Experiment 6

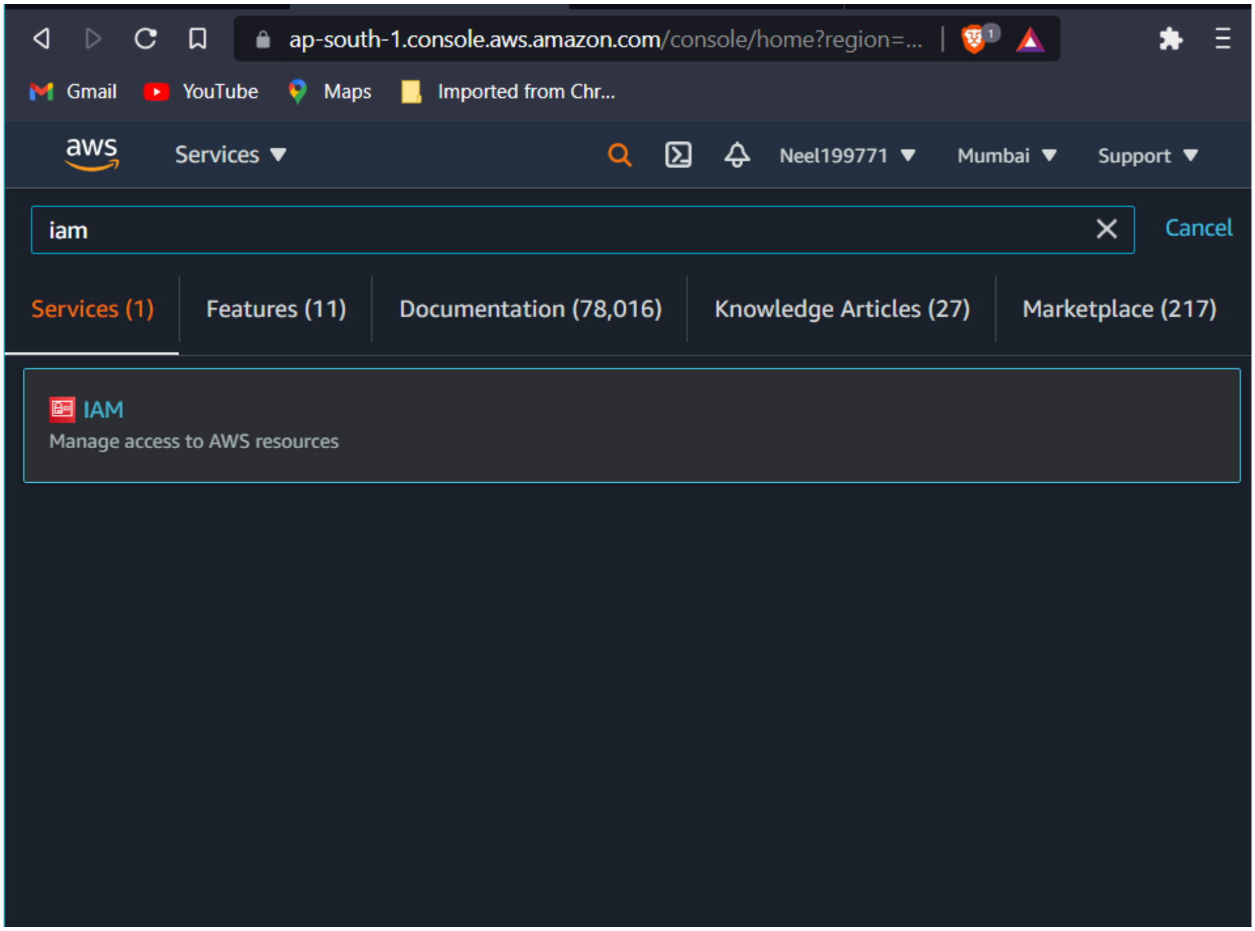
Aim :-

To understand terraform lifecycle and to build, change, and destroy AWS infrastructure using Terraform

Steps :-

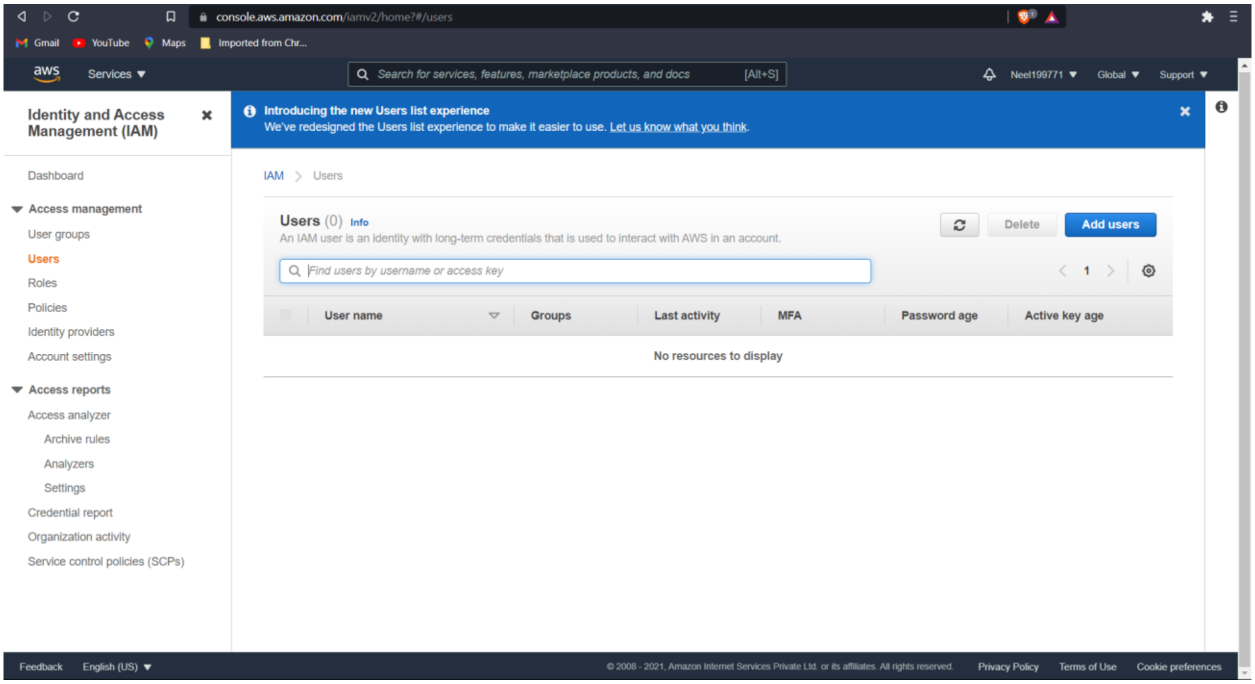
Terraform is an infrastructure as code (IaC) tool that allows you to build, change, and version infrastructure safely and efficiently.

1) Open And Login to your AWS console-And search IAM and click on it



ap-south-1.console.aws.amazon.com/console/home?region=...

2) Now click on Add Users in The User Section as shown in the image


console.aws.amazon.com/iamv2/home?#/users
Gmail
YouTube
Maps
Imported from Chr...
aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
Neel199771 ▼
Global ▼
Support ▼

Identity and Access Management (IAM)

Introducing the new Users list experience
We've redesigned the Users list experience to make it easier to use. Let us know what you think.

Users (0) Info

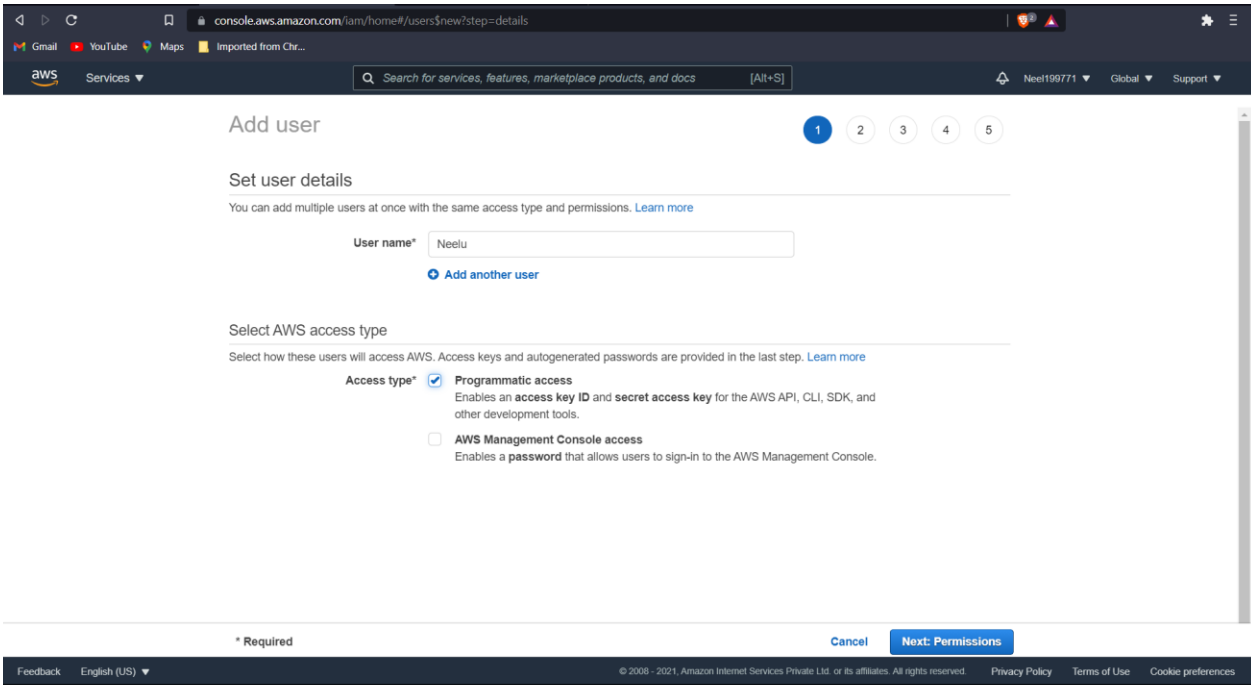
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

User name	Groups	Last activity	MFA	Password age	Active key age

No resources to display

Feedback
English (US) ▼
© 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences

3) Now give any name For username And Check The Programmatic Access field shown below


console.aws.amazon.com/iam/home#/users$new?step=details
Gmail
YouTube
Maps
Imported from Chr...
aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
Neel199771 ▼
Global ▼
Support ▼
1
2
3
4
5

Add user

Set user details

You can add multiple users at once with the same access type and permissions. Learn more

User name* Neelu

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. Learn more

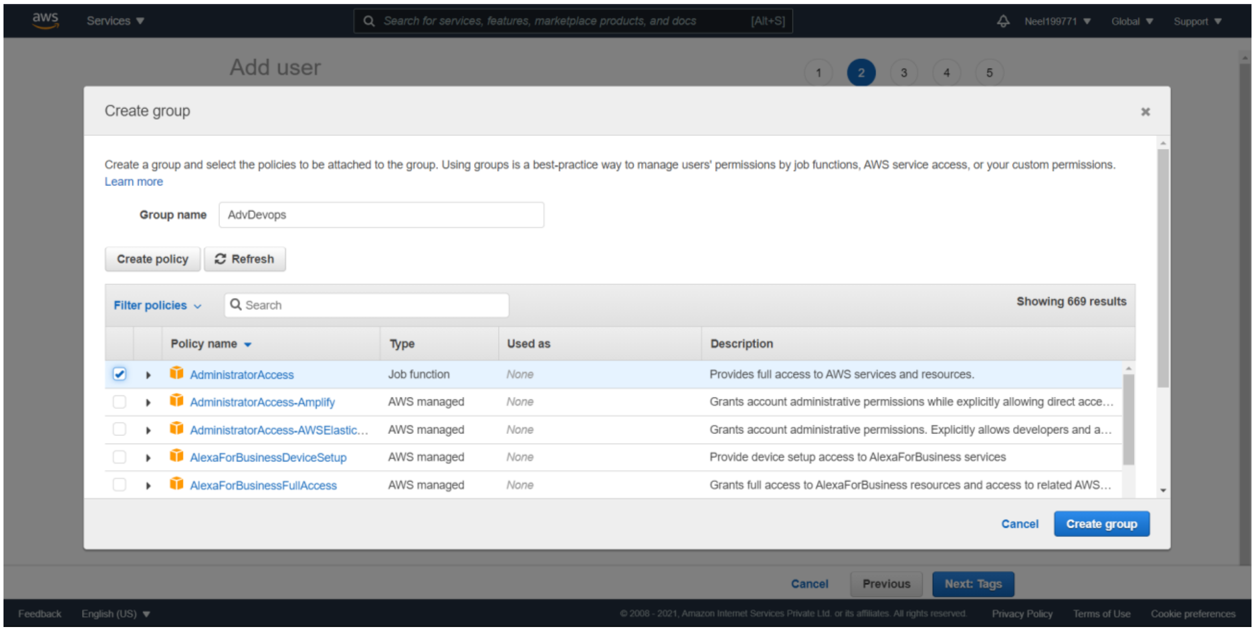
Access type*
☑ Programmatic access
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

□ AWS Management Console access
Enables a **password** that allows users to sign-in to the AWS Management Console.

* Required

Cancel
Next: Permissions
Feedback
English (US) ▼
© 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences

4) Add Group name and Check the first Policy Name


aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
Neel199771 ▼
Global ▼
Support ▼
Add user
1
2
3
4
5

Create group

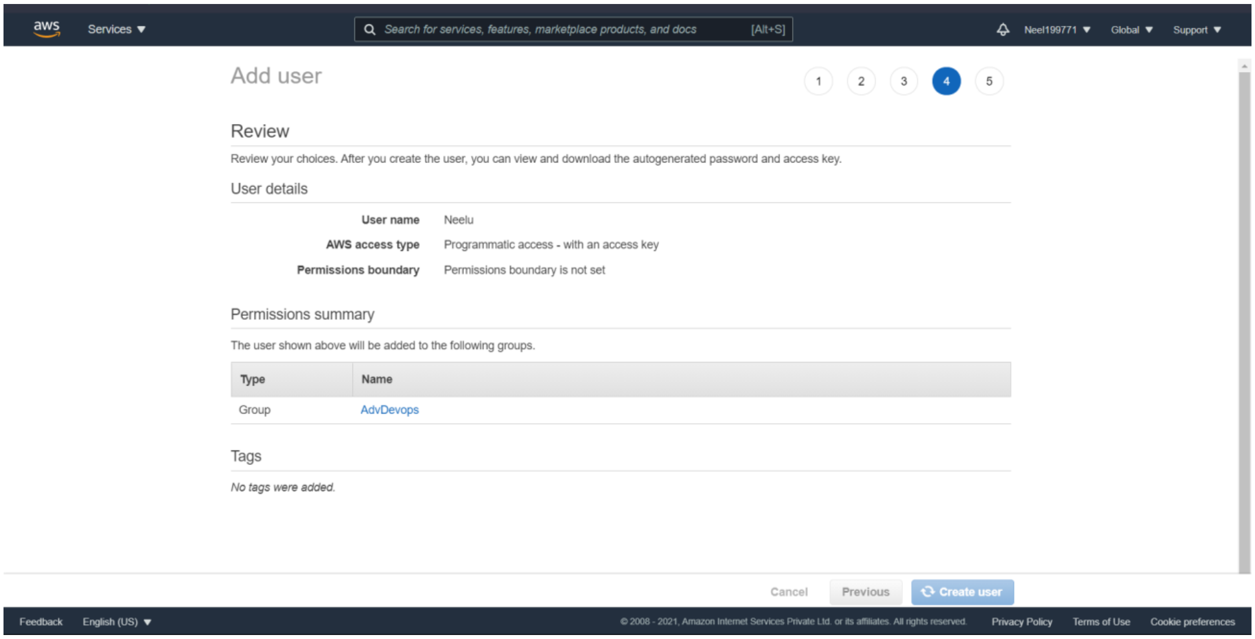
Create a group and select the policies to be attached to the group. Using groups is a best-practice way to manage users' permissions by job functions, AWS service access, or your custom permissions. Learn more

Group name AdvDevops

Showing 669 results

		Policy name	Type	Used as	Description
☑	▸	AdministratorAccess	Job function	None	Provides full access to AWS services and resources.
□	▸	AdministratorAccess-Amplify	AWS managed	None	Grants account administrative permissions while explicitly allowing direct acce...
□	▸	AdministratorAccess-AWSElastic...	AWS managed	None	Grants account administrative permissions. Explicitly allows developers and a...
□	▸	AlexaForBusinessDeviceSetup	AWS managed	None	Provide device setup access to AlexaForBusiness services
□	▸	AlexaForBusinessFullAccess	AWS managed	None	Grants full access to AlexaForBusiness resources and access to related AWS...

5) Don’t add tags


aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
Neel199771 ▼
Global ▼
Support ▼
1
2
3
4
5

Add user

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name	Neelu
AWS access type	Programmatic access - with an access key
Permissions boundary	Permissions boundary is not set

Permissions summary

The user shown above will be added to the following groups.

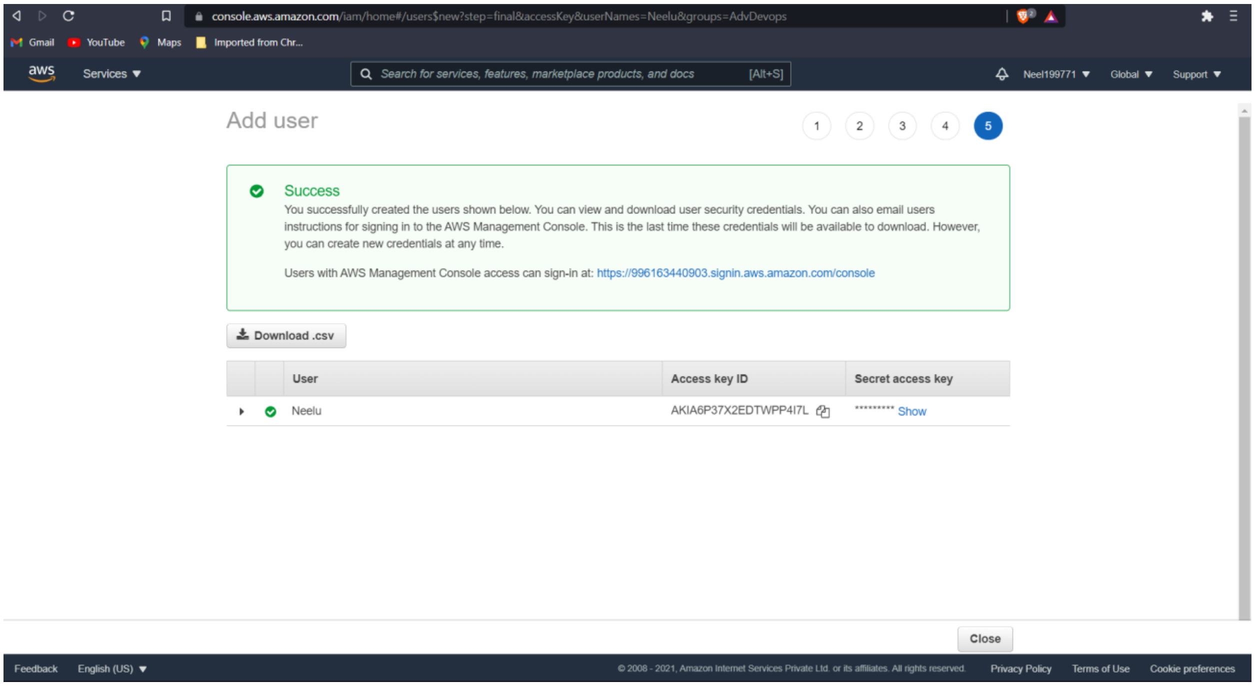
Type	Name
Group	AdvDevops

Tags

No tags were added.

Cancel
Previous
Create user
Feedback
English (US) ▼
© 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences

6) Now Download .csv file



console.aws.amazon.com/iam/home#/users$new?step=final&accessKey&userNames=Neelu&groups=AdvDevops
Gmail
YouTube
Maps
Imported from Chr...
aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
Neel199771 ▼
Global ▼
Support ▼
1
2
3
4
5
Download .csv
Close
Feedback
English (US) ▼
© 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences

Add user

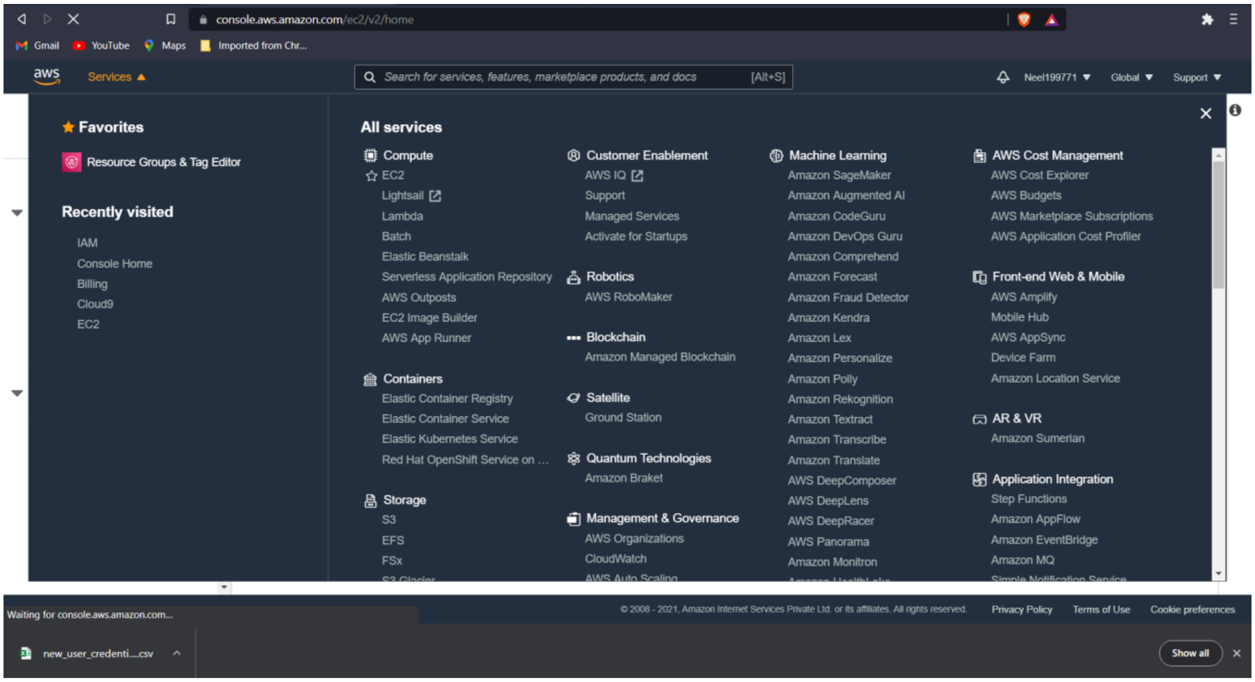
Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

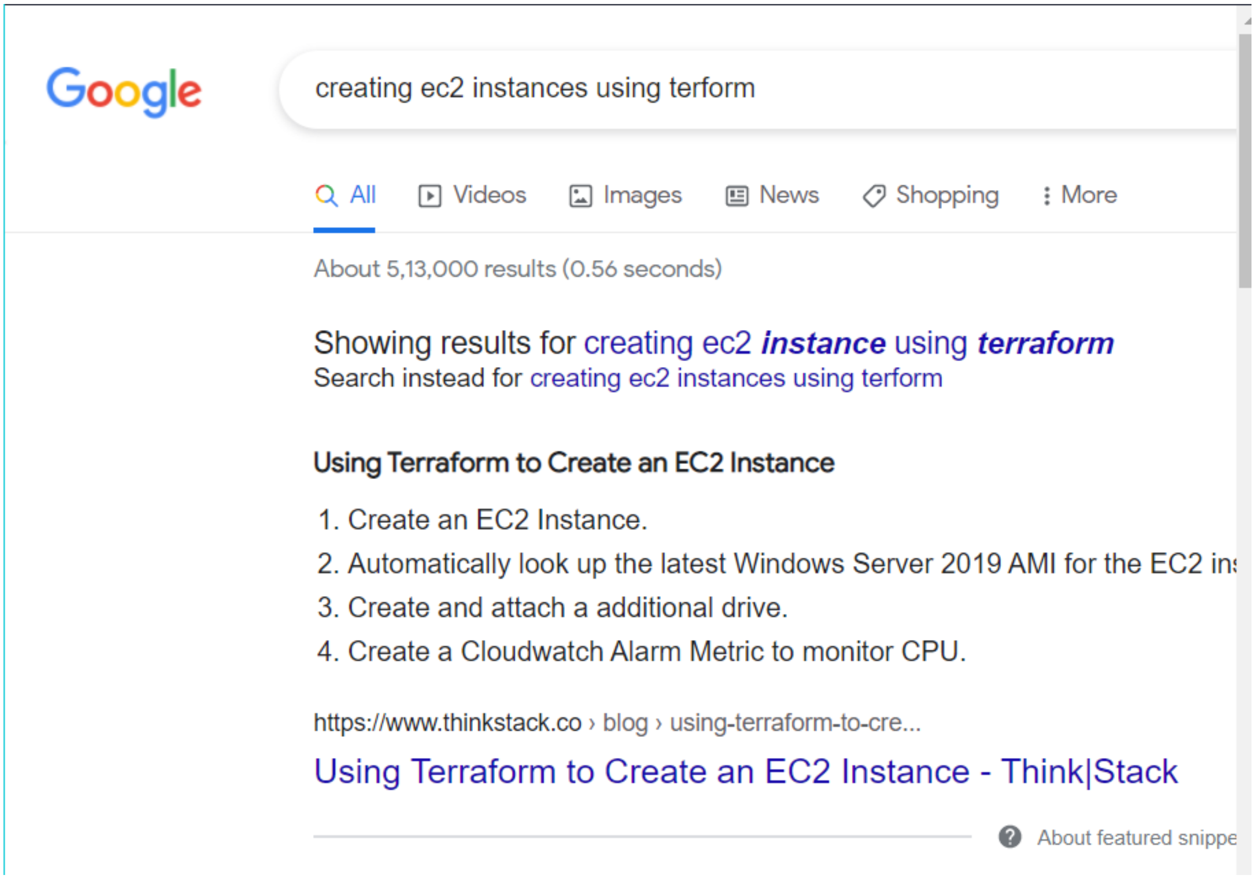
Users with AWS Management Console access can sign-in at: https://996163440903.signin.aws.amazon.com/console

		User	Access key ID	Secret access key
▸	✓	Neelu	AKIA6P37X2EDTWPP4I7L	********* Show

7) Go to services and Ec2



8) Again google search the following terms



About 5,13,000 results (0.56 seconds)

Showing results for creating ec2 ***instance*** using ***terraform***
Search instead for creating ec2 instances using terform

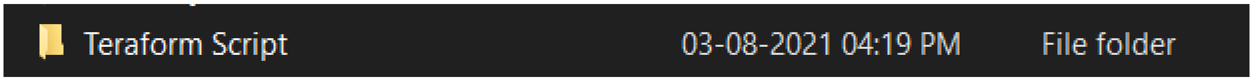
Using Terraform to Create an EC2 Instance

1. Create an EC2 Instance.
2. Automatically look up the latest Windows Server 2019 AMI for the EC2 ins
3. Create and attach a additional drive.
4. Create a Cloudwatch Alarm Metric to monitor CPU.

https://www.thinkstack.co › blog › using-terraform-to-cre...

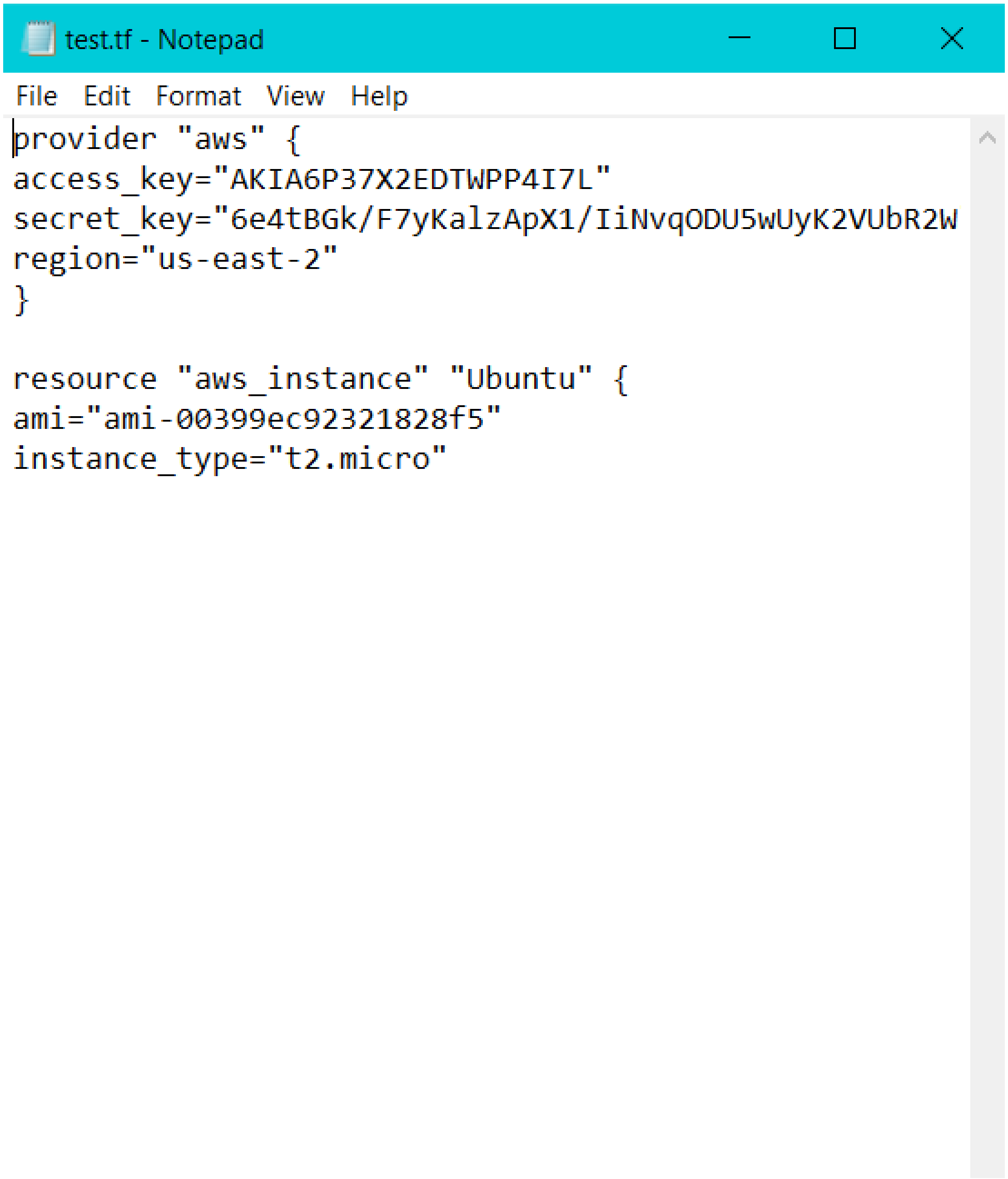
Using Terraform to Create an EC2 Instance - Think|Stack

Created a folder Name Terraform Scripts in the C drive where the AdvDevops folder was created



Teraform Script 03-08-2021 04:19 PM File folder

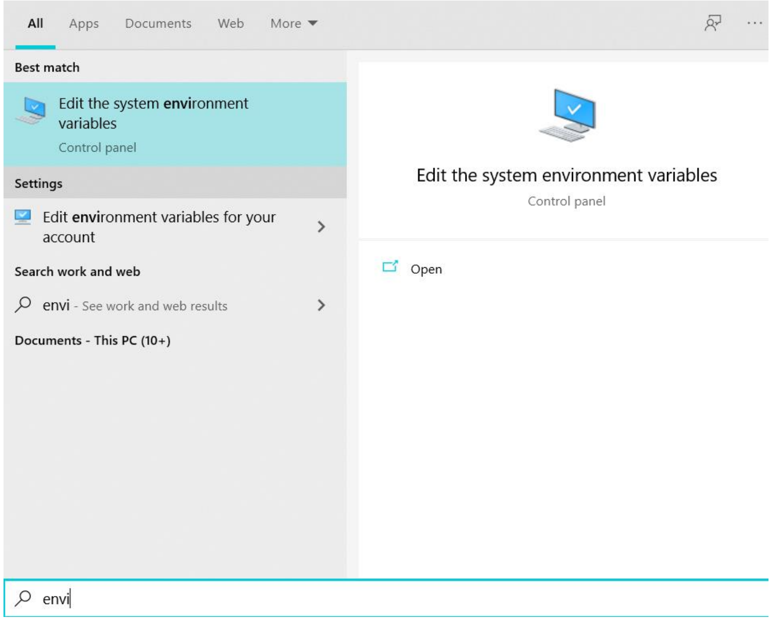
9) Now Go to note pad And Type the below Details properly But before It Just Change the ACCESS KEY AND SECRECT KEY TO THE ONE IN YOUR .csv File . Set region same as below if you want MUMBAI as your region.



```
provider "aws" {
access_key="AKIA6P37X2EDTWPP4I7L"
secret_key="6e4tBGk/F7yKalzApX1/IiNvqODU5wUyK2VUbR2W
region="us-east-2"
}

resource "aws_instance" "Ubuntu" {
ami="ami-00399ec92321828f5"
instance_type="t2.micro"
```

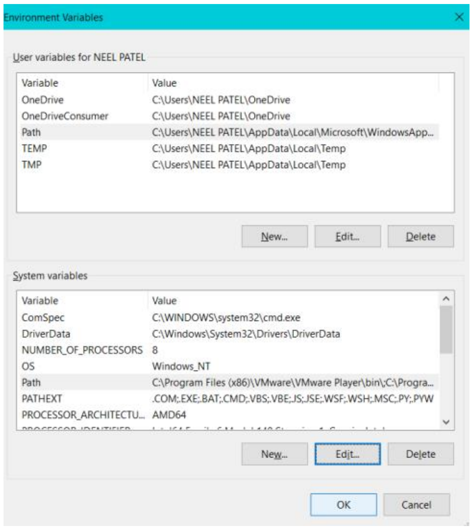
10) Now search EDIT THE SYSTEM ENVIRONMENT VARIABLES in your windows search.

Open it


All
Apps
Documents
Web
More
Best match
Edit the system environment variables
Control panel
Settings
Edit environment variables for your account
Search work and web
envi - See work and web results
Documents - This PC (10+)
Edit the system environment variables
Control panel
Open
envi

11) Now click on PATH OF USER VARIBLES, then click on Edit option Now go to edit and then add new path C:\AdvDevOps

Repeat same procedure for system variables.



User variables for NEEL PATEL

Variable	Value
OneDrive	C:\Users\NEEL PATEL\OneDrive
OneDriveConsumer	C:\Users\NEEL PATEL\OneDrive
Path	C:\Users\NEEL PATEL\AppData\Local\Microsoft\WindowsApp...
TEMP	C:\Users\NEEL PATEL\AppData\Local\Temp
TMP	C:\Users\NEEL PATEL\AppData\Local\Temp

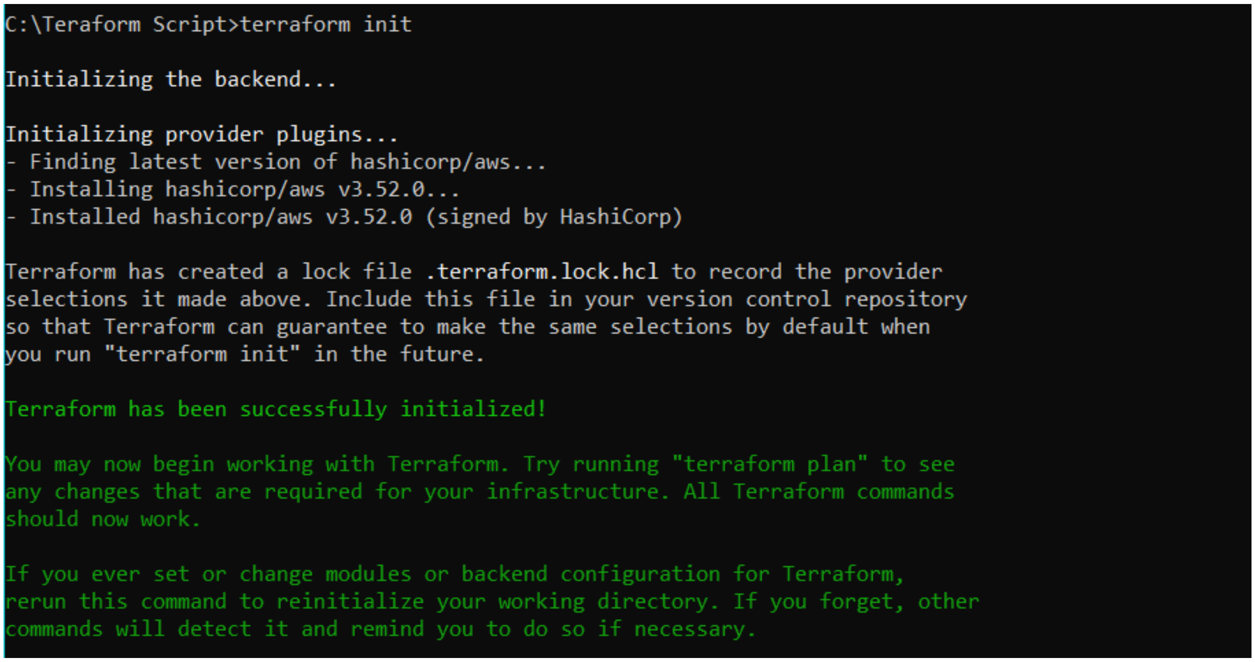
System variables

Variable	Value
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	8
OS	Windows_NT
Path	C:\Program Files (x86)\VMware\VMware Player\bin\;C:\Progra...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC;.PY;.PYW
PROCESSOR_ARCHITECTU...	AMD64

12) Now Open Command Prompt and then pate the path of Terraform script

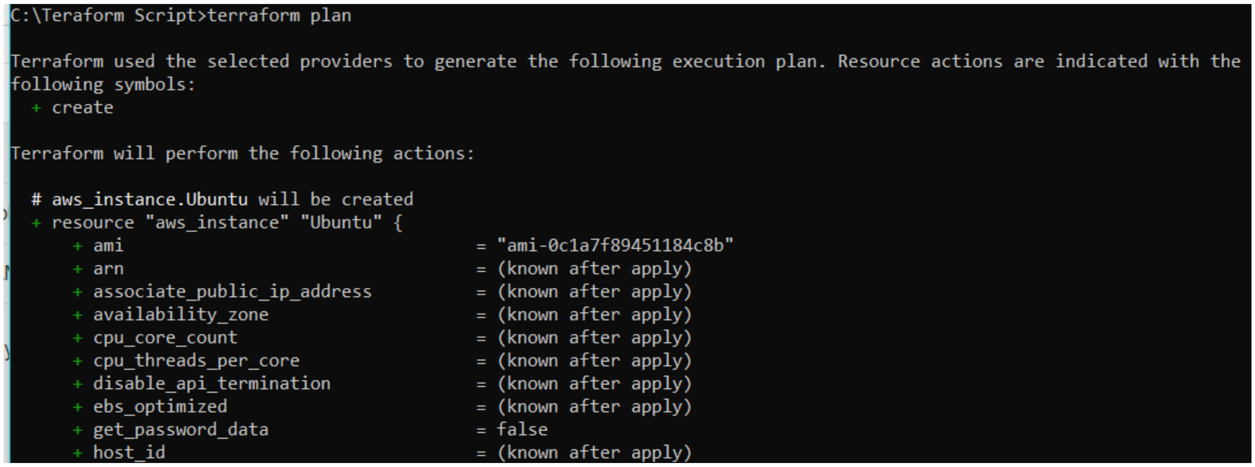
Eg. CD C:\Terraform Script as shown below

Now type Terraform Init command



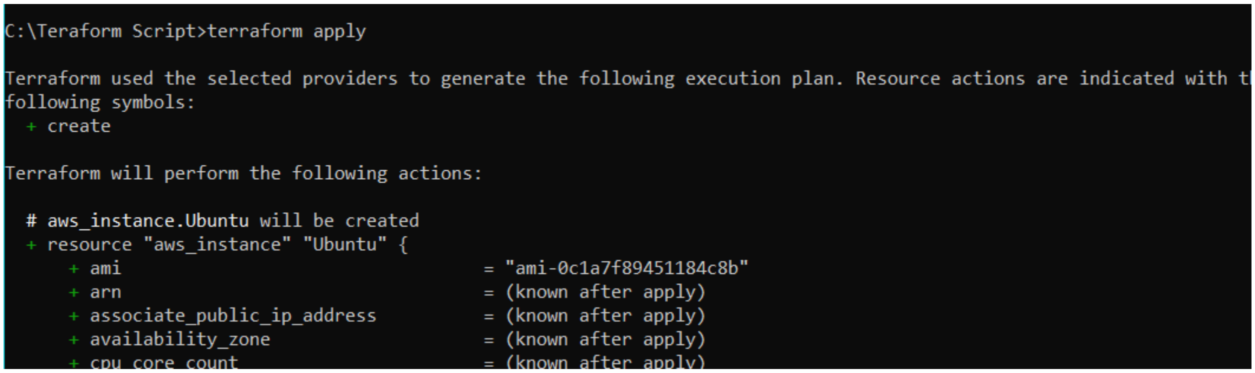
```
C:\Teraform Script>terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v3.52.0...
- Installed hashicorp/aws v3.52.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

Then if there are no errors type Terraform Plan as shown below (type YES when command prompt ask)

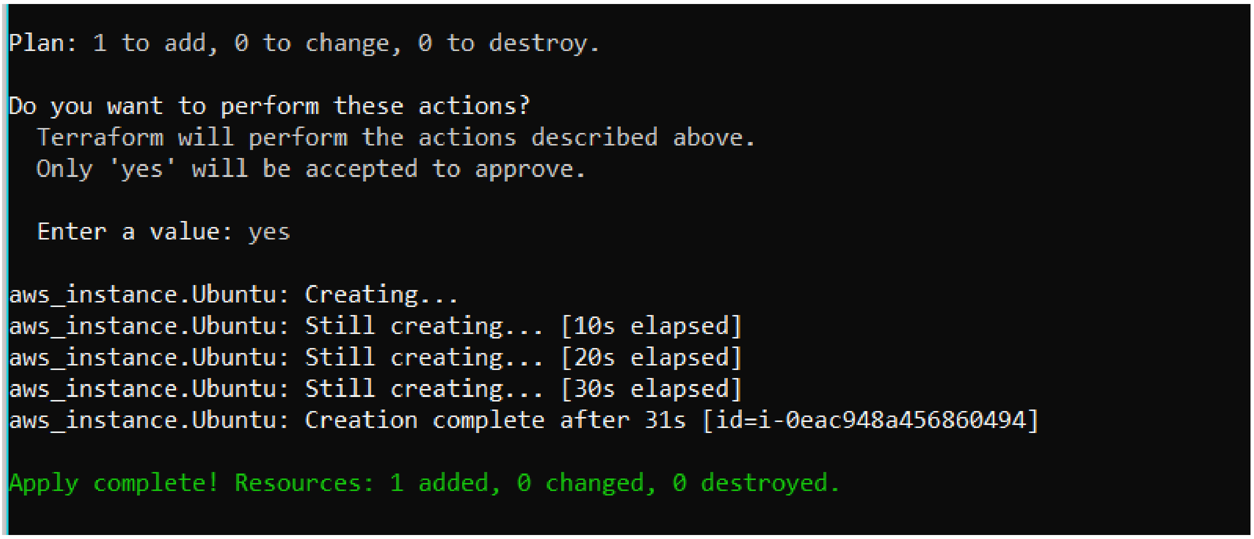


```
C:\Teraform Script>terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the
following symbols:
  + create

Terraform will perform the following actions:

  # aws_instance.Ubuntu will be created
  + resource "aws_instance" "Ubuntu" {
      + ami                                  = "ami-0c1a7f89451184c8b"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
```

Now Finally Type Terraform Apply



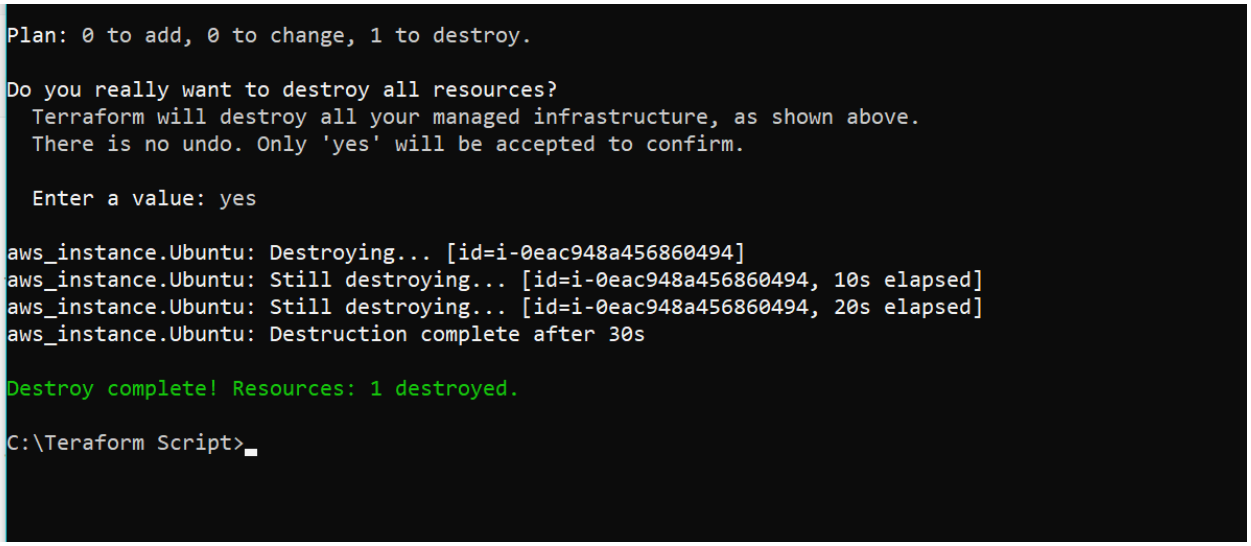
```
C:\Teraform Script>terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with t
following symbols:
  + create

Terraform will perform the following actions:

  # aws_instance.Ubuntu will be created
  + resource "aws_instance" "Ubuntu" {
      + ami                                  = "ami-0c1a7f89451184c8b"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
```



```
Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_instance.Ubuntu: Creating...
aws_instance.Ubuntu: Still creating... [10s elapsed]
aws_instance.Ubuntu: Still creating... [20s elapsed]
aws_instance.Ubuntu: Still creating... [30s elapsed]
aws_instance.Ubuntu: Creation complete after 31s [id=i-0eac948a456860494]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

13) Now go to EC2 and check that is an instance created by the name of UBUNTU and is it in running status or not If it is in Running Status then Come back to Command prompt And Terminate the Instance by

Typing - Terraform destroy



```
Plan: 0 to add, 0 to change, 1 to destroy.

Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

aws_instance.Ubuntu: Destroying... [id=i-0eac948a456860494]
aws_instance.Ubuntu: Still destroying... [id=i-0eac948a456860494, 10s elapsed]
aws_instance.Ubuntu: Still destroying... [id=i-0eac948a456860494, 20s elapsed]
aws_instance.Ubuntu: Destruction complete after 30s

Destroy complete! Resources: 1 destroyed.

C:\Teraform Script>_
```

14) Now go back to EC2 if the instance is terminated, if yes then logout of the Aws Console.

And close the command prompt!

Conclusion :-

Terraform is a powerful Infrastructure as Code (IaC) tool that automates the provisioning, management, and destruction of AWS infrastructure. It can help you to save time, reduce errors, and improve the consistency of your infrastructure.