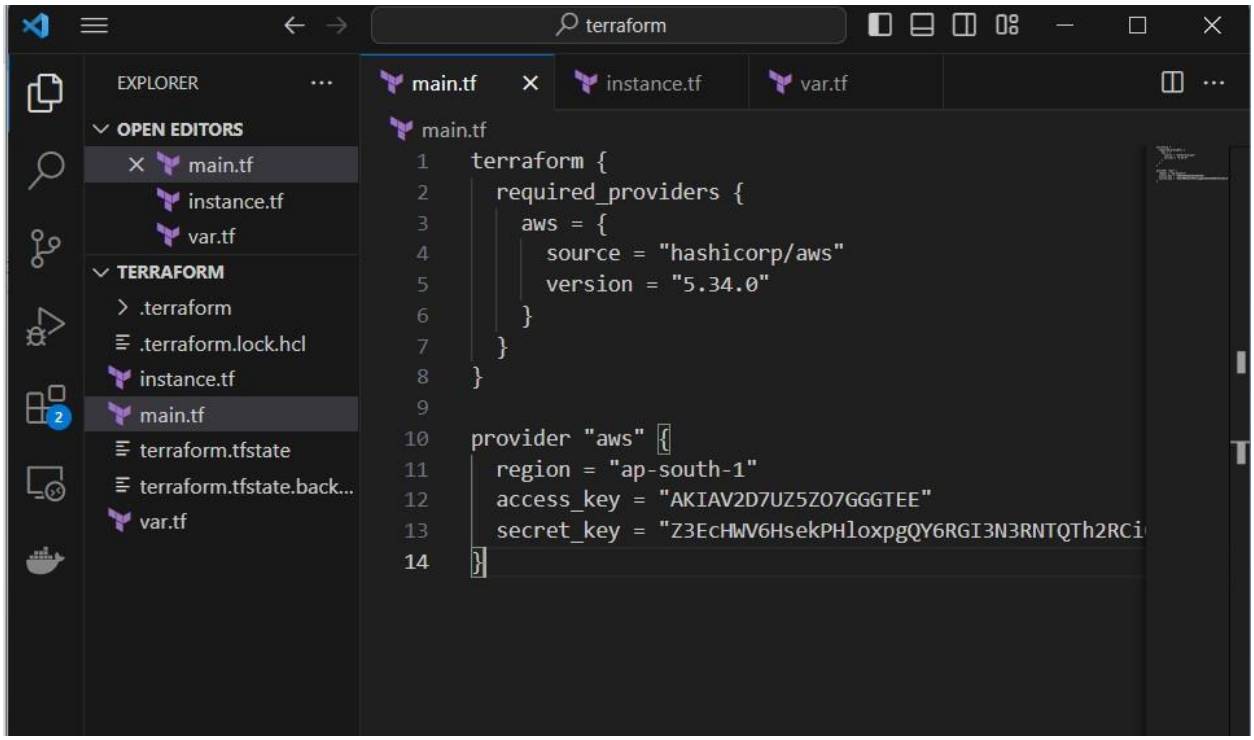
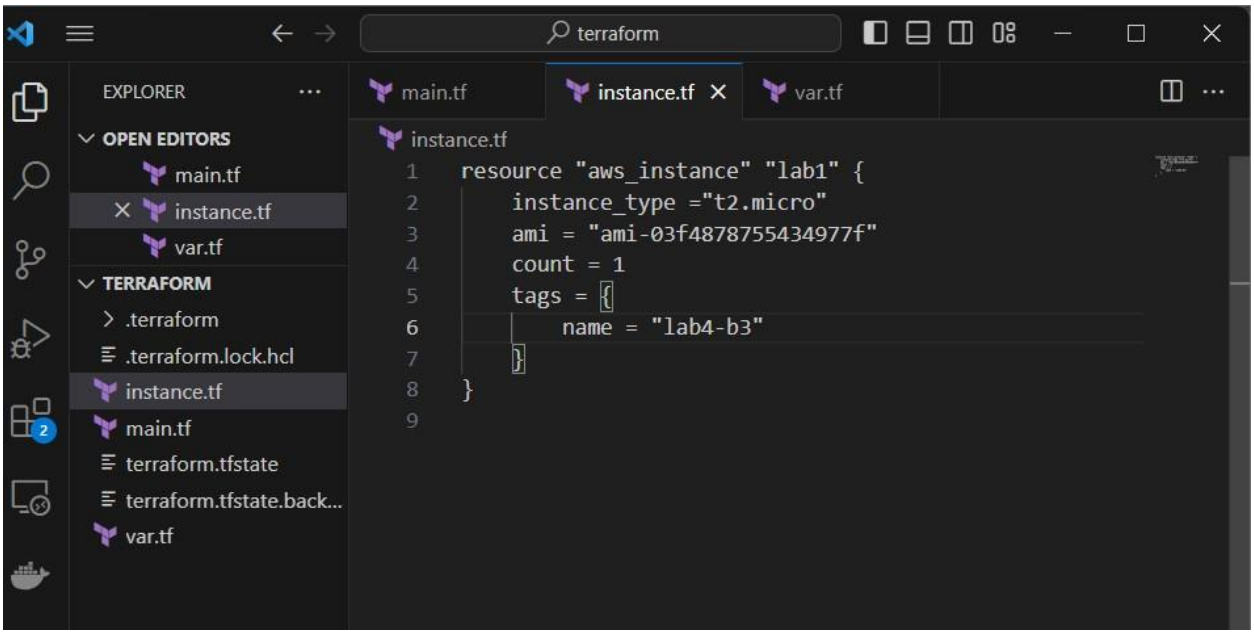
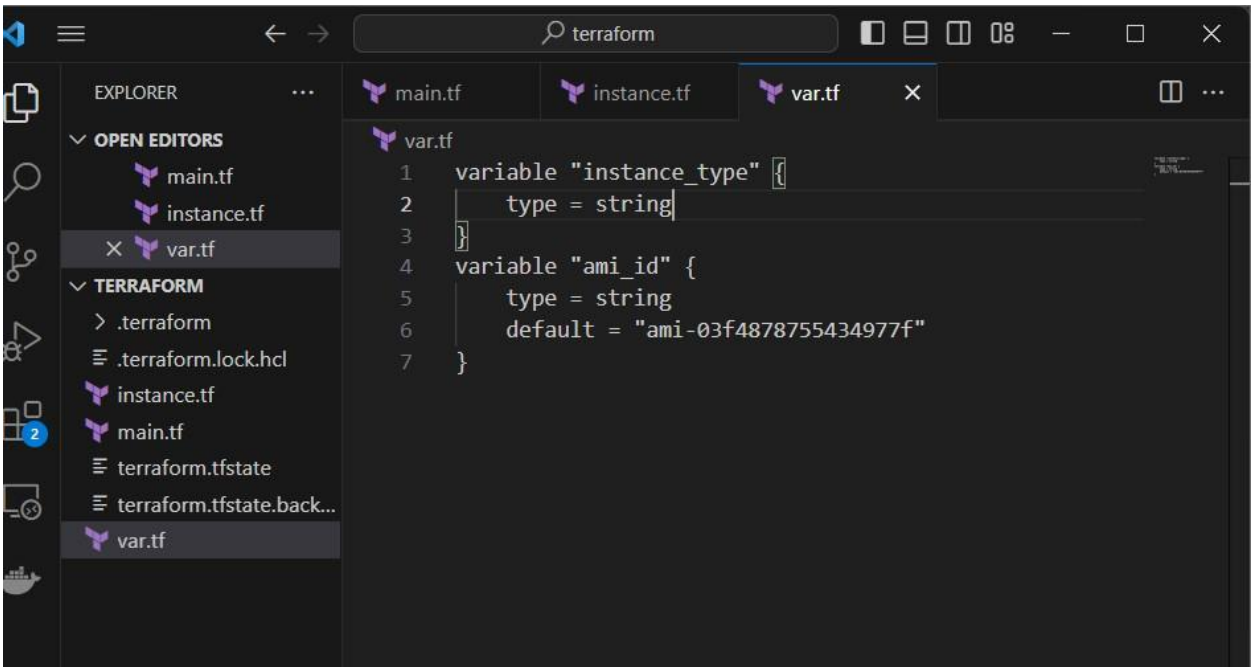
LAB-5

Terraform Variable with Command Line Argument

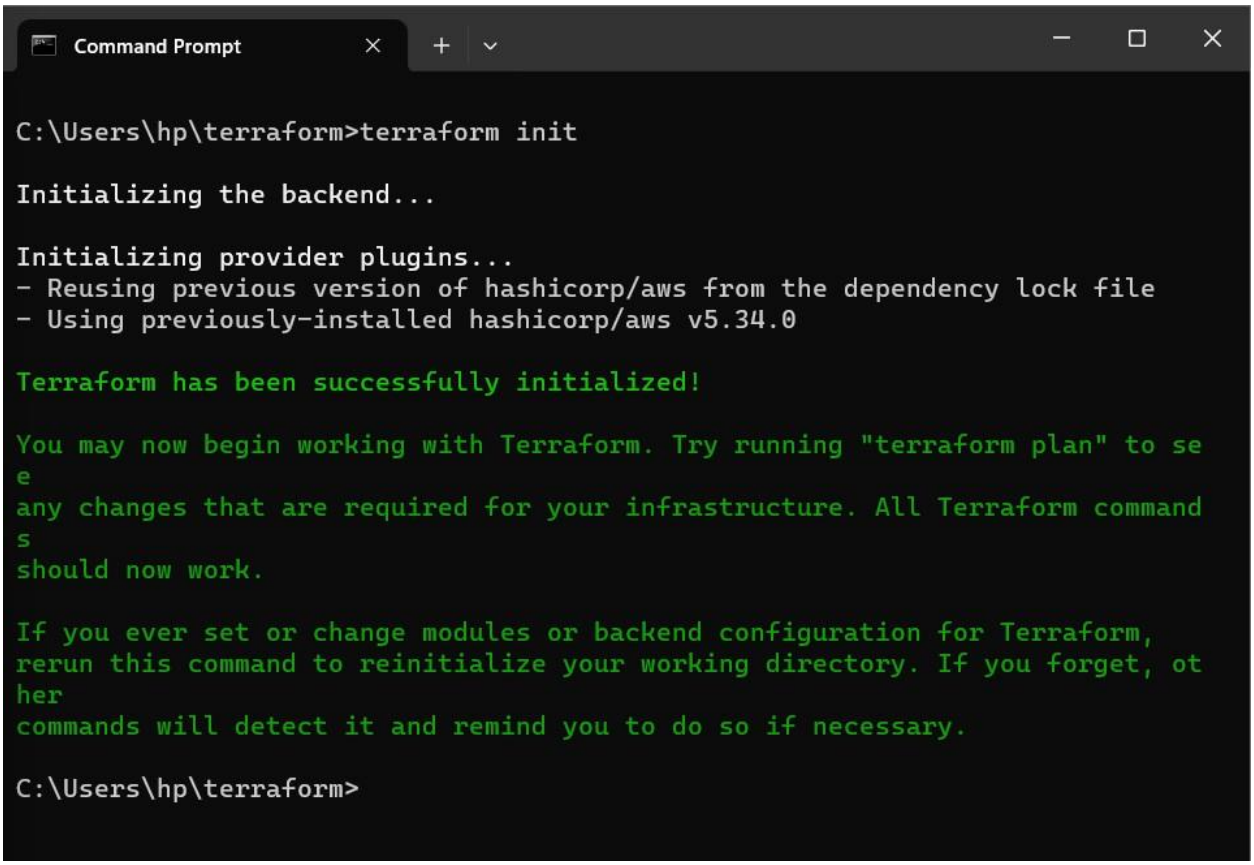
Step 1: Make changes in var.tf file





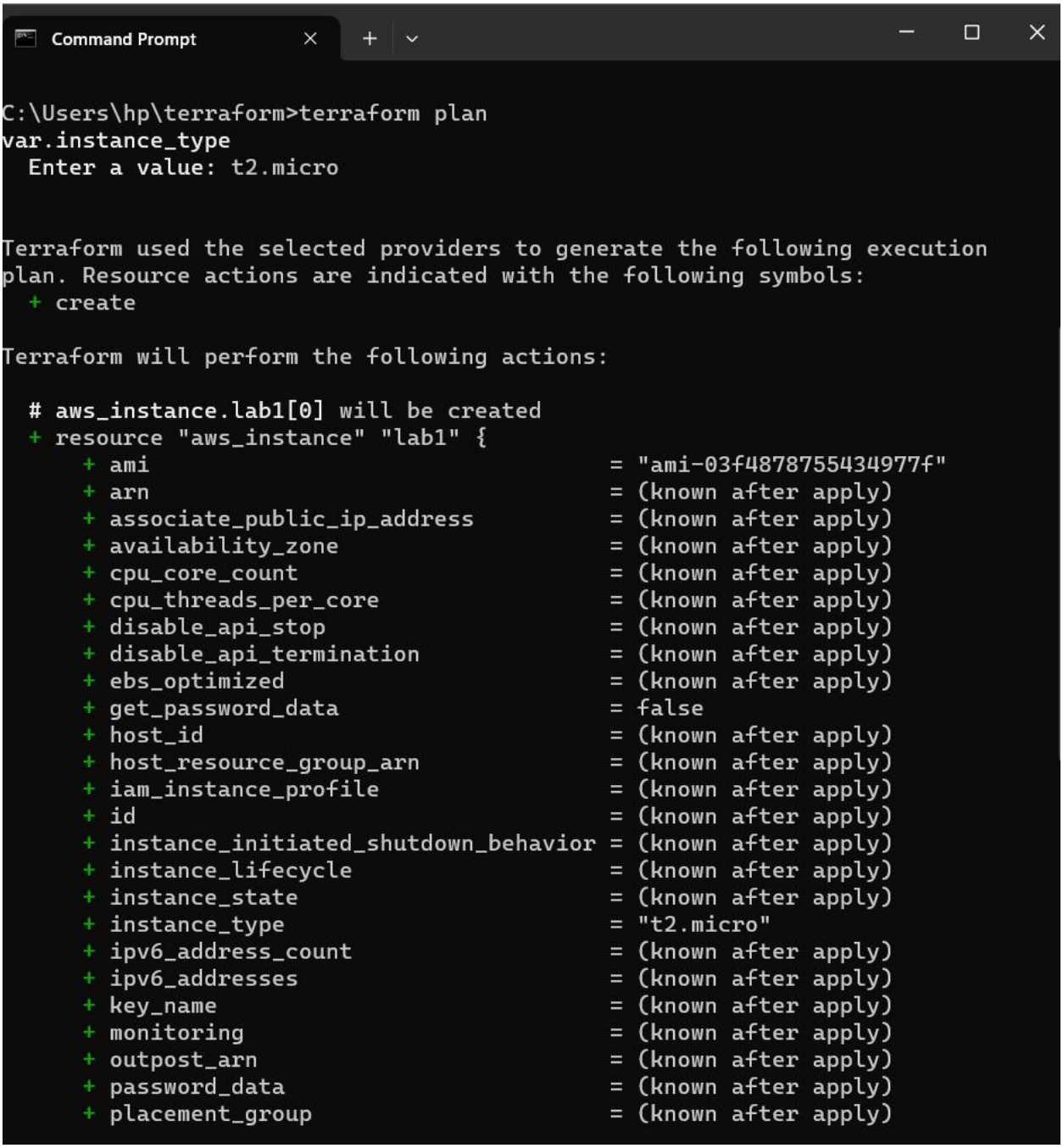


Step 2: Now we need to run terraform cycle

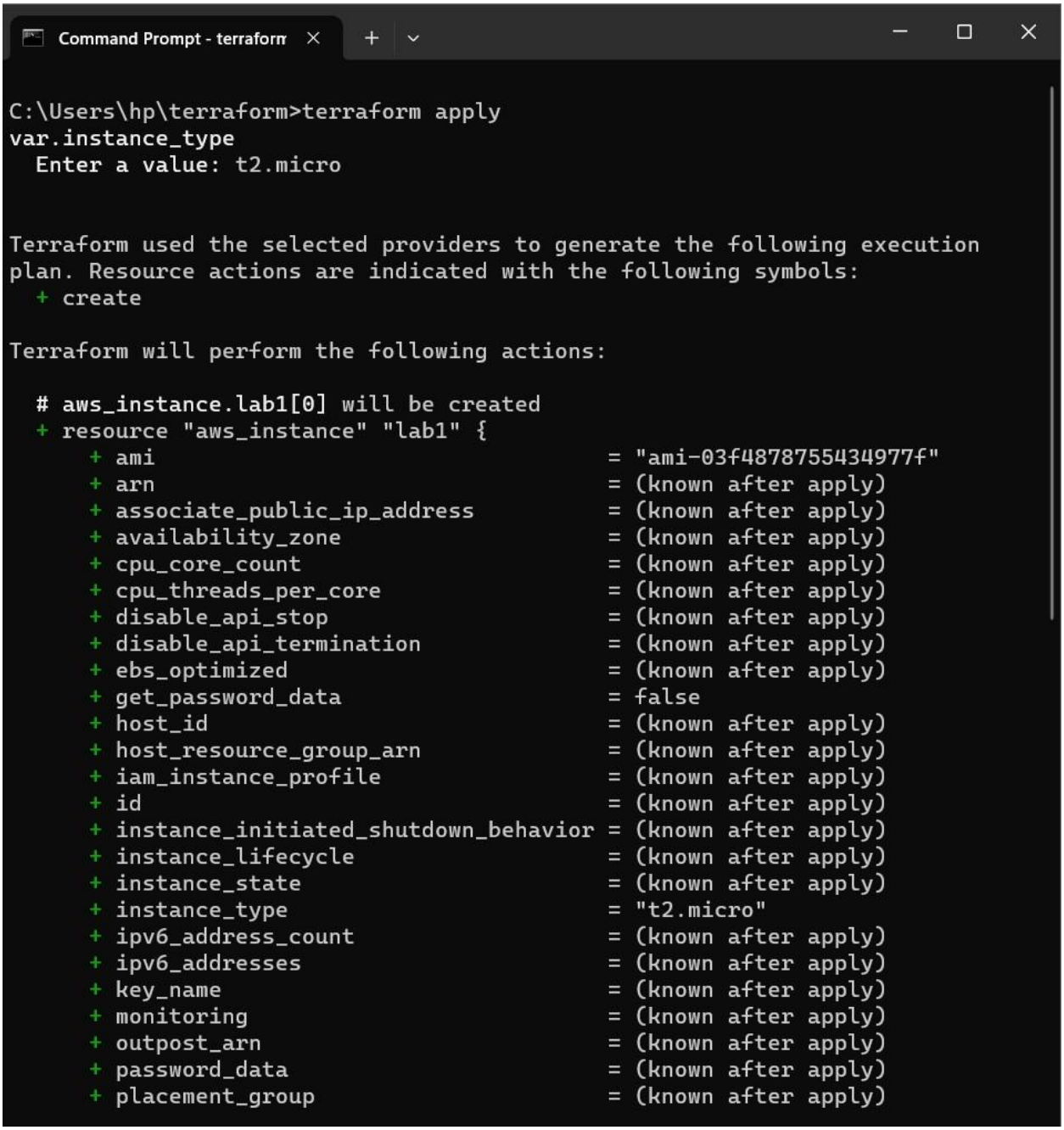


Now we have to ways to declare variable in CLI

First: We can give value after running terraform plan

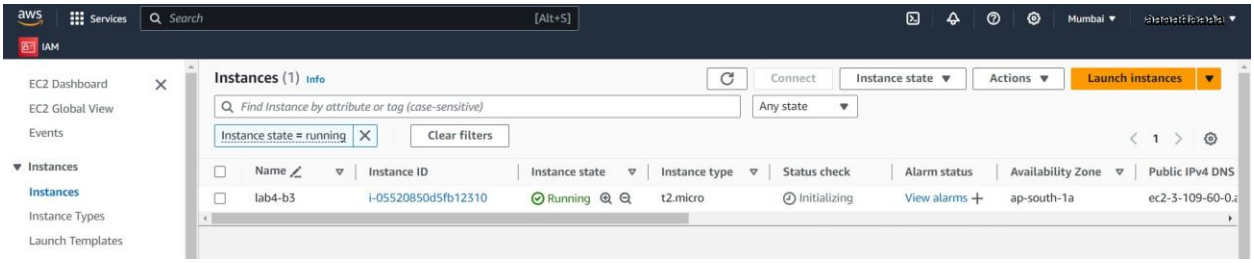


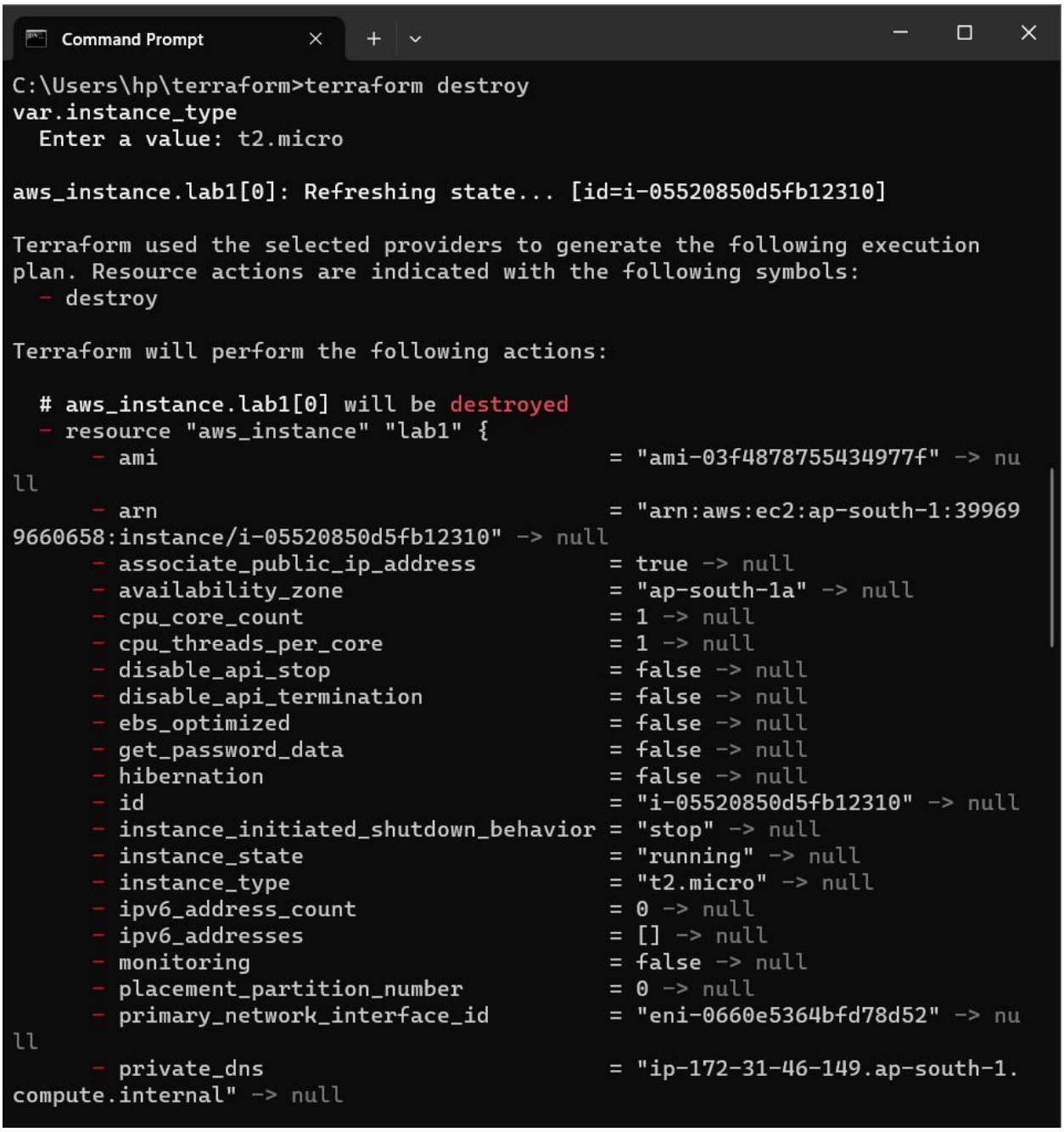
Second: By declaring variable during running command

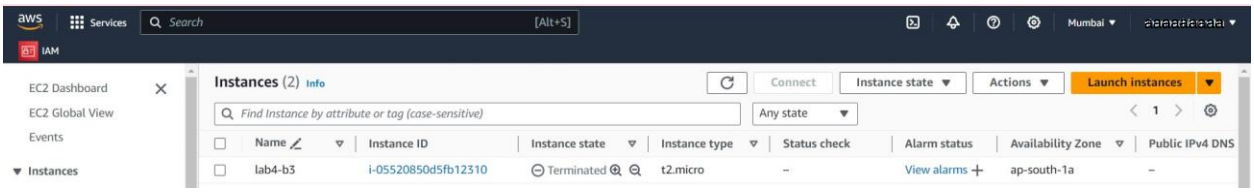


Command Prompt

```
      + 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
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "name" = "lab4-b3"
        }
      + tags_all                             = {
          + "name" = "lab4-b3"
        }
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (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.lab1[0]: Creating...
aws_instance.lab1[0]: Still creating... [10s elapsed]
aws_instance.lab1[0]: Still creating... [20s elapsed]
aws_instance.lab1[0]: Still creating... [30s elapsed]
aws_instance.lab1[0]: Creation complete after 33s [id=i-05520850d5fb12310]

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

C:\Users\hp\terraform>
```




Command Prompt
C:\Users\hp\terraform>terraform destroy
var.instance_type
  Enter a value: t2.micro
aws_instance.lab1[0]: Refreshing state... [id=i-05520850d5fb12310]
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.lab1[0] will be destroyed
  - resource "aws_instance" "lab1" {
      - ami = "ami-03f48787554349777f" -> null
      - arn = "arn:aws:ec2:ap-south-1:399699660658:instance/i-05520850d5fb12310" -> 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-05520850d5fb12310" -> 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
      - placement_partition_number = 0 -> null
      - primary_network_interface_id = "eni-0660e5364bfd78d52" -> null
      - private_dns = "ip-172-31-46-149.ap-south-1.compute.internal" -> null

Command Prompt

```
          - 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-0529c4704b8bb77d8" -> 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.lab1[0]: Destroying... [id=i-05520850d5fb12310]
aws_instance.lab1[0]: Still destroying... [id=i-05520850d5fb12310, 10s elaps
ed]
aws_instance.lab1[0]: Still destroying... [id=i-05520850d5fb12310, 20s elaps
ed]
aws_instance.lab1[0]: Still destroying... [id=i-05520850d5fb12310, 30s elaps
ed]
aws_instance.lab1[0]: Still destroying... [id=i-05520850d5fb12310, 40s elaps
ed]
aws_instance.lab1[0]: Destruction complete after 41s

Destroy complete! Resources: 1 destroyed.

C:\Users\hp\terraform>
```