

Terraform

Infrastructure as code

Infrastructure

All cloud resources that need to be present and configured when you want to create your entire application from scratch

Cloud resources

Resources you can create through the **Google Cloud API** (using **gcloud** or the **web console**) are cloud resources.

Infrastructure as code

You use source code to describe your entire infrastructure and use a management tool to create, update, or delete those resources.

Source code

can be a simple configuration language, a fully fledged programming language, or something in between.

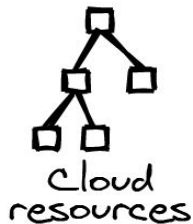
Why IaC?

- Easy to see the configuration of the system
- Easy to maintain
- Can be added to version control and collaborate with your team to build infrastructure. Every infrastructural change can be reviewed using a pull request flow.
- Reproducibility is a big benefit as well. If you create a new Google Cloud project, you can have your IaC tool set everything up. Similarly, it is also easy to tear down everything you've created, which helps to avoid unexpected costs.
- Focus is on defining the desired state of your cloud resources instead and not the commands that will create that resources

Manual



Apply one-off commands
to manage resources



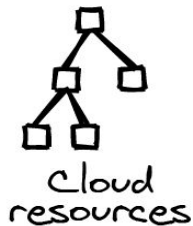
Automation



Create and
run a script



Contains logic
and multiple commands



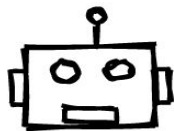
Infrastructure as Code



Edit

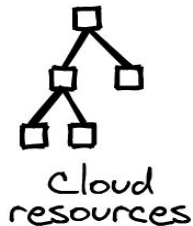


Read by



Management Tool

Plans and
applies
changes



Terraform

- an open source infrastructure as code tool by HashiCorp company
- uses a **high-level, declarative, configuration** language to describe cloud resources and the relations between them
- stable and mature tool with great support for Google Cloud and other cloud providers

gcloud credentials

There are two separate credentials managed by the gcloud CLI:

- When you execute gcloud commands, gcloud uses the credentials that are saved when you ran **gcloud auth login** .
- When you talk to Google Cloud from a client library in your software or an application like Terraform, it uses the application default credentials. Those are the credentials that were saved when you ran **gcloud auth application-default login**.

main.tf

```
provider google { ❶  
  project = var.project_id  
  version = "~> 3.0"  
}
```

```
variable "project_id" { ❷  
  type          = string  
  description = "The Google Cloud Project ID to use"  
}
```


Terraform configuration language

- HashiCorp **Configuration Language (HCL)** (Not a programming language)
- Has a limited set of “**programming language**” features: you can create **loops**, pass **parameters**, and **create modules to capture patterns you want to reuse**
- The **resource types** are supplied by **the provider**—in this case, **the Google provider** and are unique to each provider; changing from Google Cloud to AWS would require a full rewrite.
- offers **unified workflow** that lets you combine **infrastructure configuration for multiple providers** in one codebase.

Terraform commands

```
terraform init -backend-config="bucket=tf-state-demo-project"  
-backend-config="impersonate_service_account=terraform-iac-pipeline@ign  
eous-stone-322108.iam.gserviceaccount.com"
```

```
terraform plan -var-file ./vars/dev.tfvars
```

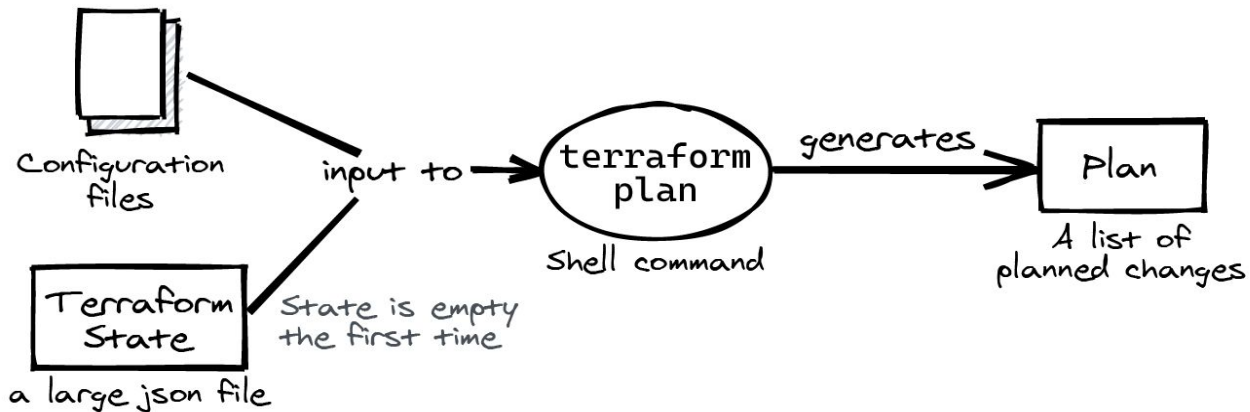
```
terraform apply -var-file ./vars/dev.tfvars
```

- *'The state of your infrastructure has been saved to the path **terraform.tfstate**'*

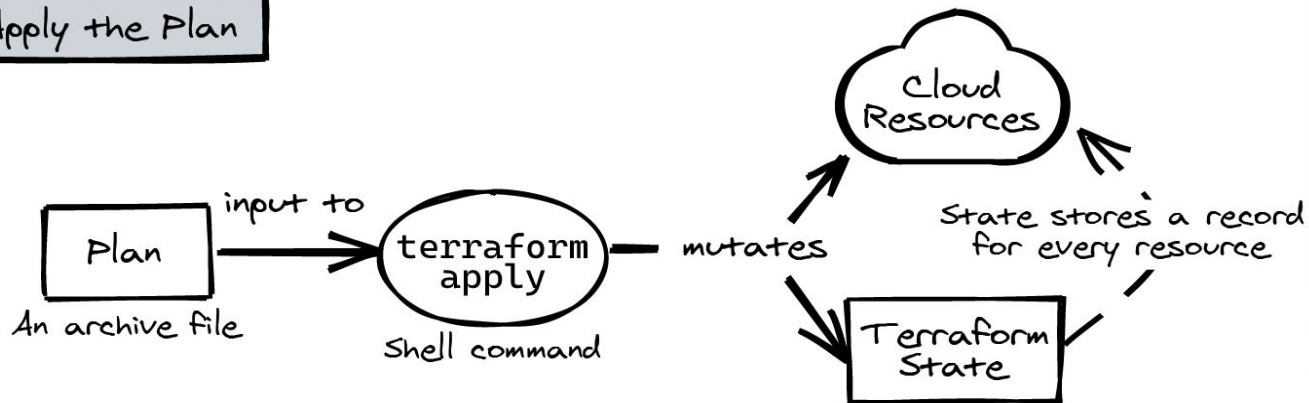
```
terraform show
```

```
terraform destroy -target X -var-file ./vars/dev.tfvars
```

Create a Plan



Apply the Plan



Terraform references (dependencies)

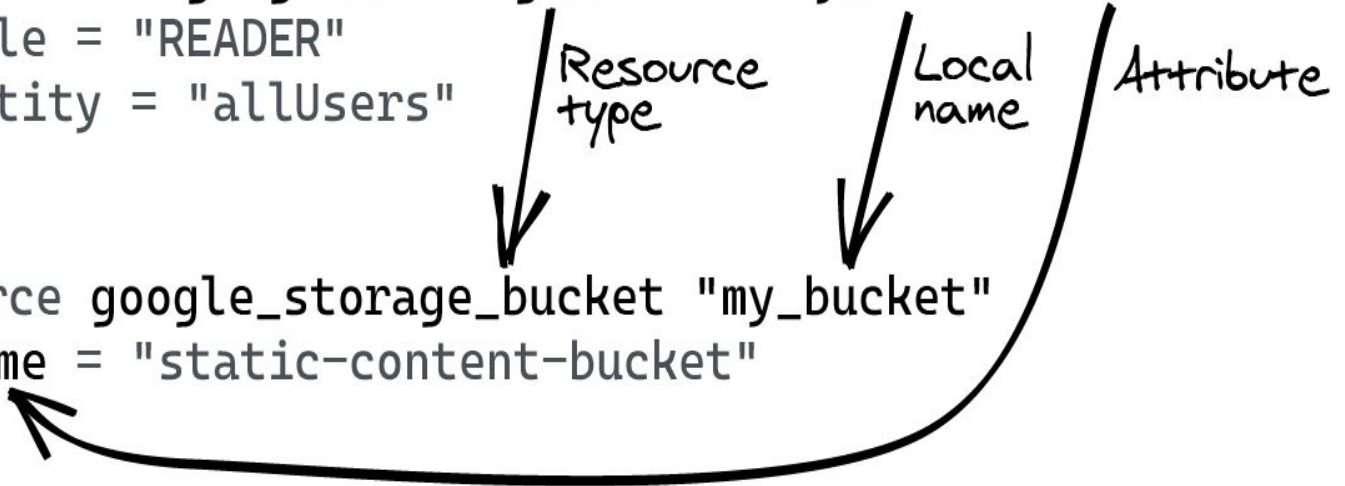
```
resource google_storage_bucket_access_control "public_rule" {  
  bucket = google_storage_bucket.my_bucket.name  
  role = "READER"  
  entity = "allUsers"  
}
```

```
resource google_storage_bucket "my_bucket"  
  name = "static-content-bucket"  
}
```

Resource
type

Local
name

Attribute



Google Cloud Platform provider

<https://registry.terraform.io/providers/hashicorp/google/latest/docs>

<https://github.com/GoogleCloudPlatform/cloud-foundation-fabric>

Ssh access

<https://kloudle.com/academy/5-ways-to-connect-to-your-gcp-vm-instances-using-ssh/>