Import an existing VPC through Python

Task 1.4 → Using terraform import

Step:1 I mentioned the **main.tf** file for terraform configurations

```
provider "aws" {
  alias  = "us_east_1"
  region = "us-east-1"
}

provider "aws" {
  alias  = "us_east_2"
  region = "us-east-2"
}

resource "aws_vpc" "my_vpc_1" {
  provider   = aws.us_east_1
  cidr_block = "10.0.0.0/24"
}

resource "aws_vpc" "my_vpc_2" {
  provider   = aws.us_east_2
  cidr_block = "10.2.0.0/24"
}
```

Step:2 After running the **imp.py** file, it will automatically be initializing the terraform

```
import boto3
import subprocess

def get_vpcs():
    """Fetch all VPCs from all regions."""
    vpc_list = []
    ec2 = boto3.client('ec2')

    # Get all regions
    regions = ec2.describe_regions()['Regions']

```

```
PS C:\Users\Dell\OneDrive - Codin City\Desktop\Terra-Auto\1.5 py> python imp.py
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.69.0...
- Installed hashicorp/aws v5.69.0 (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.
```

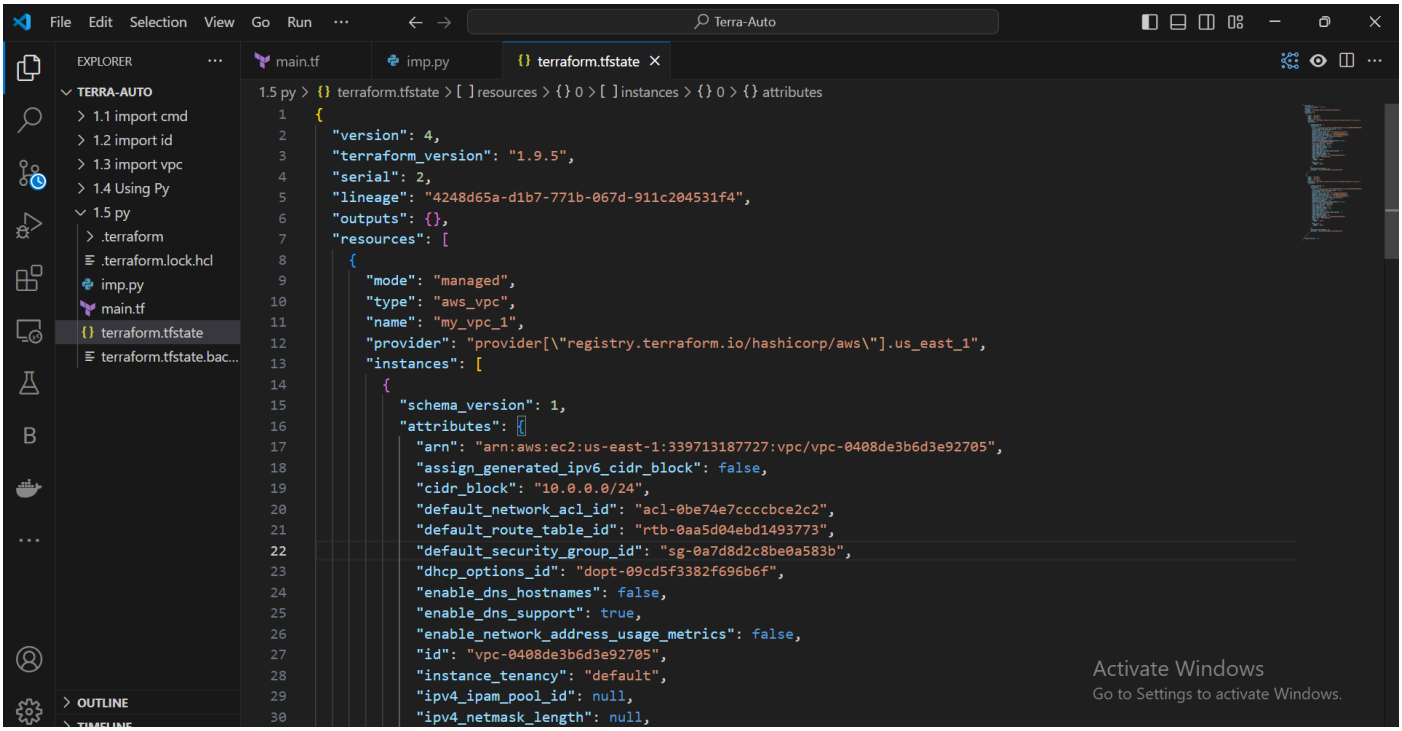
Step:3 It will list out the VPC at specified regions

```
Checking region: us-east-1
Found VPC: vpc-0408de3b6d3e92705 in region: us-east-1
Found VPC: vpc-0ac3883de5bde45b6 in region: us-east-1
Checking region: us-east-2
Found VPC: vpc-001cedaa66b3d63f6 in region: us-east-2
```

Step:4 After it all VPC has been importing to the statefile

```
aws_vpc.my_vpc_1: Importing from ID "vpc-0408de3b6d3e92705"...
aws_vpc.my_vpc_1: Import prepared!
  Prepared aws_vpc for import
aws_vpc.my_vpc_1: Refreshing state... [id=vpc-0408de3b6d3e92705]

Import successful!

The resources that were imported are shown above. These resources are now in
your Terraform state and will henceforth be managed by Terraform.

Successfully imported aws_vpc.my_vpc_1 with ID vpc-0408de3b6d3e92705.
Resource aws_vpc.my_vpc_1 is already managed by Terraform. Skipping import.
aws_vpc.my_vpc_2: Importing from ID "vpc-001cedaa66b3d63f6"...
aws_vpc.my_vpc_2: Import prepared!
  Prepared aws_vpc for import
aws_vpc.my_vpc_2: Refreshing state... [id=vpc-001cedaa66b3d63f6]

Import successful!

The resources that were imported are shown above. These resources are now in
Import successful!
```

Step:5 We can see the statefile all resource has been imported



Step:6 Also we can list out the state

```
PROBLEMS 5   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   SEARCH ERROR   AZURE

PS C:\Users\Dell\OneDrive - Codin City\Desktop\Terra-Auto\1.5 py> terraform state list
aws_vpc.my_vpc_1
aws_vpc.my_vpc_2
PS C:\Users\Dell\OneDrive - Codin City\Desktop\Terra-Auto\1.5 py>
```