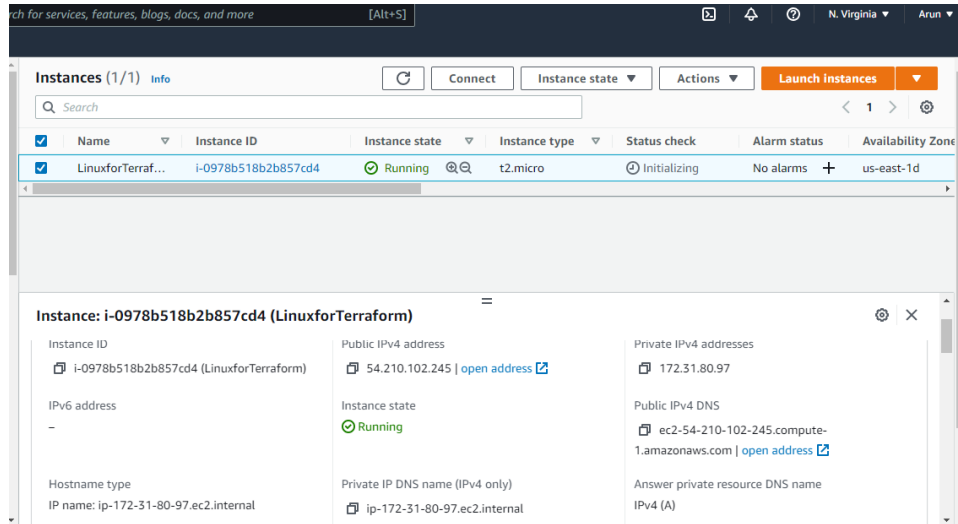


# Installation of Terraform in Linux, and creation of ec2 with the help of terraform

Step1 : Create one Linux EC2 for installing Terraform.



Step 2 : Connect to SSH perform the following commands

Sudo su -

wget [https://releases.hashicorp.com/terraform/0.13.5/terraform\\_0.13.5\\_linux\\_amd64.zip](https://releases.hashicorp.com/terraform/0.13.5/terraform_0.13.5_linux_amd64.zip)

unzip terraform\_0.13.5\_linux\_amd64.zip

./terraform

Terraform -version

Mkdir projects

Cd projects

Touch main.tf

Vim main.tf

This will create a file inside the folder projects

```
us-east-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-0978b518b2b857cd4

archive: terraform_0.13.5_linux_amd64.zip
inflating: terraform
root@ip-172-31-80-97 ~]# ./terraform
Usage: terraform [-version] [-help] <command> [args]

The available commands for execution are listed below.
The most common, useful commands are shown first, followed by
less common or more advanced commands. If you're just getting
started with Terraform, stick with the common commands. For the
other commands, please read the help and docs before usage.

Common commands:
  apply          Builds or changes infrastructure
  console        Interactive console for Terraform interpolations
  destroy        Destroy Terraform-managed infrastructure
  env            Workspace management
  fmt            Rewrites config files to canonical format
  get            Download and install modules for the configuration
  graph          Create a visual graph of Terraform resources
  import         Import existing infrastructure into Terraform
  init           Initialize a Terraform working directory
  login          Obtain and save credentials for a remote host
  logout         Remove locally-stored credentials for a remote host
  output         Read an output from a state file
  plan           Generate and show an execution plan
  providers      Prints a tree of the providers used in the configuration
```

i-0978b518b2b857cd4 (LinuxforTerraform)

Step 3 : in the main.tf enter the following commands

```
provider "aws" {
  profile = "default"
  region  = "us-east-1"

  access_key = "AKIA6LFFGBWBBSIMLF53J"
  secret_key = "not written here for security purpose"
}

resource "aws_instance" "name" {
  ami = "ami-0c293f3f676ec4f90"
  instance_type = "t2.micro"
}
```

```
--> Package terraform.x86_64 0:1.1.7-1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

Package Arch Version Repository
-----
Installing:
terraform x86_64 1.1.7-1 hashicorp

Transaction Summary
Install 1 Package

Total download size: 12 M
Installed size: 60 M
Downloading packages:
warning: /var/cache/yum/x86_64/2/hashicorp/packages/terraform-1.1.7-1.x86_64.rpm: Header V4 RSA/SHA512 Signature, key ID a3219f7b: NOKEY
Public key for terraform-1.1.7-1.x86_64.rpm is not installed
terraform-1.1.7-1.x86_64.rpm
Retrieving key from https://rpm.releases.hashicorp.com/gpg
Importing GPG key 0xA3219F7B:
Userid : HashiCorp Security (HashiCorp Package Signing) <security@packaging.hashicorp.com>
Fingerprint: e8a0 32e0 94d8 eb4e a189 d270 da41 8c88 a321 9f7b
From : https://rpm.releases.hashicorp.com/gpg
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : terraform-1.1.7-1.x86_64
Verifying : terraform-1.1.7-1.x86_64

Installed:
terraform.x86_64 0:1.1.7-1

complete!
root@ip-172-31-88-124 ~]# terraform -v
Terraform v1.1.7
on linux_amd64
root@ip-172-31-88-124 ~]# |
```

Step 4 : After checking terraform version, after entering above commands in main.tf . Enter terraform init

```
root@ip-172-31-88-124 ~# terraform -v
terraform v1.1.7
root@ip-172-31-88-124 ~# cd projects
root@ip-172-31-88-124 ~# mkdir project1
root@ip-172-31-88-124 project1# cd project1
root@ip-172-31-88-124 project1# touch main.tf
root@ip-172-31-88-124 project1# cat main.tf
provider "aws" {
  region = "us-east-1"
  access_key = "AKIAQ3LMD3G4CN7DMCP5"
  secret_key = "S0u2t4vKdJ77xTWCsqd7bGh3K44F0zFDOFymaA1"
}

resource "aws_instance" "name" {
  ami = "ami-0c3933f67e6ca490"
  instance_type = "t2.micro"
  tags = {
    Name = "terraform1"
  }
}

root@ip-172-31-88-124 project1# terraform init
initializing the backend...

initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v4.4.0...
- Installed hashicorp/aws v4.4.0 (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,
run this command to initialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
root@ip-172-31-88-124 project1#
```

Step 5 : terraform add

```
+ ephemeral_block_device {
+   device_name = (known after apply)
+   no_device   = (known after apply)
+   virtual_name = (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_interface_id = (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.name: Creating...
aws_instance.name: Still creating... [10s elapsed]
aws_instance.name: Still creating... [20s elapsed]
aws_instance.name: Still creating... [30s elapsed]
aws_instance.name: Still creating... [40s elapsed]
aws_instance.name: Creation complete after 42s [id= i-09d44519b8167948c]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
root@ip-172-31-88-124 project1#
```

Step 6 : Now resources created and go to console check the new instance created

th for services, features, blogs, docs, and more

[Alt+S]

N. Virginia

Arun

Instances (1/2) Info

Connect

Instance state

Actions

Launch instances

Search

< 1 >

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	LinuxforTerraform...	i-0978b518b2b857cd4	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d
<input checked="" type="checkbox"/>	-	i-0ebf9e01f82929ee3	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d

Instance: i-0ebf9e01f82929ee3

Details

Security

Networking

Storage

Status checks

Monitoring

Tags

▼ Instance summary Info

Instance ID	Public IPv4 address	Private IPv4 addresses
<div>i-0ebf9e01f82929ee3</div>	<div>18.206.251.1   <a href="#">open address</a></div>	<div>172.31.92.183</div>
IPv6 address	Instance state	Public IPv4 DNS
-	Running	<div>ec2-18-206-251-1.compute-</div>