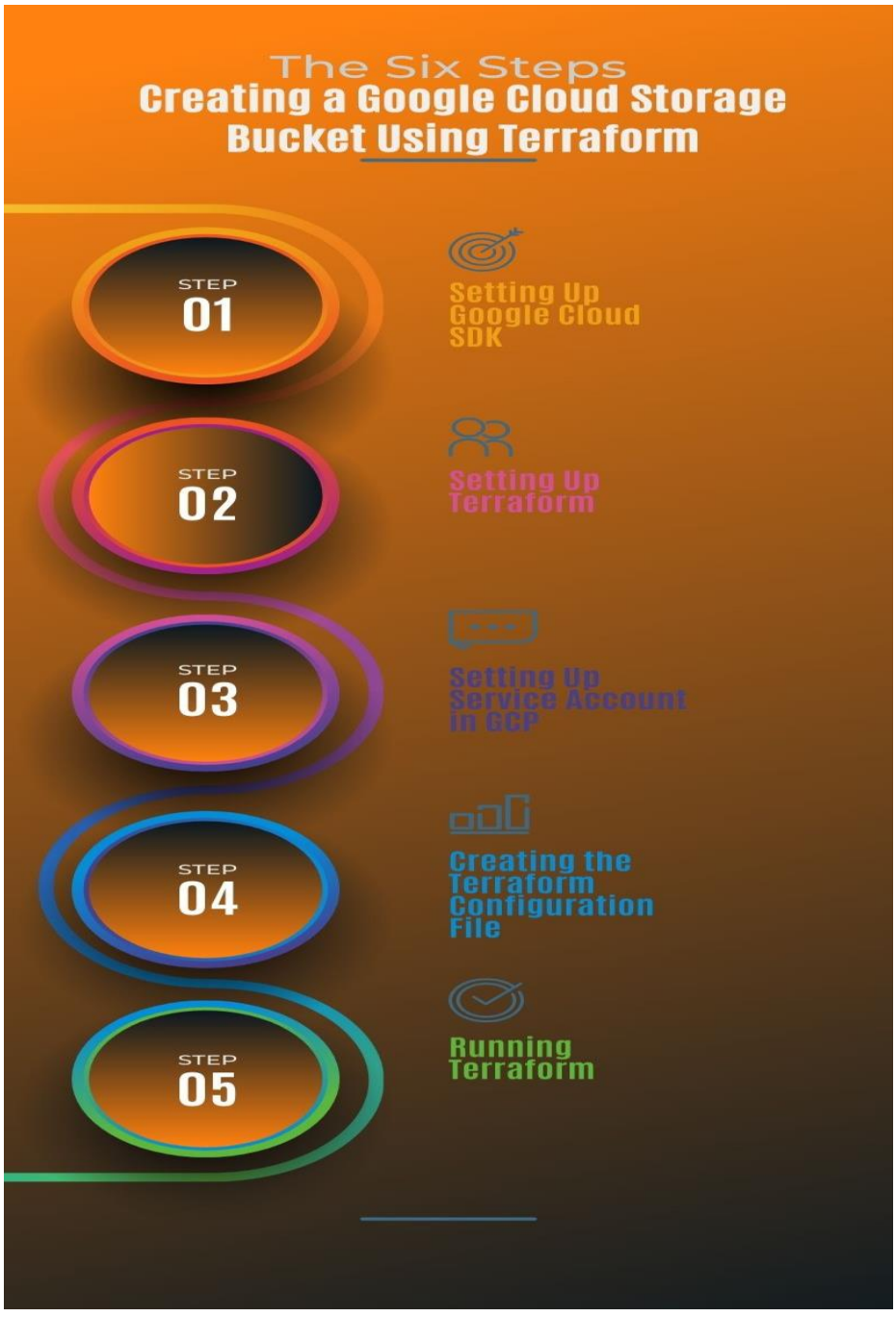
Guide to Setting Up Google Cloud Storage with Terraform

orange™

Creating a Google Cloud Storage Bucket Using Terraform:

This guide will walk you through the steps to create a Google Cloud Storage Bucket using Terraform.



Prerequisites:

- Google Cloud SDK installed
- Terraform installed

- Google Cloud Platform (GCP) account

Step-by-Step Guide:

1. Setting Up Google Cloud SDK

Open your terminal in your Linux virtual box and run gcloud init to initialize the Google Cloud SDK. Follow the prompt to log into your Google account and select your project.

2. Setting Up Terraform

If Terraform is not yet installed, use these commands to download and install it:

Copy and paste the following on linux terminal:

```
wget https://releases.hashicorp.com/terraform/1.0.6/terraform_1.0.6_linux_amd64.zip
unzip terraform_1.0.6_linux_amd64.zip
mv terraform /usr/local/bin/
```

After installation, you can verify it with terraform version.

3. Setting Up Service Account in GCP

- In the GCP console, navigate to IAM & Admin > Service Accounts.
- Click Create Service Account, provide a name, and click Create.

- Grant the service account Storage Admin role and click Continue.
- Click Done to finish creating the service account.
- Click on the newly created service account. Under the Keys tab, click Add Key, then Create new key.
- Select JSON as the key type and click Create. A JSON key file will be downloaded; keep this file secure.

4. Creating the Terraform Configuration File

- Create a new directory and navigate into it, for example:

Copy and paste the following on linux terminal:

```
mkdir terraform-gcp
cd terraform-gcp
```

- Create a new file named main.tf and open it in a text editor.
- Paste the following into main.tf. Replace path_to_your_service_account_key.json with the path to your downloaded JSON key file, and your_bucket_name with your desired bucket name.

```
// The provider block configures the named provider, in this case "google"
provider "google" {
  // Path to the service account key file
  // Uncomment the line below and replace <PATH_TO_YOUR_SERVICE_ACCOUNT_KEY> w
ith your own key path
  // credentials = file("<PATH_TO_YOUR_SERVICE_ACCOUNT_KEY>")

  // ID of the project that will be used for managing resources
  project = "extended-altar-394912"

  // Default region for managing resources
  // This is updated to a region in France
  region  = "europe-west 2"
}

```

```
// Declare a variable named "bucket_name"
variable "bucket_name" {
  // Description of the variable
  description = "The name of the bucket"

  // Type of the variable
  type        = string

  // Default value of the variable
  // The bucket name has been updated
  default     = "multi-cloud-team-std-bucket"
}

// Declare a variable named "location"
variable "location" {
  // Description of the variable
  description = "The location of the bucket"

  // Type of the variable
  type        = string

  // Default value of the variable
  // The location has been updated to Europe
  default     = "EU"
}

// Declare a variable named "storage_class"
variable "storage_class" {
  // Description of the variable
  description = "The storage class of the bucket"

  // Type of the variable
  type        = string

  // Default value of the variable
  default     = "STANDARD"
}

// Declare a resource of type "google_storage_bucket"
resource "google_storage_bucket" "bucket" {
  // Name of the bucket
  name = var.bucket_name

  // Location of the bucket
  location = var.location

  // Storage class of the bucket
  storage_class = var.storage_class}
```

- Save and close the file.

5. Running Terraform

- In the terminal, run terraform init to initialize your Terraform workspace.
- Run terraform plan to see what Terraform will do before actually doing it.
- If everything looks good, run terraform apply to create the storage bucket. Confirm the action by typing yes when prompted.

6. Verifying the Bucket Creation

- In the GCP console, navigate to Storage > Browser.
- Your newly created bucket should appear in the list.

Congratulations! You have successfully created a Google Cloud Storage Bucket using Terraform.