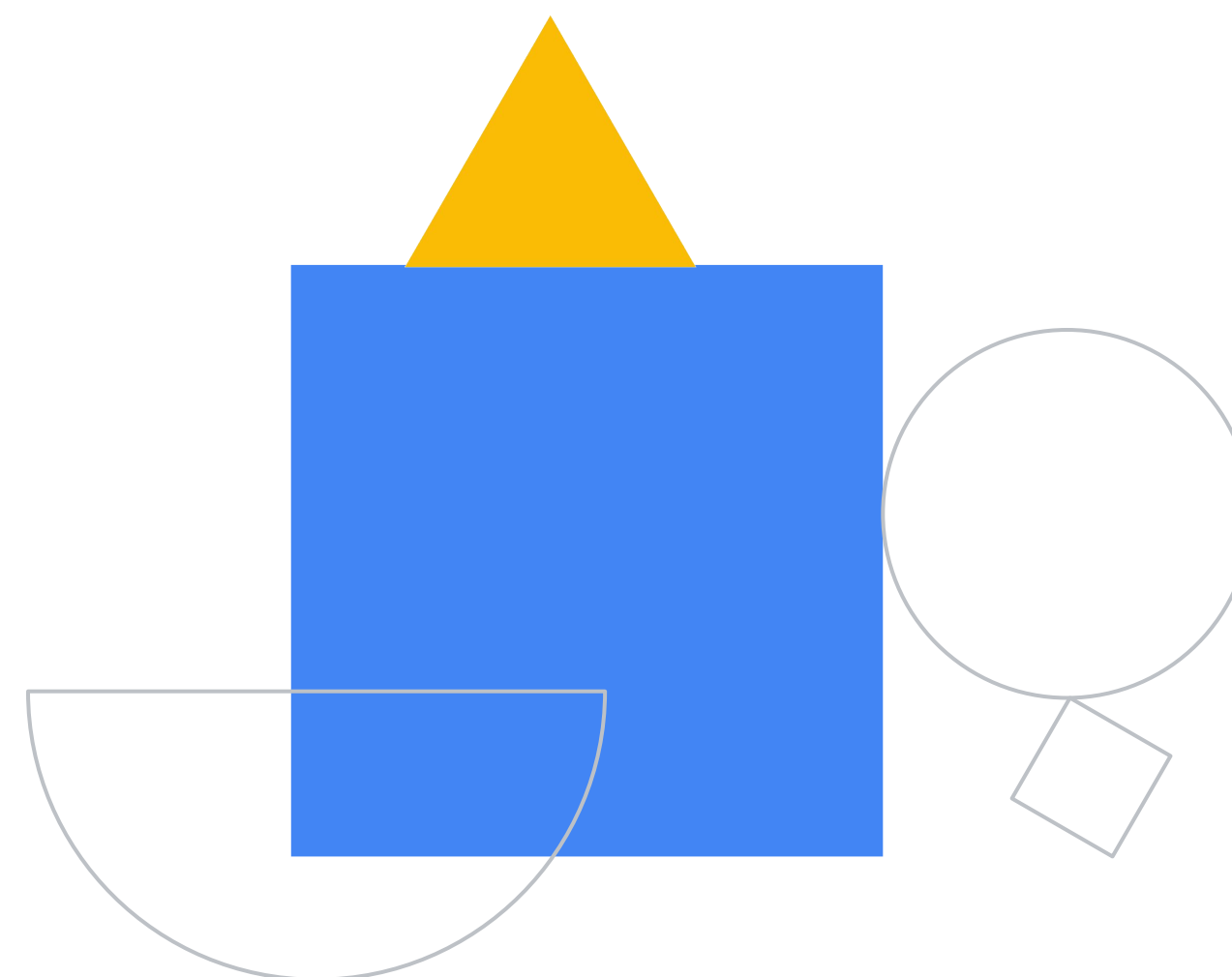
Don't modify state manually

Use the terraform state command when when you need to modify a state.



# Lab

Creating a remote backend



# Quiz



# Quiz | Question 1

## Question

Select the three benefits of storing a Terraform state file remotely.

- A. Usage Analytics
- B. Locking
- C. Secure access
- D. Sharing and delegation
- E. Automatic insights

# Quiz | Question 1

## Answer

Select the three benefits of storing a Terraform state file remotely.

- A. Usage Analytics
- B. Locking
- C. Secure access
- D. Sharing and delegation
- E. Automatic insights



# Quiz | Question 2

## Question

A state file is stored by default in a local file named `terraform.tfstate`.

- A. True
- B. False

# Quiz | Question 2

## Answer

A state file is stored by default in a local file named `terraform.tfstate`.

- A. True
- B. False



# Module Review

- 01 Define Terraform State.
- 02 List the benefits of storing a state file remotely.
- 03 Explain how to store the state in a Google Cloud Storage Bucket.
- 04 Explain Terraform state best practices.

