**TERRAFORM**

Terraform is an open-source infrastructure as a code (IAC) software tool created by HashiCorp. It enables users to define and provision a datacenter infrastructure using a high-level configuration language known as Hashicorp Configuration Language (HCL).

List of Infrastructure as a code software (IAC) tool

* Terraform Supports multiple cloud provider
* CloudFormation Supports only AWS (Owned by AWS)
* Ansible
* SaltStack
* Chef
* Puppet

Tools like Terraform and CloudFormation primarily used for Infrastructure Orchestration and they can do configuration management only for certain part.

Whereas, Ansible/Chef/Puppet are only designed for configuration management , and they can also orchestrate infra for certain part only.

Terraform Installation in Linux

wget <https://releases.hashicorp.com/terraform/0.12.24/terraform_0.12.24_linux_amd64.zip>

unzip terraform\_0.12.24\_linux\_amd64.zip

mv terraform /bin/

terraform version

Some useful commands

**terraform init**

**terraform validate** (To Check Configuration)

**terraform plan** (Desired State == Current State)

**terraform apply**

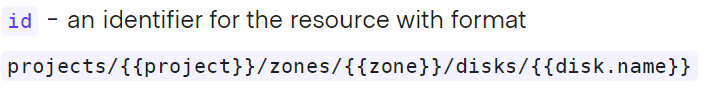
**terraform destroy** (To delete all terraform resources)

**terraform destroy -target <>** (To destroy specific resource)

**terraform refresh** (Update state file against real resources. This will not modify your infrastructure, but it can modify your state)

**terraform show** command show terraform state file configuration in easier way instead of opening terraform.tfstate file again and again.

**Attributes**



URI vs URL   
URI 🡪 Identifies complete address of resource

URL 🡪 Identifies only location of resource

So id here only refers to URI.

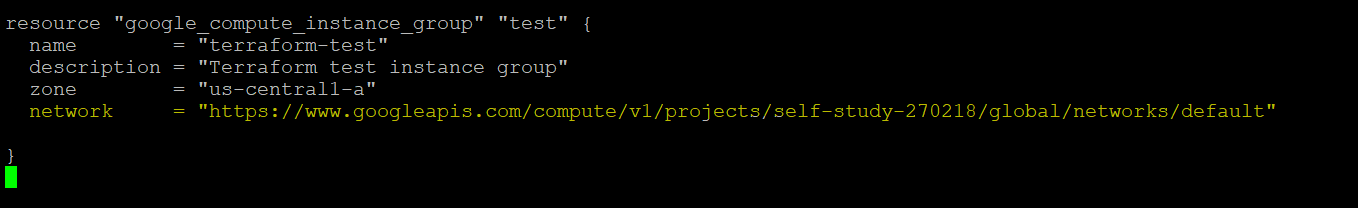
Example 🡪

id=projects/analysis-project-273210/zones/europe-west2-b/instances/terraform-machine2

id=projects/analysis-project-273210/zones/europe-west2-b/disks/disk1

**Note: To use an existing resource from GCP in terraform script, user needs to use complete URI of the resource.**

For e.g. : Below is the code snippet for creating an instance group in an existing GCP network named default.

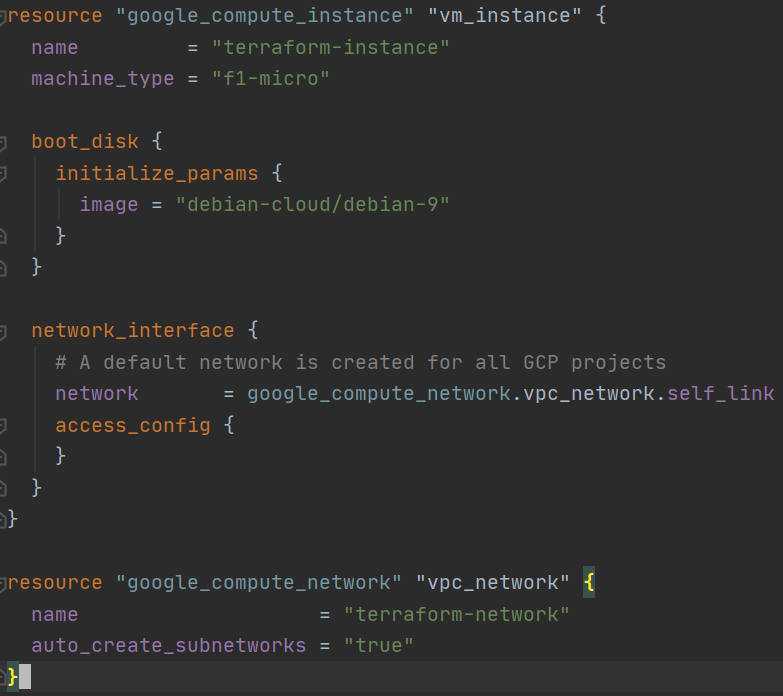


List of existing networks in GCP can be found by:

gcloud compute networks list --uri

**Linking GCP resources**

By “self\_link”

Example 

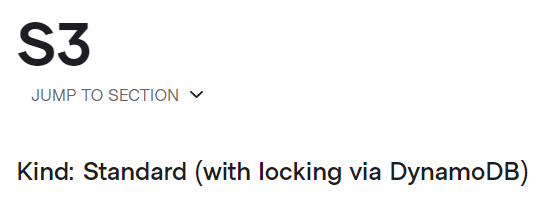
**Advanced Terraform Concepts**

**terraform.tfstate** have all clear sensitive (password, secrets etc) and insensitive data. So , never commit tfstate file in any repo. We can use **remote backend** to store terraform.tfstate file remotely not locally.

We have multiple types of remote backend – artifactory, consul, etcd, gcs , s3, swift etc.

**But make sure to use that remote backend which support state file locking feature.**

State file locking feature allow to run “terraform plan” sequentially not parallelly. Else, it can corrupt **terraform.tfstate.**

S3 support locking only with DynamoDB (<https://www.terraform.io/docs/backends/types/s3.html>)  


GCS support locking (<https://www.terraform.io/docs/backends/types/gcs.html>)

