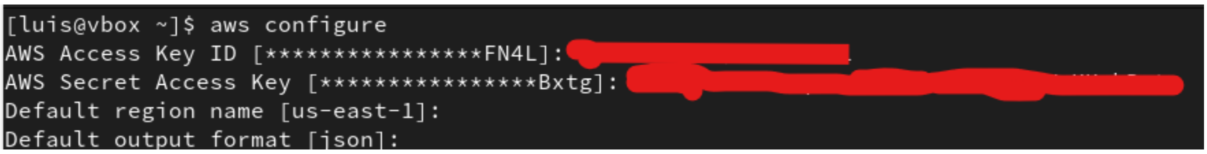
TERRAFORM BASIC EC2 INSTANCE CREATION.

- **Prerequisites**
  - Linux distribution (CentOS 9 in this case)
  - AWS CLI installed
  - Terraform installed
  - Root account in AWS platform

- **Creation of EC2 instance through Terraform**

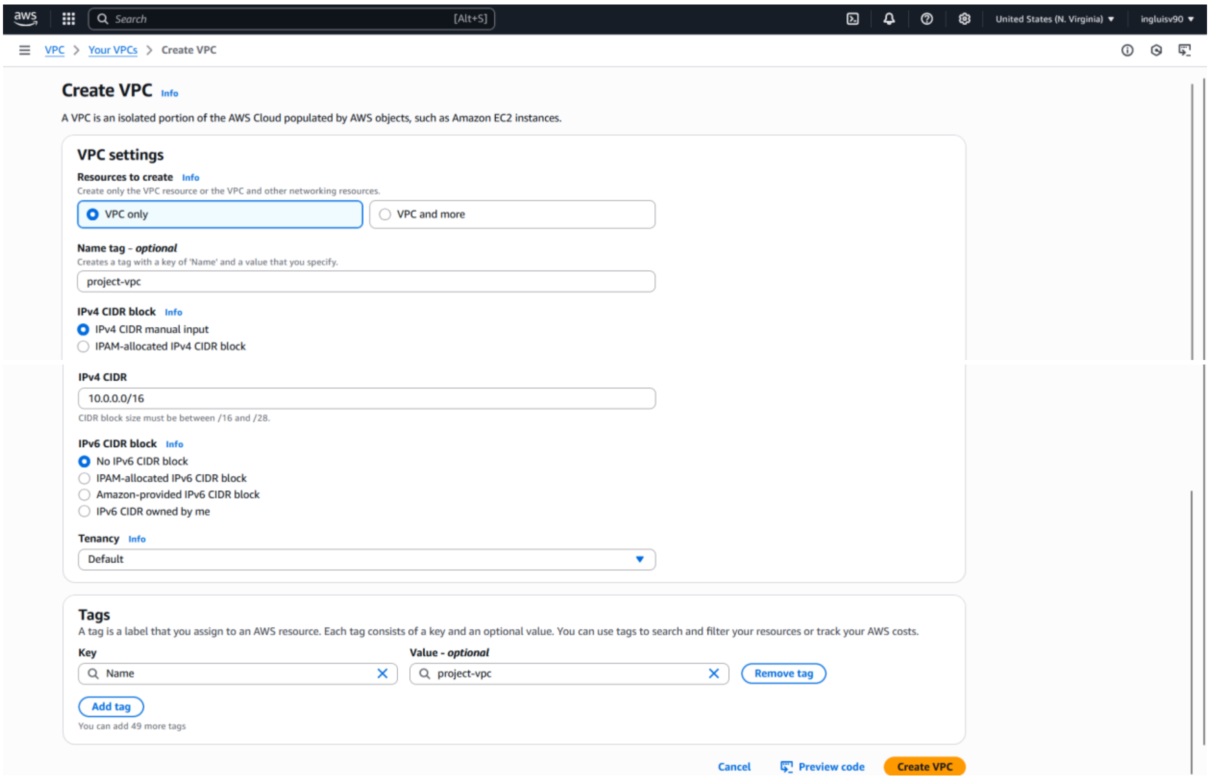
1. **Configuring AWS:** Setting up the connection with AWS console execute the next command:

aws configure



```
[luis@vbox ~]$ aws configure
AWS Access Key ID [****************FN4L]: [redacted]
AWS Secret Access Key [****************Bxtg]: [redacted]
Default region name [us-east-1]:
Default output format [json]:
```

2. **VPC creation:** this step is necessary in order to create an AWS private space called Virtual Private Cloud where our instances can be deployed.
   - Login with root account in the AWS Console
   - Search for VPC -> Create VPC
   - Provide the details as follow and hit Create VPC:



Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

Resources to create Info
Create only the VPC resource or the VPC and other networking resources.
● VPC only ○ VPC and more

Name tag - *optional*
Creates a tag with a key of 'Name' and a value that you specify.
project-vpc

IPv4 CIDR block Info
● IPv4 CIDR manual input
○ IPAM-allocated IPv4 CIDR block

IPv4 CIDR
10.0.0.0/16
CIDR block size must be between /16 and /28.

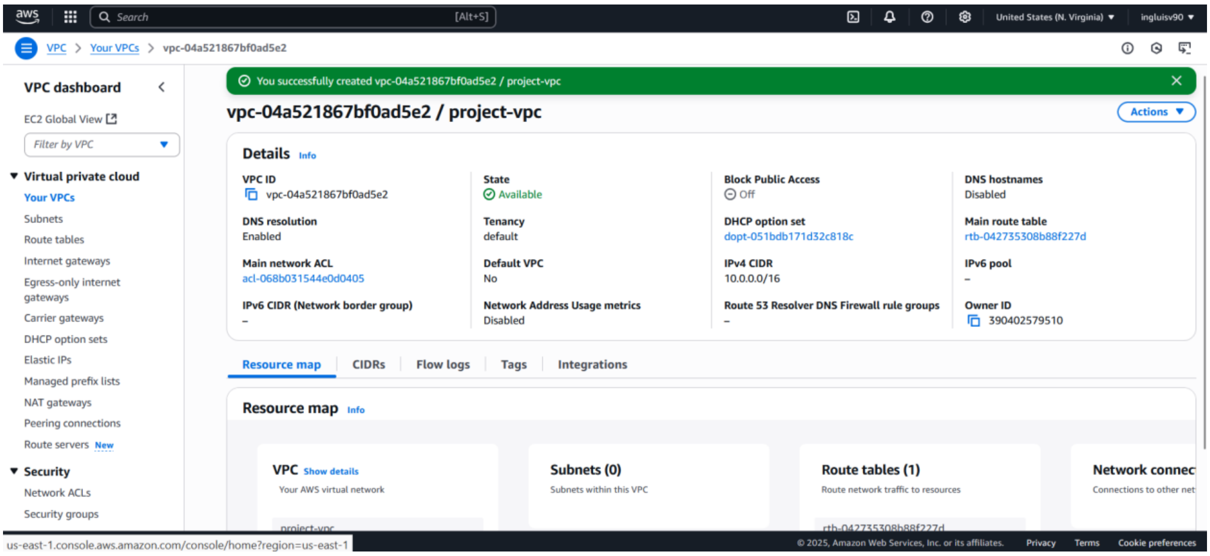
IPv6 CIDR block Info
● No IPv6 CIDR block
○ IPAM-allocated IPv6 CIDR block
○ Amazon-provided IPv6 CIDR block
○ IPv6 CIDR owned by me

Tenancy Info
Default ▾

Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - *optional*
Name	project-vpc

- Now the VPC is created as vpc-04a521867bf0ad5e2 / project-vpc



You successfully created vpc-04a521867bf0ad5e2 / project-vpc

vpc-04a521867bf0ad5e2 / project-vpc

Details Info

VPC ID vpc-04a521867bf0ad5e2	State Available	Block Public Access Off	DNS hostnames Disabled
DNS resolution Enabled	Tenancy default	DHCP option set dopt-051bdb171d32c818c	Main route table rtb-042735308b88f227d
Main network ACL acl-068b031544e0d0405	Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool –
IPv6 CIDR (Network border group) –	Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups –	Owner ID 390402579510

Resource map Info

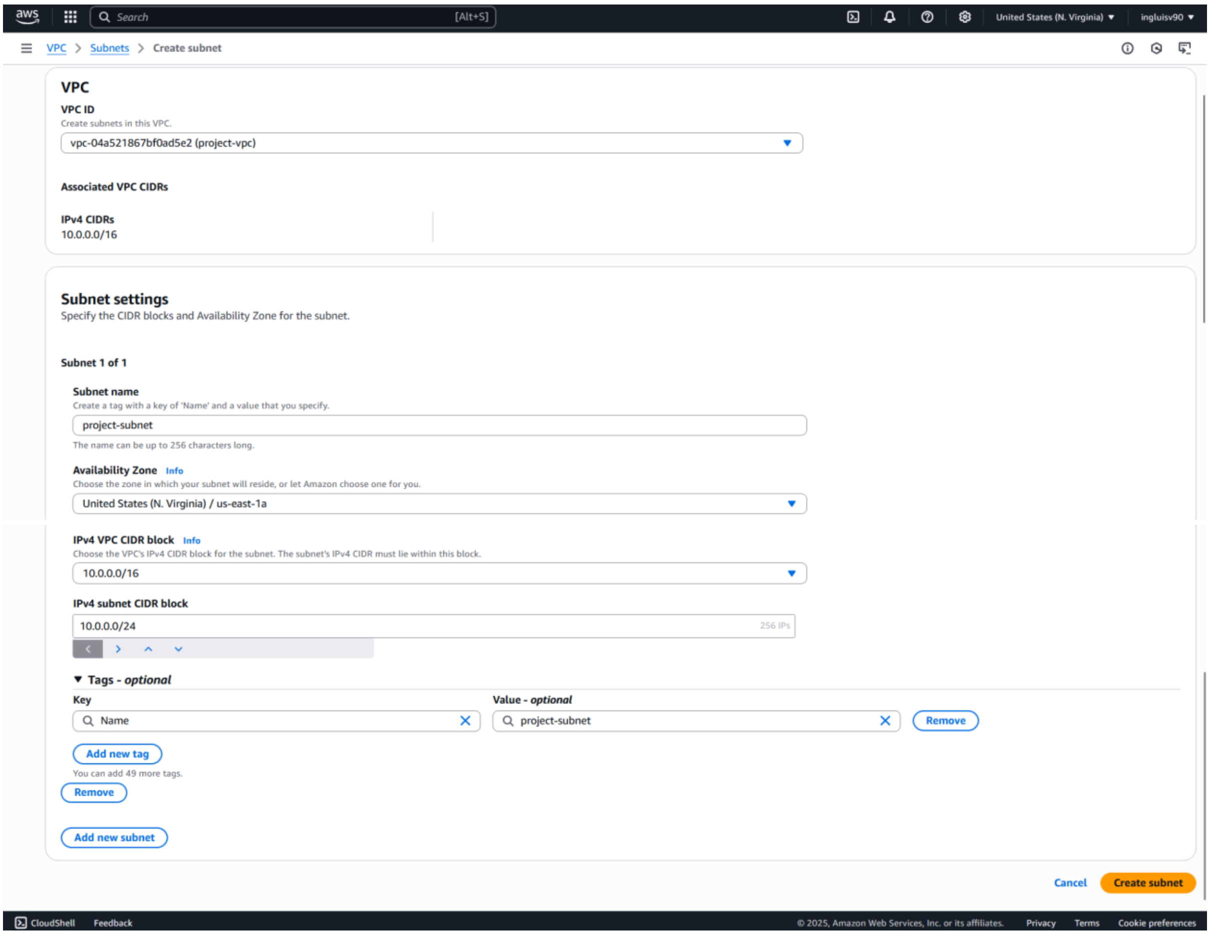
VPC Show details
Your AWS virtual network
project-vpc

Subnets (0)
Subnets within this VPC

Route tables (1)
Route network traffic to resources
rtb-042735308b88f227d

Network connec
Connections to other net

3. **Subnet creation:** It's necessary to create a new subnet in order to provide communication between the objects inside the VPC
    - In the VPC dashboard -> click on Subnets
    - Provide the information as follow and hit Create Subnet



VPC

VPC ID
Create subnets in this VPC.
vpc-04a521867bf0ad5e2 (project-vpc)

Associated VPC CIDRs

IPv4 CIDRs
10.0.0.0/16

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
project-subnet
The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
United States (N. Virginia) / us-east-1a

IPv4 VPC CIDR block Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
10.0.0.0/16

IPv4 subnet CIDR block
10.0.0.0/24 256 IPs

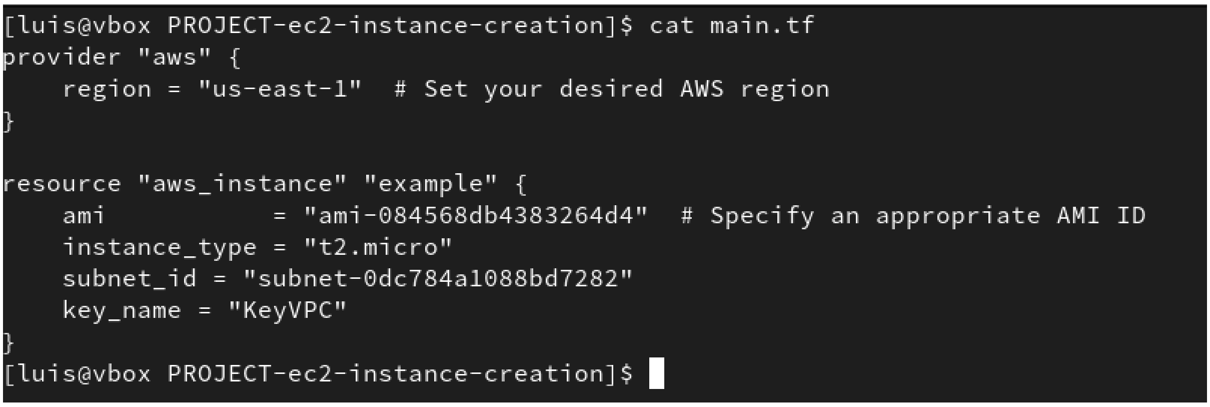
▾ Tags - *optional*

Key	Value - *optional*
Name	project-subnet

You can add 49 more tags.

4. **Terraform file:** Back in our CentOS environment, we are in the folder containing our main.tf file, which is responsible for communicating our commands to Terraform

    - File main.tf:



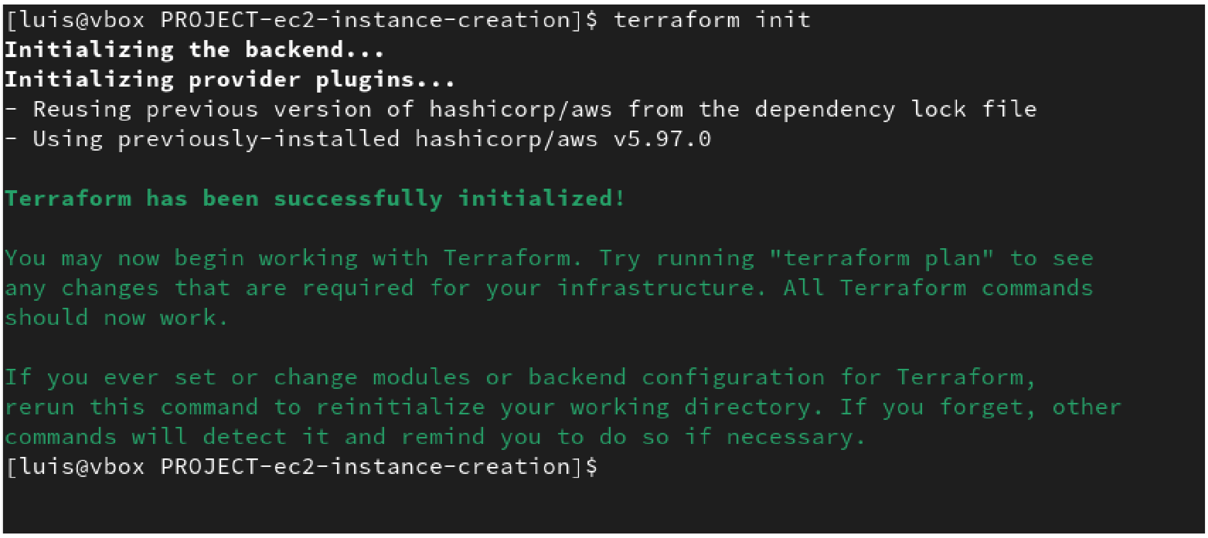
```
[luis@vbox PROJECT-ec2-instance-creation]$ cat main.tf
provider "aws" {
    region = "us-east-1"  # Set your desired AWS region
}

resource "aws_instance" "example" {
    ami           = "ami-084568db4383264d4"  # Specify an appropriate AMI ID
    instance_type = "t2.micro"
    subnet_id = "subnet-0dc784a1088bd7282"
    key_name = "KeyVPC"
}
[luis@vbox PROJECT-ec2-instance-creation]$ 
```

Here is the content of the Terraform file that we will use:

- The first lines declare the provider as “aws” and the desired region as “us-east-1”.
- After that, we must specify the kind of resource that will be created, the name, AMI ID, instance type (t2.micro [free use]), subnet, and a key name.

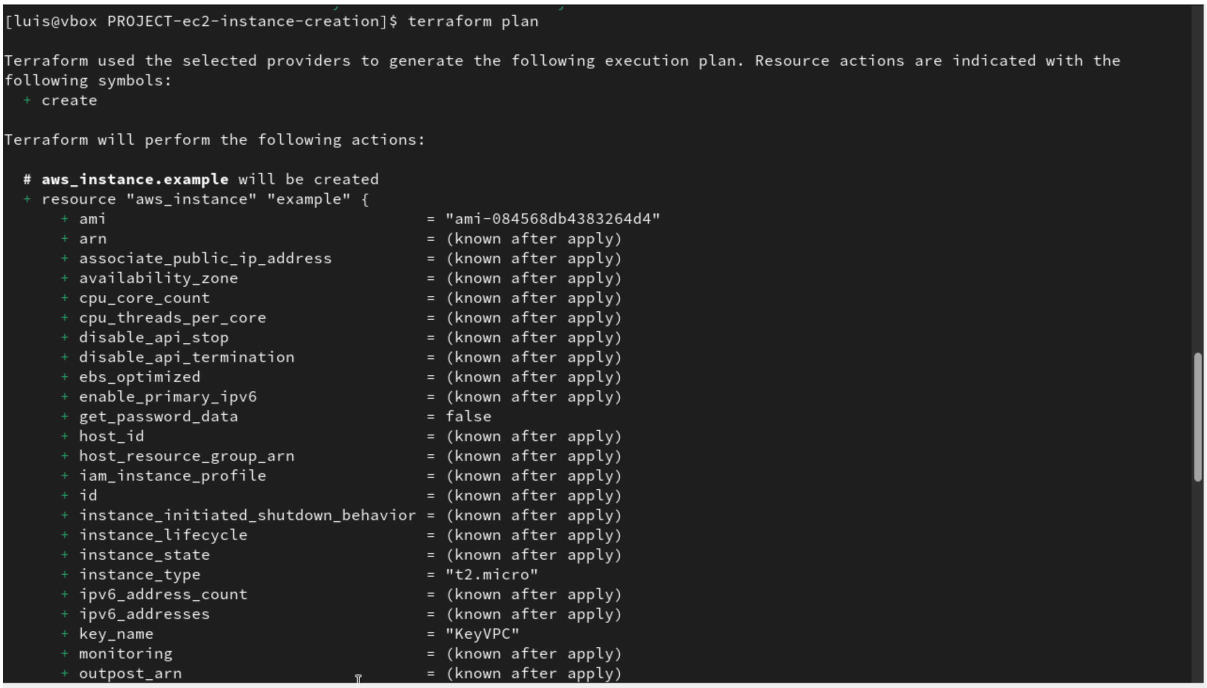
In this case, we will create an EC2 instance inside our VPC using the Subnet created.

5. **Terraform init:** Initializes a working directory and downloads the necessary provider plugins and modules, and sets up the backend for storing your infrastructure's state.

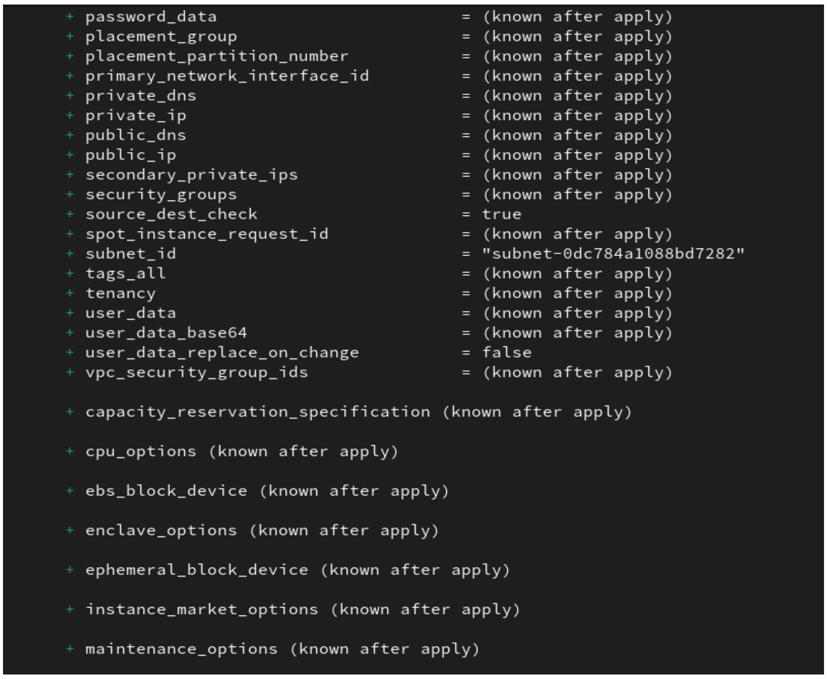


```
[luis@vbox PROJECT-ec2-instance-creation]$ terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.97.0

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.
[luis@vbox PROJECT-ec2-instance-creation]$
```

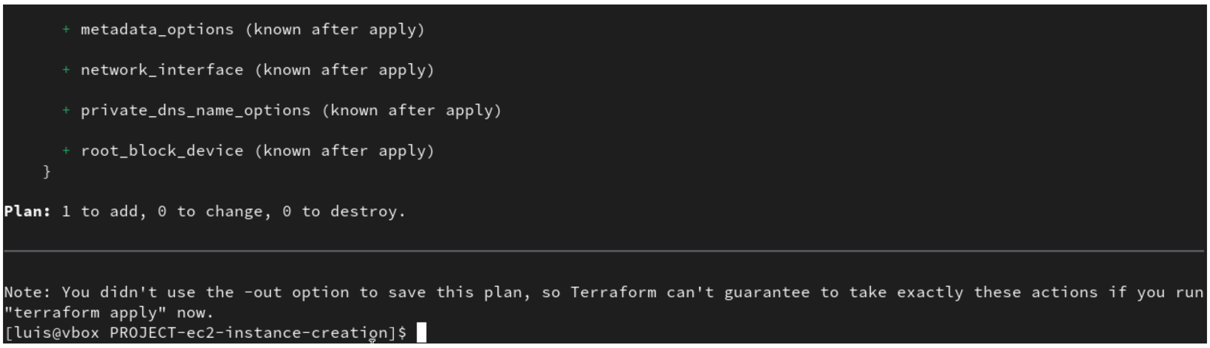
6. **Terraform plan:** It will generate an execution plan, showing us what actions will be taken without actually performing the planned actions.



```
[luis@vbox PROJECT-ec2-instance-creation]$ 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.example will be created
  + resource "aws_instance" "example" {
      + ami                                  = "ami-084568db4383264d4"
      + 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)
      + enable_primary_ipv6                  = (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"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = "KeyVPC"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
```

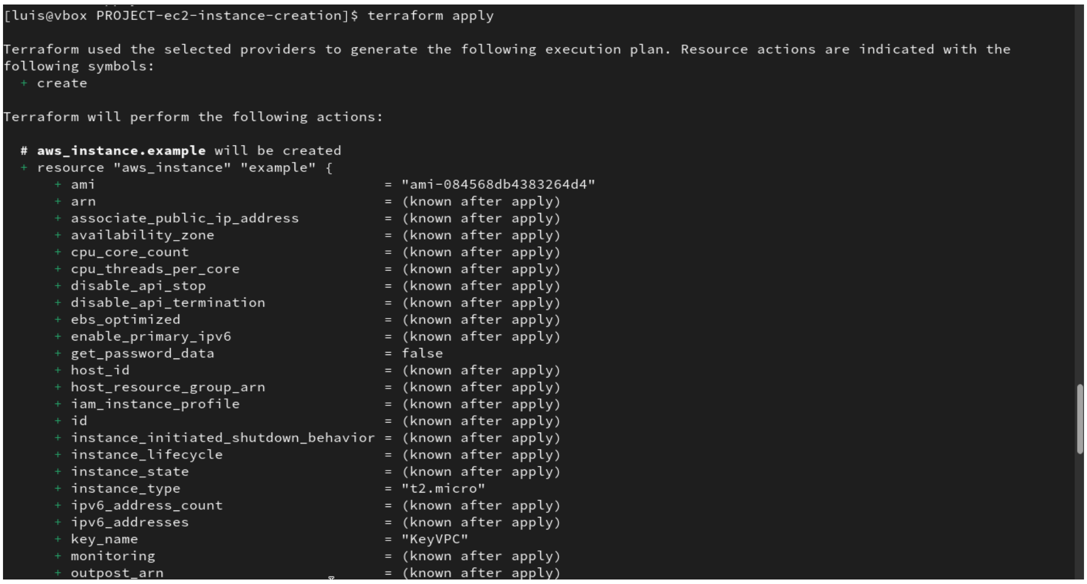


```
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = "subnet-0dc784a1088bd7282"
      + tags_all                             = (known after apply)
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)

      + capacity_reservation_specification (known after apply)

      + cpu_options (known after apply)

      + ebs_block_device (known after apply)

      + enclave_options (known after apply)

      + ephemeral_block_device (known after apply)

      + instance_market_options (known after apply)

      + maintenance_options (known after apply)
```

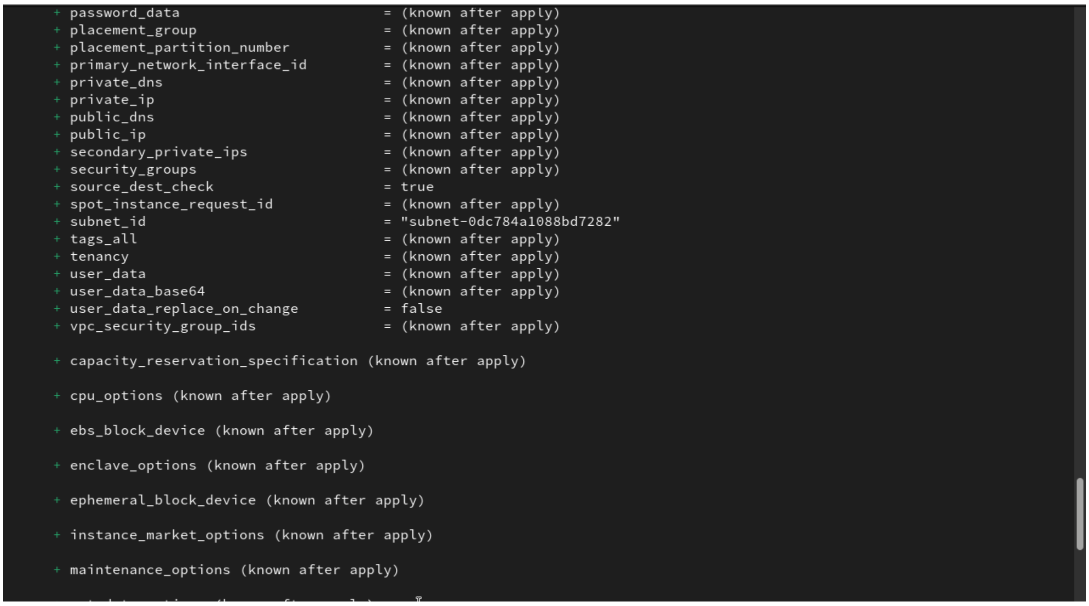


```
      + metadata_options (known after apply)

      + network_interface (known after apply)

      + private_dns_name_options (known after apply)

      + root_block_device (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 can't guarantee to take exactly these actions if you run
"terraform apply" now.
[luis@vbox PROJECT-ec2-instance-creation]$ 
```

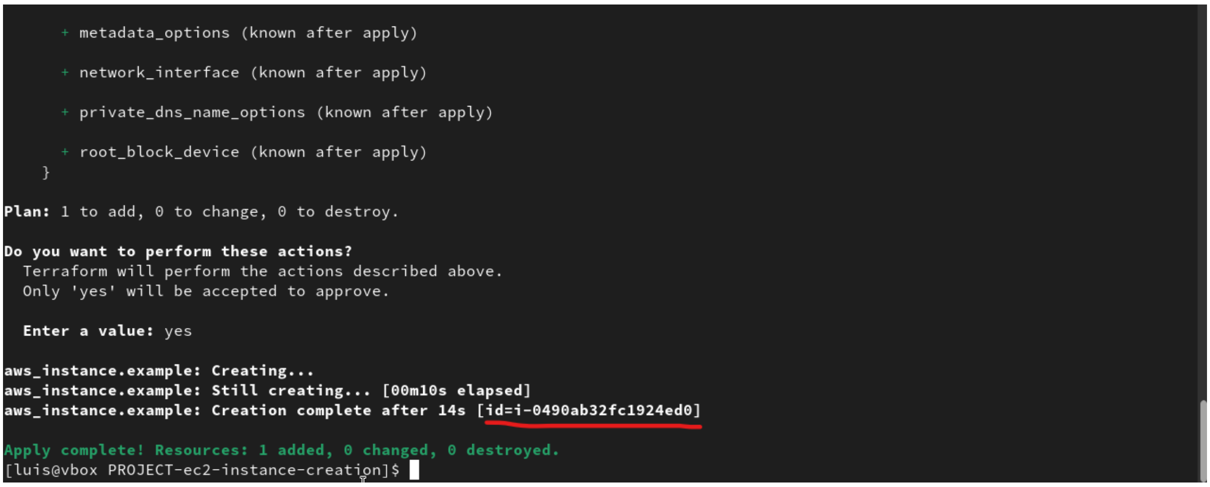
7. **Terraform apply:** Create or update infrastructure depending on the configuration files. By default, a plan will be generated first and will need to be approved before it is applied.



```
[luis@vbox PROJECT-ec2-instance-creation]$ terraform apply

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.example will be created
  + resource "aws_instance" "example" {
      + ami                                  = "ami-084568db4383264d4"
      + 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)
      + enable_primary_ipv6                  = (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"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = "KeyVPC"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
```

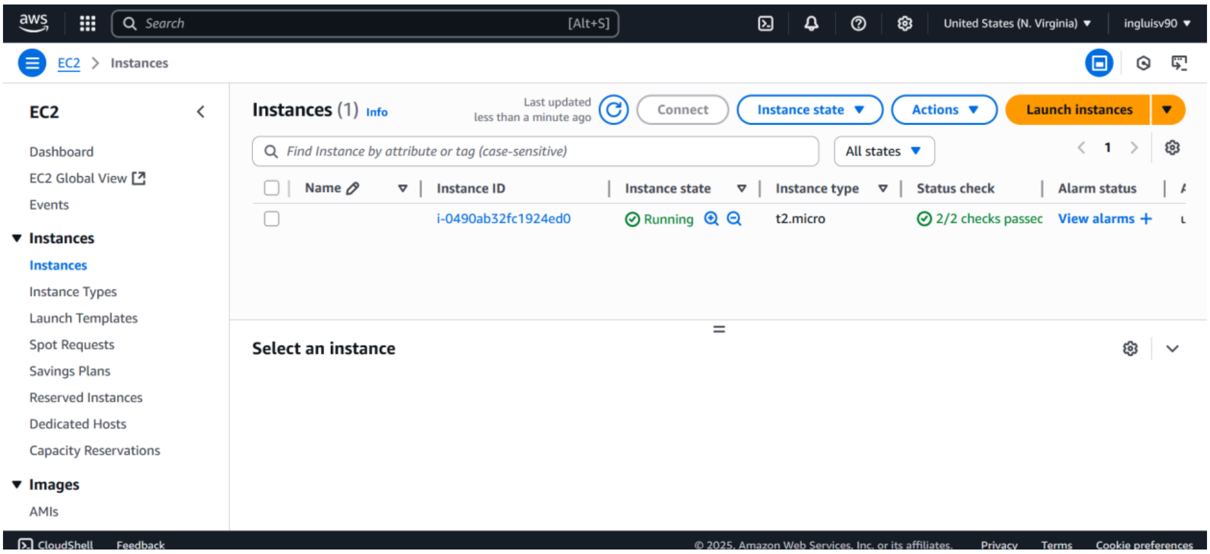


```
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = "subnet-0dc784a1088bd7282"
      + tags_all                             = (known after apply)
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)

      + capacity_reservation_specification (known after apply)

      + cpu_options (known after apply)

      + ebs_block_device (known after apply)

      + enclave_options (known after apply)

      + ephemeral_block_device (known after apply)

      + instance_market_options (known after apply)

      + maintenance_options (known after apply)
```



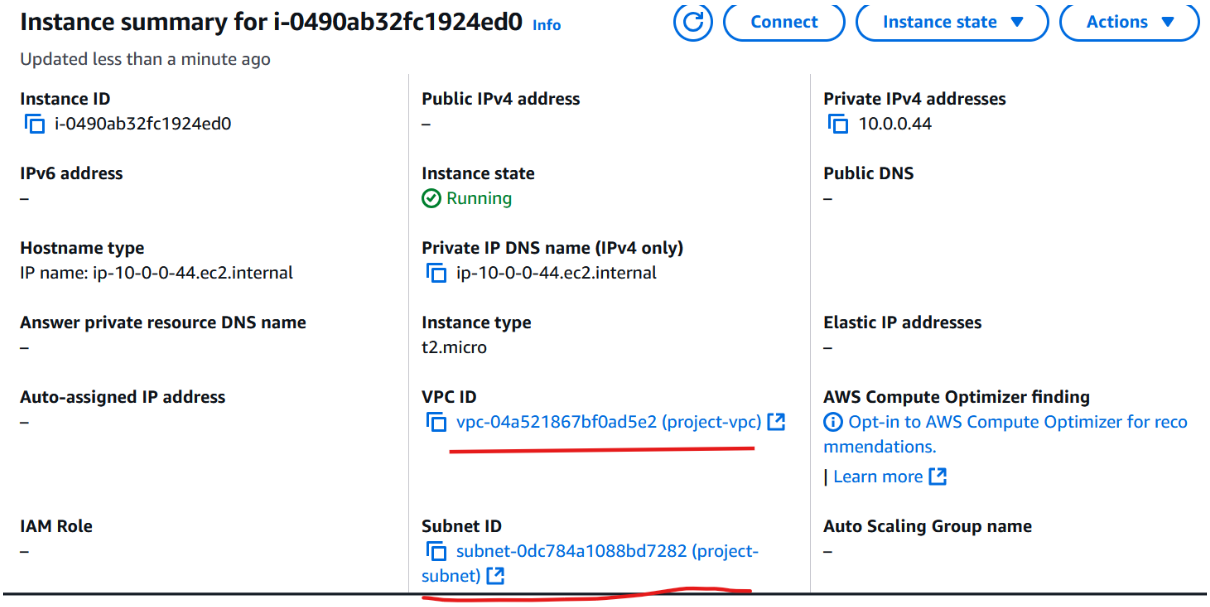
```
      + metadata_options (known after apply)

      + network_interface (known after apply)

      + private_dns_name_options (known after apply)

      + root_block_device (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.example: Creating...
aws_instance.example: Still creating... [00m10s elapsed]
aws_instance.example: Creation complete after 14s [id=i-0490ab32fc1924ed0]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
[luis@vbox PROJECT-ec2-instance-creation]$ 
```

- Once *terraform apply* is done, we see our EC2 instance ID: i-0490ab32fc1924ed0, checking into the AWS Console, the same ID is there.



□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
□		i-0490ab32fc1924ed0	Running	t2.micro	2/2 checks passec	View alarms +

- We can observe the VPC and Subnet that this EC2 instance is using.



Instance summary for i-0490ab32fc1924ed0

Updated less than a minute ago

Instance ID: i-0490ab32fc1924ed0

Public IPv4 address: –

Private IPv4 addresses: 10.0.0.44

IPv6 address: –

Instance state: Running

Public DNS: –

Hostname type: IP name: ip-10-0-0-44.ec2.internal

Private IP DNS name (IPv4 only): ip-10-0-0-44.ec2.internal

Answer private resource DNS name: –

Instance type: t2.micro

Elastic IP addresses: –

Auto-assigned IP address: –

VPC ID: vpc-04a521867bf0ad5e2 (project-vpc)

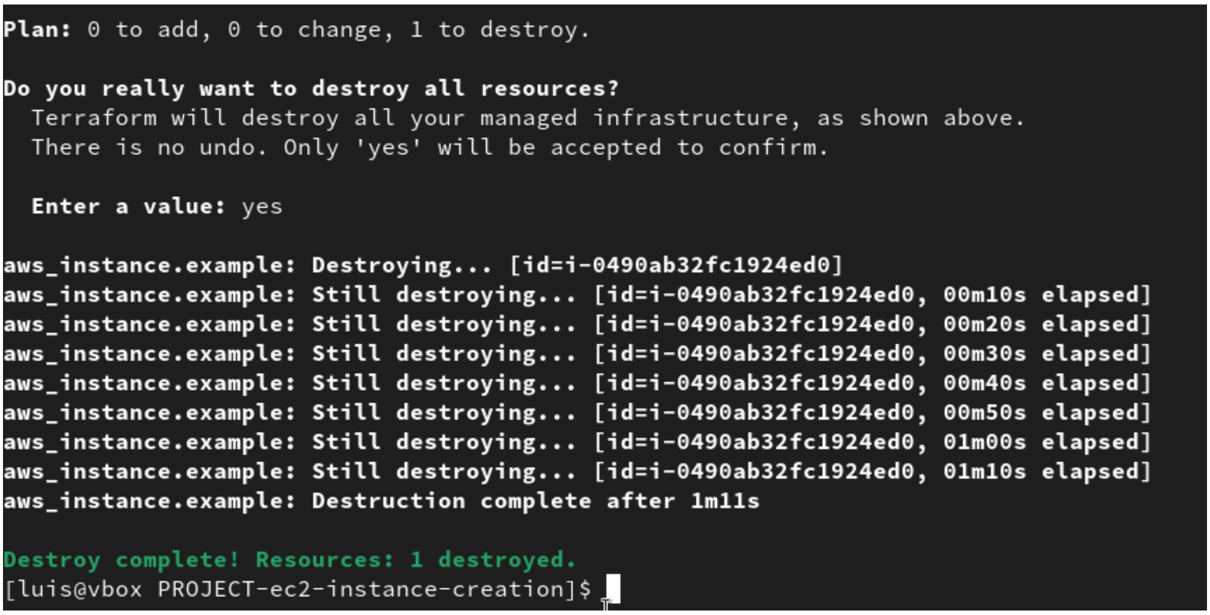
AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more

IAM Role: –

Subnet ID: subnet-0dc784a1088bd7282 (project-subnet)

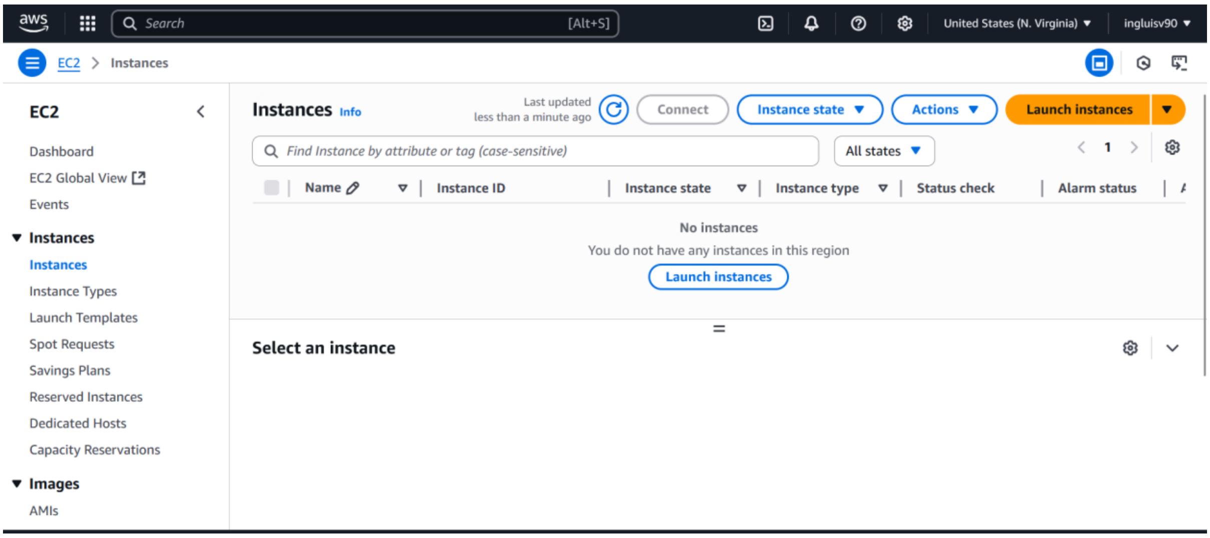
Auto Scaling Group name: –

8. **Terraform destroy:** Destroy the infrastructure managed by Terraform.



```
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.example: Destroying... [id=i-0490ab32fc1924ed0]
aws_instance.example: Still destroying... [id=i-0490ab32fc1924ed0, 00m10s elapsed]
aws_instance.example: Still destroying... [id=i-0490ab32fc1924ed0, 00m20s elapsed]
aws_instance.example: Still destroying... [id=i-0490ab32fc1924ed0, 00m30s elapsed]
aws_instance.example: Still destroying... [id=i-0490ab32fc1924ed0, 00m40s elapsed]
aws_instance.example: Still destroying... [id=i-0490ab32fc1924ed0, 00m50s elapsed]
aws_instance.example: Still destroying... [id=i-0490ab32fc1924ed0, 01m00s elapsed]
aws_instance.example: Still destroying... [id=i-0490ab32fc1924ed0, 01m10s elapsed]
aws_instance.example: Destruction complete after 1m11s

Destroy complete! Resources: 1 destroyed.
[luis@vbox PROJECT-ec2-instance-creation]$ 
```

- Checking into the AWS Console again, the instance was deleted successfully.



No instances

You do not have any instances in this region