SCHOOL OF COMPUTER SCIENCE

UNIVERSITY OF PETROLEUM AND ENERGY STUDIES
DEHRADUN, UTTARAKHAND

SYSTEM MONITORING AND CONFIGURATION MANAGEMENT

LAB FILE

(2024-2025)

6TH SEMESTER

Submitted To:

Dr. Hitesh Kumar Sharma

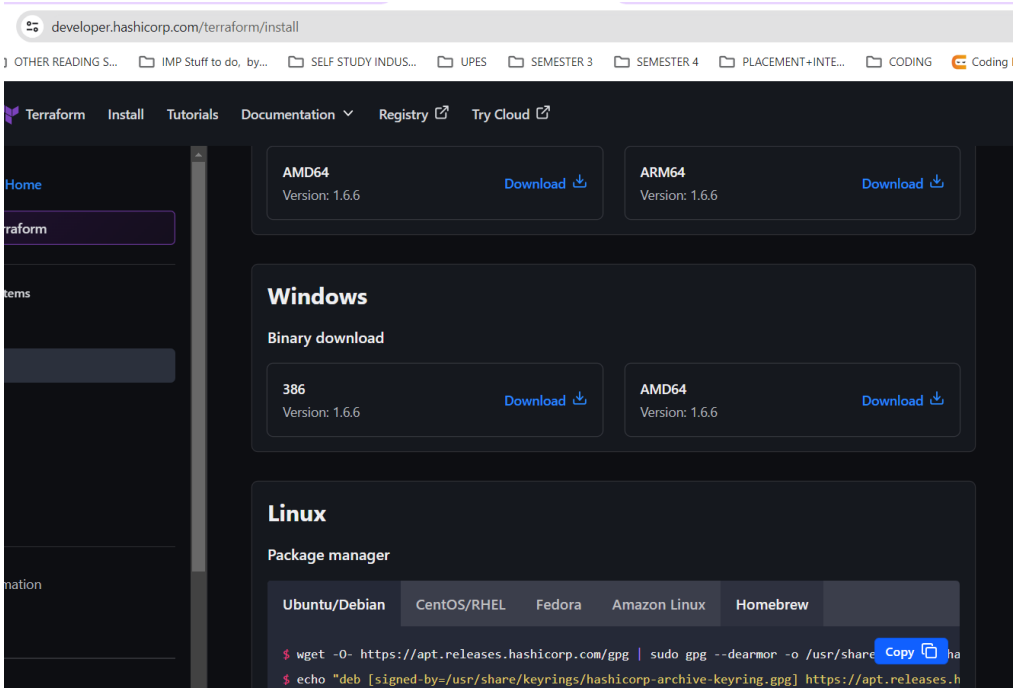
Submitted By:

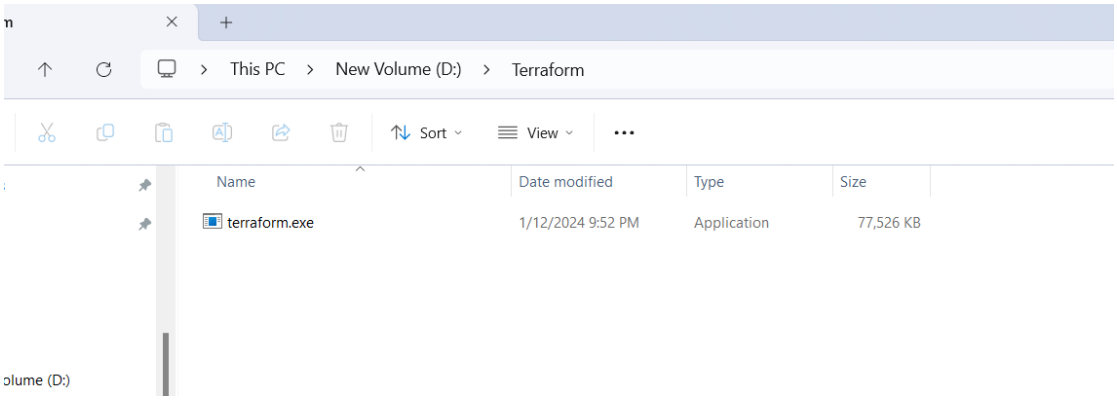
Siddhi Jain
B. Tech CSE DevOps
Sap id- 500090875
Batch 1
R2142210770

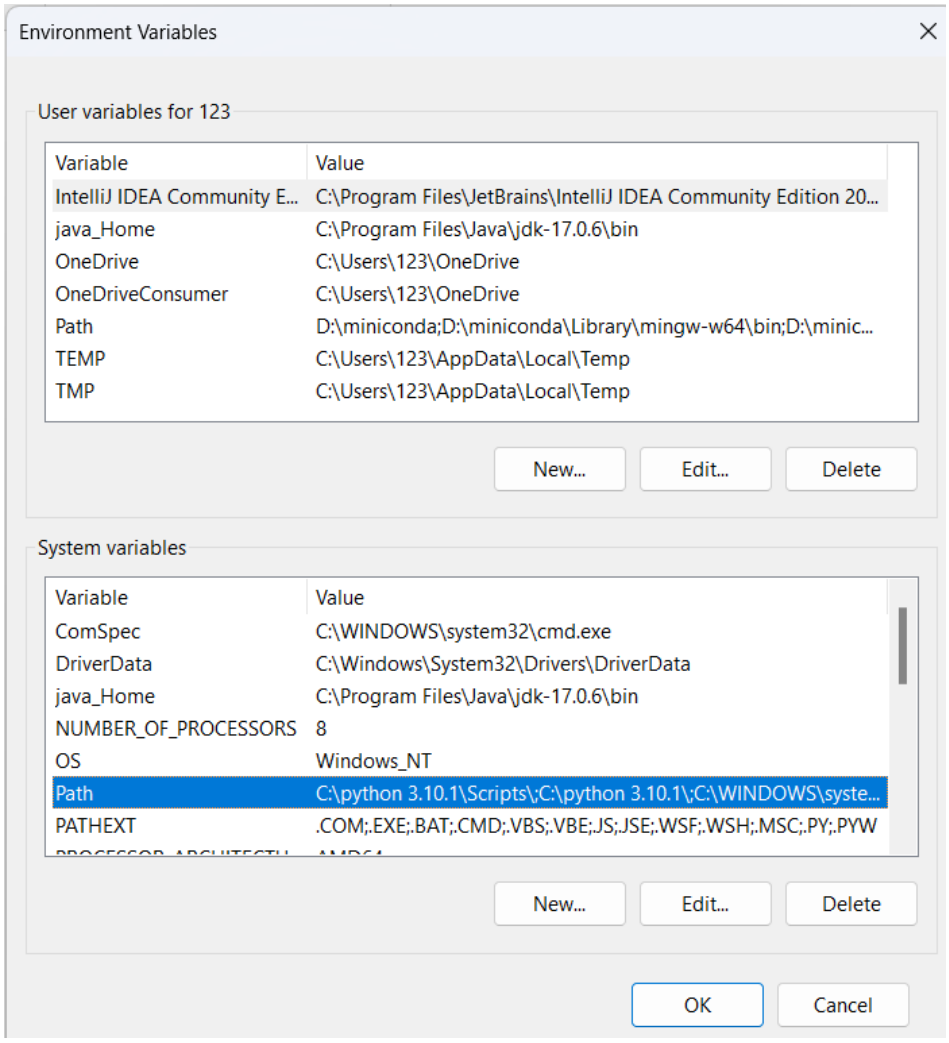
LAB EXERCISE 1

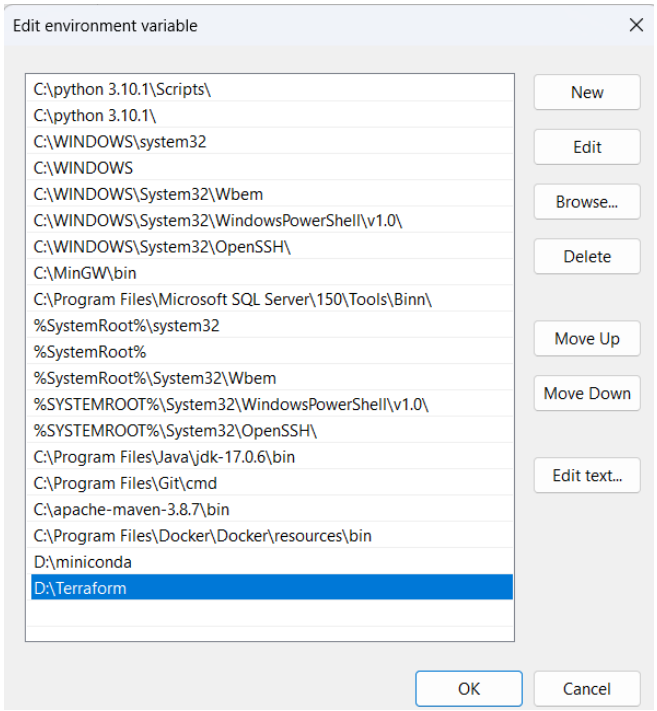
Aim: Install Terraform on Windows

Download Terraform, add it to path and verify install.






Environment Variables
User variables for 123
Variable
Value
IntelliJ IDEA Community E...
C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 20...
java_Home
C:\Program Files\Java\jdk-17.0.6\bin
OneDrive
C:\Users\123\OneDrive
OneDriveConsumer
C:\Users\123\OneDrive
Path
D:\miniconda;D:\miniconda\Library\mingw-w64\bin;D:\minic...
TEMP
C:\Users\123\AppData\Local\Temp
TMP
C:\Users\123\AppData\Local\Temp
New...
Edit...
Delete
System variables
Variable
Value
ComSpec
C:\WINDOWS\system32\cmd.exe
DriverData
C:\Windows\System32\Drivers\DriverData
java_Home
C:\Program Files\Java\jdk-17.0.6\bin
NUMBER_OF_PROCESSORS
8
OS
Windows_NT
Path
C:\python 3.10.1\Scripts\;C:\python 3.10.1\;C:\WINDOWS\syste...
PATHEXT
.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC;.PY;.PYW
New...
Edit...
Delete
OK
Cancel


Edit environment variable
C:\python 3.10.1\Scripts\
C:\python 3.10.1\
C:\WINDOWS\system32
C:\WINDOWS
C:\WINDOWS\System32\Wbem
C:\WINDOWS\System32\WindowsPowerShell\v1.0\
C:\WINDOWS\System32\OpenSSH\
C:\MinGW\bin
C:\Program Files\Microsoft SQL Server\150\Tools\Binn\
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files\Java\jdk-17.0.6\bin
C:\Program Files\Git\cmd
C:\apache-maven-3.8.7\bin
C:\Program Files\Docker\Docker\resources\bin
D:\miniconda
D:\Terraform
New
Edit
Browse...
Delete
Move Up
Move Down
Edit text...
OK
Cancel

On cmd run this command for the successful installation.

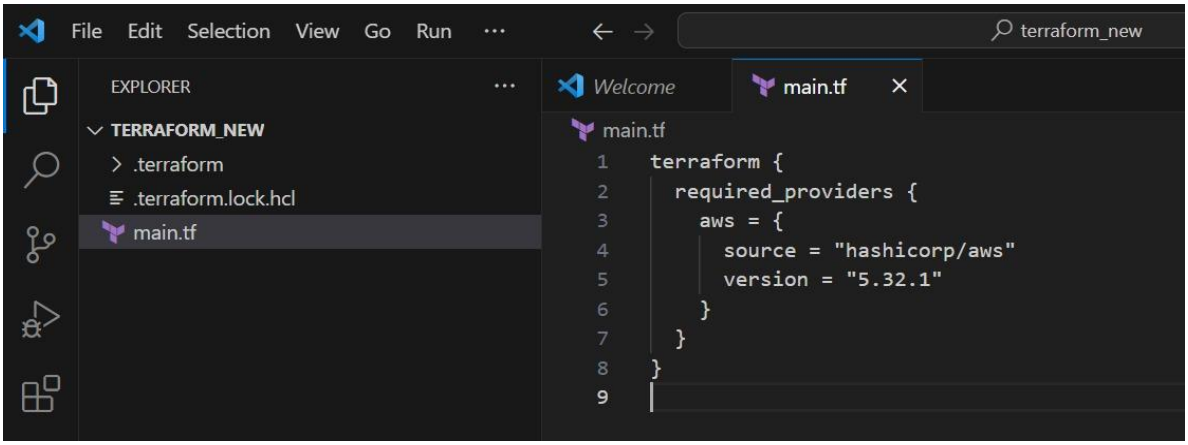
```
Microsoft Windows [Version 10.0.22621.3085]
(c) Microsoft Corporation. All rights reserved.

C:\Users\jains>terraform --version
Terraform v1.6.6
on windows_amd64
```

*******End of Experiment-1*******

LAB EXERCISE 2

Aim: Terraform AWS Provider and IAM User Setting

Step1: Create Terraform Configuration File (main.tf)



Step 2: Initialize Terraform:

```
C:\Users\jains>cd "C:\Users\jains\Desktop\terraform_new"

C:\Users\jains\Desktop\terraform_new>terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.32.1"...
- Installing hashicorp/aws v5.32.1...
- Installed hashicorp/aws v5.32.1 (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.
```

*******End of Experiment-2*******

LAB EXERCISE 3

Aim: Provisioning an EC2 Instance on AWS

Step 1: Create Terraform Configuration File (main.tf)

```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "5.32.1"
    }
  }
}
provider "aws" {
region = "ap-south-1"
access_key = "AAKIA6GBMF5EUSILM7YFD"
secret_key = "O4bcqpe7uZr3BLP1rHtAIC769MzmA9fGhiFmpxch"
}
```

Step 2: Create Terraform Configuration File for EC2 instance (instance.tf)

```
resource "aws_instance" "My_instance" {
  ami                    = "ami-00952f27cf14db9cd"
  instance_type          = "t2.micro"
  count = 1
  tags = {
    Name = "UPES-EC2-Instance"
  }
}
```

Step 3: Initialize Terraform:

```
C:\Users\jains>cd "C:\Users\jains\Desktop\terraform_new"

C:\Users\jains\Desktop\terraform_new>terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.32.1"...
- Installing hashicorp/aws v5.32.1...
- Installed hashicorp/aws v5.32.1 (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.
```

Step 4: Apply Validate

```
C:\Users\jains\Desktop\terraform_new>terraform validate

  Error: Unsupported block type

    on instance.tf line 5, in resource "aws_instance" "My_instance":
     5:   tags{

  Blocks of type "tags" are not expected here. Did you mean to define argument "tags"? If so, use the equals sign to
  assign it a value.


C:\Users\jains\Desktop\terraform_new>terraform validate
Success! The configuration is valid.
```

Step 5: Review Plan:

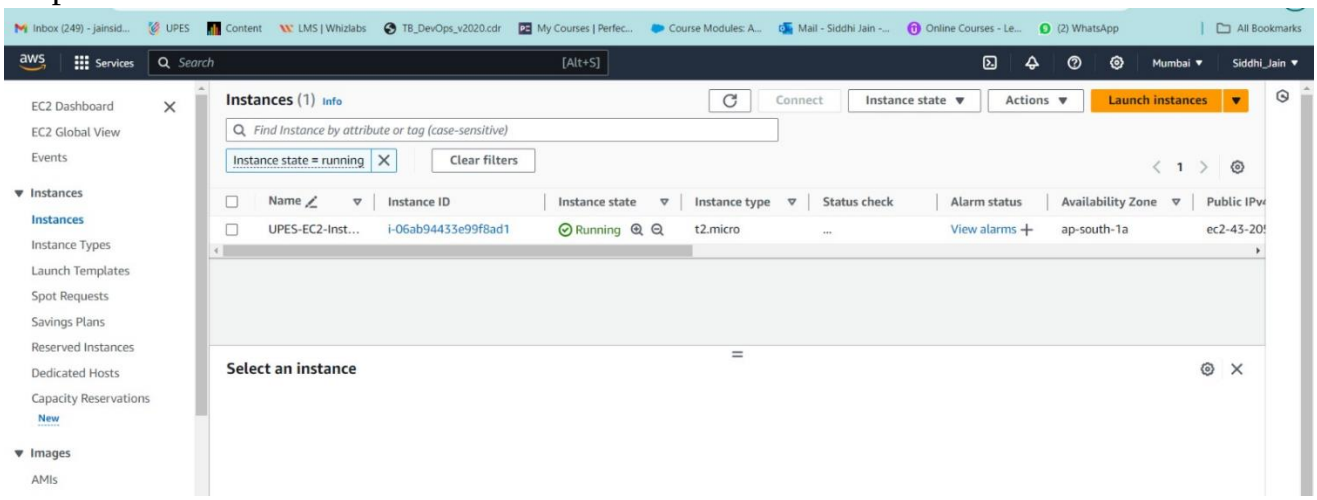
```
C:\Users\jains\Desktop\terraform_new>terraform plan

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_instance[0] will be created
  + resource "aws_instance" "My_instance" {
      + ami                                  = "ami-03f4878755434977f"
      + 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)
```

Step 6: Apply Changes

```
C:\Users\jains\Desktop\terraform_new>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_instance[0] will be created
  + resource "aws_instance" "My_instance" {
      + ami                                  = "ami-03f4878755434977f"
      + 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)
```

```
    }

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_instance[0]: Creating...
aws_instance.My_instance[0]: Still creating... [10s elapsed]
aws_instance.My_instance[0]: Still creating... [20s elapsed]
aws_instance.My_instance[0]: Creation complete after 24s [id=i-06ab94433e99f8ad1]

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

Step 7: Search on AWS whether or not the instance has been created.

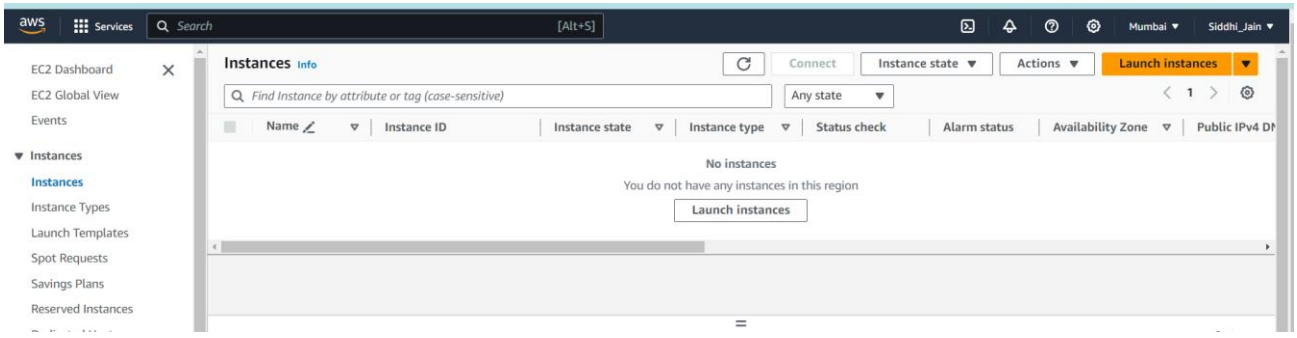


Step 8: Cleanup Resources

```
C:\Users\jains\Desktop\terraform_new>terraform destroy
aws_instance.My_instance[0]: Refreshing state... [id=i-06ab94433e99f8ad1]

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_instance[0] will be destroyed
  - resource "aws_instance" "My_instance" {
      - ami                                  = "ami-03f4878755434977f" -> null
      - arn                                  = "arn:aws:ec2:ap-south-1:975050238249:instance/i-06ab94433e99f8ad1" -> 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-06ab94433e99f8ad1" -> 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.My_instance[0]: Destroying... [id=i-06ab94433e99f8ad1]
aws_instance.My_instance[0]: Still destroying... [id=i-06ab94433e99f8ad1, 10s elapsed]
aws_instance.My_instance[0]: Still destroying... [id=i-06ab94433e99f8ad1, 20s elapsed]
aws_instance.My_instance[0]: Still destroying... [id=i-06ab94433e99f8ad1, 30s elapsed]
aws_instance.My_instance[0]: Destruction complete after 31s

Destroy complete! Resources: 1 destroyed.

C:\Users\jains\Desktop\terraform_new>
```

Step 9: Check if the instance is destroyed or not.



*******End of Experiment-3******

LAB EXERCISE 4

Aim: Terraform Variables

Step 1: Create a main file & terraform configuration file for EC2 Instance (instance.tf)

```
terrafrom-variables
  .terraform\providers\registry.terraform.i...
    5.32.1
    5.34.0
  .terraform.lock.hcl
  index.html
  instance.tf
  main.tf
  variable.tf
```

```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "5.32.1"
    }
  }
}
provider "aws" {
region = "ap-south-1"
access_key = "AAKIA6GBMF5EUSILM7YFD"
secret_key = "O4bcqpe7uZr3BLP1rHtAIC769MzmA9fGhiFmpxch"
}
```

```
provider "aws" {
region = "ap-south-1"
access_key = "AAKIA6GBMF5EUSILM7YFD"
secret_key = "O4bcqpe7uZr3BLP1rHtAIC769MzmA9fGhiFmpxch"
}
resource "aws_instance" "My_instance" {
 ami = "ami-0d63de463e6604d0a"
 instance_type = "t2.micro"
 count= 1
 tags = {
   Name = "Exp4-instance"
 }
}
```

main.tf | variable.tf | instance.tf ● | index.html ●

terrafrom-variables > instance.tf > resource "aws_instance" "My_instance" > tags > Name

```
resource "aws_instance" "My_instance" {
  ami = var.ami
  instance_type = var.instance_type
  count = 1
  tags = {
    Name = "Exp4-instance"
  }
}
```

2. Open a new file named variables.tf. Define variables for region, ami, secret_key, access_key and instance_type.

```
terrafrom-variables > variable.tf > ...
variable "region" {
 type = string
 description = "AWS region"
 default = "ap-south-1"
}

variable "ami" {
 type = string
 description = "AMI ID"
 default = "ami-0d63de463e6604d0a"
}

variable "instance_type" {
 type = string
 description = "EC2 Instance Type"
 default = "t2.micro"
}
```

3. Modify main.tf & instance.tf to use the variables.

```
main.tf   variable.tf   instance.tf ●   index.html ●
terrafrom-variables > instance.tf > resource "aws_instance" "My_instance" > tags > Name
resource "aws_instance" "My_instance" {
  ami = var.ami
  instance_type = var.instance_type
  count = 1
  tags = {
    Name = "Exp4-instance"
  }
}
```

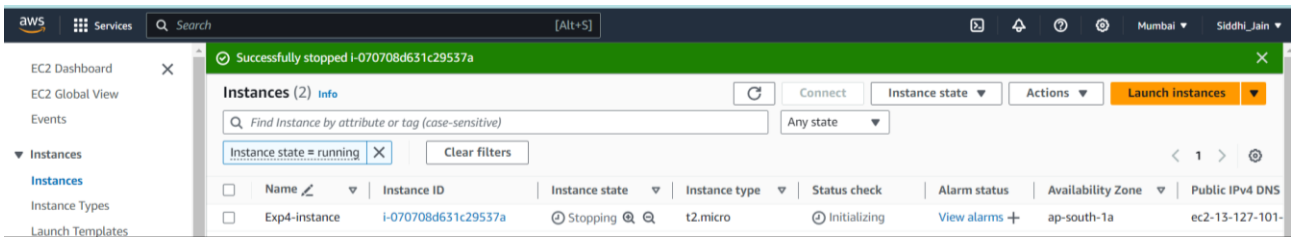
```
resource "aws_instance" "My_instance" {
 ami = var.ami
 instance_type = var.instance_type
 count= 1
 tags = {
   Name = "Exp4-instance"
 }
}
```

4. Run the following Terraform commands to initialize and apply the configuration.

```
C:\Users\jains\Desktop\terraform\terrafrom-variables>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_instance[0] will be created
  + resource "aws_instance" "My_instance" {
      + ami                                  = "ami-0d63de463e6604d0a"
      + 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)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (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)
```

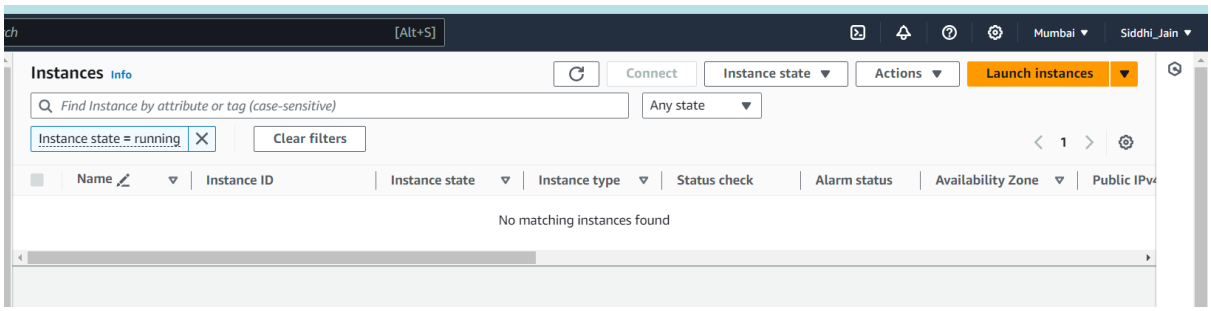
```
  Enter a value: yes

aws_instance.New_instance[0]: Creating...
aws_instance.My_instance[0]: Creating...
aws_instance.New_instance[0]: Still creating... [10s elapsed]
aws_instance.My_instance[0]: Still creating... [10s elapsed]
aws_instance.New_instance[0]: Still creating... [20s elapsed]
aws_instance.My_instance[0]: Still creating... [20s elapsed]
aws_instance.My_instance[0]: Still creating... [30s elapsed]
aws_instance.New_instance[0]: Still creating... [30s elapsed]
aws_instance.New_instance[0]: Creation complete after 32s [id=i-070708d631c29537a]
aws_instance.My_instance[0]: Creation complete after 32s [id=i-03d5976af87b936a9]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
```

5. Verify Resources



6. Cleanup Resources

```
C:\Users\jains\Desktop\terraform\terrafrom-variables>terraform destroy
aws_instance.New_instance[0]: Refreshing state... [id=i-070708d631c29537a]
aws_instance.My_instance[0]: Refreshing state... [id=i-03d5976af87b936a9]

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_instance[0] will be destroyed
  - resource "aws_instance" "My_instance" {
      - ami                                  = "ami-0d63de463e6604d0a" -> null
      - arn                                  = "arn:aws:ec2:ap-south-1:975050238249:instance/i-03d5976af87b936a9" -> 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-03d5976af87b936a9" -> 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
```



*****End of Experiment-4****