Don Bosco Institute of Technology, Mumbai 400070

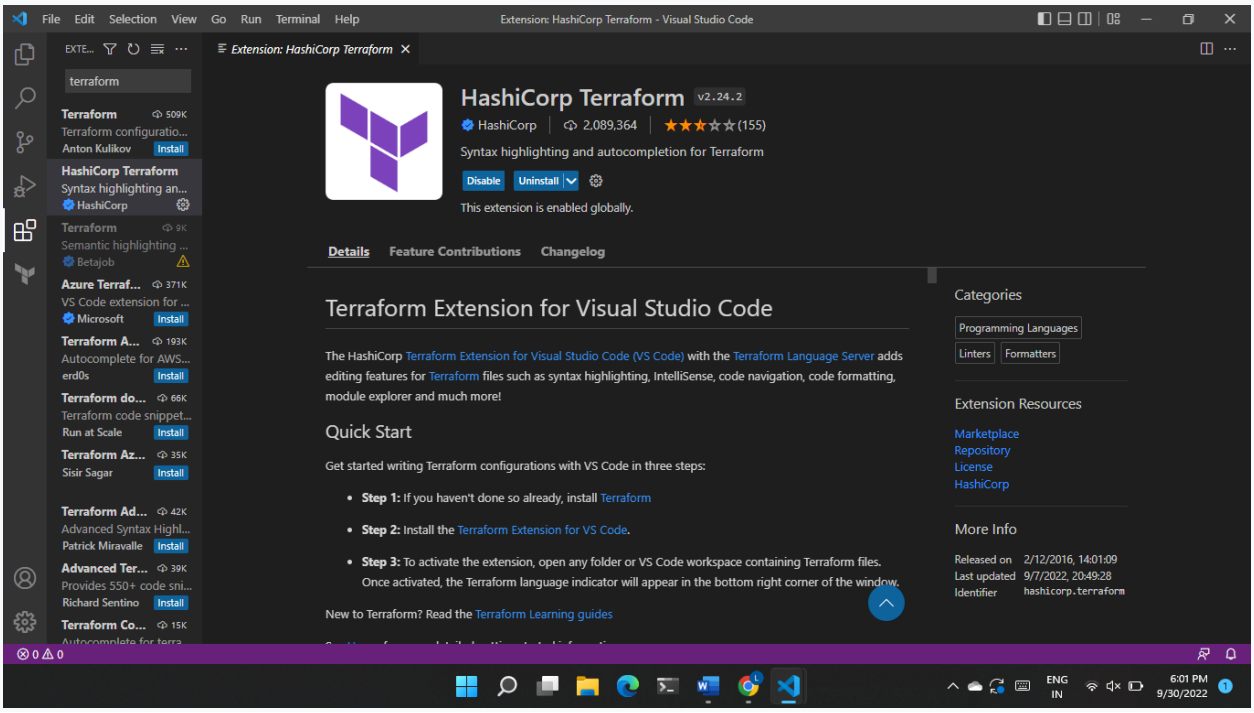
Department of Information Technology

Experiment 6: Terraform

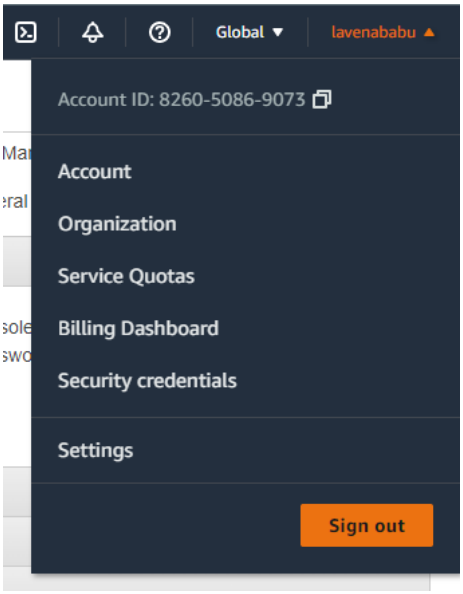
Name: Lavena Babu Roll No.:29

Aim: To Build, change, and destroy AWS / GCP /Microsoft Azure/ DigitalOcean infrastructure Using Terraform

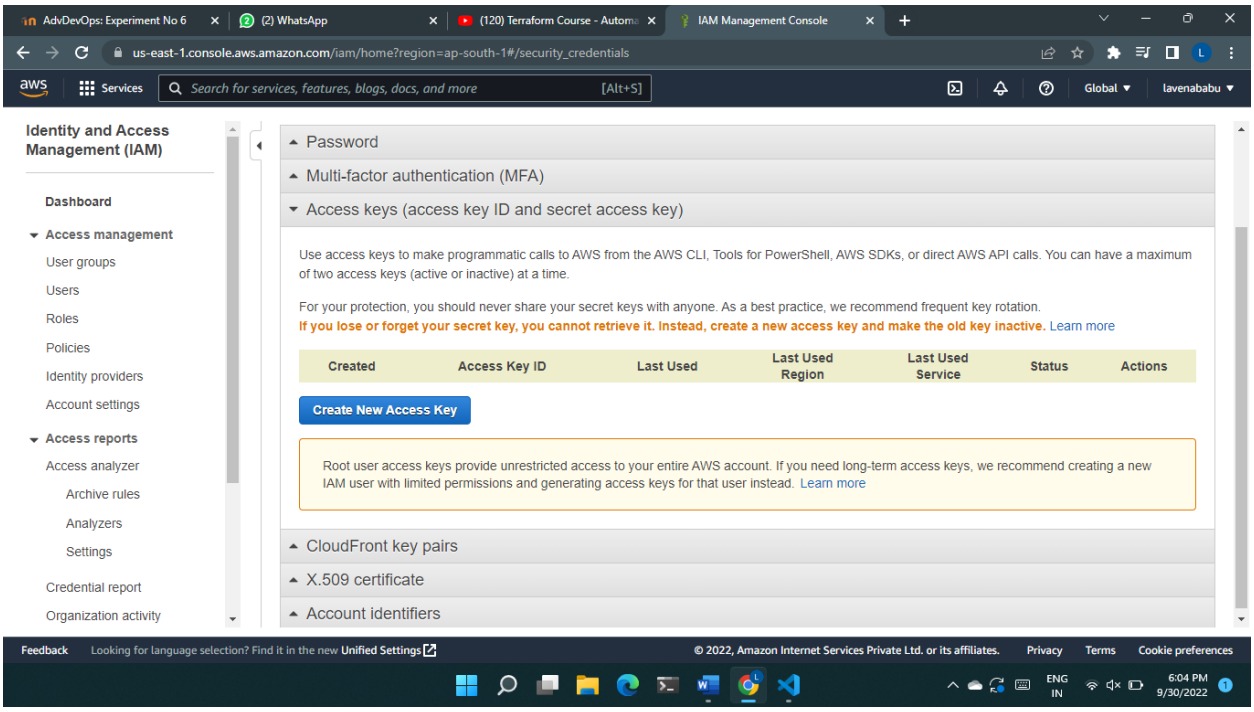
Step 1: Install the terraform extension in VS code



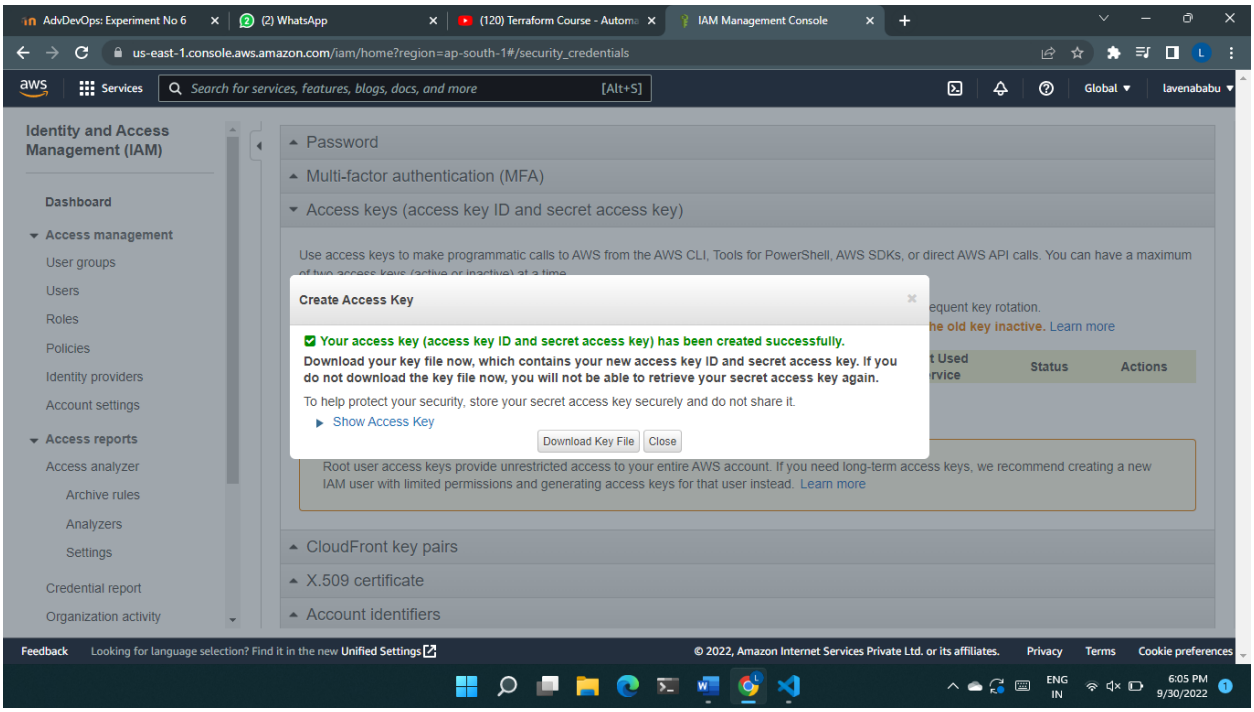
Step 2: Open aws account, go to your account and click on security credentials



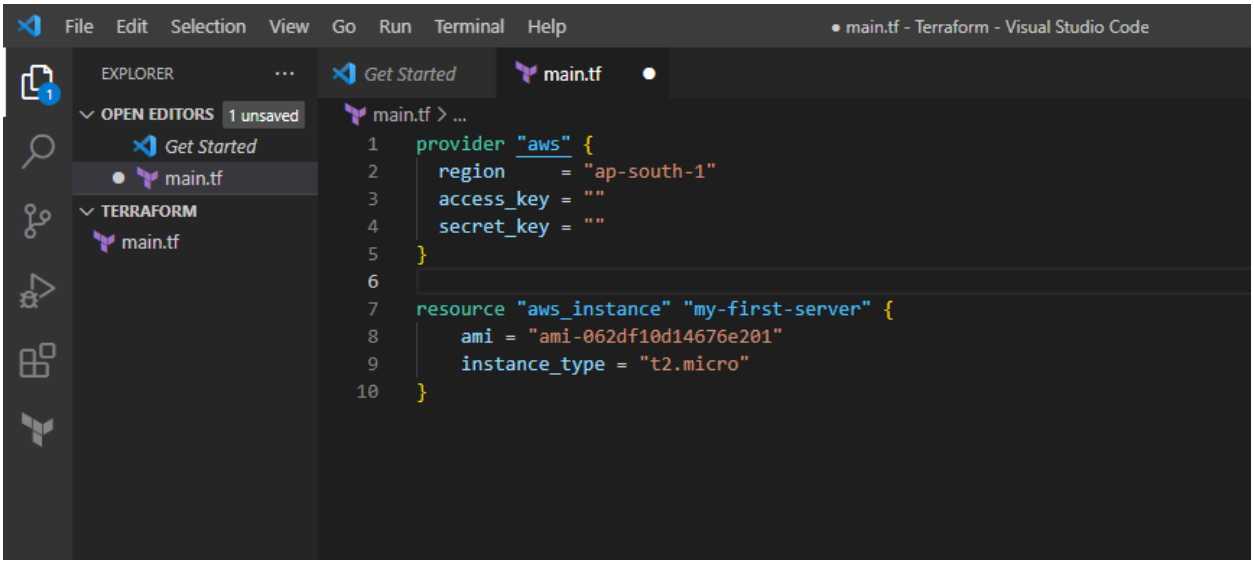
Step 3: Click on Access key if you have a key you can use that only if you don't have a key you have to create a new access key



Step 4: Create new access key and also download the root key folder



Step 5: Create new folder named terraform and also create .tf file, Now add the code with the access key and secret key along with ami of your choice and the instance type



Step 6: Use the command “terraform init” to initialize terraform

```
PROBLEMS   TERMINAL   OUTPUT   DEBUG CONSOLE

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

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.
PS C:\Users\Lavena\OneDrive\Documents\A_DevOps\Terraform>
```

Step 7: Dry-run your production network with “terraform plan”

```
PS C:\Users\Lavena\OneDrive\Documents\A_DevOps\Terraform> 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.my-first-server will be created
  + resource "aws_instance" "my-first-server" {
      + ami                                  = "ami-062df10d14676e201"
      + 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_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (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)
```

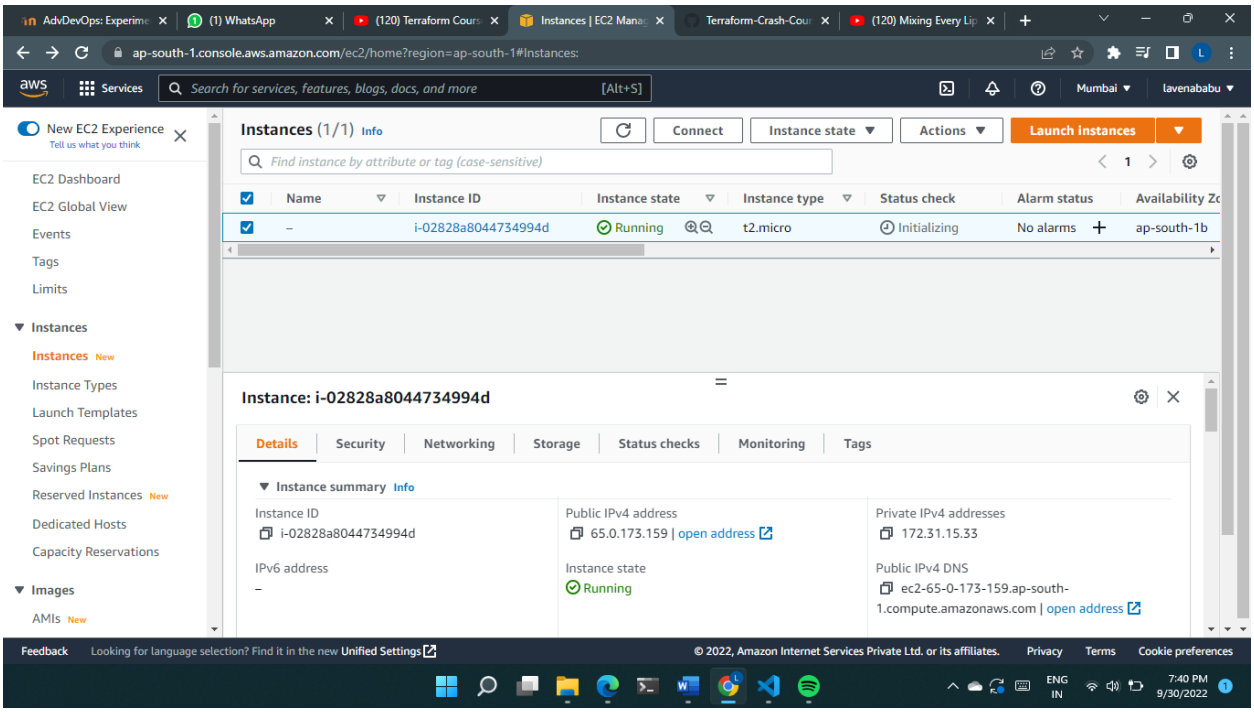
Step 8: Run our code using " terraform apply"

```
PS C:\Users\Lavena\OneDrive\Documents\A_DevOps\Terraform> 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.my-first-server will be created
  + resource "aws_instance" "my-first-server" {
      + ami                                  = "ami-062df10d14676e201"
      + 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_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (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)
```

Step 9: Enter "yes"

```
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.my-first-server: Creating...
aws_instance.my-first-server: Still creating... [10s elapsed]
aws_instance.my-first-server: Still creating... [20s elapsed]
aws_instance.my-first-server: Still creating... [30s elapsed]
aws_instance.my-first-server: Creation complete after 31s [id=i-02828a8044734994d]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\Users\Lavena\OneDrive\Documents\A_DevOps\Terraform>
```

A new instance has been created



Step 10: Destroy the instance using “terraform destroy”

```
PS C:\Users\Lavena\OneDrive\Documents\A_DevOps\Terraform> terraform destroy
aws_instance.my-first-server: Refreshing state... [id=i-02828a8044734994d]

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.my-first-server will be destroyed
  - resource "aws_instance" "my-first-server" {
      - ami                                  = "ami-062df10d14676e201" -> null
      - arn                                  = "arn:aws:ec2:ap-south-1:826050869073:instance/i-02828a8044734994d" -> 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_stop                     = false -> null
      - disable_api_termination              = false -> null
      - ebs_optimized                        = false -> null
      - get_password_data                    = false -> null
      - hibernation                          = false -> null
      - id                                   = "i-02828a8044734994d" -> null
      - instance_initiated_shutdown_behavior = "stop" -> null
      - instance_state                       = "running" -> null
      - instance_type                        = "t2.micro" -> null
      - ipv6_address_count                   = 0 -> null
      - ipv6_addresses                       = [] -> null
      - monitoring                           = false -> null
      - primary_network_interface_id         = "eni-0a72b165b4560945e" -> null
      - private_dns                          = "ip-172-31-15-33.ap-south-1.compute.internal" -> null
      - private_ip                           = "172.31.15.33" -> null
      - public_dns                           = "ec2-65-0-173-159.ap-south-1.compute.amazonaws.com" -> null
```

Step 11: Enter “yes”

```
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.my-first-server: Destroying... [id=i-02828a8044734994d]
aws_instance.my-first-server: Still destroying... [id=i-02828a8044734994d, 10s elapsed]
aws_instance.my-first-server: Still destroying... [id=i-02828a8044734994d, 20s elapsed]
aws_instance.my-first-server: Still destroying... [id=i-02828a8044734994d, 30s elapsed]
aws_instance.my-first-server: Destruction complete after 40s

Destroy complete! Resources: 1 destroyed.
PS C:\Users\Lavena\OneDrive\Documents\A_DevOps\Terraform>
```

The instance is terminated