Terraform 05&06

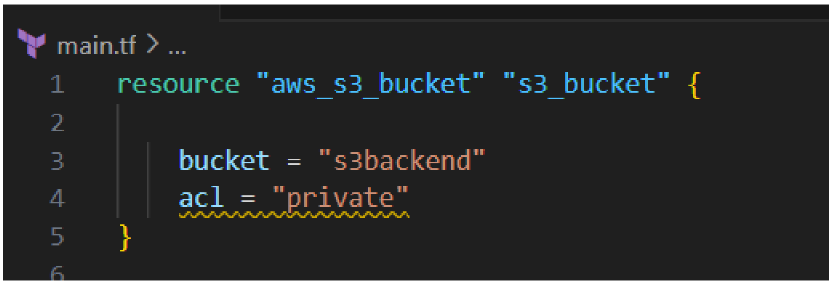
1) Watch terraform-05 video.

--completed

2) Execute the script shown in video.

Create AWS s3 using terraform:

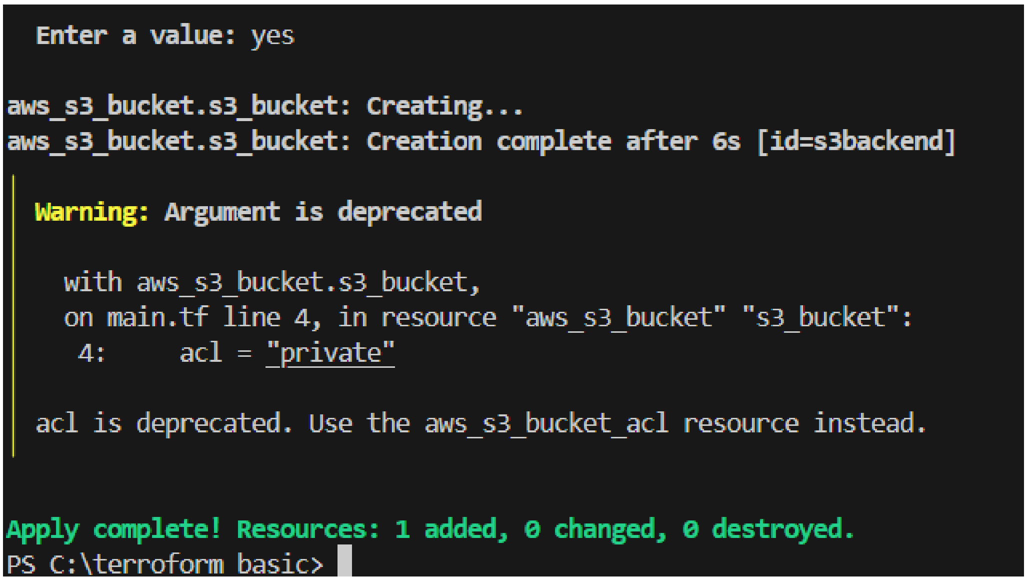
--main.tf template



main.tf > ...

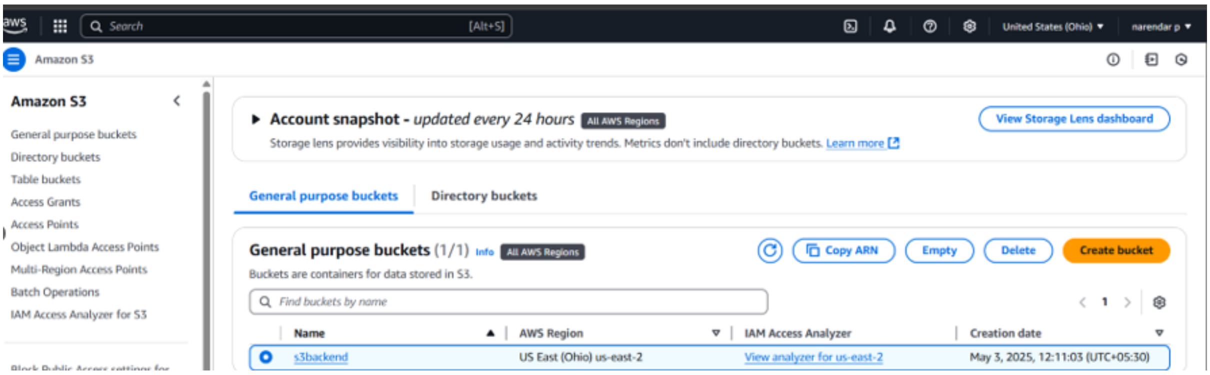
```
resource "aws_s3_bucket" "s3_bucket" {

    bucket = "s3backend"
    acl = "private"
}
```

--execution



```
  Enter a value: yes

aws_s3_bucket.s3_bucket: Creating...
aws_s3_bucket.s3_bucket: Creation complete after 6s [id=s3backend]

  Warning: Argument is deprecated

    with aws_s3_bucket.s3_bucket,
    on main.tf line 4, in resource "aws_s3_bucket" "s3_bucket":
     4:     acl = "private"

  acl is deprecated. Use the aws_s3_bucket_acl resource instead.


Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\terroform basic>
```

--check aws s3 now



Account snapshot - *updated every 24 hours* All AWS Regions

Storage lens provides visibility into storage usage and activity trends. Metrics don't include directory buckets. Learn more

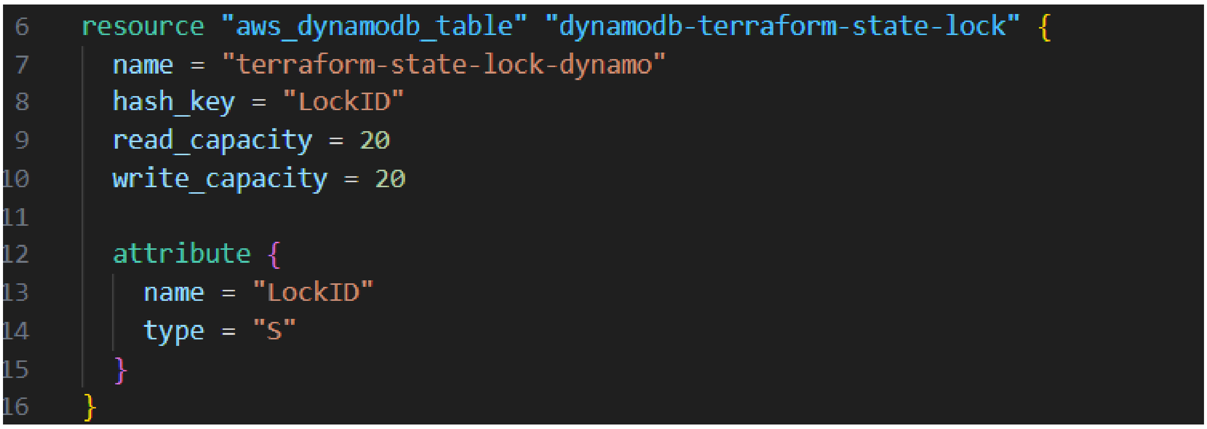
General purpose buckets (1/1) Info All AWS Regions

Buckets are containers for data stored in S3.

	Name	AWS Region	IAM Access Analyzer	Creation date
●	s3backend	US East (Ohio) us-east-2	View analyzer for us-east-2	May 3, 2025, 12:11:03 (UTC+05:30)

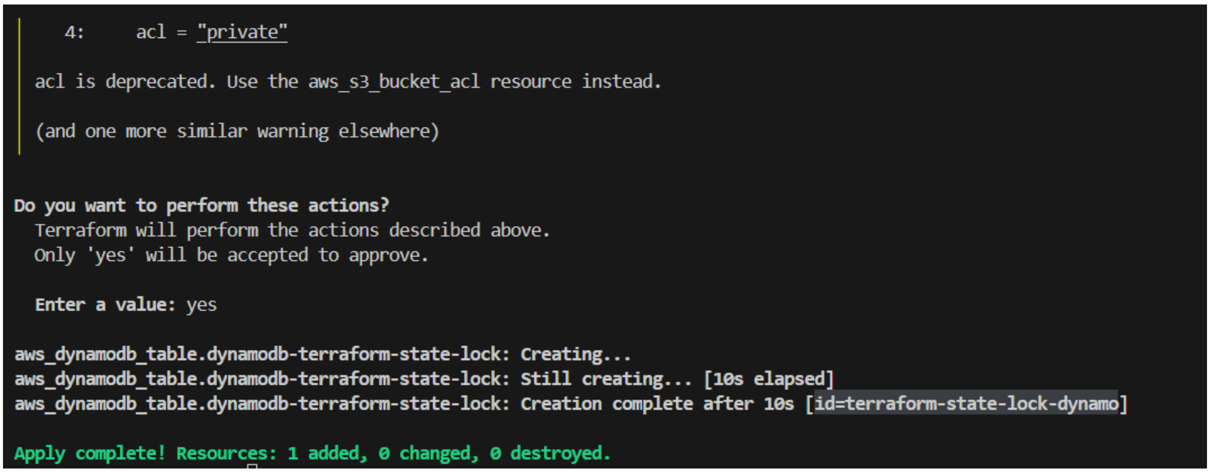
Create AWS dynamo db using terraform:

--template



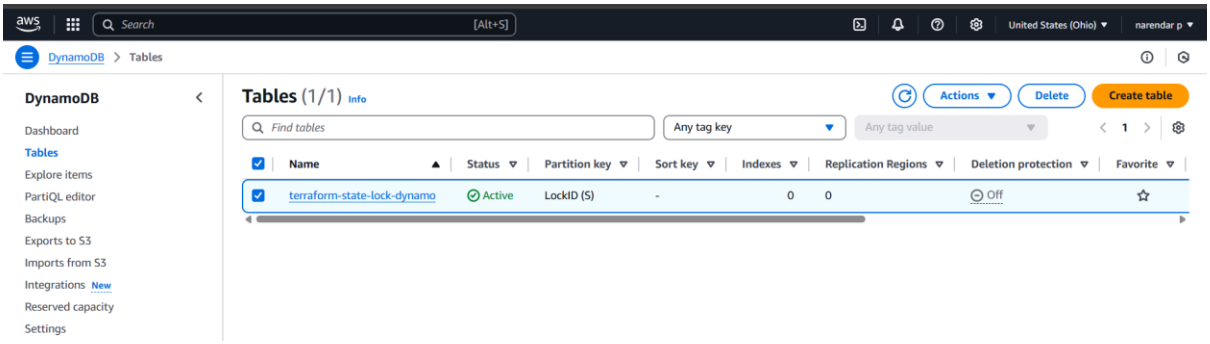
```
resource "aws_dynamodb_table" "dynamodb-terraform-state-lock" {
  name = "terraform-state-lock-dynamo"
  hash_key = "LockID"
  read_capacity = 20
  write_capacity = 20

  attribute {
    name = "LockID"
    type = "S"
  }
}
```

--execution



```
│   4:     acl = "private"
│
│ acl is deprecated. Use the aws_s3_bucket_acl resource instead.
│
│ (and one more similar warning elsewhere)


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_dynamodb_table.dynamodb-terraform-state-lock: Creating...
aws_dynamodb_table.dynamodb-terraform-state-lock: Still creating... [10s elapsed]
aws_dynamodb_table.dynamodb-terraform-state-lock: Creation complete after 10s [id=terraform-state-lock-dynamo]

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

--check in AWS Dynamo DB

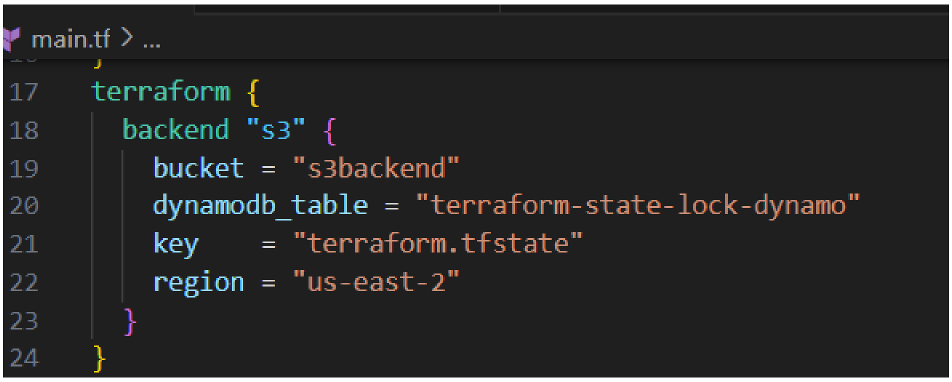


Tables (1/1) Info

☑	Name	Status	Partition key	Sort key	Indexes	Replication Regions	Deletion protection	Favorite
☑	terraform-state-lock-dynamo	✓ Active	LockID (S)	-	0	0	⊖ Off	☆

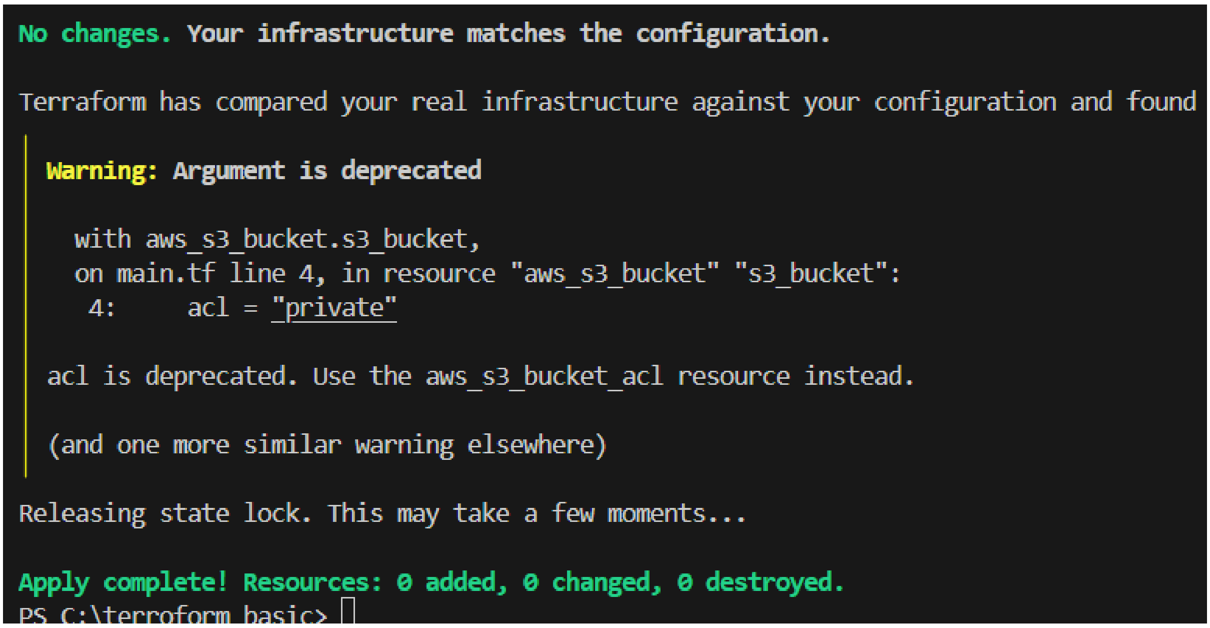
S3 as backend for terraform.tfstate file:

----template



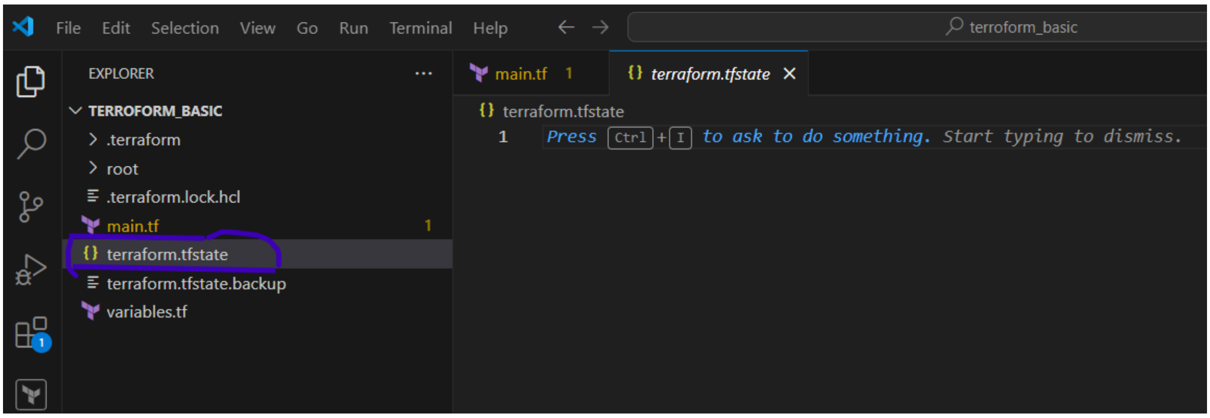
```
terraform {
  backend "s3" {
    bucket = "s3backend"
    dynamodb_table = "terraform-state-lock-dynamo"
    key     = "terraform.tfstate"
    region = "us-east-2"
  }
}
```

--exection

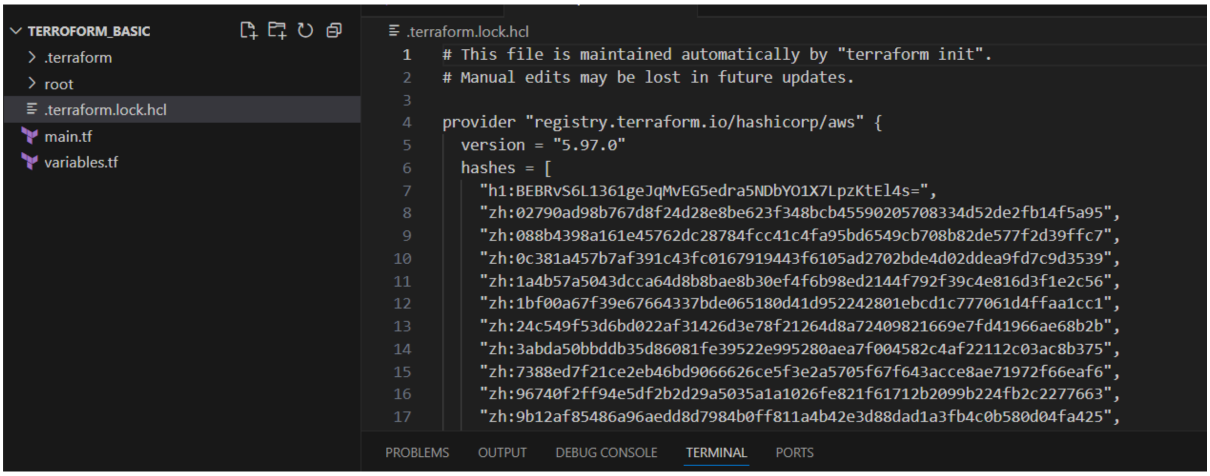


```
No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and found

  Warning: Argument is deprecated

    with aws_s3_bucket.s3_bucket,
    on main.tf line 4, in resource "aws_s3_bucket" "s3_bucket":
     4:     acl = "private"

  acl is deprecated. Use the aws_s3_bucket_acl resource instead.

  (and one more similar warning elsewhere)

Releasing state lock. This may take a few moments...

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
PS C:\terroform_basic> 
```

Check state file its locked



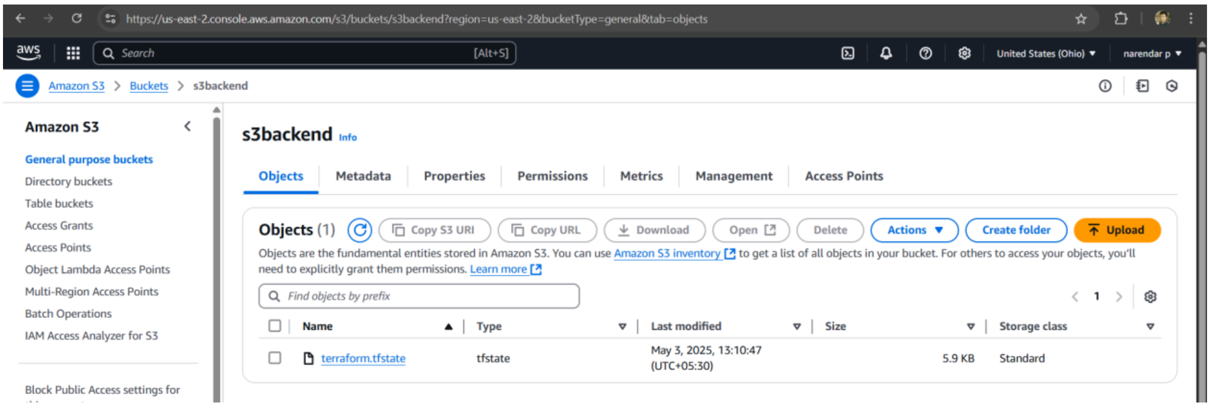
```
1  Press Ctrl+I to ask to do something. Start typing to dismiss.
```



```
# This file is maintained automatically by "terraform init".
# Manual edits may be lost in future updates.

provider "registry.terraform.io/hashicorp/aws" {
  version = "5.97.0"
  hashes = [
    "h1:BEBRvS6L1361geJqMvEG5edra5NDbYO1X7LpzKtEl4s=",
    "zh:02790ad98b767d8f24d28e8be623f348bcb45590205708334d52de2fb14f5a95",
    "zh:088b4398a161e45762dc28784fcc41c4fa95bd6549cb708b82de577f2d39ffc7",
    "zh:0c381a457b7af391c43fc0167919443f6105ad2702bde4d02ddea9fd7c9d3539",
    "zh:1a4b57a5043dcca64d8b8bae8b30ef4f6b98ed2144f792f39c4e816d3f1e2c56",
    "zh:1bf00a67f39e67664337bde065180d41d952242801ebcd1c777061d4ffaa1cc1",
    "zh:24c549f53d6bd022af31426d3e78f21264d8a72409821669e7fd41966ae68b2b",
    "zh:3abda50bbddb35d86081fe39522e995280aea7f004582c4af22112c03ac8b375",
    "zh:7388ed7f21ce2eb46bd9066626ce5f3e2a5705f67f643acce8ae71972f66eaf6",
    "zh:96740f2ff94e5df2b2d29a5035a1a1026fe821f61712b2099b224fb2c2277663",
    "zh:9b12af85486a96aedd8d7984b0ff811a4b42e3d88dad1a3fb4c0b580d04fa425",
```

Check aws account

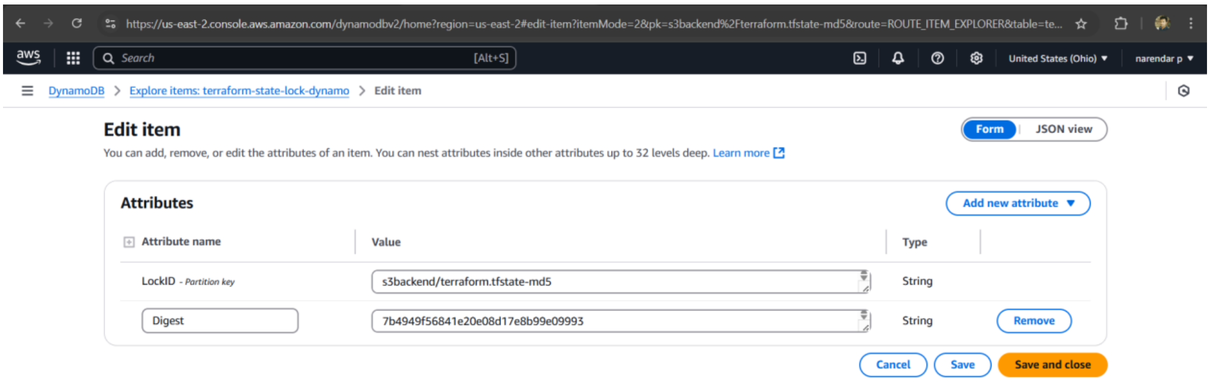
S3 terraform.tfstate file got created



Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name ▲	Type	Last modified	Size	Storage class
□	terraform.tfstate	tfstate	May 3, 2025, 13:10:47 (UTC+05:30)	5.9 KB	Standard

Dynabo db table created with lock file


https://us-east-2.console.aws.amazon.com/dynamodbv2/home?region=us-east-2#edit-item?itemMode=2&pk=s3backend%2Fterraform.tfstate-md5&route=ROUTE_ITEM_EXPLORER&table=te...
aws
Search
[Alt+S]
United States (Ohio) ▼
narendar p ▼
DynamoDB > Explore items: terraform-state-lock-dynamo > Edit item
Form
JSON view
Add new attribute ▼
Remove
Cancel
Save
Save and close

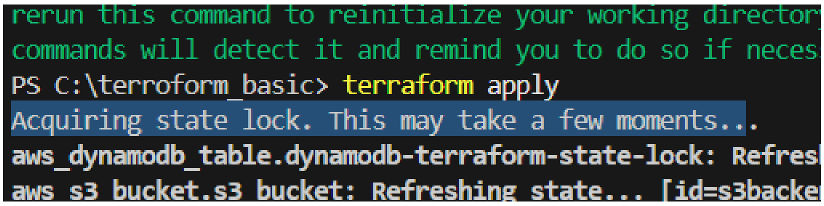
Edit item

You can add, remove, or edit the attributes of an item. You can nest attributes inside other attributes up to 32 levels deep. Learn more

Attributes

Attribute name	Value	Type
LockID - *Partition key*	s3backend/terraform.tfstate-md5	String
Digest	7b4949f56841e20e08d17e8b99e09993	String

Test if someone wants to perform any action through an error or takes time one complete then others one start

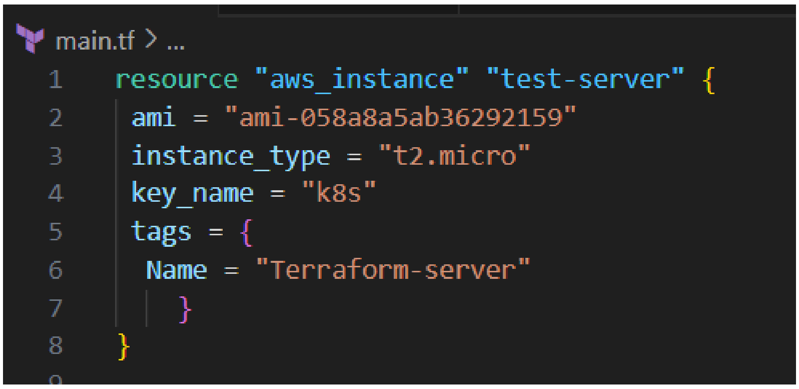


```
rerun this command to reinitialize your working director
commands will detect it and remind you to do so if neces
PS C:\terroform_basic> terraform apply
Acquiring state lock. This may take a few moments...
aws_dynamodb_table.dynamodb-terraform-state-lock: Refres
aws_s3_bucket.s3_bucket: Refreshing state... [id=s3backe
```

Terraform Provisioners:

--provision one ec2 instance using terraform template

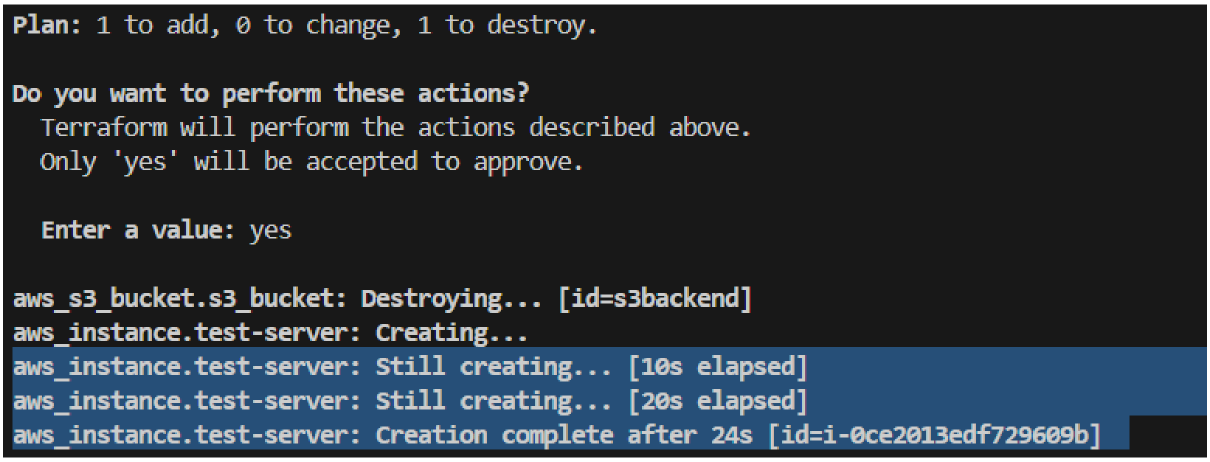
-template



main.tf > ...

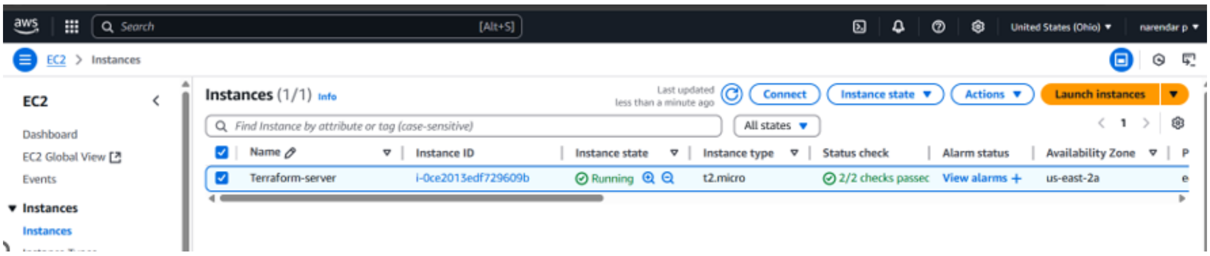
```
resource "aws_instance" "test-server" {
  ami = "ami-058a8a5ab36292159"
  instance_type = "t2.micro"
  key_name = "k8s"
  tags = {
   Name = "Terraform-server"
    }
}
```

-execution



```
Plan: 1 to add, 0 to change, 1 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.s3_bucket: Destroying... [id=s3backend]
aws_instance.test-server: Creating...
aws_instance.test-server: Still creating... [10s elapsed]
aws_instance.test-server: Still creating... [20s elapsed]
aws_instance.test-server: Creation complete after 24s [id=i-0ce2013edf729609b]
```

-Check ec2 creation in aws



Instances (1/1)

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	P
☑	Terraform-server	i-0ce2013edf729609b	Running	t2.micro	2/2 checks passec	View alarms +	us-east-2a	e

---Remote provisioners

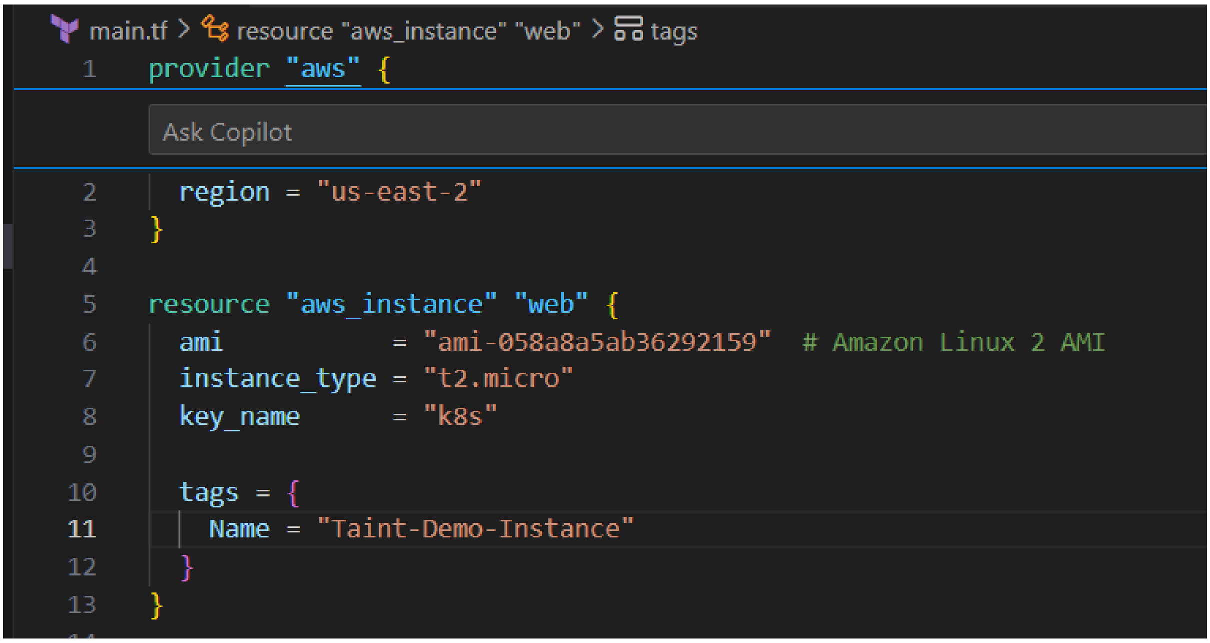
---Local provisioners

---Terraform taint and untaint:

These are used to force recreation of a resource, even if the code hasn't changed.

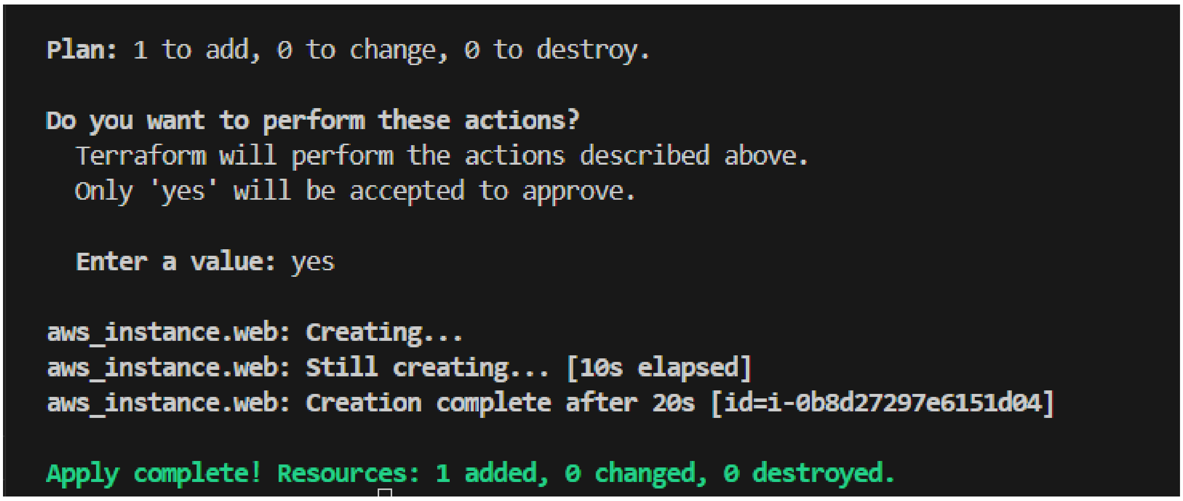
--Create an EC2 instance

Template



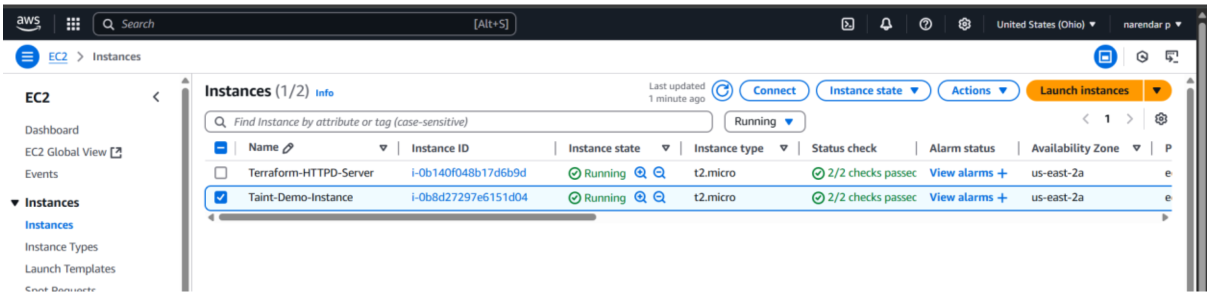
```
provider "aws" {
  region = "us-east-2"
}

resource "aws_instance" "web" {
  ami           = "ami-058a8a5ab36292159"  # Amazon Linux 2 AMI
  instance_type = "t2.micro"
  key_name      = "k8s"

  tags = {
    Name = "Taint-Demo-Instance"
  }
}
```

Execution



```
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.web: Creating...
aws_instance.web: Still creating... [10s elapsed]
aws_instance.web: Creation complete after 20s [id=i-0b8d27297e6151d04]

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

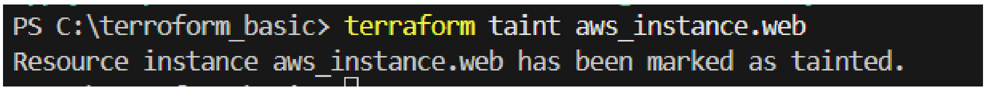
Check



Instances (1/2) Info

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	P
□	Terraform-HTTPD-Server	i-0b140f048b17d6b9d	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	us-east-2a	e
☑	Taint-Demo-Instance	i-0b8d27297e6151d04	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	us-east-2a	e

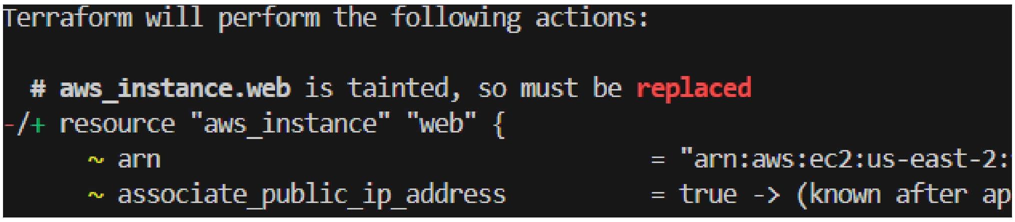
--Force Recreation with terraform taint its tainted



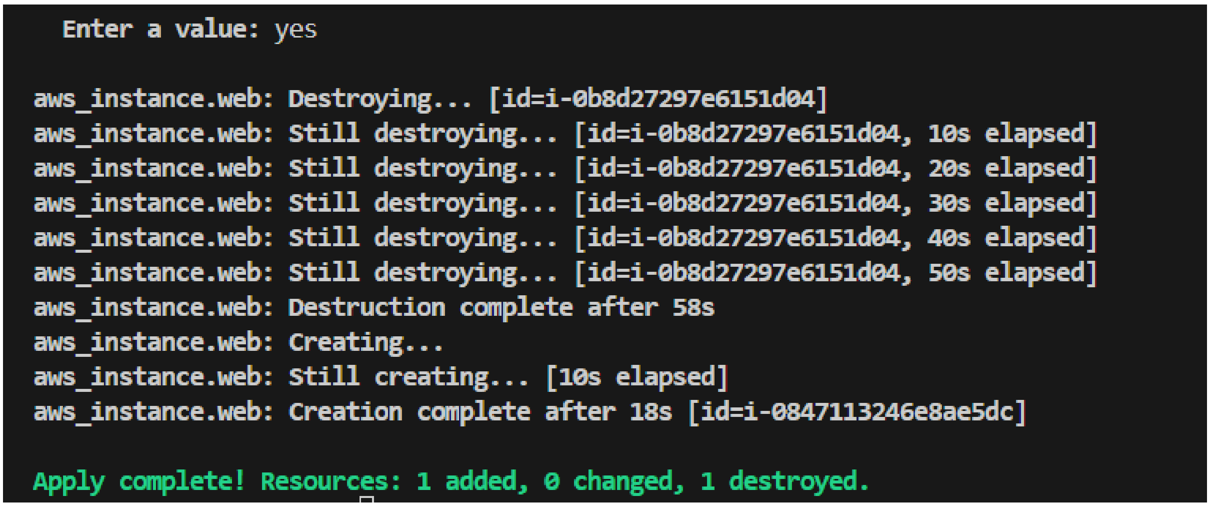
```
PS C:\terroform_basic> terraform taint aws_instance.web
Resource instance aws_instance.web has been marked as tainted.
```

Apply again

Terraform will now destroy the instance and recreate it automatically



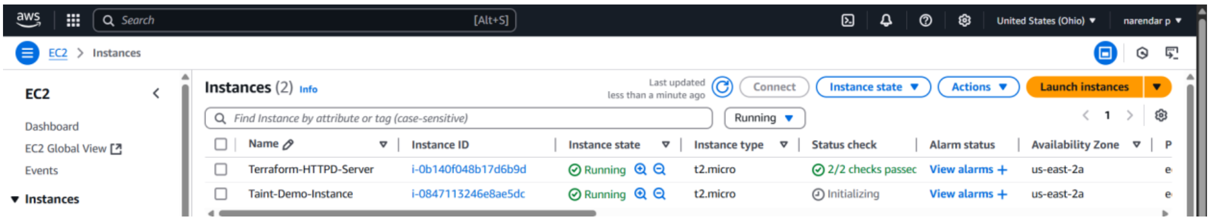
```
Terraform will perform the following actions:

  # aws_instance.web is tainted, so must be replaced
-/+ resource "aws_instance" "web" {
      ~ arn                                  = "arn:aws:ec2:us-east-2:
      ~ associate_public_ip_address          = true -> (known after ap
```



```
    Enter a value: yes

  aws_instance.web: Destroying... [id=i-0b8d27297e6151d04]
  aws_instance.web: Still destroying... [id=i-0b8d27297e6151d04, 10s elapsed]
  aws_instance.web: Still destroying... [id=i-0b8d27297e6151d04, 20s elapsed]
  aws_instance.web: Still destroying... [id=i-0b8d27297e6151d04, 30s elapsed]
  aws_instance.web: Still destroying... [id=i-0b8d27297e6151d04, 40s elapsed]
  aws_instance.web: Still destroying... [id=i-0b8d27297e6151d04, 50s elapsed]
  aws_instance.web: Destruction complete after 58s
  aws_instance.web: Creating...
  aws_instance.web: Still creating... [10s elapsed]
  aws_instance.web: Creation complete after 18s [id=i-0847113246e8ae5dc]

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

Check instance

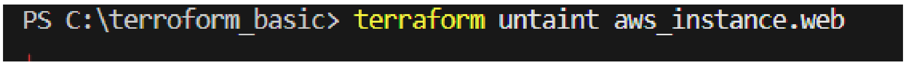
Shutdown & recreate instance



Instances (2) Info

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	P
□	Terraform-HTTPD-Server	i-0b140f048b17d6b9d	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	us-east-2a	e
□	Taint-Demo-Instance	i-0847113246e8ae5dc	✓ Running	t2.micro	Initializing	View alarms +	us-east-2a	e

-- terraform untaint



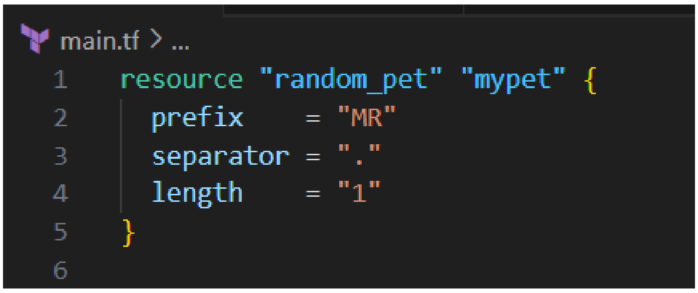
```
PS C:\terroform_basic> terraform untaint aws_instance.web
```

So now terraform untaint aws_instance.web

Now, Terraform will not destroy the instance — it will just validate the existing state.

Debugging:

Create a template



main.tf > ...

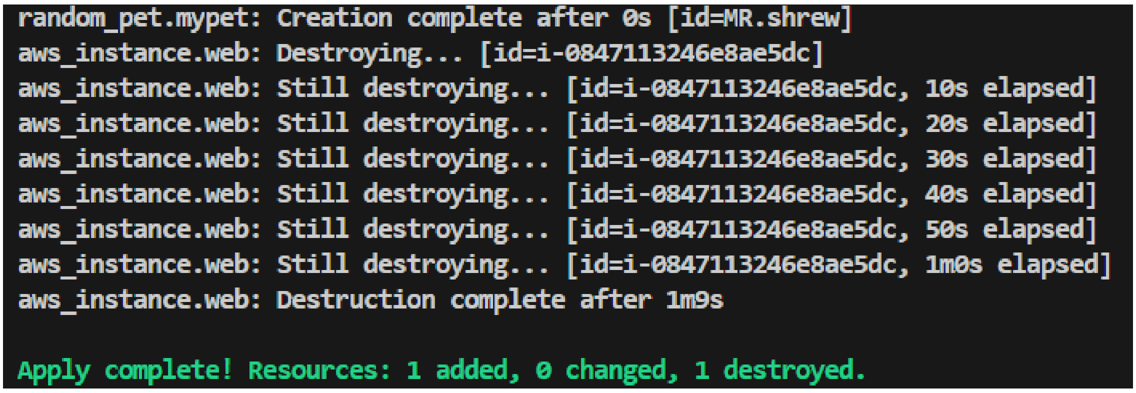
```
resource "random_pet" "mypet" {
  prefix    = "MR"
  separator = "."
  length    = "1"
}

```

Then apply these two commands

$env:TF_LOG = "DEBUG"

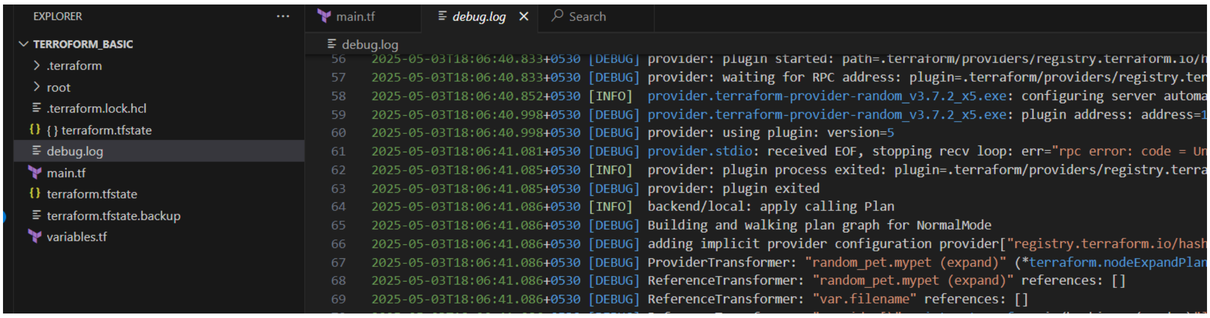
$env:TF_LOG_PATH = "debug.log"

Then execute



```
random_pet.mypet: Creation complete after 0s [id=MR.shrew]
aws_instance.web: Destroying... [id=i-0847113246e8ae5dc]
aws_instance.web: Still destroying... [id=i-0847113246e8ae5dc, 10s elapsed]
aws_instance.web: Still destroying... [id=i-0847113246e8ae5dc, 20s elapsed]
aws_instance.web: Still destroying... [id=i-0847113246e8ae5dc, 30s elapsed]
aws_instance.web: Still destroying... [id=i-0847113246e8ae5dc, 40s elapsed]
aws_instance.web: Still destroying... [id=i-0847113246e8ae5dc, 50s elapsed]
aws_instance.web: Still destroying... [id=i-0847113246e8ae5dc, 1m0s elapsed]
aws_instance.web: Destruction complete after 1m9s

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

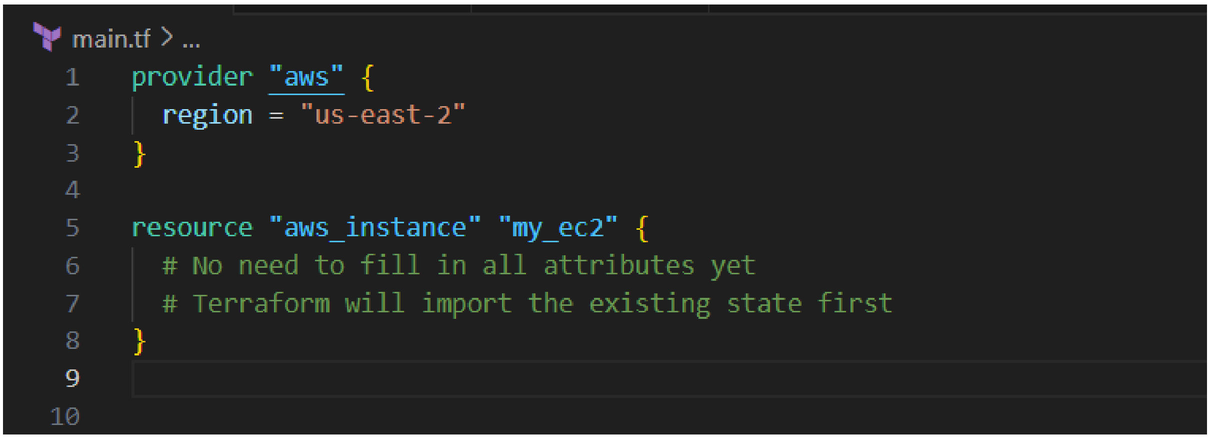
debug.log file got created click on that check log



```
2025-05-03T18:06:40.833+0530 [DEBUG] provider: plugin started: path=.terraform/providers/registry.terraform.io/h
2025-05-03T18:06:40.833+0530 [DEBUG] provider: waiting for RPC address: plugin=.terraform/providers/registry.ter
2025-05-03T18:06:40.852+0530 [INFO]  provider.terraform-provider-random_v3.7.2_x5.exe: configuring server automa
2025-05-03T18:06:40.998+0530 [DEBUG] provider.terraform-provider-random_v3.7.2_x5.exe: plugin address: address=1
2025-05-03T18:06:40.998+0530 [DEBUG] provider: using plugin: version=5
2025-05-03T18:06:41.081+0530 [DEBUG] provider.stdio: received EOF, stopping recv loop: err="rpc error: code = Un
2025-05-03T18:06:41.085+0530 [INFO]  provider: plugin process exited: plugin=.terraform/providers/registry.terra
2025-05-03T18:06:41.085+0530 [DEBUG] provider: plugin exited
2025-05-03T18:06:41.086+0530 [INFO]  backend/local: apply calling Plan
2025-05-03T18:06:41.086+0530 [DEBUG] Building and walking plan graph for NormalMode
2025-05-03T18:06:41.086+0530 [DEBUG] adding implicit provider configuration provider["registry.terraform.io/hash
2025-05-03T18:06:41.086+0530 [DEBUG] ProviderTransformer: "random_pet.mypet (expand)" (*terraform.nodeExpandPlan
2025-05-03T18:06:41.086+0530 [DEBUG] ReferenceTransformer: "random_pet.mypet (expand)" references: []
2025-05-03T18:06:41.086+0530 [DEBUG] ReferenceTransformer: "var.filename" references: []
```

Terraform import:

Create template



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

resource "aws_instance" "my_ec2" {
  # No need to fill in all attributes yet
  # Terraform will import the existing state first
}

```

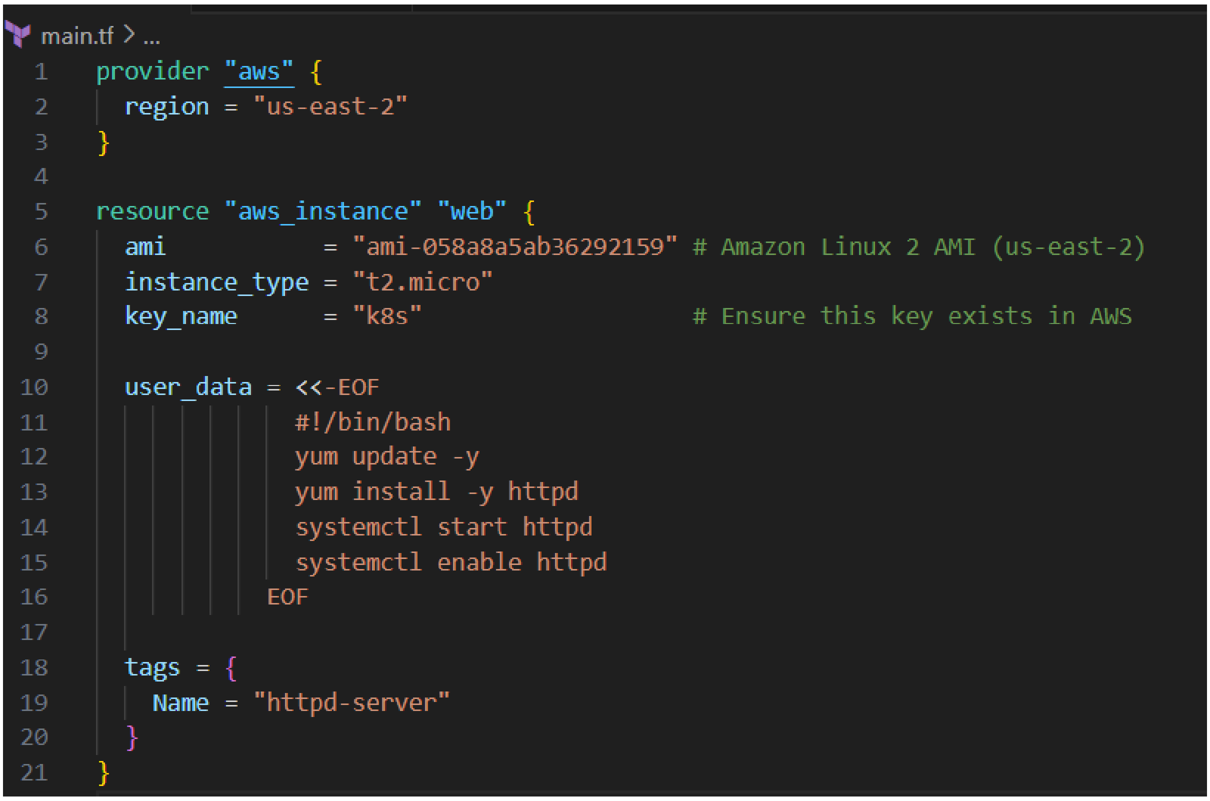
Apply import command

terraform import aws_instance.my_ec2 i-0abc1234def567890

This command maps the existing EC2 instance to the Terraform resource aws_instance.my_ec2.

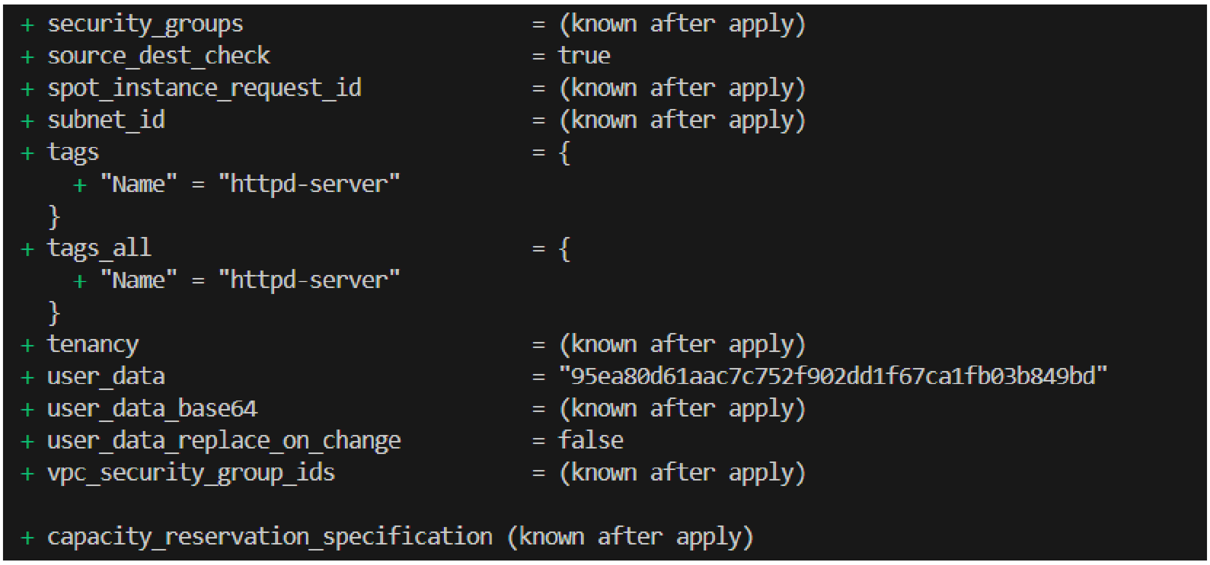
3)Create one ec2 instance with httpd installed using terraform script.

--terraform template Create one ec2 instance with httpd installed using terraform script

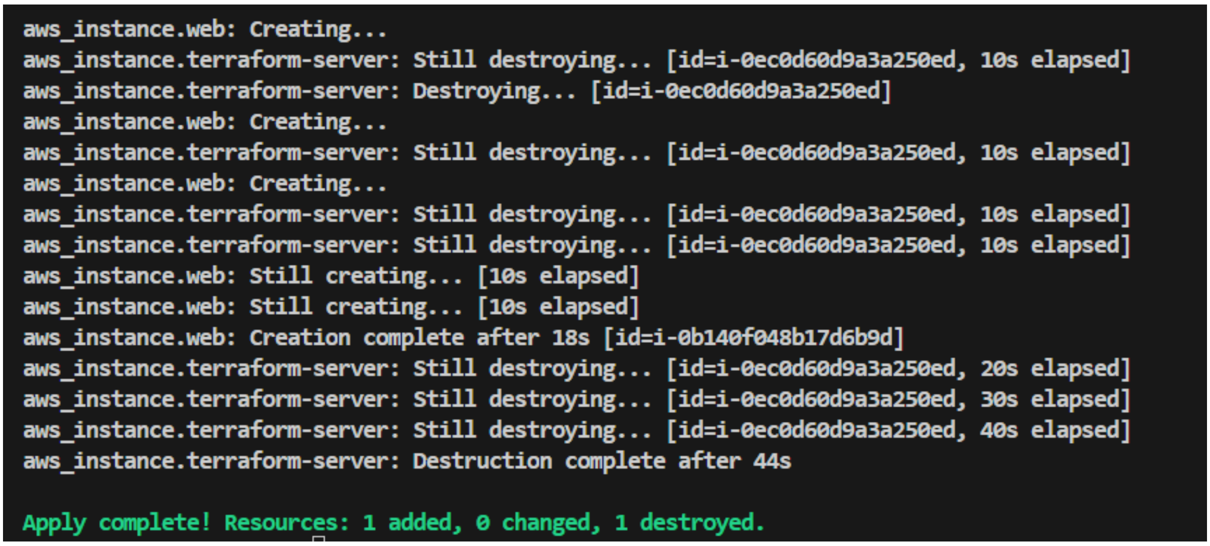


```
provider "aws" {
  region = "us-east-2"
}

resource "aws_instance" "web" {
  ami           = "ami-058a8a5ab36292159" # Amazon Linux 2 AMI (us-east-2)
  instance_type = "t2.micro"
  key_name      = "k8s"                   # Ensure this key exists in AWS

  user_data = <<-EOF
            #!/bin/bash
            yum update -y
            yum install -y httpd
            systemctl start httpd
            systemctl enable httpd
          EOF

  tags = {
    Name = "httpd-server"
  }
}
```

--execution



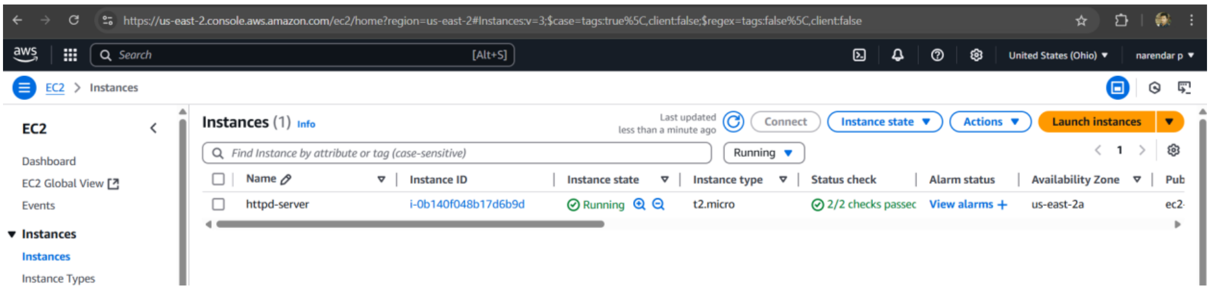
```
+ security_groups                      = (known after apply)
+ source_dest_check                    = true
+ spot_instance_request_id             = (known after apply)
+ subnet_id                            = (known after apply)
+ tags                                 = {
    + "Name" = "httpd-server"
  }
+ tags_all                             = {
    + "Name" = "httpd-server"
  }
+ tenancy                              = (known after apply)
+ user_data                            = "95ea80d61aac7c752f902dd1f67ca1fb03b849bd"
+ user_data_base64                     = (known after apply)
+ user_data_replace_on_change          = false
+ vpc_security_group_ids               = (known after apply)

+ capacity_reservation_specification (known after apply)
```



```
aws_instance.web: Creating...
aws_instance.terraform-server: Still destroying... [id=i-0ec0d60d9a3a250ed, 10s elapsed]
aws_instance.terraform-server: Destroying... [id=i-0ec0d60d9a3a250ed]
aws_instance.web: Creating...
aws_instance.terraform-server: Still destroying... [id=i-0ec0d60d9a3a250ed, 10s elapsed]
aws_instance.web: Creating...
aws_instance.terraform-server: Still destroying... [id=i-0ec0d60d9a3a250ed, 10s elapsed]
aws_instance.terraform-server: Still destroying... [id=i-0ec0d60d9a3a250ed, 10s elapsed]
aws_instance.web: Still creating... [10s elapsed]
aws_instance.web: Still creating... [10s elapsed]
aws_instance.web: Creation complete after 18s [id=i-0b140f048b17d6b9d]
aws_instance.terraform-server: Still destroying... [id=i-0ec0d60d9a3a250ed, 20s elapsed]
aws_instance.terraform-server: Still destroying... [id=i-0ec0d60d9a3a250ed, 30s elapsed]
aws_instance.terraform-server: Still destroying... [id=i-0ec0d60d9a3a250ed, 40s elapsed]
aws_instance.terraform-server: Destruction complete after 44s

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

--check aws ec2 created created or not

-its created

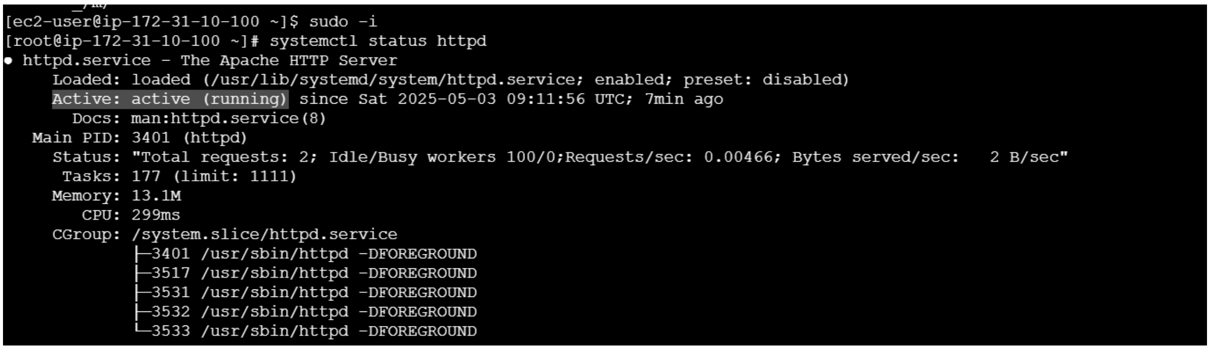


https://us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#Instances:v=3;$case=tags:true%5C,client:false;$regex=tags:false%5C,client:false

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Pub
□	httpd-server	i-0b140f048b17d6b9d	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	us-east-2a	ec2-

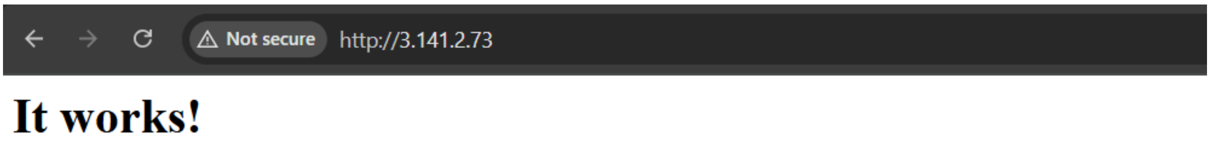
-now connect to ec2 check httpd running or not

-its running



```
[ec2-user@ip-172-31-10-100 ~]$ sudo -i
[root@ip-172-31-10-100 ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
     Active: active (running) since Sat 2025-05-03 09:11:56 UTC; 7min ago
       Docs: man:httpd.service(8)
   Main PID: 3401 (httpd)
     Status: "Total requests: 2; Idle/Busy workers 100/0;Requests/sec: 0.00466; Bytes served/sec:   2 B/sec"
      Tasks: 177 (limit: 1111)
     Memory: 13.1M
        CPU: 299ms
     CGroup: /system.slice/httpd.service
             ├─3401 /usr/sbin/httpd -DFOREGROUND
             ├─3517 /usr/sbin/httpd -DFOREGROUND
             ├─3531 /usr/sbin/httpd -DFOREGROUND
             ├─3532 /usr/sbin/httpd -DFOREGROUND
             └─3533 /usr/sbin/httpd -DFOREGROUND
```

-check in browser

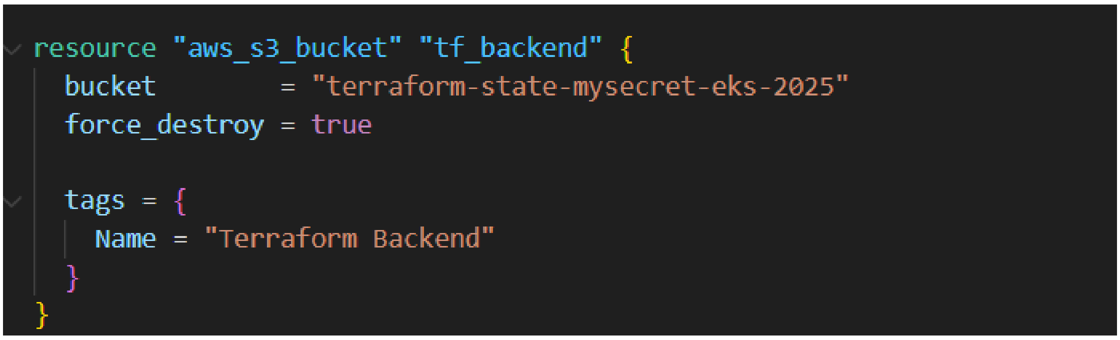


http://3.141.2.73

It works!

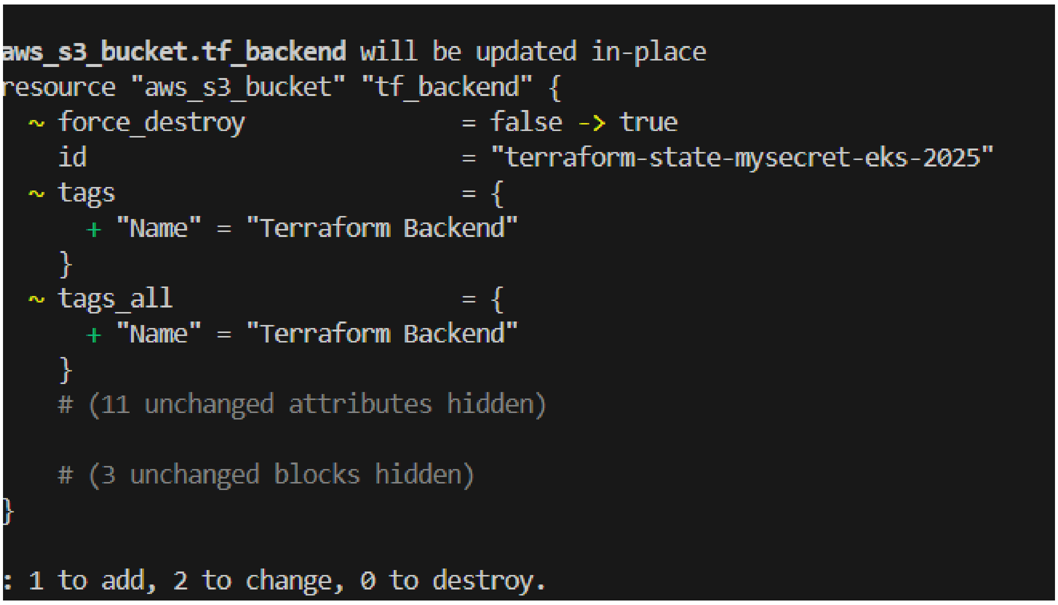
4) Setup s3 as backend to the task 3.

--template



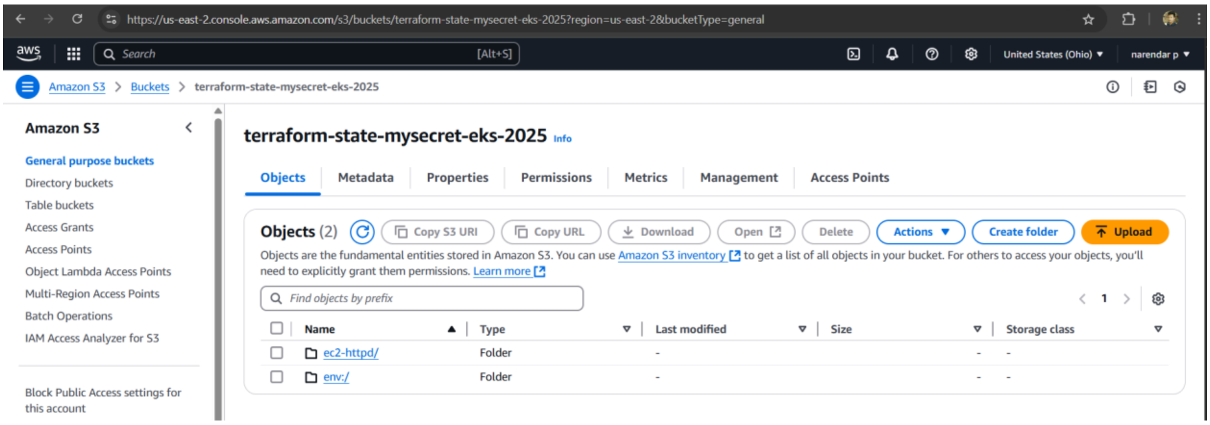
```
resource "aws_s3_bucket" "tf_backend" {
  bucket        = "terraform-state-mysecret-eks-2025"
  force_destroy = true

  tags = {
    Name = "Terraform Backend"
  }
}
```

--execution



```
aws_s3_bucket.tf_backend will be updated in-place
resource "aws_s3_bucket" "tf_backend" {
  ~ force_destroy                  = false -> true
    id                             = "terraform-state-mysecret-eks-2025"
  ~ tags                           = {
      + "Name" = "Terraform Backend"
    }
  ~ tags_all                       = {
      + "Name" = "Terraform Backend"
    }
    # (11 unchanged attributes hidden)

    # (3 unchanged blocks hidden)
}

: 1 to add, 2 to change, 0 to destroy.
```

--check aws backend got created in s3



https://us-east-2.console.aws.amazon.com/s3/buckets/terraform-state-mysecret-eks-2025?region=us-east-2&bucketType=general

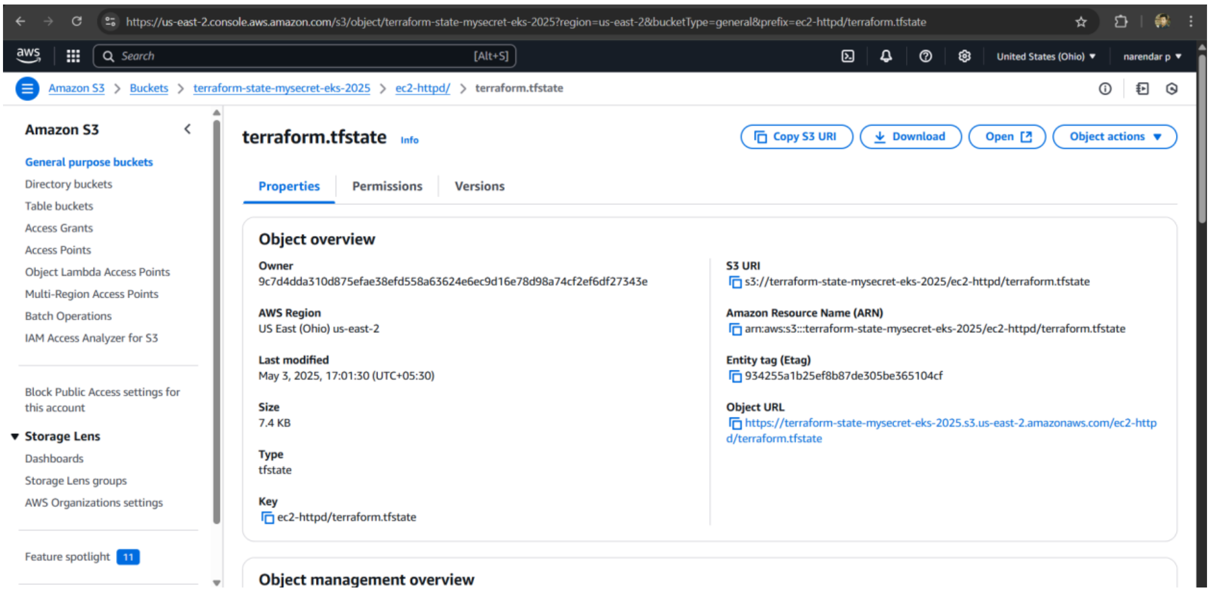
Amazon S3 > Buckets > terraform-state-mysecret-eks-2025

terraform-state-mysecret-eks-2025 Info

Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

	Name	Type	Last modified	Size	Storage class
□	ec2-httpd/	Folder	-	-	-
□	env:/	Folder	-	-	-



https://us-east-2.console.aws.amazon.com/s3/object/terraform-state-mysecret-eks-2025?region=us-east-2&bucketType=general&prefix=ec2-httpd/terraform.tfstate
aws
Search
[Alt+S]
United States (Ohio)
narendar p
Amazon S3 > Buckets > terraform-state-mysecret-eks-2025 > ec2-httpd/ > terraform.tfstate
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 11
Info
Copy S3 URI
Download
Open
Object actions
Properties
Permissions
Versions

terraform.tfstate

Object overview

Owner
9c7d4dda310d875efae38efd558a63624e6ec9d16e78d98a74cf2ef6df27343e

AWS Region
US East (Ohio) us-east-2

Last modified
May 3, 2025, 17:01:30 (UTC+05:30)

Size
7.4 KB

Type
tfstate

Key
ec2-httpd/terraform.tfstate

S3 URI
s3://terraform-state-mysecret-eks-2025/ec2-httpd/terraform.tfstate

Amazon Resource Name (ARN)
arn:aws:s3:::terraform-state-mysecret-eks-2025/ec2-httpd/terraform.tfstate

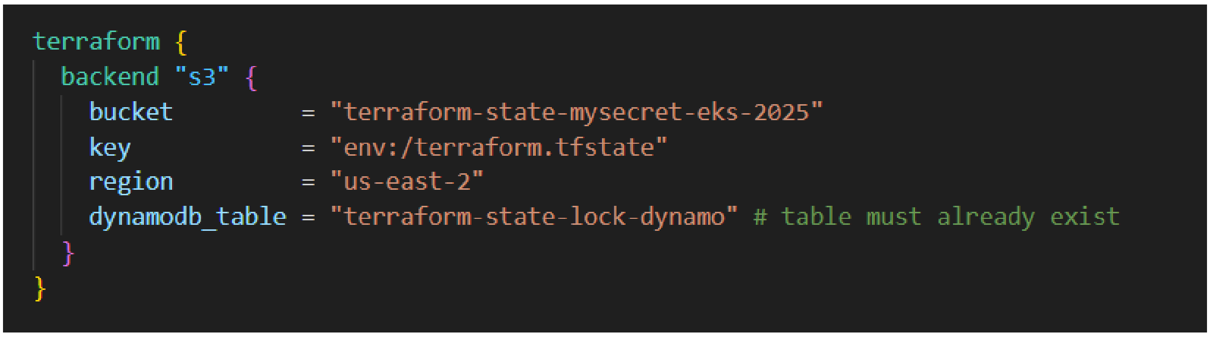
Entity tag (Etag)
934255a1b25ef8b87de305be365104cf

Object URL
https://terraform-state-mysecret-eks-2025.s3.us-east-2.amazonaws.com/ec2-httpd/terraform.tfstate

Object management overview

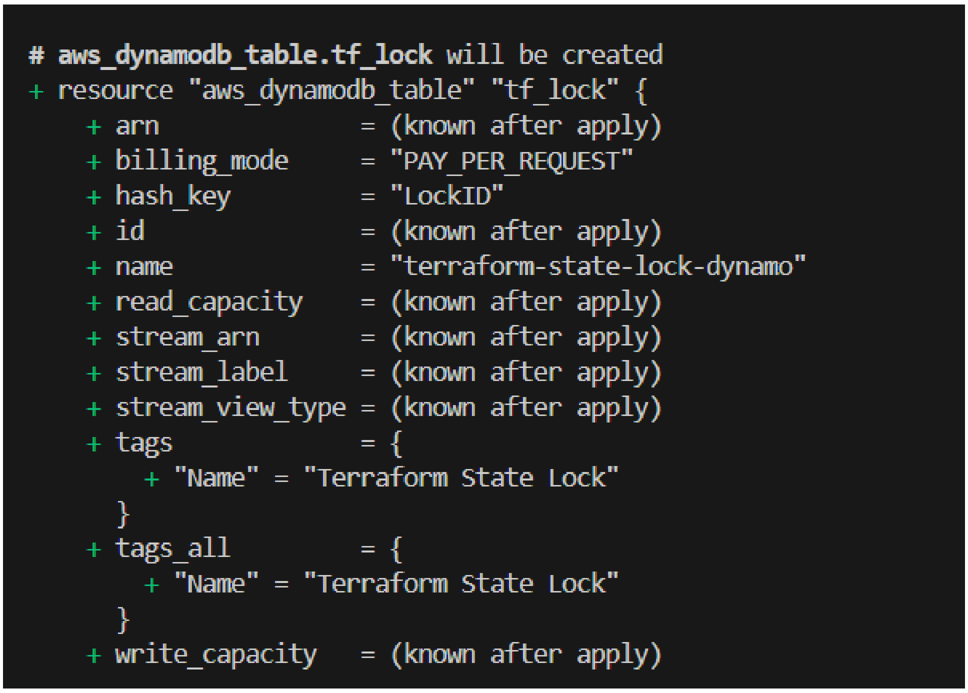
5) Setup dynamo db locking for task3.

--template



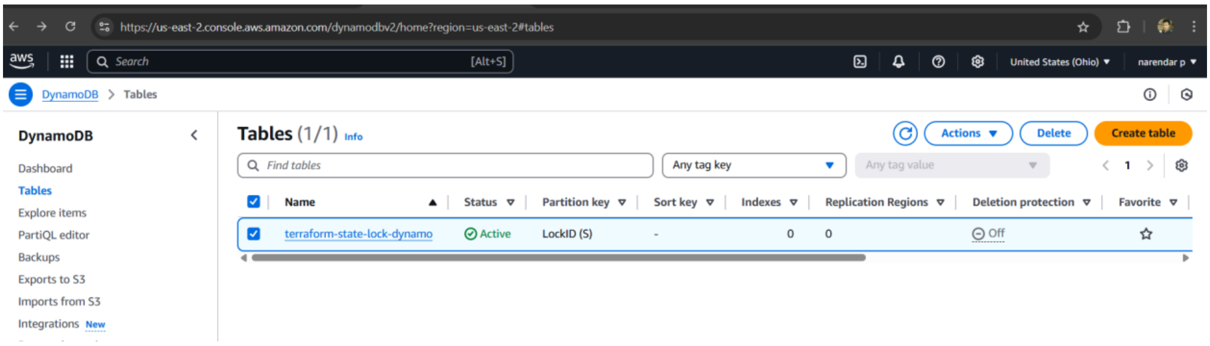
```
terraform {
  backend "s3" {
    bucket         = "terraform-state-mysecret-eks-2025"
    key            = "env:/terraform.tfstate"
    region         = "us-east-2"
    dynamodb_table = "terraform-state-lock-dynamo" # table must already exist
  }
}
```

--execution



```
# aws_dynamodb_table.tf_lock will be created
+ resource "aws_dynamodb_table" "tf_lock" {
    + arn              = (known after apply)
    + billing_mode     = "PAY_PER_REQUEST"
    + hash_key         = "LockID"
    + id               = (known after apply)
    + name             = "terraform-state-lock-dynamo"
    + read_capacity    = (known after apply)
    + stream_arn       = (known after apply)
    + stream_label     = (known after apply)
    + stream_view_type = (known after apply)
    + tags             = {
        + "Name" = "Terraform State Lock"
      }
    + tags_all         = {
        + "Name" = "Terraform State Lock"
      }
    + write_capacity   = (known after apply)
```

--check aws dynamo db locking created

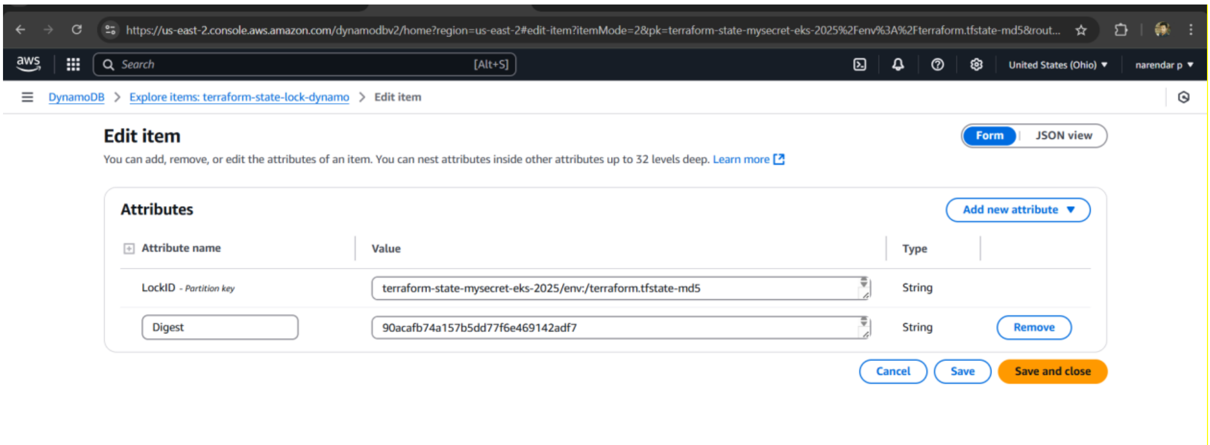


https://us-east-2.console.aws.amazon.com/dynamodbv2/home?region=us-east-2#tables

DynamoDB > Tables

Tables (1/1) Info

☑	Name	Status	Partition key	Sort key	Indexes	Replication Regions	Deletion protection	Favorite
☑	terraform-state-lock-dynamo	✓ Active	LockID (S)	-	0	0	Off	☆



https://us-east-2.console.aws.amazon.com/dynamodbv2/home?region=us-east-2#edit-item?itemMode=2&pk=terraform-state-mysecret-eks-2025%2Fenv%3A%2Fterraform.tfstate-md5&rout...

DynamoDB > Explore items: terraform-state-lock-dynamo > Edit item

Edit item

You can add, remove, or edit the attributes of an item. You can nest attributes inside other attributes up to 32 levels deep. Learn more

Attributes

Attribute name	Value	Type
LockID - Partition key	terraform-state-mysecret-eks-2025/env:/terraform.tfstate-md5	String
Digest	90acafb74a157b5dd77f6e469142adf7	String

6) Watch terraform-06 video.

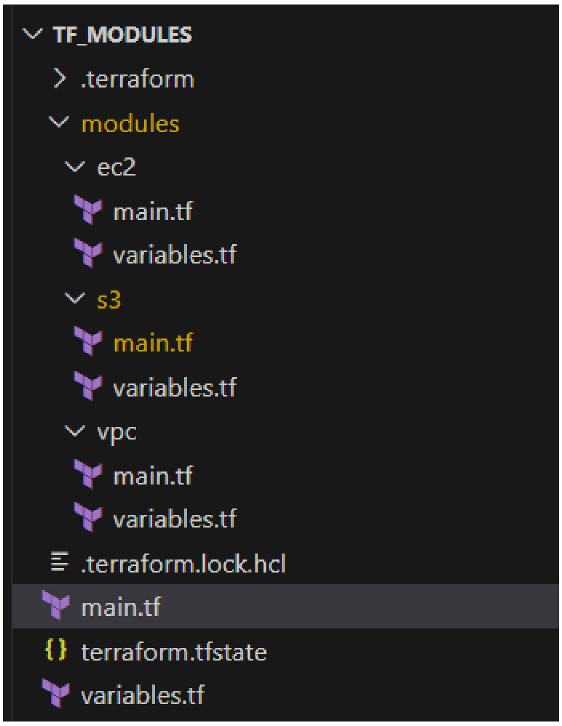
--completed

7) Execute the script shown in video.

--completed in 8th and 9th task

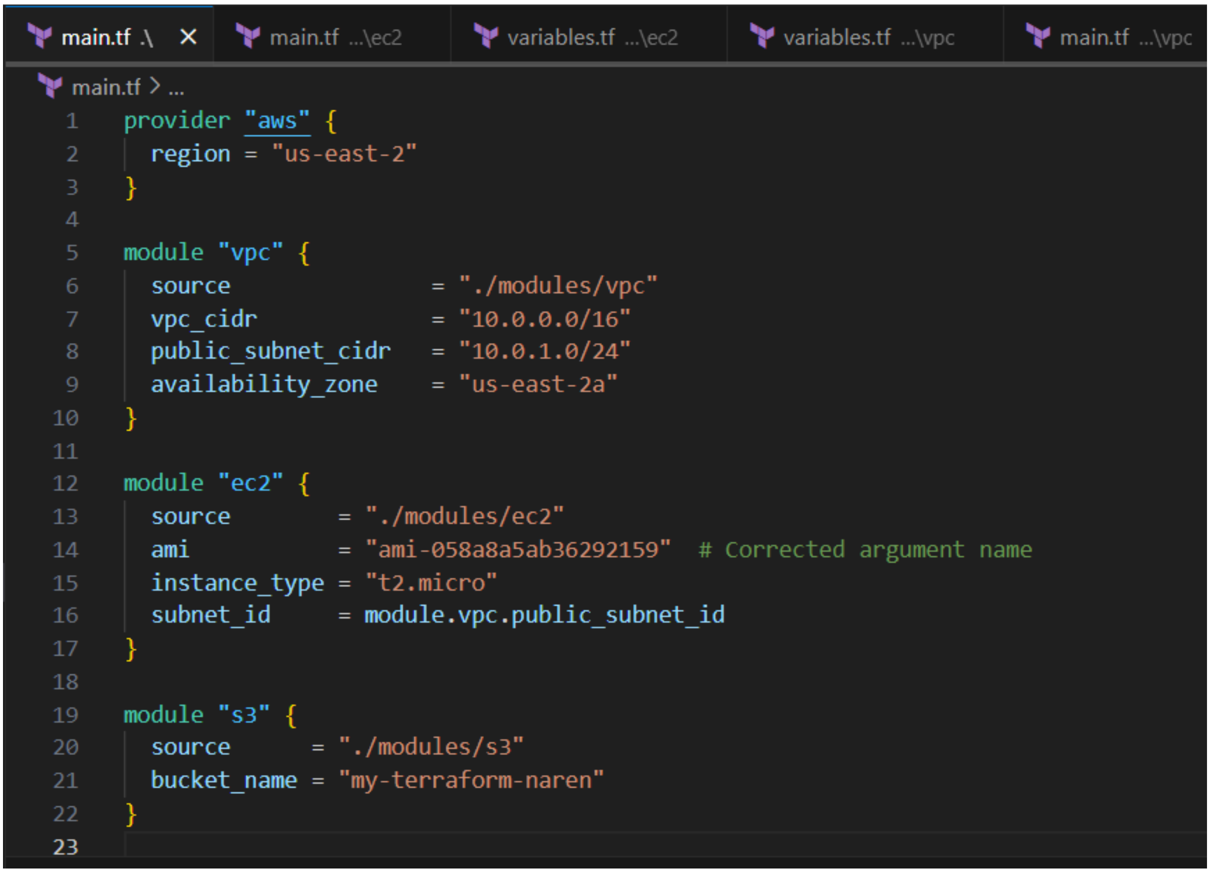
8) Provision ec2, s3 and vpc using Terraform modules.

--make module directory stucture



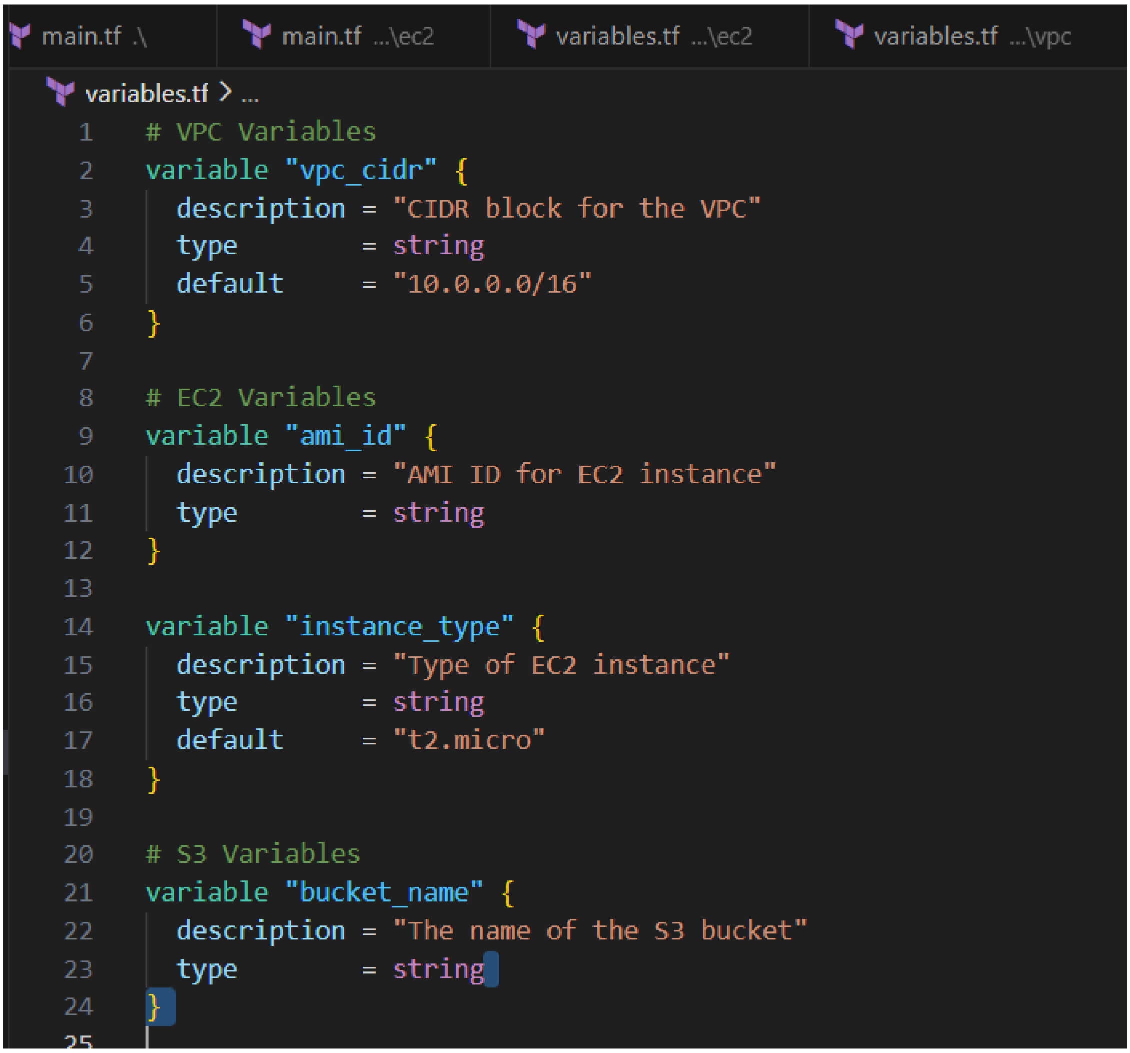
```
TF_MODULES
  .terraform
  modules
    ec2
      main.tf
      variables.tf
    s3
      main.tf
      variables.tf
    vpc
      main.tf
      variables.tf
  .terraform.lock.hcl
  main.tf
  terraform.tfstate
  variables.tf
```

--root main.tf



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

module "vpc" {
  source              = "./modules/vpc"
  vpc_cidr            = "10.0.0.0/16"
  public_subnet_cidr  = "10.0.1.0/24"
  availability_zone   = "us-east-2a"
}

module "ec2" {
  source        = "./modules/ec2"
  ami           = "ami-058a8a5ab36292159"  # Corrected argument name
  instance_type = "t2.micro"
  subnet_id     = module.vpc.public_subnet_id
}

module "s3" {
  source      = "./modules/s3"
  bucket_name = "my-terraform-naren"
}

```

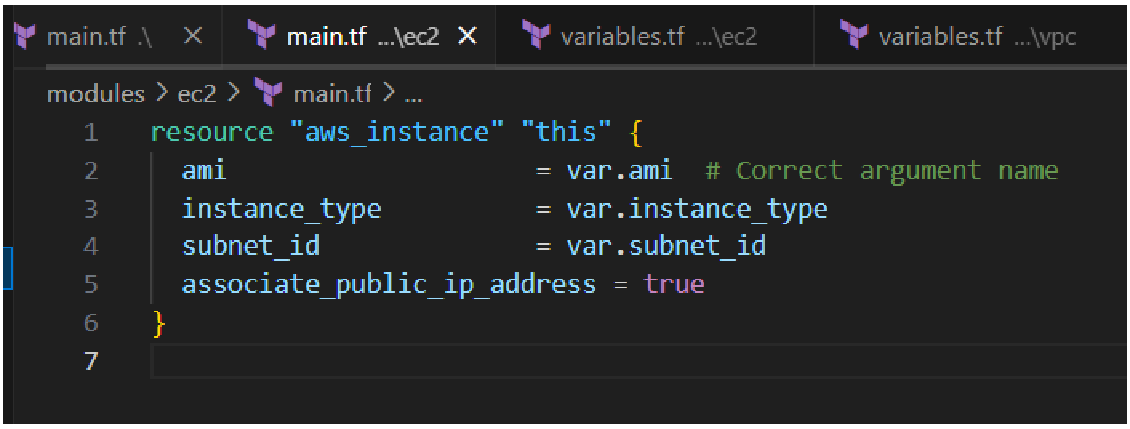
--root variable.tf



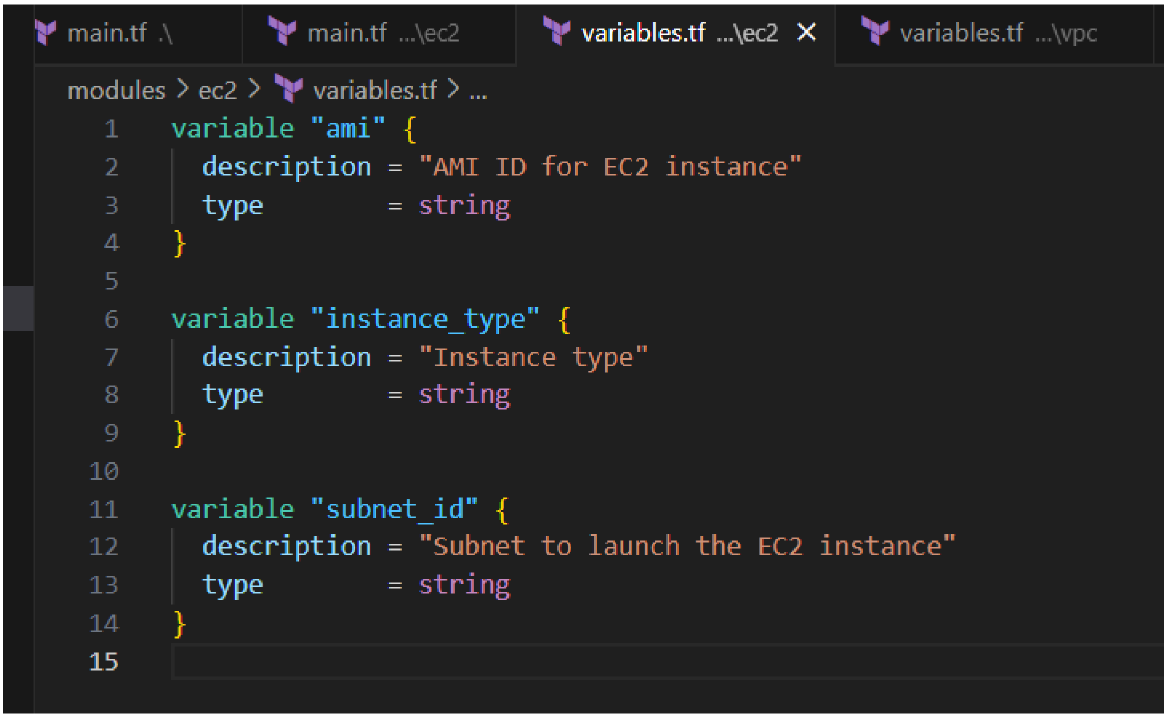
```
# VPC Variables
variable "vpc_cidr" {
  description = "CIDR block for the VPC"
  type        = string
  default     = "10.0.0.0/16"
}

# EC2 Variables
variable "ami_id" {
  description = "AMI ID for EC2 instance"
  type        = string
}

variable "instance_type" {
  description = "Type of EC2 instance"
  type        = string
  default     = "t2.micro"
}

# S3 Variables
variable "bucket_name" {
  description = "The name of the S3 bucket"
  type        = string
}
```

--make module in that make again 3 directories ec2 s3 vpc in each create main.tf and variable.tf files

-ec2

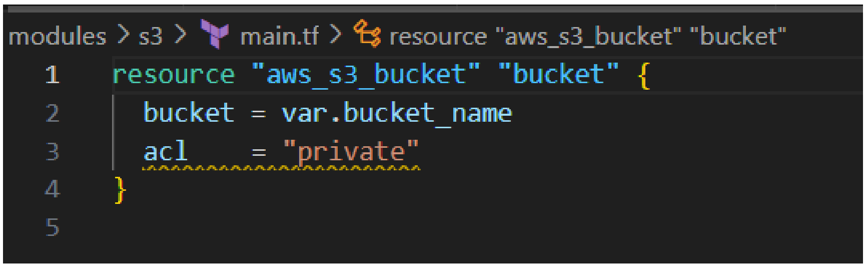


```
resource "aws_instance" "this" {
  ami                     = var.ami  # Correct argument name
  instance_type           = var.instance_type
  subnet_id               = var.subnet_id
  associate_public_ip_address = true
}
```

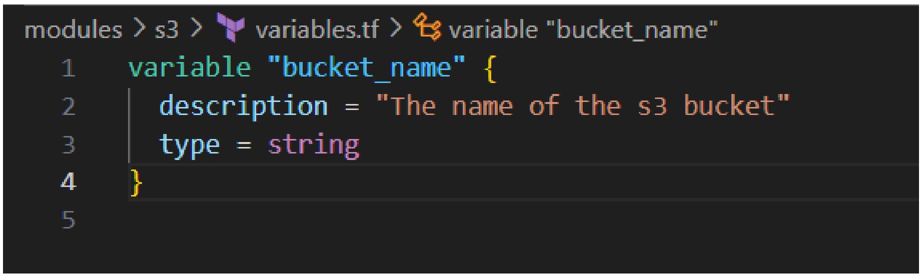


```
variable "ami" {
  description = "AMI ID for EC2 instance"
  type        = string
}

variable "instance_type" {
  description = "Instance type"
  type        = string
}

variable "subnet_id" {
  description = "Subnet to launch the EC2 instance"
  type        = string
}

```

-s3

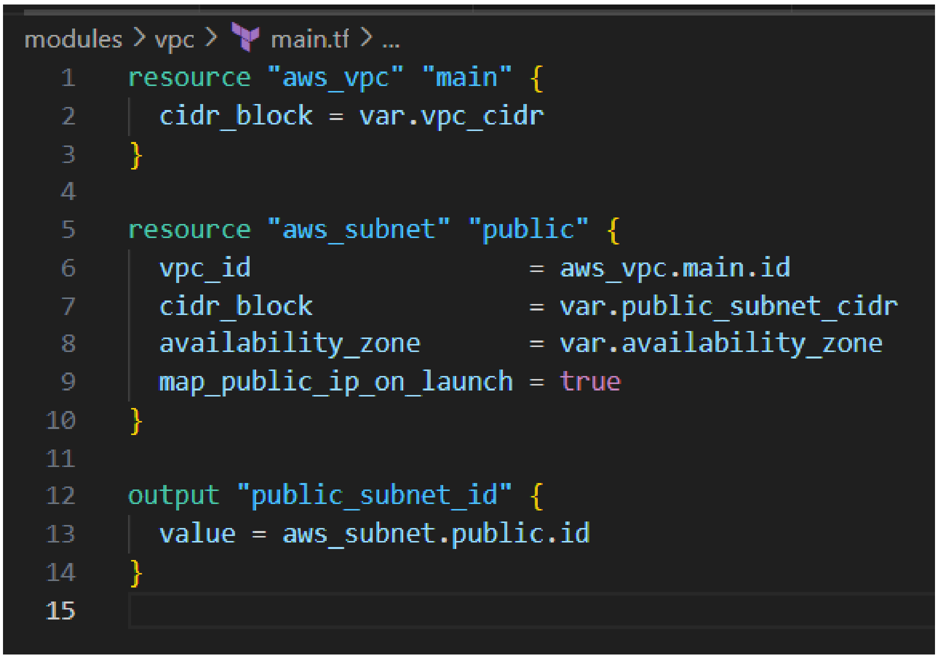


```
resource "aws_s3_bucket" "bucket" {
  bucket = var.bucket_name
  acl    = "private"
}

```

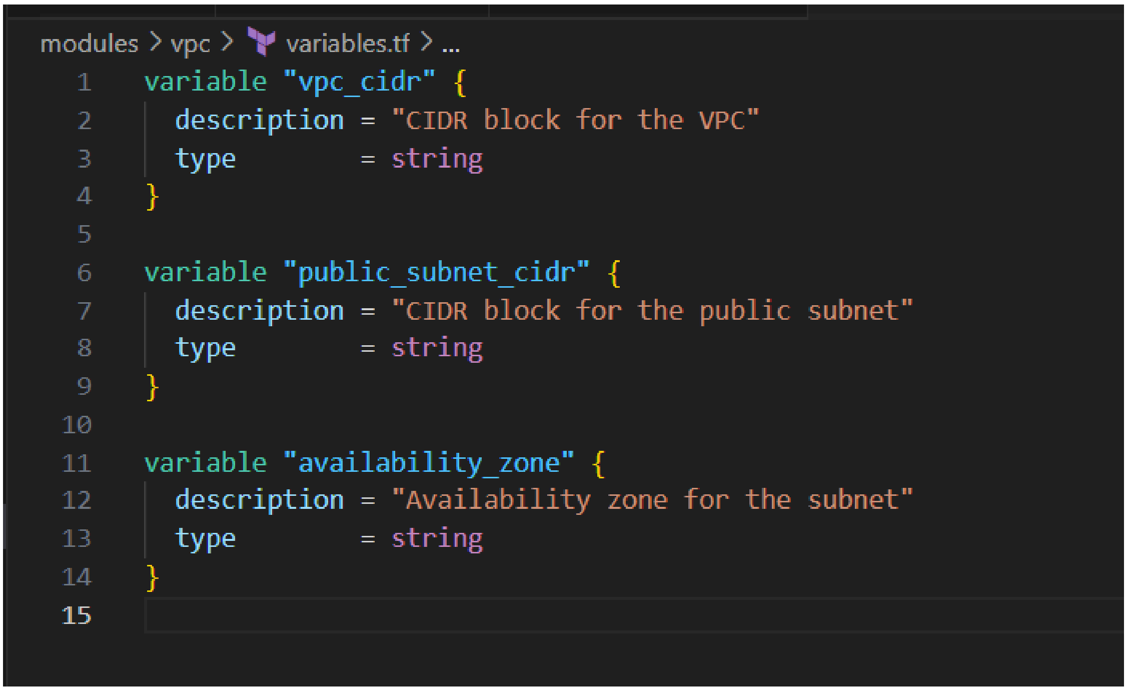


```
variable "bucket_name" {
  description = "The name of the s3 bucket"
  type = string
}

```

-vpc

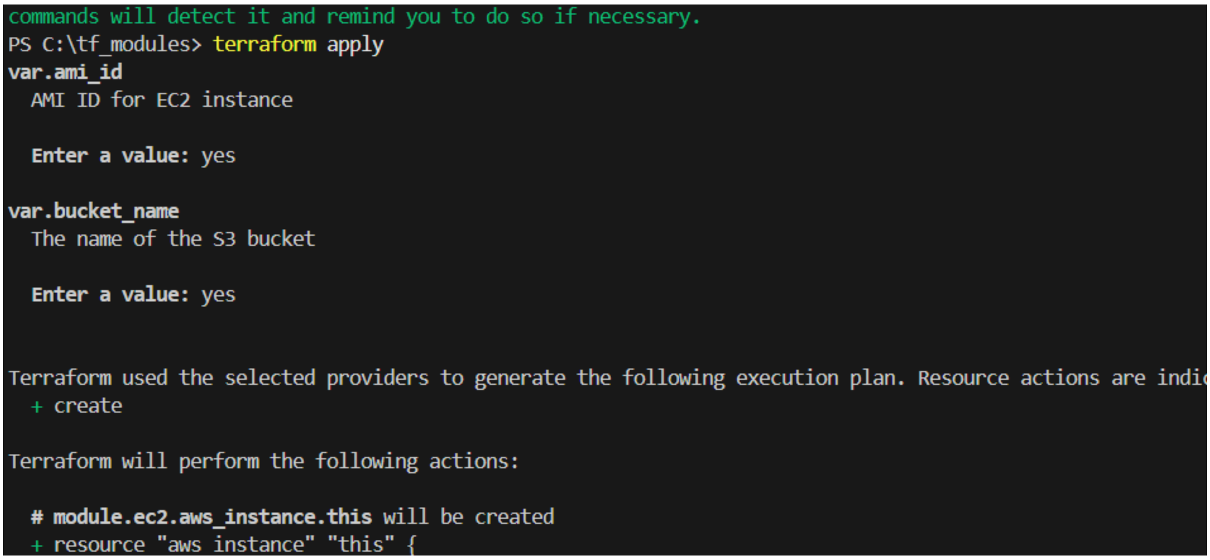


```
resource "aws_vpc" "main" {
  cidr_block = var.vpc_cidr
}

resource "aws_subnet" "public" {
  vpc_id                  = aws_vpc.main.id
  cidr_block              = var.public_subnet_cidr
  availability_zone       = var.availability_zone
  map_public_ip_on_launch = true
}

output "public_subnet_id" {
  value = aws_subnet.public.id
}

```

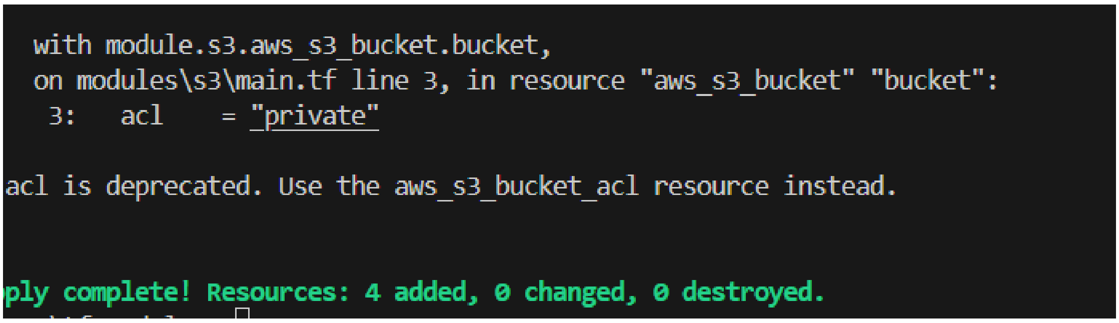


```
variable "vpc_cidr" {
  description = "CIDR block for the VPC"
  type        = string
}

variable "public_subnet_cidr" {
  description = "CIDR block for the public subnet"
  type        = string
}

variable "availability_zone" {
  description = "Availability zone for the subnet"
  type        = string
}

```

--now execute

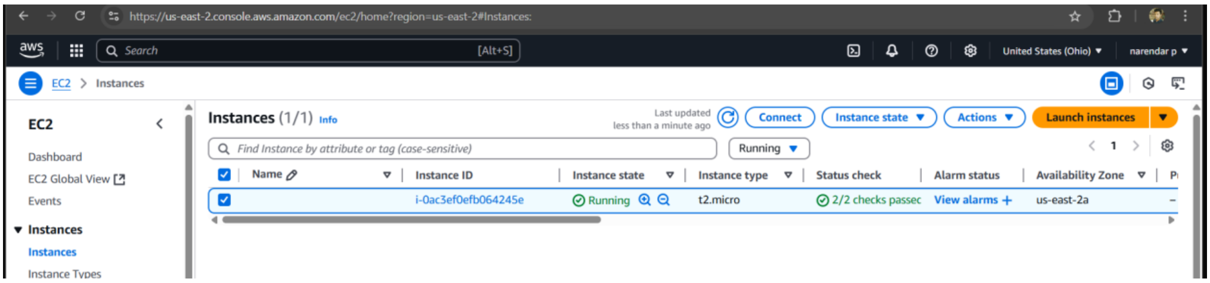


```
commands will detect it and remind you to do so if necessary.
PS C:\tf_modules> terraform apply
var.ami_id
  AMI ID for EC2 instance

  Enter a value: yes

var.bucket_name
  The name of the S3 bucket

  Enter a value: yes


Terraform used the selected providers to generate the following execution plan. Resource actions are indi
  + create

Terraform will perform the following actions:

  # module.ec2.aws_instance.this will be created
  + resource "aws_instance" "this" {
```



```
  with module.s3.aws_s3_bucket.bucket,
  on modules\s3\main.tf line 3, in resource "aws_s3_bucket" "bucket":
   3:   acl     = "private"

acl is deprecated. Use the aws_s3_bucket_acl resource instead.


ply complete! Resources: 4 added, 0 changed, 0 destroyed.
```

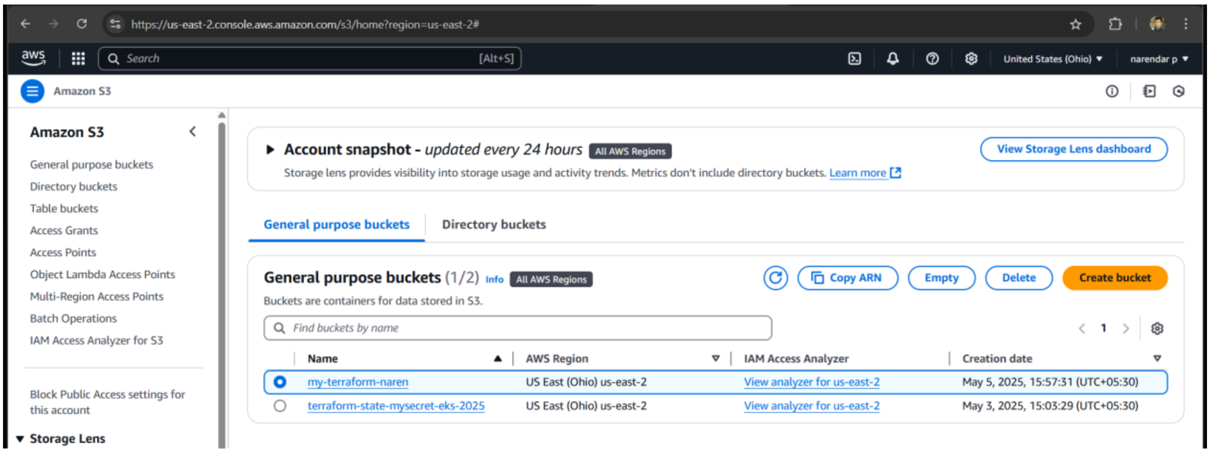
--check aws ec2



Instances (1/1) Info

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	P
☑		i-0ac3ef0efb064245e	Running	t2.micro	2/2 checks passec	View alarms +	us-east-2a	–

--check aws s3



https://us-east-2.console.aws.amazon.com/s3/home?region=us-east-2#

▸ **Account snapshot -** *updated every 24 hours* All AWS Regions

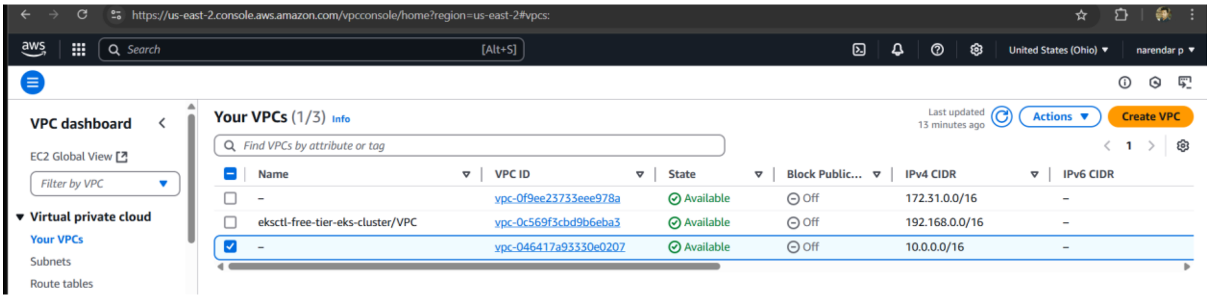
Storage lens provides visibility into storage usage and activity trends. Metrics don't include directory buckets. Learn more

General purpose buckets (1/2) Info All AWS Regions

Buckets are containers for data stored in S3.

	Name	AWS Region	IAM Access Analyzer	Creation date
●	my-terraform-naren	US East (Ohio) us-east-2	View analyzer for us-east-2	May 5, 2025, 15:57:31 (UTC+05:30)
○	terraform-state-mysecret-eks-2025	US East (Ohio) us-east-2	View analyzer for us-east-2	May 3, 2025, 15:03:29 (UTC+05:30)

--check vpc



https://us-east-2.console.aws.amazon.com/vpcconsole/home?region=us-east-2#vpcs:

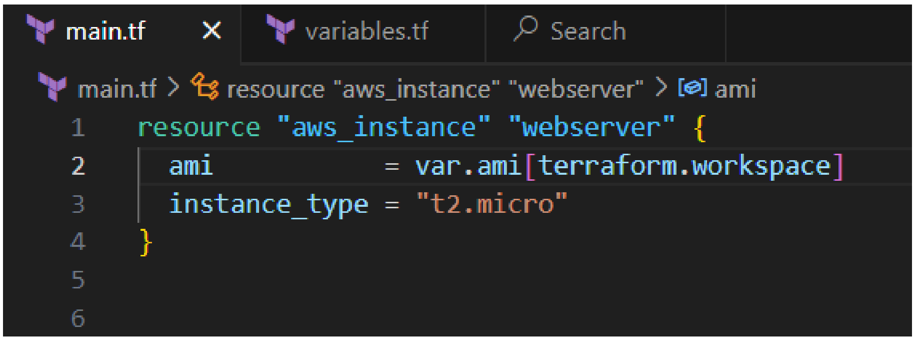
Your VPCs (1/3) Info

Last updated 13 minutes ago

	Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR
□	–	vpc-0f9ee23733eee978a	✓ Available	⊖ Off	172.31.0.0/16	–
□	eksctl-free-tier-eks-cluster/VPC	vpc-0c569f3cbd9b6eba3	✓ Available	⊖ Off	192.168.0.0/16	–
☑	–	vpc-046417a93330e0207	✓ Available	⊖ Off	10.0.0.0/16	–

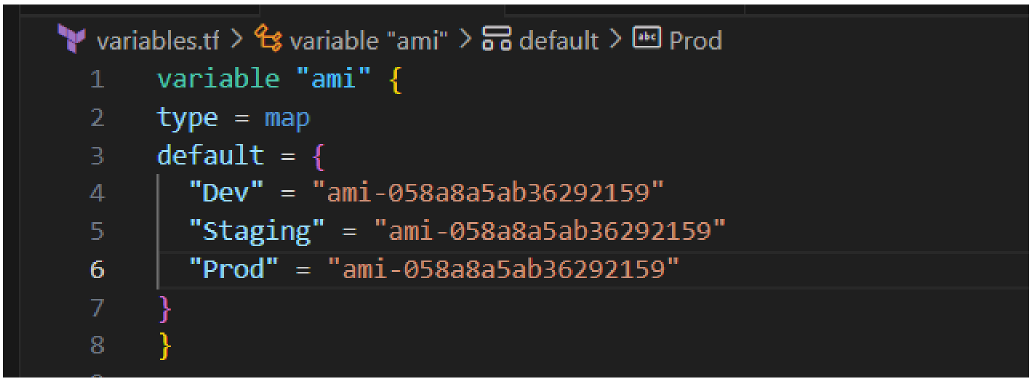
9) Provision ec2 for 3 different environments (Dev, Staging and Prod) using terraform workspaces.

--create main.tf



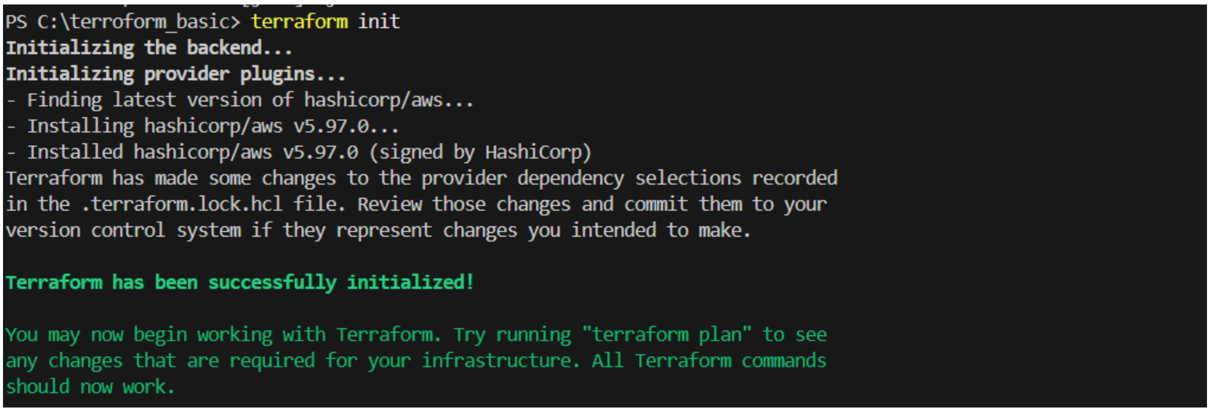
```
resource "aws_instance" "webserver" {
  ami           = var.ami[terraform.workspace]
  instance_type = "t2.micro"
}
```

--create variable.tf



```
variable "ami" {
type = map
default = {
  "Dev" = "ami-058a8a5ab36292159"
  "Staging" = "ami-058a8a5ab36292159"
  "Prod" = "ami-058a8a5ab36292159"
}
}
```

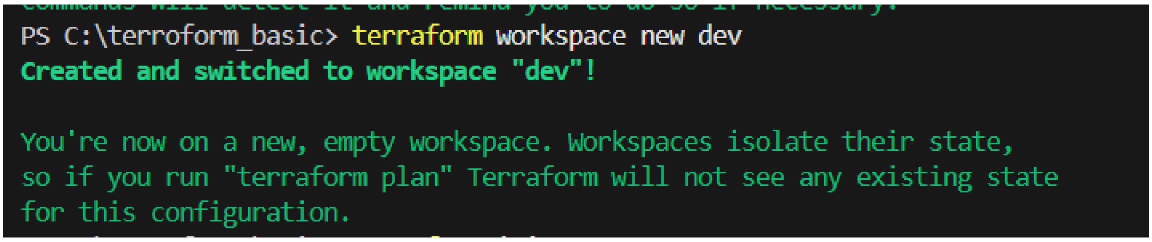
--do terraform init



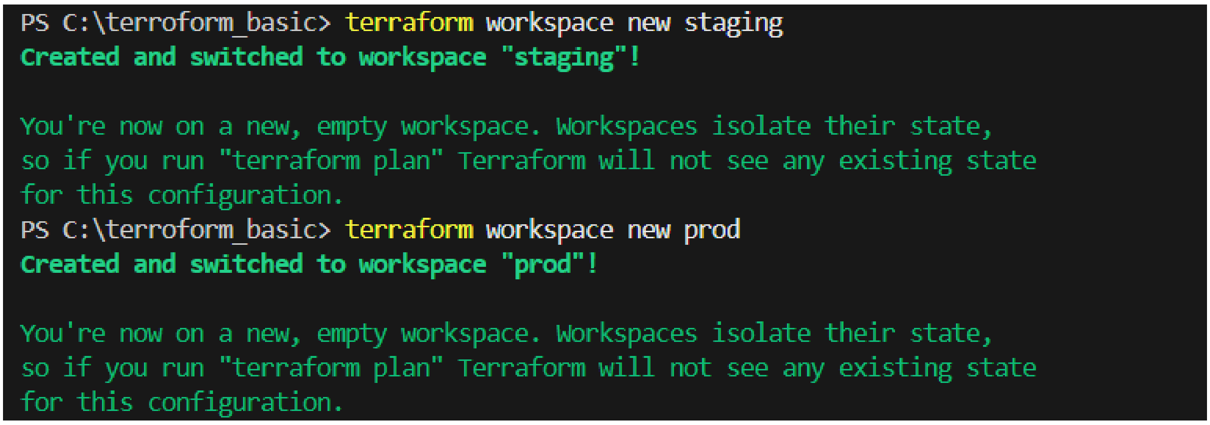
```
PS C:\terroform_basic> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.97.0...
- Installed hashicorp/aws v5.97.0 (signed by HashiCorp)
Terraform has made some changes to the provider dependency selections recorded
in the .terraform.lock.hcl file. Review those changes and commit them to your
version control system if they represent changes you intended to make.

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.
```

--create workspaces (dev staging prod) using below command

Terraform workspace new workspacename



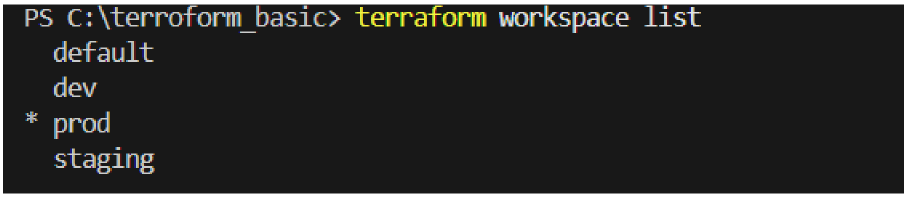
```
commands will detect it and remind you to do so if necessary.
PS C:\terroform_basic> terraform workspace new dev
Created and switched to workspace "dev"!

You're now on a new, empty workspace. Workspaces isolate their state,
so if you run "terraform plan" Terraform will not see any existing state
for this configuration.
```



```
PS C:\terroform_basic> terraform workspace new staging
Created and switched to workspace "staging"!

You're now on a new, empty workspace. Workspaces isolate their state,
so if you run "terraform plan" Terraform will not see any existing state
for this configuration.
PS C:\terroform_basic> terraform workspace new prod
Created and switched to workspace "prod"!

You're now on a new, empty workspace. Workspaces isolate their state,
so if you run "terraform plan" Terraform will not see any existing state
for this configuration.
```

--check with below command

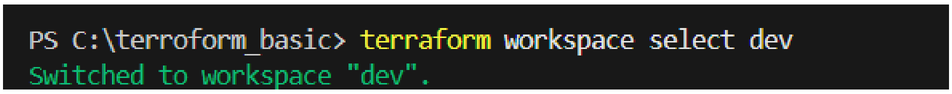
terraform workspace list



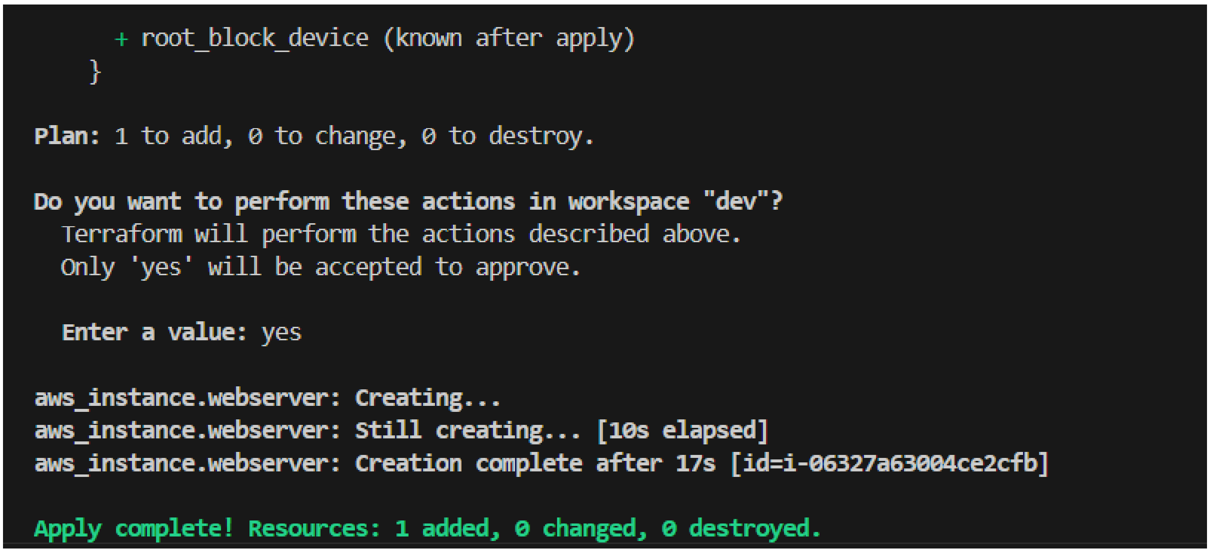
```
PS C:\terroform_basic> terraform workspace list
  default
  dev
* prod
  staging
```

--now switch to each environment and do apply

dev

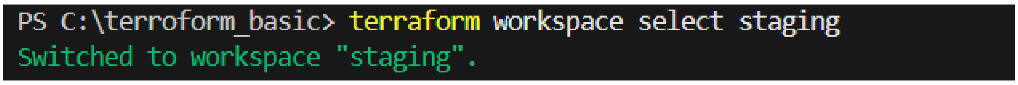


```
PS C:\terroform_basic> terraform workspace select dev
Switched to workspace "dev".
```

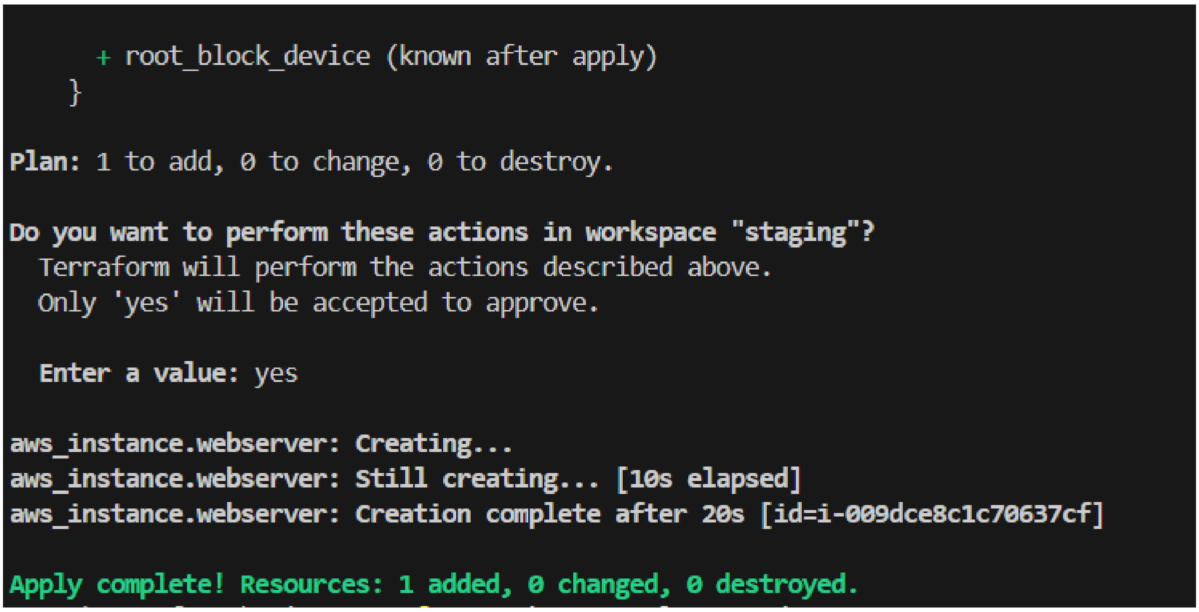


```
      + root_block_device (known after apply)
    }

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

Do you want to perform these actions in workspace "dev"?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_instance.webserver: Creating...
aws_instance.webserver: Still creating... [10s elapsed]
aws_instance.webserver: Creation complete after 17s [id=i-06327a63004ce2cfb]

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

staging

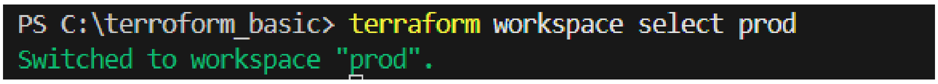


```
PS C:\terroform_basic> terraform workspace select staging
Switched to workspace "staging".
```

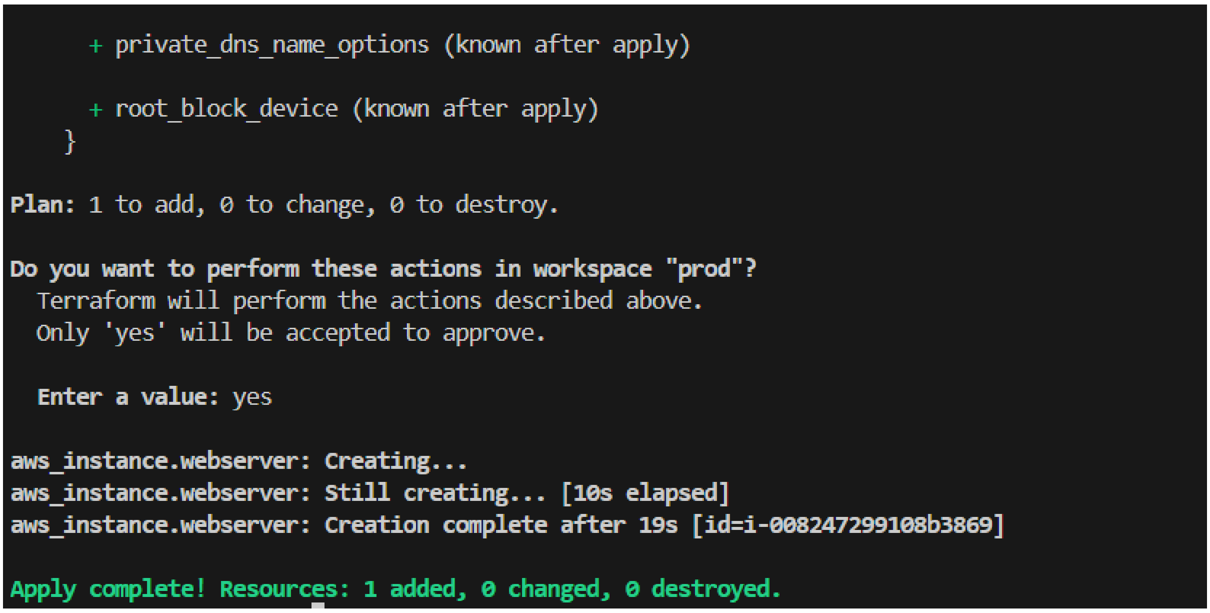


```
      + root_block_device (known after apply)
    }

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

Do you want to perform these actions in workspace "staging"?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_instance.webserver: Creating...
aws_instance.webserver: Still creating... [10s elapsed]
aws_instance.webserver: Creation complete after 20s [id=i-009dce8c1c70637cf]

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

prod

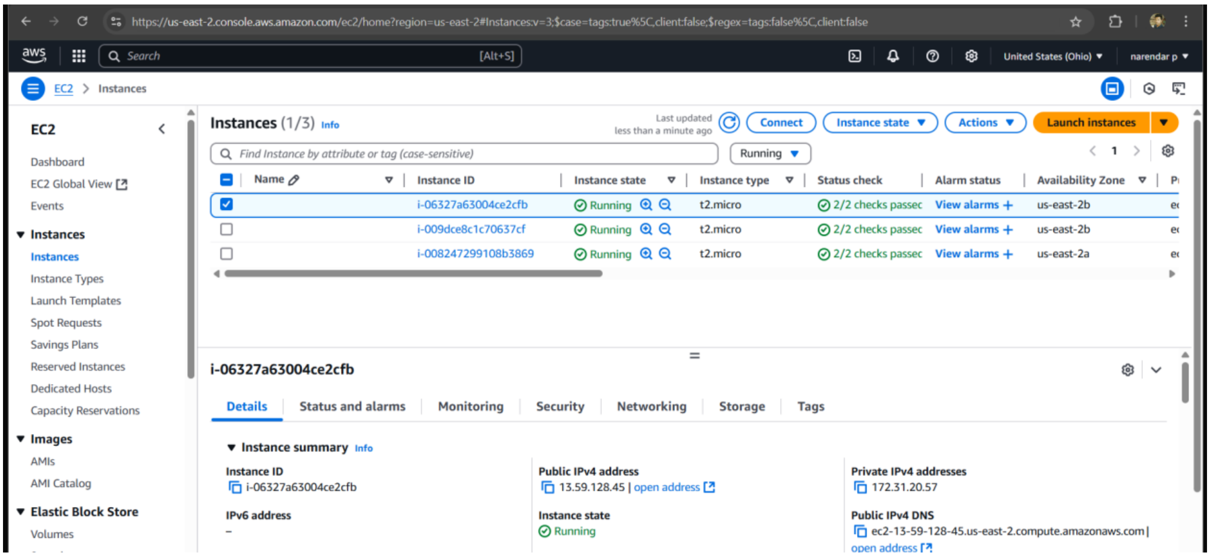


```
PS C:\terroform_basic> terraform workspace select prod
Switched to workspace "prod".
```



```
      + private_dns_name_options (known after apply)

      + root_block_device (known after apply)
    }

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

Do you want to perform these actions in workspace "prod"?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_instance.webserver: Creating...
aws_instance.webserver: Still creating... [10s elapsed]
aws_instance.webserver: Creation complete after 19s [id=i-008247299108b3869]

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

--check aws now 3 ec2s are running or not



https://us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#Instances:v=3;$case=tags:true%5C,client:false;$regex=tags:false%5C,client:false

Instances (1/3) Info

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	P
☑		i-06327a63004ce2cfb	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	us-east-2b	e
□		i-009dce8c1c70637cf	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	us-east-2b	e
□		i-008247299108b3869	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	us-east-2a	e

i-06327a63004ce2cfb

▼ Instance summary Info

Instance ID
i-06327a63004ce2cfb

Public IPv4 address
13.59.128.45 | open address

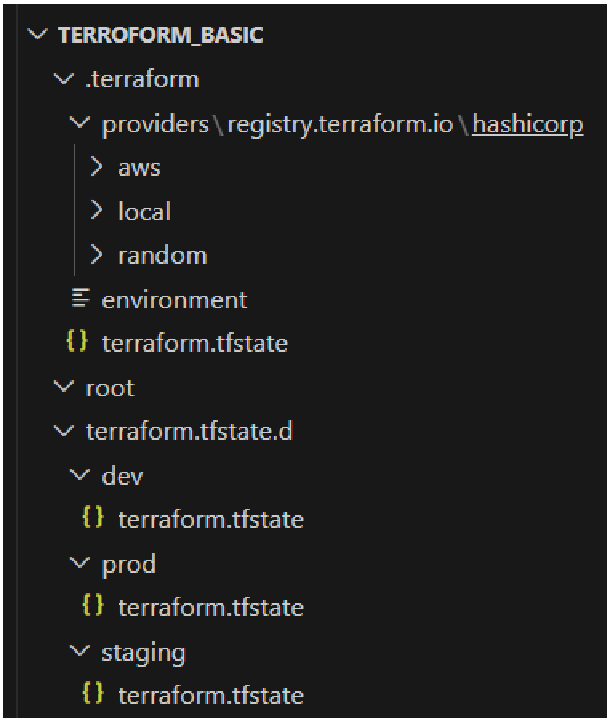
Private IPv4 addresses
172.31.20.57

IPv6 address
–

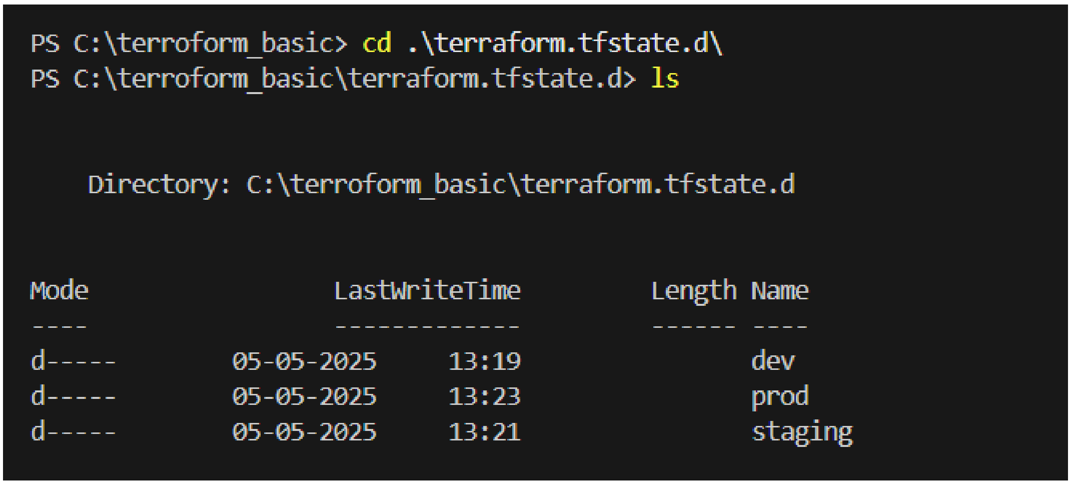
Instance state
✓ Running

Public IPv4 DNS
ec2-13-59-128-45.us-east-2.compute.amazonaws.com | open address

--check terraform.tfstate.d



```
TERROFORM_BASIC
  .terraform
    providers\registry.terraform.io\hashicorp
      aws
      local
      random
    environment
    terraform.tfstate
  root
  terraform.tfstate.d
    dev
      terraform.tfstate
    prod
      terraform.tfstate
    staging
      terraform.tfstate
```



```
PS C:\terroform_basic> cd .\terraform.tfstate.d\
PS C:\terroform_basic\terraform.tfstate.d> ls


    Directory: C:\terroform_basic\terraform.tfstate.d


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        05-05-2025     13:19                dev
d-----        05-05-2025     13:23                prod
d-----        05-05-2025     13:21                staging
```