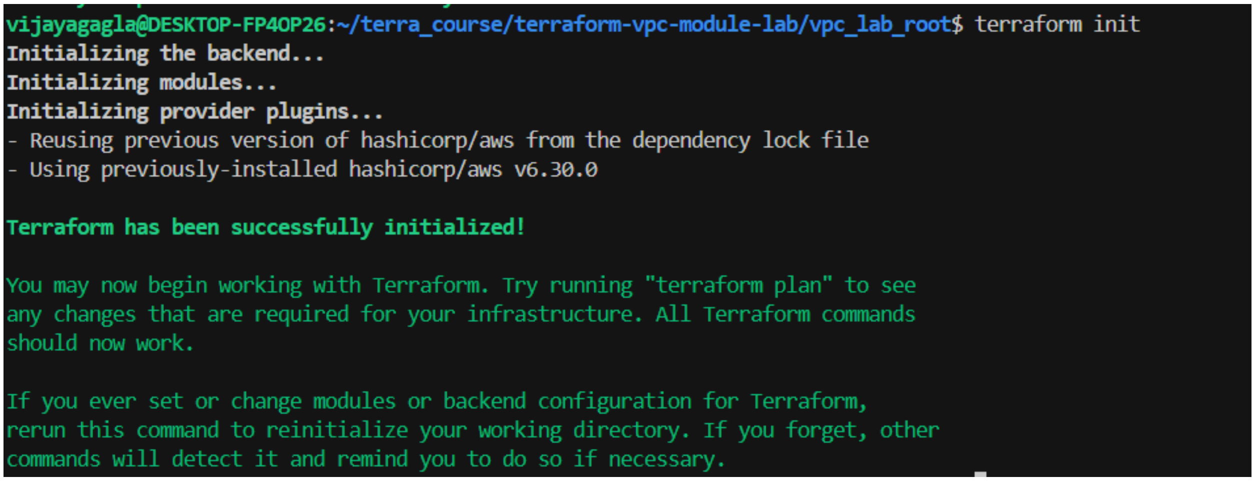
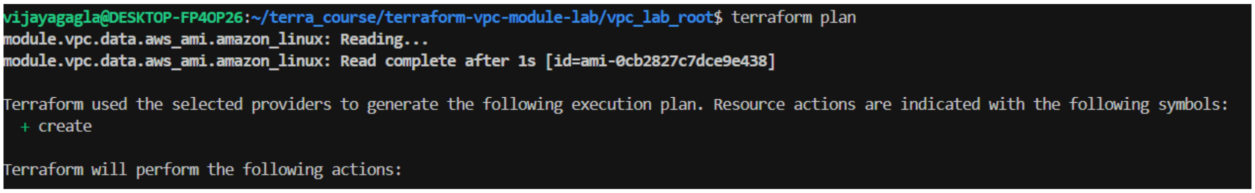
# LAB | Build a VPC as a Terraform Module

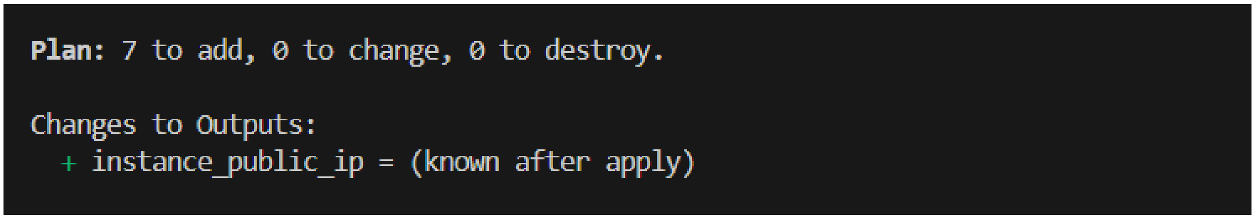
Terraform command outputs - Created fresh all the resources.



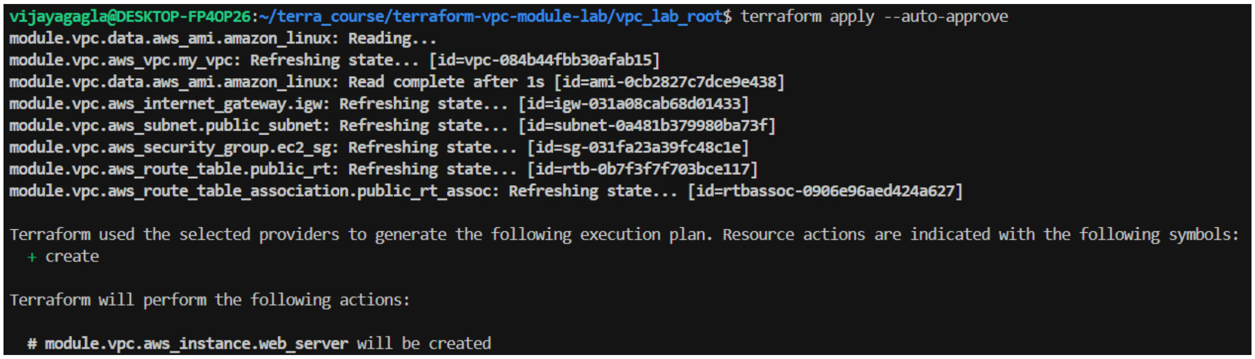
```
vijayagagla@DESKTOP-FP4OP26:~/terra_course/terraform-vpc-module-lab/vpc_lab_root$ terraform init
Initializing the backend...
Initializing modules...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v6.30.0

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.
```



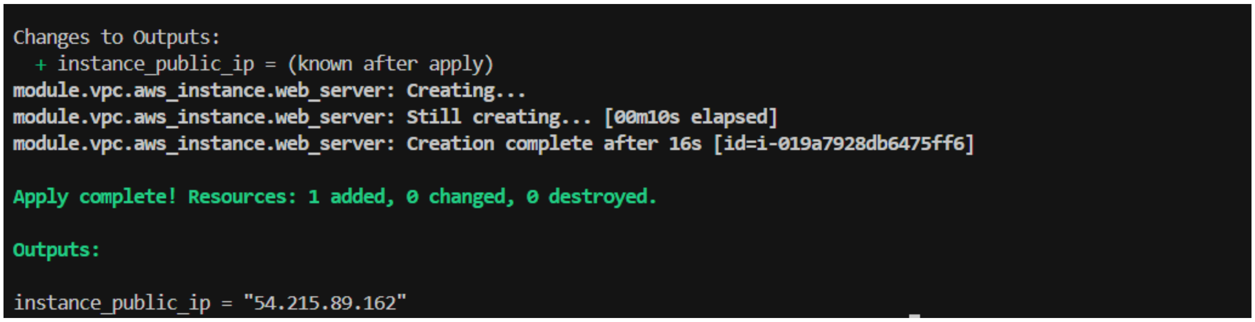
```
vijayagagla@DESKTOP-FP4OP26:~/terra_course/terraform-vpc-module-lab/vpc_lab_root$ terraform plan
module.vpc.data.aws_ami.amazon_linux: Reading...
module.vpc.data.aws_ami.amazon_linux: Read complete after 1s [id=ami-0cb2827c7dce9e438]

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:
```



```
Plan: 7 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + instance_public_ip = (known after apply)
```

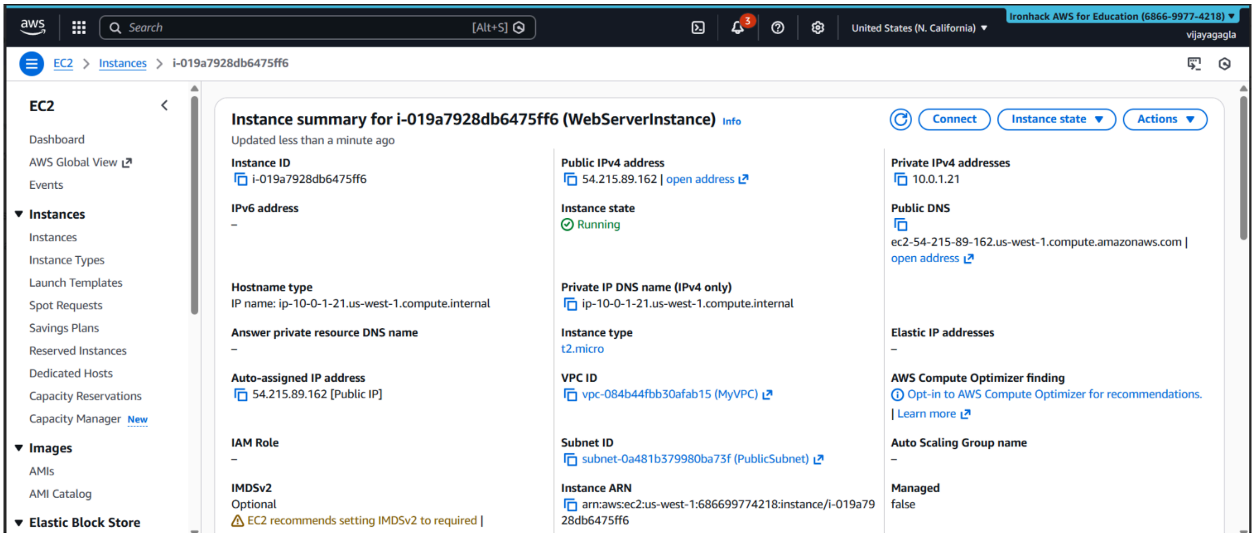


```
vijayagagla@DESKTOP-FP4OP26:~/terra_course/terraform-vpc-module-lab/vpc_lab_root$ terraform apply --auto-approve
module.vpc.data.aws_ami.amazon_linux: Reading...
module.vpc.aws_vpc.my_vpc: Refreshing state... [id=vpc-084b44fbb30afab15]
module.vpc.data.aws_ami.amazon_linux: Read complete after 1s [id=ami-0cb2827c7dce9e438]
module.vpc.aws_internet_gateway.igw: Refreshing state... [id=igw-031a08cab68d01433]
module.vpc.aws_subnet.public_subnet: Refreshing state... [id=subnet-0a481b379980ba73f]
module.vpc.aws_security_group.ec2_sg: Refreshing state... [id=sg-031fa23a39fc48c1e]
module.vpc.aws_route_table.public_rt: Refreshing state... [id=rtb-0b7f3f7f703bce117]
module.vpc.aws_route_table_association.public_rt_assoc: Refreshing state... [id=rtbassoc-0906e96aed424a627]

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:

  # module.vpc.aws_instance.web_server will be created
```



```
Changes to Outputs:
  + instance_public_ip = (known after apply)
module.vpc.aws_instance.web_server: Creating...
module.vpc.aws_instance.web_server: Still creating... [00m10s elapsed]
module.vpc.aws_instance.web_server: Creation complete after 16s [id=i-019a7928db6475ff6]

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

Outputs:

instance_public_ip = "54.215.89.162"
```

AWS Console confirming the VPC and EC2 instance



**Instance summary for i-019a7928db6475ff6 (WebServerInstance)** Info

Updated less than a minute ago

**Instance ID**
i-019a7928db6475ff6

**Public IPv4 address**
54.215.89.162 | open address

**Private IPv4 addresses**
10.0.1.21

**IPv6 address**
–

**Instance state**
✓ Running

**Public DNS**
ec2-54-215-89-162.us-west-1.compute.amazonaws.com | open address

**Hostname type**
IP name: ip-10-0-1-21.us-west-1.compute.internal

**Private IP DNS name (IPv4 only)**
ip-10-0-1-21.us-west-1.compute.internal

**Answer private resource DNS name**
–

**Instance type**
t2.micro

**Elastic IP addresses**
–

**Auto-assigned IP address**
54.215.89.162 [Public IP]

**VPC ID**
vpc-084b44fbb30afab15 (MyVPC)

**AWS Compute Optimizer finding**
Opt-in to AWS Compute Optimizer for recommendations. | Learn more

**IAM Role**
–

**Subnet ID**
subnet-0a481b379980ba73f (PublicSubnet)

**Auto Scaling Group name**
–

**IMDSv2**
Optional
⚠ EC2 recommends setting IMDSv2 to required |

**Instance ARN**
arn:aws:ec2:us-west-1:686699774218:instance/i-019a7928db6475ff6

**Managed**
false