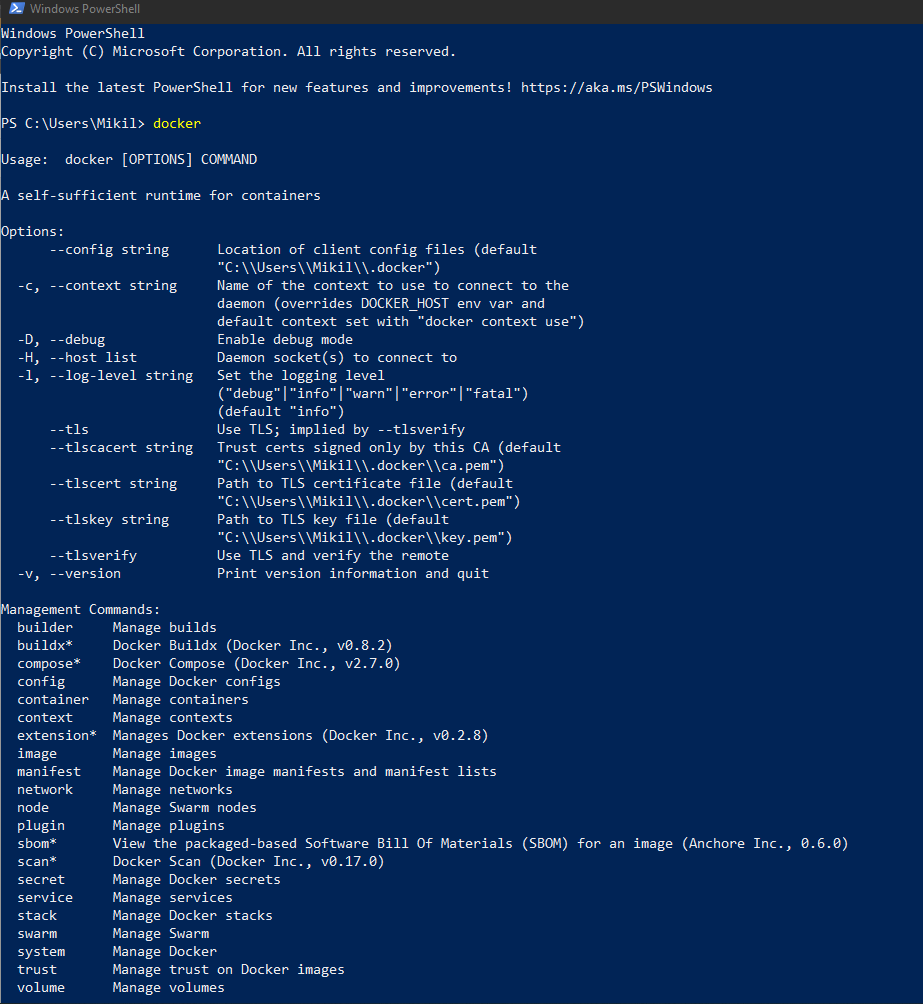
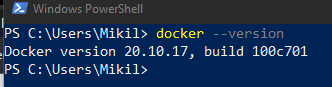
**Steps-**

**Creating Docker container using terraform**

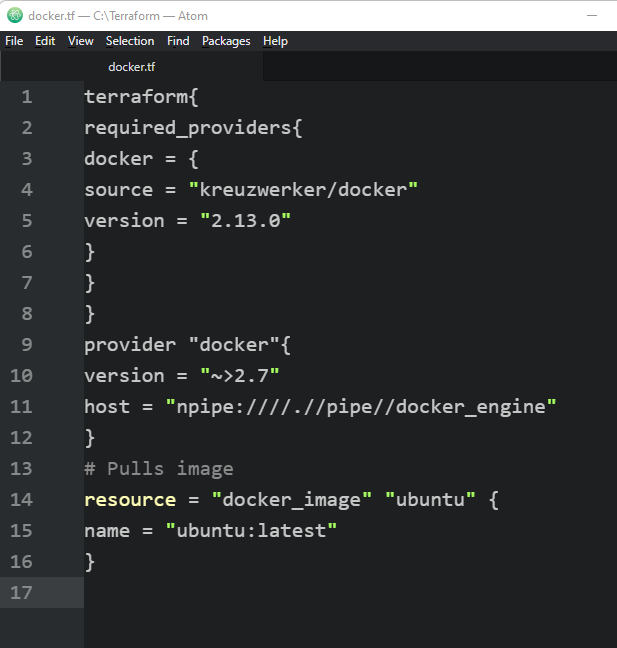
Step 1: Check the docker functionality

****

****

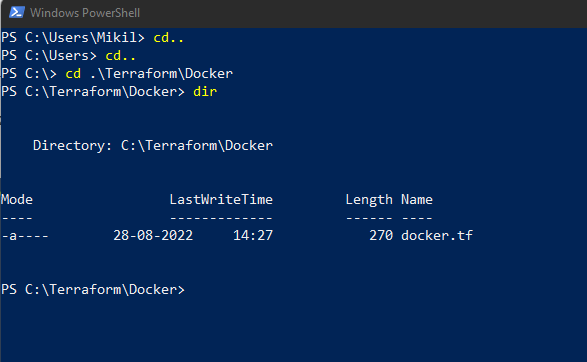
Step 2: Write a terraform script to create a Ubuntu Linux container

Create a new docker.tf file using Atom editor and write the following contents into it.

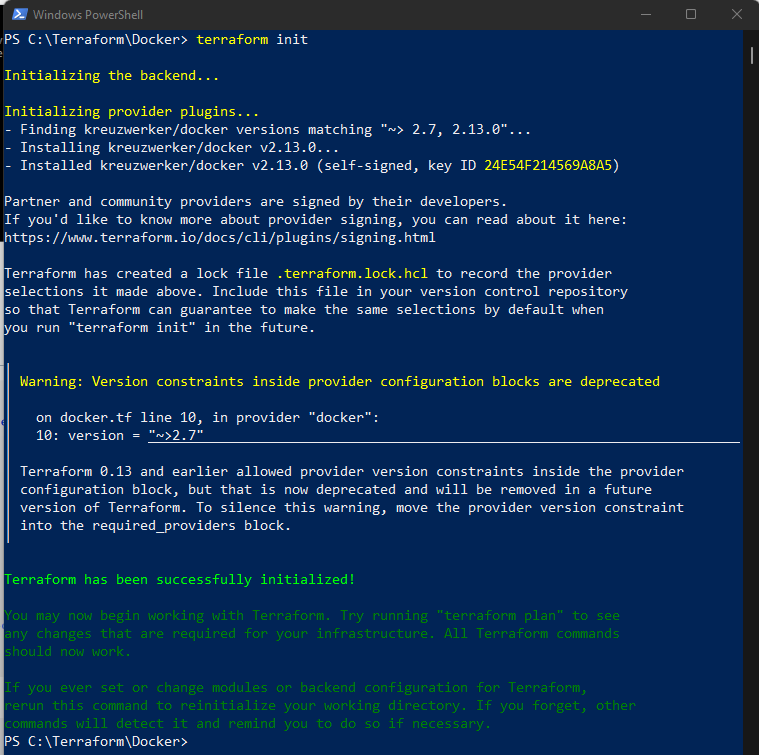
****

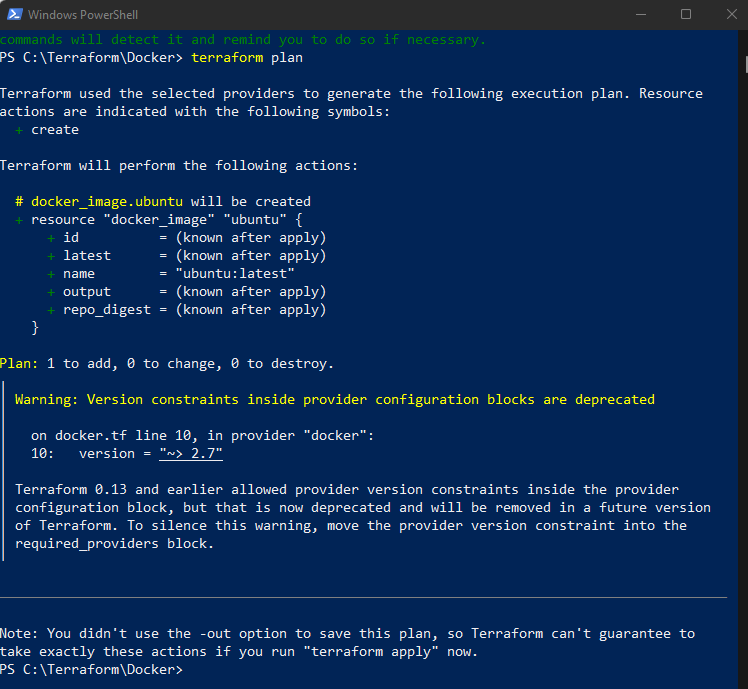
Save the file in a new directory called docker where rest of the terraform scripts are stored

Step 2: Open Command Prompt and go to Terraform\_Script\docker directory where our .tf file is stored

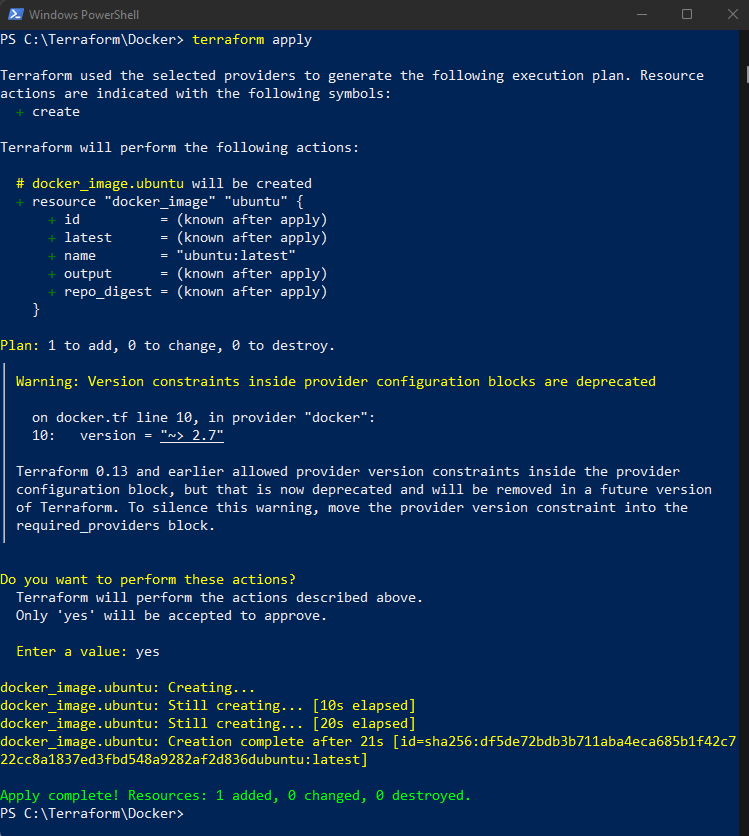
****

Step 3: Execute Terraform Init command to initialize the resources

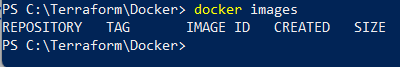
****Step 4: Execute Terraform plan to see the available resources

****

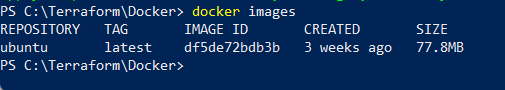
Step 5: Execute Terraform apply to apply the configuration, which will automatically create and run the ubuntu Linux container based on our configuration.

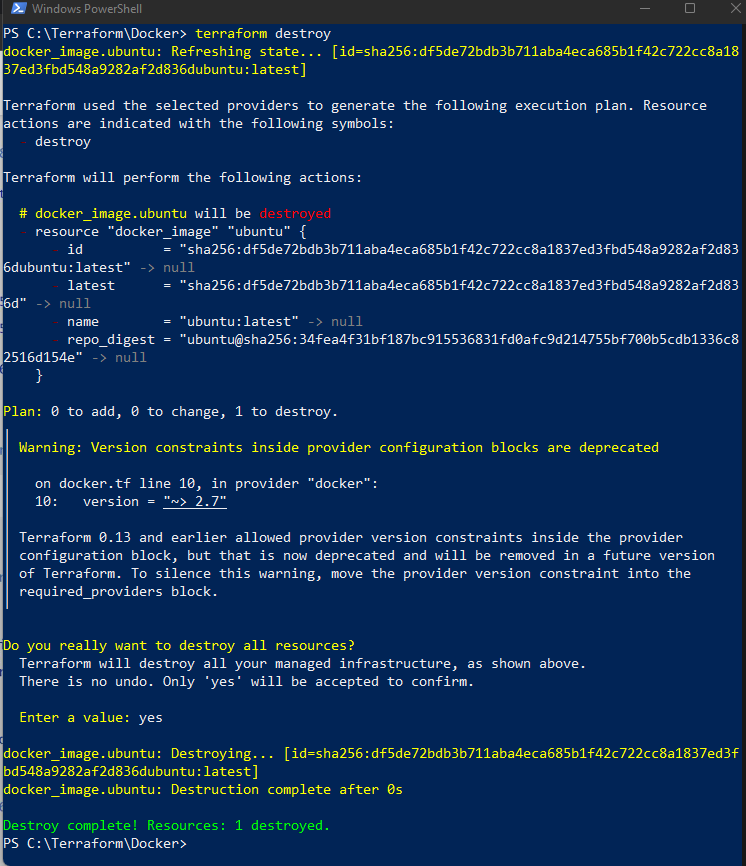
****

Docker images Before Executing Apply command:

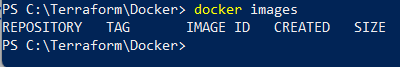
****

Docker images, After Executing Apply step:

****

Step 6: Execute Terraform destroy to delete the configuration, which will automatically delete the Ubuntu Container****

Docker images After Executing Destroy step:

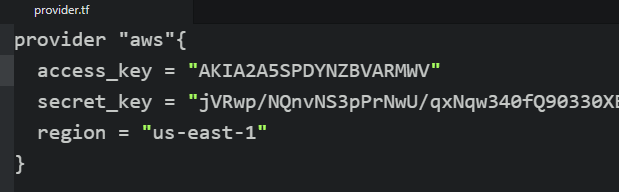
****

**Creating S3 Bucket using terraform**

Step 1: Write a Terraform Script in Atom for creating S3 Bucket on Amazon AWS

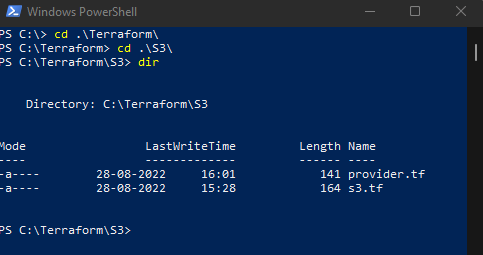
****

Create a new provider.tf file and write the following contents into it.

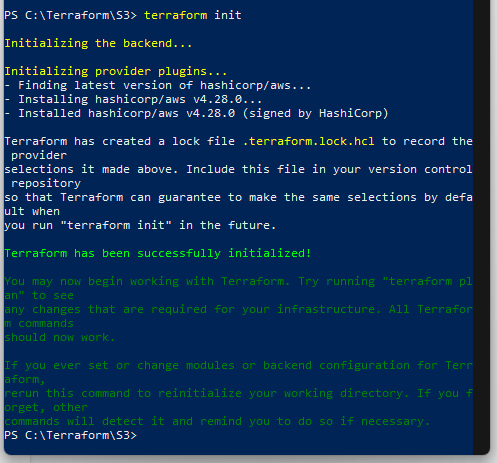
****

Save both the files in the same directory Terraform/S3

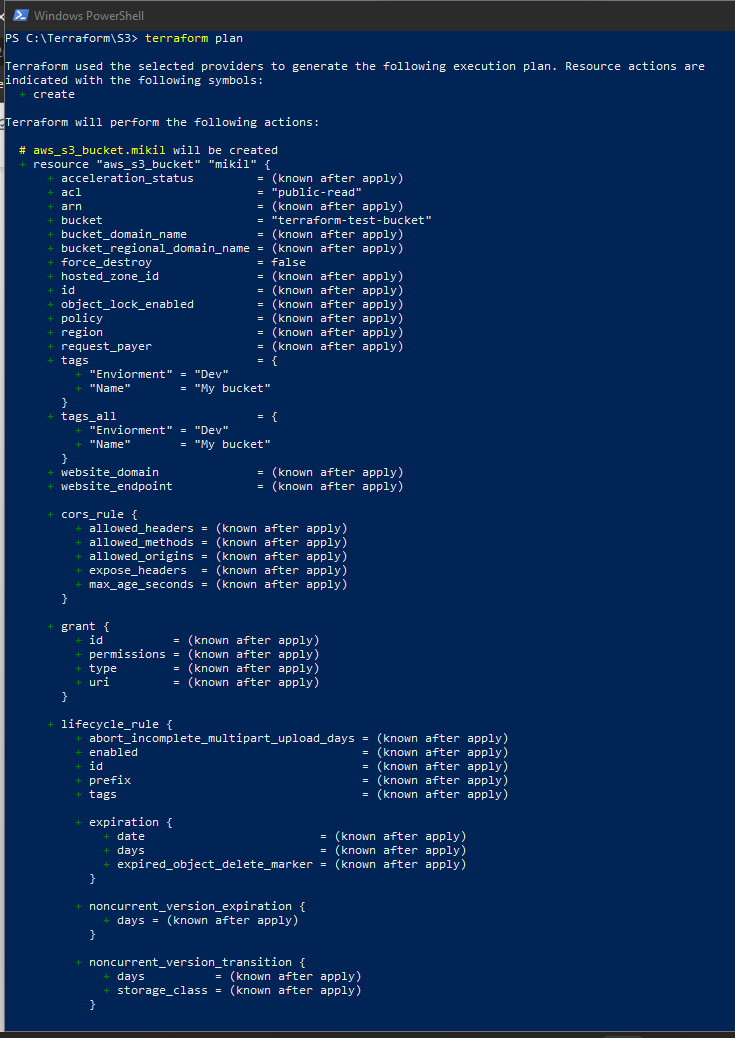
Step 2: Open Command Prompt and go to Terraform\S3 directory where our .tf files are stored

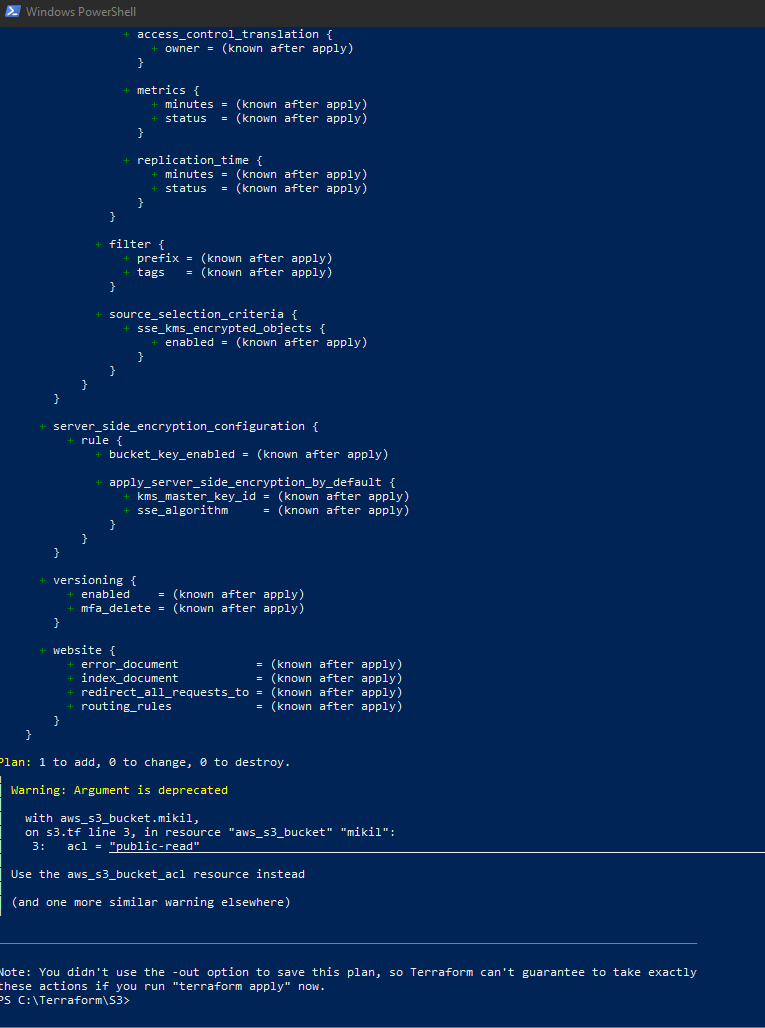
****

Step 3: Execute Terraform Init command to initialize the resources

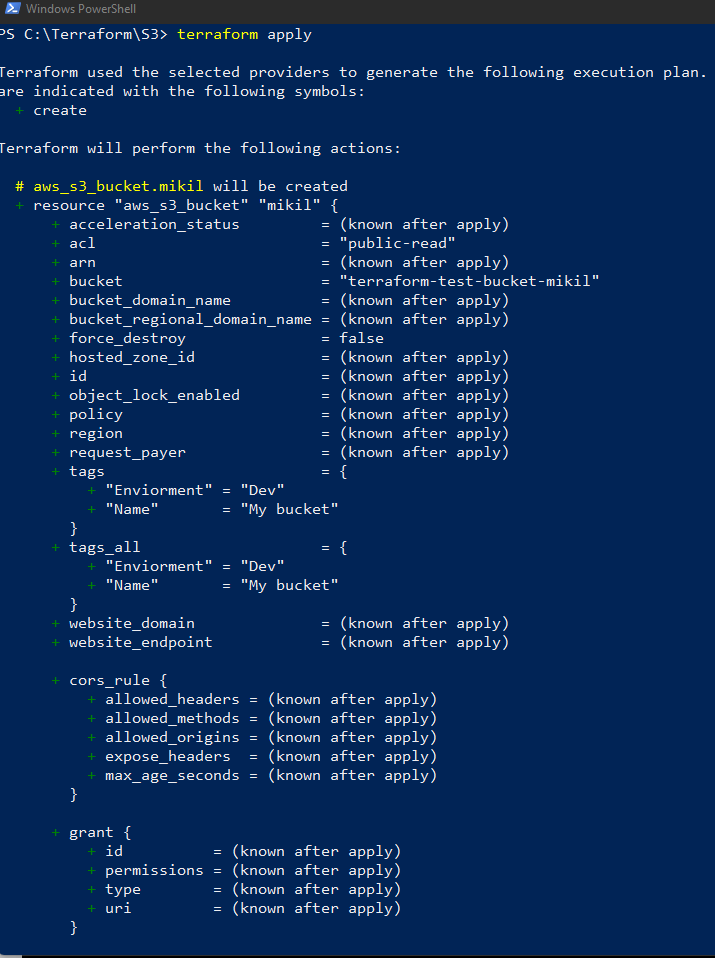
****

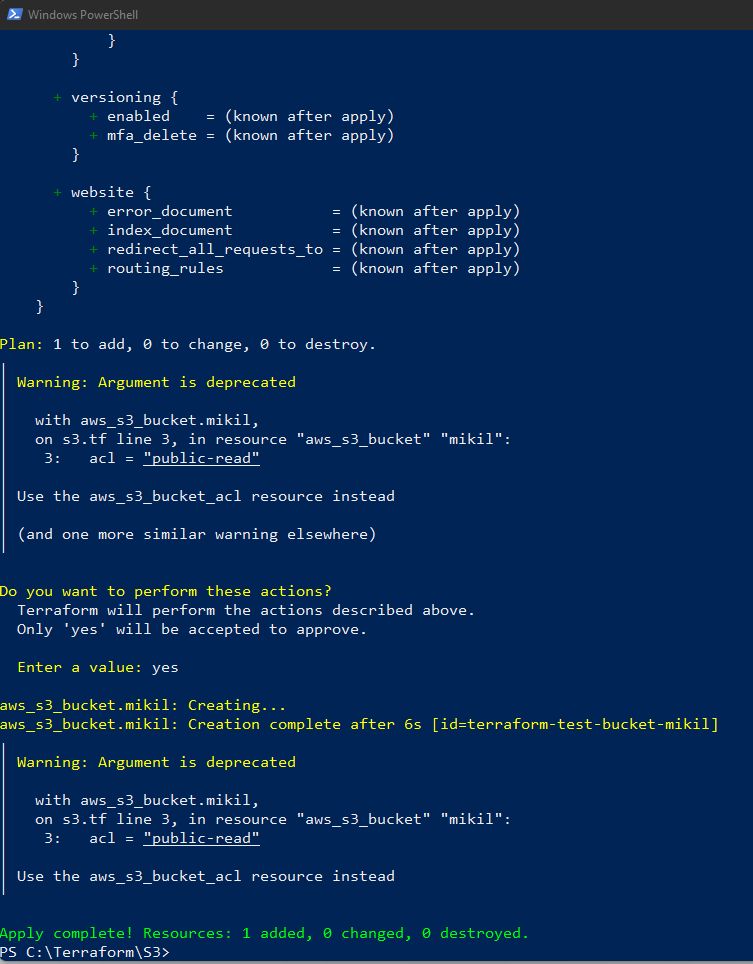
Step 4: Execute Terraform plan to see the available resources

****

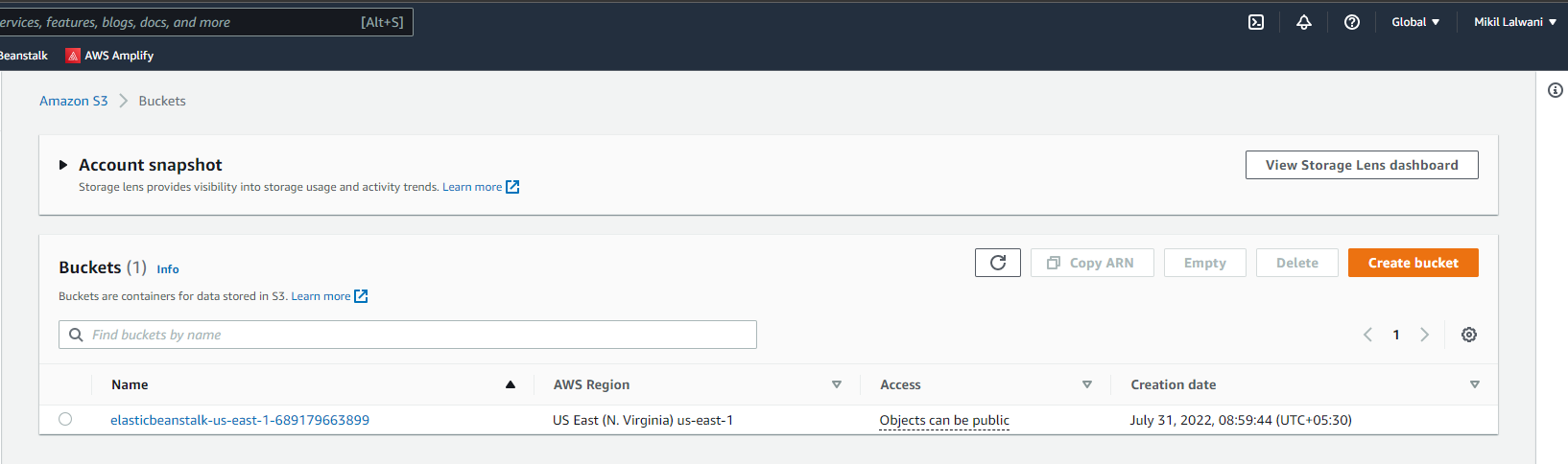
****

Step 5: Execute Terraform apply to apply the configuration, which will automatically create an S3 bucket based on our configuration.

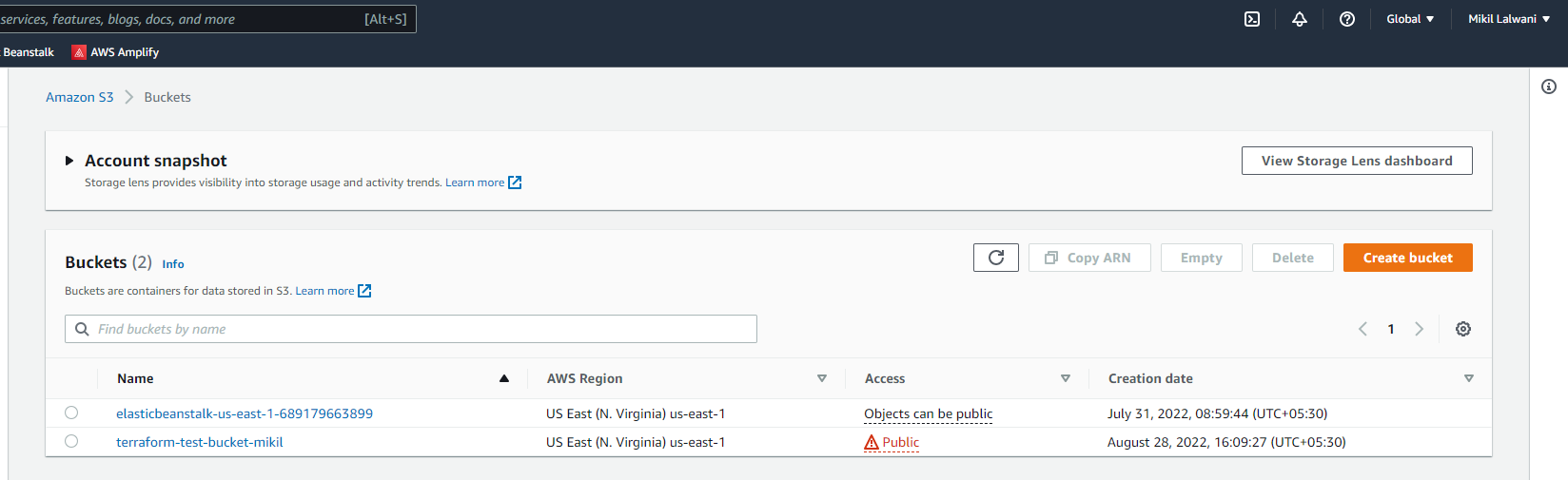
****

****

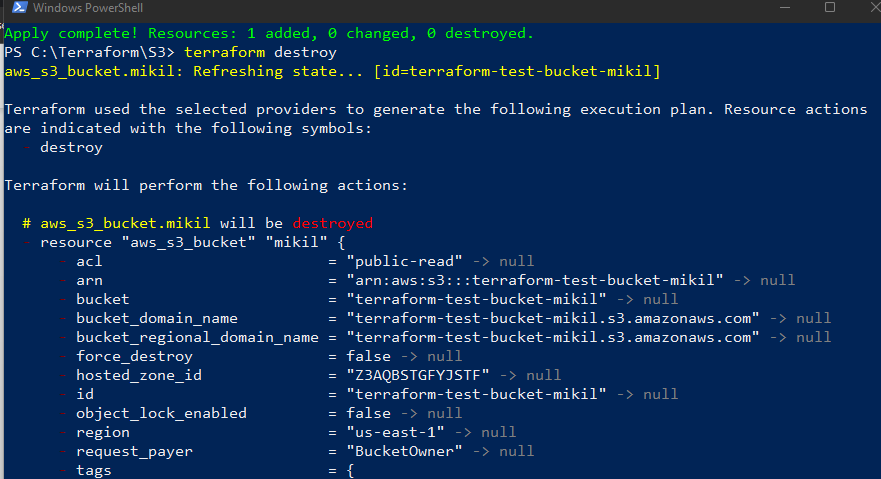
AWS S3bucket dashboard, Before Executing Apply command:

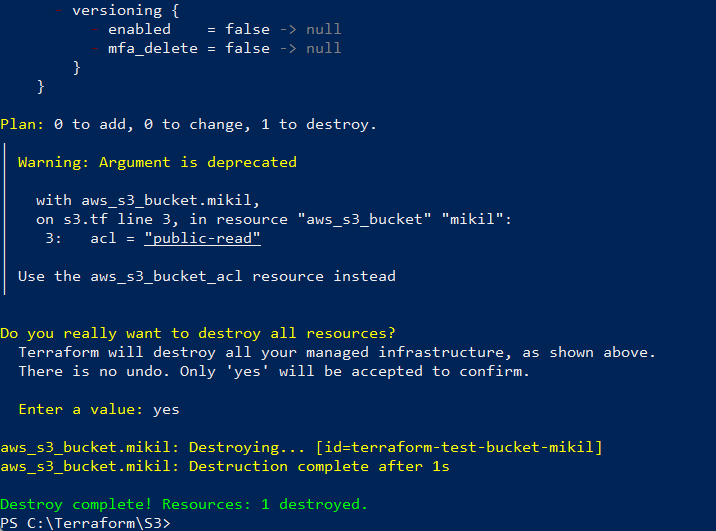
****

AWS S3 Bucket dashboard, After Executing Apply step:

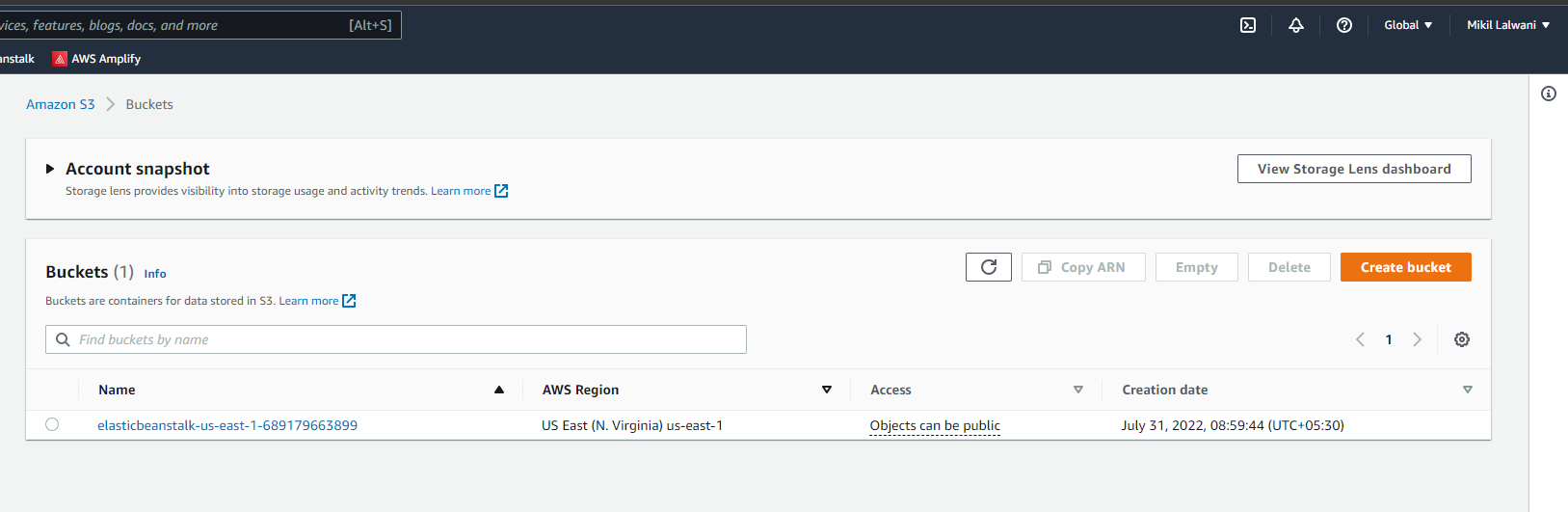
****

Step 6: Execute Terraform destroy to delete the configuration, which will automatically delete an EC2 instance

****

****

AWS EC2 dashboard, After Executing Destroy step:

****

**Creating EC2 instance using terraform**

Step 1: Create a Working directory called “Terraform\_Scripts” in C:\ drive for storing all the Terraform scripts

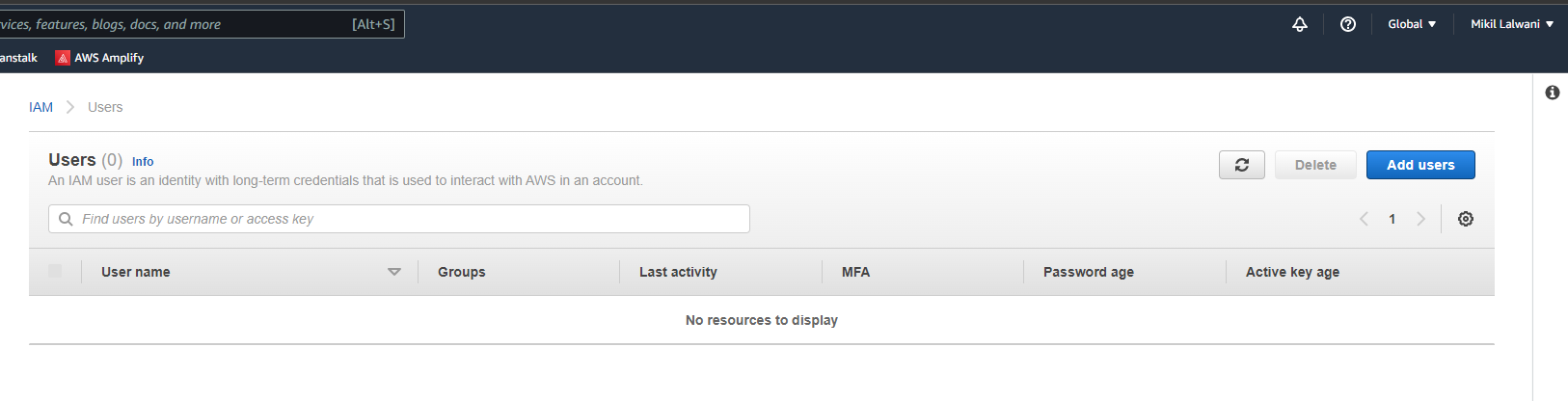
Step 2: Open Atom Editor and Open newly created folder “Terraform\_Scripts” in it

Step 3: Create a new file called “EC2\_on\_Terraform” in it

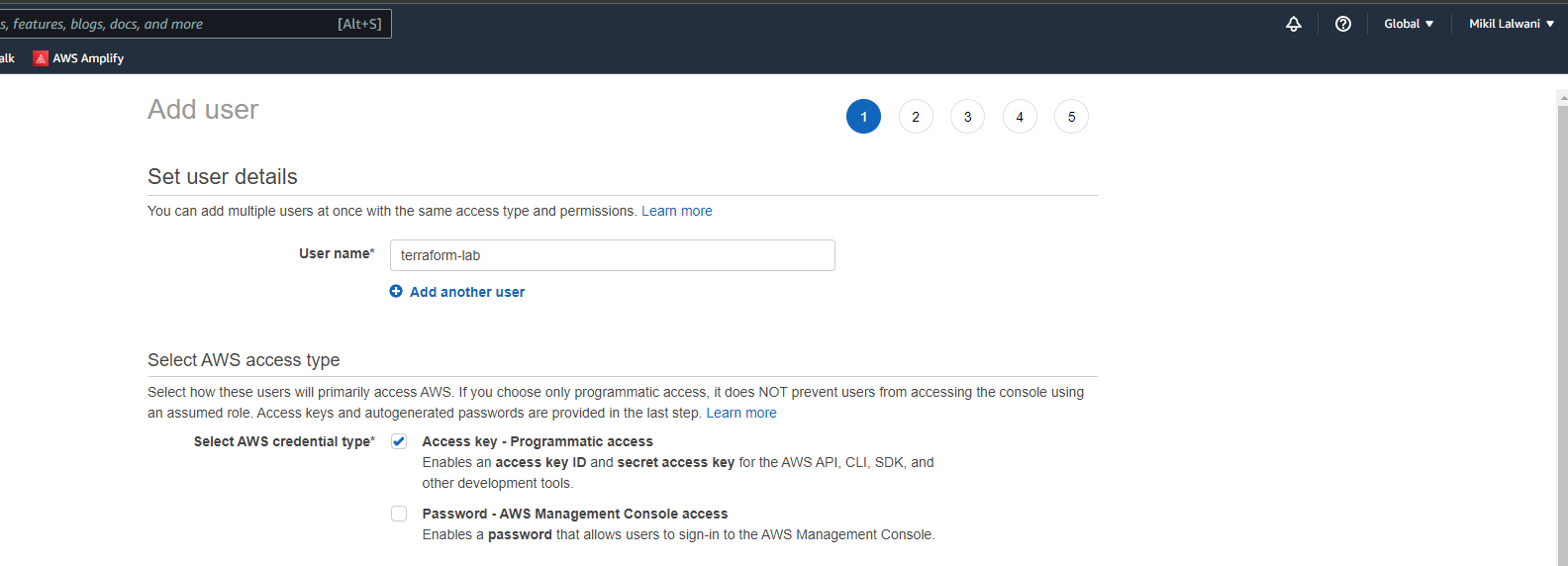
Step 4: Open AWS Console and Search for IAM to get Access Key ID and Secrete Key

Open AWS Console: <https://aws.amazon.com/console>, Provide the credentials to Log in to Cloud Portal.

Now, search for IAM to get the Access Key ID and Secrete Key

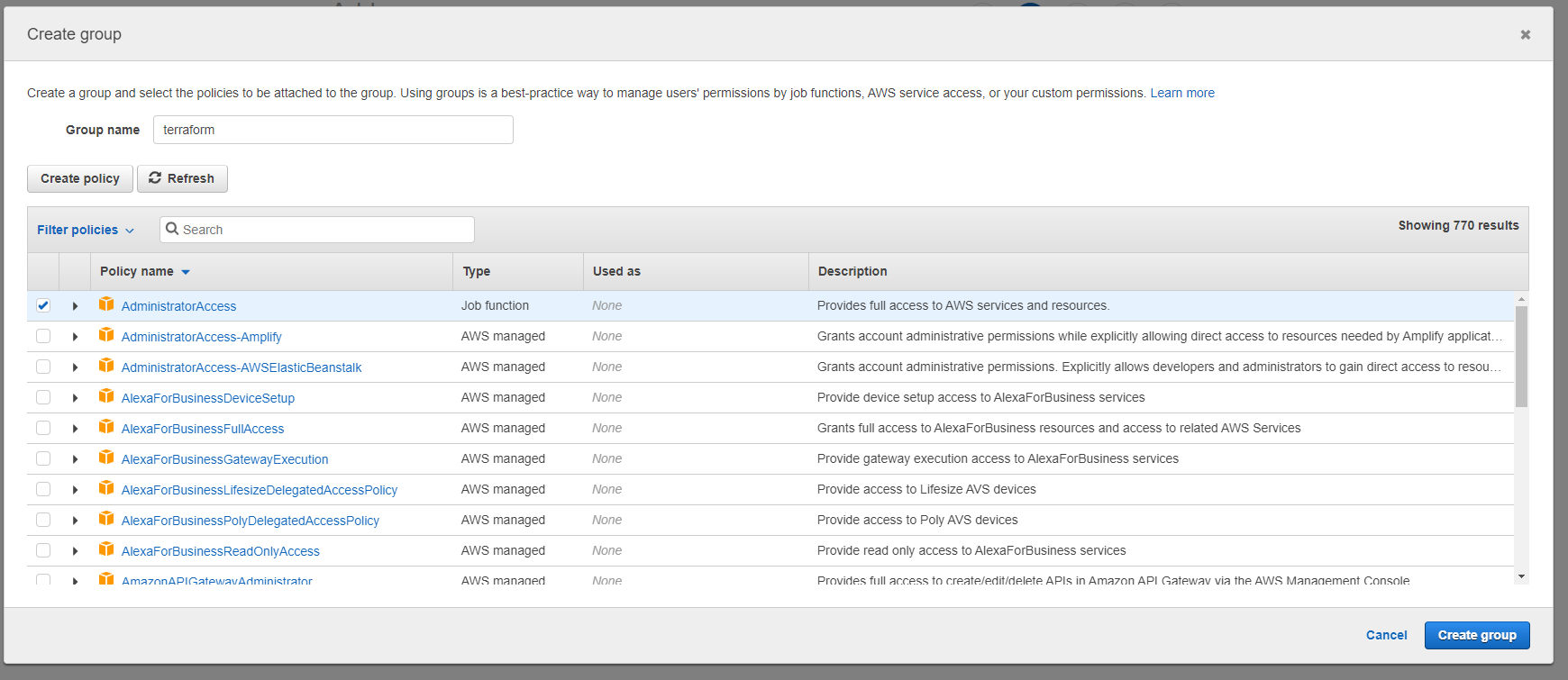
****

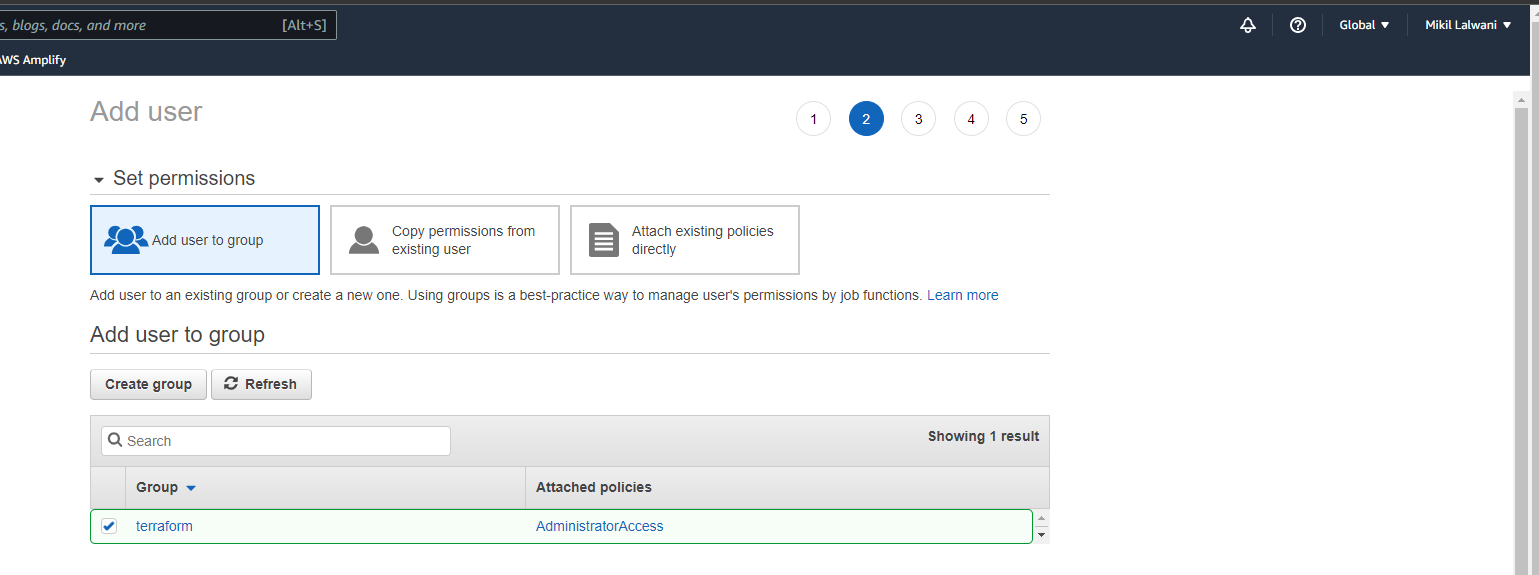
Create a new user Terraform\_user and select option Programmatic Access then click on next for “Permissions”

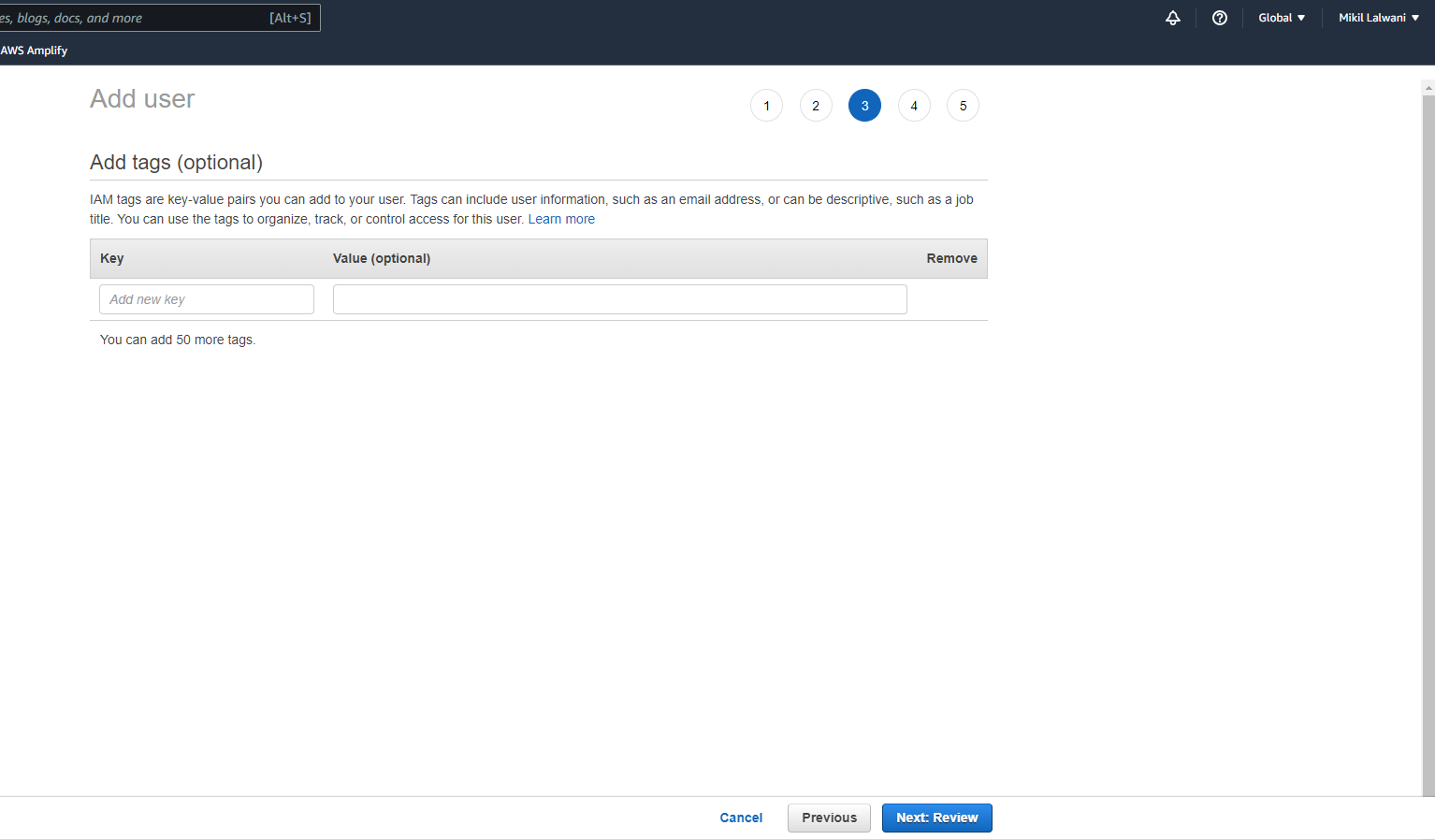
****

Now, Create a group “Terraform\_Group” and assign permission as “Administrator Access”

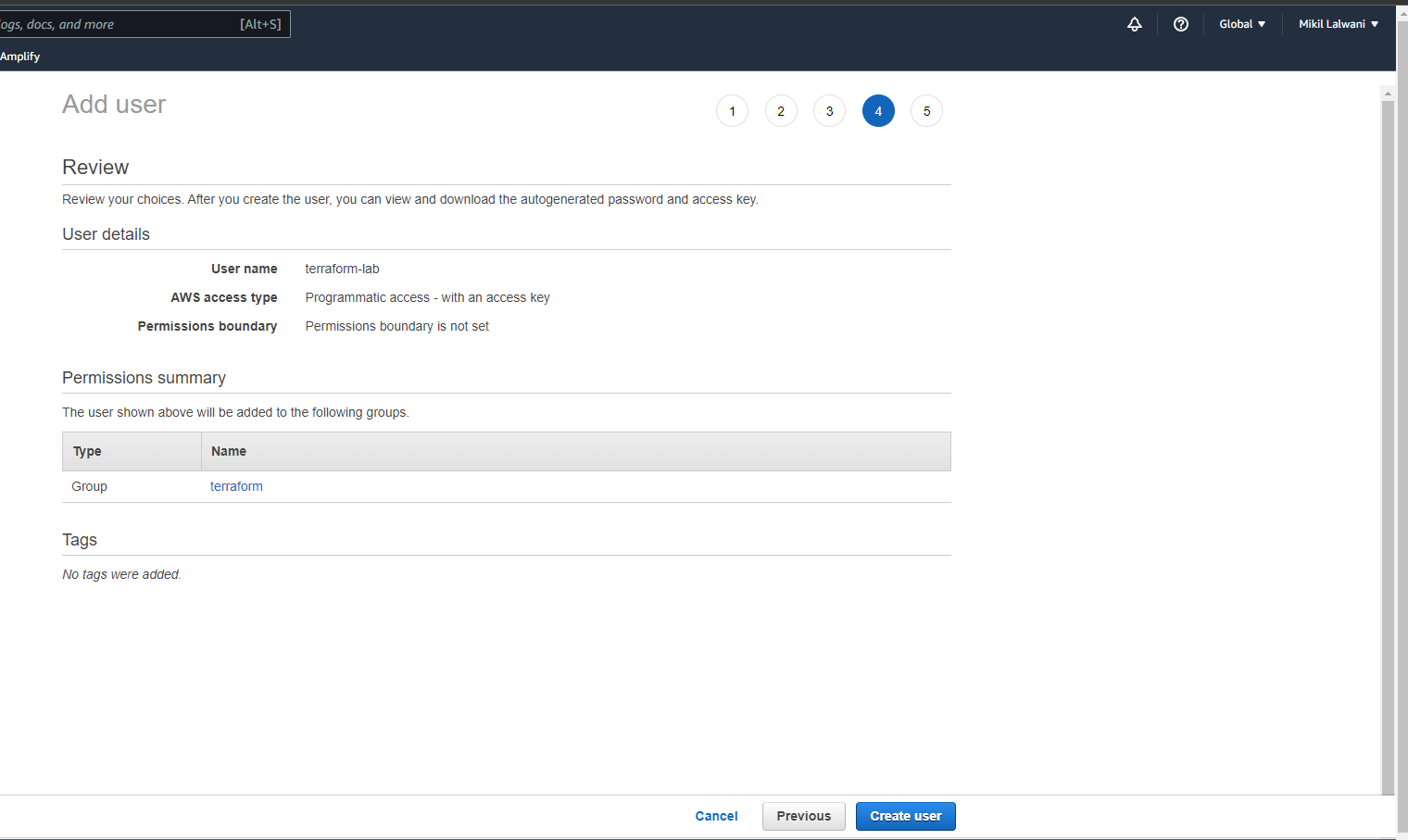
Now, Click on Create group followed by click on next for apecifying a Tag which is Optional

****

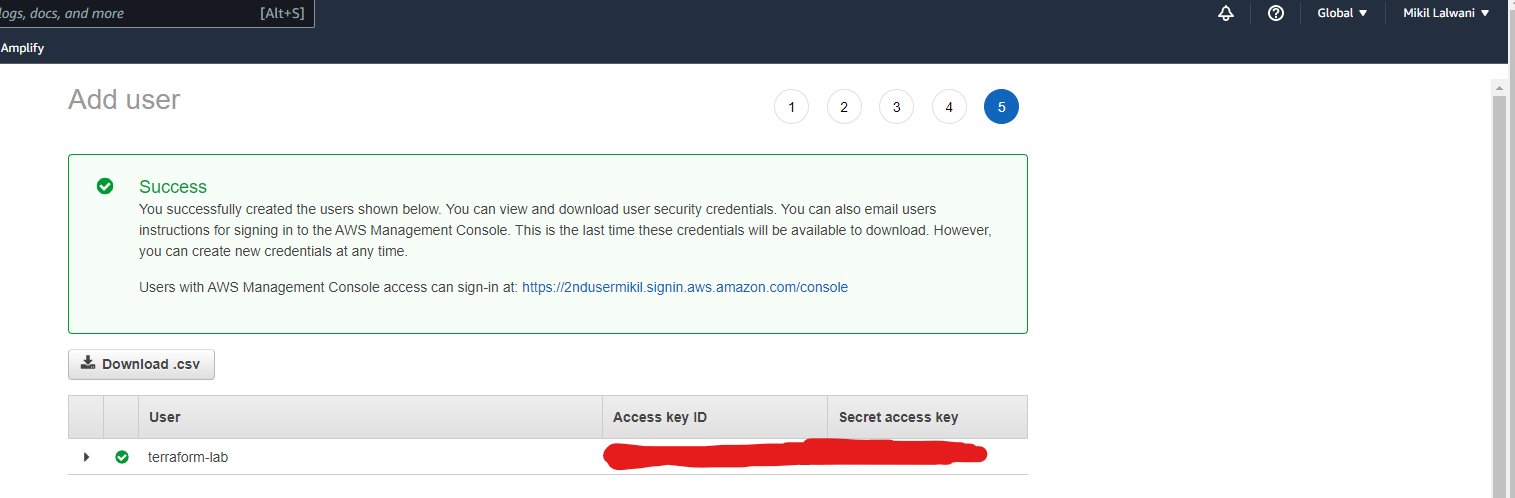
****

****

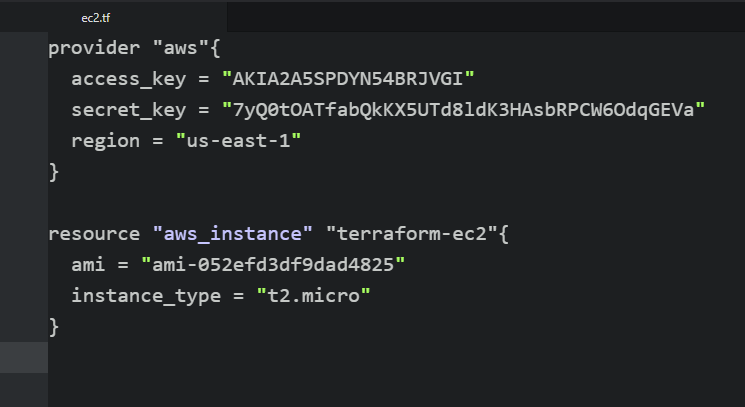
Finally, Click on Next and Create user button to Create a New User.

****

Now, Copy the Access Key ID and Secret Access Key for using it in Terraform Script,or Download .CSV file and save it for using Access Key ID and Secret Access in Key for Terraform Script

****

Step 5: Write Terraform Script in Atom for creating a EC2 instance using an automated Script

****

In this Script, access\_key = " ", secret\_key = " " and ami = " "

needs to be specified as per Operating ststem and EC2 instance id.

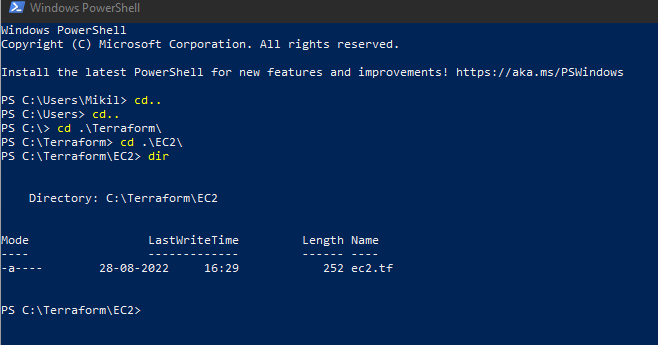
The access\_key and secret\_key can be used from previous step i.e. using CSV file or copied from user section of IAM service.

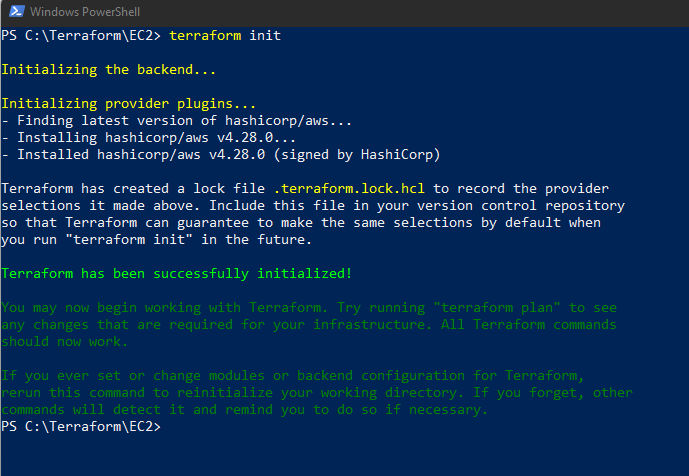
AMI stands for Amazon Machine Image which is the id of EC2 Virtual machine instance which can be copied from AWS EC2 service ami = " "

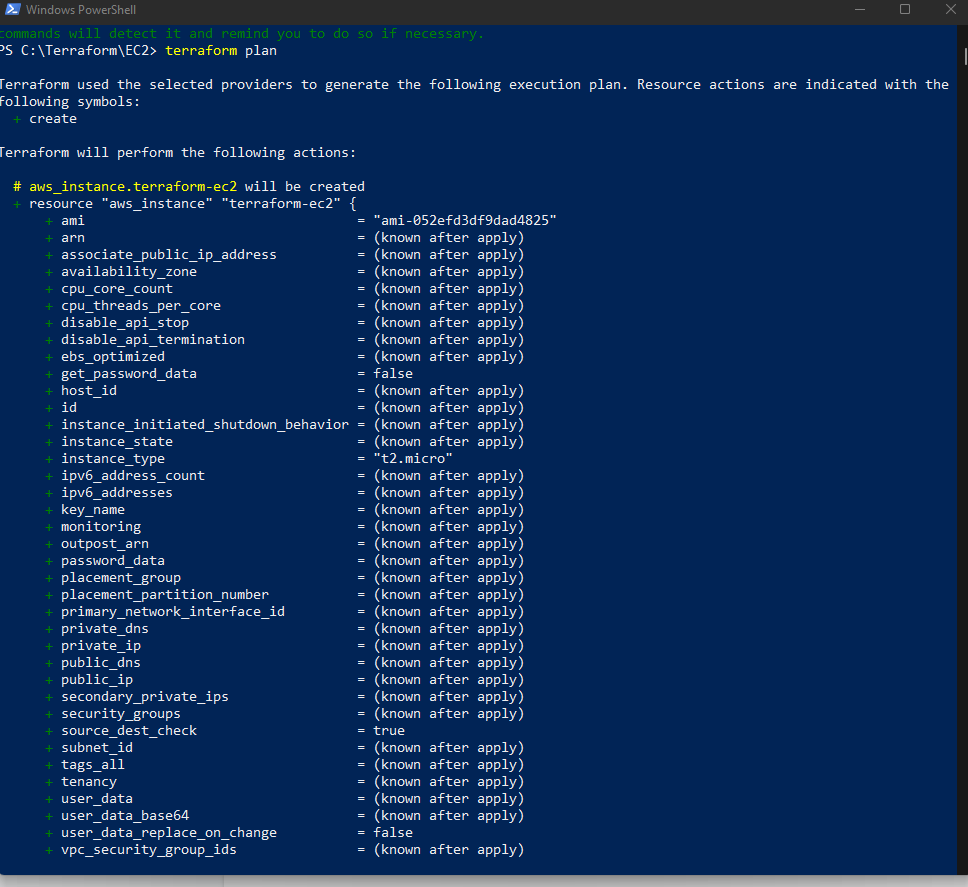
To get AMI, First open AWS console and open EC2 service.

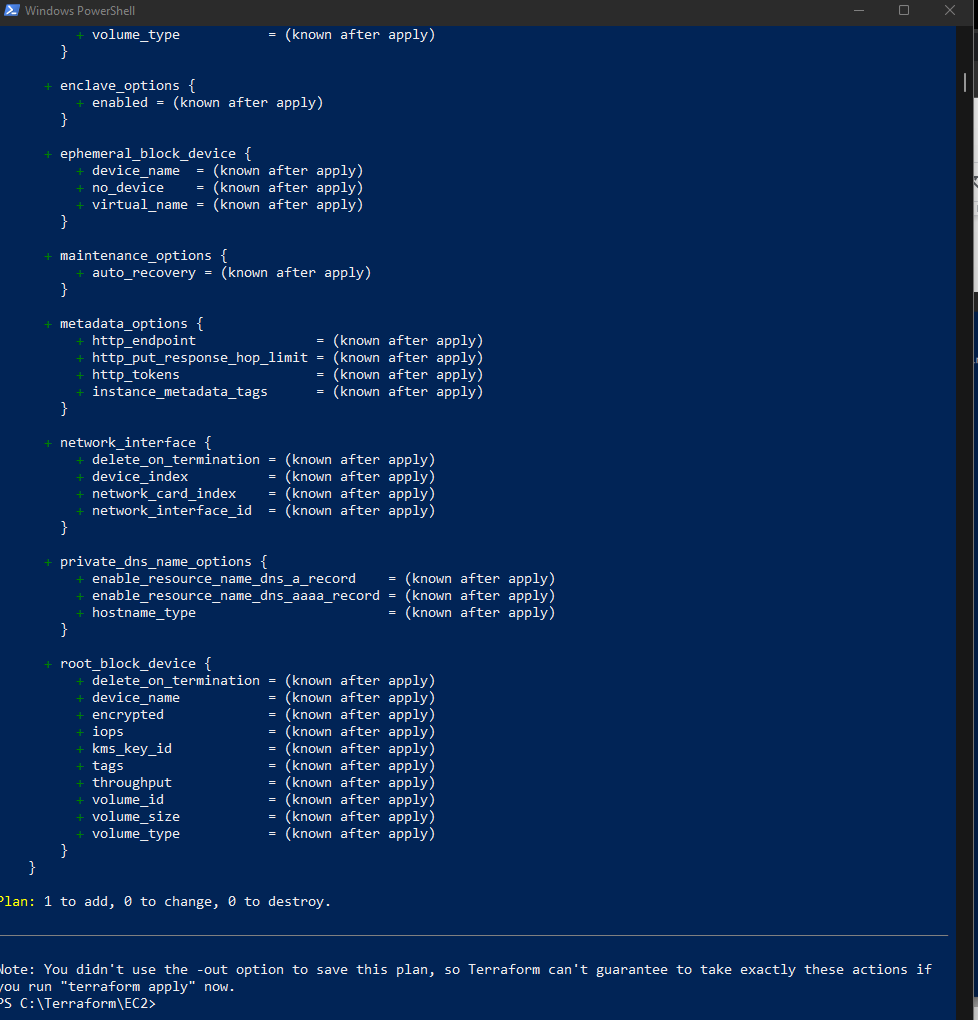
Click on Launch instance, which will show you list of Operating systems for which EC2 instance to be created. Copy the AMI id of an image for which instance to be created and paste it into our terraform Script. [Note: Ami changes region to region, so see the region before copying AMI which is mentioned in the script, in our example it is us-east-1]

Step 7: Open Command Prompt and go to Terraform\_Script directory where our .tf files are stored

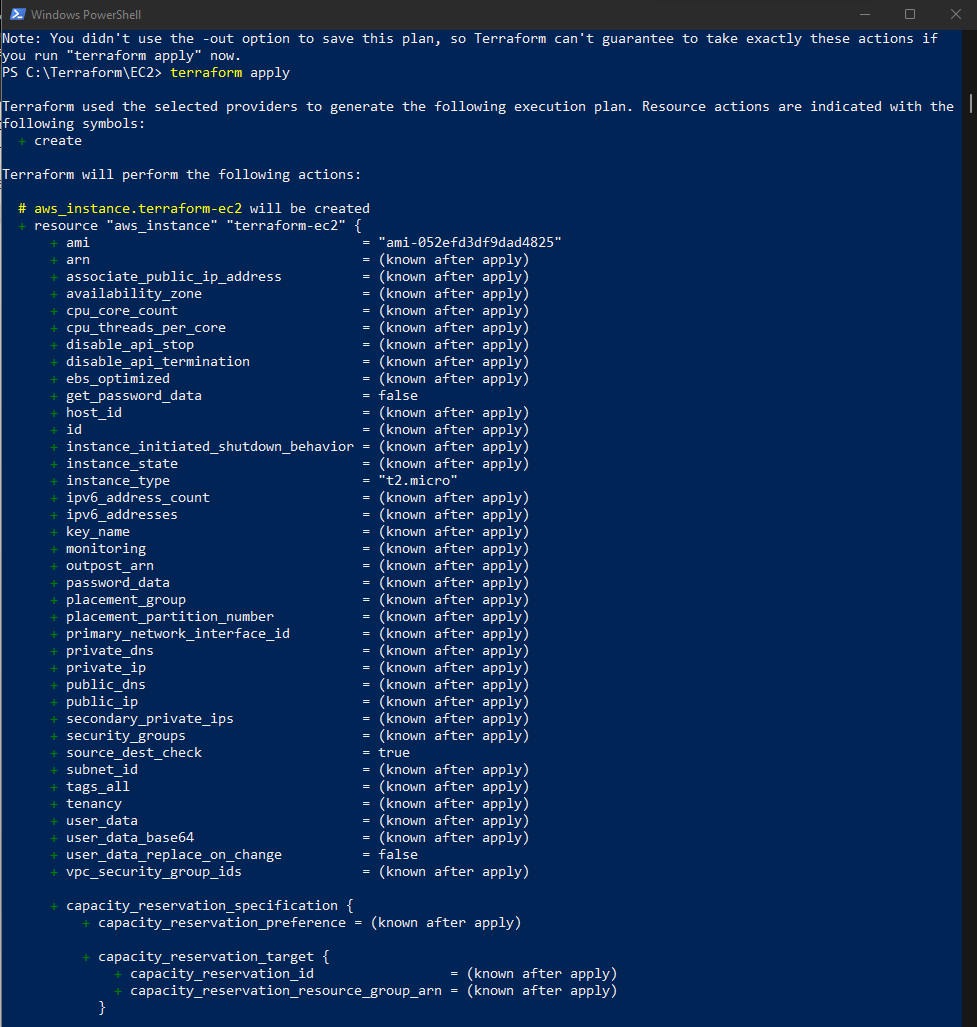
****

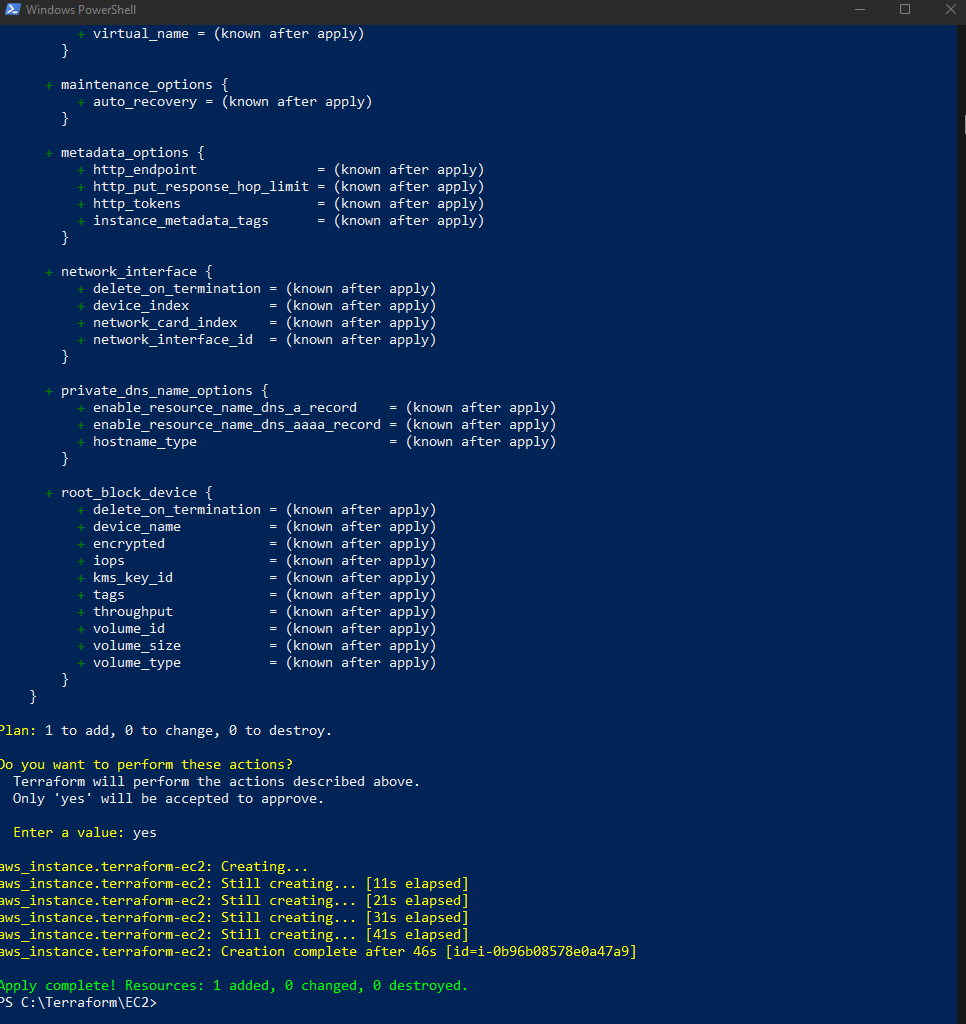
Step 8: Execute Terraform Init command to initialize the resources****

Step 8: Execute Terraform plan to see the available resources****

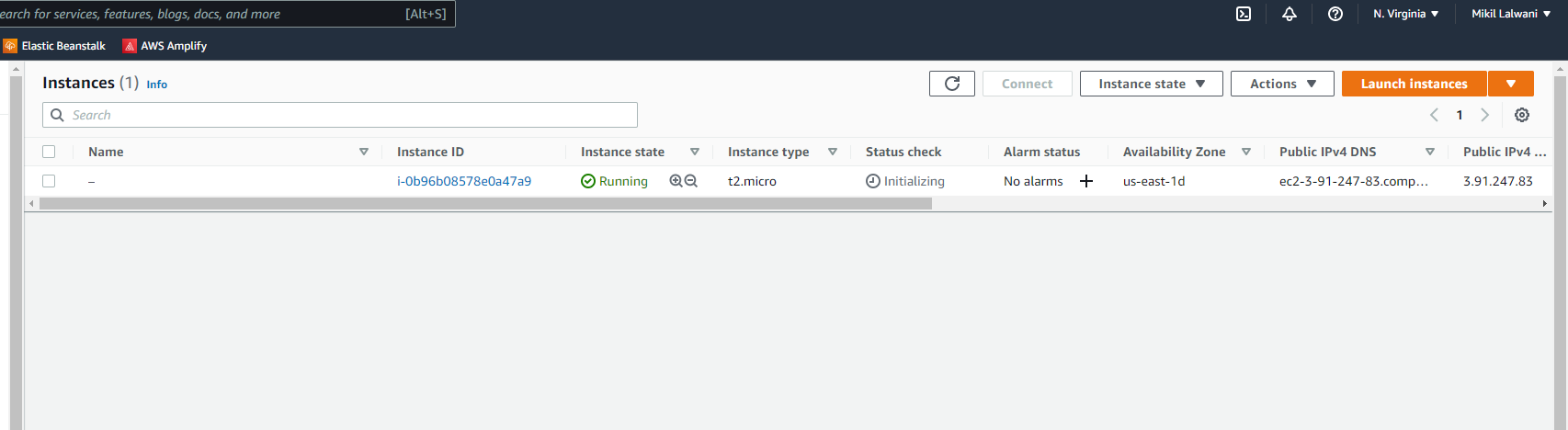
****

Step 9: Execute Terraform apply to apply the configuration, which will automatically create an EC2 instance based on our configuration.

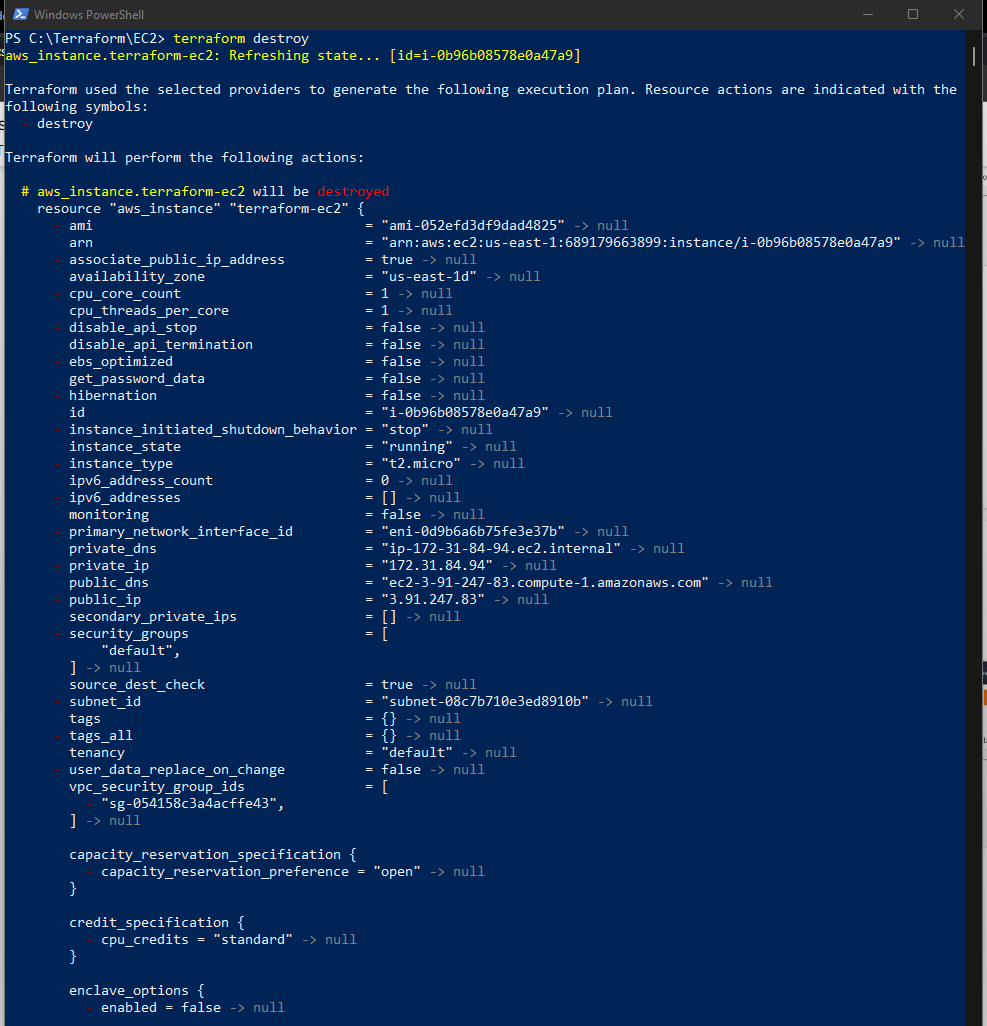
****

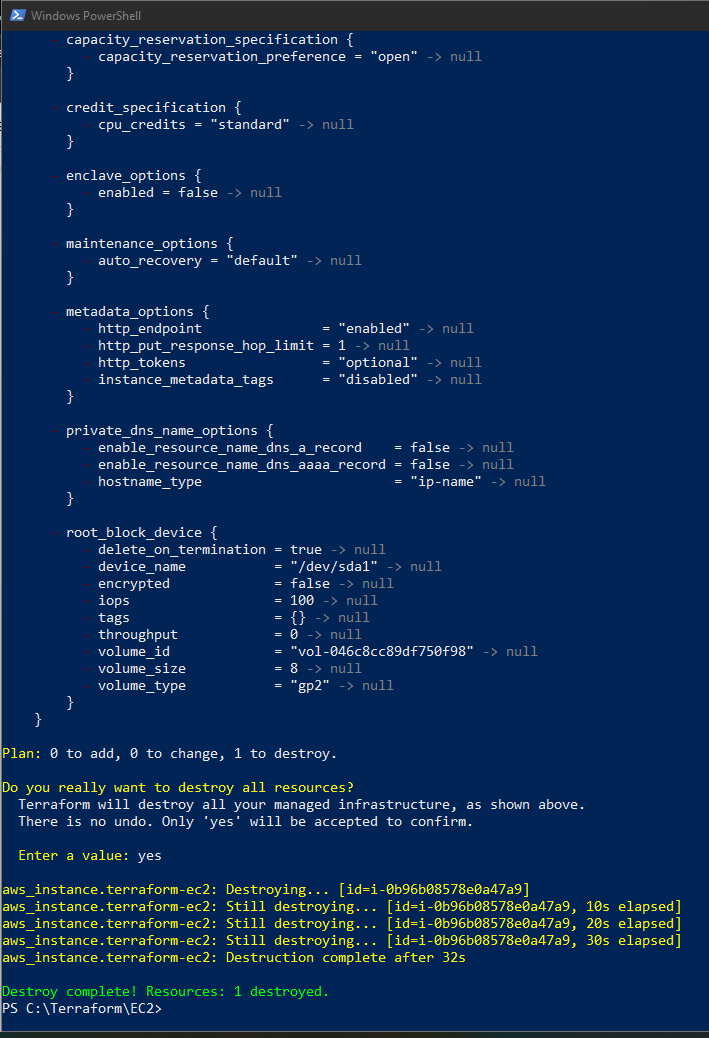
****

AWS EC2 dashboard, After Executing Apply step:

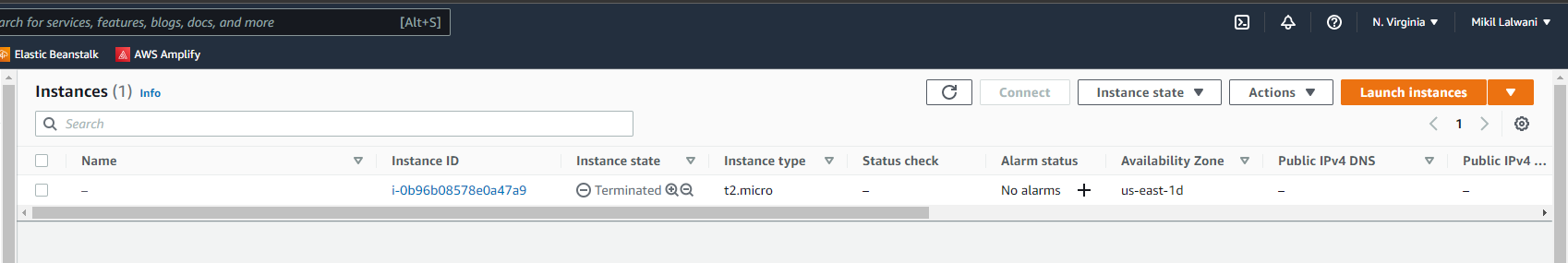
****

Step 10: Execute Terraform destroy to delete the configuration, which will automatically delete an EC2 instance

****

****

AWS EC2 dashboard, After Executing Destroy step:

****