SPCM-LAB-EXP3

1. Create a TerraForm Configuration File for EC2 instance (instance.tf).

instance.tf

```
resource "aws_instance" "Kanishka-ec2"{
    instance_type = "t2.micro"
    ami="ami-0014ce3e52359afbd"
    count=1
    tags={
        Name="Kanishka-EC2-Instance"
    }
}
```

Initialize: terraform init

```
Terraform-SPCM-LAB  terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.31.0

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.
```

Validate: terraform validate

```
→  Terraform-SPCM-LAB  terraform validate
Success! The configuration is valid.

→  Terraform-SPCM-LAB
```

Review Plan : terraform plan

```
→  Terraform-SPCM-LAB  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.Kanishka-ec2[0] will be created
  + resource "aws_instance" "Kanishka-ec2" {
      + ami                                  = "ami-0014ce3e52359afbd"
      + 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"
```

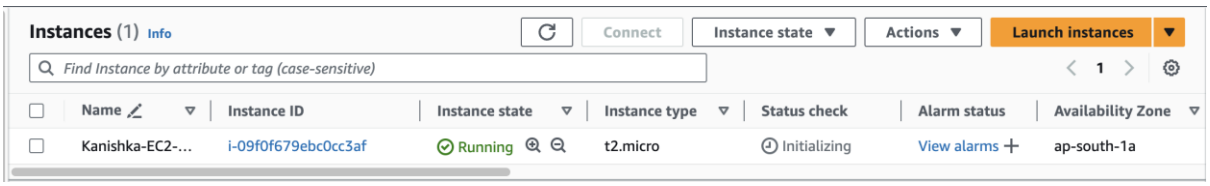
Apply Changes: terraform apply

```
  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]: Still creating... [30s elapsed]
aws_instance.My-instance[0]: Creation complete after 32s [id=i-09f0f679ebc0cc3af]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
→  Terraform-SPCM-LAB
```

Verify Resource Creation.



CleanUp Resources: terraform destroy

```
  Enter a value: yes

aws_instance.My-instance[0]: Destroying... [id=i-09f0f679ebc0cc3af]
aws_instance.My-instance[0]: Still destroying... [id=i-09f0f679ebc0cc3af, 10s elapsed]
aws_instance.My-instance[0]: Still destroying... [id=i-09f0f679ebc0cc3af, 20s elapsed]
aws_instance.My-instance[0]: Still destroying... [id=i-09f0f679ebc0cc3af, 30s elapsed]
aws_instance.My-instance[0]: Destruction complete after 30s

Destroy complete! Resources: 1 destroyed.
➜  Terraform-SPCM-LAB
```

Verify Resource Deletion