Dr. Shubhangi Kharche

Launching EC2 using terraform

You will learn to install terraform and launch EC2 instance

Requirements:

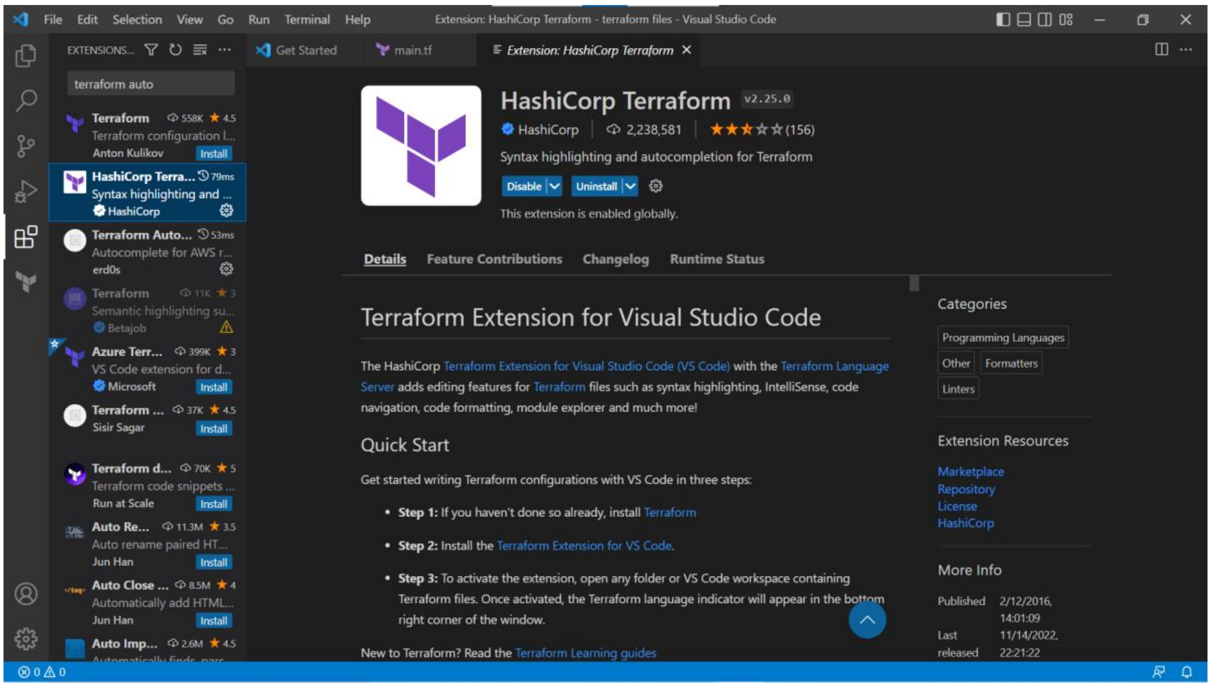
Good editor to code: visual code editor (open source)

Download visual code editor for windows and install it.

https://code.visualstudio.com/download

Install VC extension to support terraform coding

Install Hashi corp terraform and terraform autocomplete extensions and verify whether they are installed successfully.


File Edit Selection View Go Run Terminal Help
Extension: HashiCorp Terraform - terraform files - Visual Studio Code
EXTENSIONS...
Get Started
main.tf
Extension: HashiCorp Terraform
Disable
Uninstall
Details
Feature Contributions
Changelog
Runtime Status
⊗ 0 ⚠ 0

terraform auto

Terraform 558K ★ 4.5
Terraform configuration l...
Anton Kulikov Install

HashiCorp Terra... 79ms
Syntax highlighting and ...
HashiCorp

Terraform Auto... 53ms
Autocomplete for AWS r...
erd0s

Terraform 11K ★ 3
Semantic highlighting su...
Betajob

Azure Terr... 399K ★ 3
VS Code extension for d...
Microsoft Install

Terraform ... 37K ★ 4.5
Sisir Sagar Install

Terraform d... 70K ★ 5
Terraform code snippets ...
Run at Scale Install

Auto Re... 11.3M ★ 3.5
Auto rename paired HT...
Jun Han Install

Auto Close ... 8.5M ★ 4
Automatically add HTML...
Jun Han Install

Auto Imp... 2.6M ★ 4.5

HashiCorp Terraform v2.25.0
HashiCorp | 2,238,581 | ★★★☆☆ (156)
Syntax highlighting and autocompletion for Terraform

This extension is enabled globally.

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.

New to Terraform? Read the Terraform Learning guides

Categories

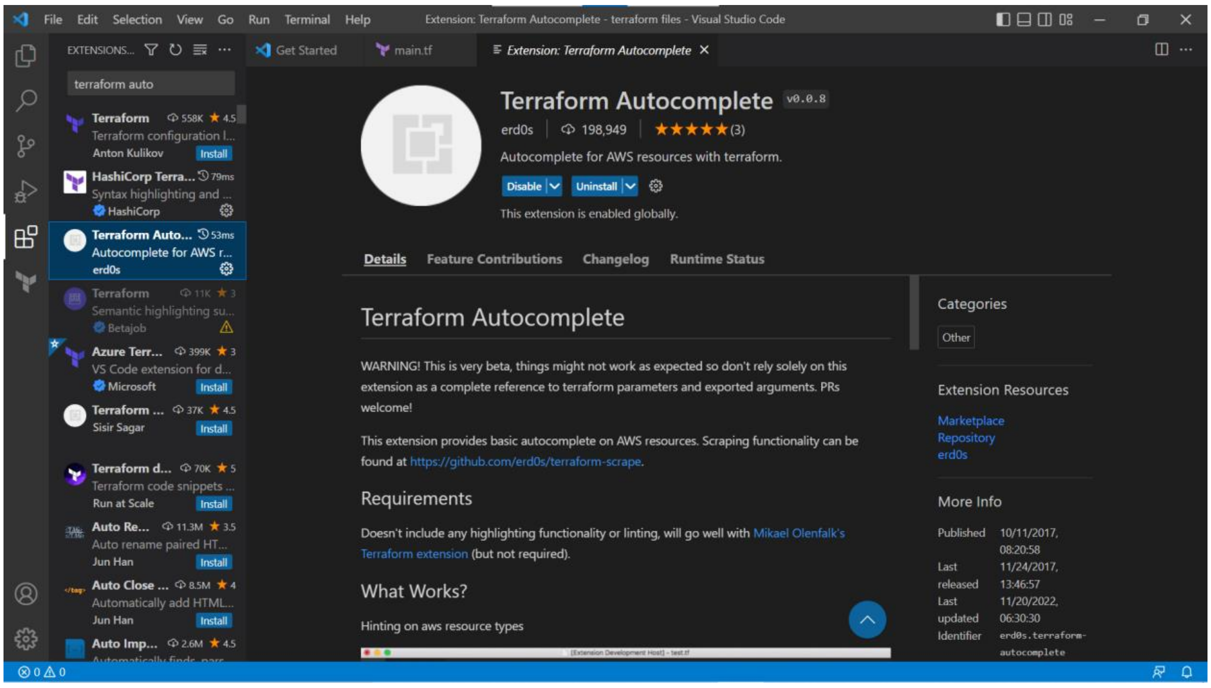
Programming Languages
Other
Formatters
Linters

Extension Resources

Marketplace
Repository
License
HashiCorp

More Info

Published	2/12/2016, 14:01:09
Last released	11/14/2022, 22:21:22



Extension: Terraform Autocomplete - terraform files - Visual Studio Code
Terraform Autocomplete v0.0.8
erd0s 198,949 (3)
Autocomplete for AWS resources with terraform.
Disable Uninstall
This extension is enabled globally.
Details Feature Contributions Changelog Runtime Status

Terraform Autocomplete

WARNING! This is very beta, things might not work as expected so don't rely solely on this extension as a complete reference to terraform parameters and exported arguments. PRs welcome!

This extension provides basic autocomplete on AWS resources. Scraping functionality can be found at https://github.com/erd0s/terraform-scrape.

Requirements

Doesn't include any highlighting functionality or linting, will go well with Mikael Olenfalk's Terraform extension (but not required).

What Works?

Hinting on aws resource types

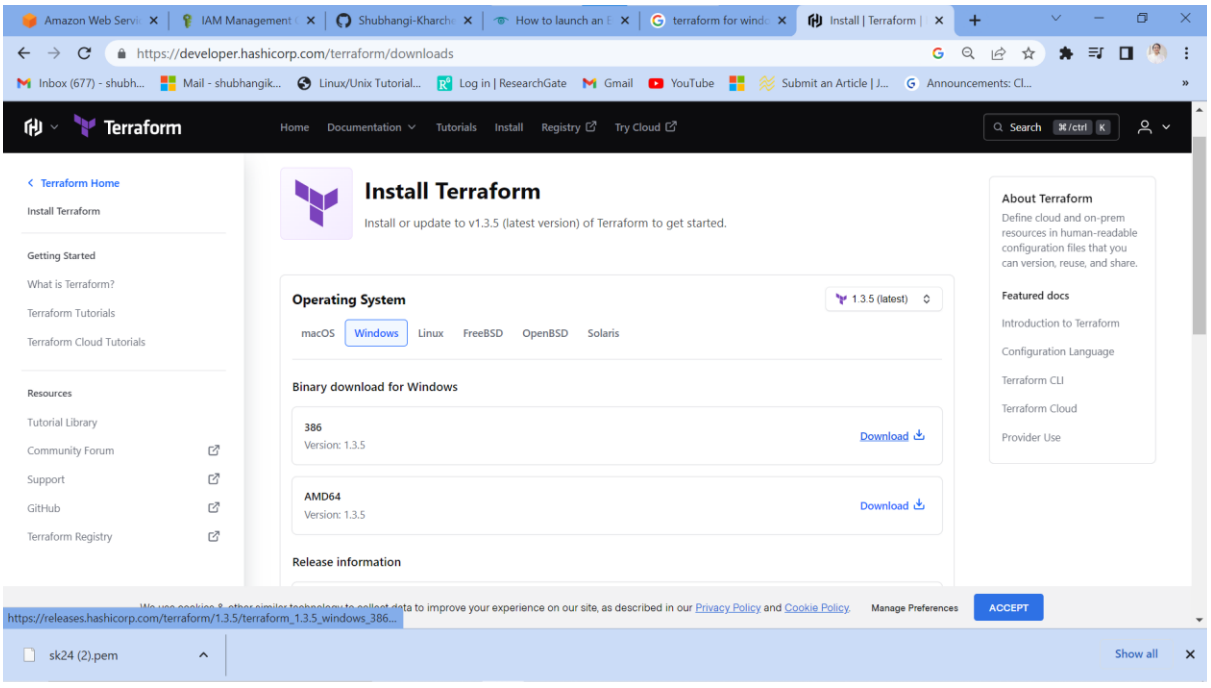
Categories: Other

Extension Resources: Marketplace, Repository, erd0s

More Info
Published 10/11/2017, 08:20:58
Last released 11/24/2017, 13:46:57
Last updated 11/20/2022, 06:30:30
Identifier erd0s.terraform-autocomplete

Download and install terraform for windows and set path variable for it.

https://developer.hashicorp.com/terraform/downloads



Install Terraform

Install or update to v1.3.5 (latest version) of Terraform to get started.

Operating System — 1.3.5 (latest)

macOS [✓ Windows] Linux FreeBSD OpenBSD Solaris

Binary download for Windows

386 Version: 1.3.5 Download

AMD64 Version: 1.3.5 Download

Release information

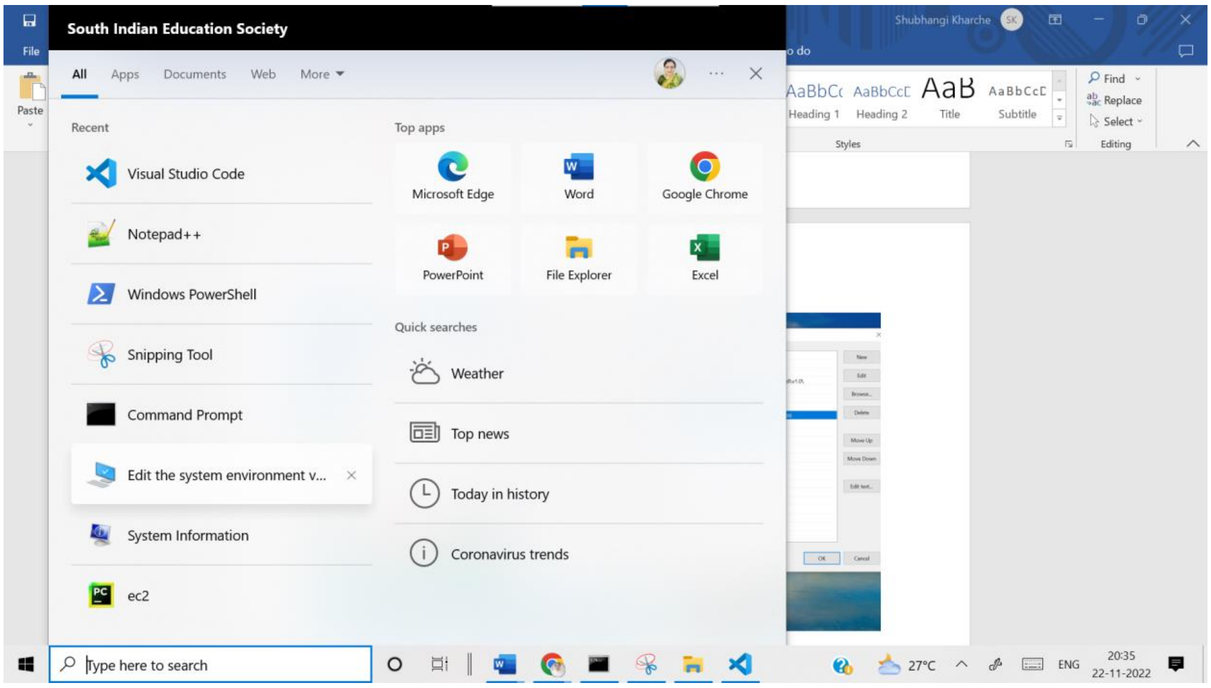
About Terraform: Define cloud and on-prem resources in human-readable configuration files that you can version, reuse, and share.

Featured docs: Introduction to Terraform, Configuration Language, Terraform CLI, Terraform Cloud, Provider Use

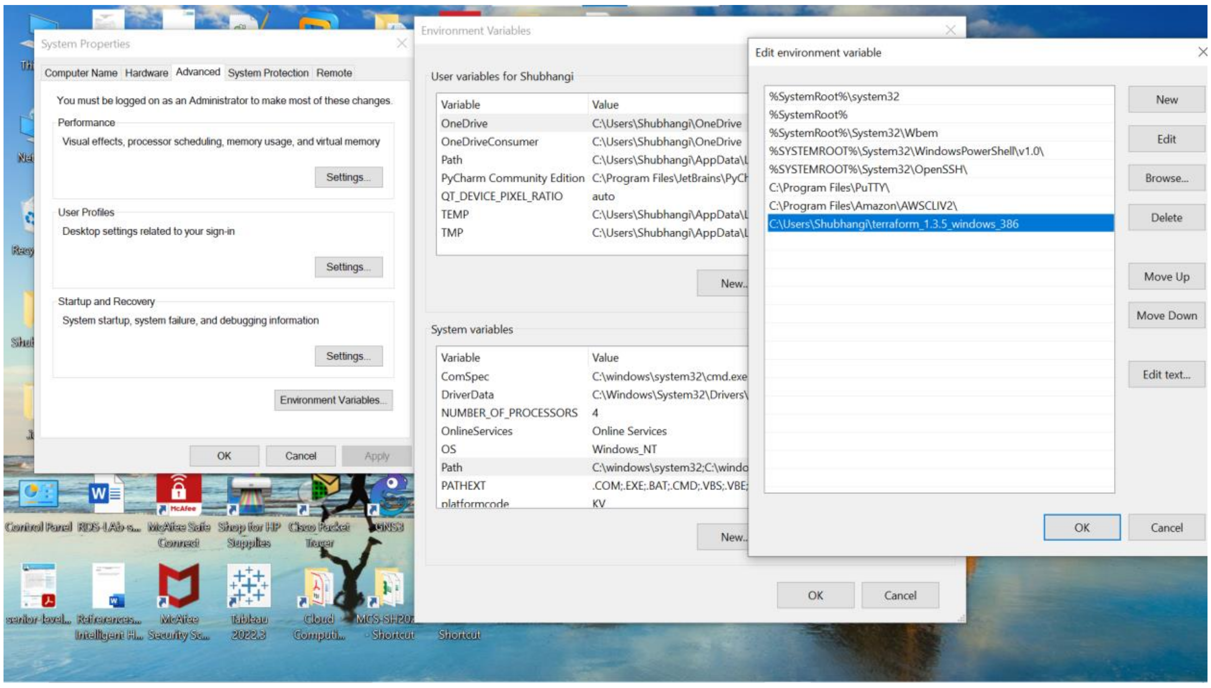
https://releases.hashicorp.com/terraform/1.3.5/terraform_1.3.5_windows_386...

Click on download for 386

Copy this path and set the environment variable



Edit environment variables


Environment Variables

User variables for Shubhangi

Variable	Value
OneDrive	C:\Users\Shubhangi\OneDrive
OneDriveConsumer	C:\Users\Shubhangi\OneDrive
Path	C:\Users\Shubhangi\AppData\L
PyCharm Community Edition	C:\Program Files\JetBrains\PyCh
QT_DEVICE_PIXEL_RATIO	auto
TEMP	C:\Users\Shubhangi\AppData\L
TMP	C:\Users\Shubhangi\AppData\L

System variables

Variable	Value
ComSpec	C:\windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\
NUMBER_OF_PROCESSORS	4
OnlineServices	Online Services
OS	Windows_NT
Path	C:\windows\system32;C:\windo
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;
platformcode	KV

Edit environment variable

```
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files\PuTTY\
C:\Program Files\Amazon\AWSCLIV2\
C:\Users\Shubhangi\terraform_1.3.5_windows_386
```

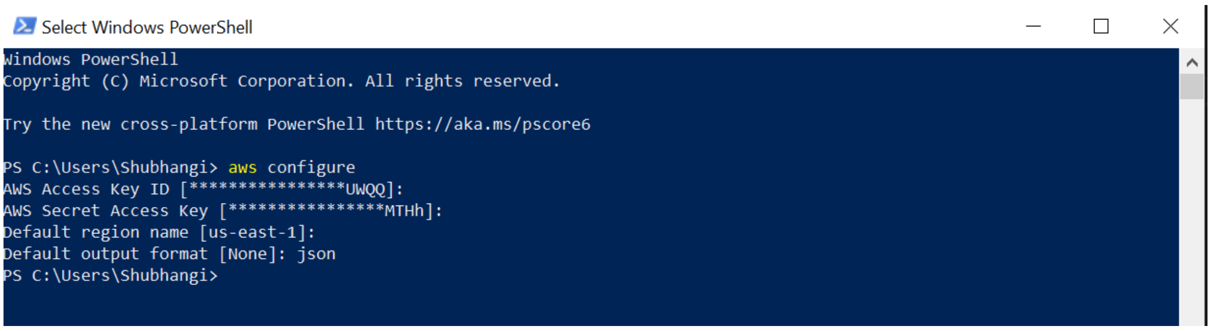
System Properties
Computer Name
Hardware
Advanced
System Protection
Remote
You must be logged on as an Administrator to make most of these changes.
Performance
Visual effects, processor scheduling, memory usage, and virtual memory
Settings...
User Profiles
Desktop settings related to your sign-in
Startup and Recovery
System startup, system failure, and debugging information
Environment Variables...
OK
Cancel
Apply
New...
New
Edit
Browse...
Delete
Move Up
Move Down
Edit text...

Set path for Terraform

Download and install awscli

You should have an AWS account. Create an IAM user, give programmatic access to it, give full access to EC2 instance, note its ACCESS KEY, SECRET ACCESS KEY, and the AWS REGION.

Run **aws configure** command in windows powershell and provide access key, secret access key and AWS region for the created user. Default output format is optional here.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Shubhangi> aws configure
AWS Access Key ID [****************UWQQ]:
AWS Secret Access Key [****************MTHh]:
Default region name [us-east-1]:
Default output format [None]: json
PS C:\Users\Shubhangi>
```

Now, the account will be connected from your terminal.

So, start coding. For this **create a folder** in your working directory to save terraform files. Create a file named **main.tf** in the created folder.

Write the following code to launch EC2 instance:

provider "aws" {

Region = "us-east-1"

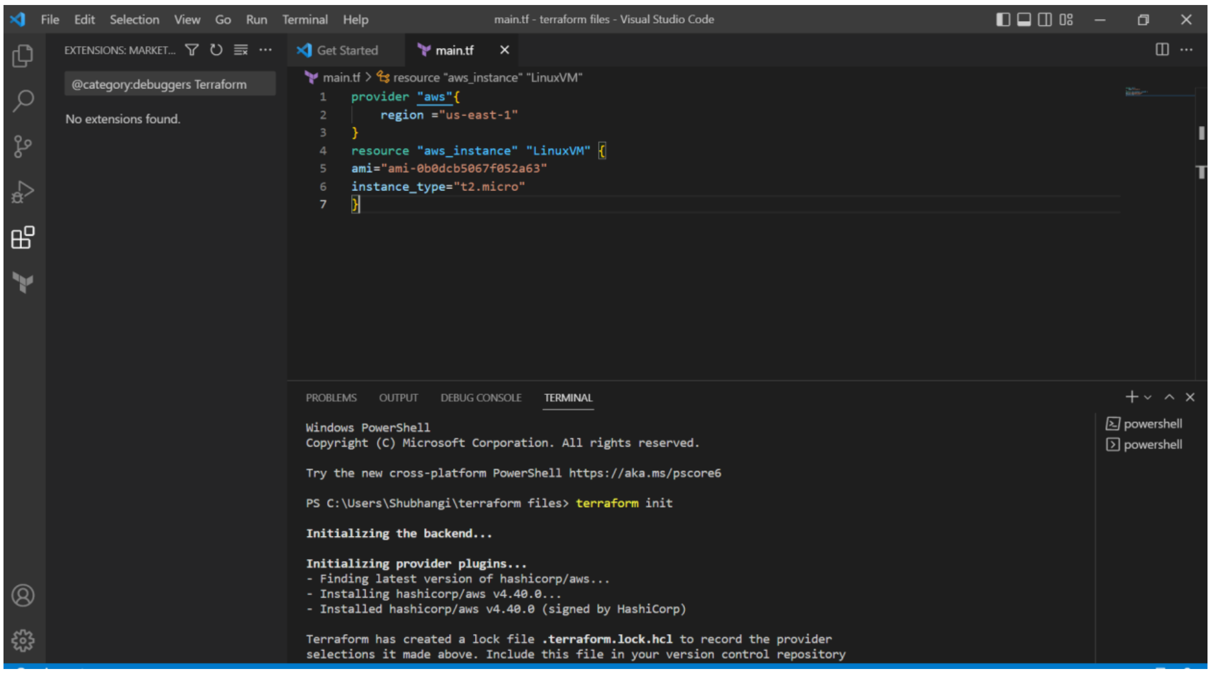
}

resource "aws_instance" "LinuxVM" {

ami = "ami-0b0dcb5067f052a63 "

instance_type = "t2.micro"

}



```
provider "aws"{
    region ="us-east-1"
}
resource "aws_instance" "LinuxVM" {
ami="ami-0b0dcb5067f052a63"
instance_type="t2.micro"
}
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Shubhangi\terraform files> terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v4.40.0...
- Installed hashicorp/aws v4.40.0 (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
```

Run the following commands in windows powershell

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

```
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Shubhangi\terraform files> terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v4.40.0...
- Installed hashicorp/aws v4.40.0 (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
you run "terraform init" in the future.



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.
PS C:\Users\Shubhangi\terraform files> terraform validate
Success! The configuration is valid.

PS C:\Users\Shubhangi\terraform files> 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.LinuxVM will be created
  + resource "aws_instance" "LinuxVM" {
      + ami                                  = "ami-0b0dcb5067f052a63"
      + 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)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = (known after apply)
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id        = (known after apply)
      + private_dns                         = (known after apply)
      + private_ip                          = (known after apply)
      + public_dns                          = (known after apply)
      + public_ip                           = (known after apply)
      + secondary_private_ips               = (known after apply)
      + security_groups                     = (known after apply)
      + source_dest_check                   = true
      + subnet_id                           = (known after apply)
      + tags_all                            = (known after apply)
      + tenancy                             = (known after apply)
      + user_data                           = (known after apply)
      + user_data_base64                    = (known after apply)
      + user_data_replace_on_change         = false
      + vpc_security_group_ids              = (known after apply)

      + capacity_reservation_specification {
        + capacity_reservation_preference = (known after apply)

        + capacity_reservation_target {
          + capacity_reservation_id                 = (known after apply)
          + capacity_reservation_resource_group_arn = (known after apply)
         }
       }

      + ebs_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)
```

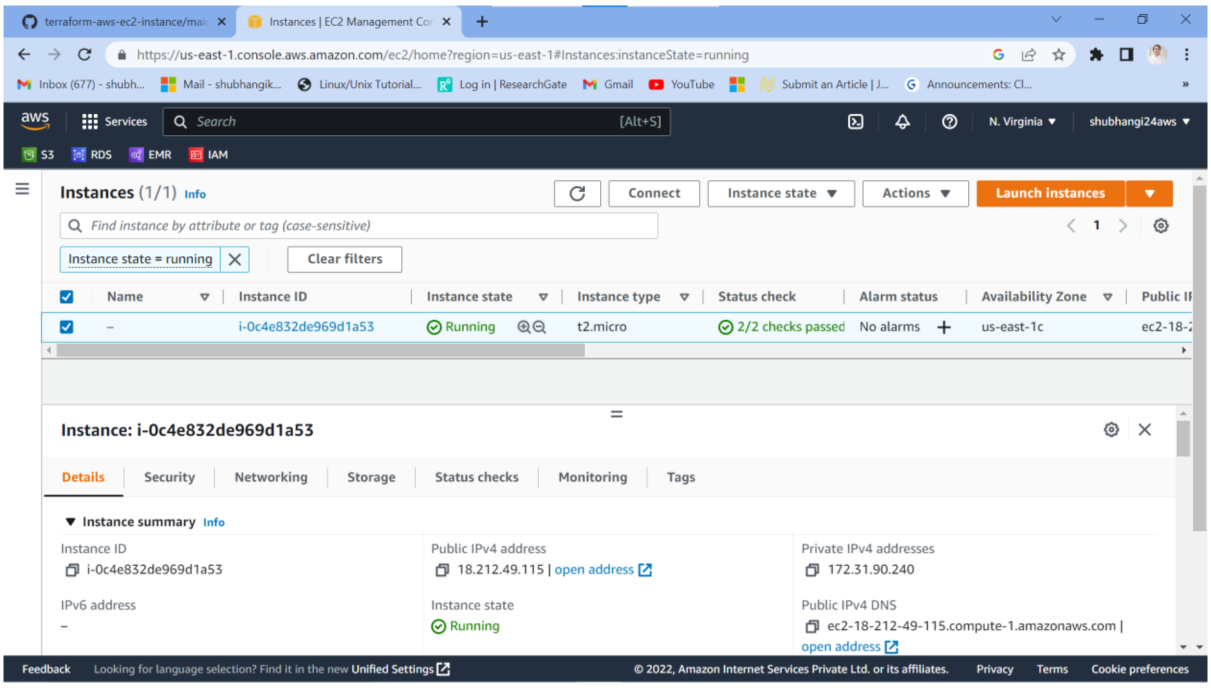
```
      + snapshot_id          = (known after apply)
      + tags                 = (known after apply)
      + throughput           = (known after apply)
      + volume_id            = (known after apply)
      + volume_size          = (known after apply)
      + volume_type          = (known after apply)
    }

  + enclave_options {
      + enabled = (known after apply)
    }

  + ephemeral_block_device {
      + device_name  = (known after apply)
      + no_device    = (known after apply)
      + virtual_name = (known after apply)
    }

  + maintenance_options {
      + auto_recovery = (known after apply)
    }

  + metadata_options {
      + http_endpoint               = (known after apply)
      + http_put_response_hop_limit = (known after apply)
      + http_tokens                 = (known after apply)
      + instance_metadata_tags      = (known after apply)
    }

  + network_interface {
      + delete_on_termination = (known after apply)
      + device_index         = (known after apply)
      + network_card_index    = (known after apply)
      + network_interface_id  = (known after apply)
    }

  + private_dns_name_options {
      + enable_resource_name_dns_a_record    = (known after apply)
      + enable_resource_name_dns_aaaa_record = (known after apply)
      + hostname_type                        = (known after apply)
    }

  + root_block_device {
      + delete_on_termination = (known after apply)
      + device_name          = (known after apply)
      + encrypted            = (known after apply)
      + iops                 = (known after apply)
      + kms_key_id           = (known after apply)
      + tags                 = (known after apply)
      + throughput           = (known after apply)
      + volume_size          = (known after apply)
      + volume_type          = (known after apply)
    }
  }

Plan: 1 to add, 0 to change, 0 to destroy.

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

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these
actions if you run "terraform apply" now.
```

```
PS C:\Users\Shubhangi\terraform files> terraform apply


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


Terraform will perform the following actions:


  # aws_instance.LinuxVM will be created
  + resource "aws_instance" "LinuxVM" {
      + ami                                  = "ami-0b0dcb5067f052a63"
      + 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)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = (known after apply)
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
      + subnet_id                            = (known after apply)
      + tags_all                             = (known after apply)
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)

      + capacity_reservation_specification {
         + capacity_reservation_preference = (known after apply)

         + capacity_reservation_target {
            + capacity_reservation_id                 = (known after apply)
            + capacity_reservation_resource_group_arn = (known after apply)
          }
       }

      + ebs_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)
    + snapshot_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)
  }

+ enclave_options {
    + enabled = (known after apply)
  }

+ ephemeral_block_device {
    + device_name  = (known after apply)
    + no_device    = (known after apply)
    + virtual_name = (known after apply)
  }

+ maintenance_options {
    + auto_recovery = (known after apply)
  }

+ metadata_options {
    + http_endpoint               = (known after apply)
    + http_put_response_hop_limit = (known after apply)
    + http_tokens                 = (known after apply)
    + instance_metadata_tags      = (known after apply)
```

```
      }

    + network_interface {
        + delete_on_termination = (known after apply)
        + device_index          = (known after apply)
        + network_card_index    = (known after apply)
        + network_interface_id  = (known after apply)
      }

    + private_dns_name_options {
        + enable_resource_name_dns_a_record    = (known after apply)
        + enable_resource_name_dns_aaaa_record = (known after apply)
        + hostname_type                        = (known after apply)
      }

    + root_block_device {
        + delete_on_termination = (known after apply)
        + device_name           = (known after apply)
        + encrypted             = (known after apply)
        + iops                  = (known after apply)
        + kms_key_id            = (known after apply)
        + tags                  = (known after apply)
        + throughput            = (known after apply)
        + volume_id             = (known after apply)
        + volume_size           = (known after apply)
        + volume_type           = (known after apply)
      }
  }

Plan: 1 to add, 0 to change, 0 to destroy.
```

```
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.LinuxVM: Creating...
aws_instance.LinuxVM: Still creating... [10s elapsed]
aws_instance.LinuxVM: Still creating... [20s elapsed]
aws_instance.LinuxVM: Still creating... [30s elapsed]
aws_instance.LinuxVM: Creation complete after 36s [id=i-0c4e832de969d1a53]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\Users\Shubhangi\terraform files>
```



Instances (1/1) Info

Find instance by attribute or tag (case-sensitive)

Instance state = running

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IF
–	i-0c4e832de969d1a53	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c	ec2-18-2

Instance: i-0c4e832de969d1a53

Instance summary Info

Instance ID
i-0c4e832de969d1a53

Public IPv4 address
18.212.49.115 | open address

Private IPv4 addresses
172.31.90.240

IPv6 address
–

Instance state
Running

Public IPv4 DNS
ec2-18-212-49-115.compute-1.amazonaws.com | open address

Successfully launched EC2 instance using Terraform