## EXPERIMENT NO : 06

AIM : To Build, change and destroy AWS/GCP/Microsoft Azure/DigitalOcean infrastructure using Terraform

THEORY :

Terraform is an open-source infrastructure as code software tools created by HashiCorp. Users define and provide data center infrastructure using a declarative configuration language known as HashiCorp Configuration Language (HCL) or optionally JSON.

Terraform supports a number of cloud infrastructure providers such Amazon Web Services, Microsoft Azure, IBM Cloud, Google Cloud Platform, DigitalOcean, Oracle Cloud Infrastructure, Yandex.Cloud, vMware vSphere and OpenStack.

Terraform has four major commands

```
$ terraform init
```

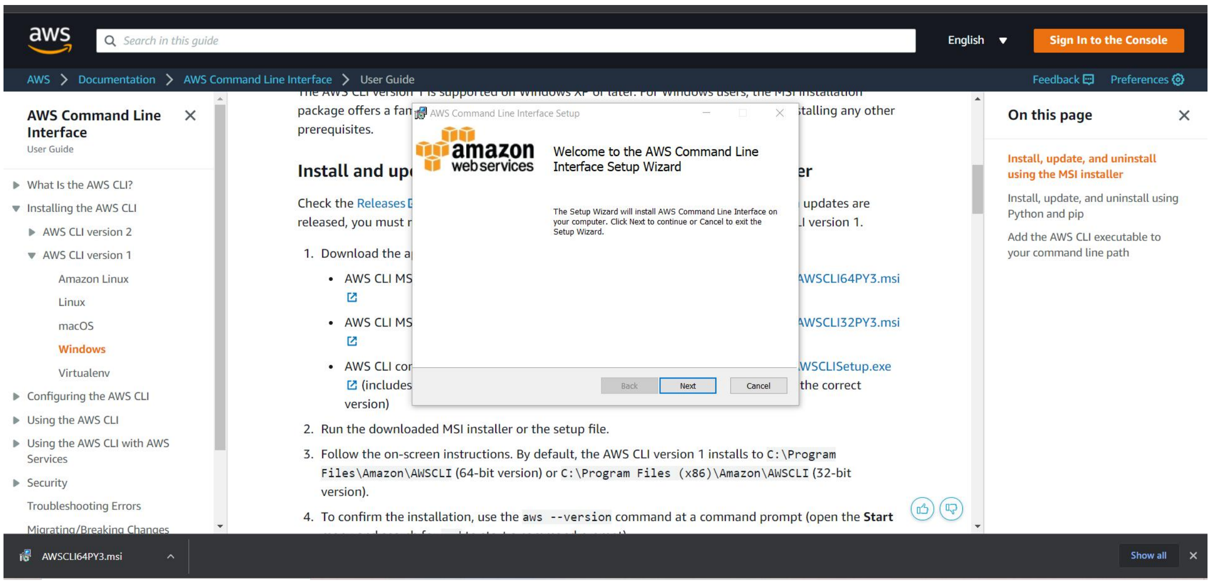
```
$ terraform plan
```

```
$ terraform apply
```

```
$ terraform destroy
```

# DESTROY AWS INFRASTRUCTURE USING TERRAFORM

## Step 1: Download AWS Cli and set environment variable


aws
Search in this guide
English
Sign In to the Console
AWS > Documentation > AWS Command Line Interface > User Guide
Feedback
Preferences
AWS Command Line Interface
User Guide
What Is the AWS CLI?
Installing the AWS CLI
AWS CLI version 2
AWS CLI version 1
Amazon Linux
Linux
macOS
Windows
Virtualenv
Configuring the AWS CLI
Using the AWS CLI
Using the AWS CLI with AWS Services
Security
Troubleshooting Errors
Migrating/Breaking Changes
On this page
Install, update, and uninstall using the MSI installer
Install, update, and uninstall using Python and pip
Add the AWS CLI executable to your command line path
AWS Command Line Interface Setup
amazon webservices
Back
Next
Cancel
AWSCLI64PY3.msi
Show all

package offers a fan… …talling any other prerequisites.

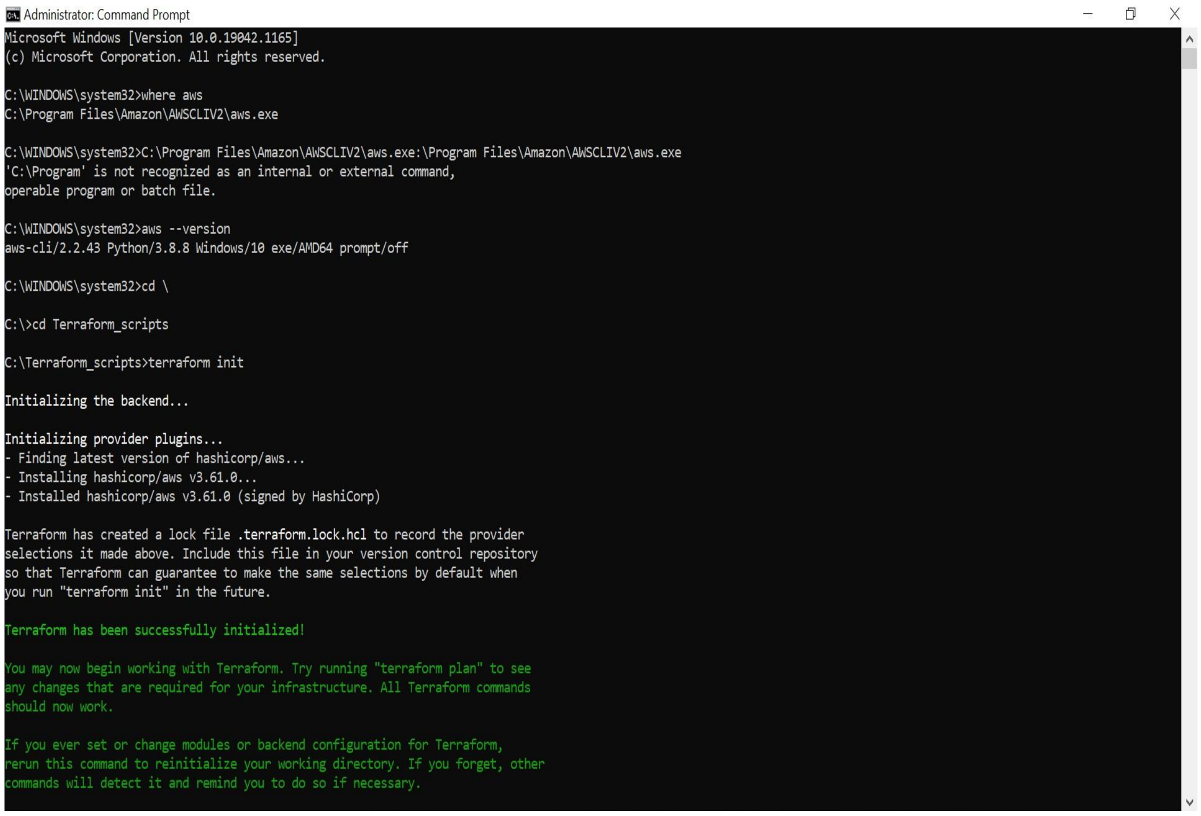
**Install and up… …er**

Check the Releases … updates are released, you must r… …I version 1.

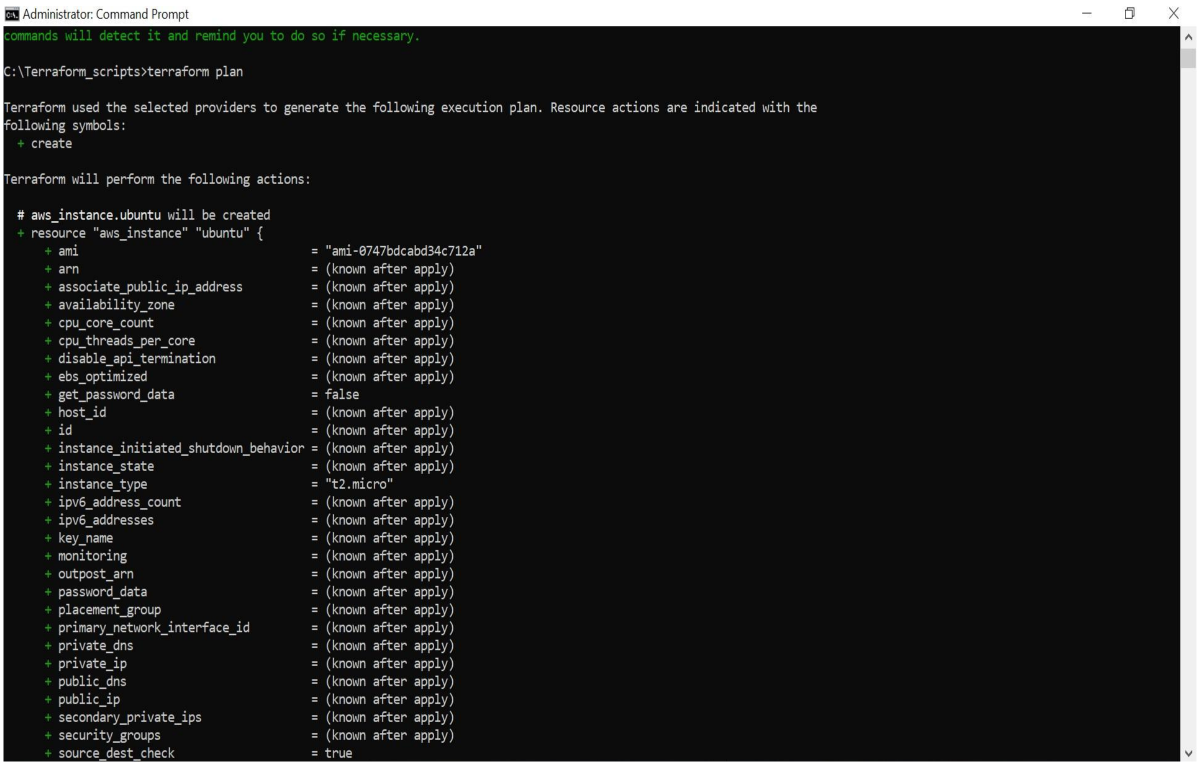
1. Download the a…
   - AWS CLI MS… AWSCLI64PY3.msi
   - AWS CLI MS… AWSCLI32PY3.msi
   - AWS CLI cor… WSCLISetup.exe (includes… the correct version)
2. Run the downloaded MSI installer or the setup file.
3. Follow the on-screen instructions. By default, the AWS CLI version 1 installs to `C:\Program Files\Amazon\AWSCLI` (64-bit version) or `C:\Program Files (x86)\Amazon\AWSCLI` (32-bit version).
4. To confirm the installation, use the `aws --version` command at a command prompt (open the **Start**

Welcome to the AWS Command Line Interface Setup Wizard

The Setup Wizard will install AWS Command Line Interface on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

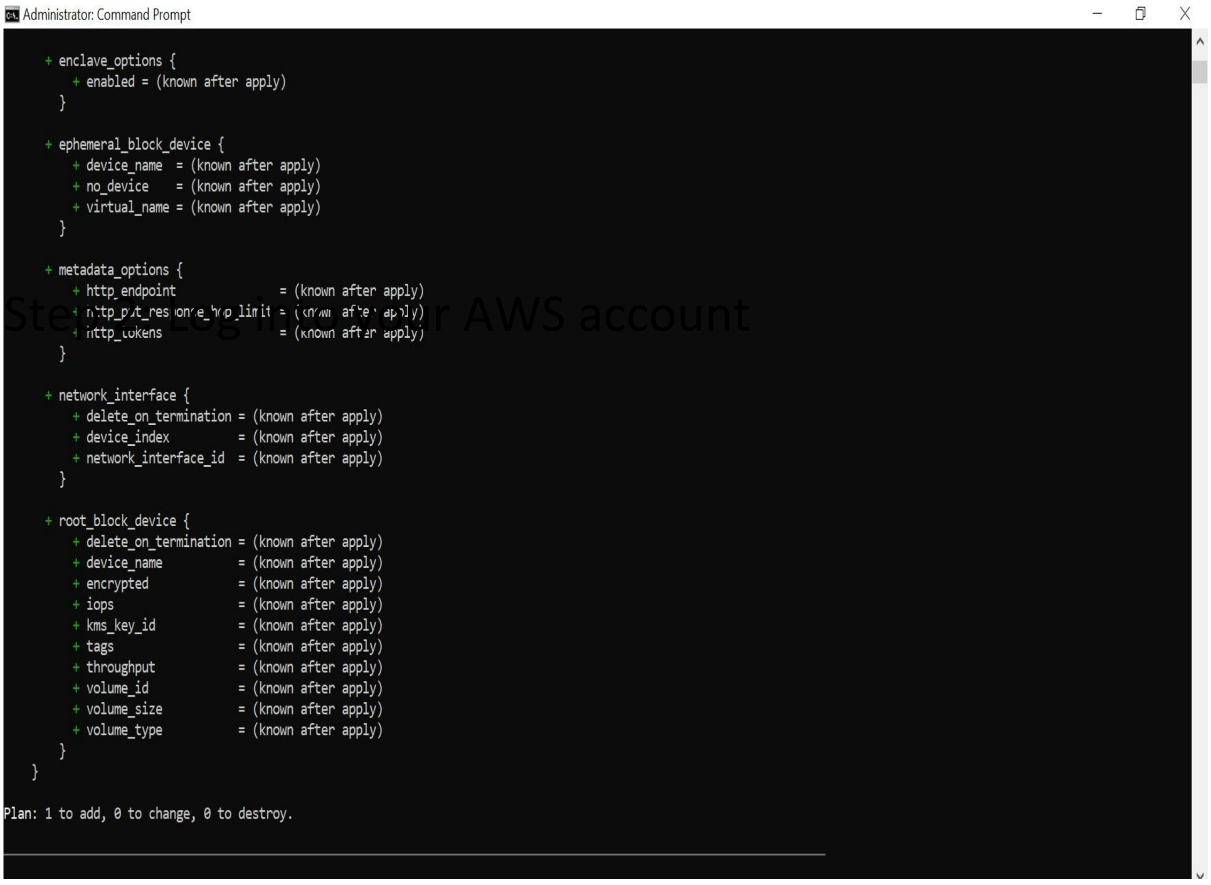

Administrator: Command Prompt

```
Microsoft Windows [Version 10.0.19042.1165]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>where aws
C:\Program Files\Amazon\AWSCLIV2\aws.exe

C:\WINDOWS\system32>C:\Program Files\Amazon\AWSCLIV2\aws.exe:\Program Files\Amazon\AWSCLIV2\aws.exe
'C:\Program' is not recognized as an internal or external command,
operable program or batch file.

C:\WINDOWS\system32>aws --version
aws-cli/2.2.43 Python/3.8.8 Windows/10 exe/AMD64 prompt/off

C:\WINDOWS\system32>cd \

C:\>cd Terraform_scripts

C:\Terraform_scripts>terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v3.61.0...
- Installed hashicorp/aws v3.61.0 (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.
```



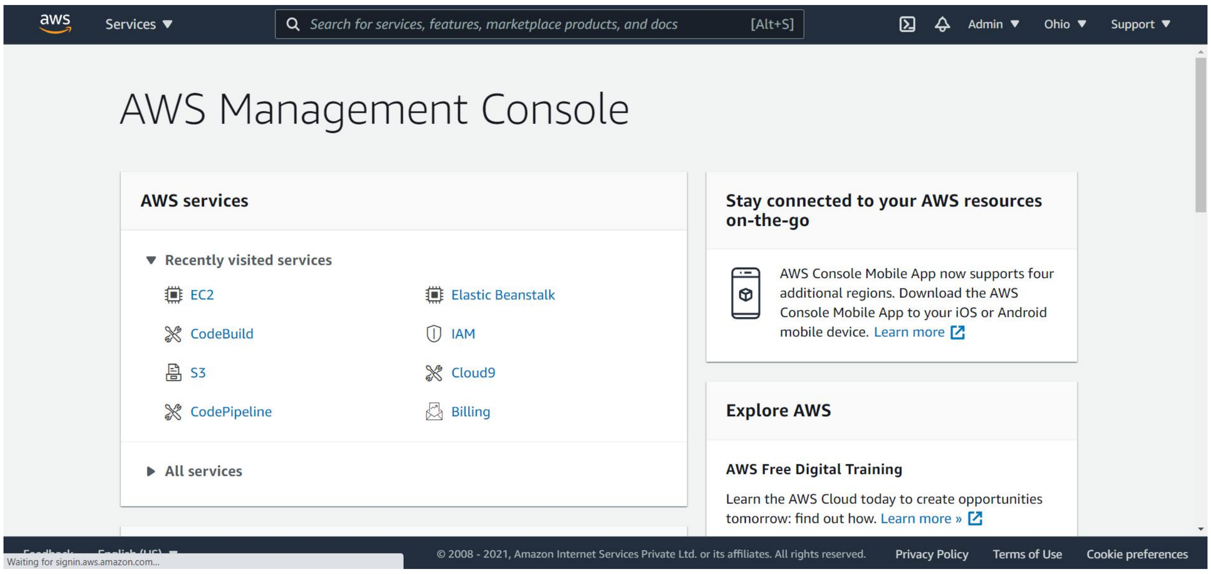
```
commands will detect it and remind you to do so if necessary.

C:\Terraform_scripts>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-0747bdcabd34c712a"
      + 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)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = (known after apply)
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
```

Ste   Log in   r AWS account



```
      + enclave_options {
          + enabled = (known after apply)
        }

      + ephemeral_block_device {
          + device_name  = (known after apply)
          + no_device    = (known after apply)
          + virtual_name = (known after apply)
        }

      + metadata_options {
          + http_endpoint               = (known after apply)
          + http_put_response_hop_limit = (known after apply)
          + http_tokens                 = (known after apply)
        }

      + network_interface {
          + delete_on_termination = (known after apply)
          + device_index          = (known after apply)
          + network_interface_id  = (known after apply)
        }

      + root_block_device {
          + delete_on_termination = (known after apply)
          + device_name           = (known after apply)
          + encrypted             = (known after apply)
          + iops                  = (known after apply)
          + kms_key_id            = (known after apply)
          + tags                  = (known after apply)
          + throughput            = (known after apply)
          + volume_id             = (known after apply)
          + volume_size           = (known after apply)
          + volume_type           = (known after apply)
        }
    }

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

## Step 2: Log into your AWS account


aws Services ▼ Search for services, features, marketplace products, and docs [Alt+S] Admin ▼ Ohio ▼ Support ▼

AWS Management Console

**AWS services**

▼ Recently visited services

| | |
|---|---|
| EC2 | Elastic Beanstalk |
| CodeBuild | IAM |
| S3 | Cloud9 |
| CodePipeline | Billing |

▶ All services

**Stay connected to your AWS resources on-the-go**

AWS Console Mobile App now supports four additional regions. Download the AWS Console Mobile App to your iOS or Android mobile device. Learn more

**Explore AWS**

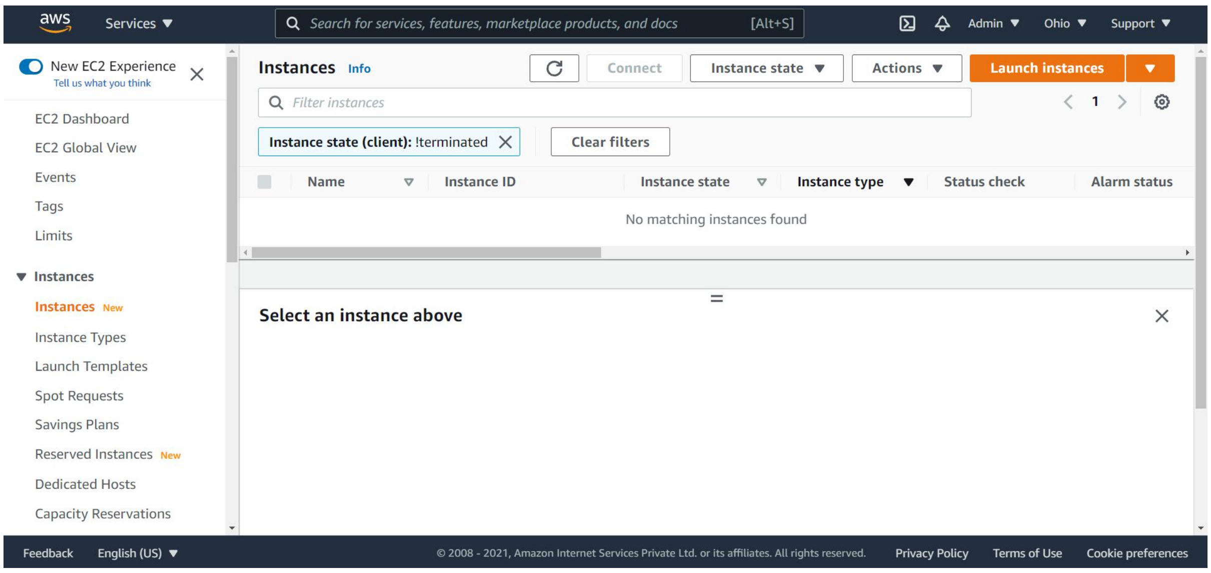
**AWS Free Digital Training**

Learn the AWS Cloud today to create opportunities tomorrow: find out how. Learn more »

Waiting for signin.aws.amazon.com...

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

## Step 3: Open the instance section in EC2 services


aws Services ▼ Search for services, features, marketplace products, and docs [Alt+S] Admin ▼ Ohio ▼ Support ▼

New EC2 Experience
Tell us what you think

EC2 Dashboard
EC2 Global View
Events
Tags
Limits

▼ Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Capacity Reservations

**Instances** Info — Connect | Instance state ▼ | Actions ▼ | Launch instances ▼

Filter instances — 1

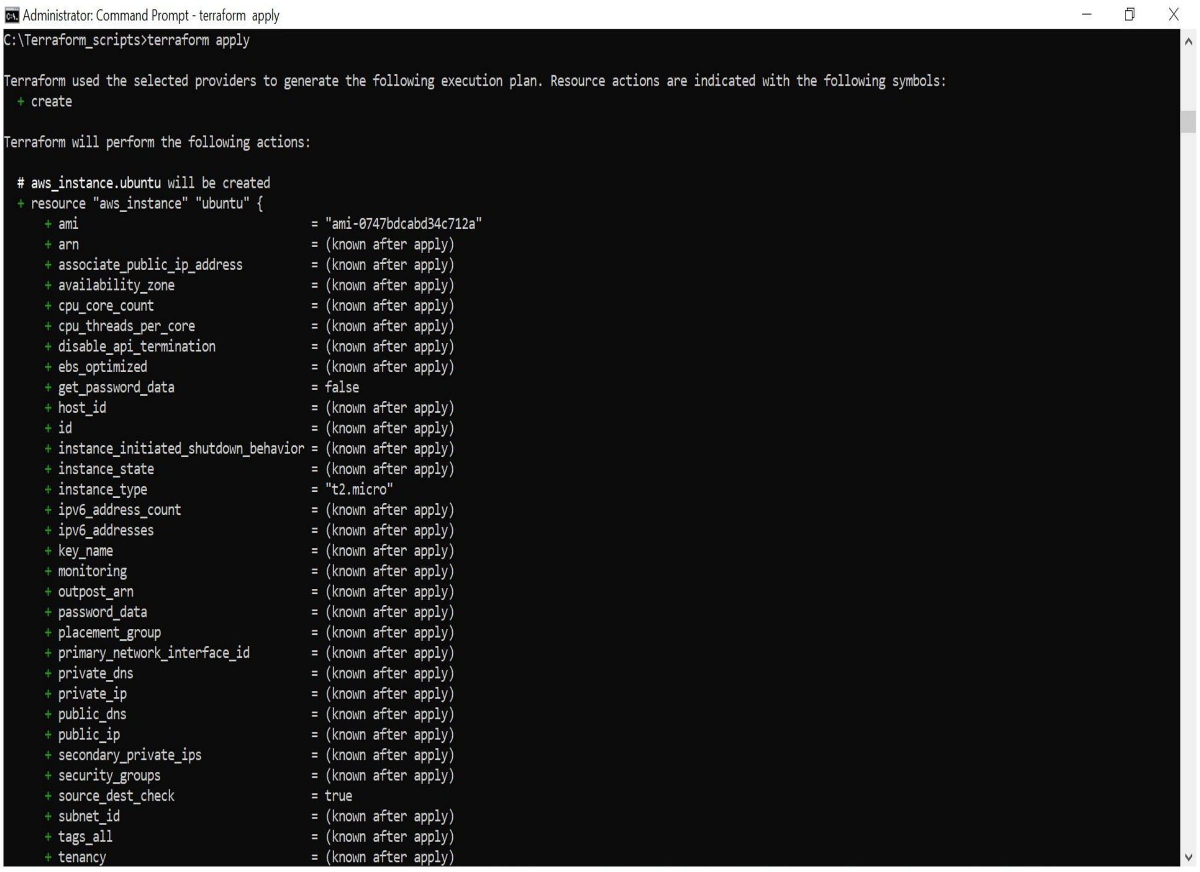
Instance state (client): !terminated ✕ — Clear filters

| Name | Instance ID | Instance state | Instance type | Status check | Alarm status |
|---|---|---|---|---|---|
| No matching instances found | | | | | |

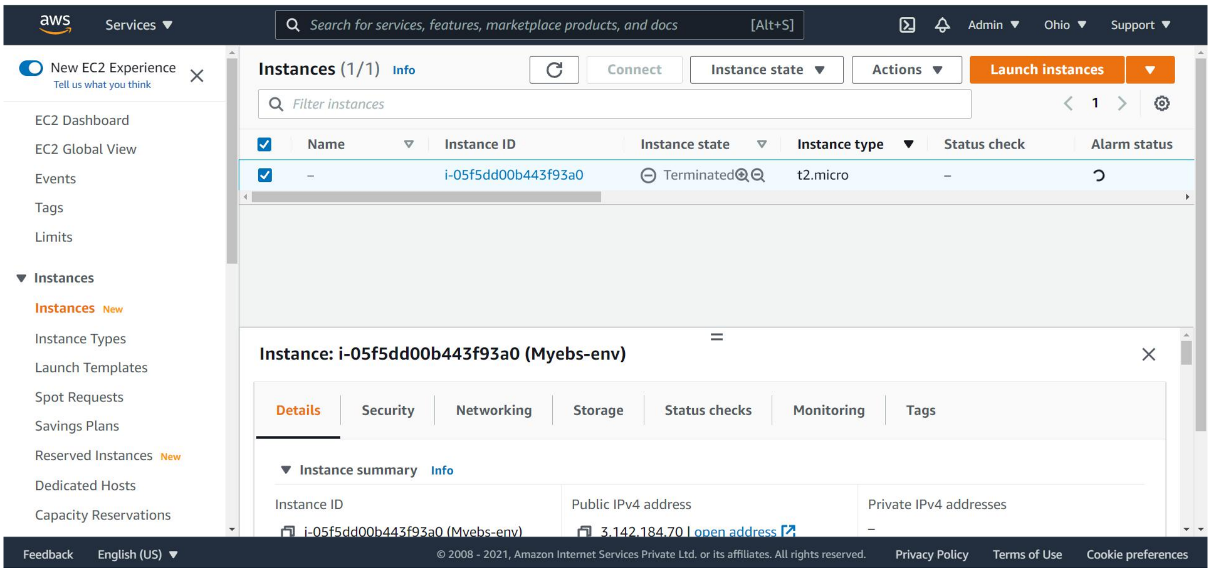
**Select an instance above**

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

## Step 4: Write command terraform apply to create a new instance


Administrator: Command Prompt - terraform apply

```
C:\Terraform_scripts>terraform apply

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-0747bdcabd34c712a"
      + 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)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = (known after apply)
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
      + subnet_id                            = (known after apply)
      + tags_all                             = (known after apply)
      + tenancy                              = (known after apply)
```



Instances (1/1) Info

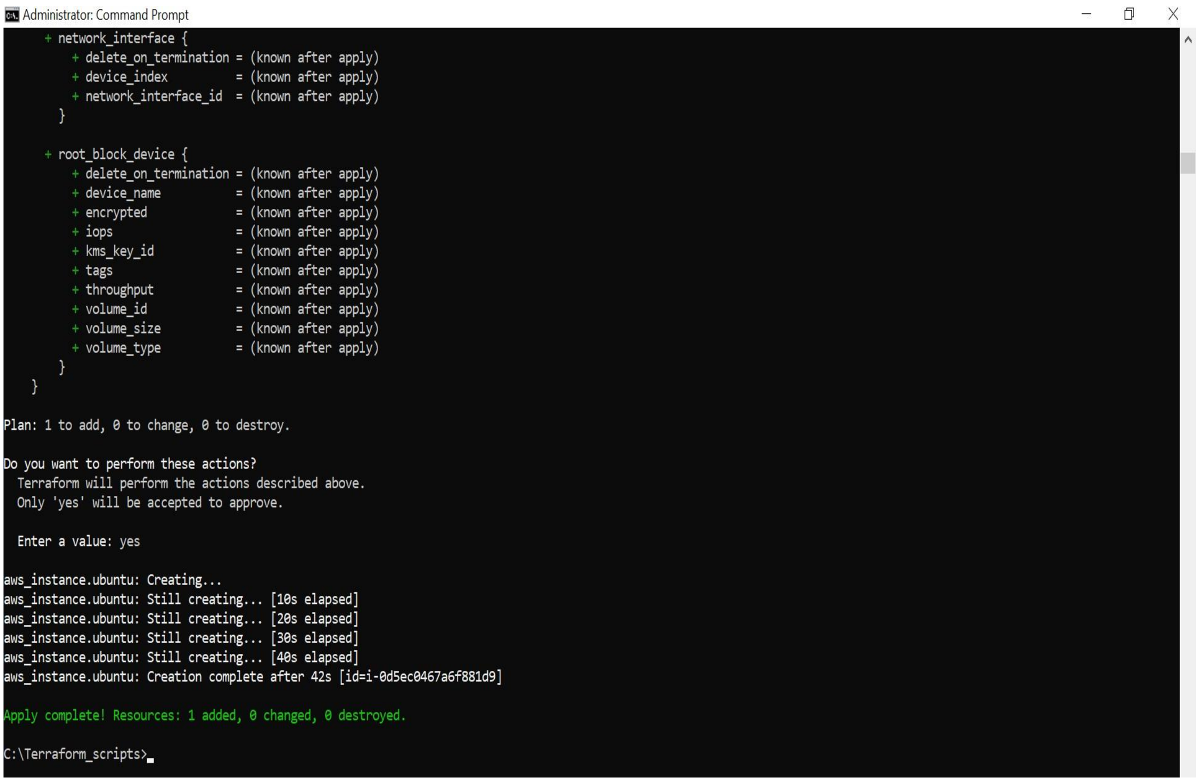
| Name | Instance ID | Instance state | Instance type | Status check | Alarm status |
|---|---|---|---|---|---|
| – | i-05f5dd00b443f93a0 | Terminated | t2.micro | – | |

Instance: i-05f5dd00b443f93a0 (Myebs-env)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

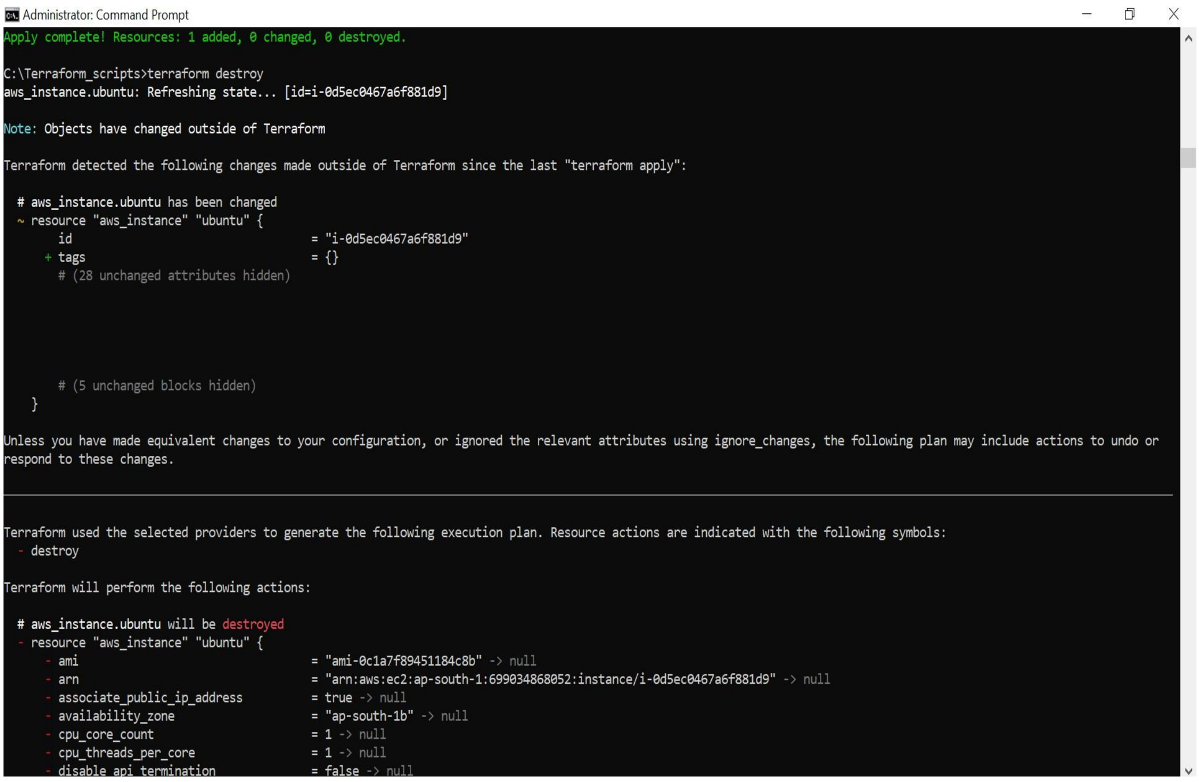
Instance summary Info

| Instance ID | Public IPv4 address | Private IPv4 addresses |
|---|---|---|
| i-05f5dd00b443f93a0 (Myebs-env) | 3.142.184.70 \| open address | – |


Administrator: Command Prompt

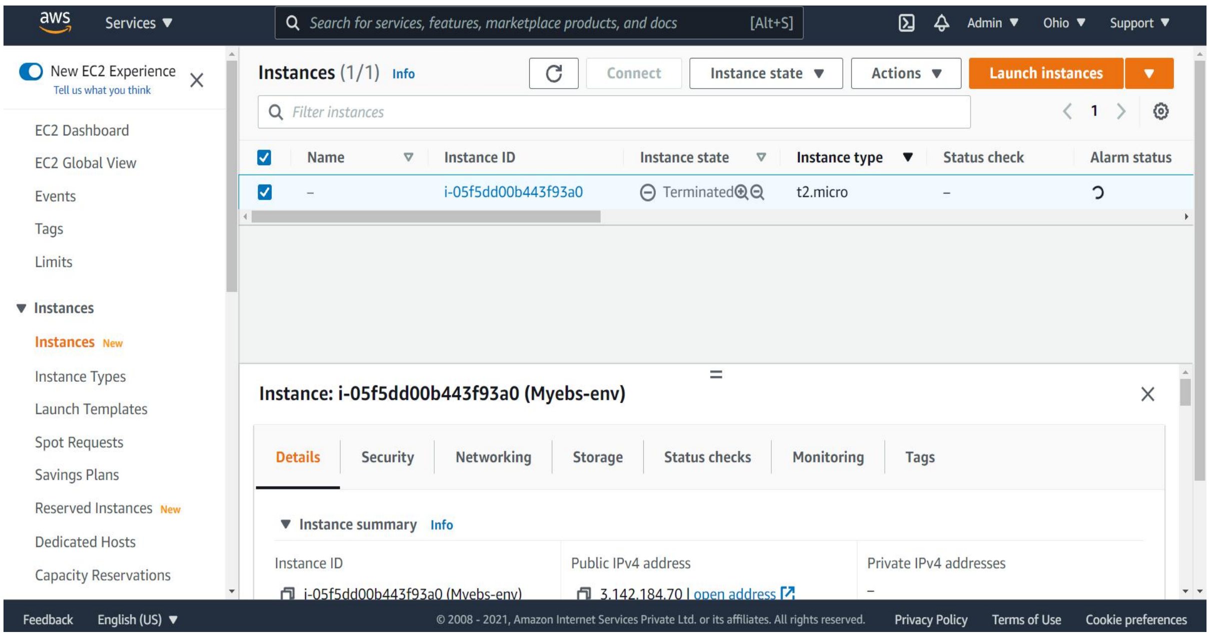
```
      + network_interface {
          + delete_on_termination = (known after apply)
          + device_index          = (known after apply)
          + network_interface_id  = (known after apply)
        }

      + root_block_device {
          + delete_on_termination = (known after apply)
          + device_name           = (known after apply)
          + encrypted             = (known after apply)
          + iops                  = (known after apply)
          + kms_key_id            = (known after apply)
          + tags                  = (known after apply)
          + throughput            = (known after apply)
          + volume_id             = (known after apply)
          + volume_size           = (known after apply)
          + volume_type           = (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: Still creating... [40s elapsed]
aws_instance.ubuntu: Creation complete after 42s [id=i-0d5ec0467a6f881d9]

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

C:\Terraform_scripts>
```

Step 5: Type the command terraform destroy to delete



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

C:\Terraform_scripts>terraform destroy
aws_instance.ubuntu: Refreshing state... [id=i-0d5ec0467a6f881d9]

Note: Objects have changed outside of Terraform

Terraform detected the following changes made outside of Terraform since the last "terraform apply":

  # aws_instance.ubuntu has been changed
  ~ resource "aws_instance" "ubuntu" {
        id                                   = "i-0d5ec0467a6f881d9"
      + tags                                 = {}
        # (28 unchanged attributes hidden)






        # (5 unchanged blocks hidden)
    }

Unless you have made equivalent changes to your configuration, or ignored the relevant attributes using ignore_changes, the following plan may include actions to undo or
respond to these changes.

─────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────

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

Terraform will perform the following actions:

  # aws_instance.ubuntu will be destroyed
  - resource "aws_instance" "ubuntu" {
      - ami                                  = "ami-0c1a7f89451184c8b" -> null
      - arn                                  = "arn:aws:ec2:ap-south-1:699034868052:instance/i-0d5ec0467a6f881d9" -> null
      - associate_public_ip_address          = true -> null
      - availability_zone                    = "ap-south-1b" -> null
      - cpu_core_count                       = 1 -> null
      - cpu_threads_per_core                 = 1 -> null
      - disable_api_termination              = false -> null
```

the instance created



| Name | Instance ID | Instance state | Instance type | Status check | Alarm status |
|---|---|---|---|---|---|
| – | i-05f5dd00b443f93a0 | Terminated | t2.micro | – | |

Instance: i-05f5dd00b443f93a0 (Myebs-env)

▼ Instance summary Info

| Instance ID | Public IPv4 address | Private IPv4 addresses |
|---|---|---|
| i-05f5dd00b443f93a0 (Myebs-env) | 3.142.184.70 \| open address | – |

© 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.

CONCLUSION : Hence we can conclude that we have learned and implemented To Build, change and destroy AWS/GCP/Microsoft Azure/DigitalOcean infrastructure using Terraform.