Lesson-End Project

Building and Testing a Terraform Module

Project agenda: To build and test a Terraform module for efficient and streamlined infrastructure management using AWS

Description: As a DevOps engineer at InnovateNow Tech, you are tasked with building and testing a Terraform module to standardize cloud resource deployment across projects. Your objectives include creating a reusable module that follows best practices, ensures security compliance, and is flexible for different requirements. Additionally, you will develop automated tests to validate the module's functionality, aiming to streamline provisioning, reduce errors, and accelerate delivery.

Tools required: Terraform and AWS CLI

Prerequisites: Refer to Lesson 09 Demo 01 to install and set up Terraform, and ensure you have the AWS CLI installed to configure AWS

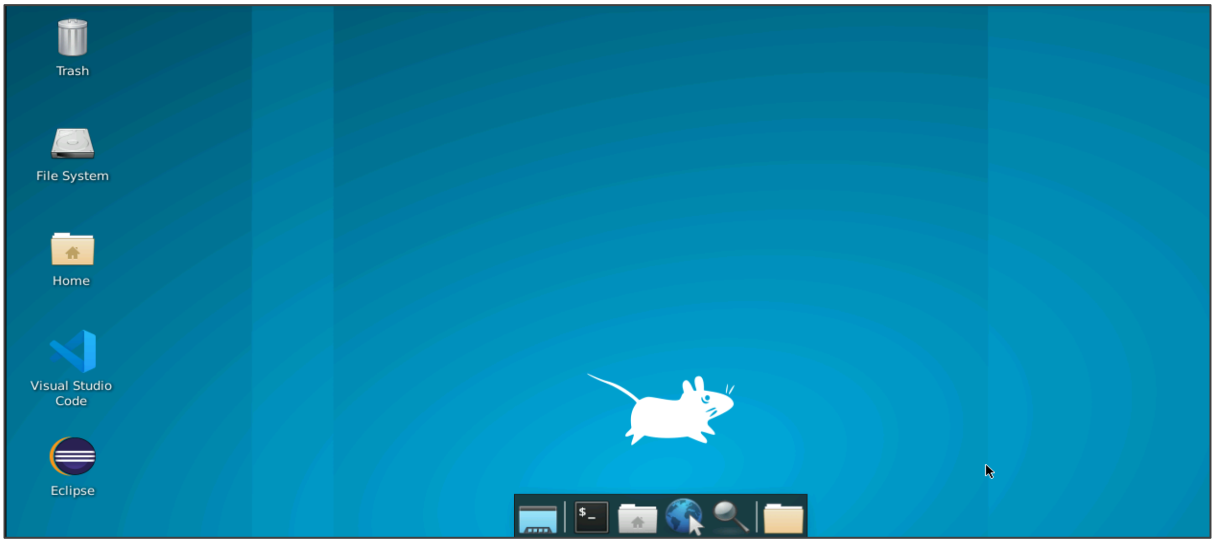
Expected deliverables: A step-by-step guide to writing the Terraform VPC module code, writing the main Terraform project code, and deploying the code to test the module.

Steps to be followed:

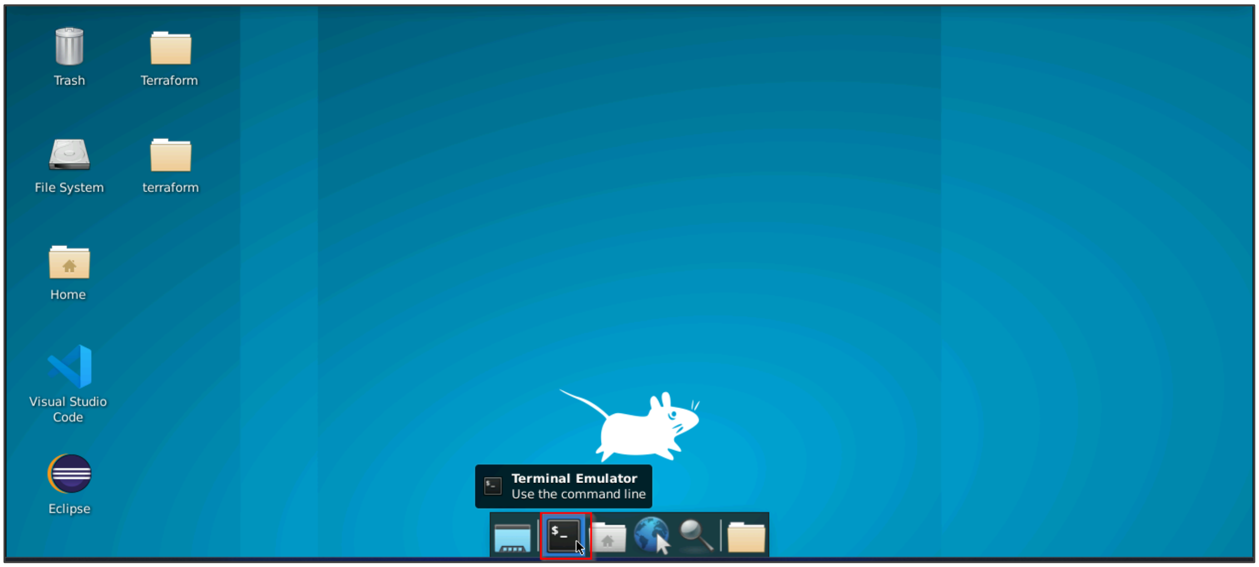
1. Configure the AWS CLI from the terminal
2. Create the directory structure for the Terraform project
3. Write the Terraform VPC module code
4. Write the main Terraform project code
5. Deploy the code and test the module

Step 1: Configure the AWS CLI from the terminal

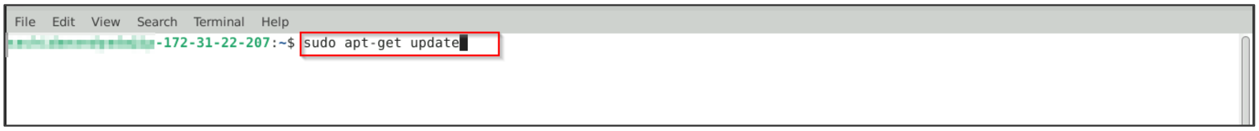
1.1 Open the DevOps Lab:



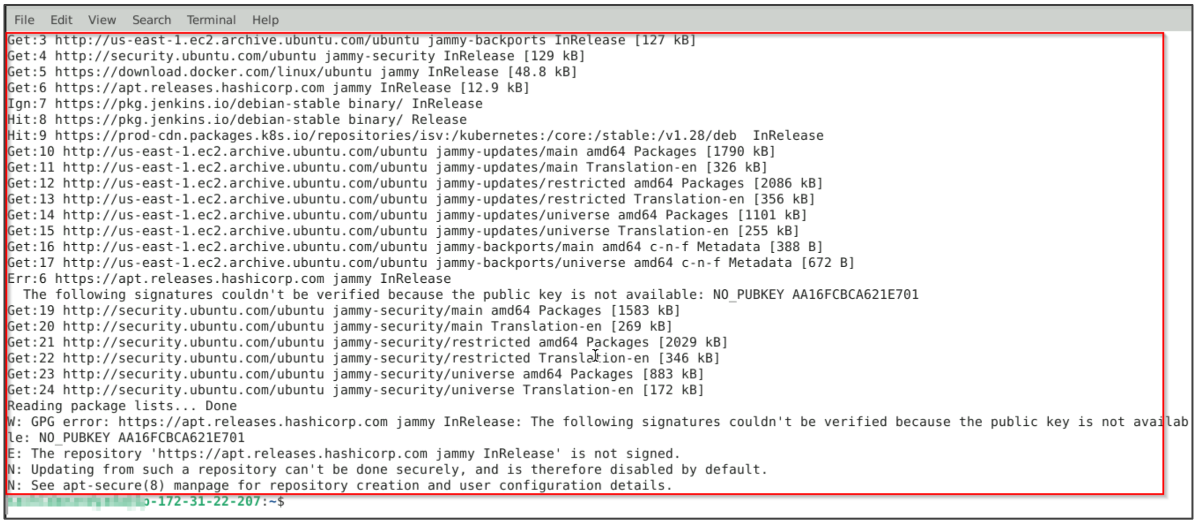
1.2 Open the terminal:



1.3 Enter the command given below to fetch the latest package information from the repositories and update the local package index:
sudo apt-get update



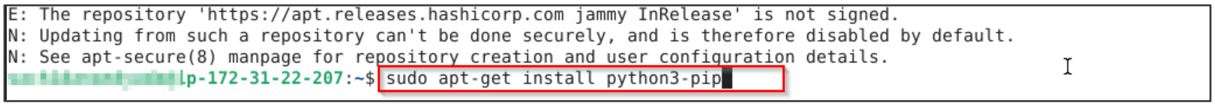
```
[redacted]-172-31-22-207:~$ sudo apt-get update
```



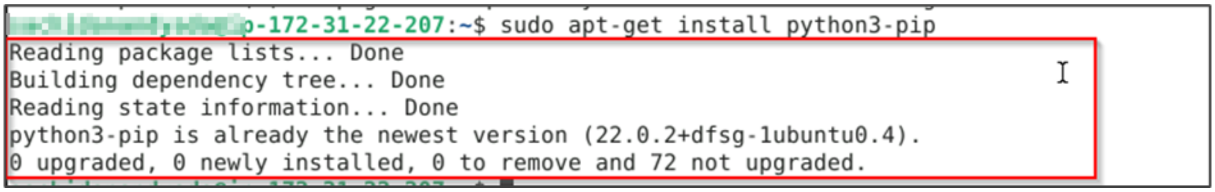
```
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:5 https://download.docker.com/linux/ubuntu jammy InRelease [48.8 kB]
Get:6 https://apt.releases.hashicorp.com jammy InRelease [12.9 kB]
Ign:7 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:8 https://pkg.jenkins.io/debian-stable binary/ Release
Hit:9 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.28/deb  InRelease
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1790 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [326 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [2086 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [356 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1101 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [255 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [388 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [672 B]
Err:6 https://apt.releases.hashicorp.com jammy InRelease
  The following signatures couldn't be verified because the public key is not available: NO_PUBKEY AA16FCBCA621E701
Get:19 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1583 kB]
Get:20 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [269 kB]
Get:21 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [2029 kB]
Get:22 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [346 kB]
Get:23 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [883 kB]
Get:24 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [172 kB]
Reading package lists... Done
W: GPG error: https://apt.releases.hashicorp.com jammy InRelease: The following signatures couldn't be verified because the public key is not availab
le: NO_PUBKEY AA16FCBCA621E701
E: The repository 'https://apt.releases.hashicorp.com jammy InRelease' is not signed.
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
[redacted]p-172-31-22-207:~$
```

The command runs successfully.

1.4 Enter the command given below to install the Python 3 package installer (pip) on your system:
sudo apt-get install python3-pip



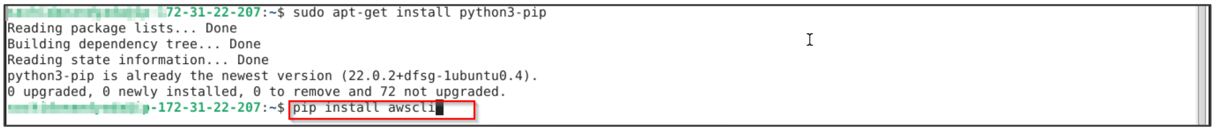
```
E: The repository 'https://apt.releases.hashicorp.com jammy InRelease' is not signed.
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
[redacted]lp-172-31-22-207:~$ sudo apt-get install python3-pip
```



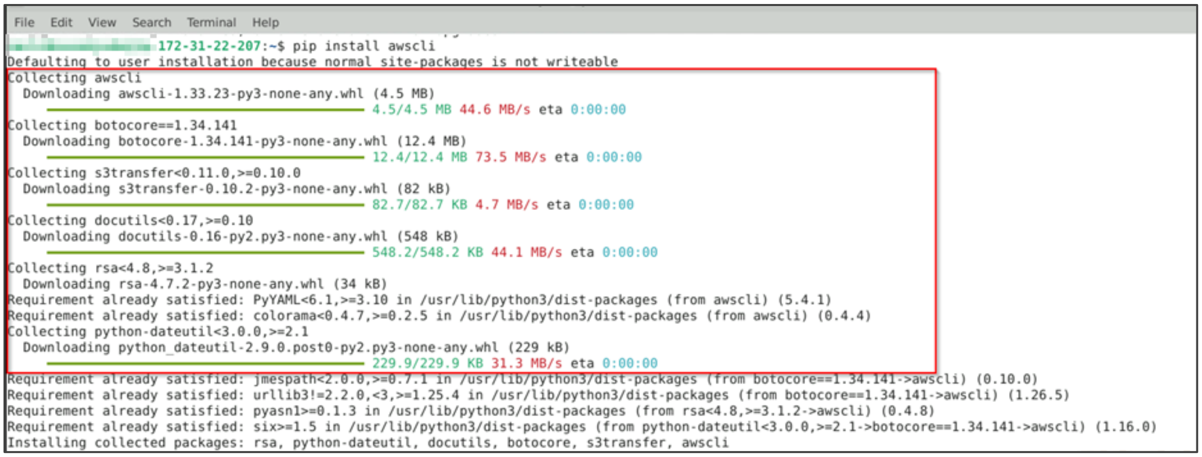
```
[redacted]p-172-31-22-207:~$ sudo apt-get install python3-pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3-pip is already the newest version (22.0.2+dfsg-1ubuntu0.4).
0 upgraded, 0 newly installed, 0 to remove and 72 not upgraded.
```

The Python package gets installed successfully.

1.5 Run the below command to install AWS CLI:
pip install awscli



```
[redacted]72-31-22-207:~$ sudo apt-get install python3-pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3-pip is already the newest version (22.0.2+dfsg-1ubuntu0.4).
0 upgraded, 0 newly installed, 0 to remove and 72 not upgraded.
[redacted]p-172-31-22-207:~$ pip install awscli
```

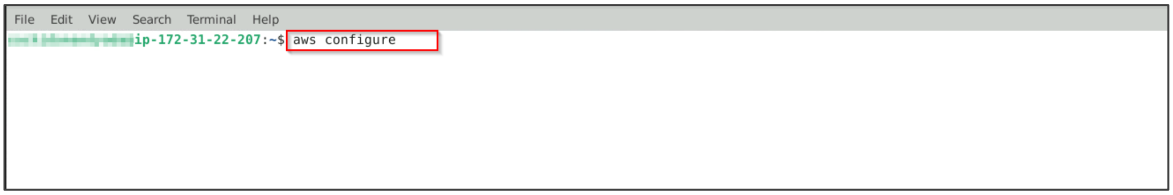


```
[redacted]172-31-22-207:~$ pip install awscli
Defaulting to user installation because normal site-packages is not writeable
Collecting awscli
  Downloading awscli-1.33.23-py3-none-any.whl (4.5 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.5/4.5 MB 44.6 MB/s eta 0:00:00
Collecting botocore==1.34.141
  Downloading botocore-1.34.141-py3-none-any.whl (12.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.4/12.4 MB 73.5 MB/s eta 0:00:00
Collecting s3transfer<0.11.0,>=0.10.0
  Downloading s3transfer-0.10.2-py3-none-any.whl (82 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 82.7/82.7 KB 4.7 MB/s eta 0:00:00
Collecting docutils<0.17,>=0.10
  Downloading docutils-0.16-py2.py3-none-any.whl (548 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 548.2/548.2 KB 44.1 MB/s eta 0:00:00
Collecting rsa<4.8,>=3.1.2
  Downloading rsa-4.7.2-py3-none-any.whl (34 kB)
Requirement already satisfied: PyYAML<6.1,>=3.10 in /usr/lib/python3/dist-packages (from awscli) (5.4.1)
Requirement already satisfied: colorama<0.4.7,>=0.2.5 in /usr/lib/python3/dist-packages (from awscli) (0.4.4)
Collecting python-dateutil<3.0.0,>=2.1
  Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 229.9/229.9 KB 31.3 MB/s eta 0:00:00
Requirement already satisfied: jmespath<2.0.0,>=0.7.1 in /usr/lib/python3/dist-packages (from botocore==1.34.141->awscli) (0.10.0)
Requirement already satisfied: urllib3!=2.2.0,<3,>=1.25.4 in /usr/lib/python3/dist-packages (from botocore==1.34.141->awscli) (1.26.5)
Requirement already satisfied: pyasn1>=0.1.3 in /usr/lib/python3/dist-packages (from rsa<4.8,>=3.1.2->awscli) (0.4.8)
Requirement already satisfied: six>=1.5 in /usr/lib/python3/dist-packages (from python-dateutil<3.0.0,>=2.1->botocore==1.34.141->awscli) (1.16.0)
Installing collected packages: rsa, python-dateutil, docutils, botocore, s3transfer, awscli
```

The AWS CLI gets installed successfully.

1.6 Execute the command below to set up your AWS credentials as environment variables:

aws configure



```
[redacted]ip-172-31-22-207:~$ aws configure
```

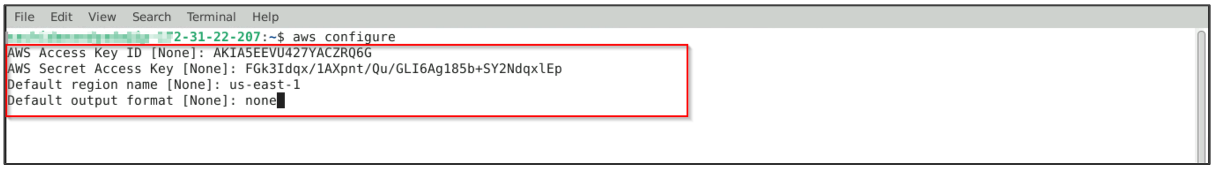
1.7 Enter the AWS credentials as per the required fields:

AWS Access Key ID: AKIA5EEVU427YACZRQ6G

AWS Secret Access Key: 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

Default Region: us-east-1

Default output format: none



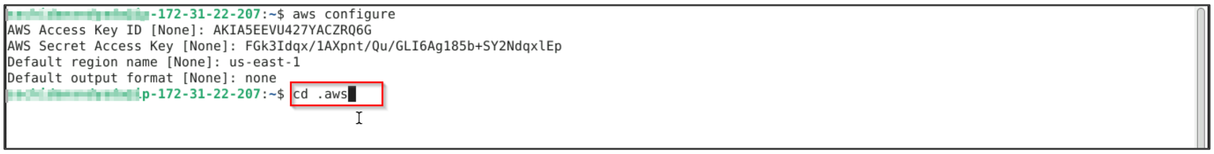
```
[redacted]2-31-22-207:~$ aws configure
AWS Access Key ID [None]: AKIA5EEVU427YACZRQ6G
AWS Secret Access Key [None]: FGk3Idqx/1AXpnt/Qu/GLI6Ag185b+SY2NdqxlEp
Default region name [None]: us-east-1
Default output format [None]: none
```

This configuration process stores the credentials in a file at **~/.aws/credentials.**

Note: Keep your security credentials available to authenticate your AWS account.

1.8 Navigate to the **aws** directory to add the security token of your AWS by giving the following command:

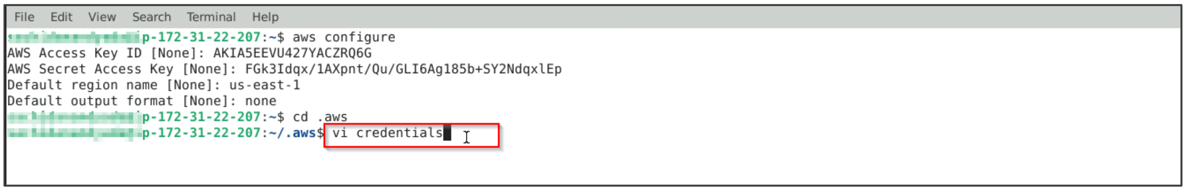
cd .aws



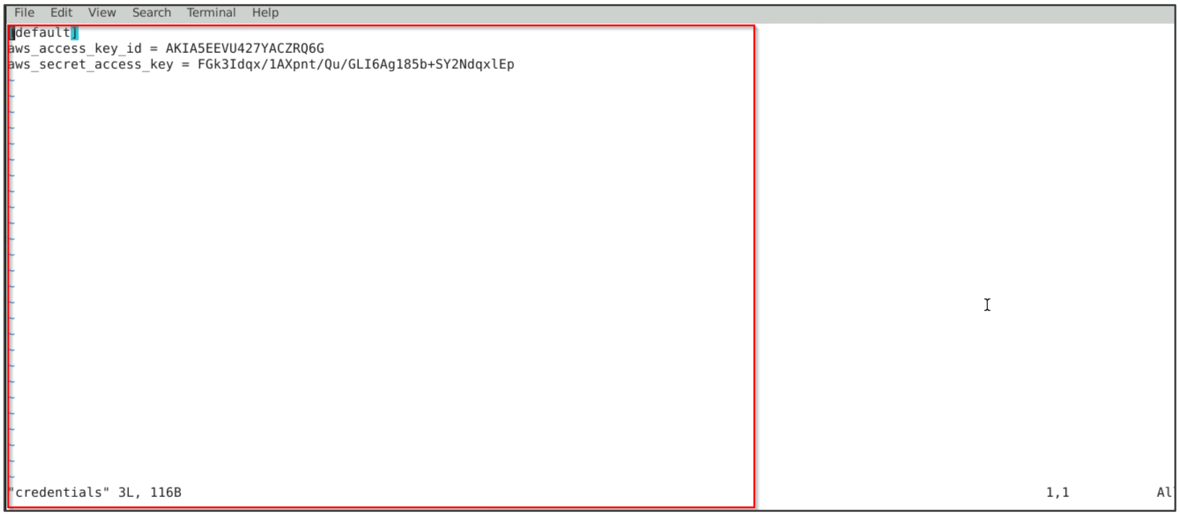
```
[redacted]-172-31-22-207:~$ aws configure
AWS Access Key ID [None]: AKIA5EEVU427YACZRQ6G
AWS Secret Access Key [None]: FGk3Idqx/1AXpnt/Qu/GLI6Ag185b+SY2NdqxlEp
Default region name [None]: us-east-1
Default output format [None]: none
[redacted]p-172-31-22-207:~$ cd .aws
```

1.9 Run the given command to check the credentials file and the secret key created:

vi credentials



```
[redacted]p-172-31-22-207:~$ aws configure
AWS Access Key ID [None]: AKIA5EEVU427YACZRQ6G
AWS Secret Access Key [None]: FGk3Idqx/1AXpnt/Qu/GLI6Ag185b+SY2NdqxlEp
Default region name [None]: us-east-1
Default output format [None]: none
[redacted]p-172-31-22-207:~$ cd .aws
[redacted]p-172-31-22-207:~/.aws$ vi credentials
```

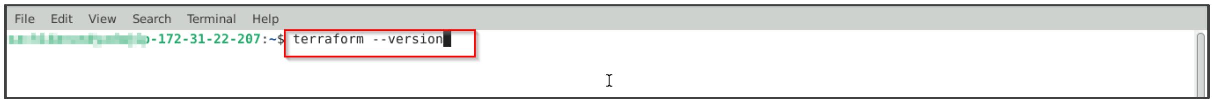


```
[default]
aws_access_key_id = AKIA5EEVU427YACZRQ6G
aws_secret_access_key = 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
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"credentials" 3L, 116B                                                    1,1           Al
```

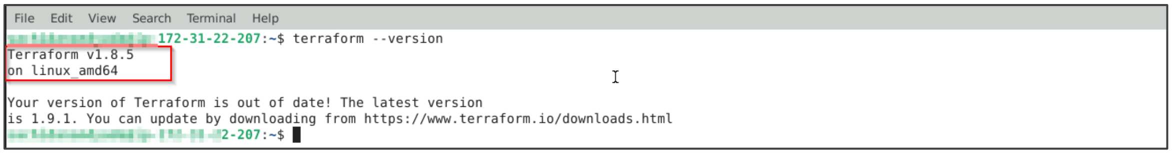
Step 2: Create the directory structure for the Terraform project

2.1 Enter the given command to check the installed version of Terraform:

terraform --version



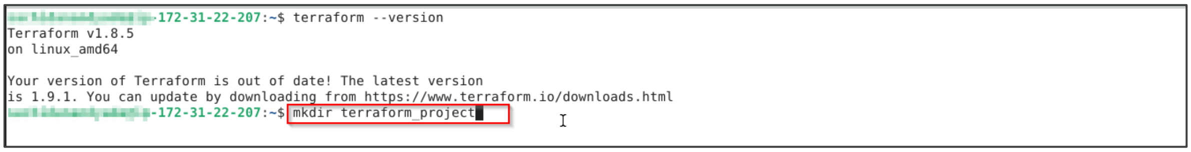
```
[redacted]p-172-31-22-207:~$ terraform --version
```



```
[redacted]172-31-22-207:~$ terraform --version
Terraform v1.8.5
on linux_amd64

Your version of Terraform is out of date! The latest version
is 1.9.1. You can update by downloading from https://www.terraform.io/downloads.html
[redacted]2-207:~$
```

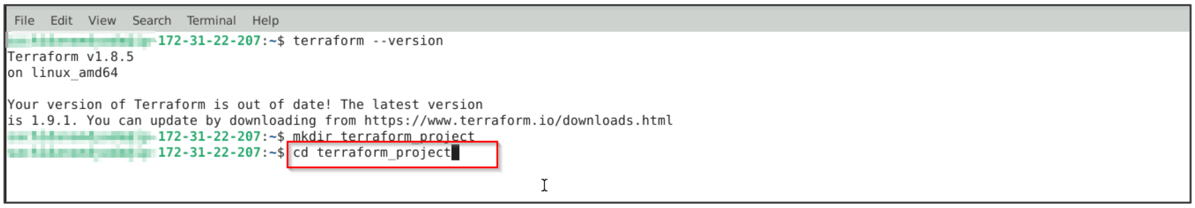
The installed version of Terraform is visible.

2.2 Enter the given command to create a new directory for your Terraform code:
mkdir terraform_project



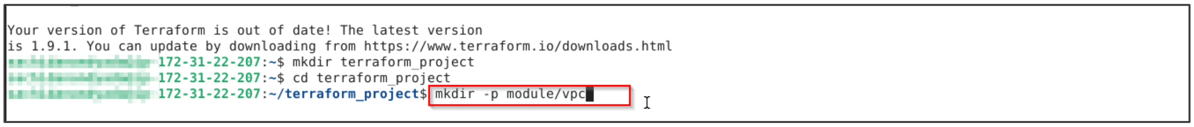
```
[redacted]-172-31-22-207:~$ terraform --version
Terraform v1.8.5
on linux_amd64

Your version of Terraform is out of date! The latest version
is 1.9.1. You can update by downloading from https://www.terraform.io/downloads.html
[redacted]-172-31-22-207:~$ mkdir terraform_project
```

2.3 Navigate to the main project directory by running the command given below:
cd terraform_project

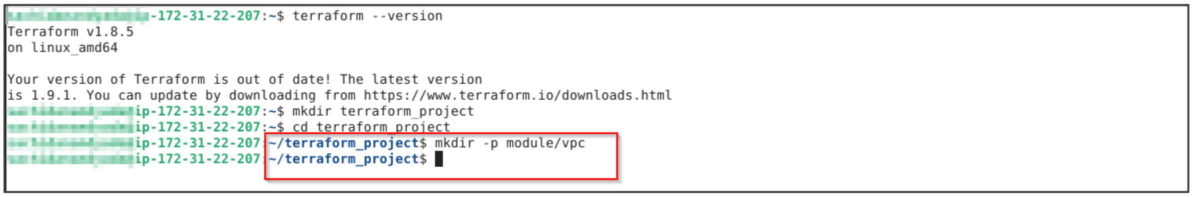


```
[redacted]172-31-22-207:~$ terraform --version
Terraform v1.8.5
on linux_amd64

Your version of Terraform is out of date! The latest version
is 1.9.1. You can update by downloading from https://www.terraform.io/downloads.html
[redacted]172-31-22-207:~$ mkdir terraform_project
[redacted]172-31-22-207:~$ cd terraform_project
```

2.4 Run the following command to create a custom directory named **module** and a directory inside it named **vpc**:
mkdir -p module/vpc



```
Your version of Terraform is out of date! The latest version
is 1.9.1. You can update by downloading from https://www.terraform.io/downloads.html
[redacted]172-31-22-207:~$ mkdir terraform_project
[redacted]172-31-22-207:~$ cd terraform_project
[redacted]172-31-22-207:~/terraform_project$ mkdir -p module/vpc
```

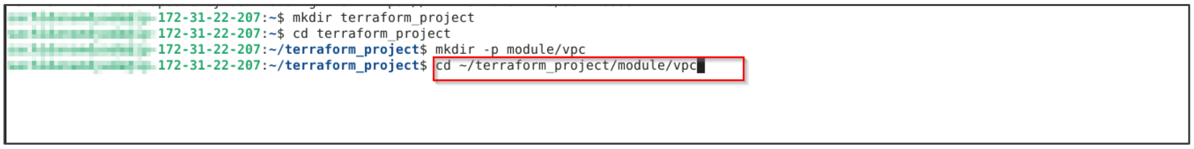


```
[redacted]-172-31-22-207:~$ terraform --version
Terraform v1.8.5
on linux_amd64

Your version of Terraform is out of date! The latest version
is 1.9.1. You can update by downloading from https://www.terraform.io/downloads.html
[redacted]ip-172-31-22-207:~$ mkdir terraform_project
[redacted]ip-172-31-22-207:~$ cd terraform_project
[redacted]ip-172-31-22-207:~/terraform_project$ mkdir -p module/vpc
[redacted]ip-172-31-22-207:~/terraform_project$
```

The custom directory named module and the directory inside it named **vpc** are created successfully.

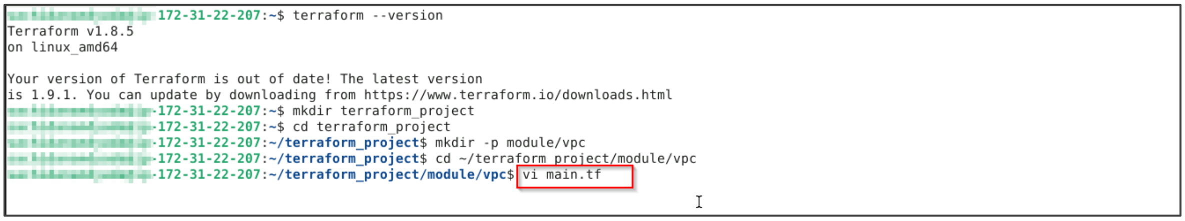
Step 3: Write the Terraform VPC module code

3.1 Navigate to the **vpc** directory by executing the following command:
cd ~/terraform_project/module/vpc



```
[redacted]-172-31-22-207:~$ mkdir terraform_project
[redacted]-172-31-22-207:~$ cd terraform_project
[redacted]-172-31-22-207:~/terraform_project$ mkdir -p module/vpc
[redacted]-172-31-22-207:~/terraform_project$ cd ~/terraform_project/module/vpc
```

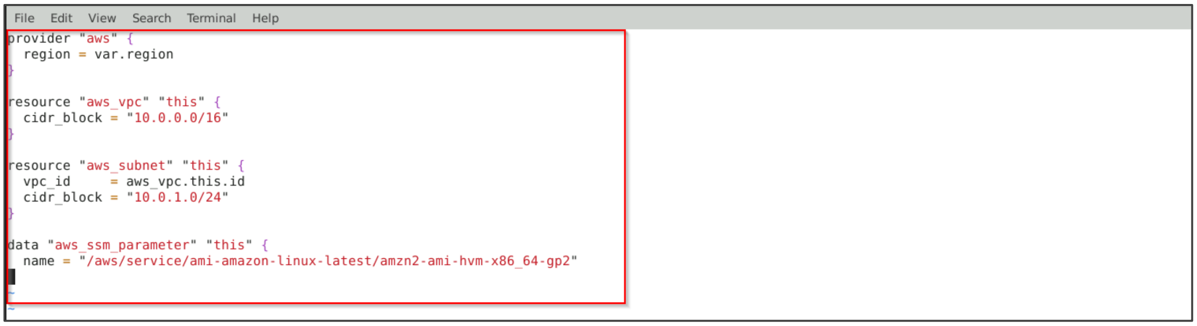
3.2 Execute the following command to create a new file named **main.tf**:
vi main.tf



```
[redacted]-172-31-22-207:~$ terraform --version
Terraform v1.8.5
on linux_amd64

Your version of Terraform is out of date! The latest version
is 1.9.1. You can update by downloading from https://www.terraform.io/downloads.html
[redacted]-172-31-22-207:~$ mkdir terraform_project
[redacted]-172-31-22-207:~$ cd terraform_project
[redacted]-172-31-22-207:~/terraform_project$ mkdir -p module/vpc
[redacted]-172-31-22-207:~/terraform_project$ cd ~/terraform_project/module/vpc
[redacted]-172-31-22-207:~/terraform_project/module/vpc$ vi main.tf
```

3.3 Enter the code given below in the **main.tf** file:

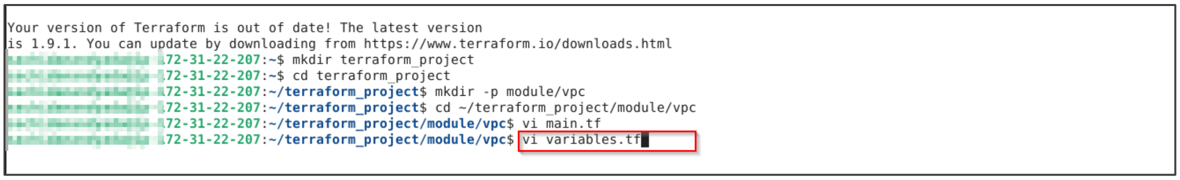
```
provider "aws" {
  region = var.region
}

resource "aws_vpc" "this" {
  cidr_block = "10.0.0.0/16"
}

resource "aws_subnet" "this" {
  vpc_id     = aws_vpc.this.id
  cidr_block = "10.0.1.0/24"
}

data "aws_ssm_parameter" "this" {
  name = "/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2"
}
```



```
provider "aws" {
  region = var.region
}

resource "aws_vpc" "this" {
  cidr_block = "10.0.0.0/16"
}

resource "aws_subnet" "this" {
  vpc_id     = aws_vpc.this.id
  cidr_block = "10.0.1.0/24"
}

data "aws_ssm_parameter" "this" {
  name = "/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2"

~
~
```

Note: Enter **:wq** in the last line of the **main.tf** file to save the file and exit to the command prompt

3.4 Enter the given command to create a new file called **variables.tf**:
vi variables.tf



```
Your version of Terraform is out of date! The latest version
is 1.9.1. You can update by downloading from https://www.terraform.io/downloads.html
[redacted]72-31-22-207:~$ mkdir terraform_project
[redacted]72-31-22-207:~$ cd terraform_project
[redacted]72-31-22-207:~/terraform_project$ mkdir -p module/vpc
[redacted]72-31-22-207:~/terraform_project$ cd ~/terraform_project/module/vpc
[redacted]72-31-22-207:~/terraform_project/module/vpc$ vi main.tf
[redacted]72-31-22-207:~/terraform_project/module/vpc$ vi variables.tf
```

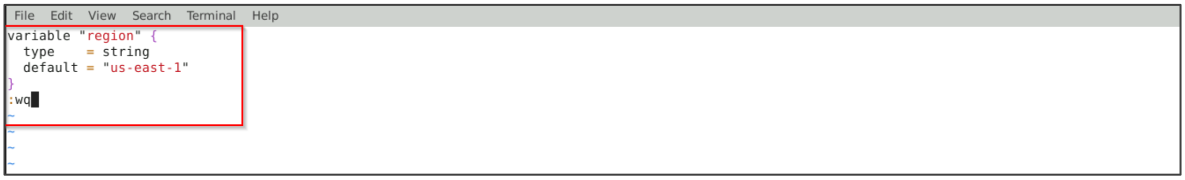
3.5 Enter the script given below in the **variables.tf** file:

variable "region" {

type = string

default = "us-east-1"

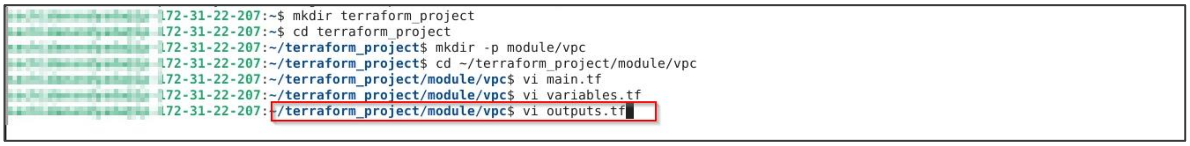
}



```
variable "region" {
  type    = string
  default = "us-east-1"
}
:wq
~
~
~
~
```

Note: Enter **:wq** in the last line of the **variables.tf** file to save the file and exit to the command prompt

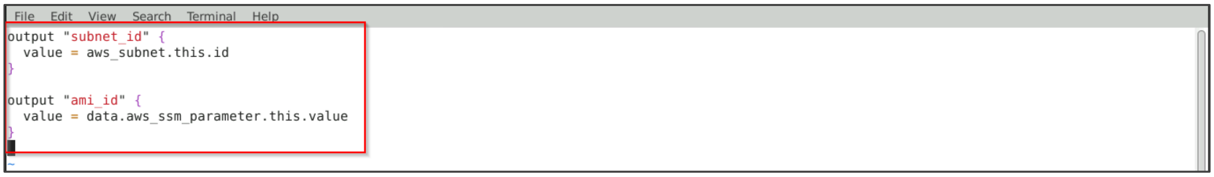
3.6 Run the given command to create a new file called **outputs.tf** and add the provided code:
vi outputs.tf



```
[redacted]172-31-22-207:~$ mkdir terraform_project
[redacted]172-31-22-207:~$ cd terraform_project
[redacted]172-31-22-207:~/terraform_project$ mkdir -p module/vpc
[redacted]172-31-22-207:~/terraform_project$ cd ~/terraform_project/module/vpc
[redacted]172-31-22-207:~/terraform_project/module/vpc$ vi main.tf
[redacted]172-31-22-207:~/terraform_project/module/vpc$ vi variables.tf
[redacted]172-31-22-207:~/terraform_project/module/vpc$ vi outputs.tf
```

3.7 Enter the script given below in the **outputs.tf** file:

```
output "subnet_id" {
  value = aws_subnet.this.id
}

output "ami_id" {
  value = data.aws_ssm_parameter.this.value
}
```

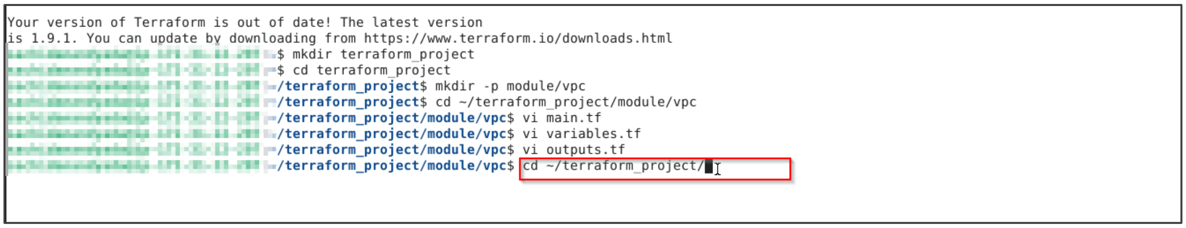


```
output "subnet_id" {
  value = aws_subnet.this.id
}

output "ami_id" {
  value = data.aws_ssm_parameter.this.value
}
~
```

> **Note**: Enter **:wq** in the last line of the **outputs.tf** file to save the file and exit to the command prompt

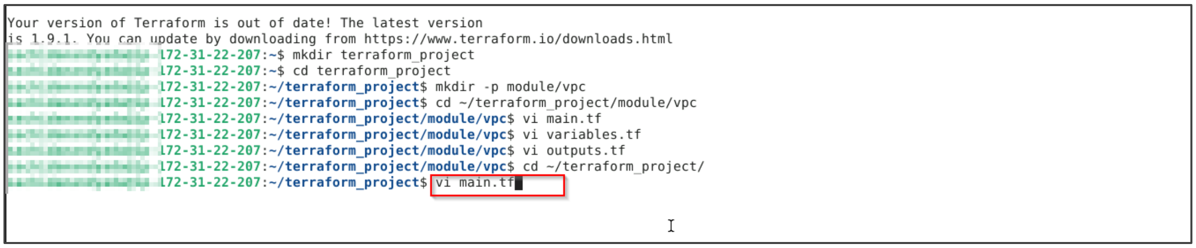
Step 4: Write the main Terraform project code

4.1 Navigate to the **terraform_project** directory using the given command:
cd ~/terraform_project/



```
Your version of Terraform is out of date! The latest version
is 1.9.1. You can update by downloading from https://www.terraform.io/downloads.html
[redacted]$ mkdir terraform_project
[redacted]$ cd terraform_project
[redacted]/terraform_project$ mkdir -p module/vpc
[redacted]/terraform_project$ cd ~/terraform_project/module/vpc
[redacted]/terraform_project/module/vpc$ vi main.tf
[redacted]/terraform_project/module/vpc$ vi variables.tf
[redacted]/terraform_project/module/vpc$ vi outputs.tf
[redacted]/terraform_project/module/vpc$ cd ~/terraform_project/
```

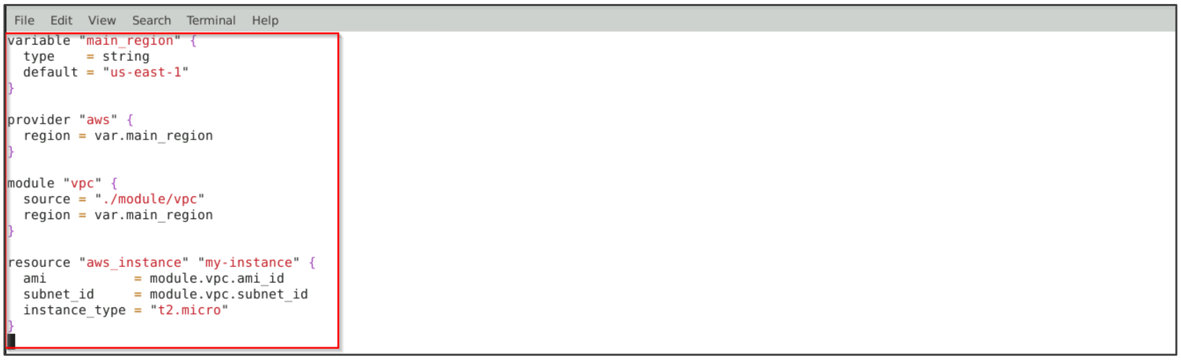
4.2 Create a new file called **main.tf** using the command given below:
vi main.tf



```
Your version of Terraform is out of date! The latest version
is 1.9.1. You can update by downloading from https://www.terraform.io/downloads.html
[redacted]72-31-22-207:~$ mkdir terraform_project
[redacted]72-31-22-207:~$ cd terraform_project
[redacted]72-31-22-207:~/terraform_project$ mkdir -p module/vpc
[redacted]72-31-22-207:~/terraform_project$ cd ~/terraform_project/module/vpc
[redacted]72-31-22-207:~/terraform_project/module/vpc$ vi main.tf
[redacted]72-31-22-207:~/terraform_project/module/vpc$ vi variables.tf
[redacted]72-31-22-207:~/terraform_project/module/vpc$ vi outputs.tf
[redacted]72-31-22-207:~/terraform_project/module/vpc$ cd ~/terraform_project/
[redacted]72-31-22-207:~/terraform_project$ vi main.tf
```

4.3 In the **main.tf** file, enter the script given below to invoke the VPC module created earlier:

```
variable "main_region" {
  type    = string
  default = "us-east-1"
}

provider "aws" {
  region = var.main_region
}

module "vpc" {
  source = "./module/vpc"
  region = var.main_region
}

resource "aws_instance" "my-instance" {
  ami           = module.vpc.ami_id
  subnet_id     = module.vpc.subnet_id
  instance_type = "t2.micro"
}
```

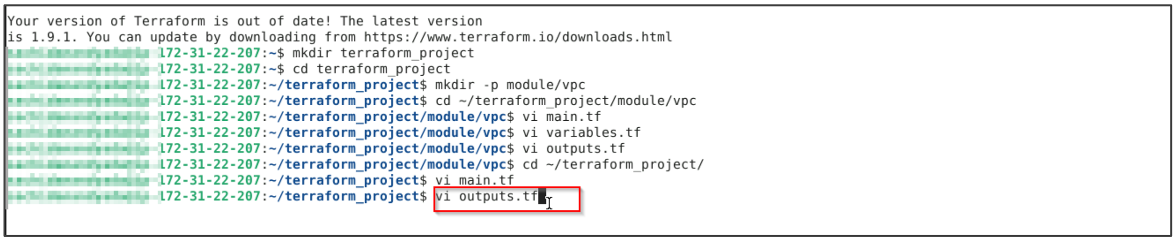


```
variable "main_region" {
  type    = string
  default = "us-east-1"
}

provider "aws" {
  region = var.main_region
}

module "vpc" {
  source = "./module/vpc"
  region = var.main_region
}

resource "aws_instance" "my-instance" {
  ami           = module.vpc.ami_id
  subnet_id     = module.vpc.subnet_id
  instance_type = "t2.micro"
}
```

Note: Enter **:wq** in the last line of the **main.tf** file to save the file and exit to the command prompt

4.4 Enter the given command to create a new file called **outputs.tf**

vi outputs.tf



```
Your version of Terraform is out of date! The latest version
is 1.9.1. You can update by downloading from https://www.terraform.io/downloads.html
[redacted]172-31-22-207:~$ mkdir terraform_project
[redacted]172-31-22-207:~$ cd terraform_project
[redacted]172-31-22-207:~/terraform_project$ mkdir -p module/vpc
[redacted]172-31-22-207:~/terraform_project$ cd ~/terraform_project/module/vpc
[redacted]172-31-22-207:~/terraform_project/module/vpc$ vi main.tf
[redacted]172-31-22-207:~/terraform_project/module/vpc$ vi variables.tf
[redacted]172-31-22-207:~/terraform_project/module/vpc$ vi outputs.tf
[redacted]172-31-22-207:~/terraform_project/module/vpc$ cd ~/terraform_project/
[redacted]172-31-22-207:~/terraform_project$ vi main.tf
[redacted]172-31-22-207:~/terraform_project$ vi outputs.tf
```

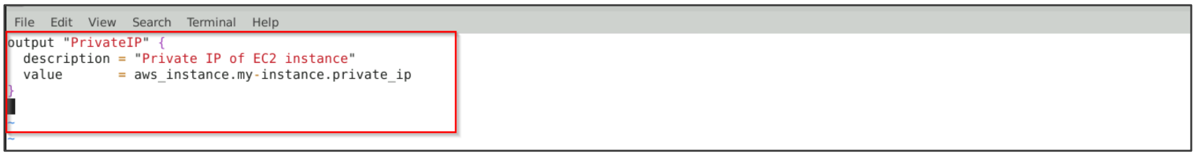
4.5 Enter the code given below in the **outputs.tf file**:

output "PrivateIP" {

description = "Private IP of EC2 instance"

value = aws_instance.my-instance.private_ip

}

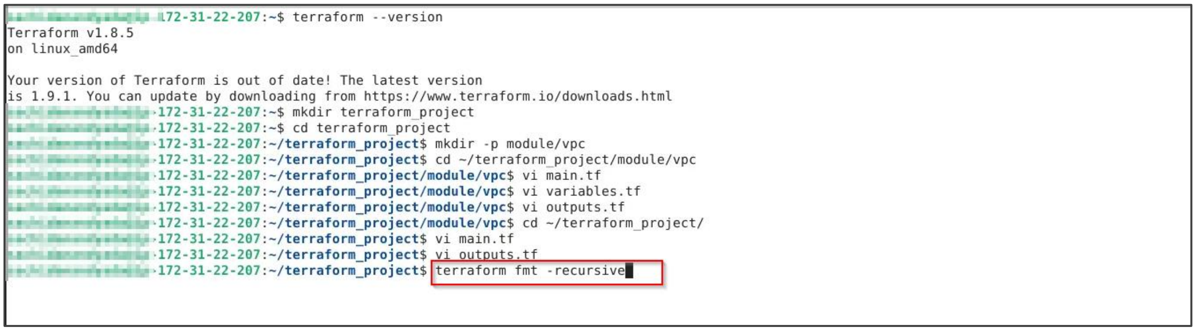


```
output "PrivateIP" {
  description = "Private IP of EC2 instance"
  value       = aws_instance.my-instance.private_ip
}
```

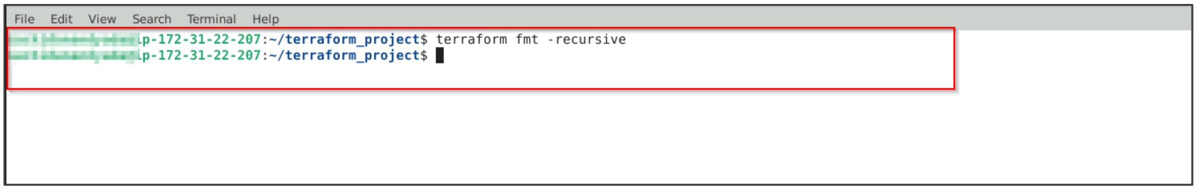
Note: Enter **:wq** in the last line of the **outputs.tf** file to save the file and exit to the command prompt

Step 5: Deploy the code and test the module

5.1 Execute the following command to format the code in all the files:
terraform fmt -recursive



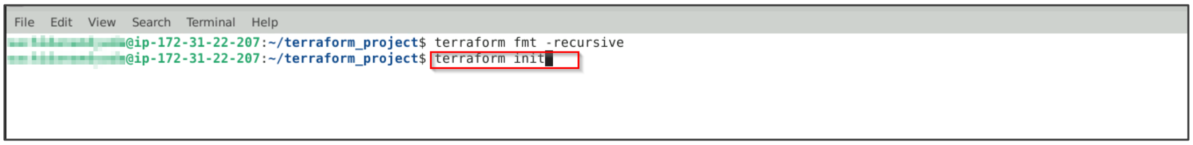
```
[redacted]172-31-22-207:~$ terraform --version
Terraform v1.8.5
on linux_amd64

Your version of Terraform is out of date! The latest version
is 1.9.1. You can update by downloading from https://www.terraform.io/downloads.html
[redacted]172-31-22-207:~$ mkdir terraform_project
[redacted]172-31-22-207:~$ cd terraform_project
[redacted]172-31-22-207:~/terraform_project$ mkdir -p module/vpc
[redacted]172-31-22-207:~/terraform_project$ cd ~/terraform_project/module/vpc
[redacted]172-31-22-207:~/terraform_project/module/vpc$ vi main.tf
[redacted]172-31-22-207:~/terraform_project/module/vpc$ vi variables.tf
[redacted]172-31-22-207:~/terraform_project/module/vpc$ vi outputs.tf
[redacted]172-31-22-207:~/terraform_project/module/vpc$ cd ~/terraform_project/
[redacted]172-31-22-207:~/terraform_project$ vi main.tf
[redacted]172-31-22-207:~/terraform_project$ vi outputs.tf
[redacted]172-31-22-207:~/terraform_project$ terraform fmt -recursive
```



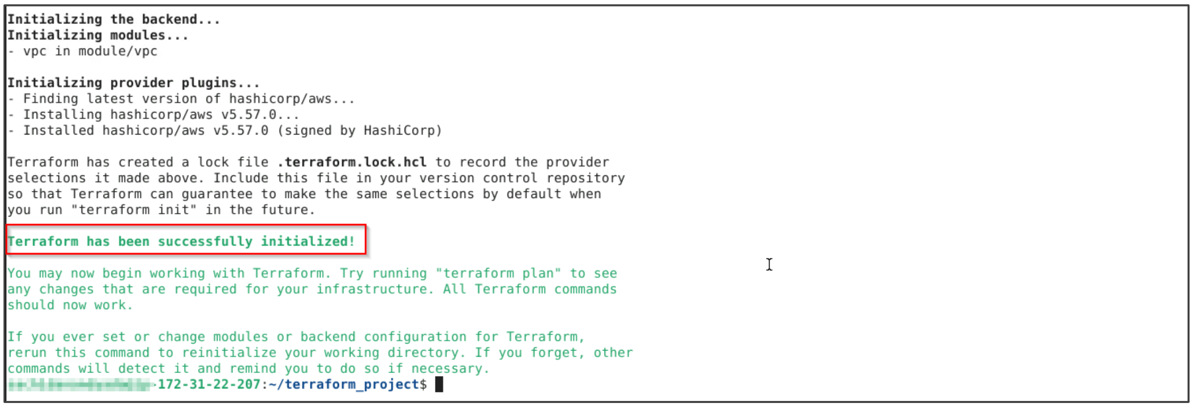
```
[redacted]p-172-31-22-207:~/terraform_project$ terraform fmt -recursive
[redacted]p-172-31-22-207:~/terraform_project$
```

The format command runs successfully.

5.2 Execute the command given below to initialize the Terraform configuration to fetch any required providers and get the code referenced in the module block:
terraform init



```
[redacted]@ip-172-31-22-207:~/terraform_project$ terraform fmt -recursive
[redacted]@ip-172-31-22-207:~/terraform_project$ terraform init
```

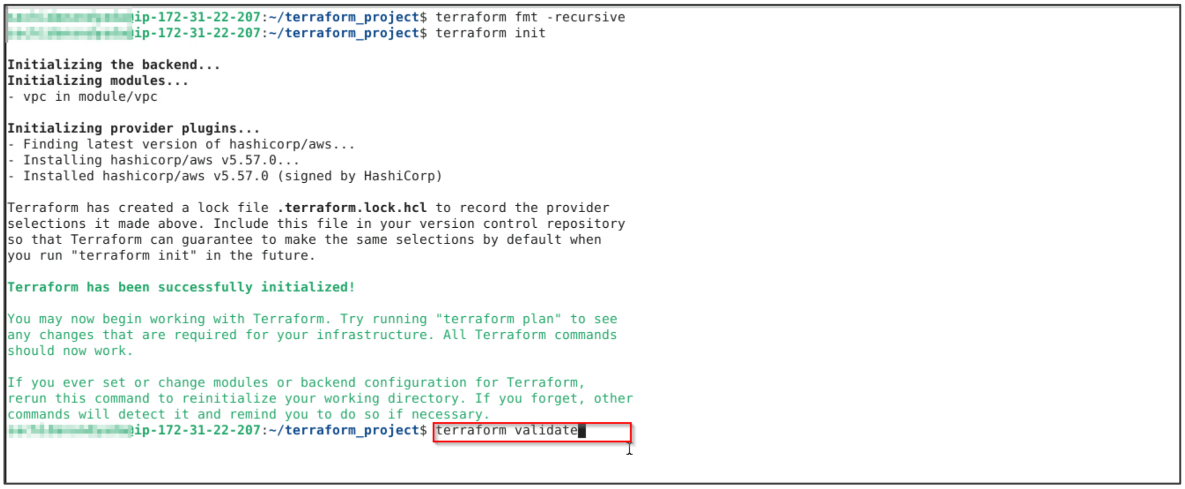


```
Initializing the backend...
Initializing modules...
- vpc in module/vpc

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.57.0...
- Installed hashicorp/aws v5.57.0 (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.
[redacted]172-31-22-207:~/terraform_project$
```

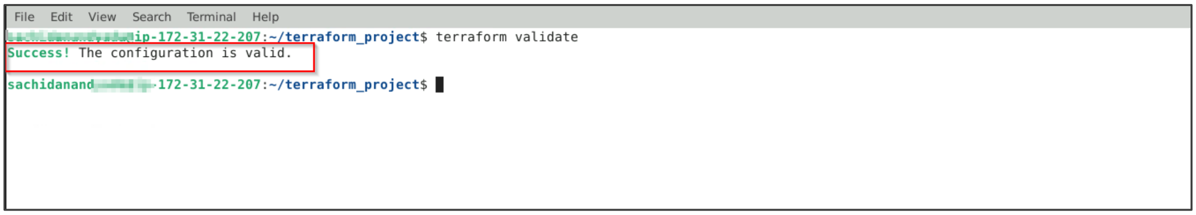
The terraform configuration file is successfully initialized.

5.3 Run the command given below to validate the code:

terraform validate



```
[redacted]ip-172-31-22-207:~/terraform_project$ terraform fmt -recursive
[redacted]ip-172-31-22-207:~/terraform_project$ terraform init

Initializing the backend...
Initializing modules...
- vpc in module/vpc

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.57.0...
- Installed hashicorp/aws v5.57.0 (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.
[redacted]ip-172-31-22-207:~/terraform_project$ terraform validate
```

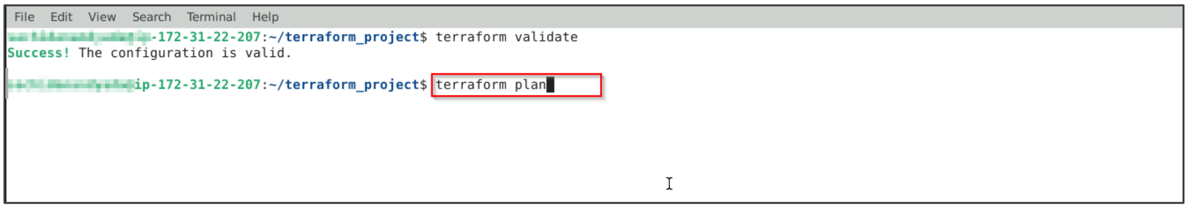


```
[redacted]ip-172-31-22-207:~/terraform_project$ terraform validate
Success! The configuration is valid.

sachidanand[redacted]-172-31-22-207:~/terraform_project$
```

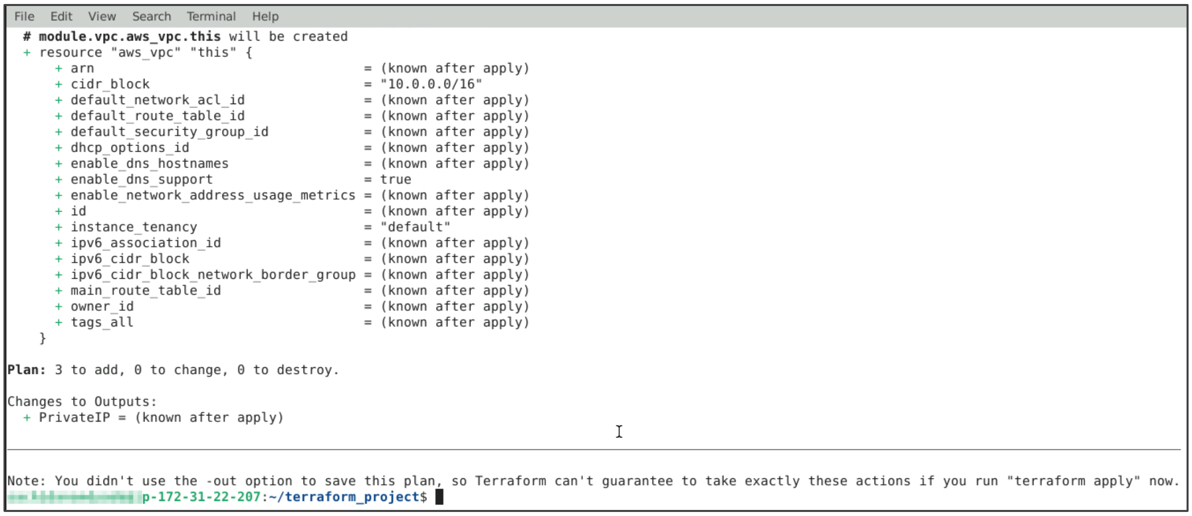
The code is successfully validated.

5.4 Execute the command given below to preview the actions that will be performed when you deploy the code:

terraform plan



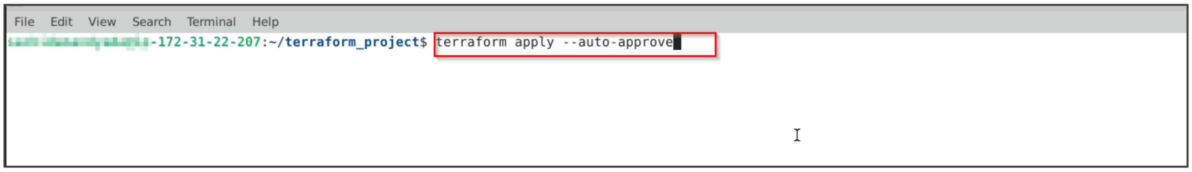
```
[redacted]-172-31-22-207:~/terraform_project$ terraform validate
Success! The configuration is valid.

[redacted]ip-172-31-22-207:~/terraform_project$ terraform plan
```



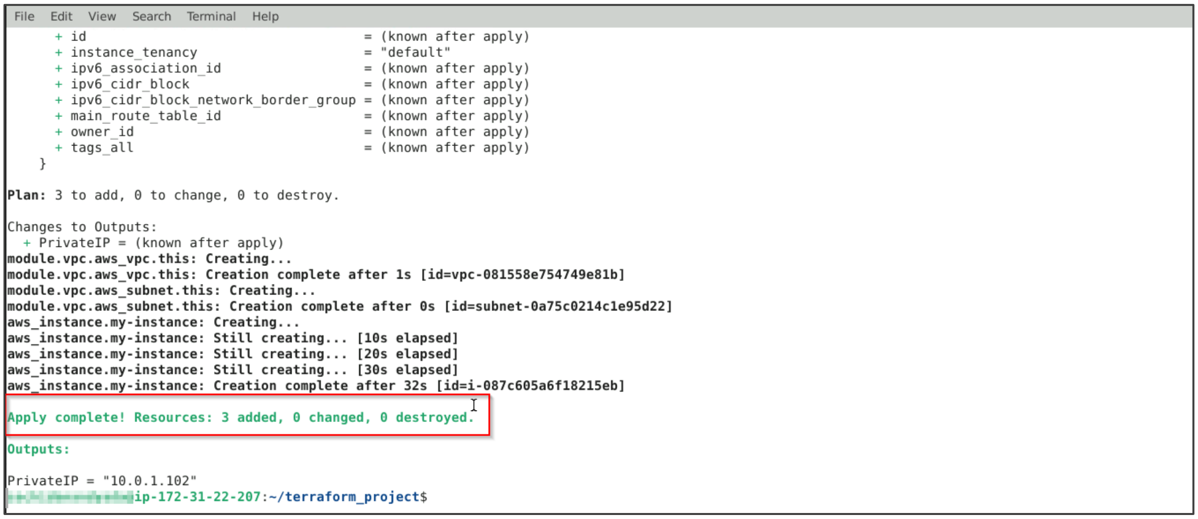
```
  # module.vpc.aws_vpc.this will be created
  + resource "aws_vpc" "this" {
      + arn                                  = (known after apply)
      + cidr_block                           = "10.0.0.0/16"
      + default_network_acl_id               = (known after apply)
      + default_route_table_id               = (known after apply)
      + default_security_group_id            = (known after apply)
      + dhcp_options_id                      = (known after apply)
      + enable_dns_hostnames                 = (known after apply)
      + enable_dns_support                   = true
      + enable_network_address_usage_metrics = (known after apply)
      + id                                   = (known after apply)
      + instance_tenancy                     = "default"
      + ipv6_association_id                  = (known after apply)
      + ipv6_cidr_block                      = (known after apply)
      + ipv6_cidr_block_network_border_group = (known after apply)
      + main_route_table_id                  = (known after apply)
      + owner_id                             = (known after apply)
      + tags_all                             = (known after apply)
    }

Plan: 3 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + PrivateIP = (known after apply)

─────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────

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.
[redacted]p-172-31-22-207:~/terraform_project$
```

The terraform plan command runs successfully.

5.5 Run the command given below to deploy the code:
terraform apply --auto-approve



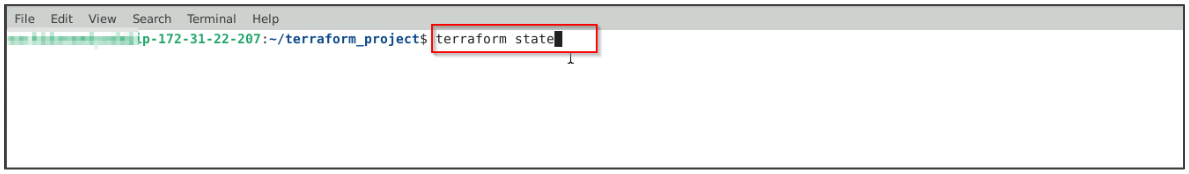
```
[redacted]-172-31-22-207:~/terraform_project$ terraform apply --auto-approve
```



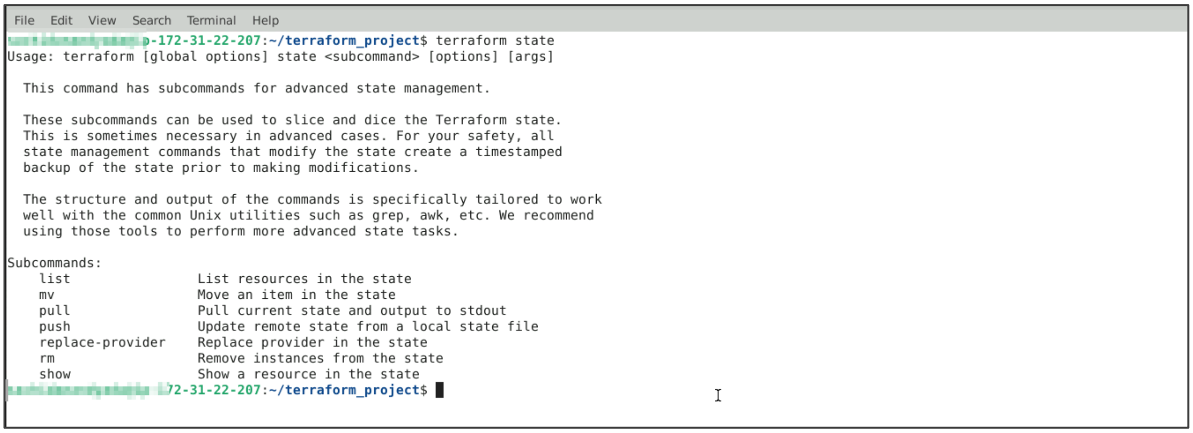
```
      + id                                   = (known after apply)
      + instance_tenancy                     = "default"
      + ipv6_association_id                  = (known after apply)
      + ipv6_cidr_block                      = (known after apply)
      + ipv6_cidr_block_network_border_group = (known after apply)
      + main_route_table_id                  = (known after apply)
      + owner_id                             = (known after apply)
      + tags_all                             = (known after apply)
    }

Plan: 3 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + PrivateIP = (known after apply)
module.vpc.aws_vpc.this: Creating...
module.vpc.aws_vpc.this: Creation complete after 1s [id=vpc-081558e754749e81b]
module.vpc.aws_subnet.this: Creating...
module.vpc.aws_subnet.this: Creation complete after 0s [id=subnet-0a75c0214c1e95d22]
aws_instance.my-instance: Creating...
aws_instance.my-instance: Still creating... [10s elapsed]
aws_instance.my-instance: Still creating... [20s elapsed]
aws_instance.my-instance: Still creating... [30s elapsed]
aws_instance.my-instance: Creation complete after 32s [id=i-087c605a6f18215eb]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

Outputs:

PrivateIP = "10.0.1.102"
[redacted]ip-172-31-22-207:~/terraform_project$
```

The **terraform apply --auto-approve** command runs successfully.

5.6 Execute the command given below to view the resources that are created:
terraform state



```
[redacted]ip-172-31-22-207:~/terraform_project$ terraform state
```



```
[redacted]p-172-31-22-207:~/terraform_project$ terraform state
Usage: terraform [global options] state <subcommand> [options] [args]

  This command has subcommands for advanced state management.

  These subcommands can be used to slice and dice the Terraform state.
  This is sometimes necessary in advanced cases. For your safety, all
  state management commands that modify the state create a timestamped
  backup of the state prior to making modifications.

  The structure and output of the commands is specifically tailored to work
  well with the common Unix utilities such as grep, awk, etc. We recommend
  using those tools to perform more advanced state tasks.

Subcommands:
    list                List resources in the state
    mv                  Move an item in the state
    pull                Pull current state and output to stdout
    push                Update remote state from a local state file
    replace-provider    Replace provider in the state
    rm                  Remove instances from the state
    show                Show a resource in the state
[redacted]72-31-22-207:~/terraform_project$
```

By following these steps, you will have successfully built and tested a Terraform module, ensuring efficient infrastructure management within AWS.