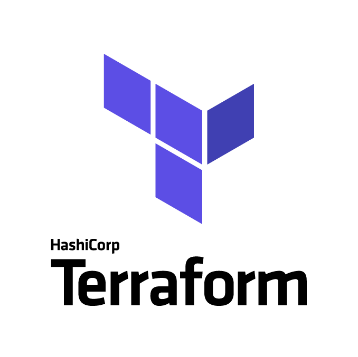
**TERRAFORM**

**Infrastructure As a Code (IaaS)**

**Introduction**

* IAAC | Automate Infrastructure
* Define Infrastructure State
* Ansible, Puppet or Chef automates mostly OS related tasks
  + Define machine state
* Terraform automates infrastructure itself
  + AWS, Azure, GCP, Digital Ocean etc.
* Terraform works with automation tools like ansible after infra is setup and ready
* Terraform has its own language called HCL (Hashicorp Language), its similar to JSON Syntax

**Installation**

<https://developer.hashicorp.com/terraform/downloads>

**Exercise 1**

1. Write instance.tf file
2. Apply the terraform file – instance.tf
3. Launch Instance
4. Make some changes to instance.tf file
5. Re-apply the terraform file – instance.tf

Step 1: Download and Install Terraform on the machine where you want to execute the terraform scripts

Step 2: Create an IAM user with administrator access in AWS cloud and configure it on the local machine using AWS configure command

**$ aws configure**

Refer **exercise\_1.tf** file to understand how you can write terraform script to provision EC2 instance on AWS cloud using terraform.

$ terraform fmt

$ terraform init

$ terraform validate

$ terraform plan

$ terraform apply

$ terraform destroy

**Variables**

* Move secrets to another file
* Values that change
  + AMI, Tags, Keypairs etc
* Reusability of the code

**Exercise 2:** You can refer exercise 2 directory to understand how to use variables and separate each block into different files

**Provisioners**

* Use Standard Image and use provisioner to setup software and files
  + File uploads
  + Remote-exec, local-exec
  + Ansible, Chef, Puppet etc.

**Exercise 3:**

$ ssh-keygen

Pass the keyname like – mykey

Refer exercise 3 directory to understand it clearly

**Exercise 4:**

Refer Exercise 4 for output.tf

**Exercise 5:**

Refer Exercise 5 for understand how to configure backend using s3 bucket to manage tfstate file.