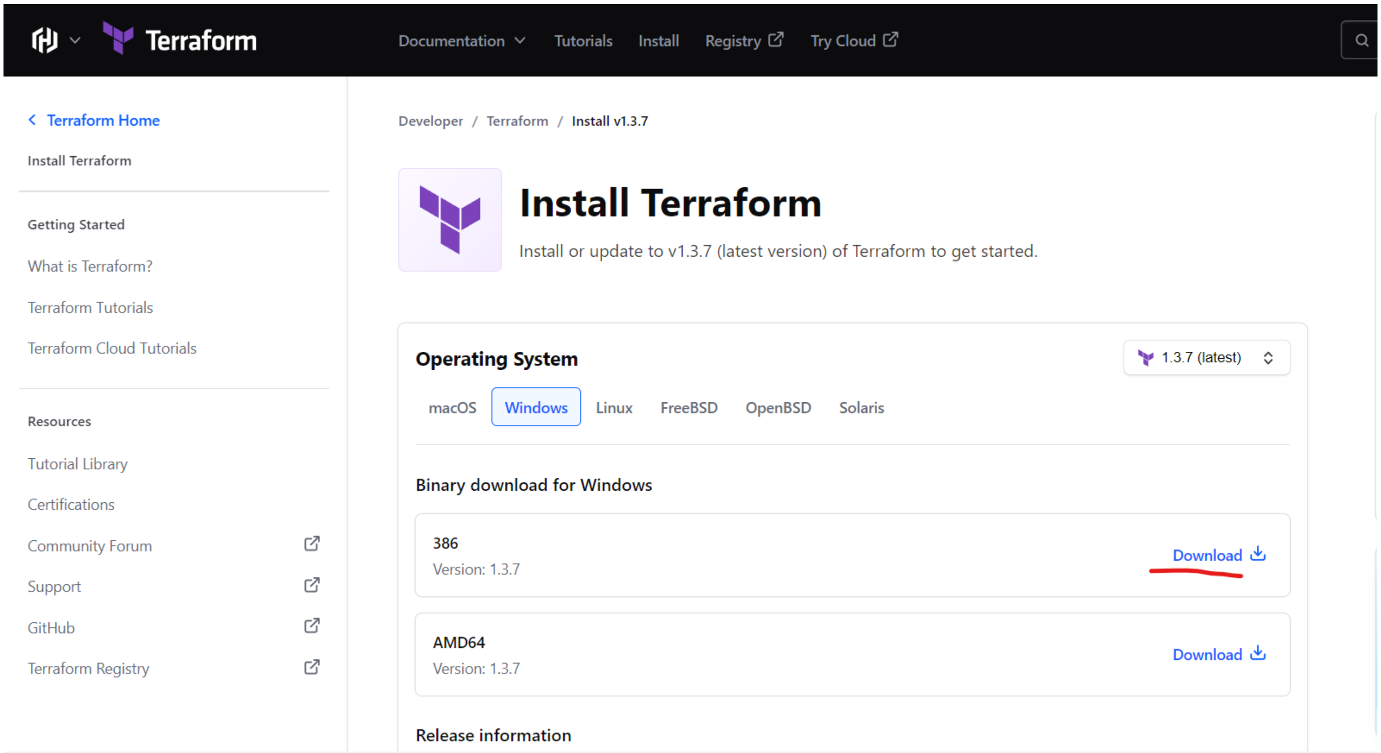
TERRAFORM LAB - By Kishan Ray

Installation of Terraform:-

To download Terraform click on this link :- Download Terraform



Install Terraform

Install or update to v1.3.7 (latest version) of Terraform to get started.

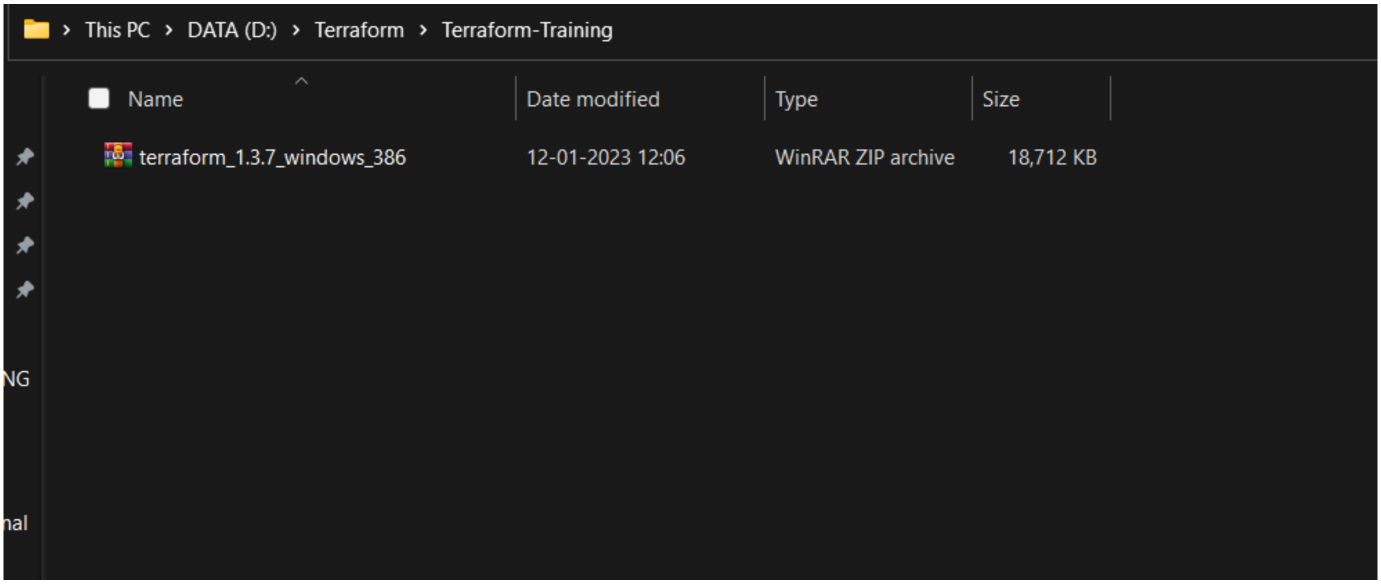
Operating System

Binary download for Windows

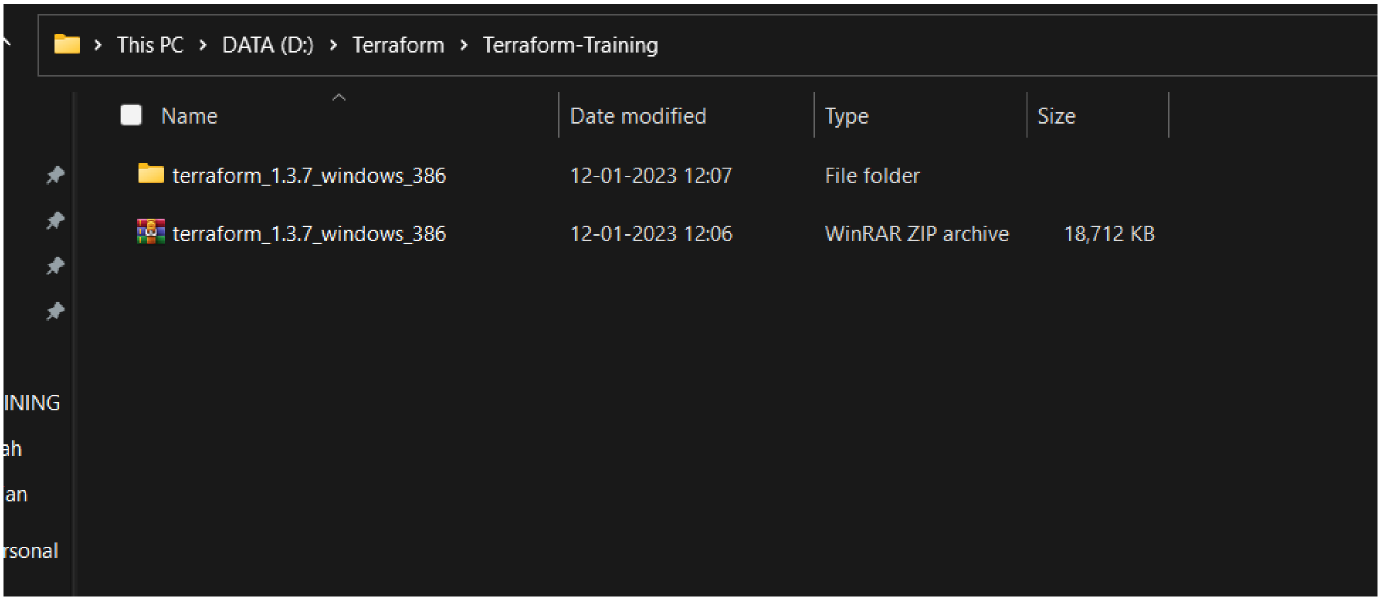
386
Version: 1.3.7

AMD64
Version: 1.3.7

Release information

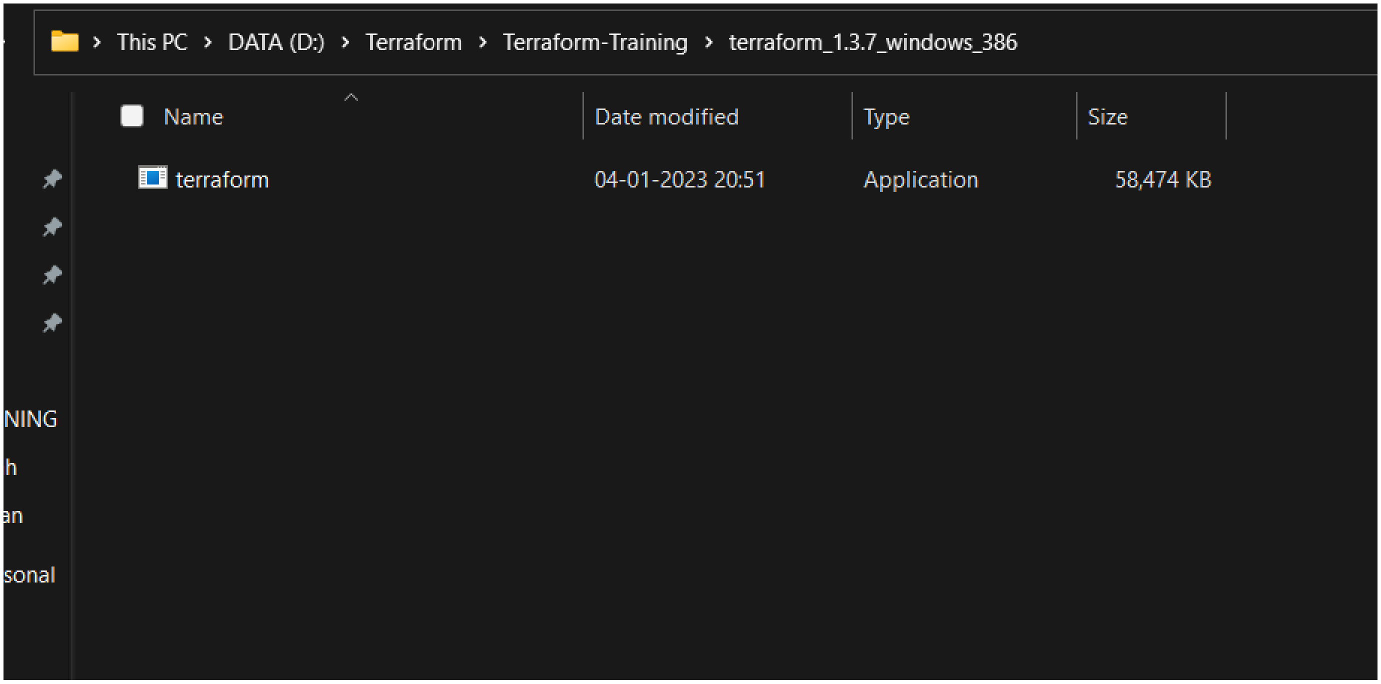


Name	Date modified	Type	Size
terraform_1.3.7_windows_386	12-01-2023 12:06	WinRAR ZIP archive	18,712 KB



This PC › DATA (D:) › Terraform › Terraform-Training

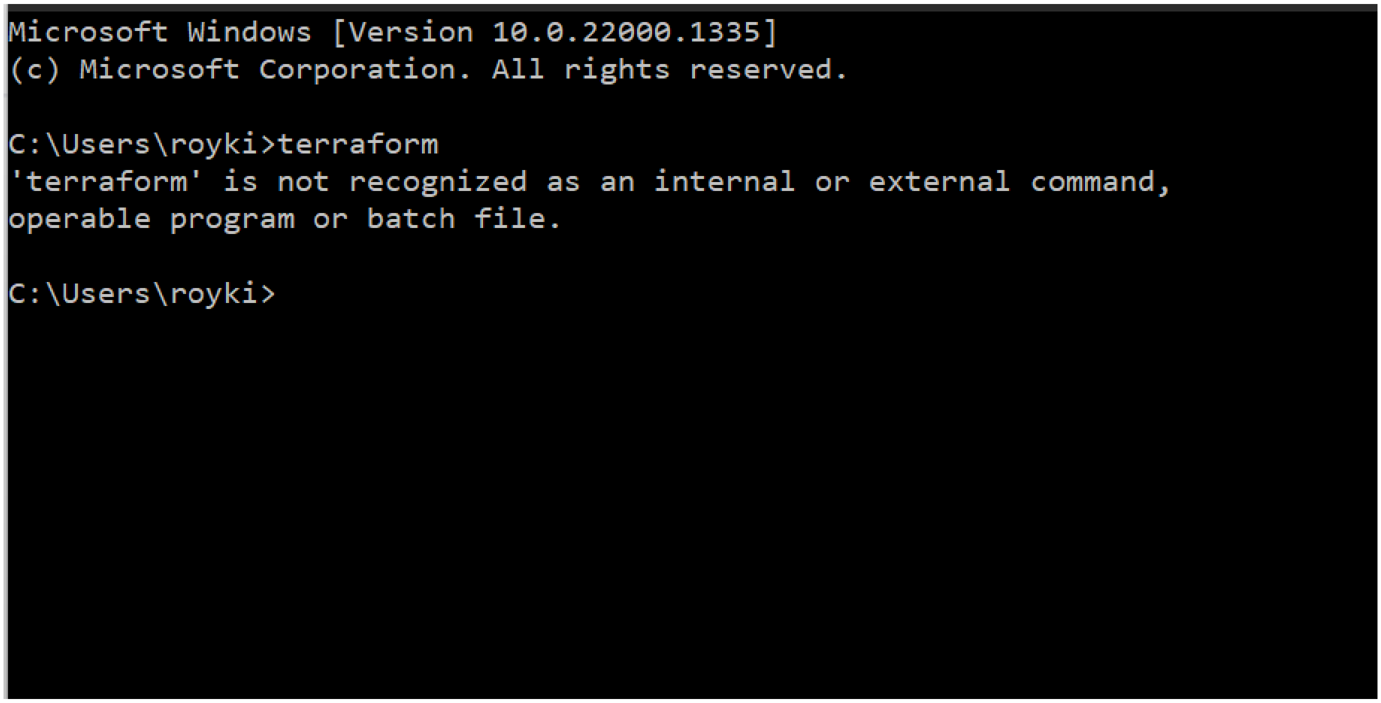
Name	Date modified	Type	Size
terraform_1.3.7_windows_386	12-01-2023 12:07	File folder	
terraform_1.3.7_windows_386	12-01-2023 12:06	WinRAR ZIP archive	18,712 KB



This PC › DATA (D:) › Terraform › Terraform-Training › terraform_1.3.7_windows_386

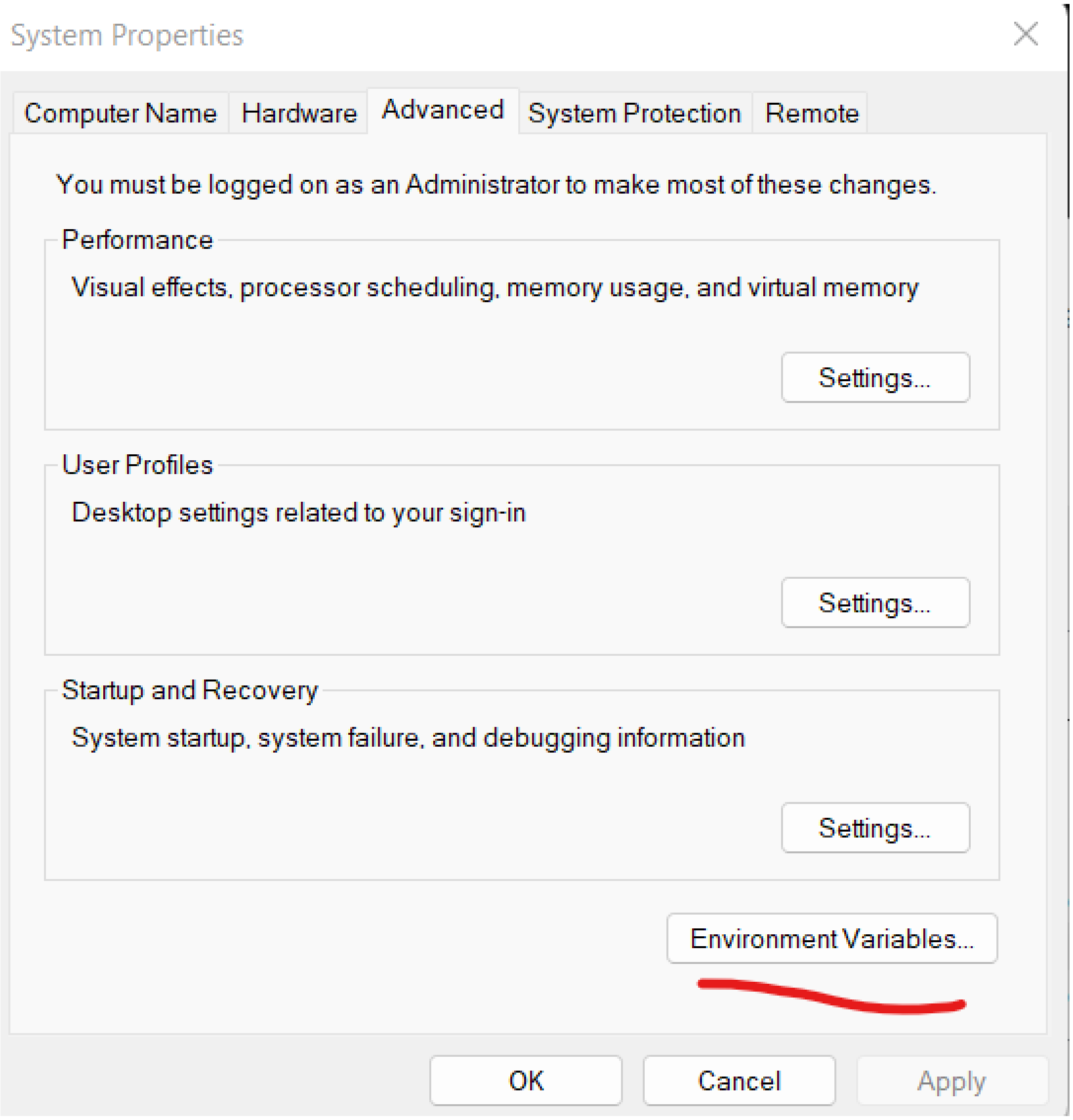
Name	Date modified	Type	Size
terraform	04-01-2023 20:51	Application	58,474 KB

Now Open “cmd” and try to use “terraform” command:



```
Microsoft Windows [Version 10.0.22000.1335]
(c) Microsoft Corporation. All rights reserved.

C:\Users\royki>terraform
'terraform' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\royki>
```

We have to add terraform as an Environment variable then only windows cmd will able to understand.

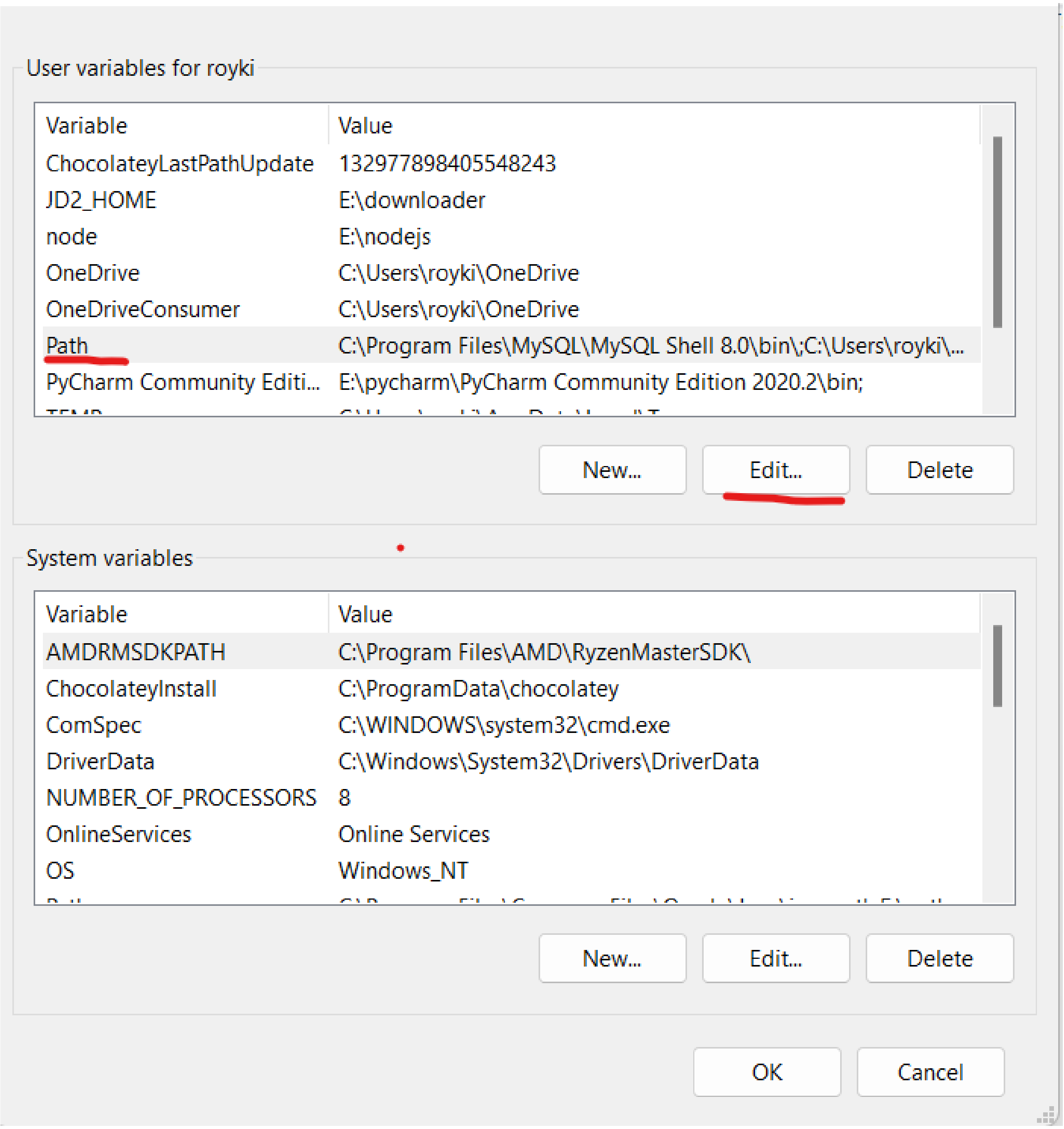


You must be logged on as an Administrator to make most of these changes.

Visual effects, processor scheduling, memory usage, and virtual memory

Desktop settings related to your sign-in

System startup, system failure, and debugging information

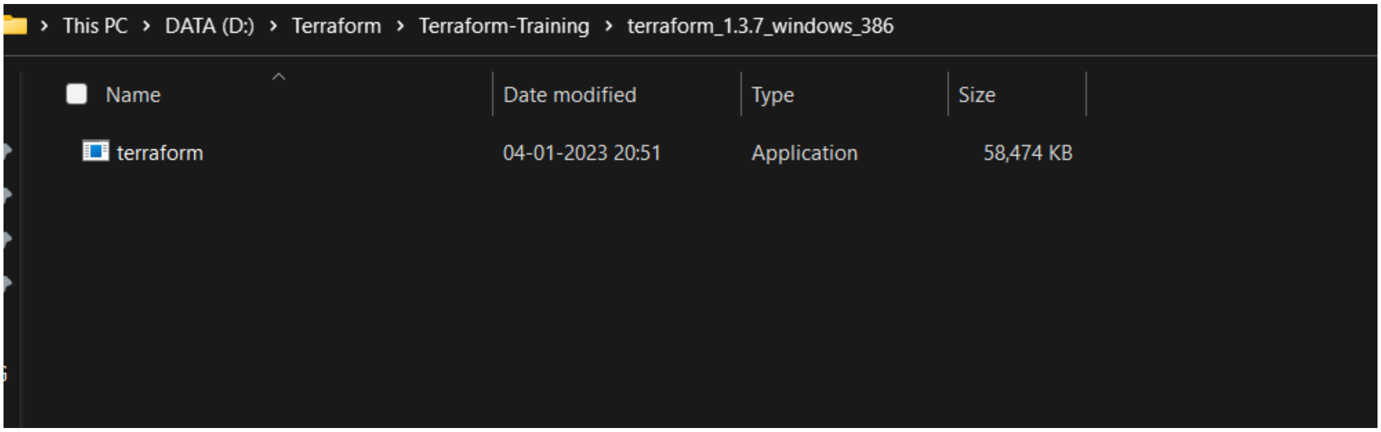


User variables for royki

Variable	Value
ChocolateyLastPathUpdate	132977898405548243
JD2_HOME	E:\downloader
node	E:\nodejs
OneDrive	C:\Users\royki\OneDrive
OneDriveConsumer	C:\Users\royki\OneDrive
Path	C:\Program Files\MySQL\MySQL Shell 8.0\bin\;C:\Users\royki\...
PyCharm Community Editi...	E:\pycharm\PyCharm Community Edition 2020.2\bin;

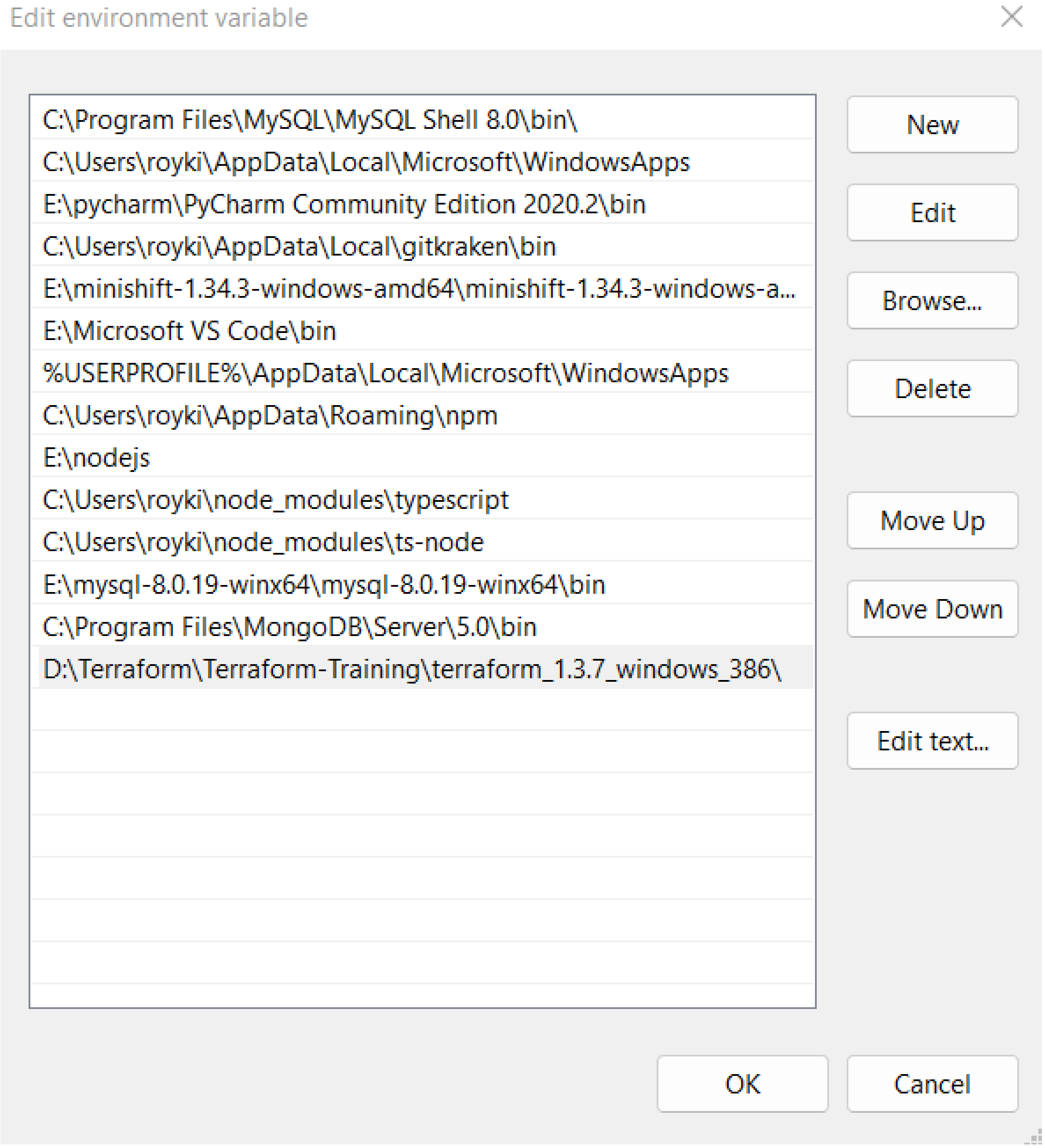
System variables

Variable	Value
AMDRMSDKPATH	C:\Program Files\AMD\RyzenMasterSDK\
ChocolateyInstall	C:\ProgramData\chocolatey
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	8
OnlineServices	Online Services
OS	Windows_NT



This PC › DATA (D:) › Terraform › Terraform-Training › terraform_1.3.7_windows_386

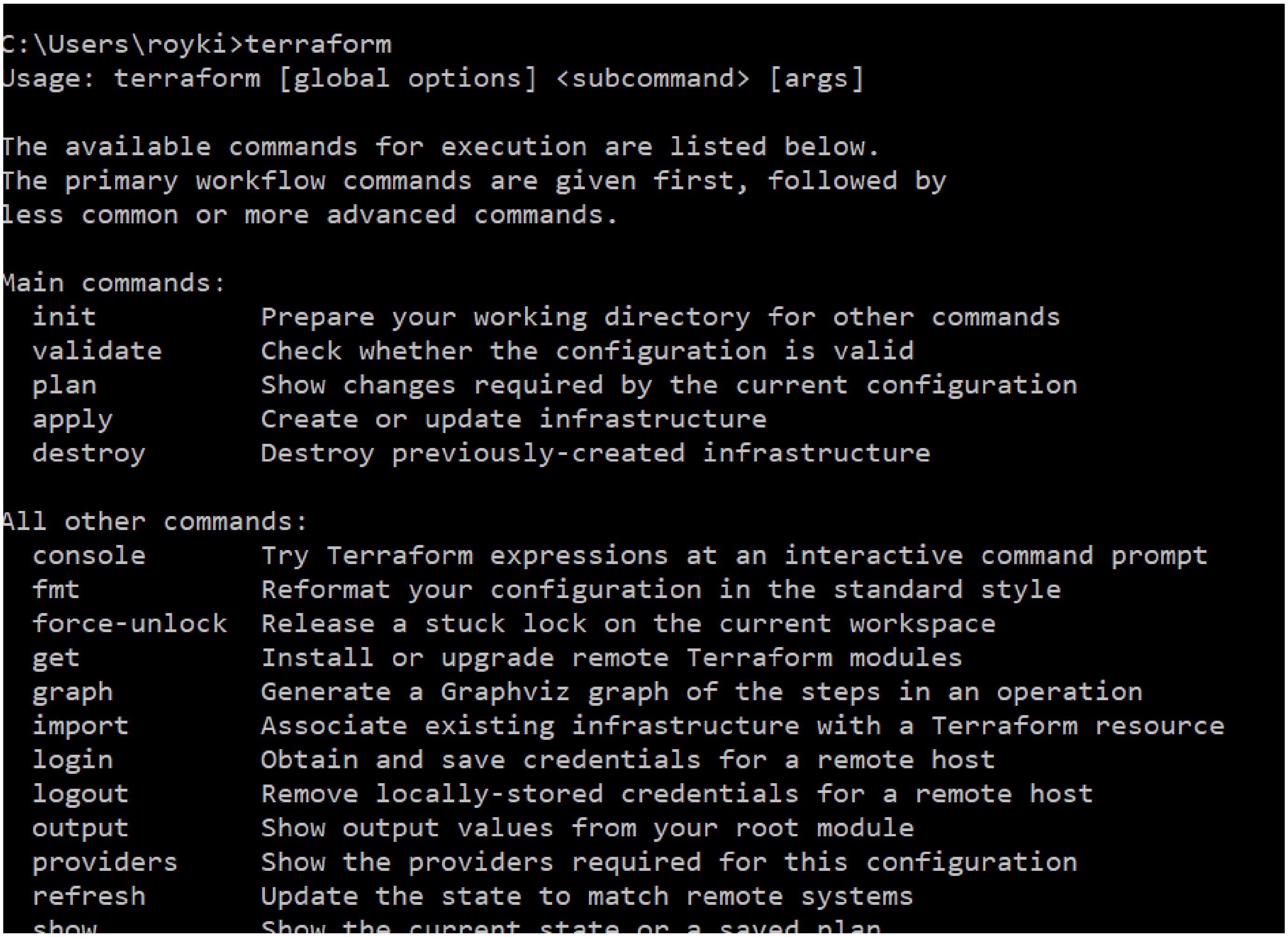
Name	Date modified	Type	Size
terraform	04-01-2023 20:51	Application	58,474 KB



Edit environment variable

```
C:\Program Files\MySQL\MySQL Shell 8.0\bin\
C:\Users\royki\AppData\Local\Microsoft\WindowsApps
E:\pycharm\PyCharm Community Edition 2020.2\bin
C:\Users\royki\AppData\Local\gitkraken\bin
E:\minishift-1.34.3-windows-amd64\minishift-1.34.3-windows-a...
E:\Microsoft VS Code\bin
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\Users\royki\AppData\Roaming\npm
E:\nodejs
C:\Users\royki\node_modules\typescript
C:\Users\royki\node_modules\ts-node
E:\mysql-8.0.19-winx64\mysql-8.0.19-winx64\bin
C:\Program Files\MongoDB\Server\5.0\bin
D:\Terraform\Terraform-Training\terraform_1.3.7_windows_386\
```

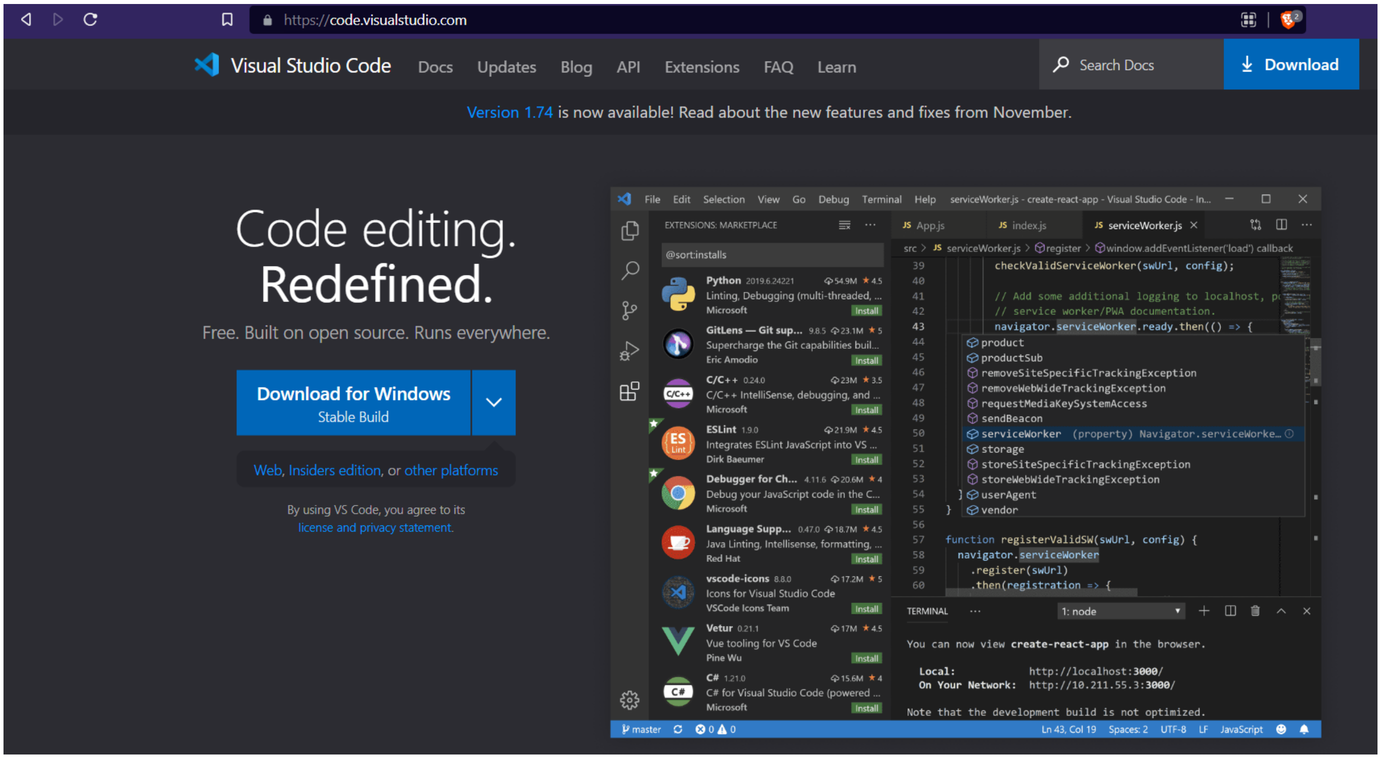
After path again enter “terraform” in cmd.



```
C:\Users\royki>terraform
Usage: terraform [global options] <subcommand> [args]

The available commands for execution are listed below.
The primary workflow commands are given first, followed by
less common or more advanced commands.

Main commands:
  init          Prepare your working directory for other commands
  validate      Check whether the configuration is valid
  plan          Show changes required by the current configuration
  apply         Create or update infrastructure
  destroy       Destroy previously-created infrastructure

All other commands:
  console       Try Terraform expressions at an interactive command prompt
  fmt           Reformat your configuration in the standard style
  force-unlock  Release a stuck lock on the current workspace
  get           Install or upgrade remote Terraform modules
  graph         Generate a Graphviz graph of the steps in an operation
  import        Associate existing infrastructure with a Terraform resource
  login         Obtain and save credentials for a remote host
  logout        Remove locally-stored credentials for a remote host
  output        Show output values from your root module
  providers     Show the providers required for this configuration
  refresh       Update the state to match remote systems
  show          Show the current state or a saved plan
```

Install VS Code Editor

Download it from this link.


https://code.visualstudio.com
Visual Studio Code
Docs
Updates
Blog
API
Extensions
FAQ
Learn
Search Docs
Download
Download for Windows
Stable Build
Web, Insiders edition, or other platforms
File Edit Selection View Go Debug Terminal Help
serviceWorker.js - create-react-app - Visual Studio Code - In...
EXTENSIONS: MARKETPLACE
@sort:installs
Python 2019.6.24221
Linting, Debugging (multi-threaded, ...
Microsoft
GitLens — Git sup... 9.8.5
Supercharge the Git capabilities buil...
Eric Amodio
C/C++ 0.24.0
C/C++ IntelliSense, debugging, and ...
Microsoft
ESLint 1.9.0
Integrates ESLint JavaScript into VS ...
Dirk Baeumer
Debugger for Ch... 4.11.6
Debug your JavaScript code in the C...
Microsoft
Language Supp... 0.47.0
Java Linting, Intellisense, formatting, ...
Red Hat
vscode-icons 8.8.0
Icons for Visual Studio Code
VSCode Icons Team
Vetur 0.21.1
Vue tooling for VS Code
Pine Wu
C# 1.21.0
C# for Visual Studio Code (powered ...
Microsoft
Install
App.js
index.js
serviceWorker.js
src > serviceWorker.js > register > window.addEventListener('load') callback
TERMINAL
1: node
master
Ln 43, Col 19 Spaces: 2 UTF-8 LF JavaScript

Version 1.74 is now available! Read about the new features and fixes from November.

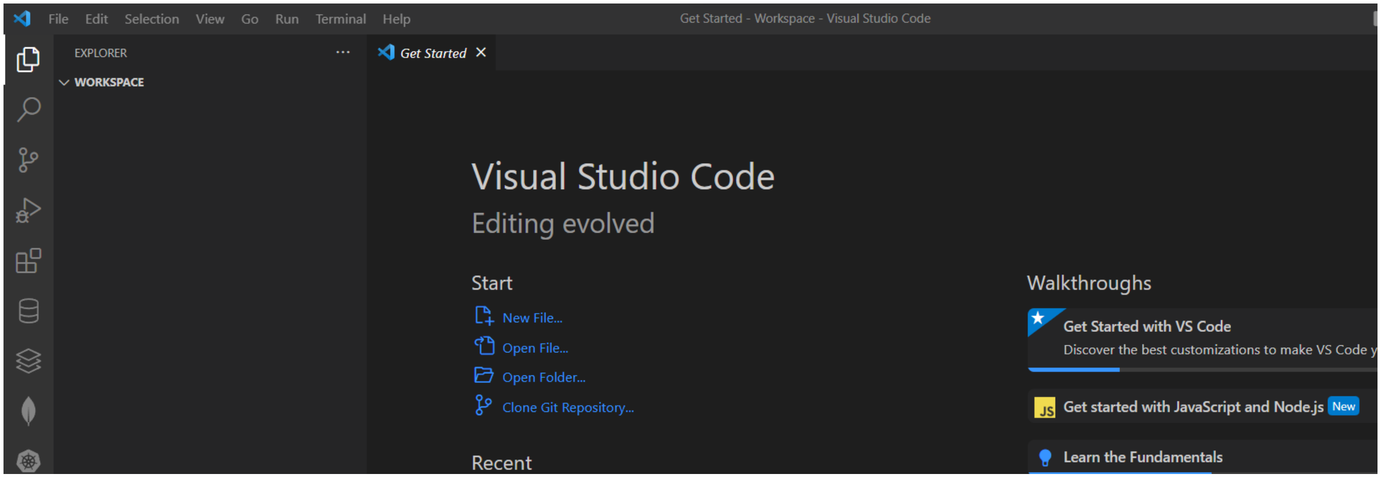
Code editing. Redefined.

Free. Built on open source. Runs everywhere.

By using VS Code, you agree to its license and privacy statement.

```
        checkValidServiceWorker(swUrl, config);

        // Add some additional logging to localhost, po
        // service worker/PWA documentation.
        navigator.serviceWorker.ready.then(() => {
          product
          productSub
          removeSiteSpecificTrackingException
          removeWebWideTrackingException
          requestMediaKeySystemAccess
          sendBeacon
          serviceWorker  (property) Navigator.serviceWorke...
          storage
          storeSiteSpecificTrackingException
          storeWebWideTrackingException
      } userAgent
    }   vendor

function registerValidSW(swUrl, config) {
  navigator.serviceWorker
    .register(swUrl)
    .then(registration => {
```

```
You can now view create-react-app in the browser.

  Local:            http://localhost:3000/
  On Your Network:  http://10.211.55.3:3000/

Note that the development build is not optimized.
```



Visual Studio Code

Editing evolved

Start

Walkthroughs

Get Started with VS Code

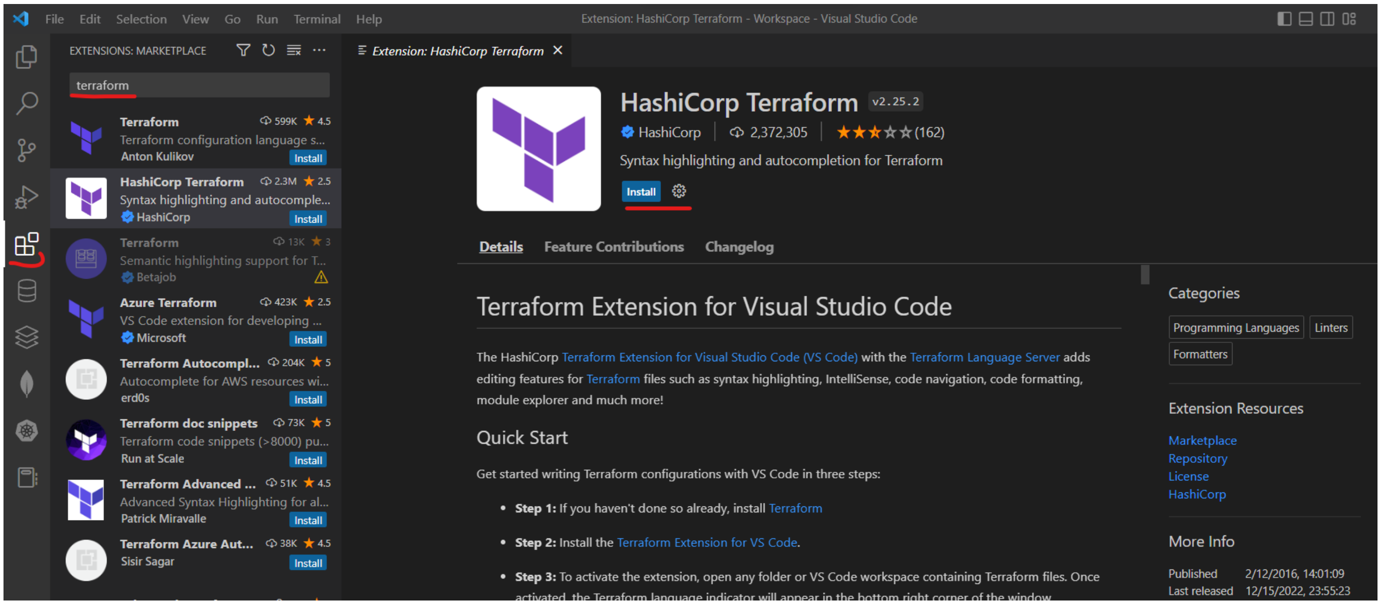
Discover the best customizations to make VS Code y

Get started with JavaScript and Node.js

Learn the Fundamentals

Recent

Install terraform extension


File Edit Selection View Go Run Terminal Help
Extension: HashiCorp Terraform - Workspace - Visual Studio Code
EXTENSIONS: MARKETPLACE
terraform
Terraform 599K ★4.5
Terraform configuration language s...
Anton Kulikov
Install
HashiCorp Terraform 2.3M ★2.5
Syntax highlighting and autocomple...
HashiCorp
Install
Terraform 13K ★3
Semantic highlighting support for T...
Betajob
Azure Terraform 423K ★2.5
VS Code extension for developing ...
Microsoft
Install
Terraform Autocompl... 204K ★5
Autocomplete for AWS resources wi...
erd0s
Install
Terraform doc snippets 73K ★5
Terraform code snippets (>8000) pu...
Run at Scale
Install
Terraform Advanced ... 51K ★4.5
Advanced Syntax Highlighting for al...
Patrick Miravalle
Install
Terraform Azure Aut... 38K ★4.5
Sisir Sagar
Install
Extension: HashiCorp Terraform
HashiCorp Terraform v2.25.2
HashiCorp | 2,372,305 | ★★★☆☆ (162)
Syntax highlighting and autocompletion for Terraform
Install
Details Feature Contributions Changelog
Categories
Programming Languages Linters Formatters
Extension Resources
Marketplace
Repository
License
HashiCorp
More Info
Published 2/12/2016, 14:01:09
Last released 12/15/2022, 23:55:23

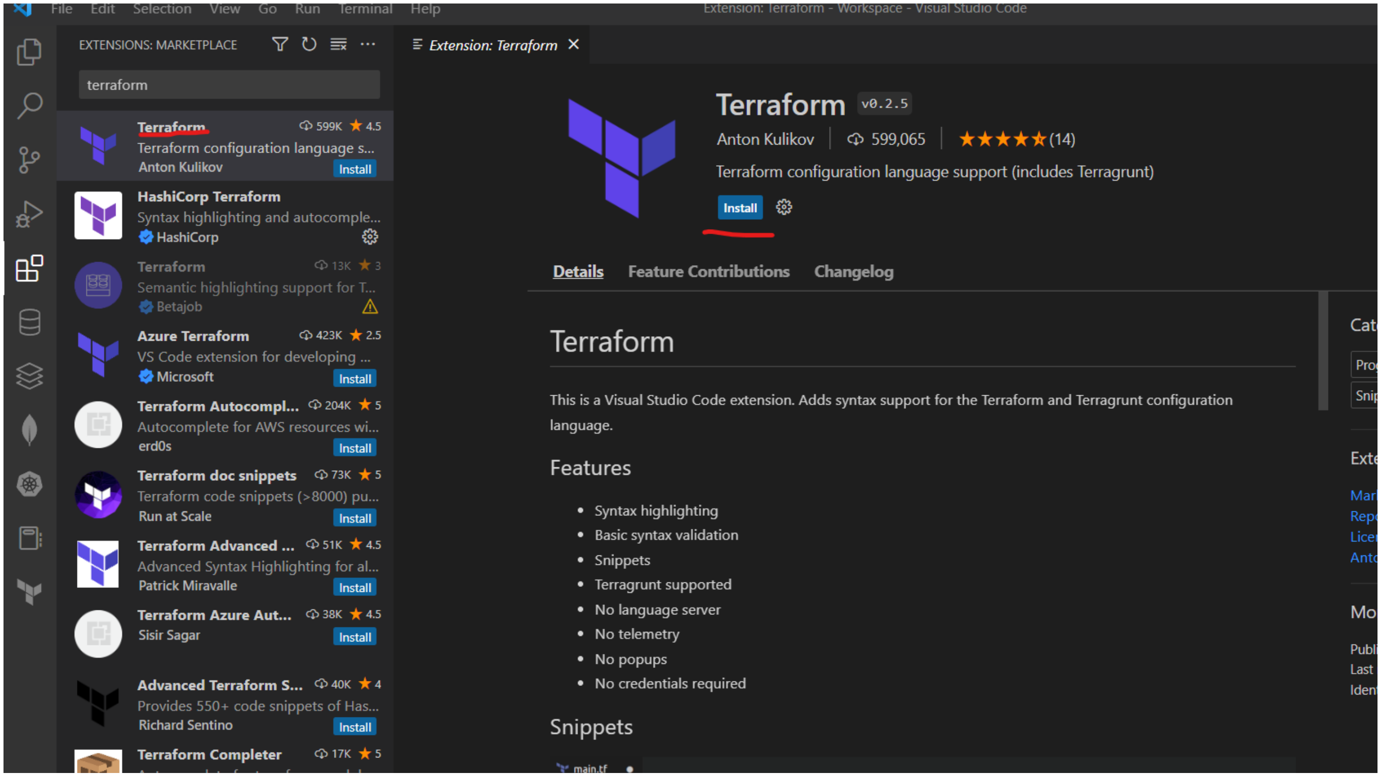
Terraform Extension for Visual Studio Code

The HashiCorp Terraform Extension for Visual Studio Code (VS Code) with the Terraform Language Server adds editing features for Terraform files such as syntax highlighting, IntelliSense, code navigation, code formatting, module explorer and much more!

Quick Start

Get started writing Terraform configurations with VS Code in three steps:

- **Step 1:** If you haven't done so already, install Terraform
- **Step 2:** Install the Terraform Extension for VS Code.
- **Step 3:** To activate the extension, open any folder or VS Code workspace containing Terraform files. Once activated, the Terraform language indicator will appear in the bottom right corner of the window.


File Edit Selection View Go Run Terminal Help
Extension: Terraform - Workspace - Visual Studio Code
EXTENSIONS: MARKETPLACE
terraform
Terraform 599K ★4.5
Terraform configuration language s...
Anton Kulikov
Install
HashiCorp Terraform
Syntax highlighting and autocomple...
HashiCorp
Terraform 13K ★3
Semantic highlighting support for T...
Betajob
Azure Terraform 423K ★2.5
VS Code extension for developing ...
Microsoft
Install
Terraform Autocompl... 204K ★5
Autocomplete for AWS resources wi...
erd0s
Install
Terraform doc snippets 73K ★5
Terraform code snippets (>8000) pu...
Run at Scale
Install
Terraform Advanced ... 51K ★4.5
Advanced Syntax Highlighting for al...
Patrick Miravalle
Install
Terraform Azure Aut... 38K ★4.5
Sisir Sagar
Install
Advanced Terraform S... 40K ★4
Provides 550+ code snippets of Has...
Richard Sentino
Install
Terraform Completer 17K ★5
Extension: Terraform
Terraform v0.2.5
Anton Kulikov | 599,065 | ★★★★★ (14)
Terraform configuration language support (includes Terragrunt)
Install
Details Feature Contributions Changelog
Cat
Pro
Snip
Exte
Mar
Rep
Lice
Anto
Mo
Publi
Last
Iden
main.tf

Terraform

This is a Visual Studio Code extension. Adds syntax support for the Terraform and Terragrunt configuration language.

Features

- Syntax highlighting
- Basic syntax validation
- Snippets
- Terragrunt supported
- No language server
- No telemetry
- No popups
- No credentials required

Snippets

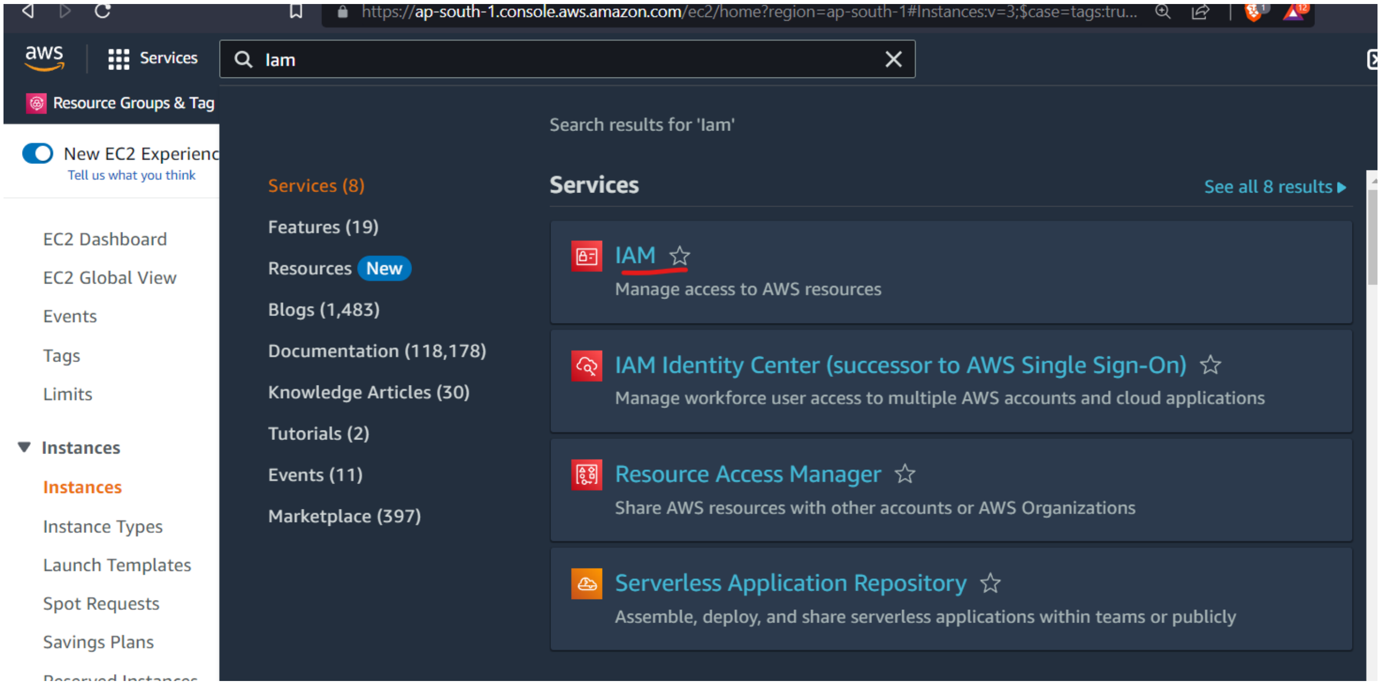
once extensions are installed then you are ready to code .

Install AWS CLI

To install aws-cli click on this link .

Once installed , create a new IAM user with power user permission and configure.

To create IAM user:-



Iam

Search results for 'Iam'

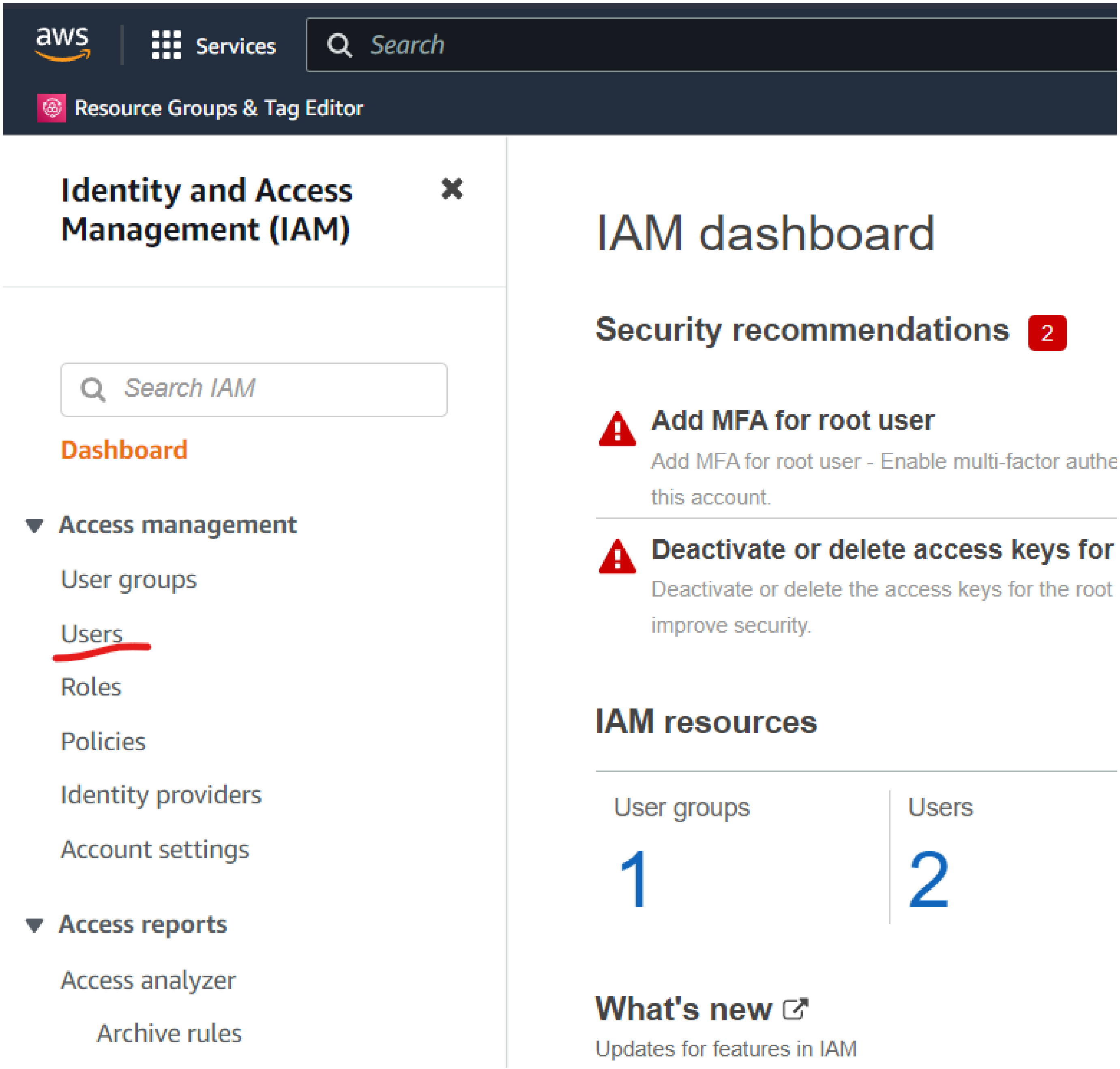
Services

IAM
Manage access to AWS resources

IAM Identity Center (successor to AWS Single Sign-On)
Manage workforce user access to multiple AWS accounts and cloud applications

Resource Access Manager
Share AWS resources with other accounts or AWS Organizations

Serverless Application Repository
Assemble, deploy, and share serverless applications within teams or publicly



IAM dashboard

Security recommendations 2

Add MFA for root user

Add MFA for root user - Enable multi-factor authe
this account.

Deactivate or delete access keys for

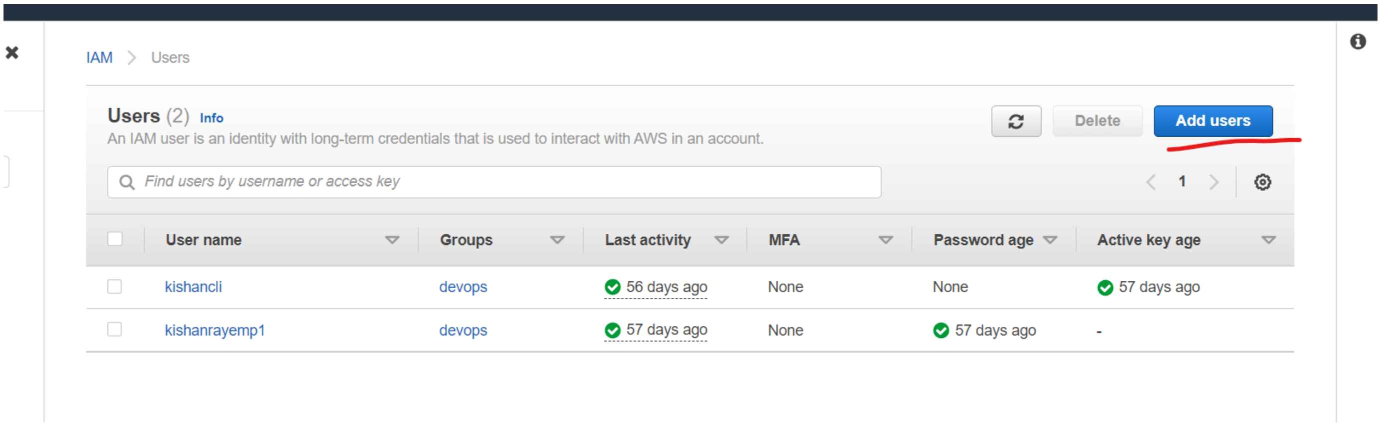
Deactivate or delete the access keys for the root
improve security.

IAM resources

User groups	Users
1	2

What's new

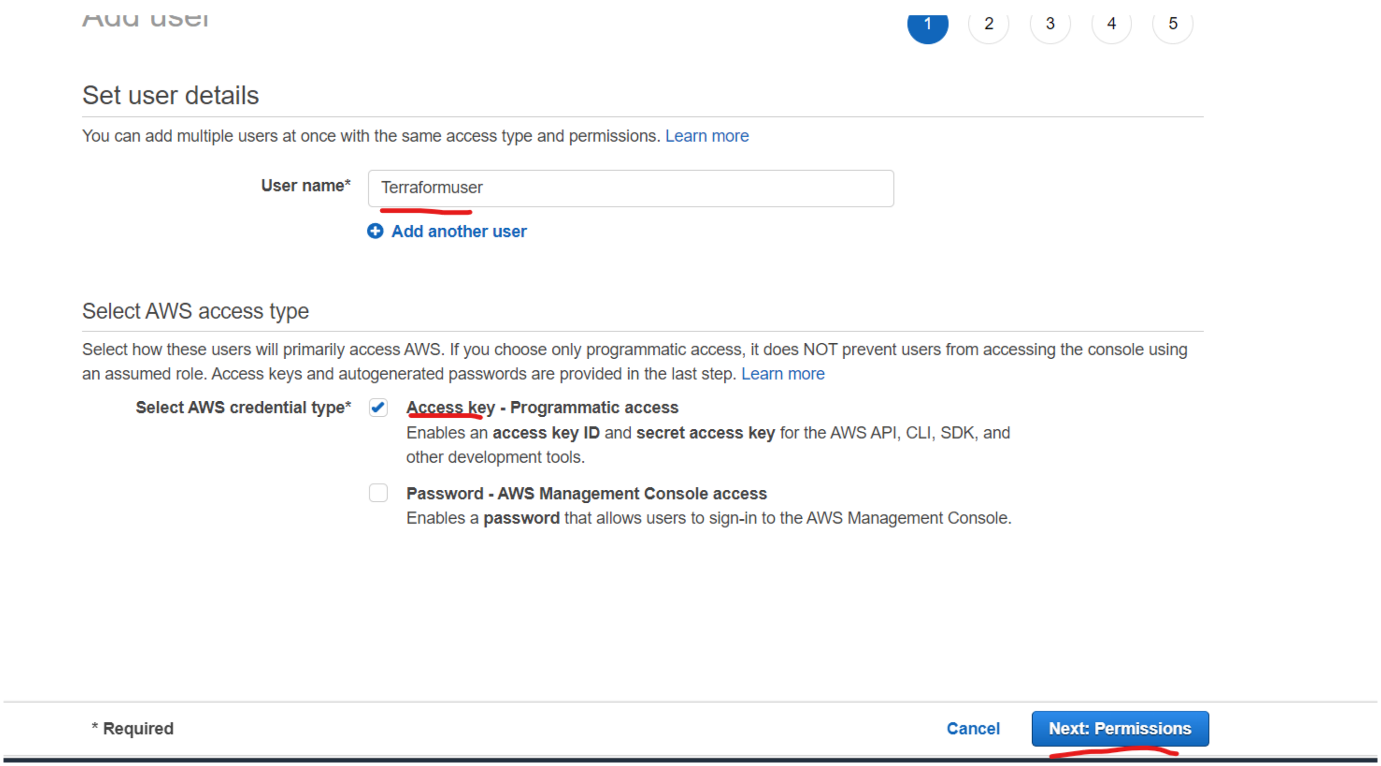
Updates for features in IAM



Users (2) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

	User name	Groups	Last activity	MFA	Password age	Active key age
□	kishancli	devops	✓ 56 days ago	None	None	✓ 57 days ago
□	kishanrayemp1	devops	✓ 57 days ago	None	✓ 57 days ago	-



Add user

Set user details

You can add multiple users at once with the same access type and permissions. Learn more

User name* Terraformuser

Select AWS access type

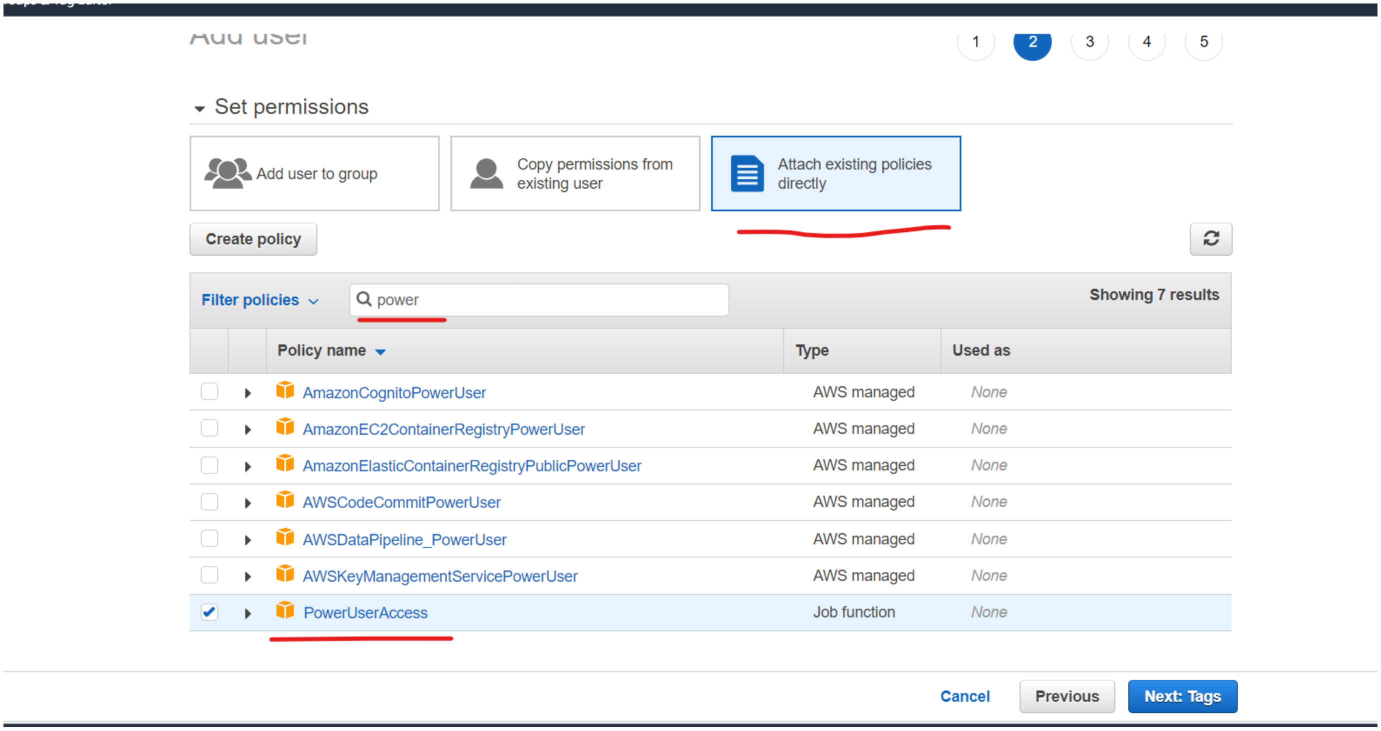
Select how these users will primarily access AWS. If you choose only programmatic access, it does NOT prevent users from accessing the console using an assumed role. Access keys and autogenerated passwords are provided in the last step. Learn more

Select AWS credential type*

☑ **Access key - Programmatic access**
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

□ **Password - AWS Management Console access**
Enables a **password** that allows users to sign-in to the AWS Management Console.

* **Required**



Add user

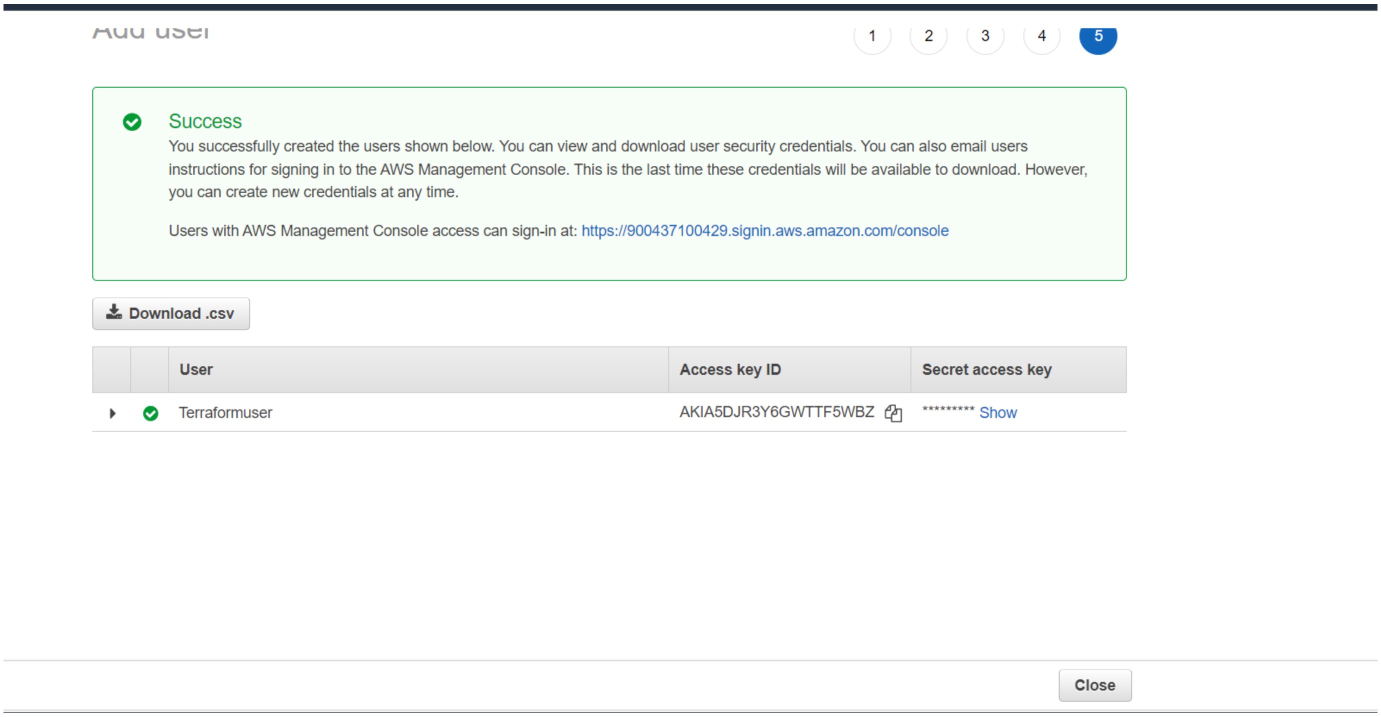
Set permissions

Add user to group
Copy permissions from existing user
Attach existing policies directly
Create policy
Filter policies

power

Showing 7 results

		Policy name	Type	Used as
□	▸	AmazonCognitoPowerUser	AWS managed	*None*
□	▸	AmazonEC2ContainerRegistryPowerUser	AWS managed	*None*
□	▸	AmazonElasticContainerRegistryPublicPowerUser	AWS managed	*None*
□	▸	AWSCodeCommitPowerUser	AWS managed	*None*
□	▸	AWSDataPipeline_PowerUser	AWS managed	*None*
□	▸	AWSKeyManagementServicePowerUser	AWS managed	*None*
☑	▸	PowerUserAccess	Job function	*None*



Add user

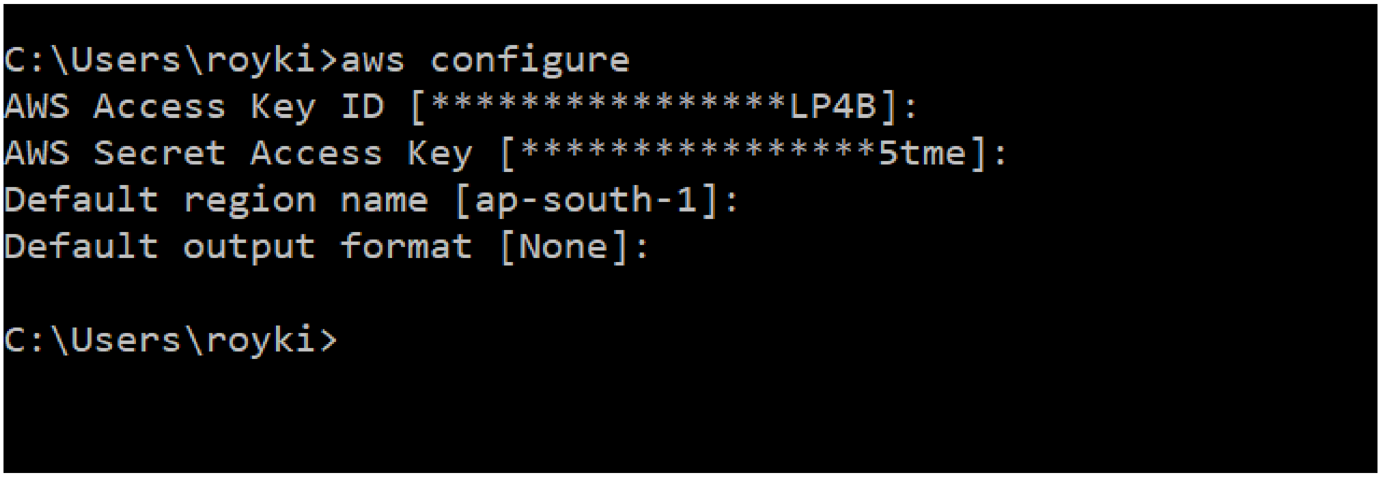
Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: https://900437100429.signin.aws.amazon.com/console

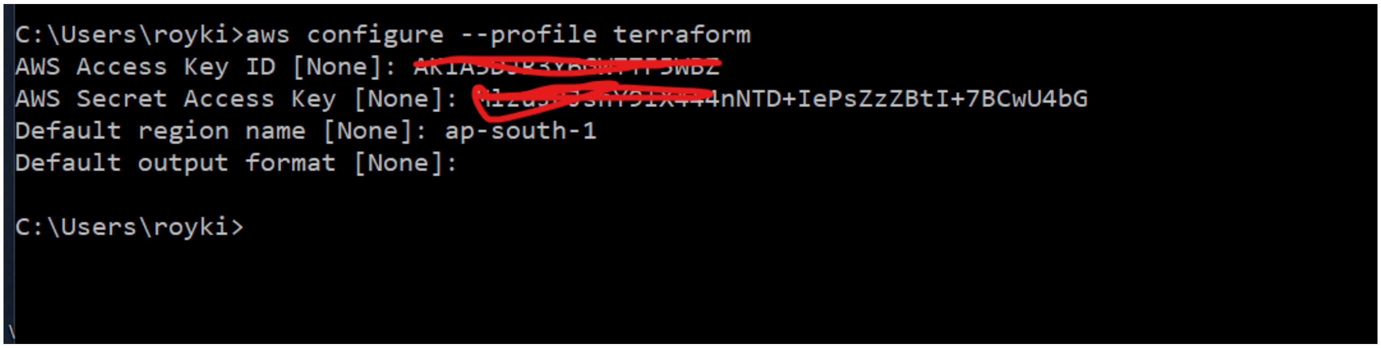
User	Access key ID	Secret access key
Terraformuser	AKIA5DJR3Y6GWTTF5WBZ	********* Show

```
aws configure   //default profile
```



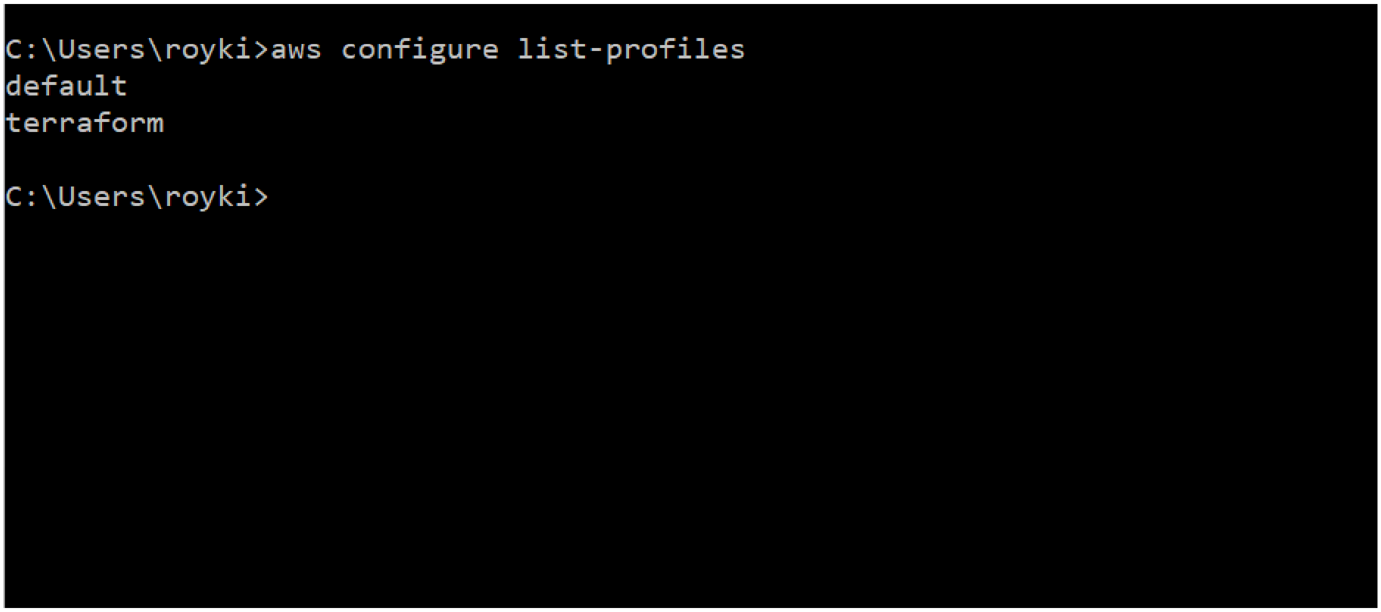
```
C:\Users\royki>aws configure
AWS Access Key ID [****************LP4B]:
AWS Secret Access Key [****************5tme]:
Default region name [ap-south-1]:
Default output format [None]:

C:\Users\royki>
```

```
aws configure --profile <"name">  //custom profile
```



```
C:\Users\royki>aws configure --profile terraform
AWS Access Key ID [None]: [redacted]
AWS Secret Access Key [None]: [redacted]4nNTD+IePsZzZBtI+7BCwU4bG
Default region name [None]: ap-south-1
Default output format [None]:

C:\Users\royki>
```

```
aws configure list-profiles
```



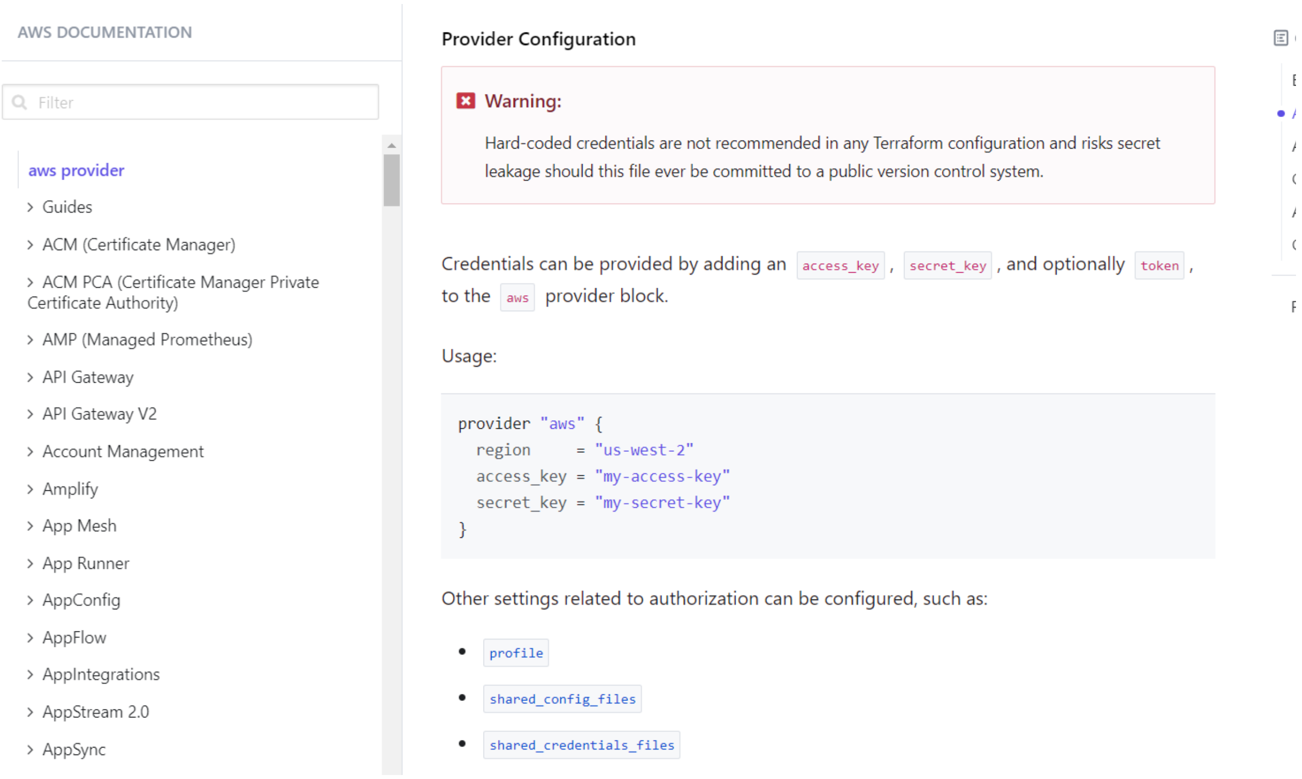
```
C:\Users\royki>aws configure list-profiles
default
terraform

C:\Users\royki>
```

To Launch an EC2-Instance using Terraform :-

Requirements:-

Create a workspace with name “Ec2-instance” in your vs code editor

Setup provider.tf using link:- https://registry.terraform.io/providers/hashicorp/aws/latest/docs



Provider Configuration

Warning:

Hard-coded credentials are not recommended in any Terraform configuration and risks secret leakage should this file ever be committed to a public version control system.

Credentials can be provided by adding an `access_key` , `secret_key` , and optionally `token` , to the `aws` provider block.

Usage:

```
provider "aws" {
  region     = "us-west-2"
  access_key = "my-access-key"
  secret_key = "my-secret-key"
}
```

Other settings related to authorization can be configured, such as:

- `profile`
- `shared_config_files`
- `shared_credentials_files`

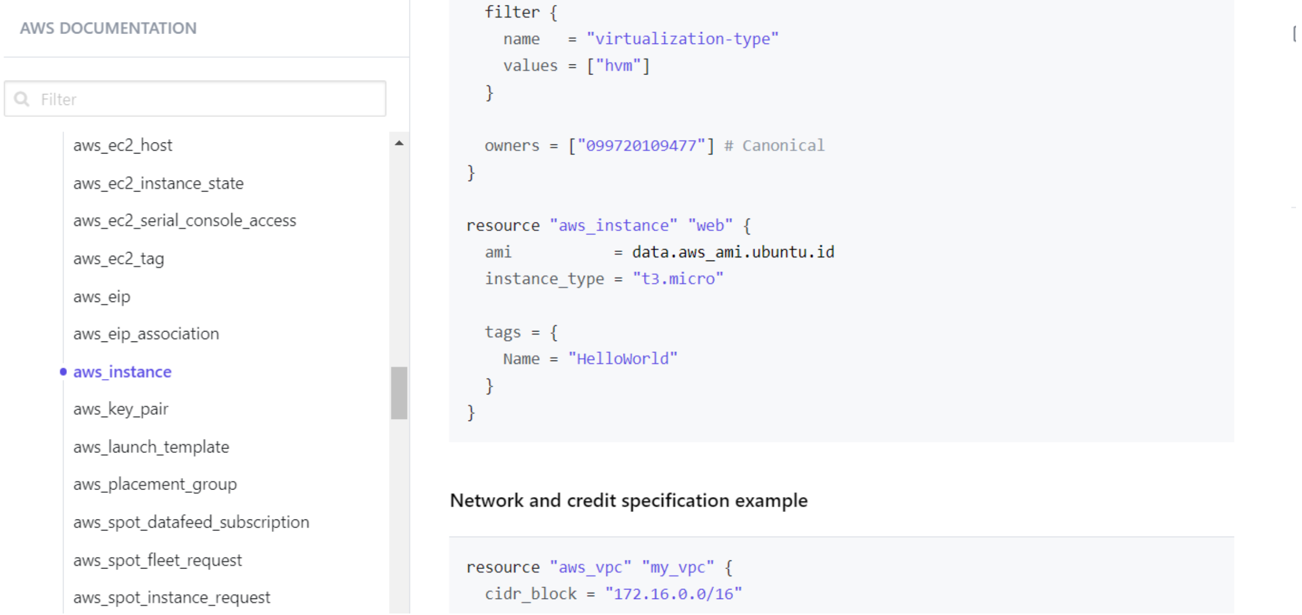
In above screenshot we are hardcoding the values of access and secret , but this is not recommended method.

So, we are going to use the profile method.

Provider.tf This file will help us to download the plugin related to provider.

```
provider "aws" {
  region = "ap-south-1"
  profile = "terraform"  //profile we created in previous step
}
```

refer:- https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/instance



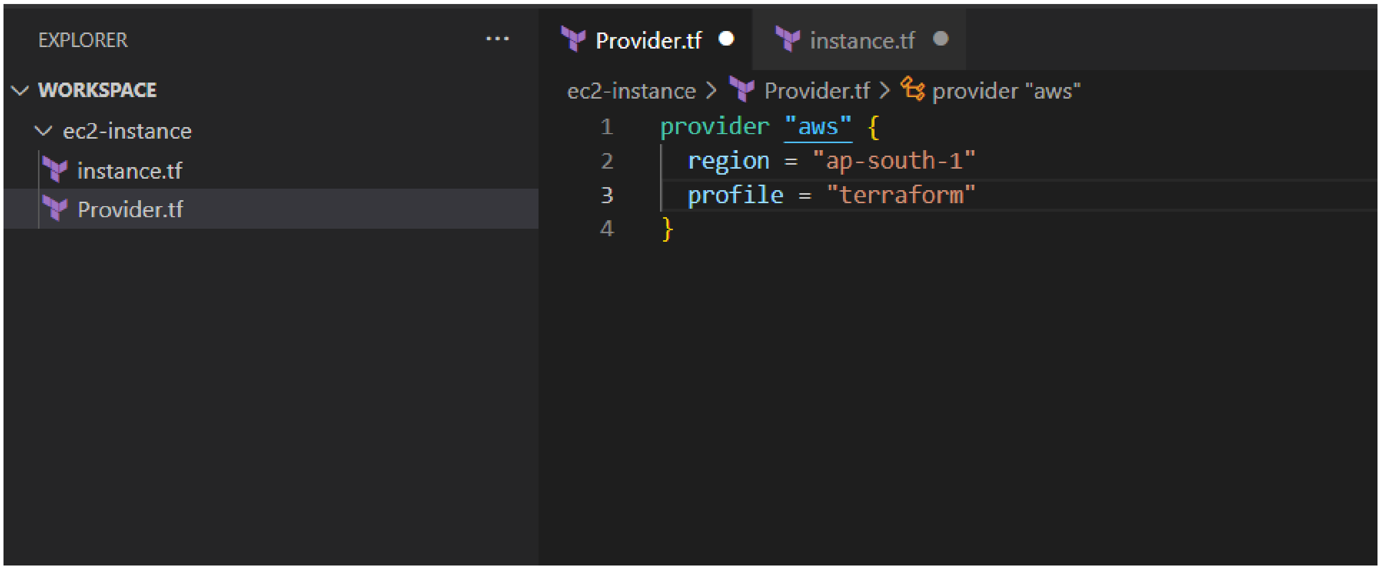
```
  filter {
    name   = "virtualization-type"
    values = ["hvm"]
  }

  owners = ["099720109477"] # Canonical
}

resource "aws_instance" "web" {
  ami           = data.aws_ami.ubuntu.id
  instance_type = "t3.micro"

  tags = {
    Name = "HelloWorld"
  }
}
```

Network and credit specification example

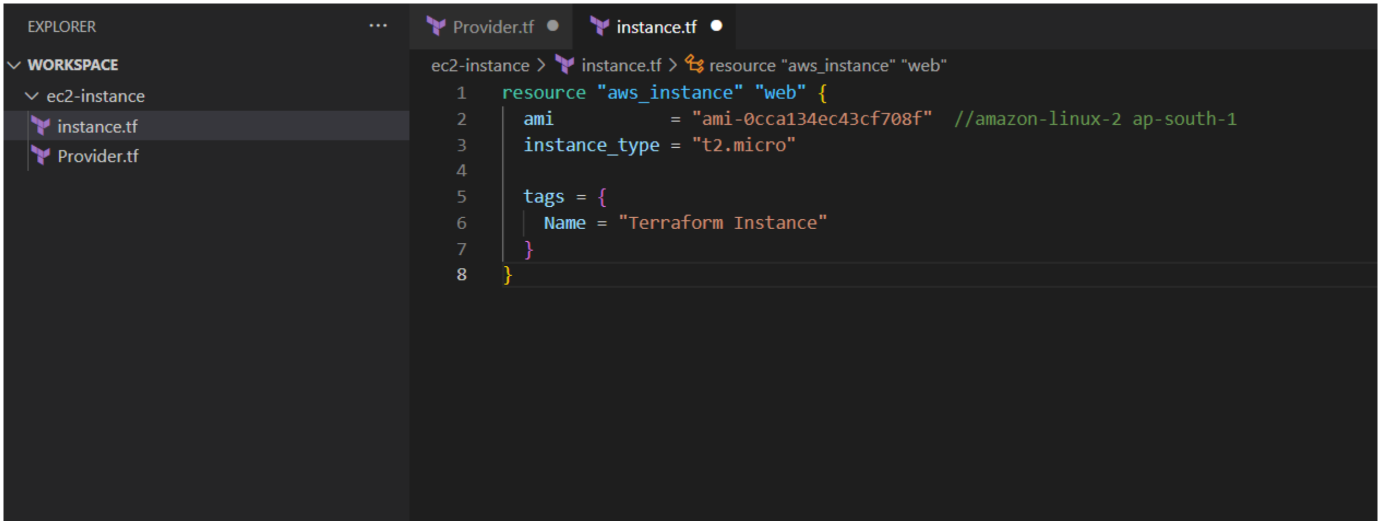
```
resource "aws_vpc" "my_vpc" {
  cidr_block = "172.16.0.0/16"
```

Create a instance.tf file

```
resource "aws_instance" "web" {
  ami           = "ami-0cca134ec43cf708f"  //amazon-linux-2 ap-south-1
  instance_type = "t2.micro"

  tags = {
    Name = "Terraform Instance"
  }
}
```

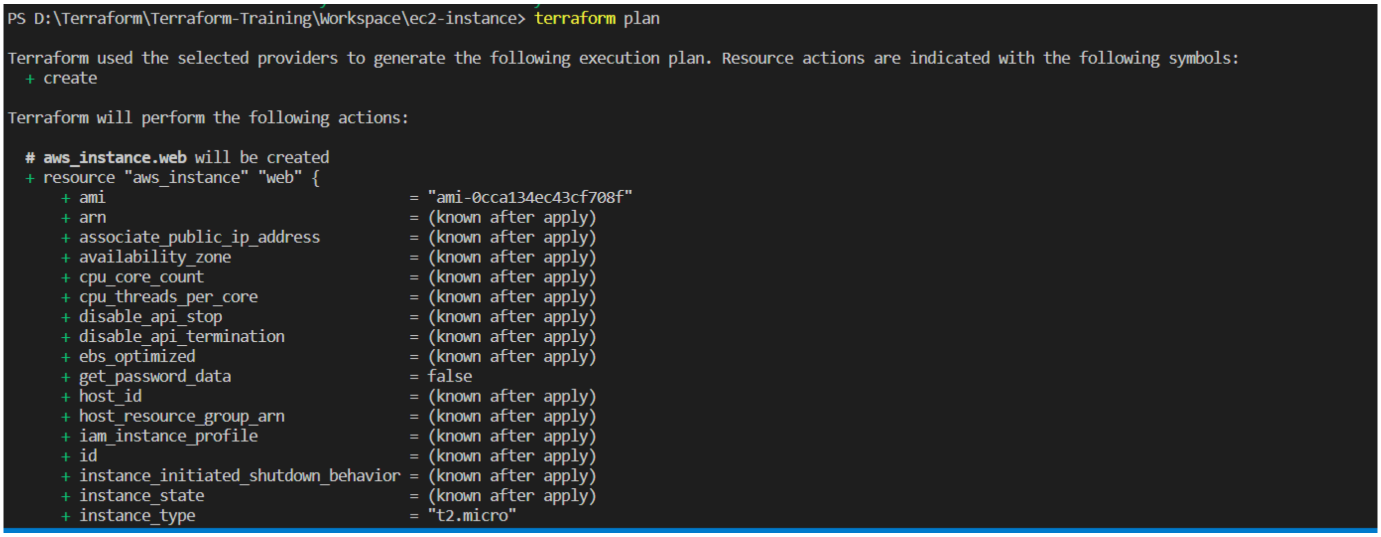


```
provider "aws" {
  region = "ap-south-1"
  profile = "terraform"
}
```



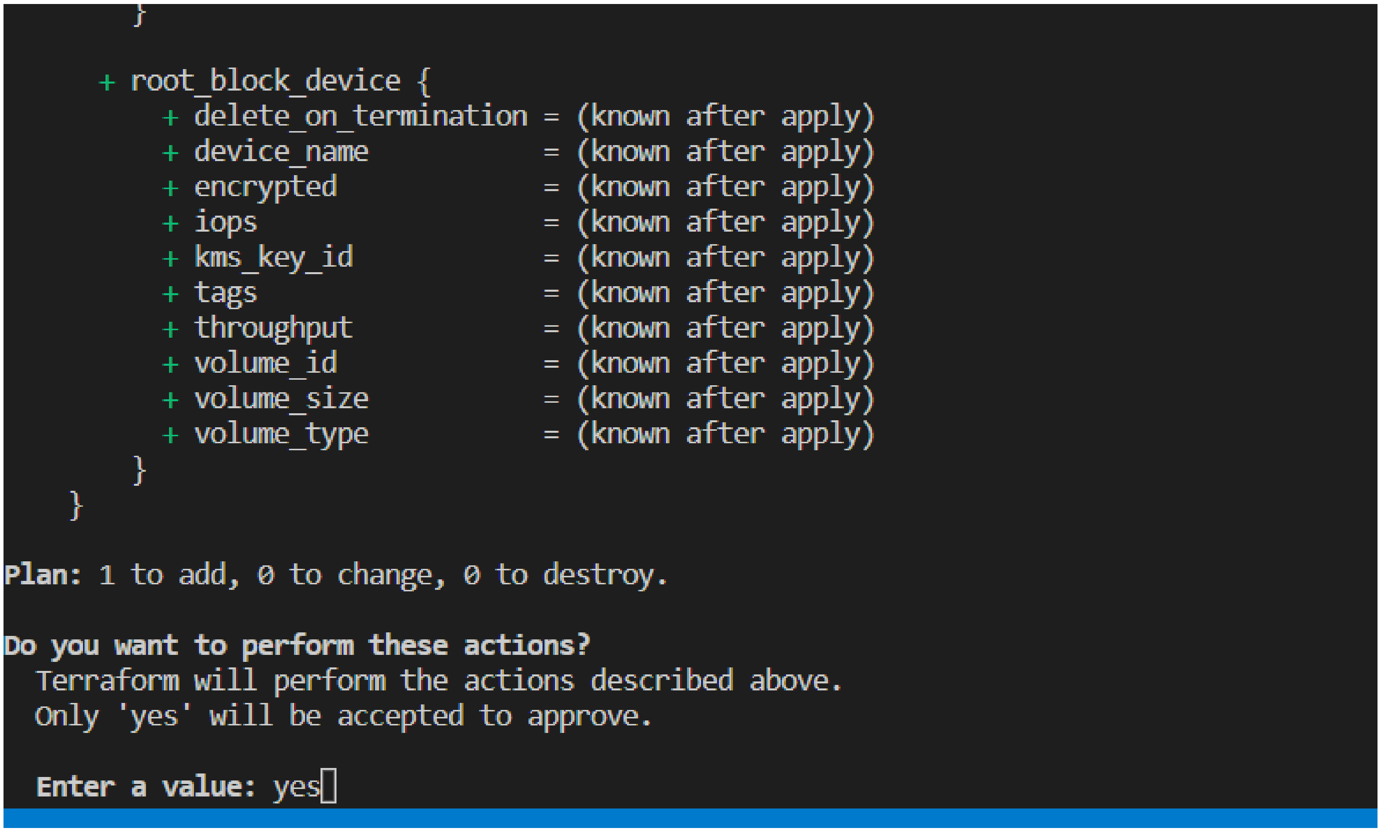
```
resource "aws_instance" "web" {
  ami           = "ami-0cca134ec43cf708f"  //amazon-linux-2 ap-south-1
  instance_type = "t2.micro"

  tags = {
    Name = "Terraform Instance"
  }
}
```

```
terraform init
terraform plan
```

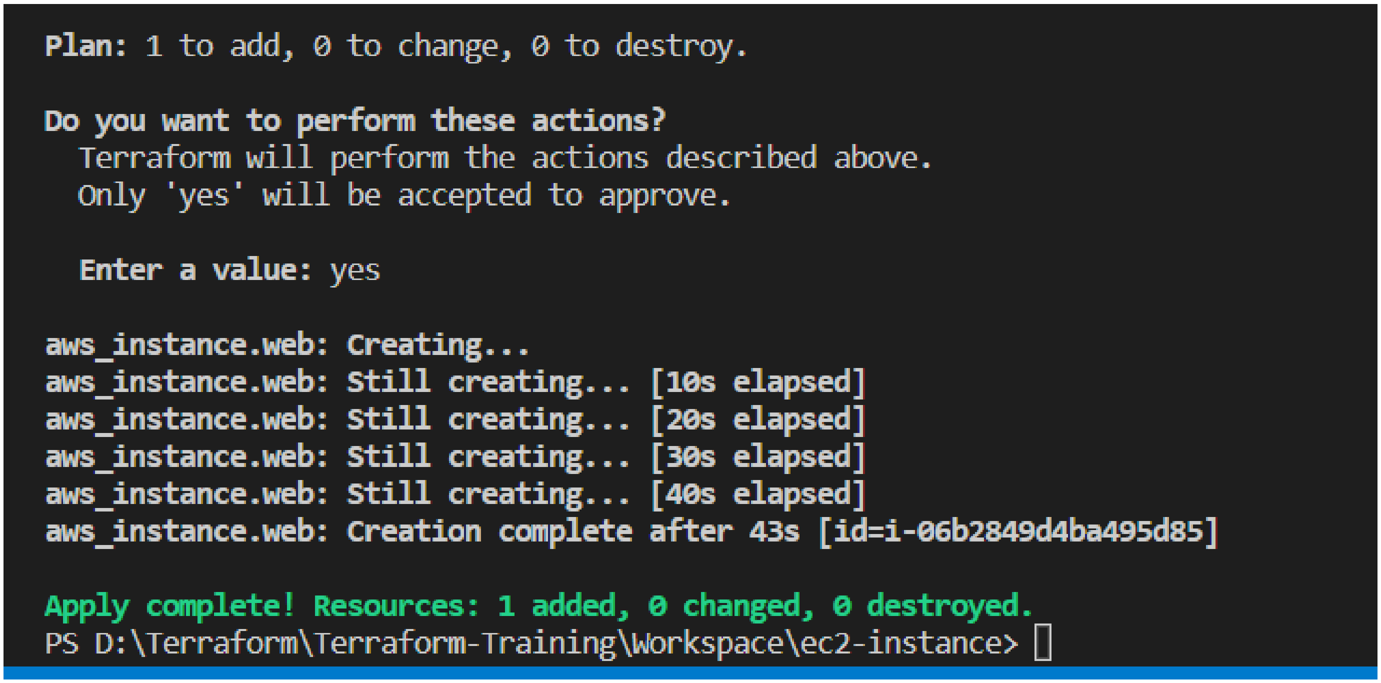


```
PS D:\Terraform\Terraform-Training\Workspace\ec2-instance> terraform plan

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.web will be created
  + resource "aws_instance" "web" {
      + ami                                  = "ami-0cca134ec43cf708f"
      + 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_state                       = (known after apply)
      + instance_type                        = "t2.micro"
```

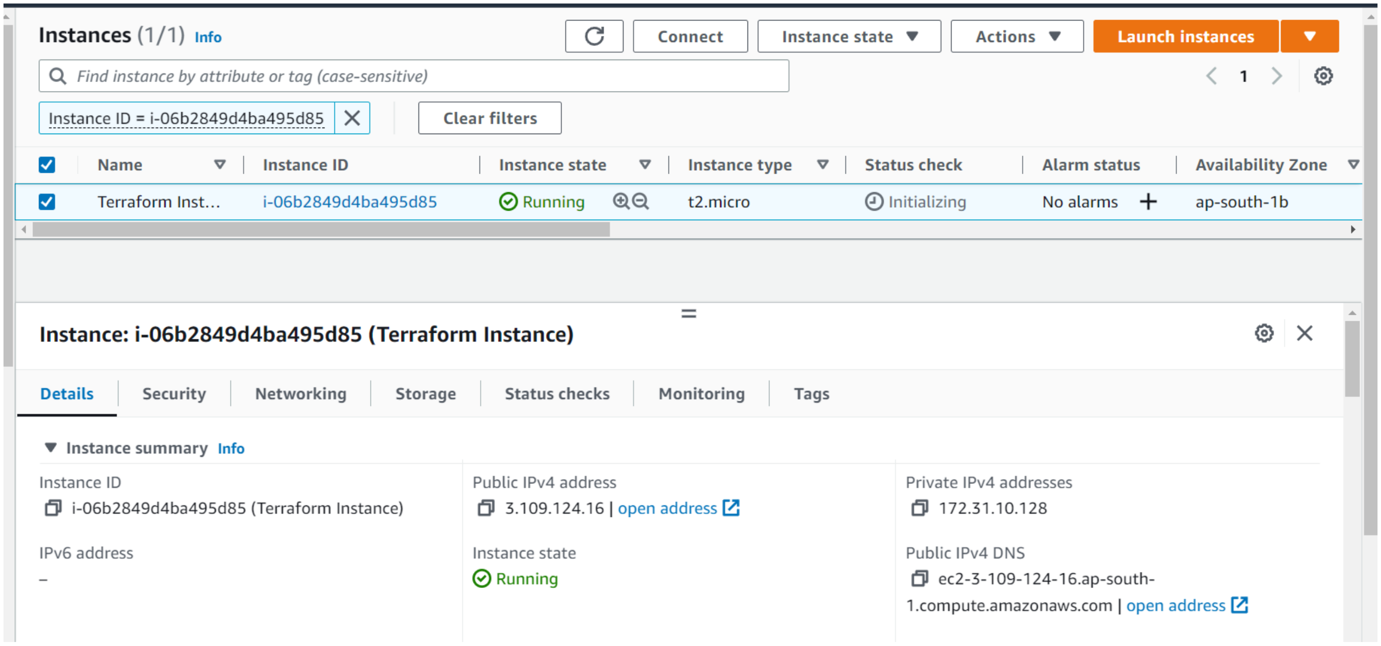
```
terraform apply
```



```
        }

      + 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.

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
```



```
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.web: Creating...
aws_instance.web: Still creating... [10s elapsed]
aws_instance.web: Still creating... [20s elapsed]
aws_instance.web: Still creating... [30s elapsed]
aws_instance.web: Still creating... [40s elapsed]
aws_instance.web: Creation complete after 43s [id=i-06b2849d4ba495d85]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS D:\Terraform\Terraform-Training\Workspace\ec2-instance>
```



Instances (1/1) Info
Connect
Instance state ▼
Actions ▼
Launch instances
Find instance by attribute or tag (case-sensitive)
Instance ID = i-06b2849d4ba495d85
Clear filters
1

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
☑	Terraform Inst...	i-06b2849d4ba495d85	Running	t2.micro	Initializing	No alarms	ap-south-1b

Instance: i-06b2849d4ba495d85 (Terraform Instance)

Instance ID: i-06b2849d4ba495d85 (Terraform Instance)

Public IPv4 address: 3.109.124.16 | open address

Private IPv4 addresses: 172.31.10.128

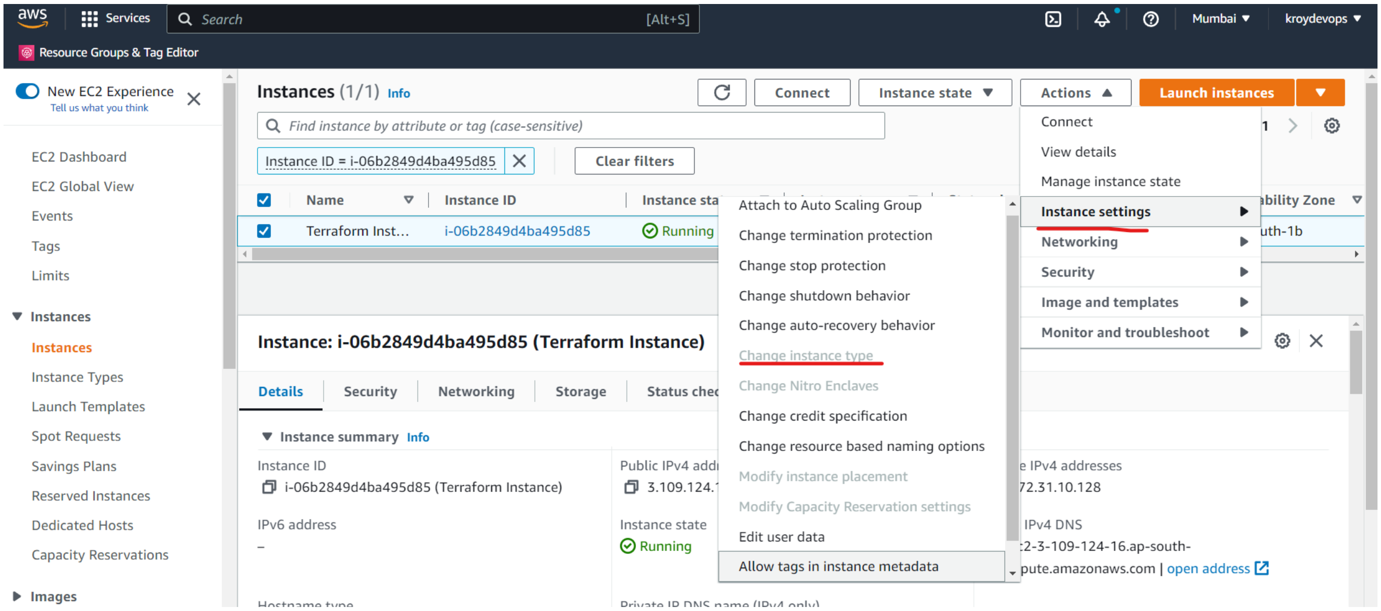
IPv6 address: –

Instance state: Running

Public IPv4 DNS: ec2-3-109-124-16.ap-south-1.compute.amazonaws.com | open address

Now let's change the instance type from t2.micro to t2.small .

You can see that without stopping instance we can not change instance type.



Instance: i-06b2849d4ba495d85 (Terraform Instance)

Instance summary Info

Instance ID
i-06b2849d4ba495d85 (Terraform Instance)

IPv6 address
–

Public IPv4 addr
3.109.124.1

Instance state
Running

e IPv4 addresses
72.31.10.128

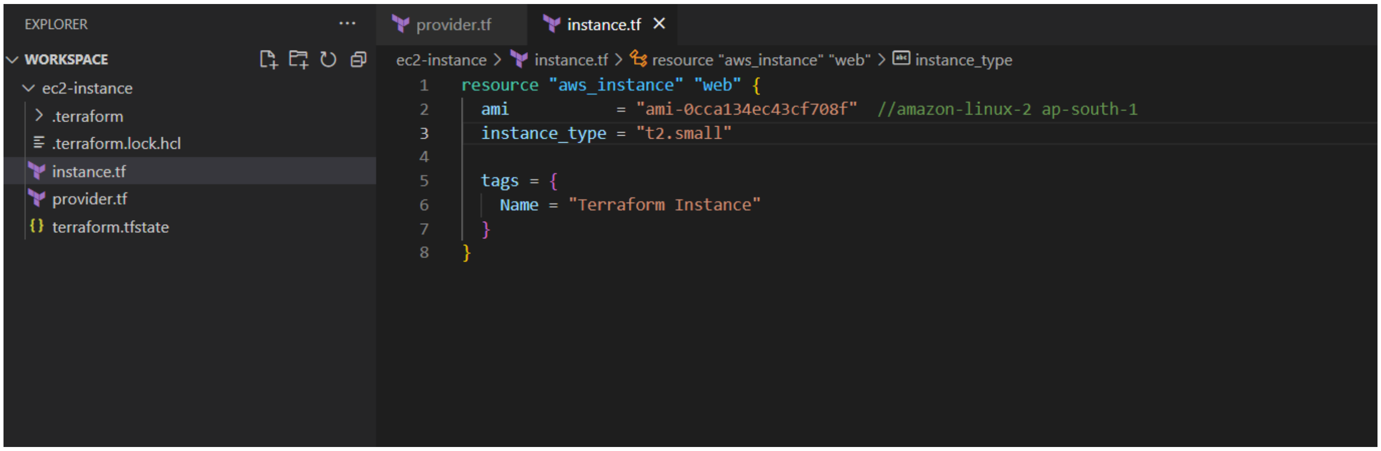
IPv4 DNS
:2-3-109-124-16.ap-south-
pute.amazonaws.com | open address

Hostname type

Private IP DNS name (IPv4 only)

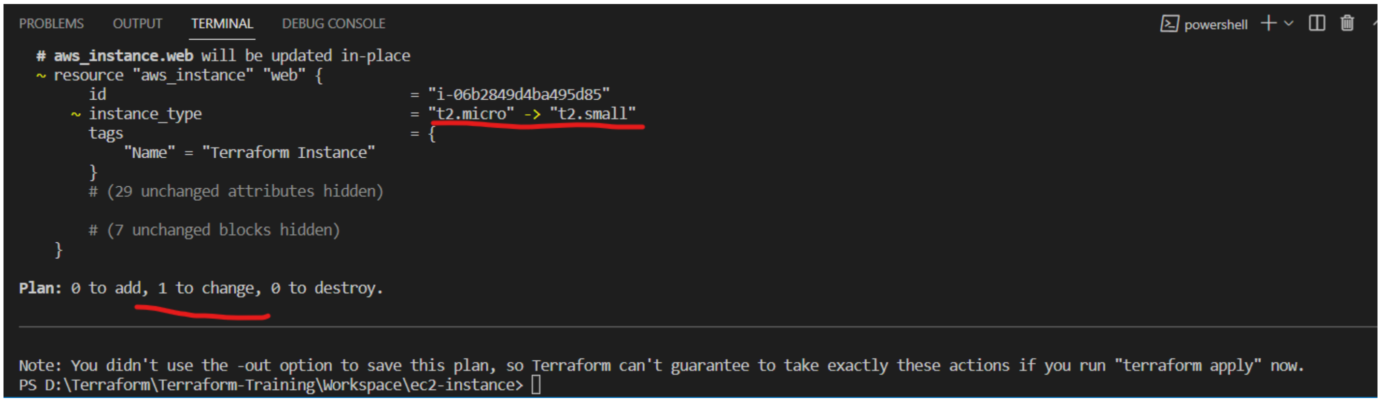
Update the code

```
resource "aws_instance" "web" {
  ami           = "ami-0cca134ec43cf708f"  //amazon-linux-2 ap-south-1
  instance_type = "t2.small"

  tags = {
    Name = "Terraform Instance"
  }
}
```



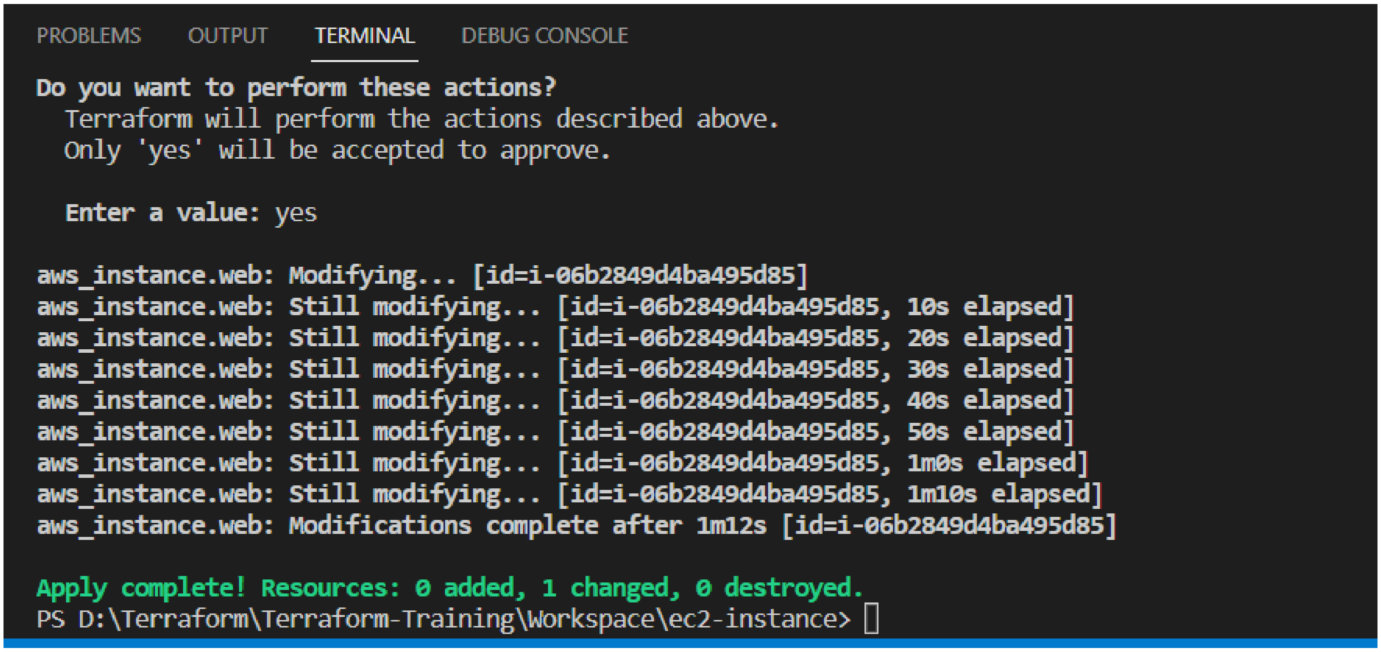
```
resource "aws_instance" "web" {
  ami           = "ami-0cca134ec43cf708f"  //amazon-linux-2 ap-south-1
  instance_type = "t2.small"

  tags = {
    Name = "Terraform Instance"
  }
}
```

```
terraform plan
```

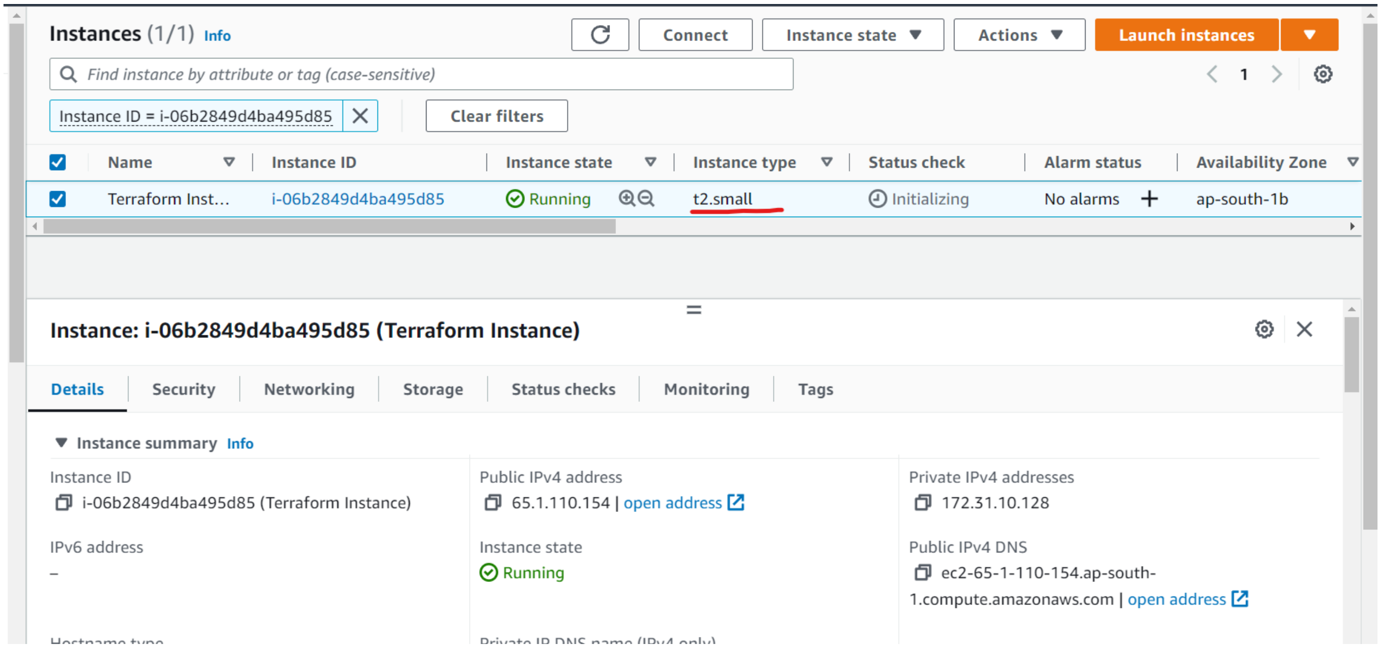


```
  # aws_instance.web will be updated in-place
  ~ resource "aws_instance" "web" {
        id                                   = "i-06b2849d4ba495d85"
      ~ instance_type                        = "t2.micro" -> "t2.small"
        tags                                 = {
            "Name" = "Terraform Instance"
        }
        # (29 unchanged attributes hidden)

        # (7 unchanged blocks hidden)
    }

Plan: 0 to add, 1 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.
PS D:\Terraform\Terraform-Training\Workspace\ec2-instance>
```

```
terraform apply
```



```
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.web: Modifying... [id=i-06b2849d4ba495d85]
aws_instance.web: Still modifying... [id=i-06b2849d4ba495d85, 10s elapsed]
aws_instance.web: Still modifying... [id=i-06b2849d4ba495d85, 20s elapsed]
aws_instance.web: Still modifying... [id=i-06b2849d4ba495d85, 30s elapsed]
aws_instance.web: Still modifying... [id=i-06b2849d4ba495d85, 40s elapsed]
aws_instance.web: Still modifying... [id=i-06b2849d4ba495d85, 50s elapsed]
aws_instance.web: Still modifying... [id=i-06b2849d4ba495d85, 1m0s elapsed]
aws_instance.web: Still modifying... [id=i-06b2849d4ba495d85, 1m10s elapsed]
aws_instance.web: Modifications complete after 1m12s [id=i-06b2849d4ba495d85]

Apply complete! Resources: 0 added, 1 changed, 0 destroyed.
PS D:\Terraform\Terraform-Training\Workspace\ec2-instance>
```



☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
☑	Terraform Inst…	i-06b2849d4ba495d85	Running	t2.small	Initializing	No alarms +	ap-south-1b

Instance: i-06b2849d4ba495d85 (Terraform Instance)

▼ Instance summary Info

Instance ID
i-06b2849d4ba495d85 (Terraform Instance)

Public IPv4 address
65.1.110.154 | open address

Private IPv4 addresses
172.31.10.128

IPv6 address
–

Instance state
Running

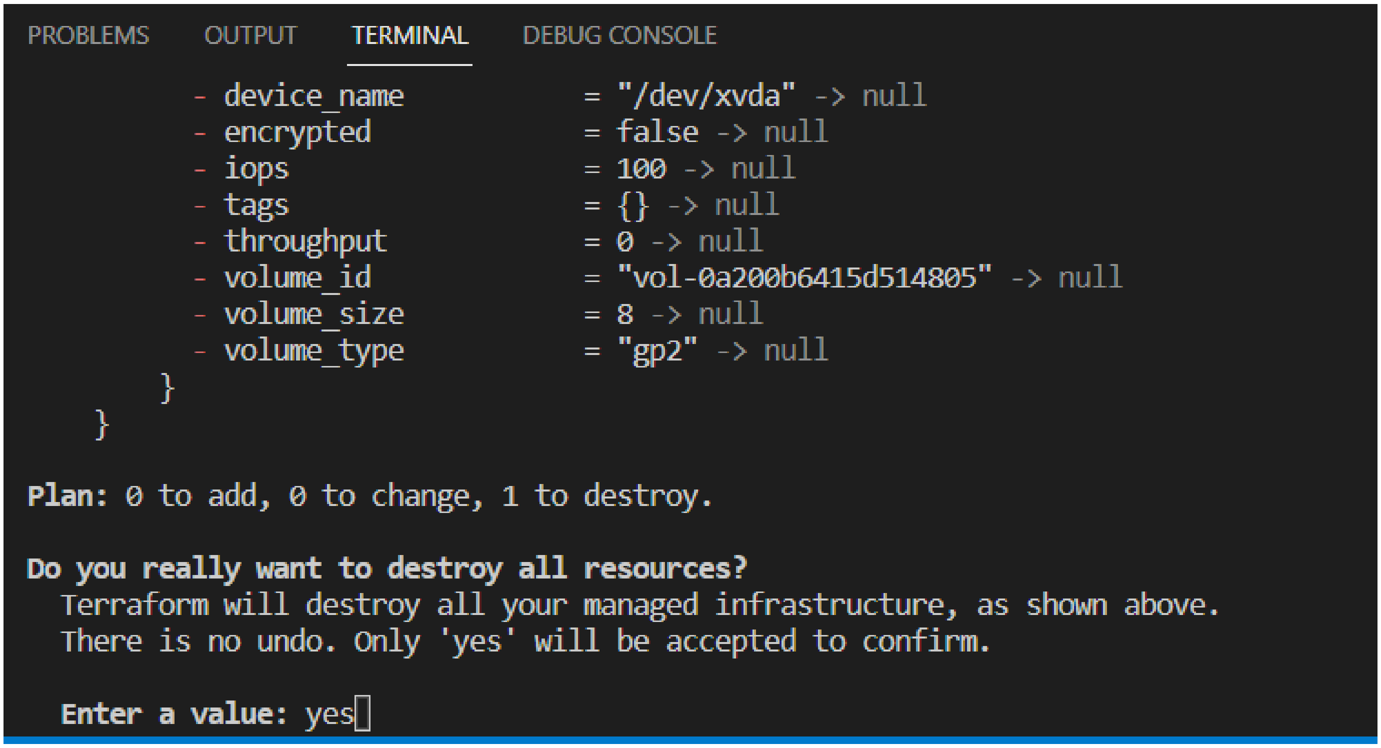
Public IPv4 DNS
ec2-65-1-110-154.ap-south-1.compute.amazonaws.com | open address

Hostname type

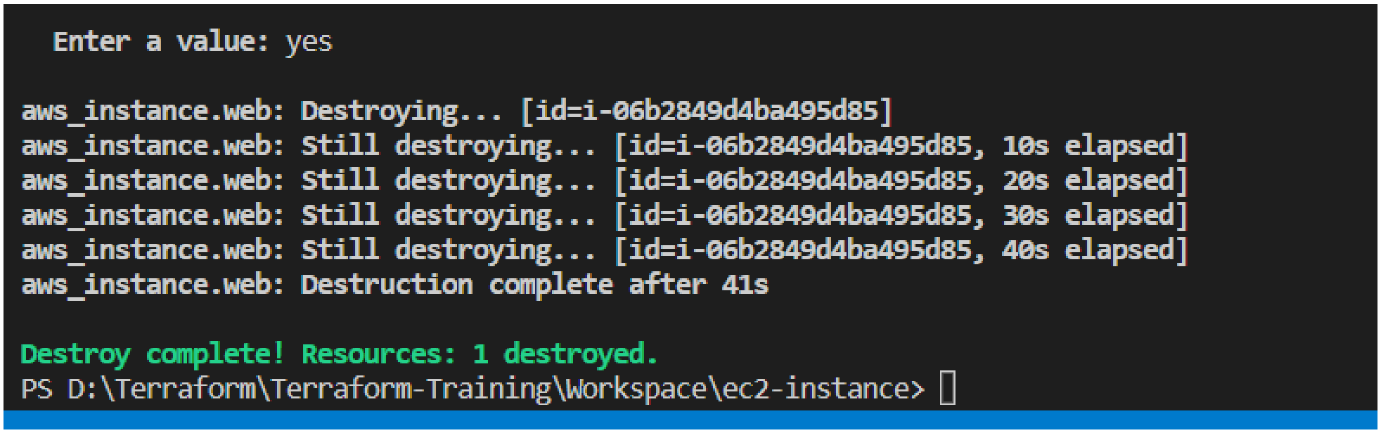
Private IP DNS name (IPv4 only)

To destroy the resources:

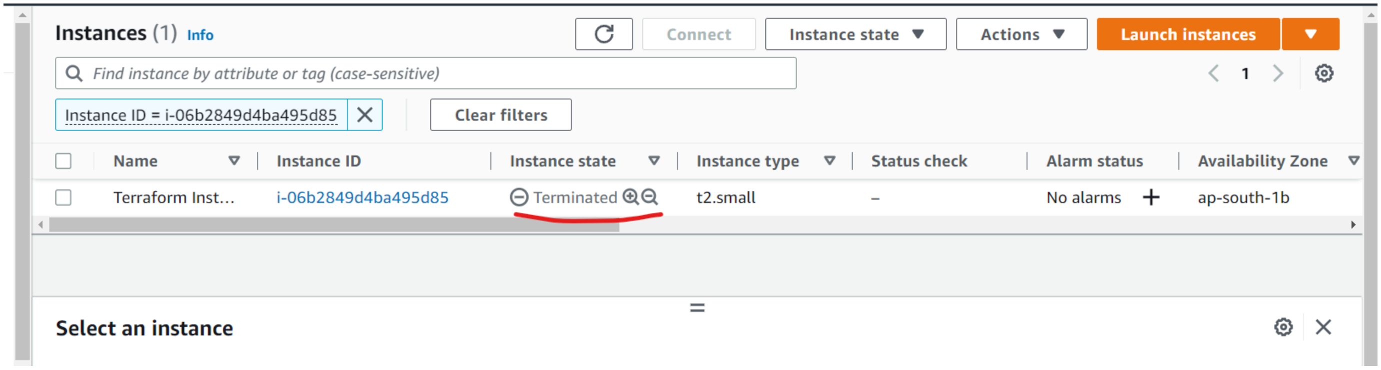
```
terraform destroy
```



```
          - device_name           = "/dev/xvda" -> null
          - encrypted             = false -> null
          - iops                  = 100 -> null
          - tags                  = {} -> null
          - throughput            = 0 -> null
          - volume_id             = "vol-0a200b6415d514805" -> 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
```



```
  Enter a value: yes

aws_instance.web: Destroying... [id=i-06b2849d4ba495d85]
aws_instance.web: Still destroying... [id=i-06b2849d4ba495d85, 10s elapsed]
aws_instance.web: Still destroying... [id=i-06b2849d4ba495d85, 20s elapsed]
aws_instance.web: Still destroying... [id=i-06b2849d4ba495d85, 30s elapsed]
aws_instance.web: Still destroying... [id=i-06b2849d4ba495d85, 40s elapsed]
aws_instance.web: Destruction complete after 41s

Destroy complete! Resources: 1 destroyed.
PS D:\Terraform\Terraform-Training\Workspace\ec2-instance> 
```



Instances (1) Info

Connect
Instance state ▼
Actions ▼
Launch instances
Find instance by attribute or tag (case-sensitive)
Instance ID = i-06b2849d4ba495d85
Clear filters
1

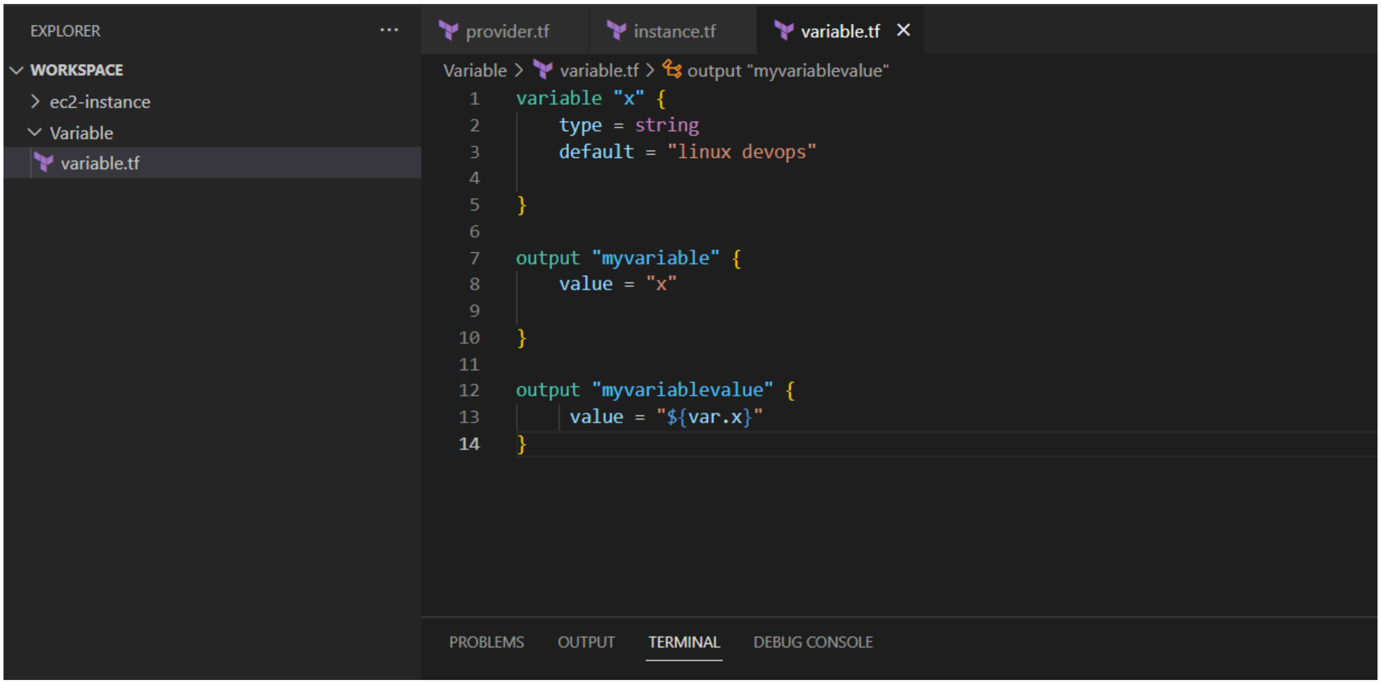
□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
□	Terraform Inst…	i-06b2849d4ba495d85	⊖ Terminated	t2.small	–	No alarms	ap-south-1b

Select an instance

Variables in Terraform

user defined variable :- reference https://developer.hashicorp.com/terraform/language/values/variables

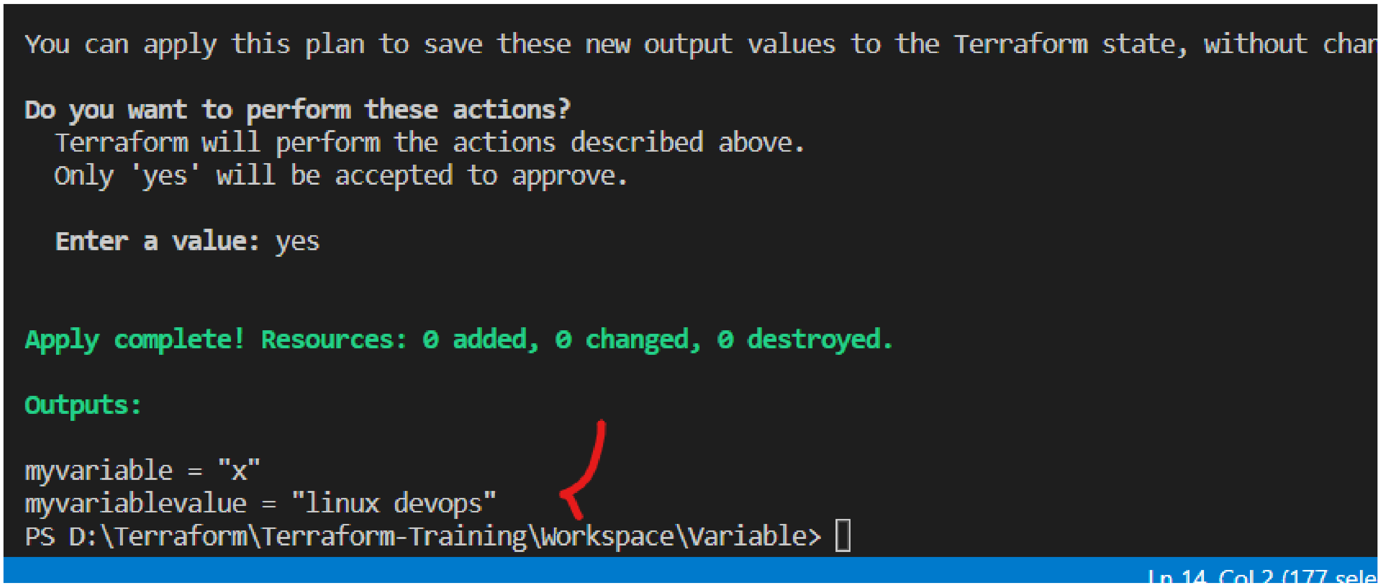
Create a new folder with name "variable" and inside it create a file with name "variable.tf"



```
variable "x" {
    type = string
    default = "linux devops"

}

output "myvariable" {
    value = "x"

}

output "myvariablevalue" {
     value = "${var.x}"
}
```

```
variable "x" {
    type = string
    default = "linux devops"

}

output "myvariable" {
    value = "x"   //will print x as it is

}

output "myvariablevalue" {
     value = "${var.x}"   //will print value of x
}
```

```
terraform apply
```



```
You can apply this plan to save these new output values to the Terraform state, without char

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


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

Outputs:

myvariable = "x"
myvariablevalue = "linux devops"
PS D:\Terraform\Terraform-Training\Workspace\Variable> 
```

To Launch Apache webserver on AWS instance using Terraform

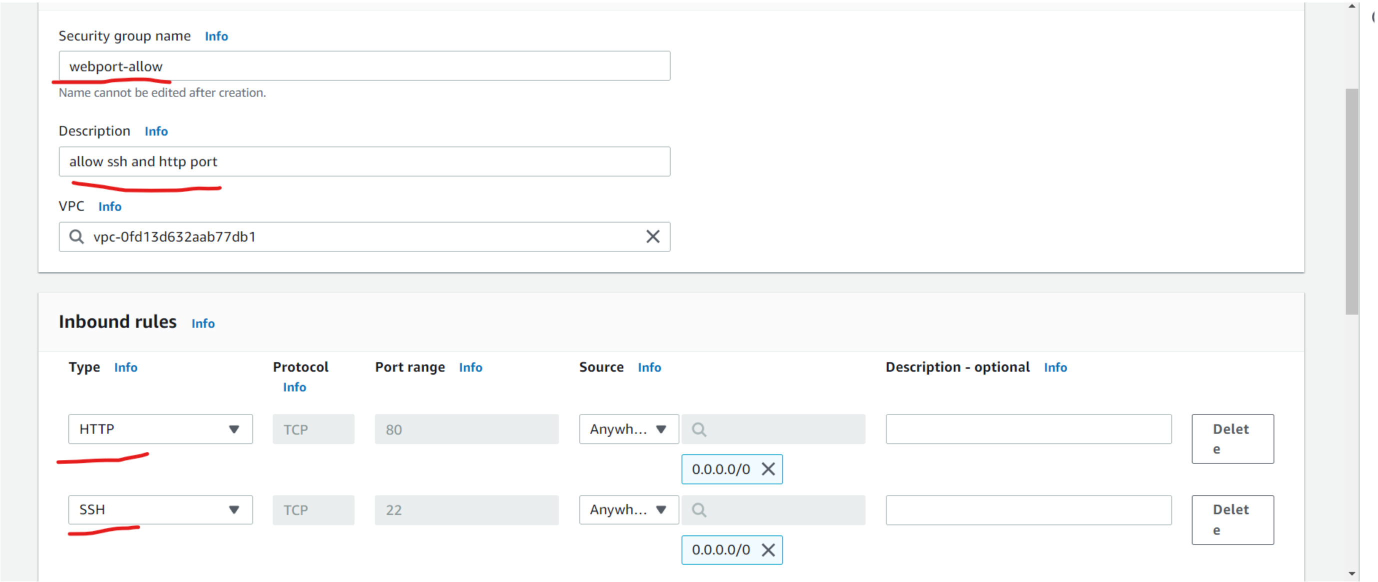
Steps:-

1. Launch an ec2-instance with http port enabled.
2. Launch an EBS volume of 1 gb in same availability zone as of ec2-instance
3. Attach the newly launched EBS volume to instance and create a partition and do format as taught during aws sessions.
4. Install php and httpd software and keep the website content copied from github to /var/www/html location.
5. Now access the webserver on public IP of instance.

Create a new workspace with name “Apache_webserver” using vs code editor.

Prerequisite:-

1. Create a security group manually with name “webport-allow” with port 80 and 22 .
2. Create a keyPair with name “terraform_key” .



Security group name Info

webport-allow

Name cannot be edited after creation.

Description Info

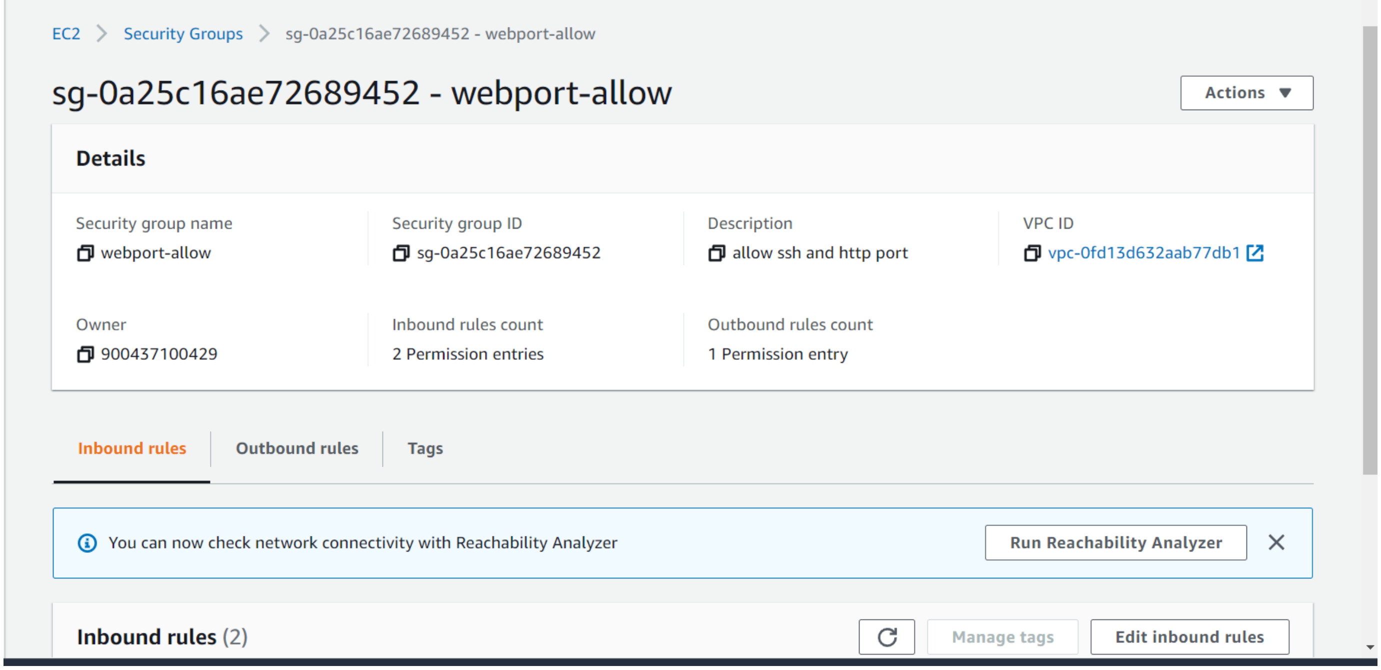
allow ssh and http port

VPC Info

vpc-0fd13d632aab77db1

Inbound rules Info

Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
HTTP	TCP	80	Anywh...	0.0.0.0/0		Delete
SSH	TCP	22	Anywh...	0.0.0.0/0		Delete



EC2 > Security Groups > sg-0a25c16ae72689452 - webport-allow

sg-0a25c16ae72689452 - webport-allow

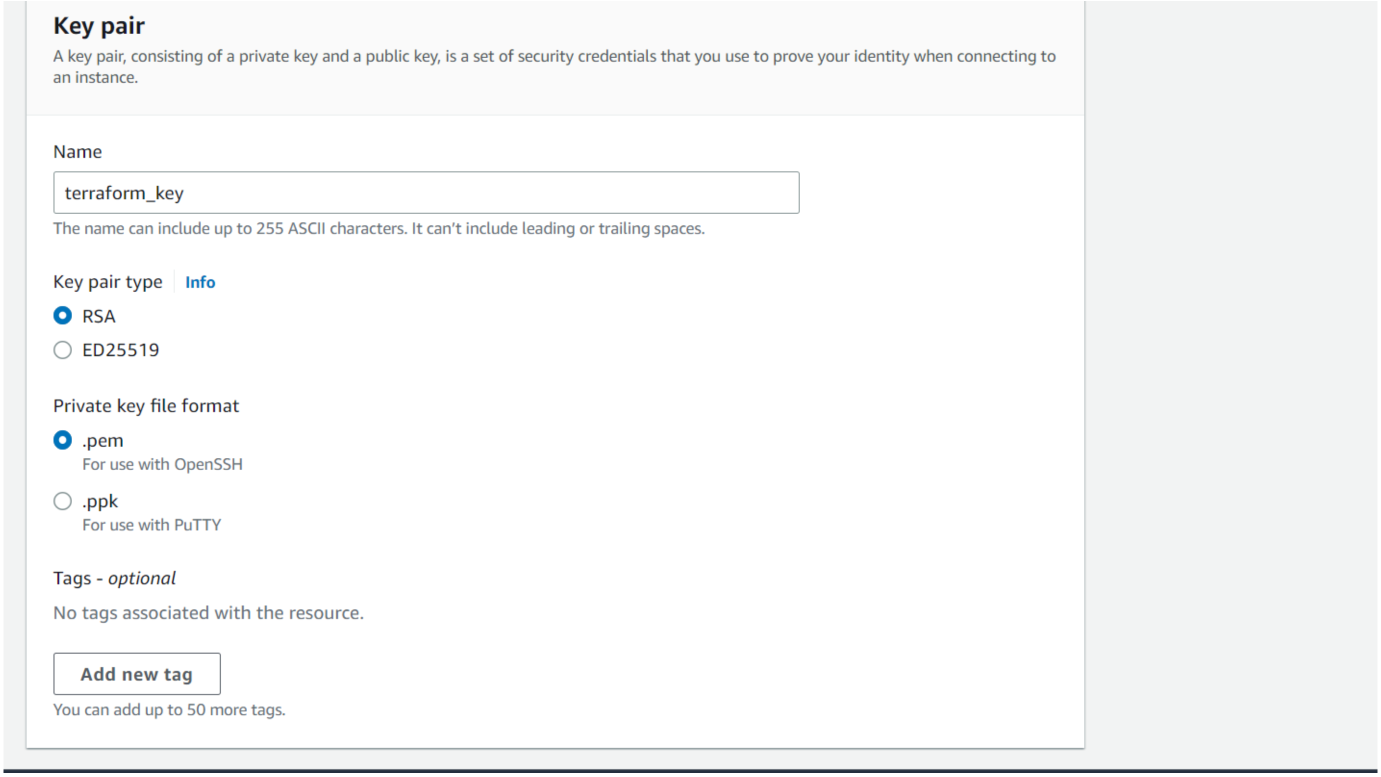
Details

Security group name	Security group ID	Description	VPC ID
webport-allow	sg-0a25c16ae72689452	allow ssh and http port	vpc-0fd13d632aab77db1
Owner	**Inbound rules count**	**Outbound rules count**	
900437100429	2 Permission entries	1 Permission entry	

You can now check network connectivity with Reachability Analyzer

Inbound rules (2)

keypair



Key pair

A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name

terraform_key

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type | Info

● RSA

○ ED25519

Private key file format

● .pem
For use with OpenSSH

○ .ppk
For use with PuTTY

Tags - *optional*

No tags associated with the resource.

You can add up to 50 more tags.

provider.tf

```
provider "aws" {
  region = "ap-south-1"
  profile = "terraform"
}
```

ec2.tf

```
resource "aws_instance" "webos1" {
  ami           = "ami-010aff33ed5991201"
  instance_type = "t2.micro"
  security_groups =  [ "webport-allow" ]
  key_name = "terraform_key"

  tags = {
    Name = "Web Server by TF"
  }

}
```

blockstorage.tf

```
resource "aws_ebs_volume" "example" {
  availability_zone = aws_instance.webos1.availability_zone
  size              = 1

  tags = {
    Name = "Web Server HD by TF"
  }
}
```

attachblock.tf

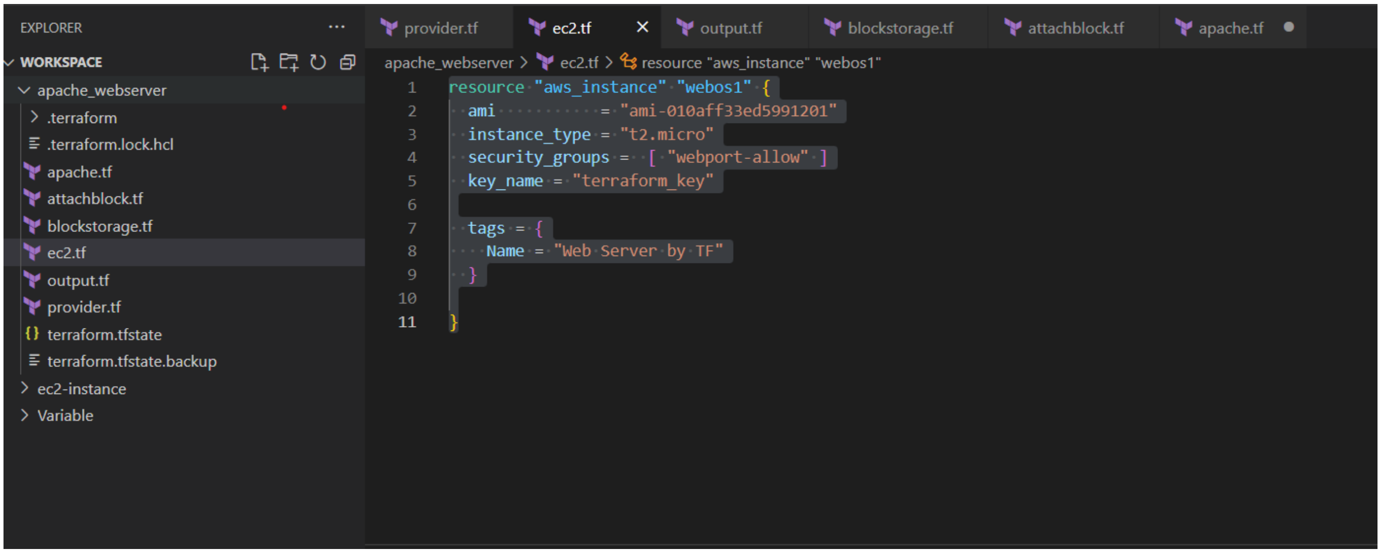
```
resource "aws_volume_attachment" "ebs_att" {
  device_name = "/dev/xvdc"
  volume_id   = aws_ebs_volume.example.id
  instance_id = aws_instance.webos1.id
  force_detach = true
}
```

apache.tf

```
resource "null_resource"  "nullremote1" {

depends_on = [
  aws_volume_attachment.ebs_att
]

connection {
    type     = "ssh"
    user     = "ec2-user"
    private_key = file("D:/Terraform/Terraform-Training/terraform_key.pem") //pem file
    host     = aws_instance.webos1.public_ip
  }

provisioner "remote-exec" {
    inline = [
      "sudo yum  install httpd  -y",
      "sudo  yum  install php  -y",
      "sudo systemctl start httpd",
      "sudo systemctl start httpd",
 "sudo mkfs.ext4 /dev/xvdc",
      "sudo  mount /dev/xvdc  /var/www/html",
  "sudo yum install git -y",
      "sudo git clone https://github.com/devopskroy/apacheWebpage.git /var/www/html/web"

    ]
  }

}
```

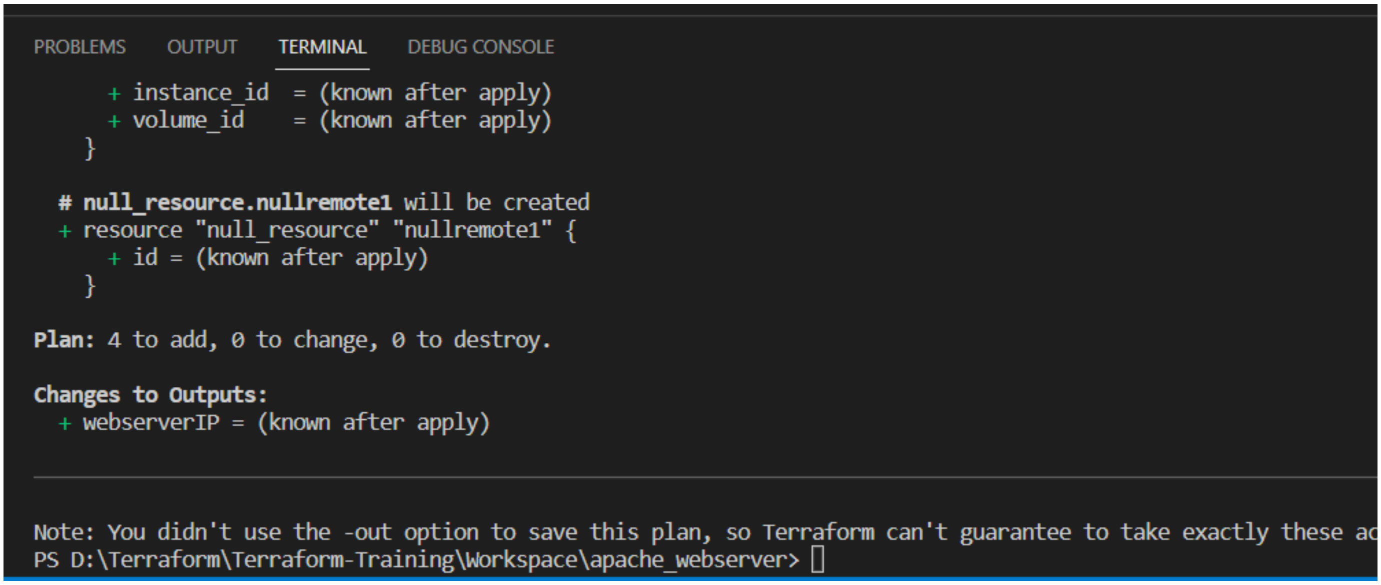
output.tf

```
output "webserverIP" {
 value = aws_instance.webos1.public_ip
}
```



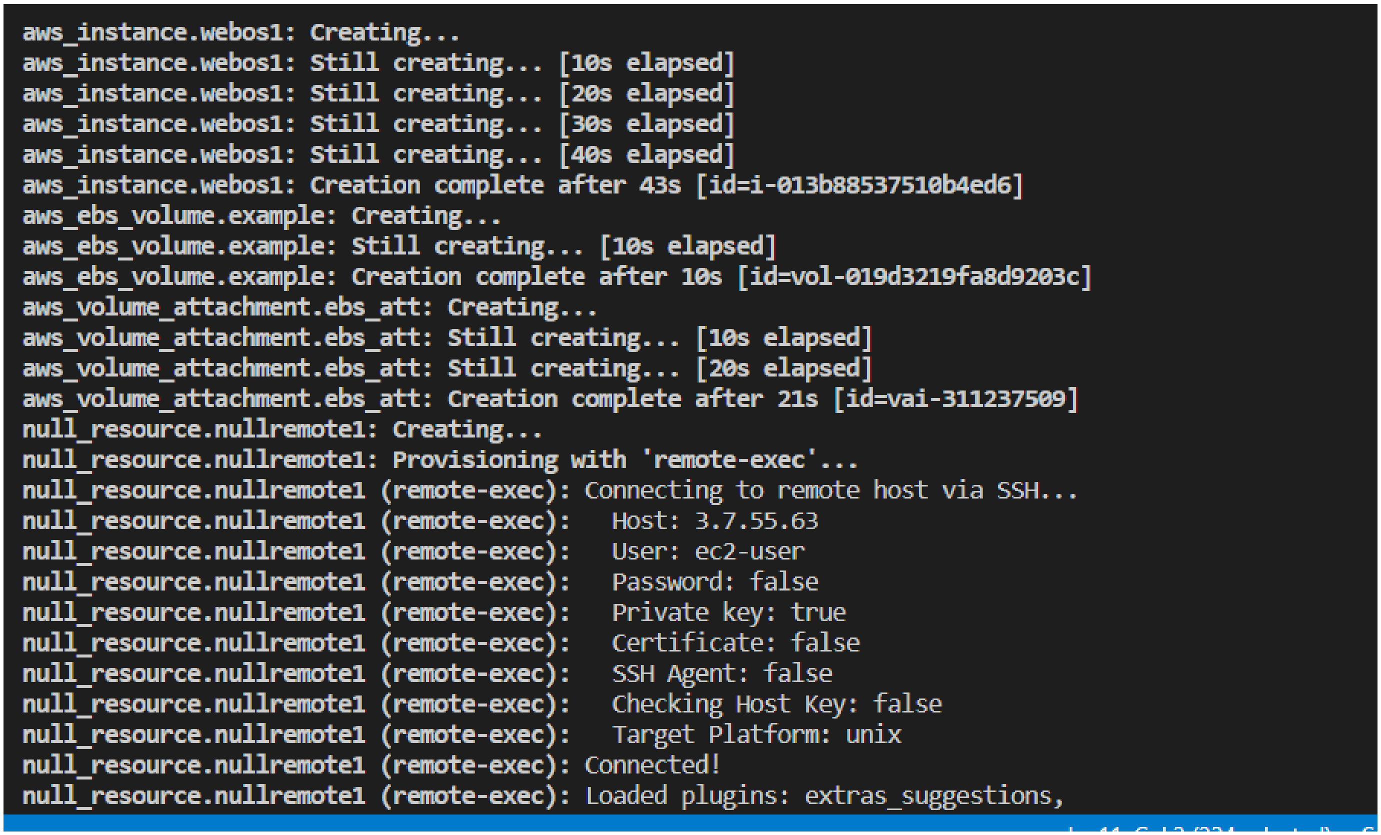
```
resource "aws_instance" "webos1" {
  ami           = "ami-010aff33ed5991201"
  instance_type = "t2.micro"
  security_groups =  [ "webport-allow" ]
  key_name = "terraform_key"

  tags = {
    Name = "Web Server by TF"
  }

}
```

```
terraform plan
```

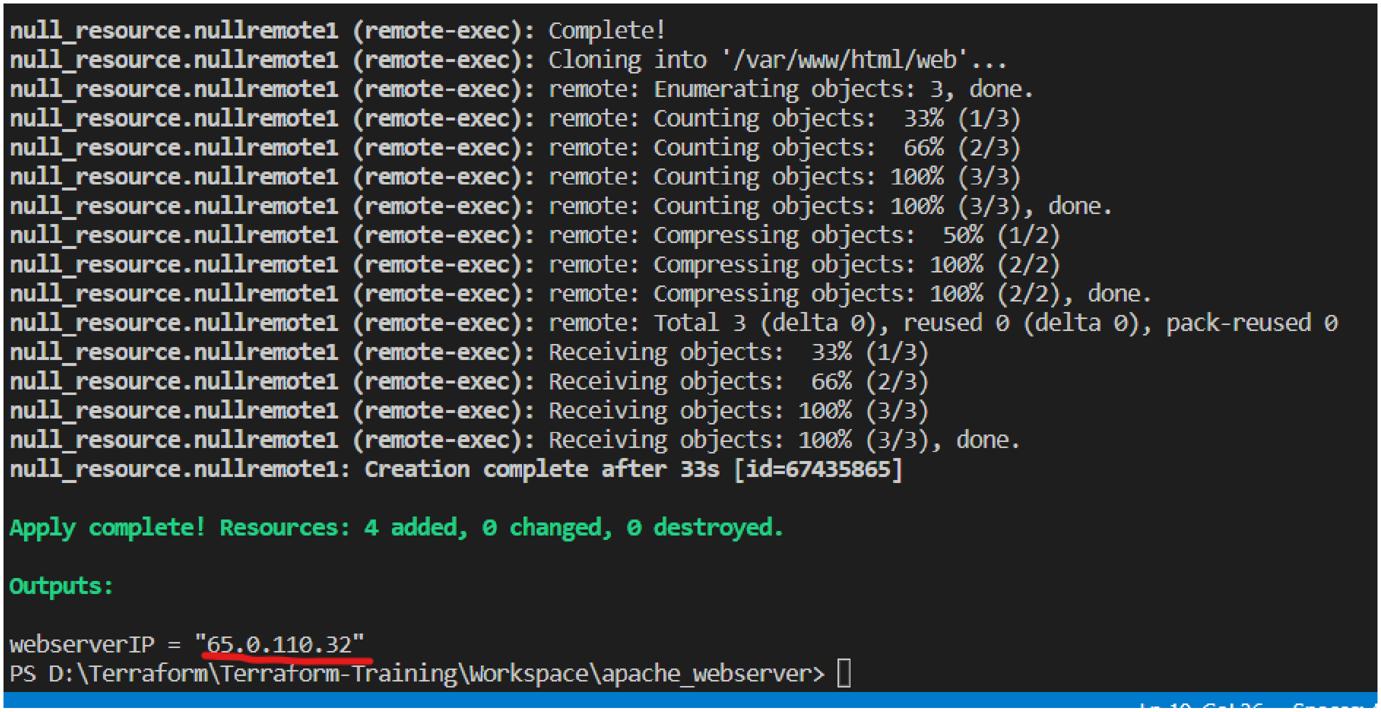


```
      + instance_id  = (known after apply)
      + volume_id    = (known after apply)
    }

  # null_resource.nullremote1 will be created
  + resource "null_resource" "nullremote1" {
      + id = (known after apply)
    }

Plan: 4 to add, 0 to change, 0 to destroy.

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

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

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these ac
PS D:\Terraform\Terraform-Training\Workspace\apache_webserver> 
```

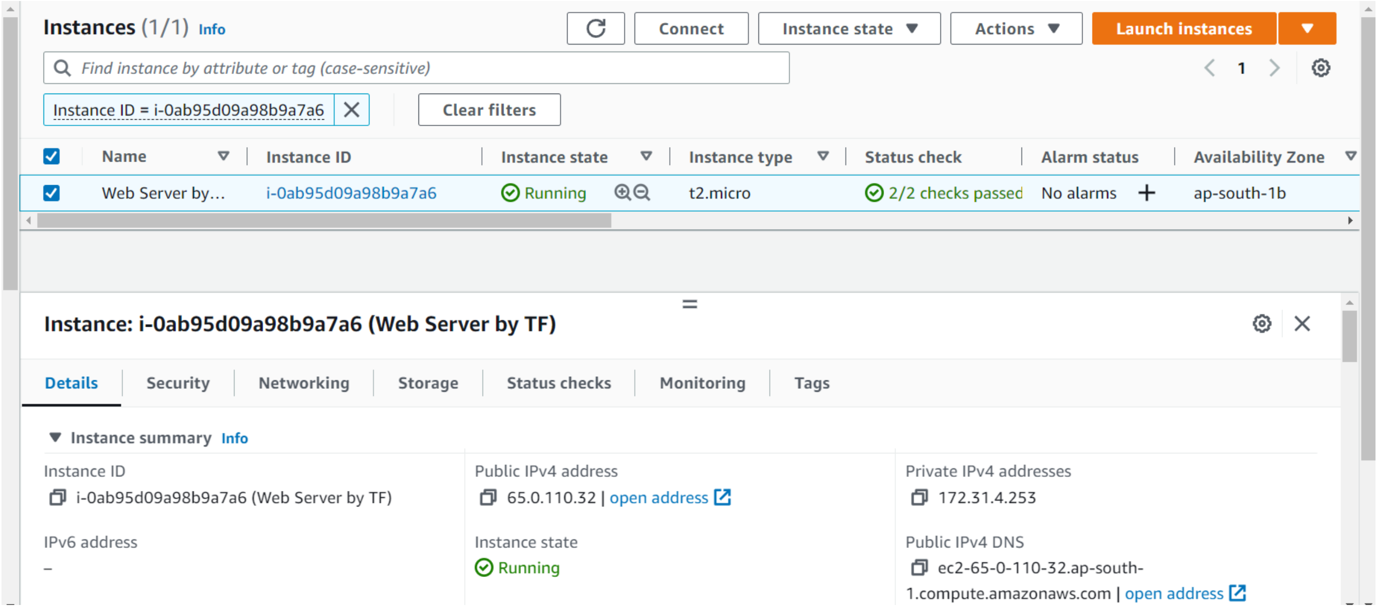
```
terraform apply
```



```
aws_instance.webos1: Creating...
aws_instance.webos1: Still creating... [10s elapsed]
aws_instance.webos1: Still creating... [20s elapsed]
aws_instance.webos1: Still creating... [30s elapsed]
aws_instance.webos1: Still creating... [40s elapsed]
aws_instance.webos1: Creation complete after 43s [id=i-013b88537510b4ed6]
aws_ebs_volume.example: Creating...
aws_ebs_volume.example: Still creating... [10s elapsed]
aws_ebs_volume.example: Creation complete after 10s [id=vol-019d3219fa8d9203c]
aws_volume_attachment.ebs_att: Creating...
aws_volume_attachment.ebs_att: Still creating... [10s elapsed]
aws_volume_attachment.ebs_att: Still creating... [20s elapsed]
aws_volume_attachment.ebs_att: Creation complete after 21s [id=vai-311237509]
null_resource.nullremote1: Creating...
null_resource.nullremote1: Provisioning with 'remote-exec'...
null_resource.nullremote1 (remote-exec): Connecting to remote host via SSH...
null_resource.nullremote1 (remote-exec):   Host: 3.7.55.63
null_resource.nullremote1 (remote-exec):   User: ec2-user
null_resource.nullremote1 (remote-exec):   Password: false
null_resource.nullremote1 (remote-exec):   Private key: true
null_resource.nullremote1 (remote-exec):   Certificate: false
null_resource.nullremote1 (remote-exec):   SSH Agent: false
null_resource.nullremote1 (remote-exec):   Checking Host Key: false
null_resource.nullremote1 (remote-exec):   Target Platform: unix
null_resource.nullremote1 (remote-exec): Connected!
null_resource.nullremote1 (remote-exec): Loaded plugins: extras_suggestions,
```



```
null_resource.nullremote1 (remote-exec): Complete!
null_resource.nullremote1 (remote-exec): Cloning into '/var/www/html/web'...
null_resource.nullremote1 (remote-exec): remote: Enumerating objects: 3, done.
null_resource.nullremote1 (remote-exec): remote: Counting objects:  33% (1/3)
null_resource.nullremote1 (remote-exec): remote: Counting objects:  66% (2/3)
null_resource.nullremote1 (remote-exec): remote: Counting objects: 100% (3/3)
null_resource.nullremote1 (remote-exec): remote: Counting objects: 100% (3/3), done.
null_resource.nullremote1 (remote-exec): remote: Compressing objects:  50% (1/2)
null_resource.nullremote1 (remote-exec): remote: Compressing objects: 100% (2/2)
null_resource.nullremote1 (remote-exec): remote: Compressing objects: 100% (2/2), done.
null_resource.nullremote1 (remote-exec): remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
null_resource.nullremote1 (remote-exec): Receiving objects:  33% (1/3)
null_resource.nullremote1 (remote-exec): Receiving objects:  66% (2/3)
null_resource.nullremote1 (remote-exec): Receiving objects: 100% (3/3)
null_resource.nullremote1 (remote-exec): Receiving objects: 100% (3/3), done.
null_resource.nullremote1: Creation complete after 33s [id=67435865]

Apply complete! Resources: 4 added, 0 changed, 0 destroyed.

Outputs:

webserverIP = "65.0.110.32"
PS D:\Terraform\Terraform-Training\Workspace\apache_webserver> 
```


Instances (1/1) Info
Connect
Instance state ▼
Actions ▼
Launch instances
Find instance by attribute or tag (case-sensitive)
Instance ID = i-0ab95d09a98b9a7a6
Clear filters
1

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
☑	Web Server by…	i-0ab95d09a98b9a7a6	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms	ap-south-1b

Instance: i-0ab95d09a98b9a7a6 (Web Server by TF)

▼ Instance summary Info

Instance ID
i-0ab95d09a98b9a7a6 (Web Server by TF)

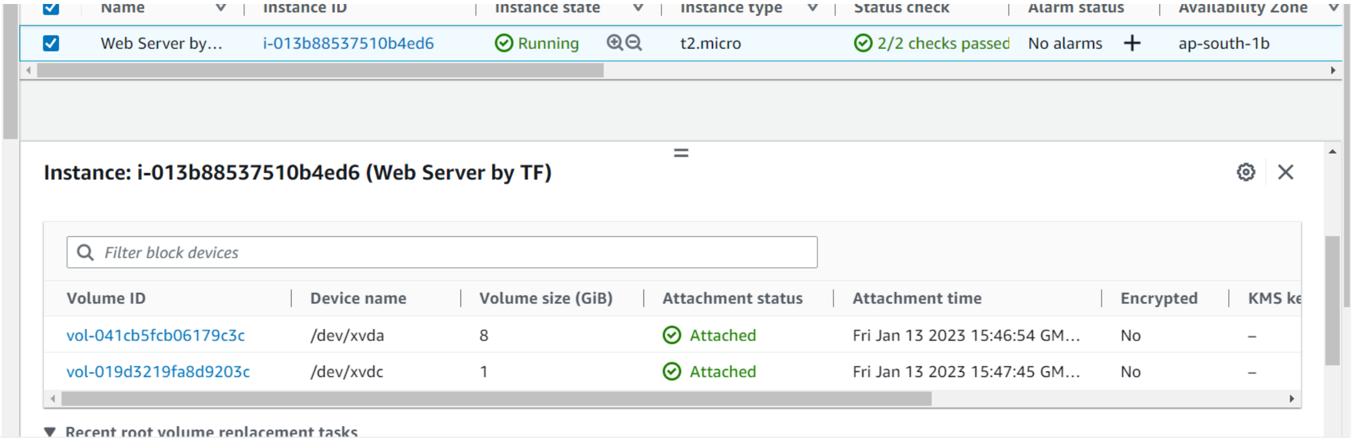
Public IPv4 address
65.0.110.32 | open address

Private IPv4 addresses
172.31.4.253

IPv6 address
–

Instance state
✓ Running

Public IPv4 DNS
ec2-65-0-110-32.ap-south-1.compute.amazonaws.com | open address



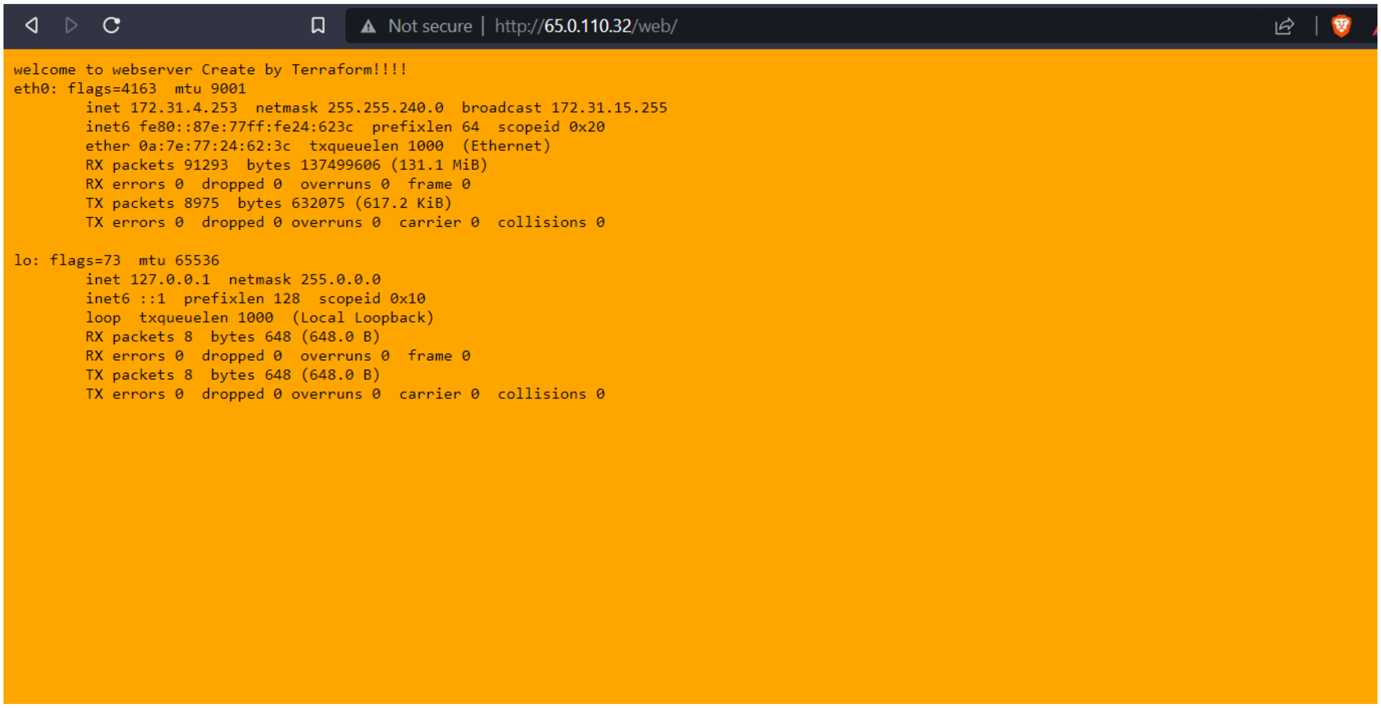
☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
☑	Web Server by…	i-013b88537510b4ed6	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms	ap-south-1b

Instance: i-013b88537510b4ed6 (Web Server by TF)

Volume ID	Device name	Volume size (GiB)	Attachment status	Attachment time	Encrypted	KMS ke
vol-041cb5fcb06179c3c	/dev/xvda	8	✓ Attached	Fri Jan 13 2023 15:46:54 GM…	No	–
vol-019d3219fa8d9203c	/dev/xvdc	1	✓ Attached	Fri Jan 13 2023 15:47:45 GM…	No	–

▼ Recent root volume replacement tasks

Enter the url in browser:- http://<public-ip>:80/web



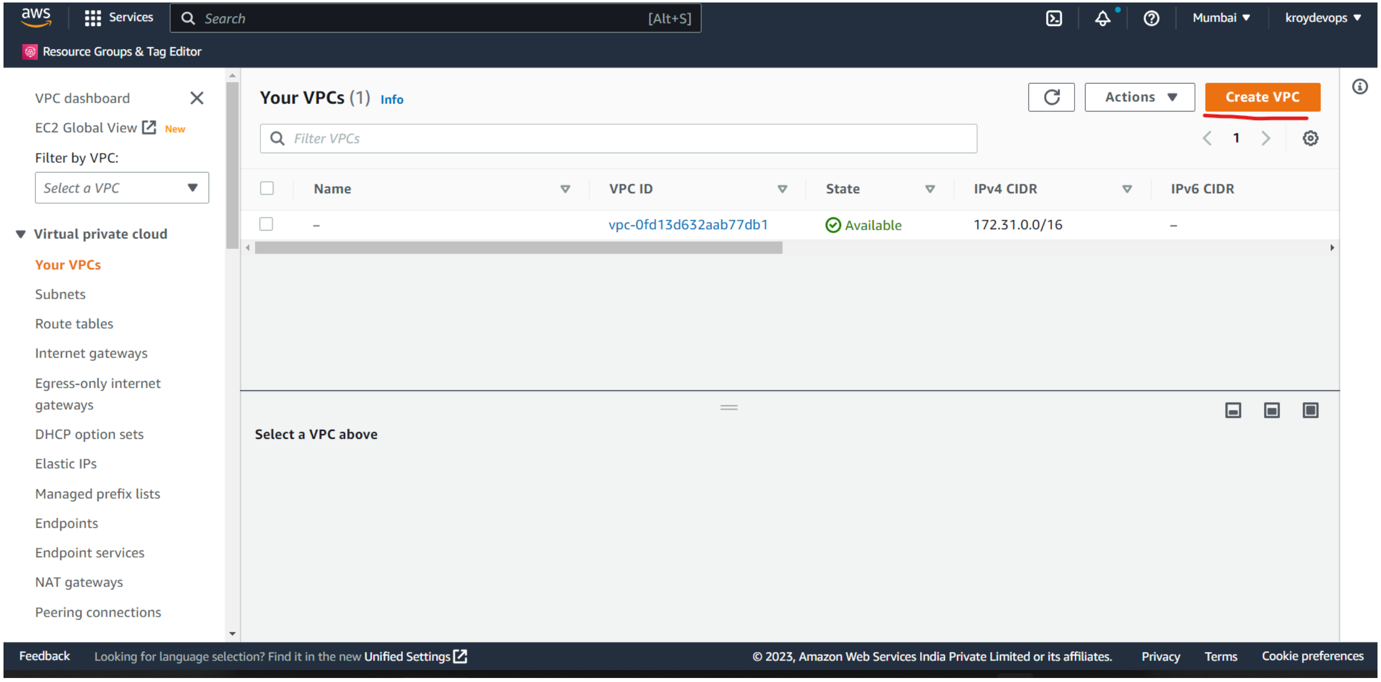
```
welcome to webserver Create by Terraform!!!!
eth0: flags=4163  mtu 9001
        inet 172.31.4.253  netmask 255.255.240.0  broadcast 172.31.15.255
        inet6 fe80::87e:77ff:fe24:623c  prefixlen 64  scopeid 0x20
        ether 0a:7e:77:24:62:3c  txqueuelen 1000  (Ethernet)
        RX packets 91293  bytes 137499606 (131.1 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8975  bytes 632075 (617.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 648 (648.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 648 (648.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

To setup VPC using Terraform

ref:- click me

1. Create a network/vpc first.

name and CIDR requires. subnet calculator



Your VPCs (1) Info

□	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
□	–	vpc-0fd13d632aab77db1	Available	172.31.0.0/16	–

Select a VPC above

2. Create an IGW which will provide internet connectivity to you private network.

3. Attach newly created IGW to you custom VPC.

4. Create two subnets {name should be unique} with range

5. Create a Routing table :- 0.0.0.0/0 go to igw

6. Associate route table to respective subnet.

provider.tf

```
provider "aws" {
  region = "ap-south-1"
  profile = "terraform"
}
```

variable.tf

```
variable "aws_region" {
    default = "ap-south-1"
}

variable "vpc_cidr" {
    default = "10.0.0.0/16"
}
variable "subnets_cidr" {
    type = list
    default = [ "10.0.1.0/24" , "10.0.2.0/24"]
}

variable "azs" {
    default = [ "ap-south-1a" , "ap-south-1b"]
}
```

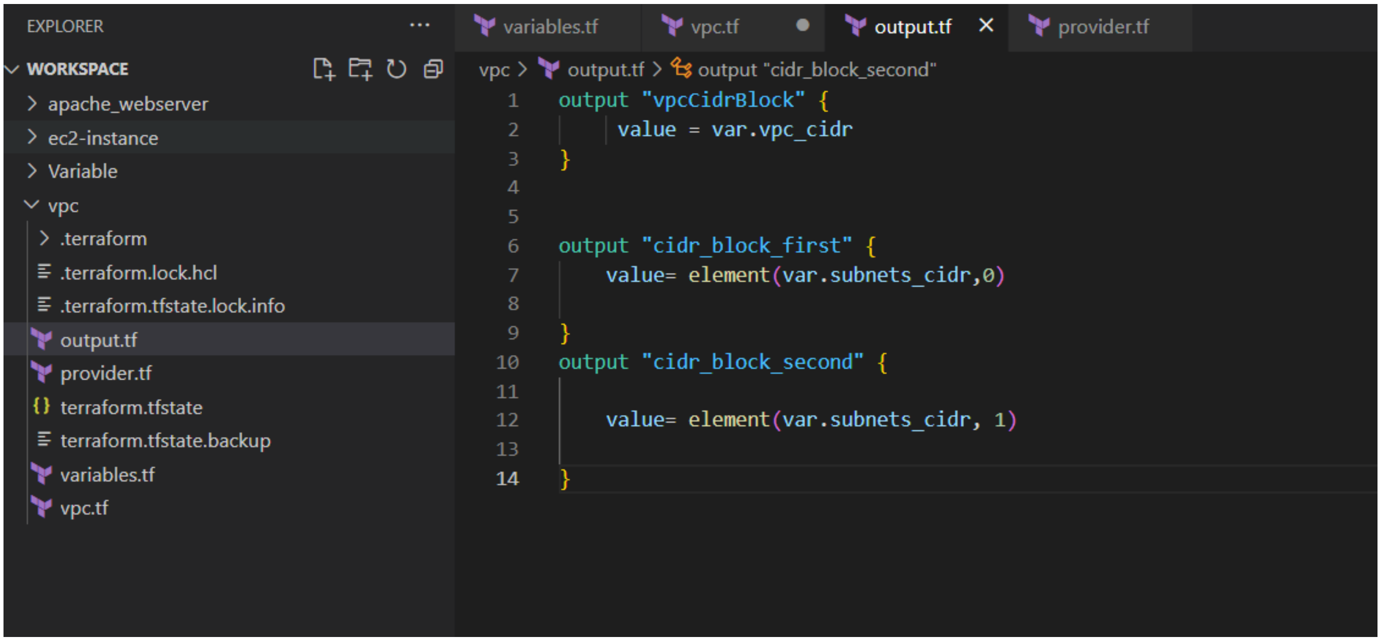
vpc.tf

```
resource "aws_vpc" "main" {
  cidr_block       = var.vpc_cidr
  tags = {
    Name = "myvpc"
  }
}

resource "aws_internet_gateway" "gw" {
  vpc_id = aws_vpc.main.id

  tags = {
    Name = "myigw"
  }
}

resource "aws_subnet" "main" {
  count = length(var.subnets_cidr)
  vpc_id     = aws_vpc.main.id
  cidr_block = element( var.subnets_cidr,  count.index  )
  availability_zone = element( var.azs , count.index )
  map_public_ip_on_launch = true
  tags = {
    Name = "Subnet-${count.index + 1}"
  }
}


resource "aws_route_table" "example" {
  vpc_id = aws_vpc.main.id

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.gw.id
  }



  tags = {
    Name = "MypublicRT"
  }
}



resource "aws_route_table_association" "a" {
     count = length(var.subnets_cidr)
  subnet_id      = element(aws_subnet.main.*.id, count.index)
  route_table_id = aws_route_table.example.id
}
```

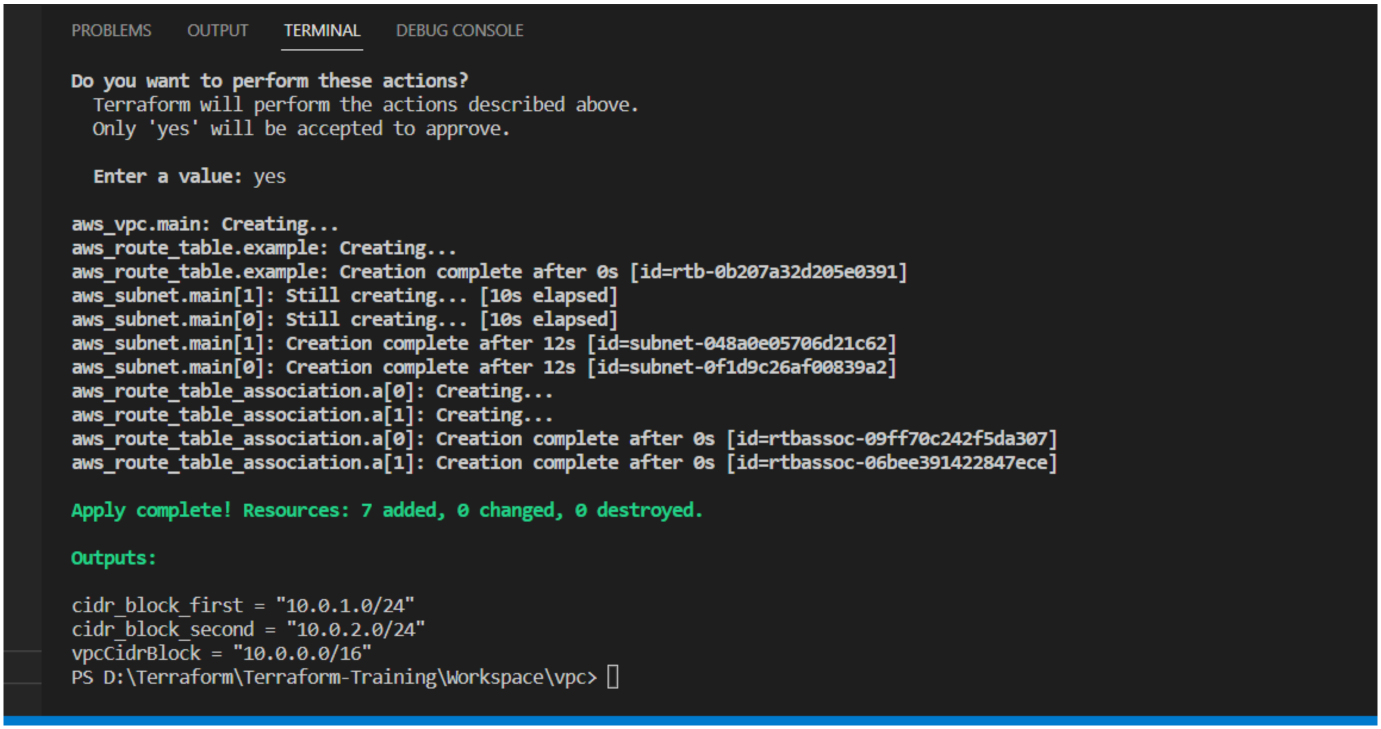
output.tf

```
output "vpcCidrBlock" {
     value = var.vpc_cidr
}


output "cidr_block_first" {
    value= element(var.subnets_cidr,0)

}
output "cidr_block_second" {

    value= element(var.subnets_cidr, 1)

}
```

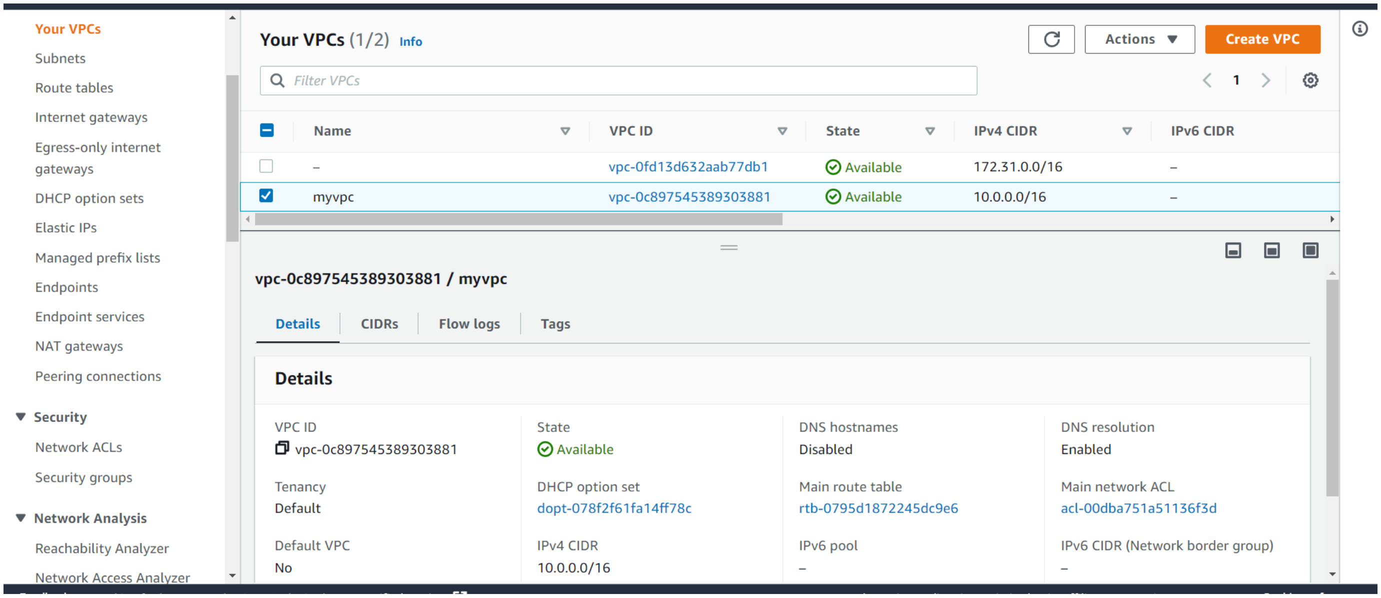
```
terraform plan
terraform apply
```



```
output "vpcCidrBlock" {
    value = var.vpc_cidr
}


output "cidr_block_first" {
    value= element(var.subnets_cidr,0)

}
output "cidr_block_second" {

    value= element(var.subnets_cidr, 1)

}
```



```
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_vpc.main: Creating...
aws_route_table.example: Creating...
aws_route_table.example: Creation complete after 0s [id=rtb-0b207a32d205e0391]
aws_subnet.main[1]: Still creating... [10s elapsed]
aws_subnet.main[0]: Still creating... [10s elapsed]
aws_subnet.main[1]: Creation complete after 12s [id=subnet-048a0e05706d21c62]
aws_subnet.main[0]: Creation complete after 12s [id=subnet-0f1d9c26af00839a2]
aws_route_table_association.a[0]: Creating...
aws_route_table_association.a[1]: Creating...
aws_route_table_association.a[0]: Creation complete after 0s [id=rtbassoc-09ff70c242f5da307]
aws_route_table_association.a[1]: Creation complete after 0s [id=rtbassoc-06bee391422847ece]

Apply complete! Resources: 7 added, 0 changed, 0 destroyed.

Outputs:

cidr_block_first = "10.0.1.0/24"
cidr_block_second = "10.0.2.0/24"
vpcCidrBlock = "10.0.0.0/16"
PS D:\Terraform\Terraform-Training\Workspace\vpc> 
```



Your VPCs (1/2) Info

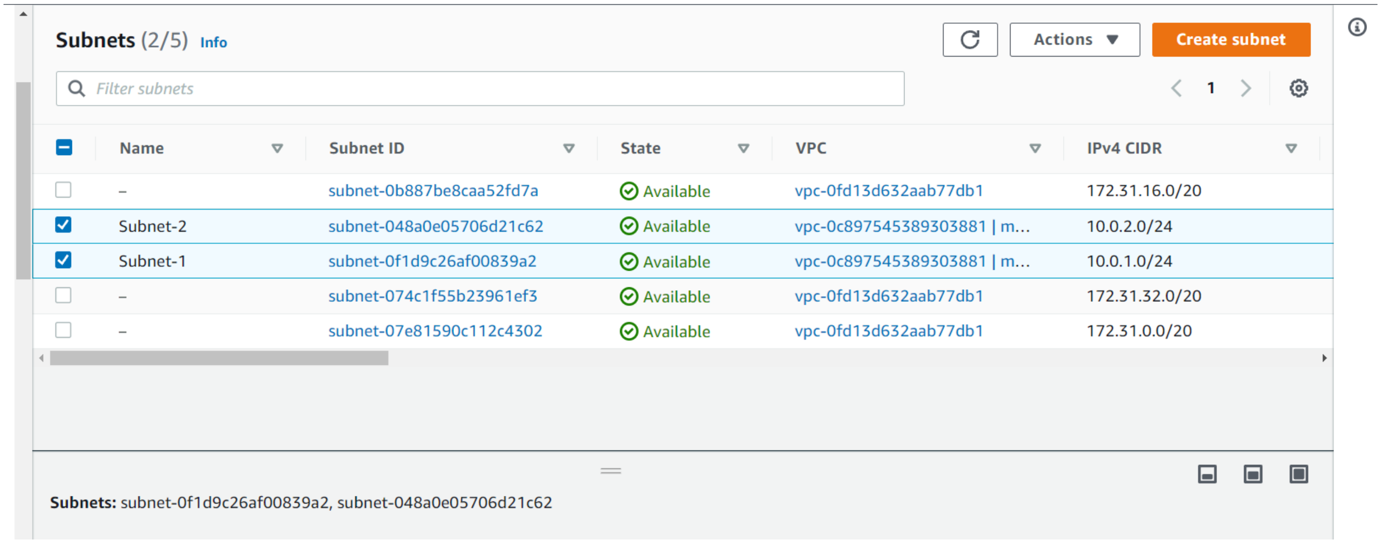
	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
□	–	vpc-0fd13d632aab77db1	✓ Available	172.31.0.0/16	–
☑	myvpc	vpc-0c897545389303881	✓ Available	10.0.0.0/16	–

vpc-0c897545389303881 / myvpc

Details | CIDRs | Flow logs | Tags

Details

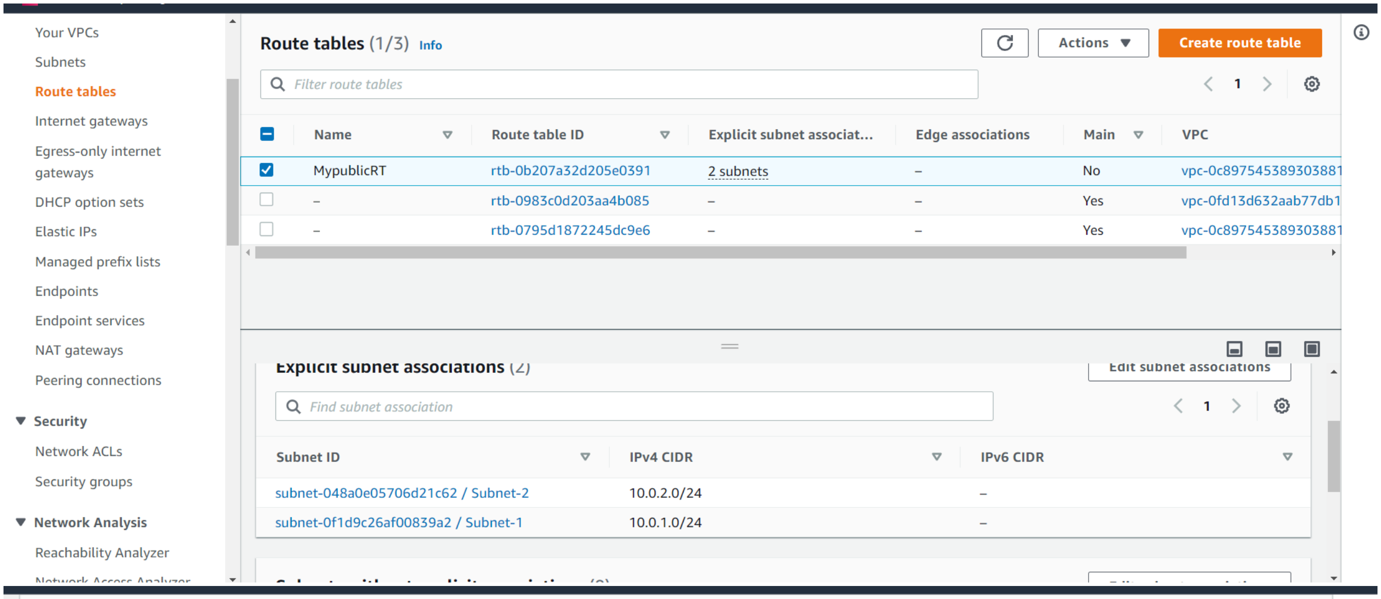
VPC ID vpc-0c897545389303881	State ✓ Available	DNS hostnames Disabled	DNS resolution Enabled
Tenancy Default	DHCP option set dopt-078f2f61fa14ff78c	Main route table rtb-0795d1872245dc9e6	Main network ACL acl-00dba751a51136f3d
Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool –	IPv6 CIDR (Network border group) –



Subnets (2/5) Info

	Name	Subnet ID	State	VPC	IPv4 CIDR
□	–	subnet-0b887be8caa52fd7a	✓ Available	vpc-0fd13d632aab77db1	172.31.16.0/20
☑	Subnet-2	subnet-048a0e05706d21c62	✓ Available	vpc-0c897545389303881 | m...	10.0.2.0/24
☑	Subnet-1	subnet-0f1d9c26af00839a2	✓ Available	vpc-0c897545389303881 | m...	10.0.1.0/24
□	–	subnet-074c1f55b23961ef3	✓ Available	vpc-0fd13d632aab77db1	172.31.32.0/20
□	–	subnet-07e81590c112c4302	✓ Available	vpc-0fd13d632aab77db1	172.31.0.0/20

Subnets: subnet-0f1d9c26af00839a2, subnet-048a0e05706d21c62

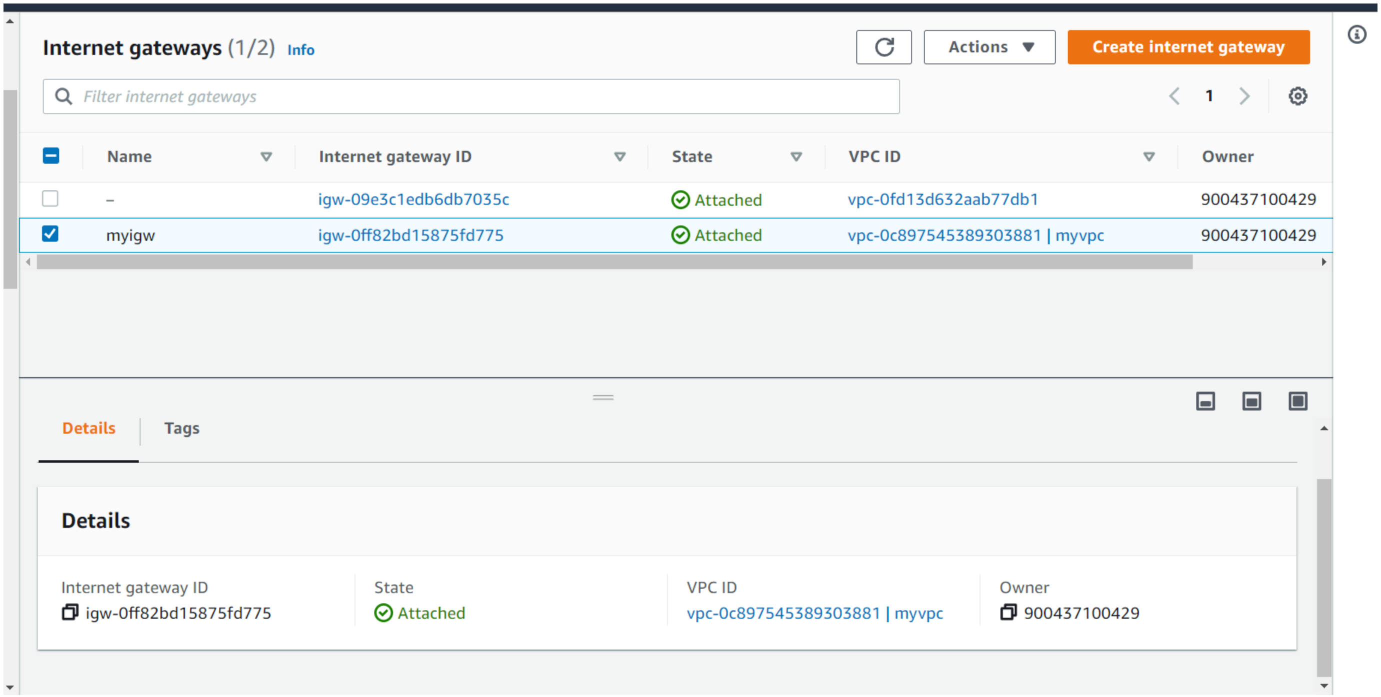


Route tables (1/3) Info

	Name	Route table ID	Explicit subnet associat...	Edge associations	Main	VPC
☑	MypublicRT	rtb-0b207a32d205e0391	2 subnets	–	No	vpc-0c897545389303881
□	–	rtb-0983c0d203aa4b085	–	–	Yes	vpc-0fd13d632aab77db1
□	–	rtb-0795d1872245dc9e6	–	–	Yes	vpc-0c897545389303881

Explicit subnet associations (2)

Subnet ID	IPv4 CIDR	IPv6 CIDR
subnet-048a0e05706d21c62 / Subnet-2	10.0.2.0/24	–
subnet-0f1d9c26af00839a2 / Subnet-1	10.0.1.0/24	–



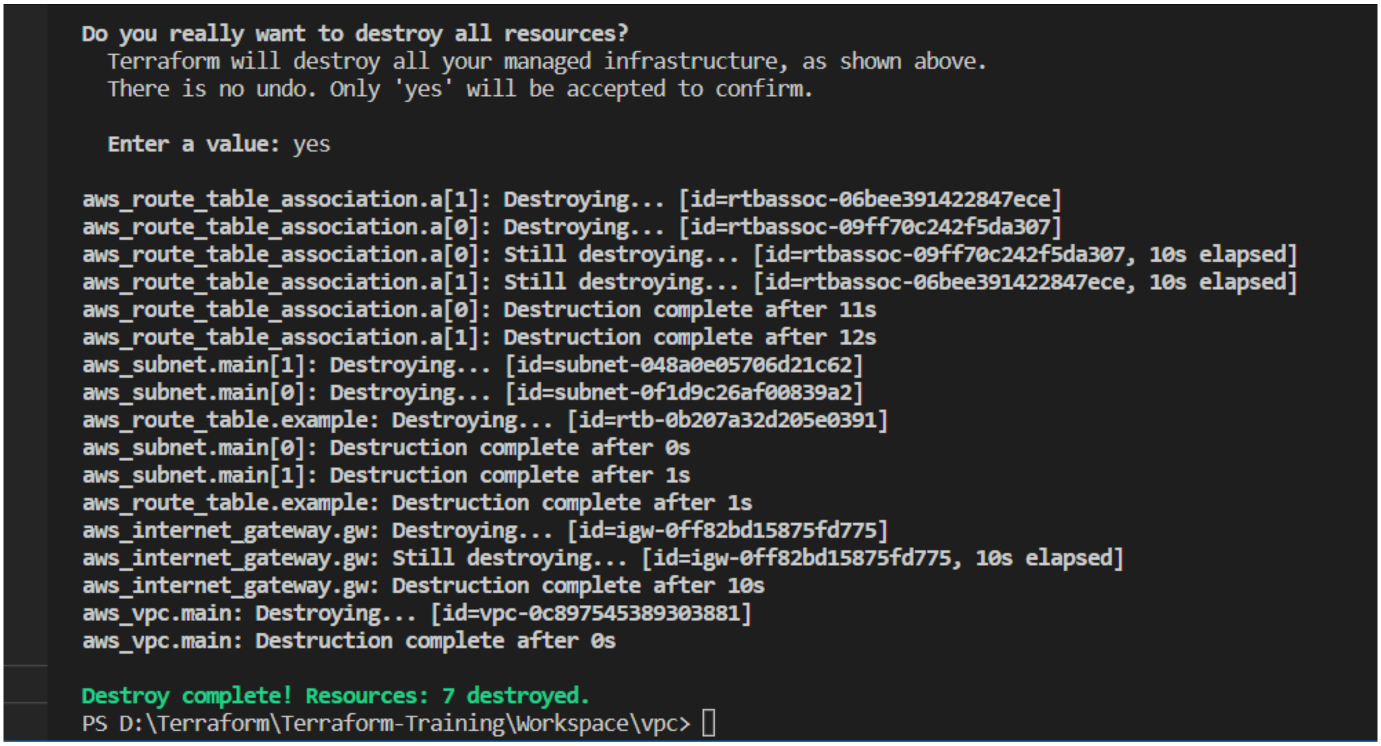
Internet gateways (1/2) Info

	Name	Internet gateway ID	State	VPC ID	Owner
□	–	igw-09e3c1edb6db7035c	✓ Attached	vpc-0fd13d632aab77db1	900437100429
☑	myigw	igw-0ff82bd15875fd775	✓ Attached	vpc-0c897545389303881 | myvpc	900437100429

Details

Internet gateway ID	State	VPC ID	Owner
igw-0ff82bd15875fd775	✓ Attached	vpc-0c897545389303881 | myvpc	900437100429

```
terraform 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_route_table_association.a[1]: Destroying... [id=rtbassoc-06bee391422847ece]
aws_route_table_association.a[0]: Destroying... [id=rtbassoc-09ff70c242f5da307]
aws_route_table_association.a[0]: Still destroying... [id=rtbassoc-09ff70c242f5da307, 10s elapsed]
aws_route_table_association.a[1]: Still destroying... [id=rtbassoc-06bee391422847ece, 10s elapsed]
aws_route_table_association.a[0]: Destruction complete after 11s
aws_route_table_association.a[1]: Destruction complete after 12s
aws_subnet.main[1]: Destroying... [id=subnet-048a0e05706d21c62]
aws_subnet.main[0]: Destroying... [id=subnet-0f1d9c26af00839a2]
aws_route_table.example: Destroying... [id=rtb-0b207a32d205e0391]
aws_subnet.main[0]: Destruction complete after 0s
aws_subnet.main[1]: Destruction complete after 1s
aws_route_table.example: Destruction complete after 1s
aws_internet_gateway.gw: Destroying... [id=igw-0ff82bd15875fd775]
aws_internet_gateway.gw: Still destroying... [id=igw-0ff82bd15875fd775, 10s elapsed]
aws_internet_gateway.gw: Destruction complete after 10s
aws_vpc.main: Destroying... [id=vpc-0c897545389303881]
aws_vpc.main: Destruction complete after 0s

Destroy complete! Resources: 7 destroyed.
PS D:\Terraform\Terraform-Training\Workspace\vpc> 
```

Use terraform modules to create VPC

ref: Click ME

Create a workspace with name “VPC Module”

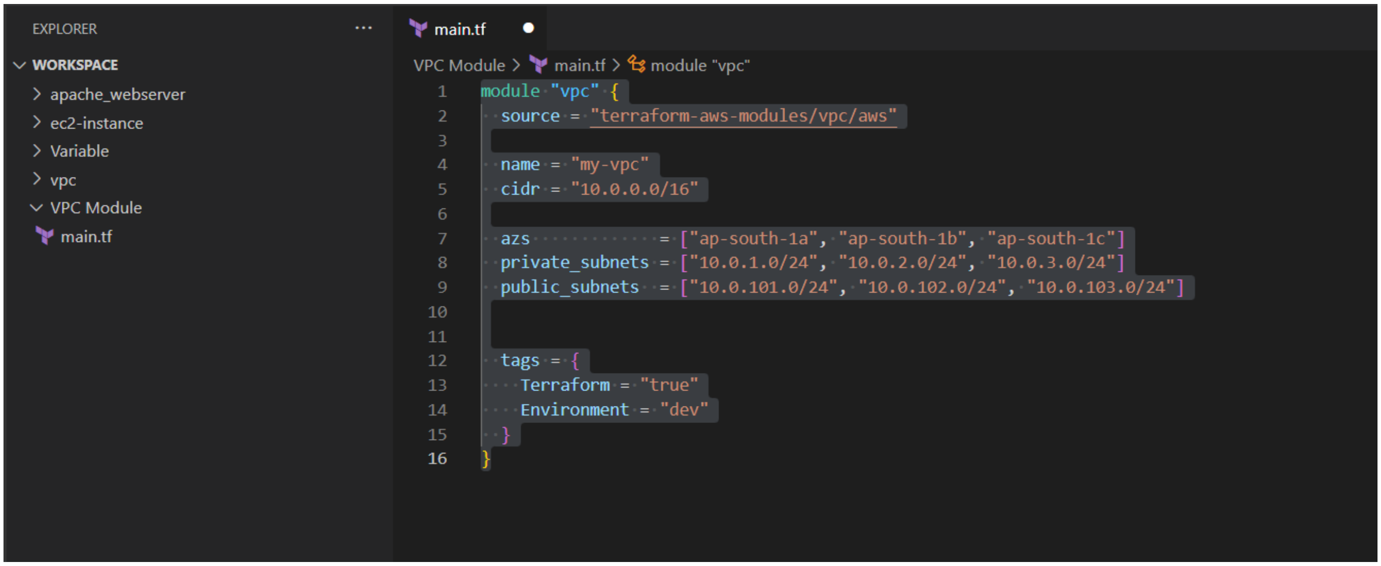
provider.tf

```
provider "aws" {
  region = "ap-south-1"
  profile = "terraform"
}
```

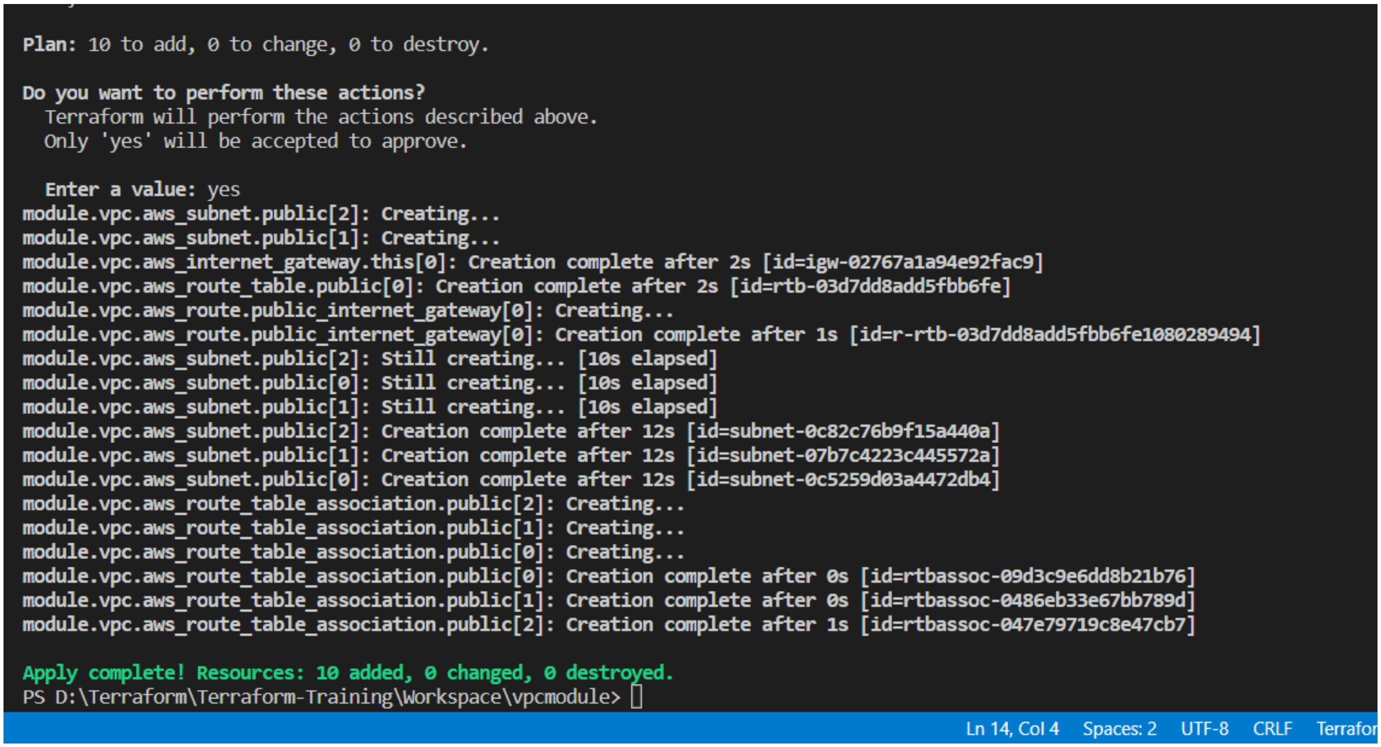
main.tf

```
module "vpc" {
  source = "terraform-aws-modules/vpc/aws"

  name = "my-vpc"
  cidr = "10.0.0.0/16"

  azs             = ["ap-south-1a", "ap-south-1b", "ap-south-1c"]
  public_subnets  = ["10.0.101.0/24", "10.0.102.0/24", "10.0.103.0/24"]


  tags = {
    Terraform = "true"
    Environment = "dev"
  }
}
```

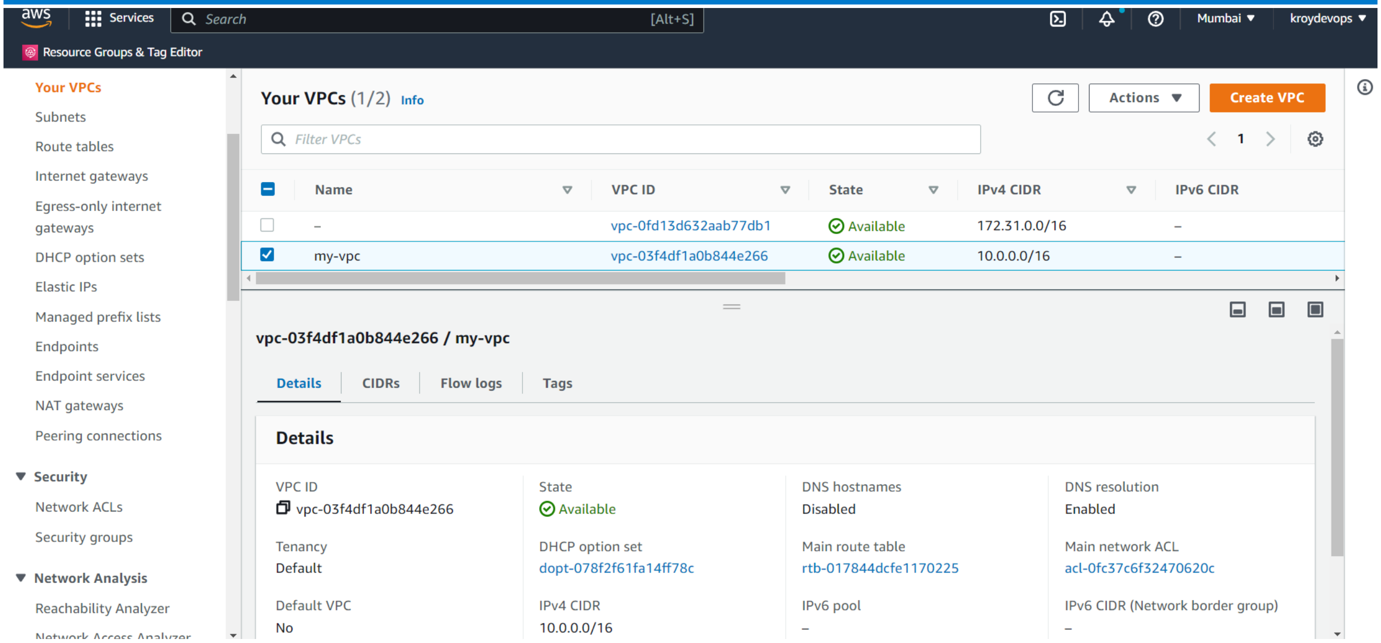
```
terraform init
terraform plan
terraform apply
```



```
module "vpc" {
  source = "terraform-aws-modules/vpc/aws"

  name = "my-vpc"
  cidr = "10.0.0.0/16"

  azs             = ["ap-south-1a", "ap-south-1b", "ap-south-1c"]
  private_subnets = ["10.0.1.0/24", "10.0.2.0/24", "10.0.3.0/24"]
  public_subnets  = ["10.0.101.0/24", "10.0.102.0/24", "10.0.103.0/24"]


  tags = {
    Terraform = "true"
    Environment = "dev"
  }
}
```



```
Plan: 10 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
module.vpc.aws_subnet.public[2]: Creating...
module.vpc.aws_subnet.public[1]: Creating...
module.vpc.aws_internet_gateway.this[0]: Creation complete after 2s [id=igw-02767a1a94e92fac9]
module.vpc.aws_route_table.public[0]: Creation complete after 2s [id=rtb-03d7dd8add5fbb6fe]
module.vpc.aws_route.public_internet_gateway[0]: Creating...
module.vpc.aws_route.public_internet_gateway[0]: Creation complete after 1s [id=r-rtb-03d7dd8add5fbb6fe1080289494]
module.vpc.aws_subnet.public[2]: Still creating... [10s elapsed]
module.vpc.aws_subnet.public[0]: Still creating... [10s elapsed]
module.vpc.aws_subnet.public[1]: Still creating... [10s elapsed]
module.vpc.aws_subnet.public[2]: Creation complete after 12s [id=subnet-0c82c76b9f15a440a]
module.vpc.aws_subnet.public[1]: Creation complete after 12s [id=subnet-07b7c4223c445572a]
module.vpc.aws_subnet.public[0]: Creation complete after 12s [id=subnet-0c5259d03a4472db4]
module.vpc.aws_route_table_association.public[2]: Creating...
module.vpc.aws_route_table_association.public[1]: Creating...
module.vpc.aws_route_table_association.public[0]: Creating...
module.vpc.aws_route_table_association.public[0]: Creation complete after 0s [id=rtbassoc-09d3c9e6dd8b21b76]
module.vpc.aws_route_table_association.public[1]: Creation complete after 0s [id=rtbassoc-0486eb33e67bb789d]
module.vpc.aws_route_table_association.public[2]: Creation complete after 1s [id=rtbassoc-047e79719c8e47cb7]

Apply complete! Resources: 10 added, 0 changed, 0 destroyed.
PS D:\Terraform\Terraform-Training\Workspace\vpcmodule> 
```



Your VPCs (1/2) Info

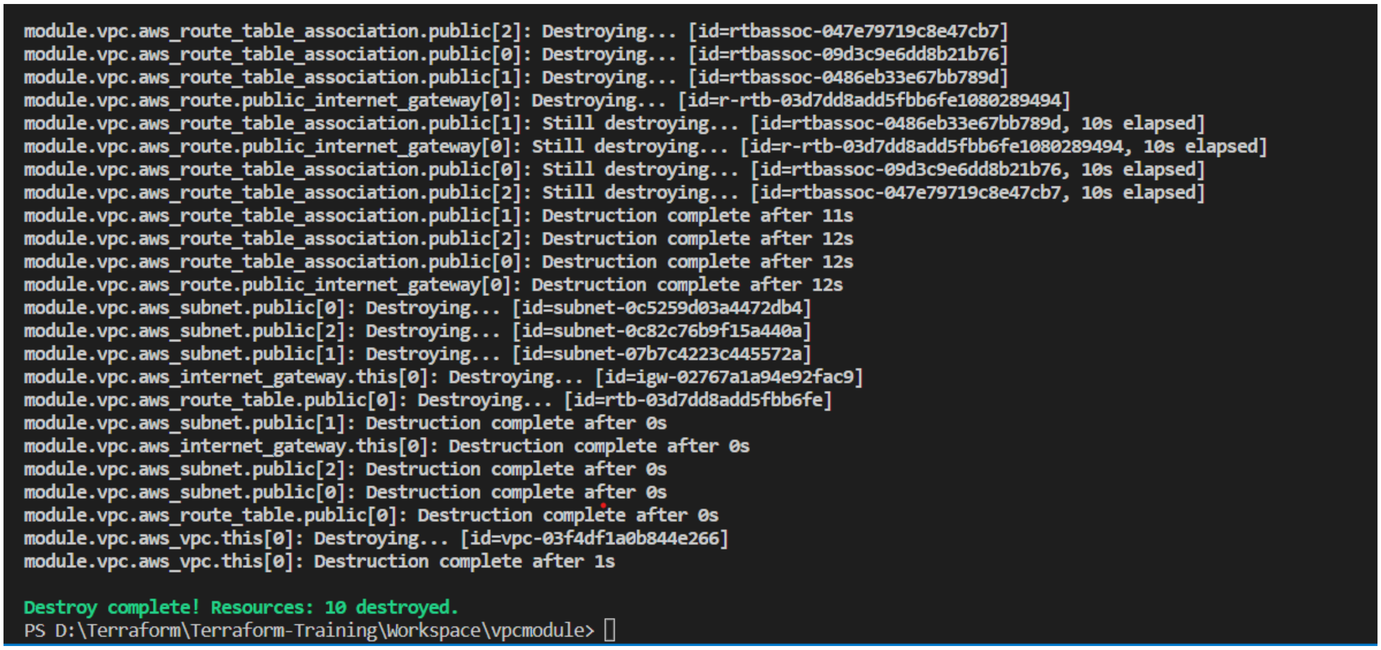
☐	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
☐	–	vpc-0fd13d632aab77db1	✓ Available	172.31.0.0/16	–
☑	my-vpc	vpc-03f4df1a0b844e266	✓ Available	10.0.0.0/16	–

vpc-03f4df1a0b844e266 / my-vpc

Details

VPC ID vpc-03f4df1a0b844e266	State ✓ Available	DNS hostnames Disabled	DNS resolution Enabled
Tenancy Default	DHCP option set dopt-078f2f61fa14ff78c	Main route table rtb-017844dcfe1170225	Main network ACL acl-0fc37c6f32470620c
Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool –	IPv6 CIDR (Network border group) –

```
terraform destroy
```



```
module.vpc.aws_route_table_association.public[2]: Destroying... [id=rtbassoc-047e79719c8e47cb7]
module.vpc.aws_route_table_association.public[0]: Destroying... [id=rtbassoc-09d3c9e6dd8b21b76]
module.vpc.aws_route_table_association.public[1]: Destroying... [id=rtbassoc-0486eb33e67bb789d]
module.vpc.aws_route.public_internet_gateway[0]: Destroying... [id=r-rtb-03d7dd8add5fbb6fe1080289494]
module.vpc.aws_route_table_association.public[1]: Still destroying... [id=rtbassoc-0486eb33e67bb789d, 10s elapsed]
module.vpc.aws_route.public_internet_gateway[0]: Still destroying... [id=r-rtb-03d7dd8add5fbb6fe1080289494, 10s elapsed]
module.vpc.aws_route_table_association.public[0]: Still destroying... [id=rtbassoc-09d3c9e6dd8b21b76, 10s elapsed]
module.vpc.aws_route_table_association.public[2]: Still destroying... [id=rtbassoc-047e79719c8e47cb7, 10s elapsed]
module.vpc.aws_route_table_association.public[1]: Destruction complete after 11s
module.vpc.aws_route_table_association.public[2]: Destruction complete after 12s
module.vpc.aws_route_table_association.public[0]: Destruction complete after 12s
module.vpc.aws_route.public_internet_gateway[0]: Destruction complete after 12s
module.vpc.aws_subnet.public[0]: Destroying... [id=subnet-0c5259d03a4472db4]
module.vpc.aws_subnet.public[2]: Destroying... [id=subnet-0c82c76b9f15a440a]
module.vpc.aws_subnet.public[1]: Destroying... [id=subnet-07b7c4223c445572a]
module.vpc.aws_internet_gateway.this[0]: Destroying... [id=igw-02767a1a94e92fac9]
module.vpc.aws_route_table.public[0]: Destroying... [id=rtb-03d7dd8add5fbb6fe]
module.vpc.aws_subnet.public[1]: Destruction complete after 0s
module.vpc.aws_internet_gateway.this[0]: Destruction complete after 0s
module.vpc.aws_subnet.public[2]: Destruction complete after 0s
module.vpc.aws_subnet.public[0]: Destruction complete after 0s
module.vpc.aws_route_table.public[0]: Destruction complete after 0s
module.vpc.aws_vpc.this[0]: Destroying... [id=vpc-03f4df1a0b844e266]
module.vpc.aws_vpc.this[0]: Destruction complete after 1s

Destroy complete! Resources: 10 destroyed.
PS D:\Terraform\Terraform-Training\Workspace\vpcmodule> 
```

By: Kishan ray