Infrastructure-as-Code using Terraform

Terraform – Assignment 2 (Create an EC2 instance on AWS Platform)

Note(s):

1. *This Lab would need you to have an account on Amazon Web Services (AWS). If you don't have an account on AWS, create one using following link:*
   Sign up for AWS

2. *Ensure the environment clean-up for your AWS resources once your project is completed to avoid incurring any unnecessary charges.*

Task 1: (Install and configure Terraform for AWS)

Install Terraform on a VM of your choice. You can also do this on your laptop. Terraform can be installed on all major platforms such as MacOS, Windows and Linux.

Check the following link(s), if needed:

https://docs.digitalocean.com/reference/terraform/getting-started/

Task 2 (Getting AWS credentials)

In order to provision resources in AWS you need to generate aws_access_key and aws_secret_key for your username you need to provide a key pair name and the specific path on your system where the private key is stored.

You must provide your AWS access keys to make programmatic calls to AWS or to use the AWS Command Line Interface or AWS Tools for PowerShell.

When you create your access keys, you create the access key ID (for example, AKIAIOSFODNN7EXAMPLE) and secret access key (for

example, wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY) as a set. The secret access key is available for download only when you create it. If you don't download your secret access key or if you lose it, you must create a new one.

Steps to get AWS access keys:

1. Sign in to the AWS Management Console as the root user. For more information, see Sign in as the root user in the *IAM User Guide*.
2. In the navigation bar on the upper right, choose your account name or number and then choose My Security Credentials.
3. Expand the Access keys (access key ID and secret access key) section.
4. To create an access key, choose Create New Access Key. If you already have two access keys, this button is disabled and you must delete an access key before you can create a new one. When prompted, choose either Show Access Key or Download Key File. This is your only opportunity to save your secret access key. After you've saved your secret access key in a secure location, chose Close.

Task 3 (Write terraform file to provision Resources in AWS)

1. Write a terraform file which shall provision an EC2 instance on AWS inside a default VPC with following specifications:
   - Instance type - t2.micro
   - AMI - AWS linux [latest version]
   - Region - us-east-1 [Not mandatory]
2. You need to create AWS security groups which will be attached to EC2 allowing to connect to the instance via SSH and open port 80 so that the server is accessible over HTTP.
3. All resources need to be created inside default VPC
4. You need to install Apache Web Server (httpd) on the provisioned EC2 instance and start the httpd to enable it as a webserver (hint: you can leverage “remote-exec” provisioner to install and start apache on the EC2 instance)

   “sudo yum install httpd -y”,
   “sudo service httpd start”

5. As Terraform Output you need to configure the public DNS of the EC2 instance.

Task 4 (Provision Resources in AWS using terraform init/plan/apply)

1. Once you have written the file and saved it, login to your terminal and go the folder where you have written the terraform file.
2. Run “terraform init” to initialize the environment and all providers. Troubleshoot, if necessary.
3. Run “terraform plan” to do a dry run of what you are about to do. Go through the output and verify changes you are about to make. Troubleshoot, if necessary.
4. Run “terraform apply” to provision the resources. Troubleshoot, if necessary.
5. Login to AWS Console on a browser of your choice and verify that the EC2 instances and other resources has been created successfully.
6. Goto terraform Output and copy the public IP of the EC2 instance and paste it in a browser as below – http://x.x.x.x:80 and check if the httpd page is appearing or not to validate if the EC2 instance is enabled as a webserver.

Task 5 (Terminate all resources)

1. Use terraform destroy command to terminate all resources provisioned as part of the assignment.
2. Login to AWS console and verify if the terraform was successfully able to terminate all resources.