Lesson 08 Demo 03

Validating Terraform Configuration File

Objective: To validate a Terraform configuration file using the **terraform validate** command

Tools required: Terraform and a Terraform script (.tf file)

Prerequisites: You need to have Terraform installed in your lab environment. This demo is incremental. Please ensure you have successfully executed Demo 02 of lesson 08 before proceeding with this demo.

Steps to be followed:

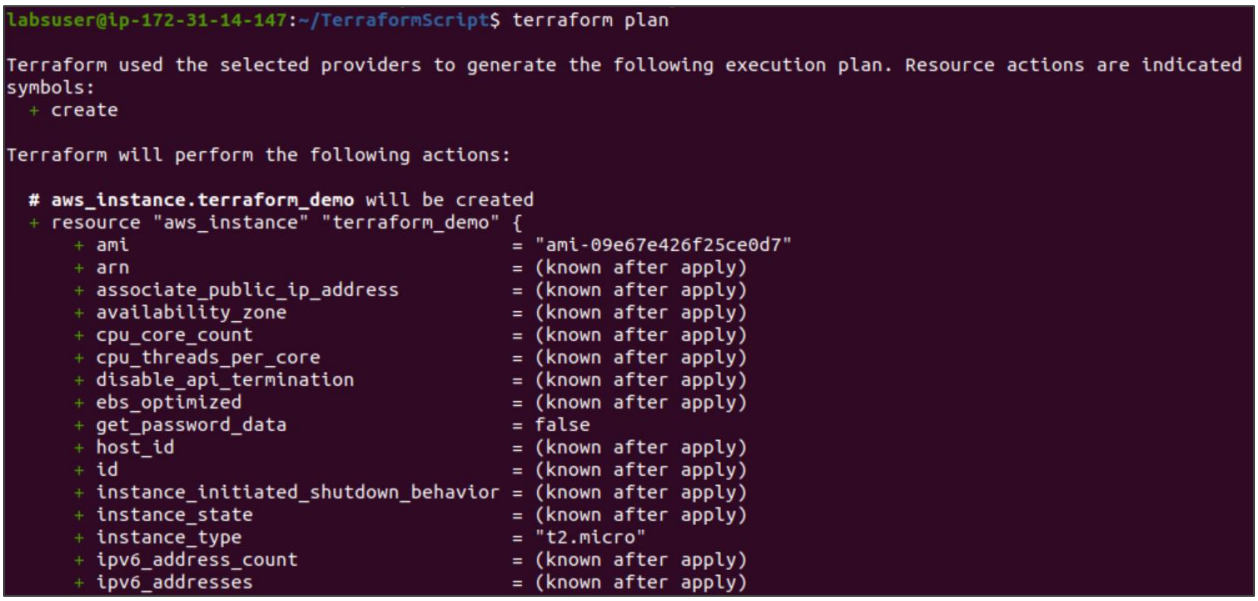
1. Validate the Terraform script

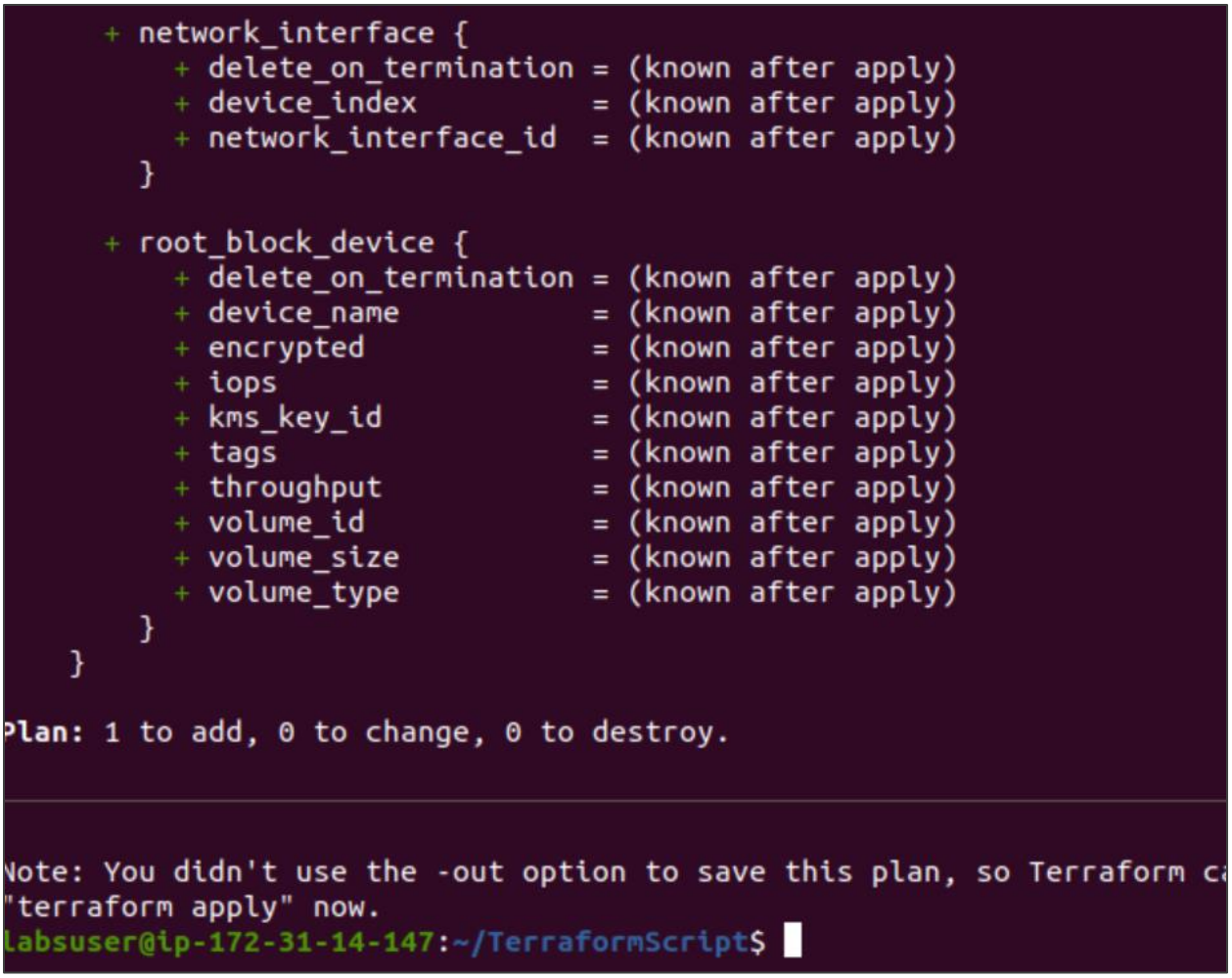
Step 1: Validate the Terraform script

1.1 Use the following command to initialize Terraform:
terraform init

```
labsuser@ip-172-31-14-147:~/TerraformScript$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v3.73.0...
- Installed hashicorp/aws v3.73.0 (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.

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.
labsuser@ip-172-31-14-147:~/TerraformScript$
```

1.2 Proceed to the planning stage, which generates the execution plan for creating and provisioning the infrastructure. Run the following command:
terraform plan




+ network_interface {
+ delete_on_termination = (known after apply)
+ device_index = (known after apply)
+ network_interface_id = (known after apply)
}
+ root_block_device {
+ delete_on_termination = (known after apply)
+ device_name = (known after apply)
+ encrypted = (known after apply)
+ iops = (known after apply)
+ kms_key_id = (known after apply)
+ tags = (known after apply)
+ throughput = (known after apply)
+ volume_id = (known after apply)
+ volume_size = (known after apply)
+ volume_type = (known after apply)
}
}
Plan: 1 to add, 0 to change, 0 to destroy.
Note: You didn't use the -out option to save this plan, so Terraform ca
"terraform apply" now.
labsuser@ip-172-31-14-147:~/TerraformScript$

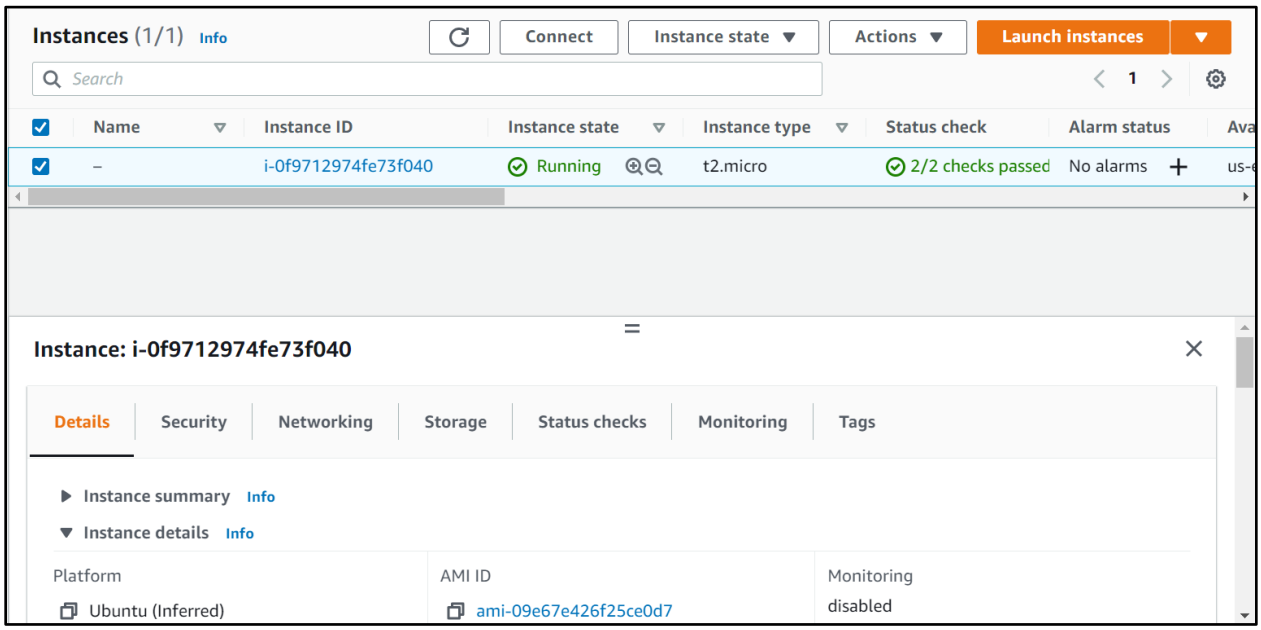
1.3 Move to the apply stage, which will execute the configuration file and launch an AWS EC2 instance. Run the following command:
terraform apply

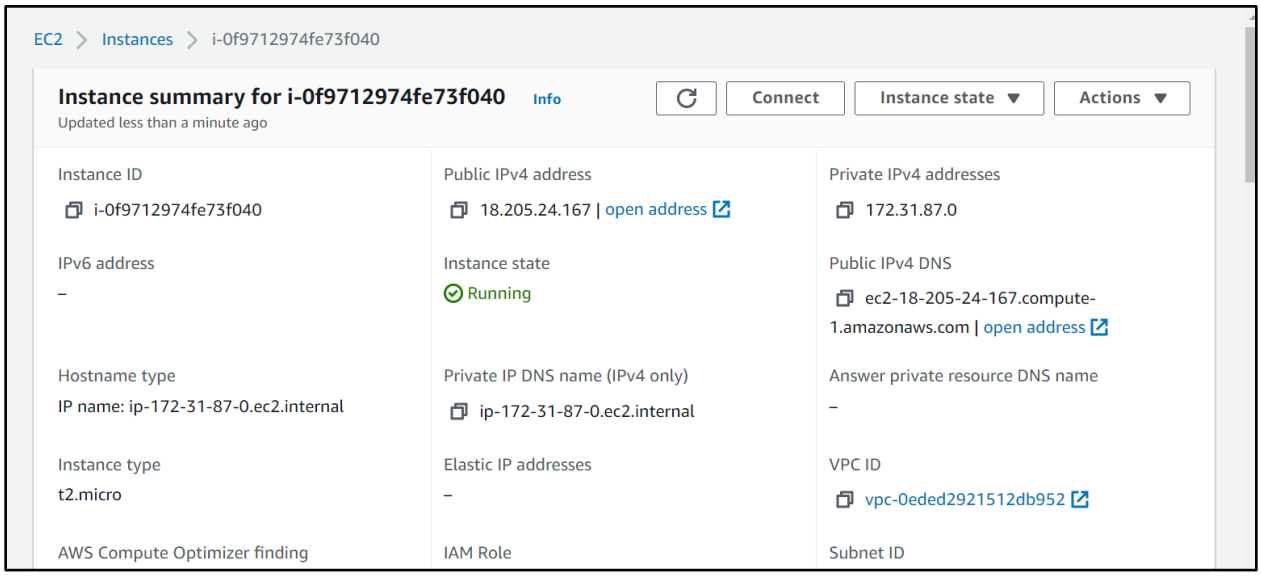
```
labsuser@ip-172-31-14-147:~/TerraformScript$ terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions
symbols:
  + create

Terraform will perform the following actions:

  # aws_instance.terraform_demo will be created
  + resource "aws_instance" "terraform_demo" {
      + ami                                  = "ami-09e67e426f25ce0d7"
      + 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_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
```

1.4 When you run the apply command, it will ask, “Do you want to perform these actions?” You need to type **yes** and press Enter

```
          + encrypted             = (known after apply)
          + iops                  = (known after apply)
          + kms_key_id            = (known after apply)
          + tags                  = (known after apply)
          + throughput            = (known after apply)
          + volume_id             = (known after apply)
          + volume_size           = (known after apply)
          + volume_type           = (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.terraform_demo: Creating...
aws_instance.terraform_demo: Still creating... [10s elapsed]
aws_instance.terraform_demo: Still creating... [20s elapsed]
aws_instance.terraform_demo: Still creating... [30s elapsed]
aws_instance.terraform_demo: Creation complete after 37s [id=i-0f9712974fe73f040]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
labsuser@ip-172-31-14-147:~/TerraformScript$
```

1.5 Navigate to the AWS EC2 dashboard, and you will see a new instance with the instance ID mentioned at the end of the **apply** command that has been created





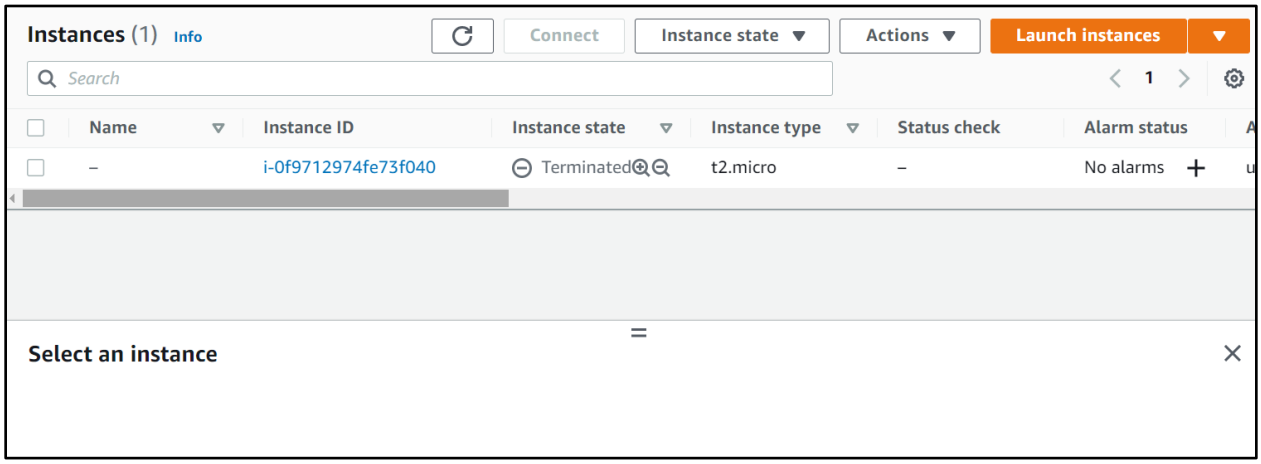
You have successfully launched an AWS EC2 instance using Terraform.

1.6 Finally, if you want to delete the instance, you need to run the destroy command:
terraform destroy

```
labsuser@ip-172-31-14-147:~/TerraformScript$ terraform destroy
aws_instance.terraform_demo: Refreshing state... [id=i-0f9712974fe73f040]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated
symbols:
  - destroy

Terraform will perform the following actions:

  # aws_instance.terraform_demo will be destroyed
  - resource "aws_instance" "terraform_demo" {
      - ami                                  = "ami-09e67e426f25ce0d7" -> null
      - arn                                  = "arn:aws:ec2:us-east-1:316828699587:instance/i-0f9712974fe73f04
      - associate_public_ip_address          = true -> null
      - availability_zone                    = "us-east-1c" -> null
      - cpu_core_count                       = 1 -> null
      - cpu_threads_per_core                 = 1 -> null
      - disable_api_termination              = false -> null
      - ebs_optimized                        = false -> null
      - get_password_data                    = false -> null
      - hibernation                          = false -> null
```

```
          - encrypted             = false -> null
          - iops                  = 100 -> null
          - tags                  = {} -> null
          - throughput            = 0 -> null
          - volume_id             = "vol-04a34748156593223" -> 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.terraform_demo: Destroying... [id=i-0f9712974fe73f040]
aws_instance.terraform_demo: Still destroying... [id=i-0f9712974fe73f040, 10s elapsed]
aws_instance.terraform_demo: Still destroying... [id=i-0f9712974fe73f040, 20s elapsed]
aws_instance.terraform_demo: Still destroying... [id=i-0f9712974fe73f040, 30s elapsed]
aws_instance.terraform_demo: Still destroying... [id=i-0f9712974fe73f040, 40s elapsed]
aws_instance.terraform_demo: Destruction complete after 42s

Destroy complete! Resources: 1 destroyed.
labsuser@ip-172-31-14-147:~/TerraformScript$
```

1.7 Now, recheck the EC2 dashboard, and you can see that the instance was terminated



By following these steps, you have successfully validated the Terraform configuration file, launched an AWS EC2 instance, and efficiently managed its lifecycle, including termination.