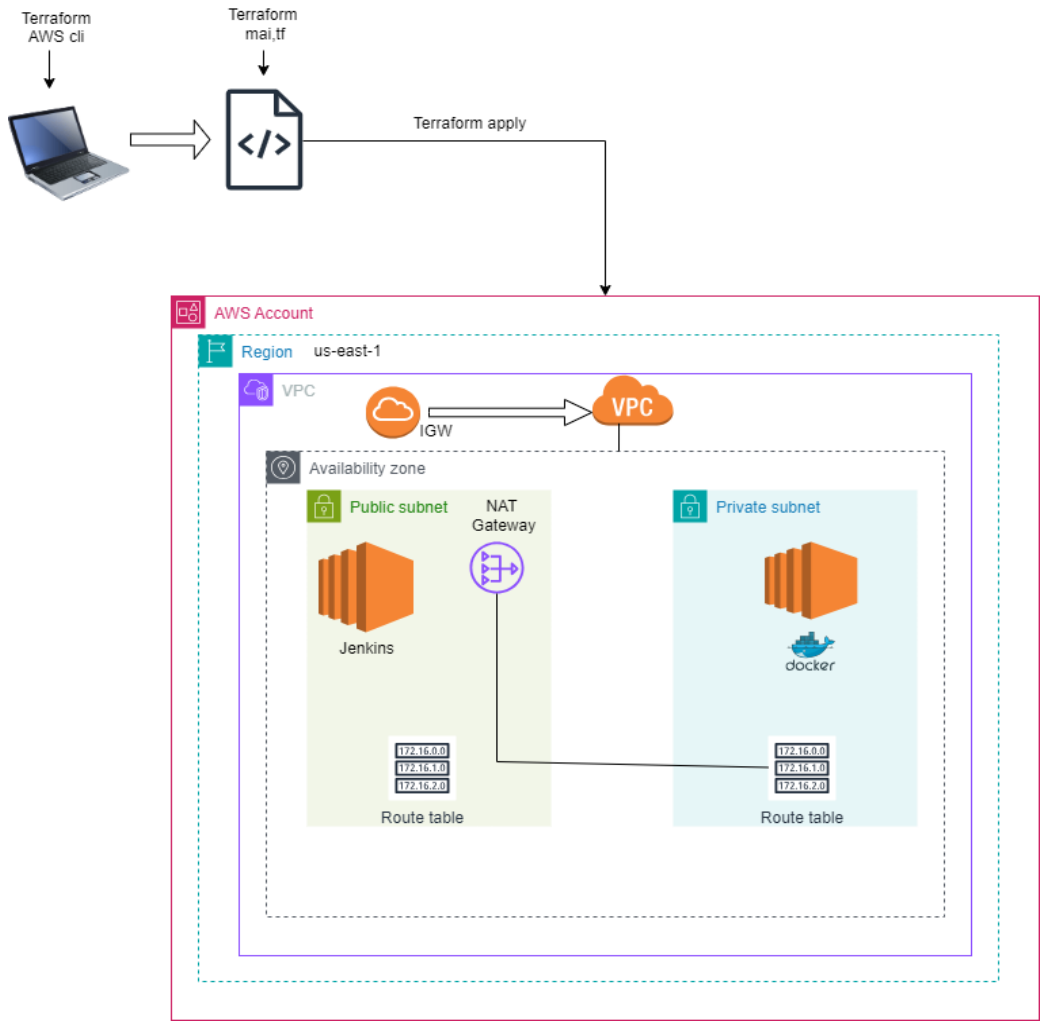
Deploying and Managing AWS resources

By using

TERRAFROM



AIM

Installing Jenkins package in Ec2 public subnet and Docker in private subnet along with security groups allowing port 22 and port 8080.

Resources : VS code, AWS cli, Terraform, Shell scripting

Procedure

1. Download and install AWS Cli in our device i.e. Laptop
   Here is the direct link for **64-bit for windows**
   https://awscli.amazonaws.com/AWSCLIV2.msi

2. Download and install **VS Code** software in our device i.e. Laptop
   Here is the direct link **for windows**
   https://code.visualstudio.com/sha/download?build=stable&os=win32-x64-user

3. Now open **VS code** and go to **extensions section** and search terraform, we will get **HarshiCOP Terraform extension** click and install it, now again search for **AWS Cli extension** and install it.

4. Now create One **IAM User** assign **AmazonFullAdministratorAccessPolicy** and **Generate Access and Secret access keys** to respective user.

5. Open IAM Service and click on Users. Select the user which we created earlier. Then it will give you Summery and click on **create Access key** Select **Command Line Interface (CLI)** and click on **next**. Then give a respected name for access key. It will show Access key and secret access key. Then **copy** that and **Save** the keys or you can **download the .csv file**.

6. Open **Terminal** in **VS code** which is located at **top 3 dots option**, and give command **aws –version** then we will get a version and then **aws configure** and give access and secret access keys which we created earlier.

7. Now **go to browser** and search for **Terraform by Harshi corp's** and install 64-bit for windows. Here is the direct link for zip folder.

   https://releases.hashicorp.com/terraform/1.9.6/terraform_1.9.6_windows_386.zip

8. After Download, select the zip folder click on **extract all** and **copy the path** of the **extracted folder**

9. Now we have to give **Environment variables to terraform**

10. Click on **windows button** and search **Edit System Environment variables** and then **select Environment variables.**

11. In **system variables select "path"** and click on **edit** and click on **new** and **paste the path of terraform** which we copied earlier, click on **OK** and close it.
12. Now run the command **terraform –version** to check weather terraform is running or not if running we will get version.
13. Now we have to create three files.
    - **provider.tf** file :- In Terraform, the **provider.tf** file is used to define and configure **providers**, which are plugins that allow Terraform to interact with external APIs, such as AWS, Azure, Google Cloud, etc. Providers are responsible for provisioning and managing the infrastructure on these platforms.
    - **main.tf** file :- This file is where you define the actual infrastructure you want to deploy. This could include things like EC2 instances, S3 buckets, databases, and more. It tells Terraform **what** to build or manage in your cloud environment (AWS, Azure, etc.)
    - **variables.tf** file :- This file contains the definitions of input variables that can be used throughout your Terraform configuration. It helps you define values that can be changed easily without modifying the core logic of your main.tf or other files.

14. Here is my GitHub project repository URL. Where the above 3 files are stored.

    https://github.com/Mamatha7316/Terraform-project.git
15. After creating the files go to terminal and run following commands
    - **terraform init** ------ Prepare your working directory for other commands
    - **terraform validate** ------ Check whether the configuration is valid
    - **terraform plan** -------- Show changes required by the current configuration
    - **terraform apply** or **terraform apply -auto-approve** ---- Create or update infrastructure

16. For verification, login to created account and check the resources which we mentioned in **main.tf** file, we will see the resources created by terraform automatically. Connect **Jenkins instance** through direct **amazon linux** and copy the public IP address in terminal and paste it on new tab with port number **:8080** and Connect **Docker instance** through **ssh** port number is **:22**

17.. If we want to delete resources which we created now then run the following command in VS code terminal so that everything which we created now will be deleted automatically
    - **terraform destroy -----** Destroy previously-created infrastructure