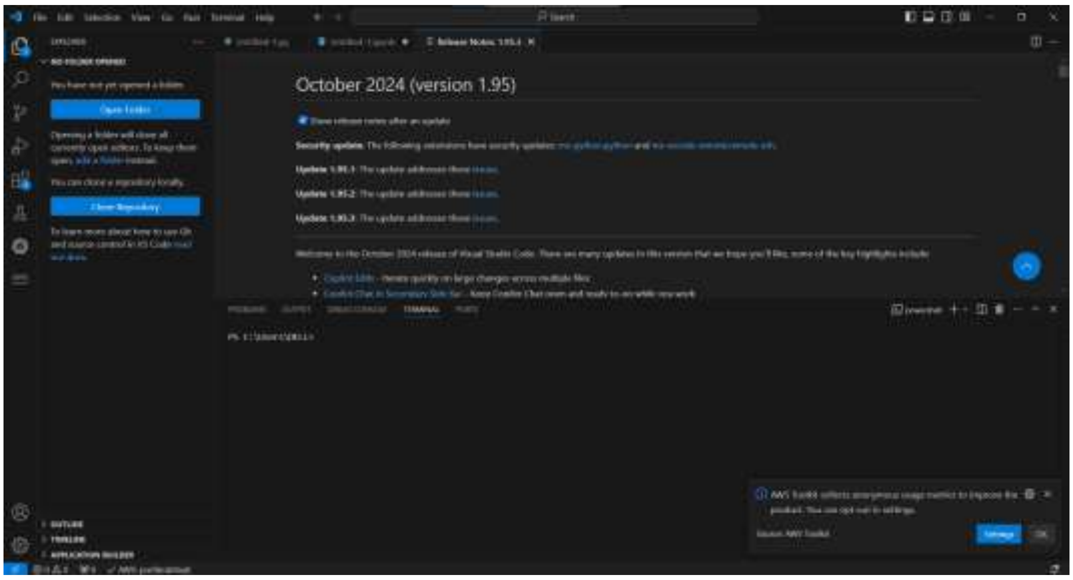
Terraform Practical

Day 1

Sakshi Shirure

1)Download VS code.



October 2024 (version 1.95)

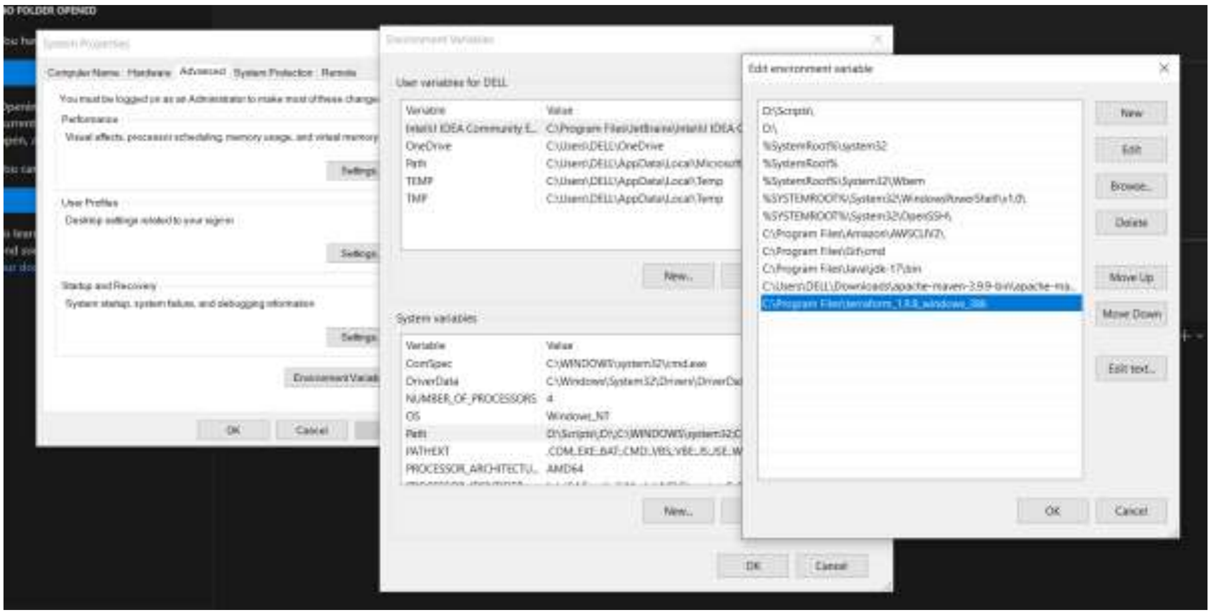
Security update: The following extensions have security updates:

Update 1.95.1: The update addresses these issues.

Update 1.95.2: The update addresses these issues.

Update 1.95.3: The update addresses these issues.

2) Download Terraform and Set the env. Settings.



User variables for DELL

Variable	Value
IntelliJ IDEA Community E...	C:\Program Files\JetBrains\IntelliJ IDEA C
OneDrive	C:\Users\DELL\OneDrive
Path	C:\Users\DELL\AppData\Local\Microsoft
TEMP	C:\Users\DELL\AppData\Local\Temp
TMP	C:\Users\DELL\AppData\Local\Temp

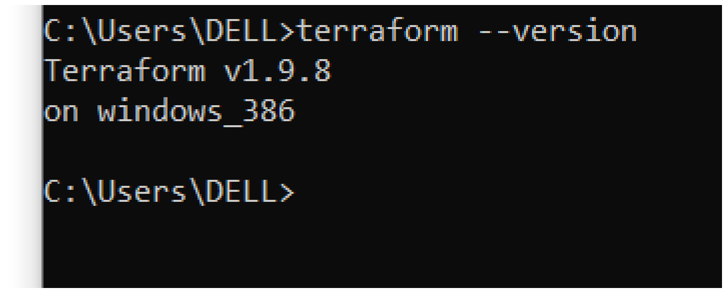
System variables

Variable	Value
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverDa
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	D:\Scripts\;D:\;C:\WINDOWS\system32;C
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.W
PROCESSOR_ARCHITECTU...	AMD64

Edit environment variable

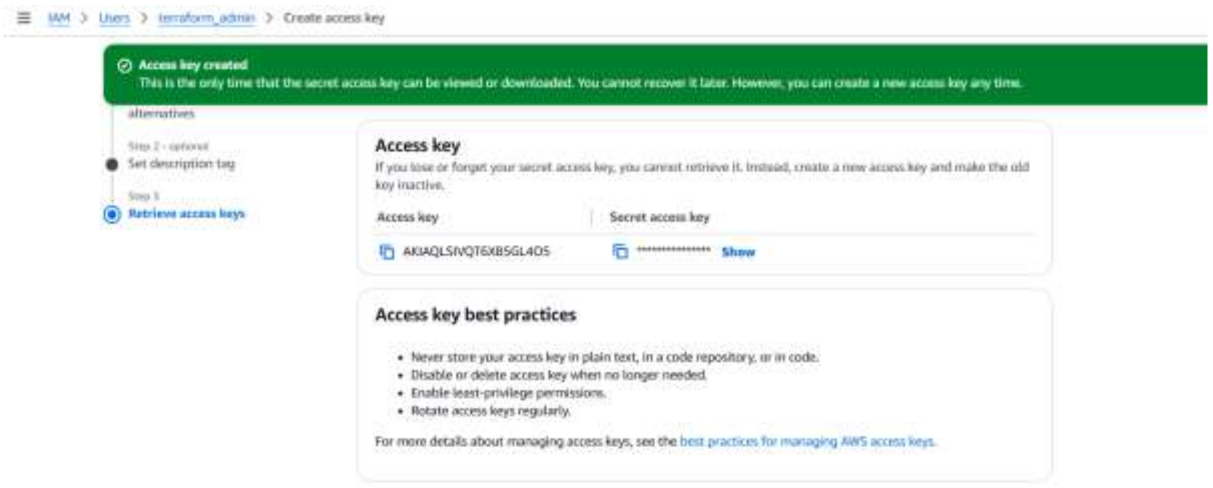
```
D:\Scripts\
D:\
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files\Amazon\AWSCLIV2\
C:\Program Files\Git\cmd
C:\Program Files\Java\jdk-17\bin
C:\Users\DELL\Downloads\apache-maven-3.9.9-bin\apache-ma...
C:\Program Files\terraform_1.9.8_windows_386
```

3) Check Terraform on windows.



```
C:\Users\DELL>terraform --version
Terraform v1.9.8
on windows_386

C:\Users\DELL>
```

4) Create IAM user with admin policy & access key secret access key.



Access key created
This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

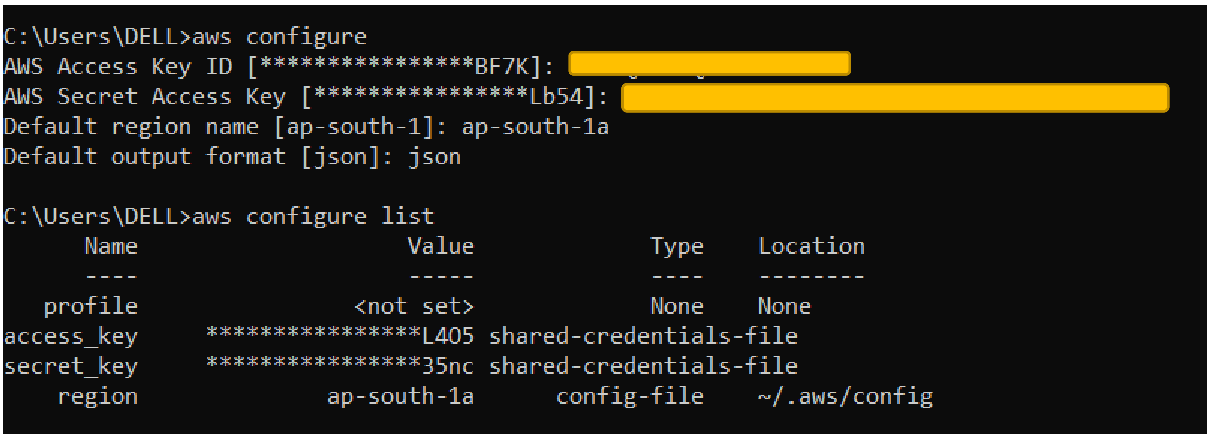
Access key
If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
AKIAQLSIVQT6XBSGL4O5	*************** Show

Access key best practices

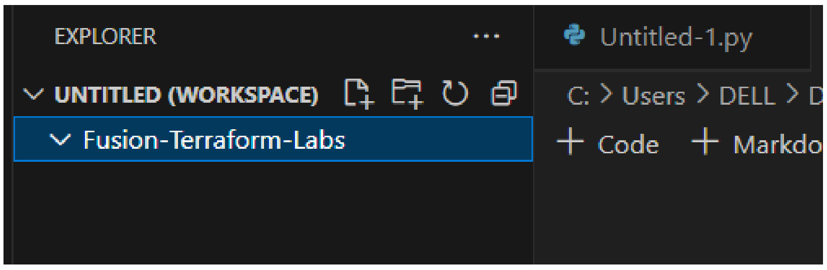
- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the best practices for managing AWS access keys.



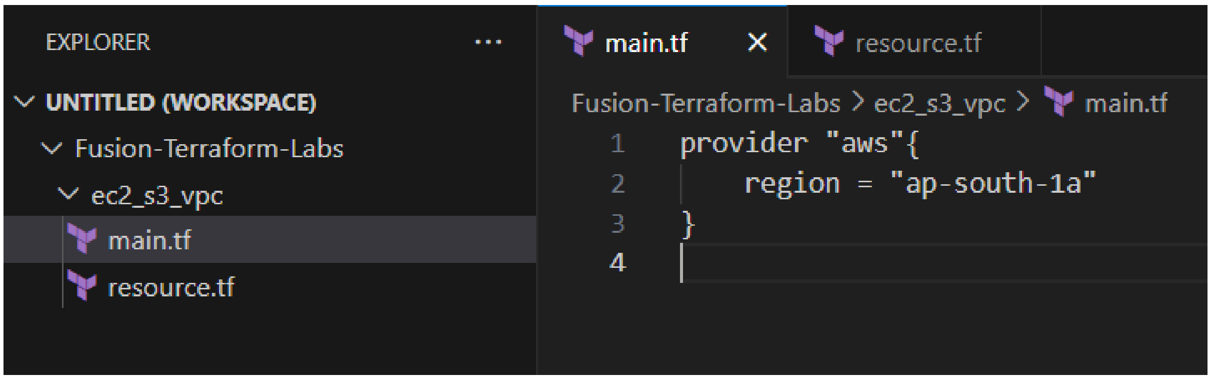
```
C:\Users\DELL>aws configure
AWS Access Key ID [****************BF7K]: [redacted]
AWS Secret Access Key [****************Lb54]: [redacted]
Default region name [ap-south-1]: ap-south-1a
Default output format [json]: json

C:\Users\DELL>aws configure list
      Name                    Value             Type    Location
      ----                    -----             ----    --------
   profile                <not set>             None    None
access_key     ****************L405 shared-credentials-file
secret_key     ****************35nc shared-credentials-file
    region              ap-south-1a      config-file    ~/.aws/config
```

5) Create a new folder as a “Fusion-Terraform-Labs” & add folder in VS code editor.



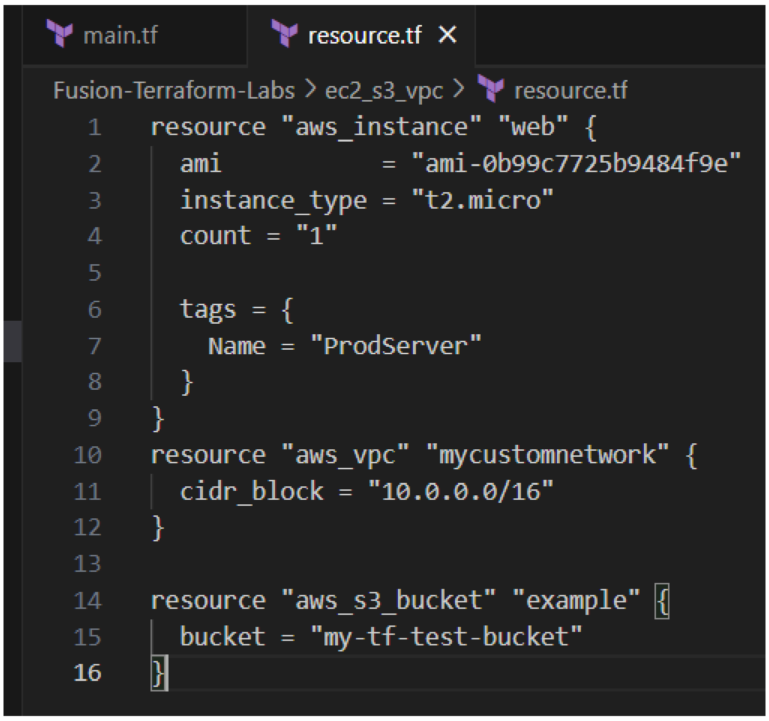
6) Create main.tf & resource.tf file.

Main.tf



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

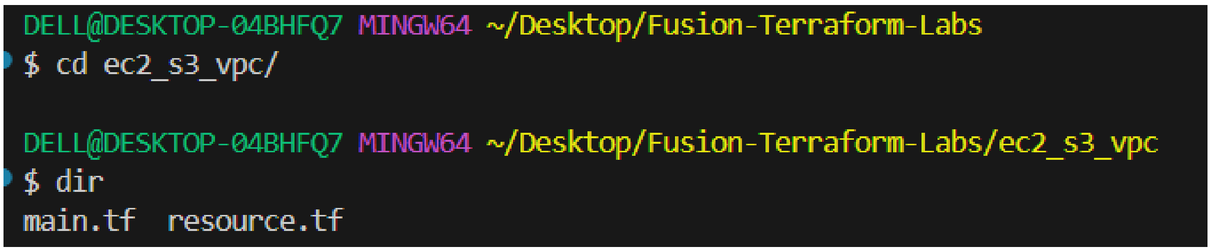
Resource.tf



```
resource "aws_instance" "web" {
  ami           = "ami-0b99c7725b9484f9e"
  instance_type = "t2.micro"
  count = "1"

  tags = {
    Name = "ProdServer"
  }
}
resource "aws_vpc" "mycustomnetwork" {
  cidr_block = "10.0.0.0/16"
}

resource "aws_s3_bucket" "example" {
  bucket = "my-tf-test-bucket"
}
```

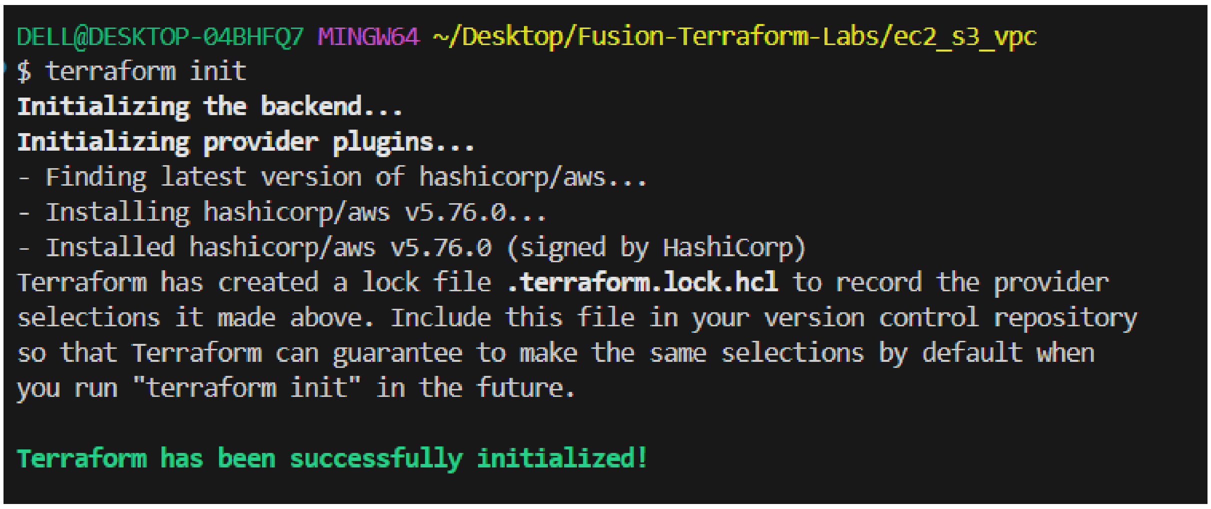
7)Run commands

1. Terraform init = Initializes the Terraform environment and downloads necessary plugins.
2. terraform validate = Validates your Terraform configuration files
3. Terraform plan = Creates an execution plan to show what actions Terraform will take to create, modify, or destroy infrastructure based on your configuration.
4. terraform apply = Applies the changes to create or update the infrastructure
5. terraform fmt = Formats configuration files for consistency and readability.
6. terraform destroy = Destroys all resources created by Terraform



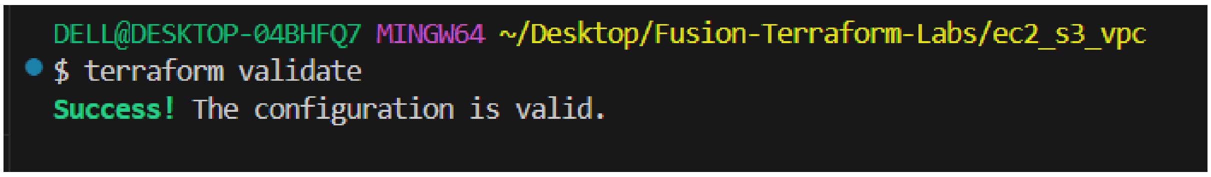
```
DELL@DESKTOP-04BHFQ7 MINGW64 ~/Desktop/Fusion-Terraform-Labs
$ cd ec2_s3_vpc/

DELL@DESKTOP-04BHFQ7 MINGW64 ~/Desktop/Fusion-Terraform-Labs/ec2_s3_vpc
$ dir
main.tf  resource.tf
```

#terraform init



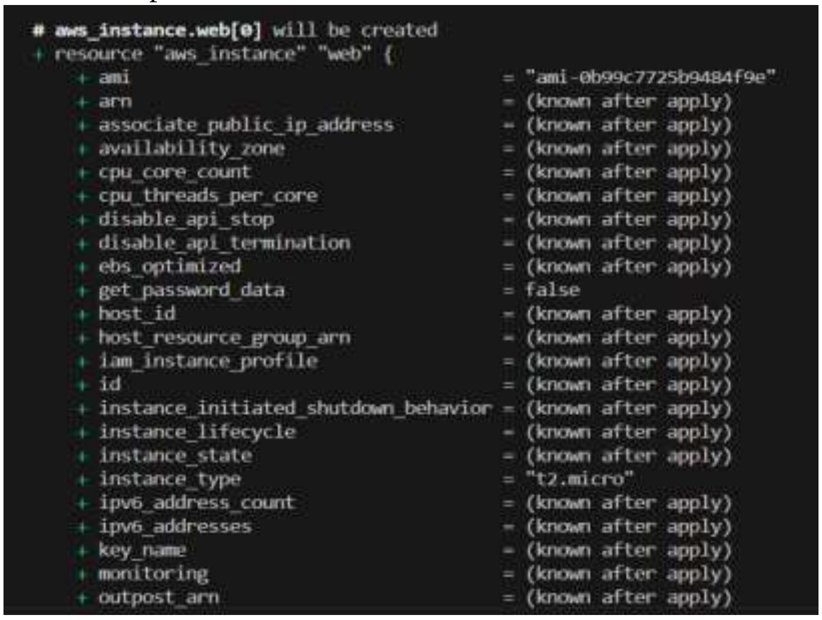
```
DELL@DESKTOP-04BHFQ7 MINGW64 ~/Desktop/Fusion-Terraform-Labs/ec2_s3_vpc
$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.76.0...
- Installed hashicorp/aws v5.76.0 (signed by HashiCorp)
Terraform 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!
```

#terraform validate

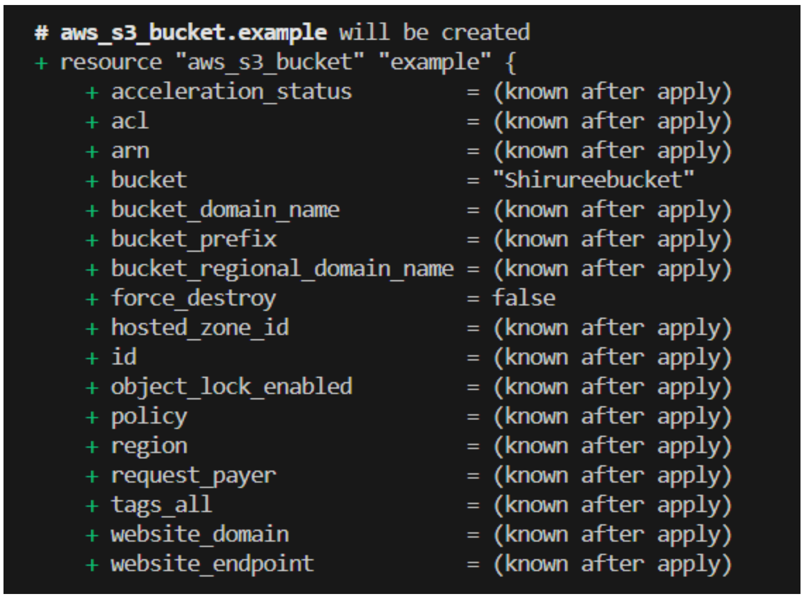


```
DELL@DESKTOP-04BHFQ7 MINGW64 ~/Desktop/Fusion-Terraform-Labs/ec2_s3_vpc
$ terraform validate
Success! The configuration is valid.
```

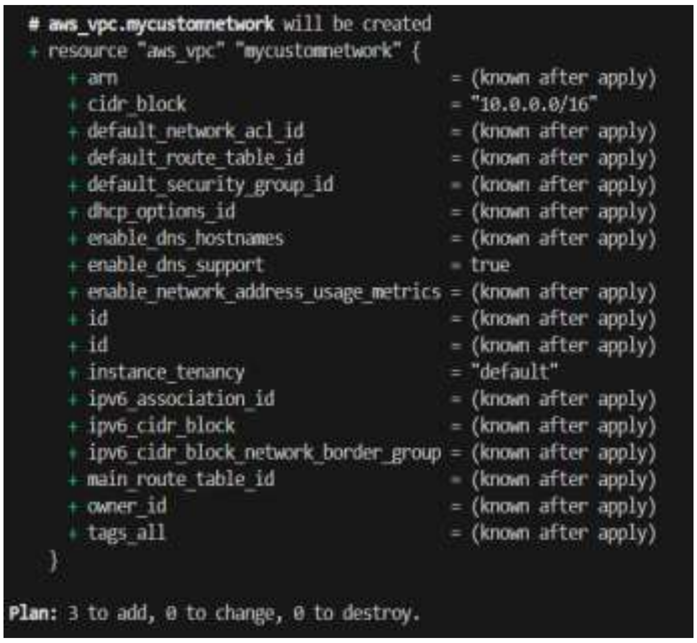
#terraform plan



```
# aws_instance.web[0] will be created
+ resource "aws_instance" "web" {
    + ami                                  = "ami-0b99c7725b9484f9e"
    + 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                             = (known after apply)
    + monitoring                           = (known after apply)
    + outpost_arn                          = (known after apply)
```

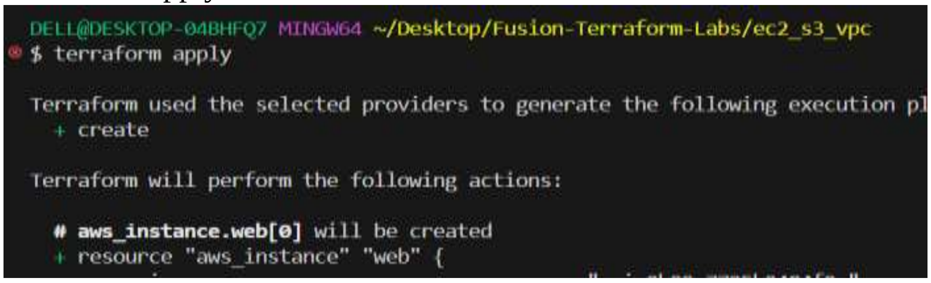


```
# aws_s3_bucket.example will be created
+ resource "aws_s3_bucket" "example" {
    + acceleration_status         = (known after apply)
    + acl                         = (known after apply)
    + arn                         = (known after apply)
    + bucket                      = "Shirureebucket"
    + bucket_domain_name          = (known after apply)
    + bucket_prefix               = (known after apply)
    + bucket_regional_domain_name = (known after apply)
    + force_destroy               = false
    + hosted_zone_id              = (known after apply)
    + id                          = (known after apply)
    + object_lock_enabled         = (known after apply)
    + policy                      = (known after apply)
    + region                      = (known after apply)
    + request_payer               = (known after apply)
    + tags_all                    = (known after apply)
    + website_domain              = (known after apply)
    + website_endpoint            = (known after apply)
```

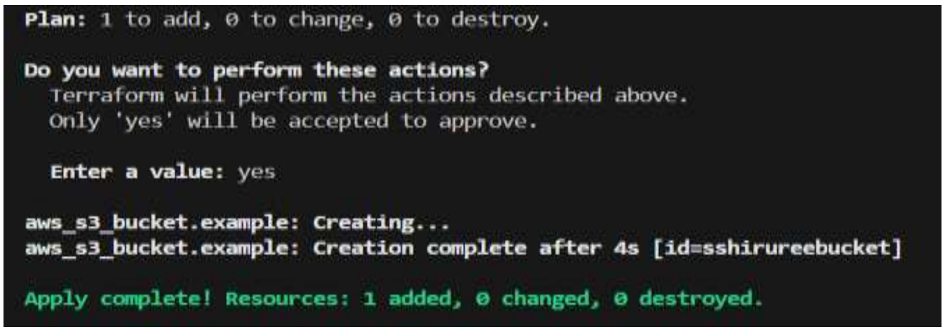


```
  # aws_vpc.mycustomnetwork will be created
  + resource "aws_vpc" "mycustomnetwork" {
      + arn                                  = (known after apply)
      + cidr_block                           = "10.0.0.0/16"
      + default_network_acl_id               = (known after apply)
      + default_route_table_id               = (known after apply)
      + default_security_group_id            = (known after apply)
      + dhcp_options_id                      = (known after apply)
      + enable_dns_hostnames                 = (known after apply)
      + enable_dns_support                   = true
      + enable_network_address_usage_metrics = (known after apply)
      + id                                   = (known after apply)
      + id                                   = (known after apply)
      + instance_tenancy                     = "default"
      + ipv6_association_id                  = (known after apply)
      + ipv6_cidr_block                      = (known after apply)
      + ipv6_cidr_block_network_border_group = (known after apply)
      + main_route_table_id                  = (known after apply)
      + owner_id                             = (known after apply)
      + tags_all                             = (known after apply)
    }

Plan: 3 to add, 0 to change, 0 to destroy.
```

#terraform apply



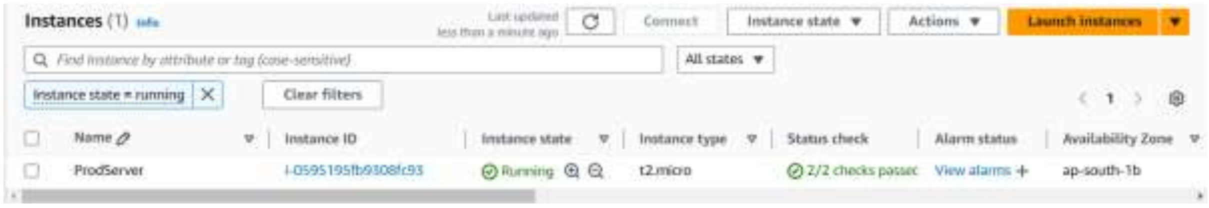
```
  DELL@DESKTOP-04BHFQ7 MINGW64 ~/Desktop/Fusion-Terraform-Labs/ec2_s3_vpc
$ terraform apply

  Terraform used the selected providers to generate the following execution pl
    + create

  Terraform will perform the following actions:

    # aws_instance.web[0] will be created
    + resource "aws_instance" "web" {
```



```
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_s3_bucket.example: Creating...
aws_s3_bucket.example: Creation complete after 4s [id=sshirureebucket]

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

8) See output on AWS console.

Ec2



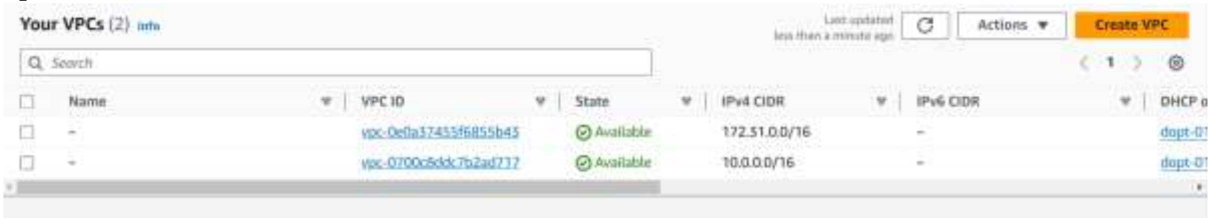
Instances (1) Info

Last updated less than a minute ago

Instance state = running

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
☐	ProdServer	i-0595195fb9308fc93	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1b

vpc

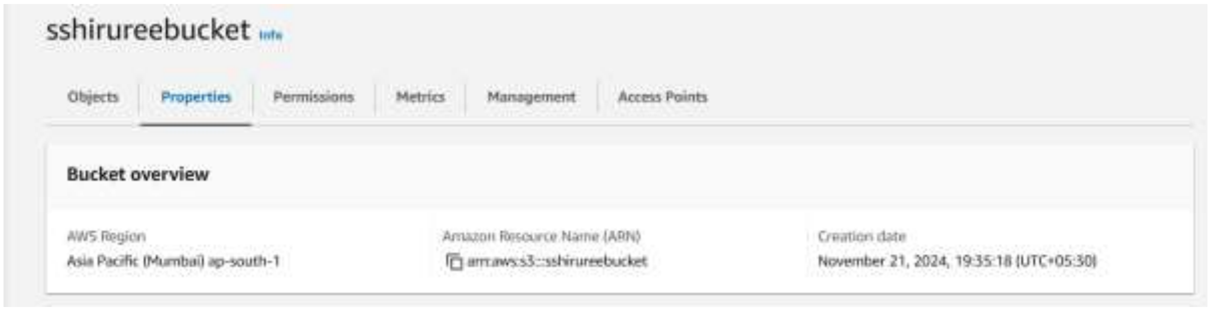


Your VPCs (2) Info

Last updated less than a minute ago

	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP o
☐	-	vpc-0e0a37455f6855b43	✓ Available	172.31.0.0/16	-	dopt-01
☐	-	vpc-0700c8ddc7b2ad717	✓ Available	10.0.0.0/16	-	dopt-01

s3 bucket

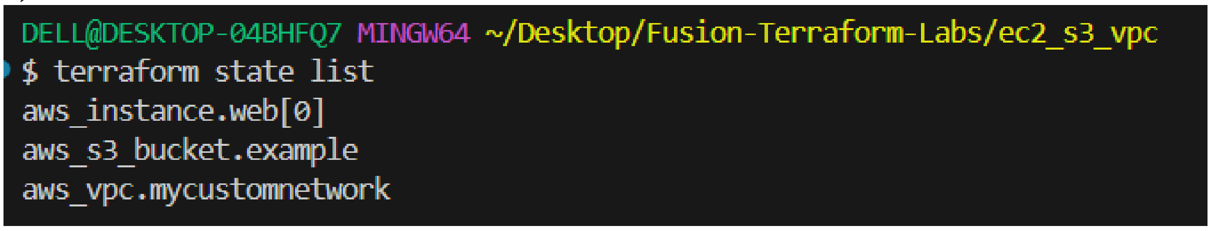


sshirureebucket Info

Bucket overview

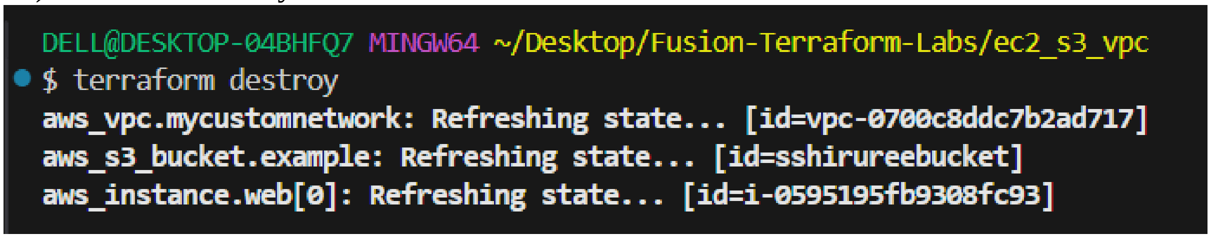
AWS Region	Amazon Resource Name (ARN)	Creation date
Asia Pacific (Mumbai) ap-south-1	arn:aws:s3:::sshirureebucket	November 21, 2024, 19:35:18 (UTC+05:30)

9) #terraform state list

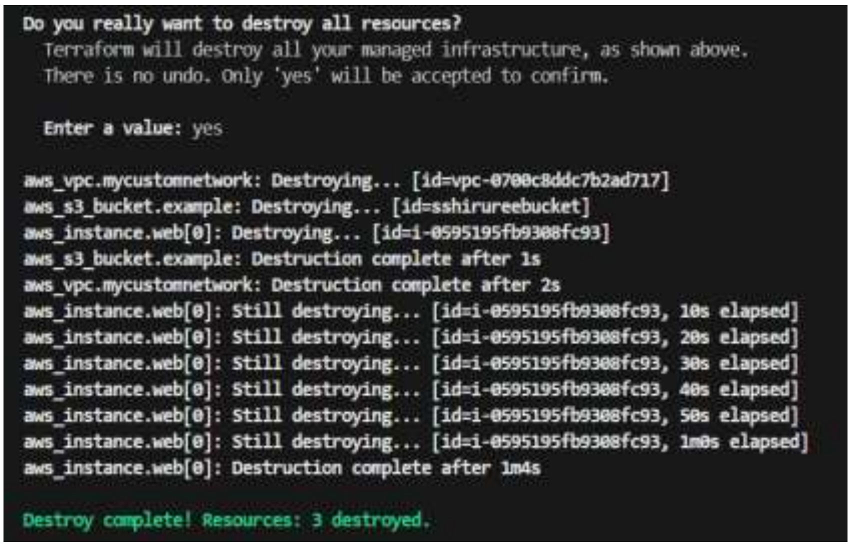


```
DELL@DESKTOP-04BHFQ7 MINGW64 ~/Desktop/Fusion-Terraform-Labs/ec2_s3_vpc
$ terraform state list
aws_instance.web[0]
aws_s3_bucket.example
aws_vpc.mycustomnetwork
```

10) #terraform destroy

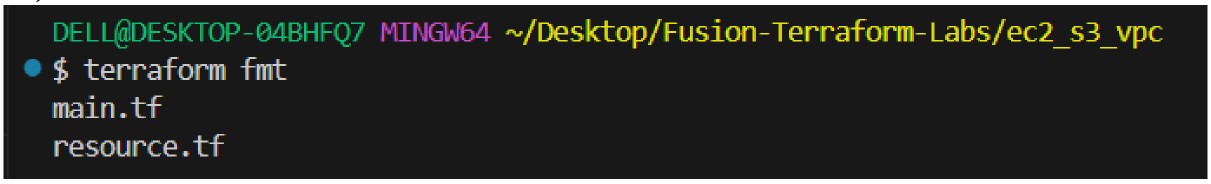


```
DELL@DESKTOP-04BHFQ7 MINGW64 ~/Desktop/Fusion-Terraform-Labs/ec2_s3_vpc
$ terraform destroy
aws_vpc.mycustomnetwork: Refreshing state... [id=vpc-0700c8ddc7b2ad717]
aws_s3_bucket.example: Refreshing state... [id=sshirureebucket]
aws_instance.web[0]: Refreshing state... [id=i-0595195fb9308fc93]
```



```
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_vpc.mycustomnetwork: Destroying... [id=vpc-0700c8ddc7b2ad717]
aws_s3_bucket.example: Destroying... [id=sshirureebucket]
aws_instance.web[0]: Destroying... [id=i-0595195fb9308fc93]
aws_s3_bucket.example: Destruction complete after 1s
aws_vpc.mycustomnetwork: Destruction complete after 2s
aws_instance.web[0]: Still destroying... [id=i-0595195fb9308fc93, 10s elapsed]
aws_instance.web[0]: Still destroying... [id=i-0595195fb9308fc93, 20s elapsed]
aws_instance.web[0]: Still destroying... [id=i-0595195fb9308fc93, 30s elapsed]
aws_instance.web[0]: Still destroying... [id=i-0595195fb9308fc93, 40s elapsed]
aws_instance.web[0]: Still destroying... [id=i-0595195fb9308fc93, 50s elapsed]
aws_instance.web[0]: Still destroying... [id=i-0595195fb9308fc93, 1m0s elapsed]
aws_instance.web[0]: Destruction complete after 1m4s

Destroy complete! Resources: 3 destroyed.
```

11) #terraform fmt



```
DELL@DESKTOP-04BHFQ7 MINGW64 ~/Desktop/Fusion-Terraform-Labs/ec2_s3_vpc
$ terraform fmt
main.tf
resource.tf
```

Thank You