

Some of the most useful Terraform commands are:

* terraform init - initializes the current directory
* terraform refresh - refreshes the state file
* terraform output - views Terraform outputs
* terraform apply - applies the Terraform code and builds stuff
* terraform destroy - destroys what has been built by Terraform
* terraform graph - creates a DOT-formatted graph
* terraform plan - a dry run to see what Terraform will do

**Terraform gcp**

**https://github.com/devbyaccident/implementing-terraform-with-gcp/tree/main/m2**

Commands used:

1. Use Application Default Credentials (ADC): gcloud auth application-default login
2. Create a new GCP Project: gcloud projects create <PROJECT\_ID>
3. List all GCP Projects: gcloud projects list
4. Set active Project: gcloud config set project <PROJECT\_ID>
5. Use Service Account Credentials:

gcloud iam service-accounts create prod-svc

gcloud projects add-iam-policy-binding carved-rock-prod --member="serviceAccount:prod-svc@carved-rock-prod.iam.gserviceaccount.com" --role="roles/owner"

gcloud iam service-accounts keys create prod-svc-creds.json --iam-account=prod-svc@carved-rock-prod.iam.gserviceaccount.com

1. Set GCP Credentials: export GOOGLE\_APPLICATION\_CREDENTIALS=<Path to service account JSON key>
2. Set ssh username (Optional): export TF\_VAR\_username=$(whoami)

https://partner.cloudskillsboost.google/course\_sessions/2251754/labs/357080 remote backend

provider "google" {

project = "qwiklabs-gcp-02-9b22f1ae18a8"

region = "us-central-1"

}

resource "google\_storage\_bucket" "test-bucket-for-state" {

name = "qwiklabs-gcp-02-9b22f1ae18a8"

location = "US"

uniform\_bucket\_level\_access = true

}

terraform {

backend "local" {

path = "terraform/state/terraform.tfstate"

}

}

**Change terraforms backend.**

terraform {

backend "gcs" {

bucket = "qwiklabs-gcp-02-9b22f1ae18a8"

prefix = "terraform/state"

}

}

terraform init -migrate-state

terraform refresh