Terraform

What is terraform

- ➢ Terraform is a tool for building, changing, and versioning infrastructure safely and efficiently.
- ➢ Free and open source IAAS(infrastructure as a service) tool build by Hashicorp.
- ➢ Terraform is used to create any cloud infrastructure.
- ➢ Written by Hashicrop Configuration Language (HCL).

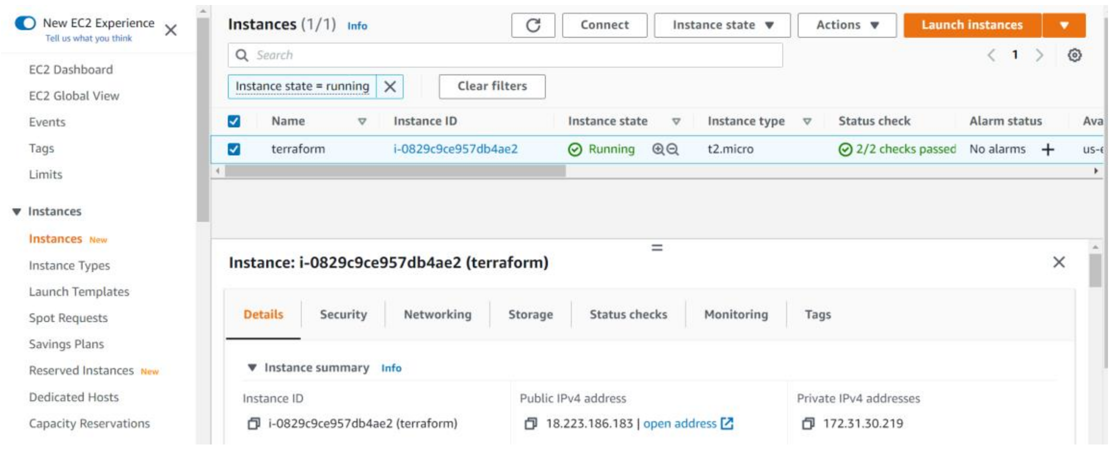
Benefits of terraform

- ➢ Once you are creating the infrastructure using terraform file, again you can use same terraform file to replicate multiple infrastructure.
- ➢ Easily identified who can make the changes in the terraform file(github commit).

Terraform setup

Step 1: installing and setup terraform

- ✧ first go to login into aws console > go to ec2 instance > create ec2 instance (linux).
- ✧ Go to IAM console > create role > attach that role to ec2 instance.



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Ava
terraform	i-0829c9ce957db4ae2	Running	t2.micro	2/2 checks passed	No alarms	us-e

Instance: i-0829c9ce957db4ae2 (terraform)

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0829c9ce957db4ae2 (terraform)	18.223.186.183 | open address	172.31.30.219

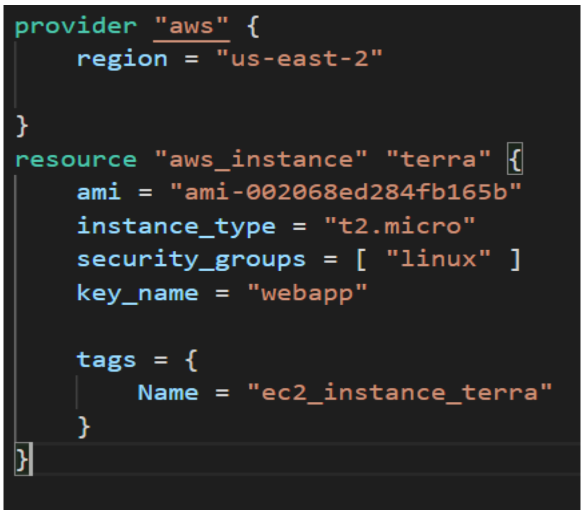
- ✧ Next install terraform

```
$ sudo yum install -y yum-utils
$ sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo
$ sudo yum -y install terraform
```

Step 2 : creating EC2 instance

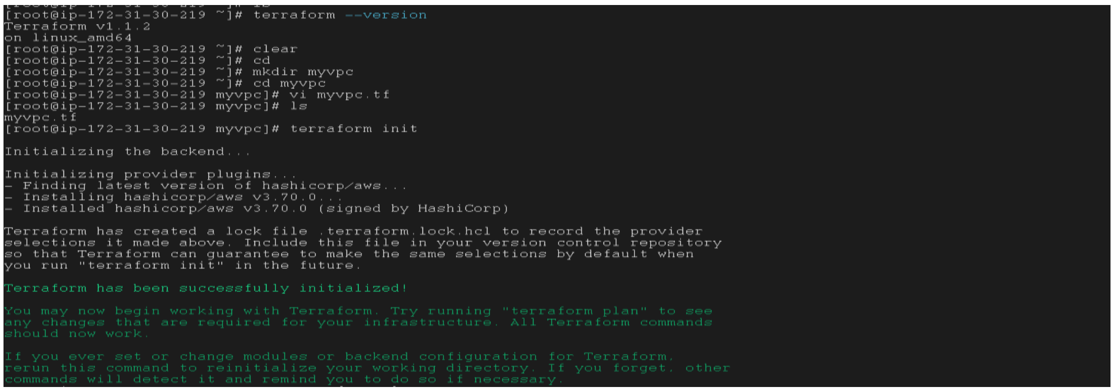
- ✧ Create directory sample ***mkdir ec2_instance*** > go to directory sample ***cd ec2_instance.***
- ✧ Create ec2_instance infrastructure file.

Main.tf



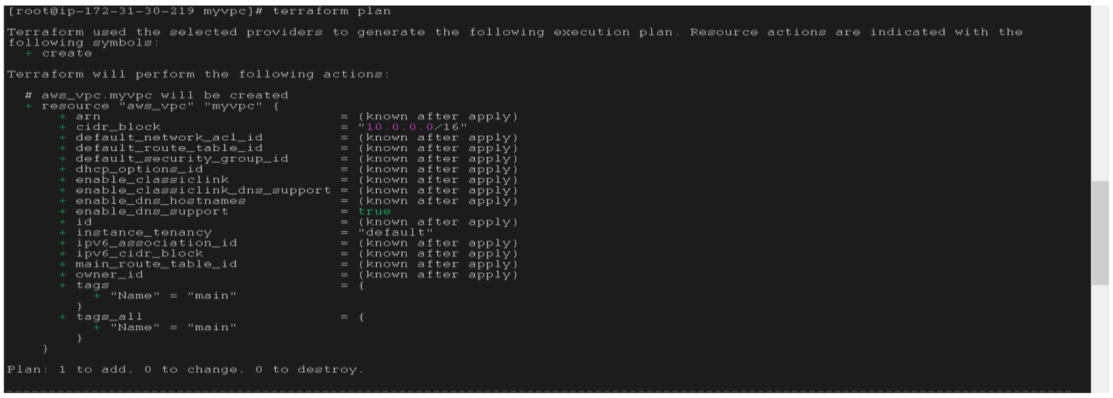
```
provider "aws" {
    region = "us-east-2"

}
resource "aws_instance" "terra" {
    ami = "ami-002068ed284fb165b"
    instance_type = "t2.micro"
    security_groups = [ "linux" ]
    key_name = "webapp"

    tags = {
        Name = "ec2_instance_terra"
    }
}
```

- ✧ Now build the infrastructure.
- ✧ Execute ***terraform init*** > initialize a working directory that contains a Terraform configuration.



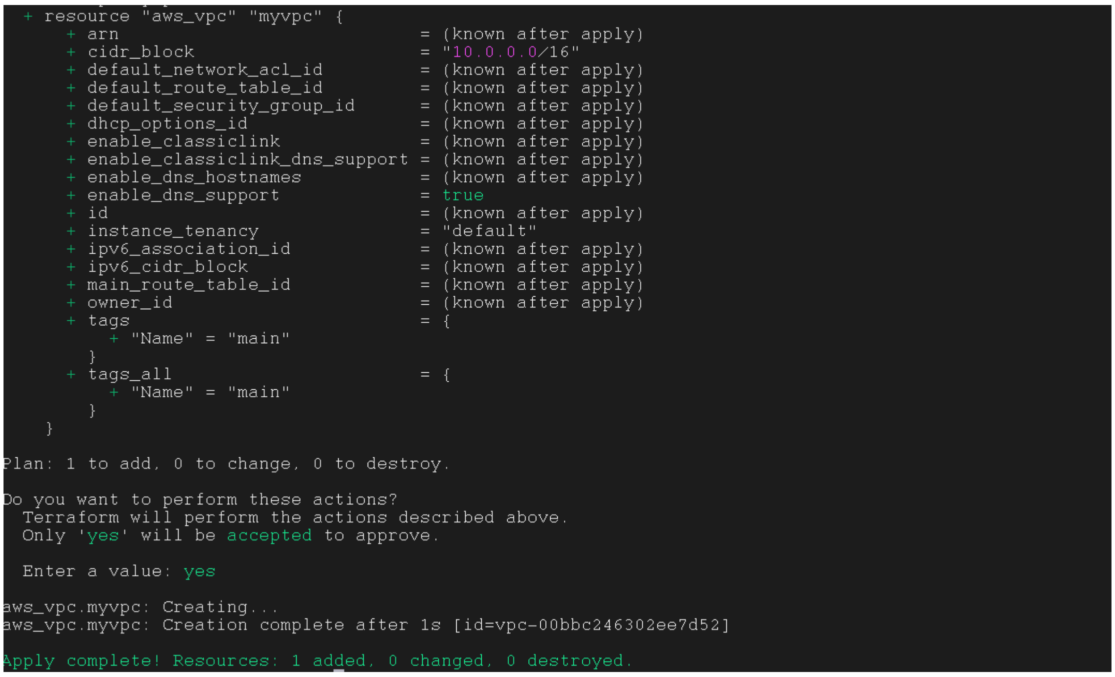
```
[root@ip-172-31-30-219 ~]# terraform --version
Terraform v1.1.2
on linux_amd64
[root@ip-172-31-30-219 ~]# clear
[root@ip-172-31-30-219 ~]# cd
[root@ip-172-31-30-219 ~]# mkdir myvpc
[root@ip-172-31-30-219 ~]# cd myvpc
[root@ip-172-31-30-219 myvpc]# vi myvpc.tf
[root@ip-172-31-30-219 myvpc]# ls
myvpc.tf
[root@ip-172-31-30-219 myvpc]# terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v3.70.0...
- Installed hashicorp/aws v3.70.0 (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.
```

- ✧ Next, execute ***terraform plan >*** creates an execution plan.



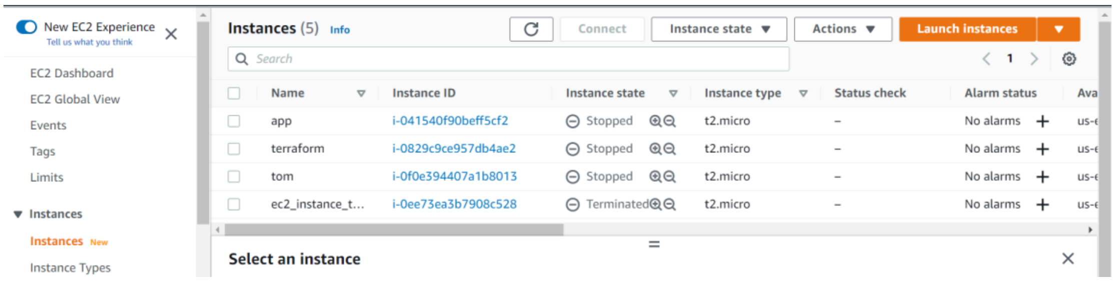
```
[root@ip-172-31-30-219 myvpc]# terraform plan

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_vpc.myvpc will be created
  + resource "aws_vpc" "myvpc" {
      + 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_classiclink               = (known after apply)
      + enable_classiclink_dns_support   = (known after apply)
      + enable_dns_hostnames             = (known after apply)
      + enable_dns_support               = true
      + id                               = (known after apply)
      + instance_tenancy                 = "default"
      + ipv6_association_id              = (known after apply)
      + ipv6_cidr_block                  = (known after apply)
      + main_route_table_id              = (known after apply)
      + owner_id                         = (known after apply)
      + tags                             = {
          + "Name" = "main"
        }
      + tags_all                         = {
          + "Name" = "main"
        }
    }

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

✧ Execute ***terraform apply*** > executes the actions proposed in a Terraform plan.



```
  + resource "aws_vpc" "myvpc" {
      + 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_classiclink               = (known after apply)
      + enable_classiclink_dns_support   = (known after apply)
      + enable_dns_hostnames             = (known after apply)
      + enable_dns_support               = true
      + id                               = (known after apply)
      + instance_tenancy                 = "default"
      + ipv6_association_id              = (known after apply)
      + ipv6_cidr_block                  = (known after apply)
      + main_route_table_id              = (known after apply)
      + owner_id                         = (known after apply)
      + tags                             = {
          + "Name" = "main"
        }
      + tags_all                         = {
          + "Name" = "main"
        }
    }

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_vpc.myvpc: Creating...
aws_vpc.myvpc: Creation complete after 1s [id=vpc-00bbc246302ee7d52]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

✧ Go to aws EC2 console and see instance is created.



Instances (5) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Ava
app	i-041540f90beff5cf2	Stopped	t2.micro	–	No alarms	us-e
terraform	i-0829c9ce957db4ae2	Stopped	t2.micro	–	No alarms	us-e
tom	i-0f0e394407a1b8013	Stopped	t2.micro	–	No alarms	us-e
ec2_instance_t...	i-0ee73ea3b7908c528	Terminated	t2.micro	–	No alarms	us-e

Select an instance