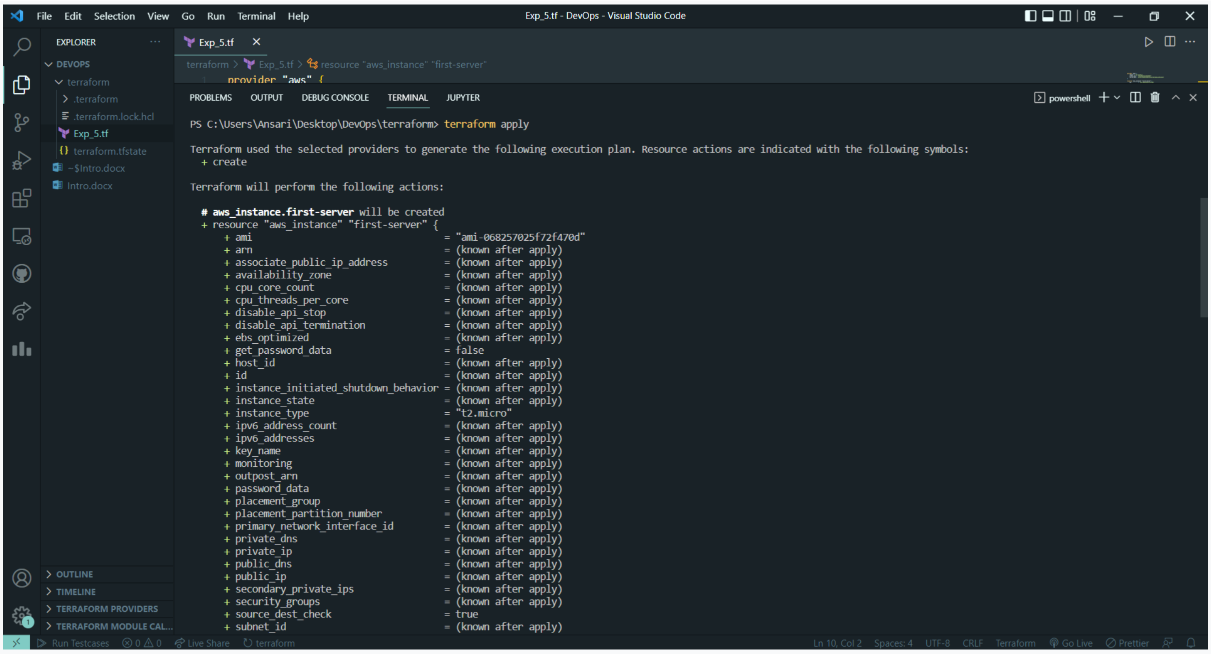
Experiment 6

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

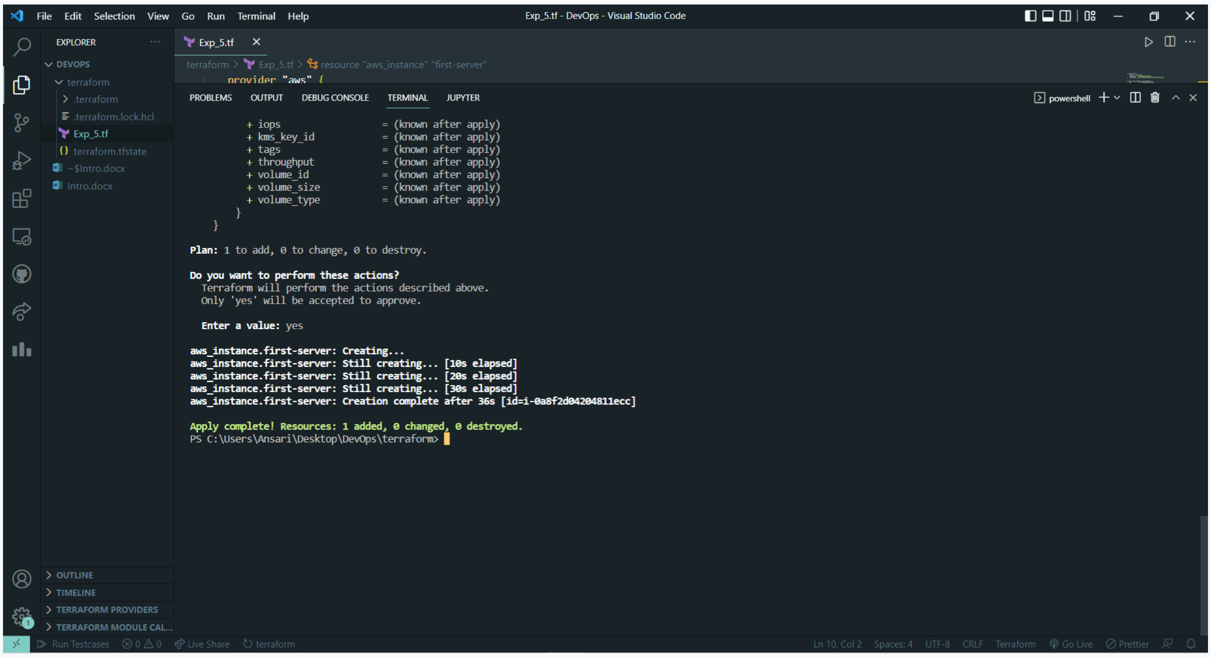
Build:

- To build a infrastructure using terraform you'll have to use the command terraform apply



```
provider "aws" {
```

```
PS C:\Users\Ansari\Desktop\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.first-server will be created
  + resource "aws_instance" "first-server" {
      + ami                                  = "ami-068257025f72f470d"
      + 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)
      + 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)
      + placement_partition_number           = (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)
```

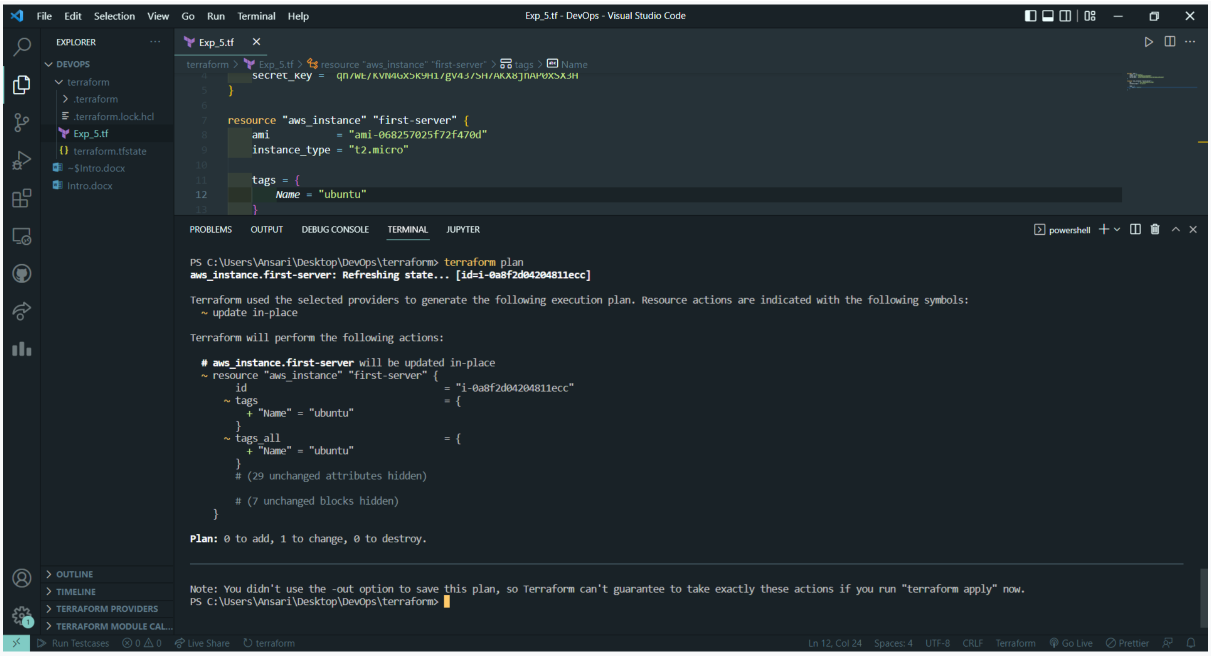


```
provider "aws" {
```

```
          + 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.first-server: Creating...
aws_instance.first-server: Still creating... [10s elapsed]
aws_instance.first-server: Still creating... [20s elapsed]
aws_instance.first-server: Still creating... [30s elapsed]
aws_instance.first-server: Creation complete after 36s [id=i-0a8f2d04204811ecc]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\Users\Ansari\Desktop\DevOps\terraform>
```

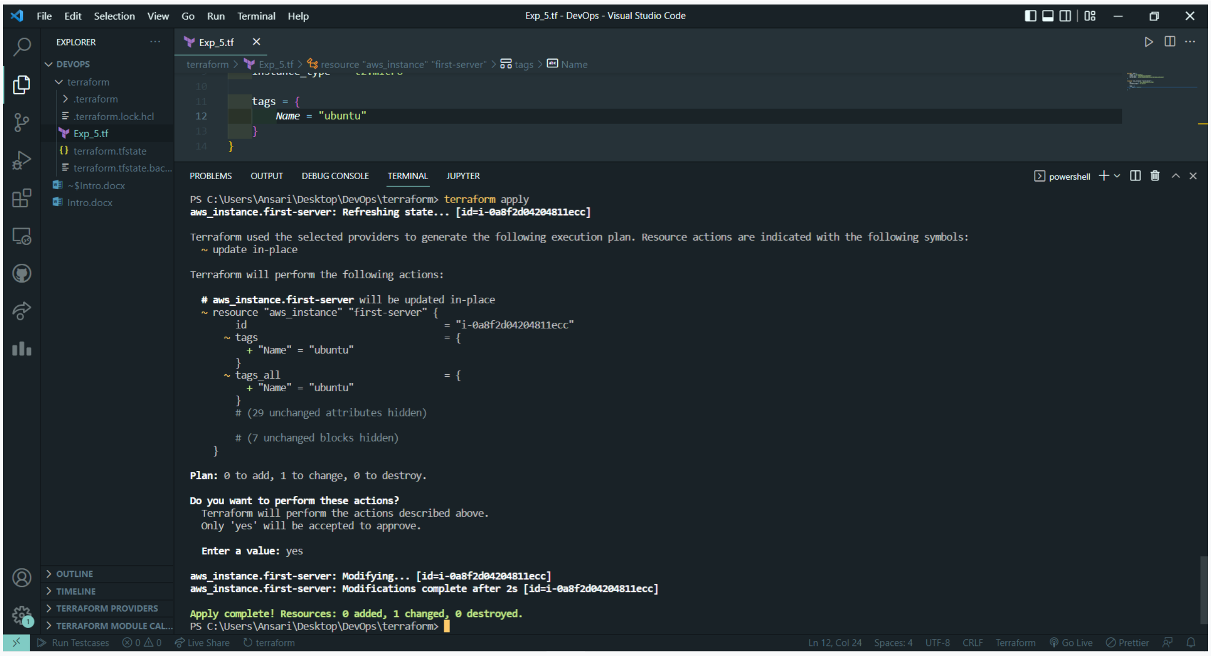
Change:

- To apply any changes to terraform we will write the changes we want to apply in the .tf file run the terraform apply command
- If you want to first see what changes will appear write the command terraform plan



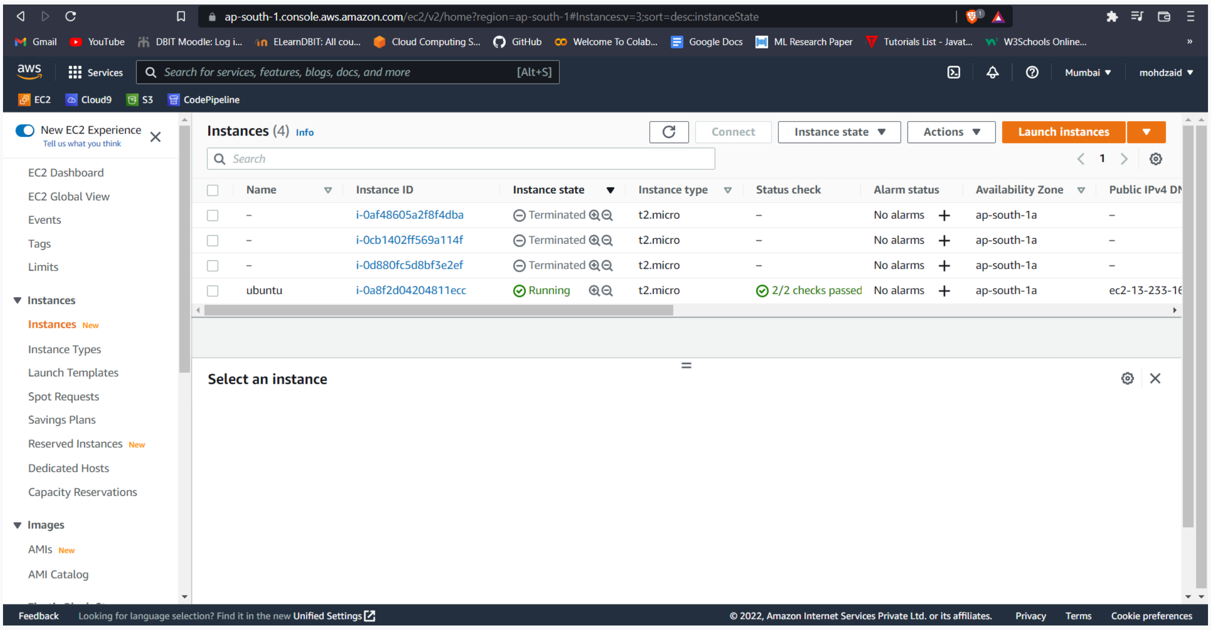
```
    secret_key =  qn7WE/KVN4GX5K9H17gV437SH7AKX8jnAP0XSX3H
}

resource "aws_instance" "first-server" {
    ami           = "ami-068257025f72f470d"
    instance_type = "t2.micro"

    tags = {
        Name = "ubuntu"
    }
```

```
PS C:\Users\Ansari\Desktop\DevOps\terraform> terraform plan
aws_instance.first-server: Refreshing state... [id=i-0a8f2d04204811ecc]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  ~ update in-place

Terraform will perform the following actions:

  # aws_instance.first-server will be updated in-place
  ~ resource "aws_instance" "first-server" {
        id                                   = "i-0a8f2d04204811ecc"
      ~ tags                                 = {
          + "Name" = "ubuntu"
        }
      ~ tags_all                             = {
          + "Name" = "ubuntu"
        }
        # (29 unchanged attributes hidden)

        # (7 unchanged blocks hidden)
    }

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

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

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
PS C:\Users\Ansari\Desktop\DevOps\terraform>
```



```

    tags = {
        Name = "ubuntu"
    }
}
```

```
PS C:\Users\Ansari\Desktop\DevOps\terraform> terraform apply
aws_instance.first-server: Refreshing state... [id=i-0a8f2d04204811ecc]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  ~ update in-place

Terraform will perform the following actions:

  # aws_instance.first-server will be updated in-place
  ~ resource "aws_instance" "first-server" {
        id                                   = "i-0a8f2d04204811ecc"
      ~ tags                                 = {
          + "Name" = "ubuntu"
        }
      ~ tags_all                             = {
          + "Name" = "ubuntu"
        }
        # (29 unchanged attributes hidden)

        # (7 unchanged blocks hidden)
    }

Plan: 0 to add, 1 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.first-server: Modifying... [id=i-0a8f2d04204811ecc]
aws_instance.first-server: Modifications complete after 2s [id=i-0a8f2d04204811ecc]

Apply complete! Resources: 0 added, 1 changed, 0 destroyed.
PS C:\Users\Ansari\Desktop\DevOps\terraform>
```



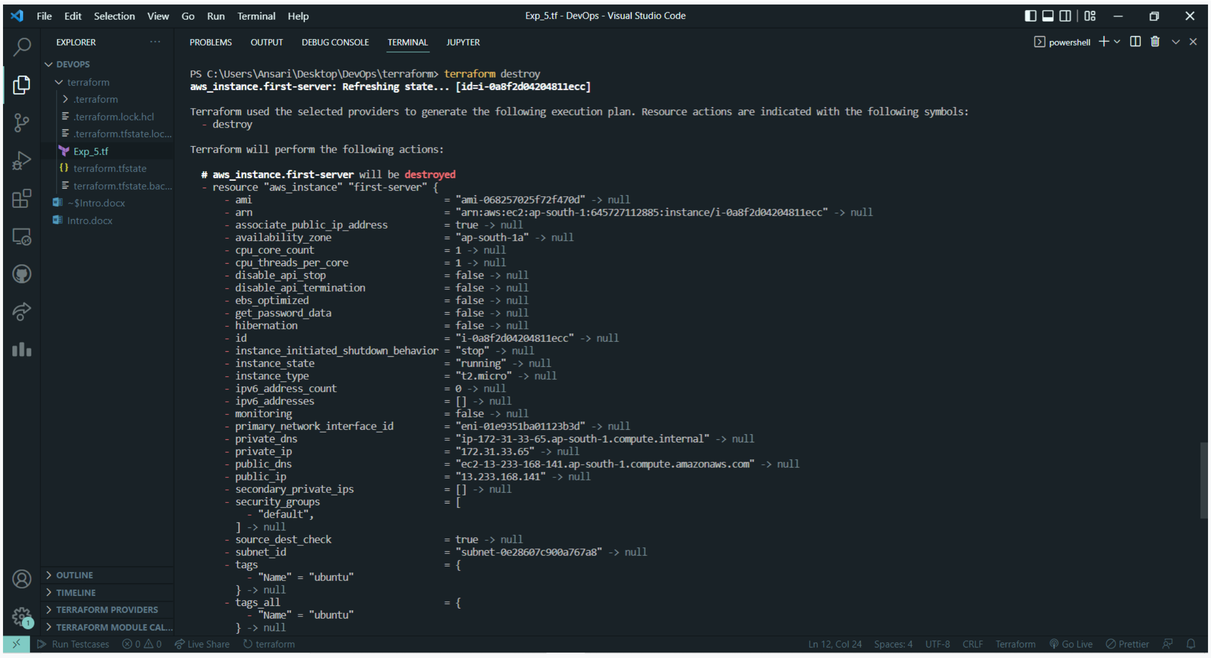
Instances (4) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DN
–	i-0af48605a2f8f4dba	Terminated	t2.micro	–	No alarms	ap-south-1a	–
–	i-0cb1402ff569a114f	Terminated	t2.micro	–	No alarms	ap-south-1a	–
–	i-0d880fc5d8bf3e2ef	Terminated	t2.micro	–	No alarms	ap-south-1a	–
ubuntu	i-0a8f2d04204811ecc	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	ec2-13-233-16

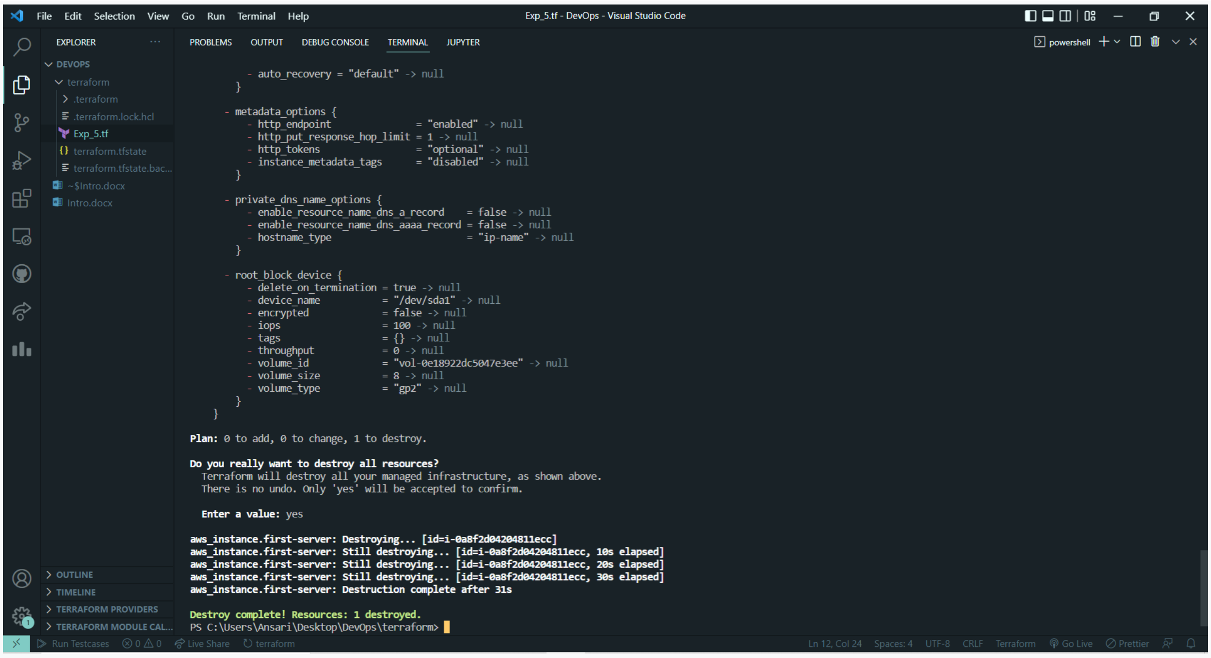
Select an instance

Destroy:

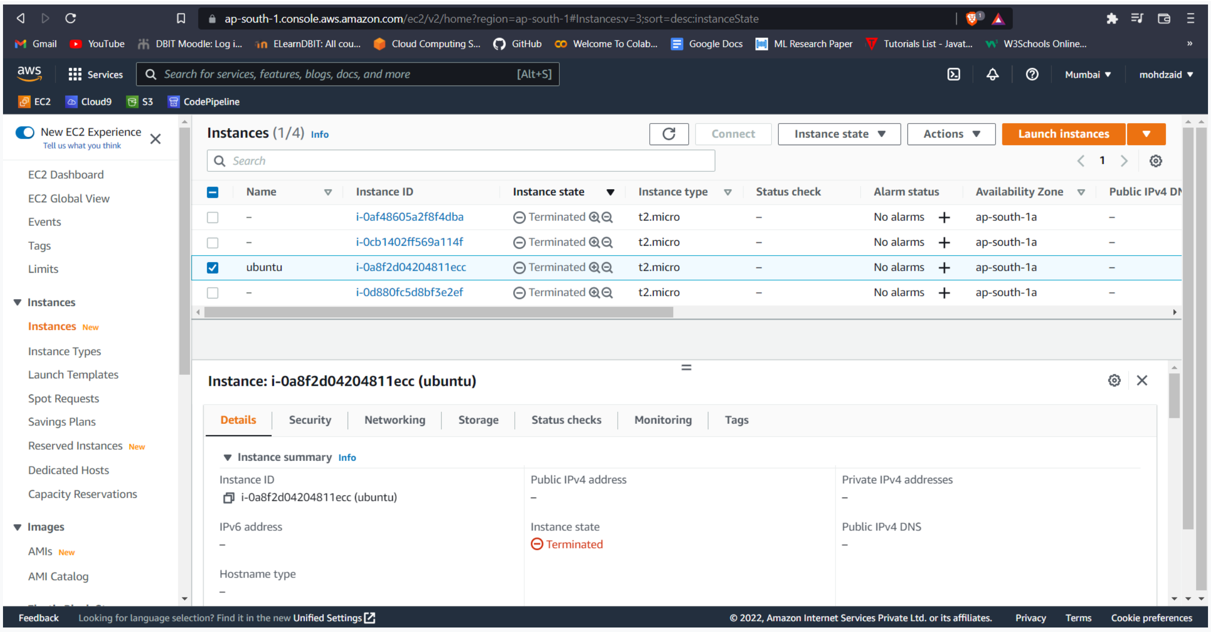
- To destroy a terraform infrastructure we will use the command terraform destroy
- There is another way to destroy terraform infrastructure just comment the resource part of the code and use the terraform apply command



```
PS C:\Users\Ansari\Desktop\DevOps\terraform> terraform destroy
aws_instance.first-server: Refreshing state... [id=i-0a8f2d04204811ecc]

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.first-server will be destroyed
  - resource "aws_instance" "first-server" {
      - ami                                  = "ami-068257025f72f470d" -> null
      - arn                                  = "arn:aws:ec2:ap-south-1:645727112885:instance/i-0a8f2d04204811ecc" -> null
      - associate_public_ip_address          = true -> null
      - availability_zone                    = "ap-south-1a" -> 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-0a8f2d04204811ecc" -> 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-01e9351ba01123b3d" -> null
      - private_dns                          = "ip-172-31-33-65.ap-south-1.compute.internal" -> null
      - private_ip                           = "172.31.33.65" -> null
      - public_dns                           = "ec2-13-233-168-141.ap-south-1.compute.amazonaws.com" -> null
      - public_ip                            = "13.233.168.141" -> null
      - secondary_private_ips                = [] -> null
      - security_groups                      = [
          - "default",
        ] -> null
      - source_dest_check                    = true -> null
      - subnet_id                            = "subnet-0e28607c900a767a8" -> null
      - tags                                 = {
          - "Name" = "ubuntu"
        } -> null
      - tags_all                             = {
          - "Name" = "ubuntu"
        } -> null
```


File Edit Selection View Go Run Terminal Help
Exp_5.tf - DevOps - Visual Studio Code
EXPLORER
DEVOPS
terraform
.terraform
.terraform.lock.hcl
Exp_5.tf
terraform.tfstate
terraform.tfstate.bac...
~$Intro.docx
Intro.docx
OUTLINE
TIMELINE
TERRAFORM PROVIDERS
TERRAFORM MODULE CAL...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
powershell
Run Testcases ⊗ 0 ⚠ 0 Live Share terraform
Ln 12, Col 24 Spaces: 4 UTF-8 CRLF Terraform Go Live Prettier

```
          - auto_recovery = "default" -> null
        }

      - metadata_options {
          - http_endpoint               = "enabled" -> null
          - http_put_response_hop_limit = 1 -> null
          - http_tokens                 = "optional" -> null
          - instance_metadata_tags      = "disabled" -> null
        }

      - private_dns_name_options {
          - enable_resource_name_dns_a_record    = false -> null
          - enable_resource_name_dns_aaaa_record = false -> null
          - hostname_type                        = "ip-name" -> null
        }

      - root_block_device {
          - delete_on_termination = true -> null
          - device_name           = "/dev/sda1" -> null
          - encrypted             = false -> null
          - iops                  = 100 -> null
          - tags                  = {} -> null
          - throughput            = 0 -> null
          - volume_id             = "vol-0e18922dc5047e3ee" -> null
          - volume_size           = 8 -> null
          - volume_type           = "gp2" -> null
        }
    }

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.first-server: Destroying... [id=i-0a8f2d04204811ecc]
aws_instance.first-server: Still destroying... [id=i-0a8f2d04204811ecc, 10s elapsed]
aws_instance.first-server: Still destroying... [id=i-0a8f2d04204811ecc, 20s elapsed]
aws_instance.first-server: Still destroying... [id=i-0a8f2d04204811ecc, 30s elapsed]
aws_instance.first-server: Destruction complete after 31s

Destroy complete! Resources: 1 destroyed.
PS C:\Users\Ansari\Desktop\DevOps\terraform> 
```


ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#Instances:v=3;sort=desc:instanceState
Gmail YouTube DBIT Moodle: Log i... ELearnDBIT: All cou... Cloud Computing S... GitHub Welcome To Colab... Google Docs ML Research Paper Tutorials List - Javat... W3Schools Online...
aws Services Search for services, features, blogs, docs, and more [Alt+S] Mumbai ▼ mohdzaid ▼
EC2 Cloud9 S3 CodePipeline
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
▼ Images
AMIs New
AMI Catalog
Connect
Instance state ▼
Actions ▼
Launch instances ▼
Search
1
Feedback Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Instances (1/4) Info

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DN
□	–	i-0af48605a2f8f4dba	⊖ Terminated	t2.micro	–	No alarms +	ap-south-1a	–
□	–	i-0cb1402ff569a114f	⊖ Terminated	t2.micro	–	No alarms +	ap-south-1a	–
☑	ubuntu	i-0a8f2d04204811ecc	⊖ Terminated	t2.micro	–	No alarms +	ap-south-1a	–
□	–	i-0d880fc5d8bf3e2ef	⊖ Terminated	t2.micro	–	No alarms +	ap-south-1a	–

Instance: i-0a8f2d04204811ecc (ubuntu)

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

▼ Instance summary Info

Instance ID i-0a8f2d04204811ecc (ubuntu)	Public IPv4 address –	Private IPv4 addresses –
IPv6 address –	Instance state ⊖ Terminated	Public IPv4 DNS –
Hostname type –		