EC2-Creation

Install Terraform on ‘Amazon linux’

1. sudo yum install -y yum-utils shadow-utils
2. sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo
3. sudo yum -y install terraform

Steps for create EC2

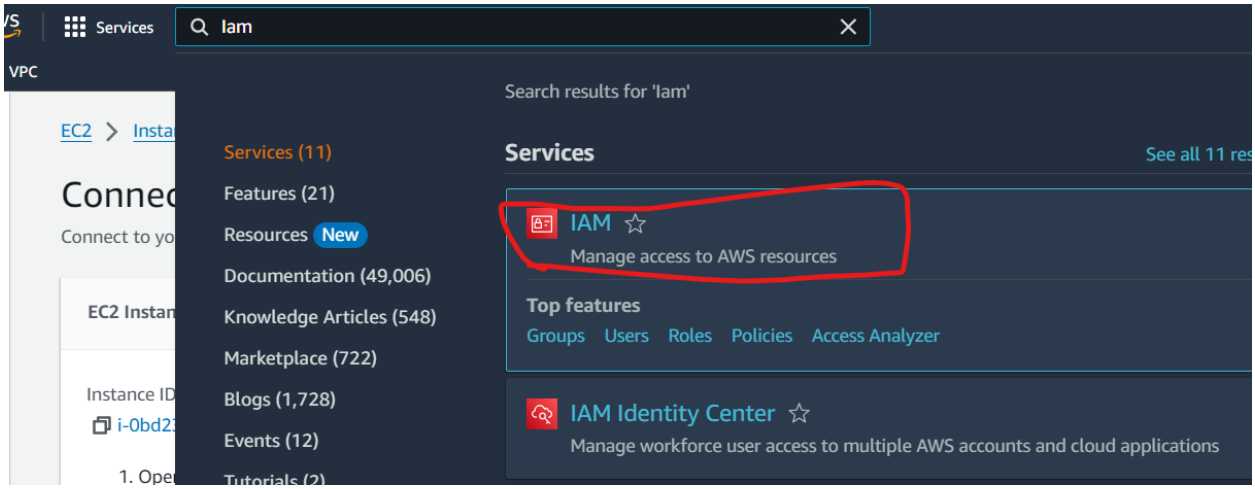
1. Create script file by extension ‘.tf’

```
provider "aws" {
   access_key = "AKIATM4VCLAT5UMA6KN5"
   secret_key = "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"
   region = "ap-south-1"
}
resource "aws_instance" "ec2_instance" {
   ami = "ami-03f4878755434977f"
   instance_type = "t2.micro"
   key_name = "terraform"
}
```

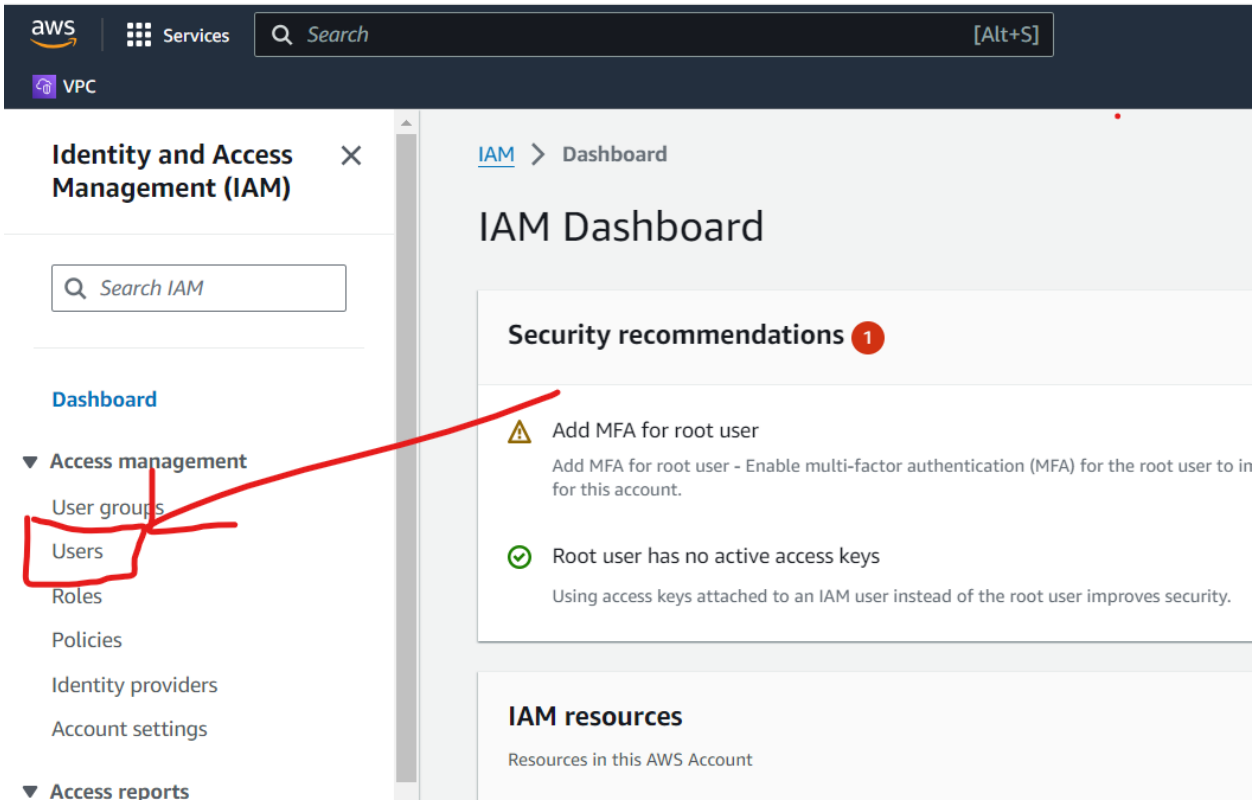
@ access key & secret key create by creating IAM user
@ ami - O.S image id
@ instance type - ram
@ key name - key pair

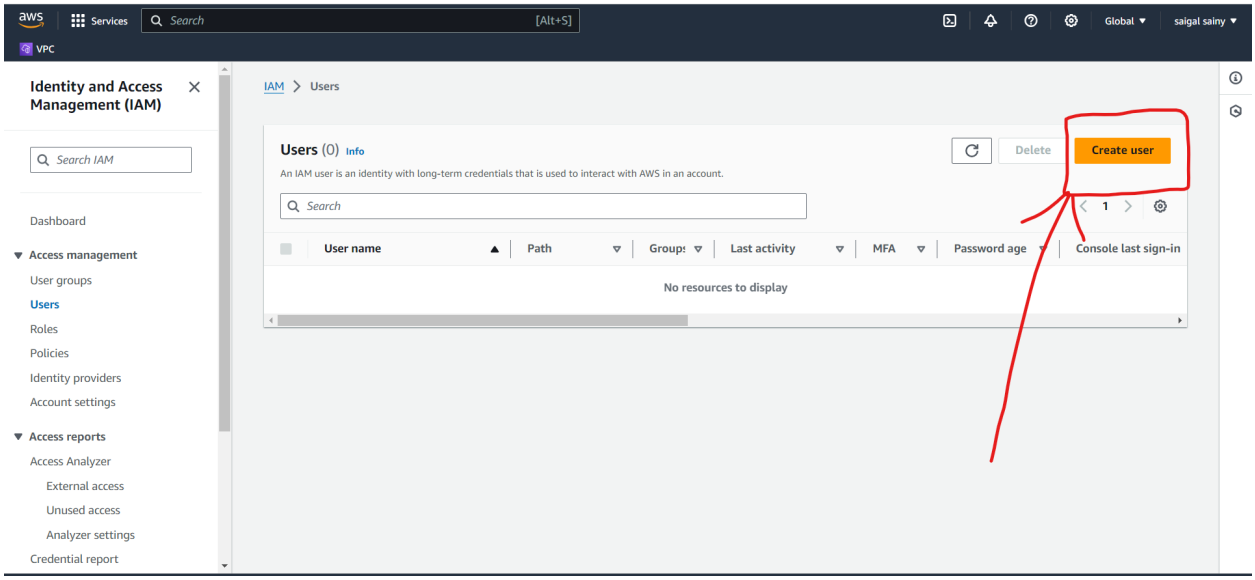
→ Create access key:-

1. Go to IAM in AWS

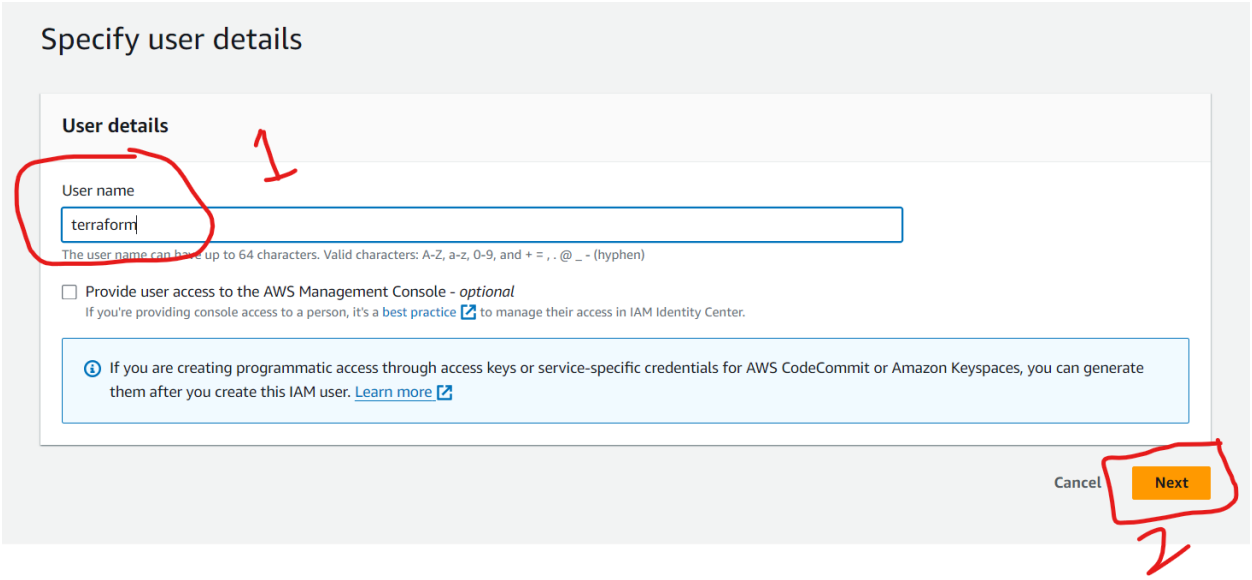


2. Create user from left side



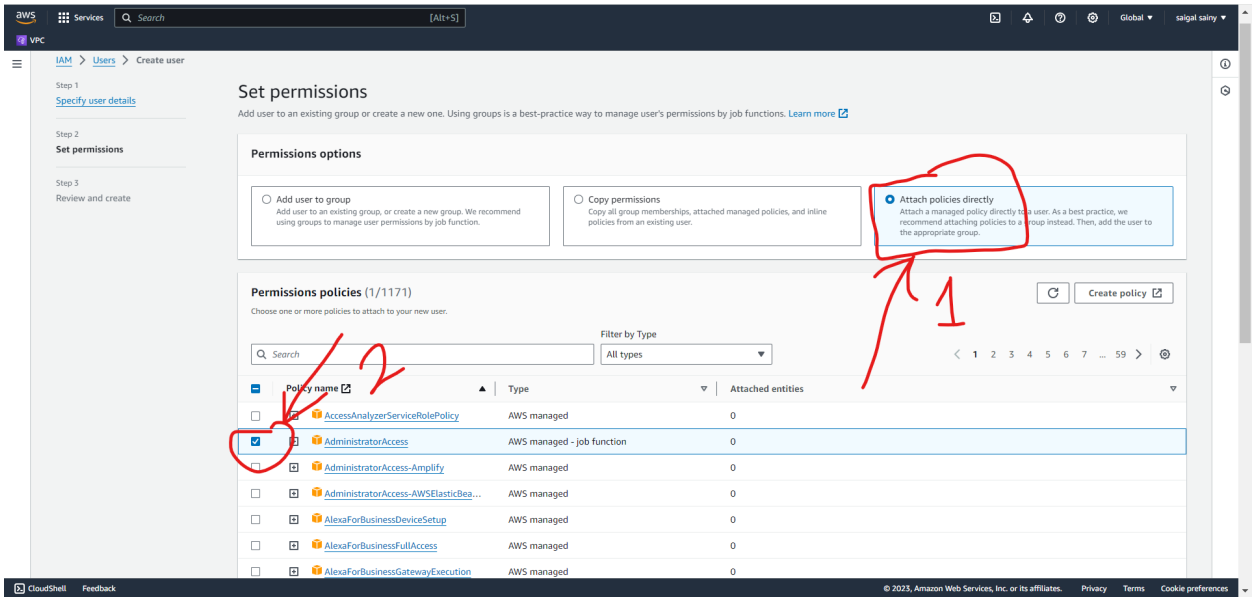


3. Write name and click save

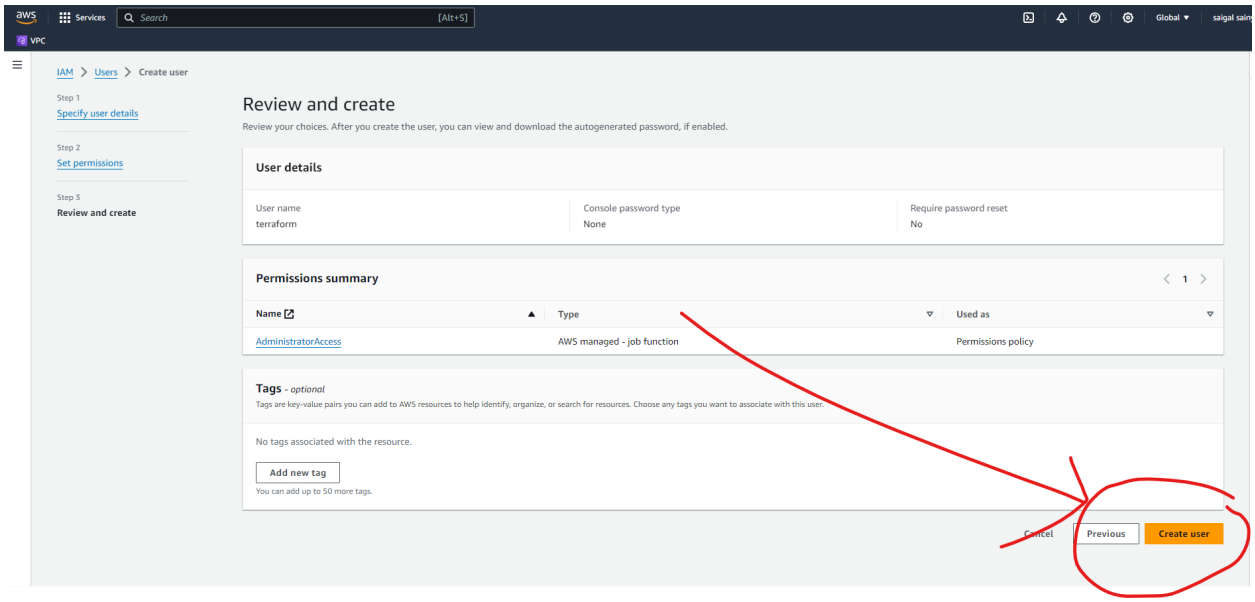


User created

4. Set permission by follow below steps and click ‘next’

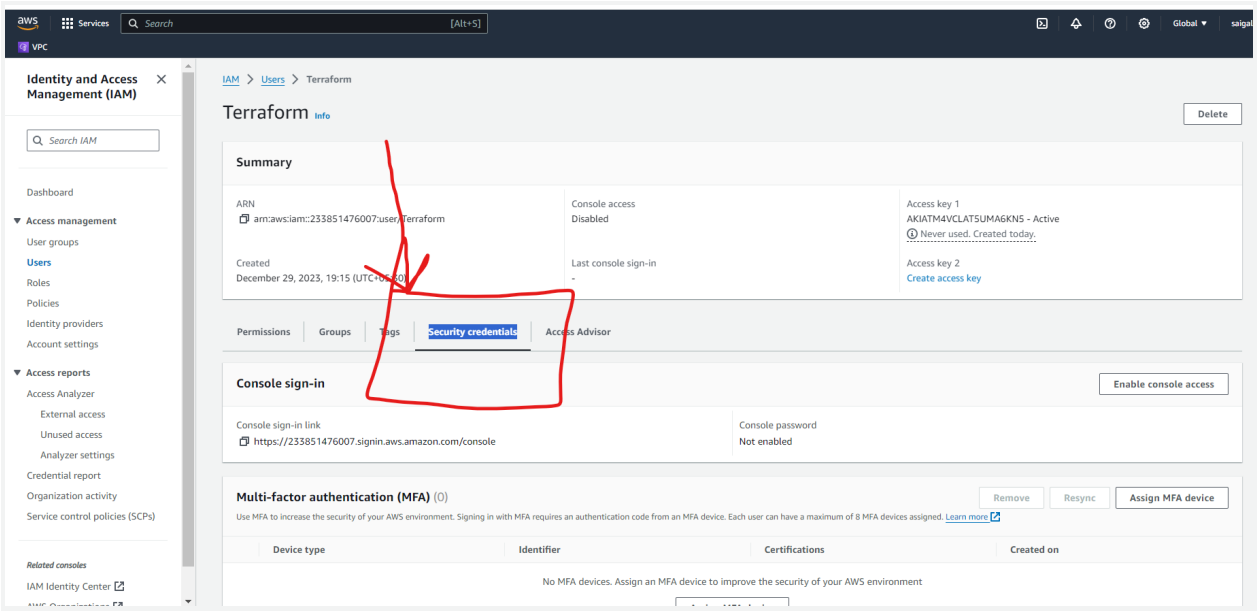


5. Create user by clicking ‘create user’

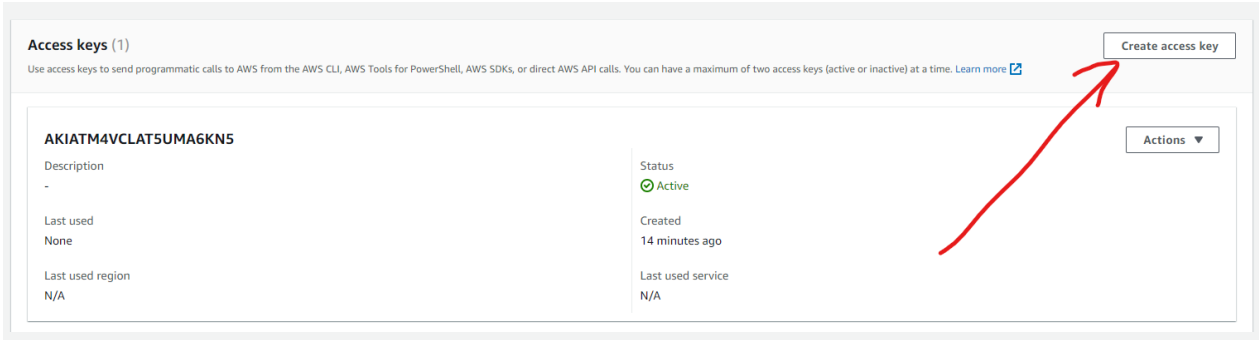


User created

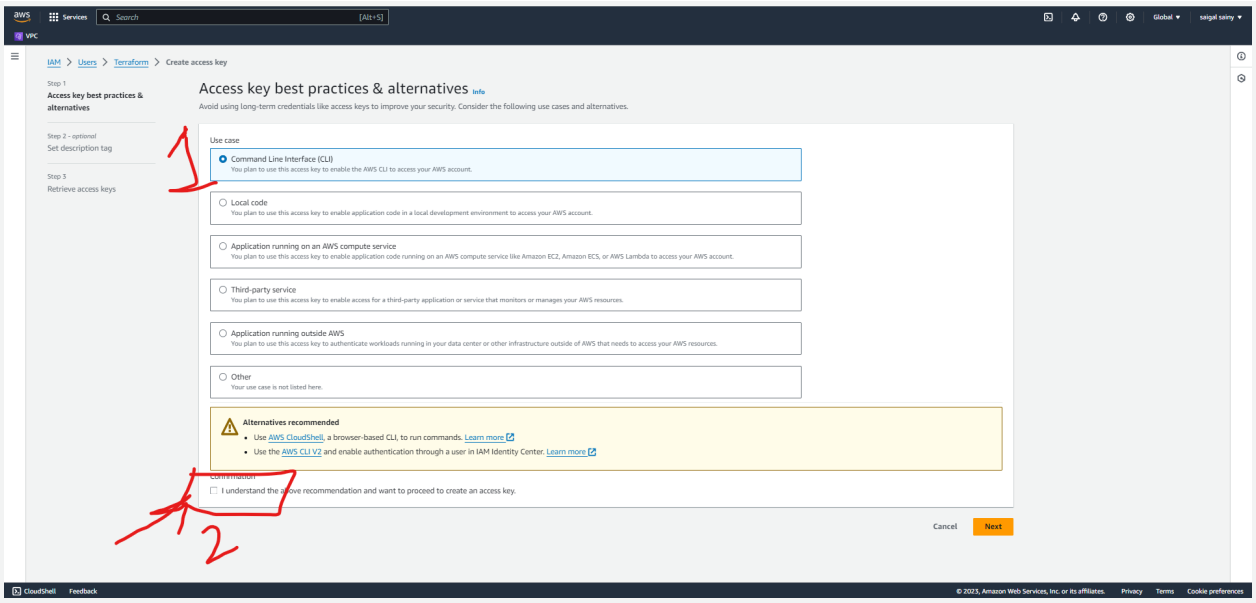
6. Create secret key for user
   →click on user and go into **Security credentials**

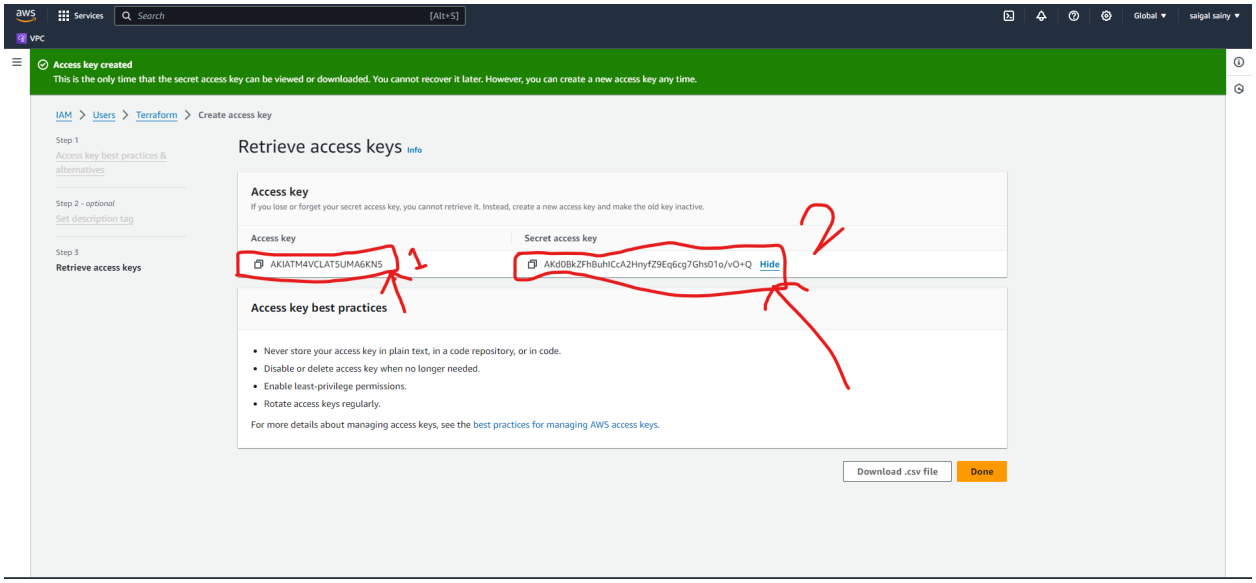


7. Scroll down and click ‘create access key’



8. Select first option and check documentation then click ‘save’





‘Access key’ and ‘secret key’ created

Run commands for create environment

1. Run this command for install plugins for create envirnment
   →terraform init

```
[root@ip-172-31-44-69 ~]# terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.31.0...
- Installed hashicorp/aws v5.31.0 (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.
[root@ip-172-31-44-69 ~]#
```

2. Run this command is used to see the changes that will take place on the infrastructure.
   →terraform plan

```
[root@ip-172-31-44-69 ~]# 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.ec2_instance will be created
  + resource "aws_instance" "ec2_instance" {
      + ami                                  = "ami-03f4878755434977f"
      + 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_lifecycle                   = (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                             = "terraform"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
```

3. Run this command for apply script and create infrastructure
→terraform apply

```
[root@ip-172-31-44-69 ~]# 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.ec2_instance will be created
  + resource "aws_instance" "ec2_instance" {
      + ami                                  = "ami-03f4878755434977f"
      + 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_lifecycle                   = (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                             = "terraform"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
```

```
      + ipv6_addresses                       = (known after apply)
      + key_name                             = "terraform"
      + 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
      + spot_instance_request_id             = (known after apply)
      + 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)
    }

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:
```

Type 'yes'

```
aws   Services   Search                                   [Alt+S]
VPC
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + 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)
    }

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.ec2_instance: Creating...
aws_instance.ec2_instance: Still creating... [10s elapsed]
aws_instance.ec2_instance: Still creating... [20s elapsed]
aws_instance.ec2_instance: Still creating... [30s elapsed]
aws_instance.ec2_instance: Creation complete after 32s [id=i-0f0aca040d91a8dff]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
[root@ip-172-31-44-69 ~]#
```

EC2 created

Instances (1/2) Info | Connect | Instance state | Actions | Launch instances

Find Instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instance type	Status check	Alarm sta
	i-0f0aca040d91a8dff	Running	t2.micro	2/2 checks passed	No alarms
terraform	i-0bd23769baa2e5a73	Running	t2.micro	2/2 checks passed	No alarms

EC2-Deletion

Run below command for delete EC2 via terraform script

→terraform destroy

```
[root@ip-172-31-44-69 ~]# terraform destroy
aws_instance.ec2_instance: Refreshing state... [id=i-0f0aca040d91a8dff]

Terraform used the selected providers to generate the following execution plan. Resource actions are
 symbols:
  - destroy

Terraform will perform the following actions:

  # aws_instance.ec2_instance will be destroyed
  - resource "aws_instance" "ec2_instance" {
      - ami                                  = "ami-0a0f1259dd1c90938" -> null
      - arn                                  = "arn:aws:ec2:ap-south-1:233851476007:instance/i-0f0aca
      - associate_public_ip_address          = true -> null
      - availability_zone                    = "ap-south-1b" -> null
      - cpu_core_count                       = 1 -> null
      - cpu_threads_per_core                 = 1 -> null
      - disable_api_stop                     = false -> null
      - disable_api_termination              = false -> null
      - ebs_optimized                        = false -> null
      - get_password_data                    = false -> null
      - hibernation                          = false -> null
      - id                                   = "i-0f0aca040d91a8dff" -> null
      - instance_initiated_shutdown_behavior = "stop" -> null
      - instance_state                       = "running" -> null
      - instance_type                        = "t2.micro" -> null
```

Type 'yes'

```
          - device_name           = "/dev/xvda" -> null
          - encrypted             = false -> null
          - iops                  = 3000 -> null
          - tags                  = {} -> null
          - throughput            = 125 -> null
          - volume_id             = "vol-0814eec0040f106b8" -> null
          - volume_size           = 8 -> null
          - volume_type           = "gp3" -> 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

aws_instance.ec2_instance: Destroying... [id=i-0f0aca040d91a8dff]
aws_instance.ec2_instance: Still destroying... [id=i-0f0aca040d91a8dff, 10s elapsed]
aws_instance.ec2_instance: Still destroying... [id=i-0f0aca040d91a8dff, 20s elapsed]
aws_instance.ec2_instance: Destruction complete after 29s

Destroy complete! Resources: 1 destroyed.
[root@ip-172-31-44-69 ~]#
```