SPCM Lab-5

Objective : Terraform variables with commaned line arguments

1. Create Terraform directory.

```
vidhant@psyches-safehouse:~$ mkdir terraform-variables
vidhant@psyches-safehouse:~$ cd terraform-variables/
vidhant@psyches-safehouse:~/terraform-variables$
```

2. Create terraform configuration file (main.tf) :

main.tf 2 | variables.tf

Lab-4 > main.tf > ...

```
provider "aws" {
  region = var.region
  access_key = "AKIA232UVZYDK5TANG62"
  secret_key = "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"
}

resource "aws_instance" "vidhant-ec2" {
  ami = var.ami
  instance_type = var.instance_type
}
```

3. Define Variables : create a new file variables.tf and define variables for region, ami and instance_type.

main.tf | variables.tf

Lab-4 > variables.tf > variable "instance_type"

```
variable "region" {
  description = "AWS region"
  default = "us-east-1"
}

variable "ami" {
    description = "AMI ID"
    default = "ami-0277155c3f0ab2930"
}

variable "instance_type" {
  description = "EC2 instance type"
  default = "t2.micro"
}
```

4 . Initialize and Apply configuration : (terraform init, terraform validate and terraform apply)

```
vidhant@psyches-safehouse:~/Documents/Terraform/Lab-4$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.35.0...
```

```
vidhant@psyches-safehouse:~/Documents/Terraform/Lab-4$ terraform validate
Success! The configuration is valid.
```

Command : (terraform apply with cmd args)

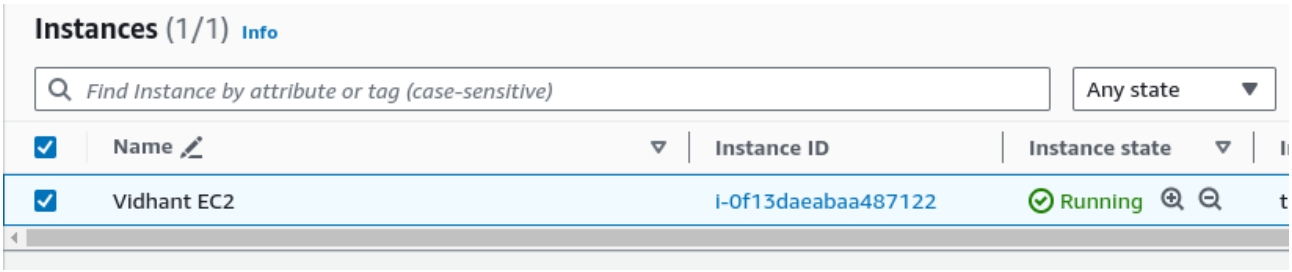
terraform apply -var 'region=us-east-1' -var 'ami=ami-12343432' -var 'instance_type=t2.micro'

```
vidhant@psyches-safehouse:~/Documents/Terraform/Lab-4$ terraform apply -var 'region=us-
east-1' -var 'ami=ami-0c7217cdde317cfec' -var 'instance_type=t2.micro'

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:
```

6. Checking aws console :



7. Cleaning up resources (terraform destroy)

```
vidhant@psyches-safehouse:~/Documents/Terraform/Lab-4$ terraform destroy
aws_instance.vidhant-ec2: Refreshing state... [id=i-0f13daeabaa487122]

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.vidhant-ec2 will be destroyed
  - resource "aws_instance" "vidhant-ec2" {
```