Read Me

Libraries/Dependencies needed for running Terraform on your machine and deploying infrastructure on AWS.

1. python-pip
2. AWS CLI
3. ssh-keygen
4. Terraform

Steps required to run the Terraform files and create the infrastructure on AWS.

Step 1) Get all the files under Terraform repo <<Link>> and save it under one location.

Step 2) Brief description about the files and its usage.

main.tf 🡪 this file has infrastructure code written by using Terraform libraries for AWS. It defines the AWS components that will be deployed with Terraform. It defines the VPC, AZ, IGW, RT, Subnets, number of in each AZ EC2, type of instance Public/Private. Also through this file the relation of each component in VPC is described, for example EC2 in each AZ, how to link each subnet and Security Group, how to link Subnet with Route Tables, etc.

terraform.tfvars 🡪 Here the user can input all the dynamic variables according the requirement. Like the AWS Region in which the Terraform will deploy, CIDR Block, Public Subnet range, Key for EC2 instances and most importantly the AMI image ID.

Note: AMI image ID will change according to the region. So for example, if it is N. Virginia region then the AMI would be different than the AMI ID for same type of EC2 instance in Ohio. The user should change the region as well as the AMI ID for the desired EC2 instance type.

variable.tf 🡪 This file links the dynamic value defined in the terraform.tfvars to the main.tf by the variable definition in this file. Also, it helps to hide the dynamic value of the terraform.tfvars from other users.

Step 3) Now make the AWS CLI profile which will be used by Terraform to access the AWS through AWS CLI.

* First create a user ‘TerraformUser’ with proper permission to use CLI.
* Then download the credentials for that user in form of .csv file. This file has Access Key ID, Secret Access Key ID in it.
* Then go to the directory where all the terraform files are stored and create the AWS CLI profile for Terraform
* Commands: aws configure -- profile harsh\_narvar (When the user hits enter, it will ask for Access Key ID and Secret Access Key ID)

Step 4) Generate a SSH key pair to assign it to the EC2 user when Terraform creates it. Make sure to save the key at proper location and update that path and key name in terraform.tfvars file.

Step 5) Do Terraform init. This command will initialize the directory where all the terraform configuration.

Step 6) Do terraform plan, this will create and execution plan and This command is a convenient way to check whether the execution plan for a set of changes matches your expectations without making any changes to real resources or to the state.

Also this will create the state file (.tfvarsstate file). Terraform must store state about your managed infrastructure and configuration. This state is used by Terraform to map real world resources to your configuration, keep track of metadata, and to improve performance for large infrastructures.

Step 7) Do terraform apply. The terraform apply command is used to apply the changes required to reach the desired state of the configuration, or the pre-determined set of actions generated by a terraform plan execution plan.