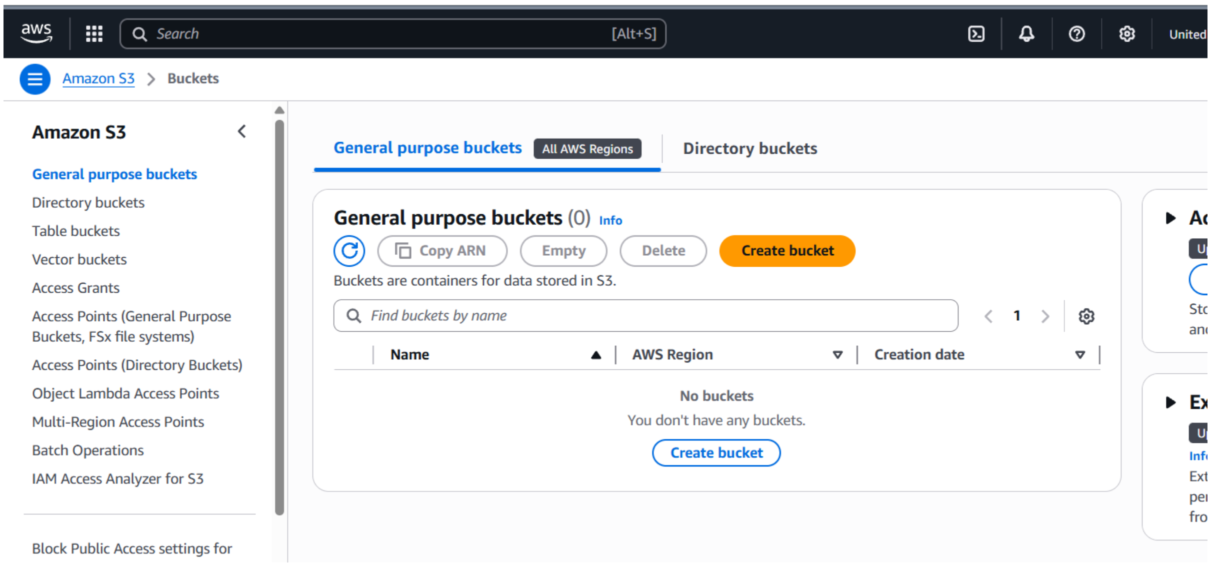
1. **Watch the Terraform-05 video.**
2. **Execute the Script Shown in the Video**

Create a s3 bucket by this script.

```
resource "aws_s3_bucket" "s3_bucket" {


  bucket = "s3backend2345678"

  acl = "private"

}
```


aws
Search [Alt+S]
United
Amazon S3 > Buckets
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Vector buckets
Access Grants
Access Points (General Purpose Buckets, FSx file systems)
Access Points (Directory Buckets)
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for
General purpose buckets All AWS Regions
Directory buckets
Info
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
1

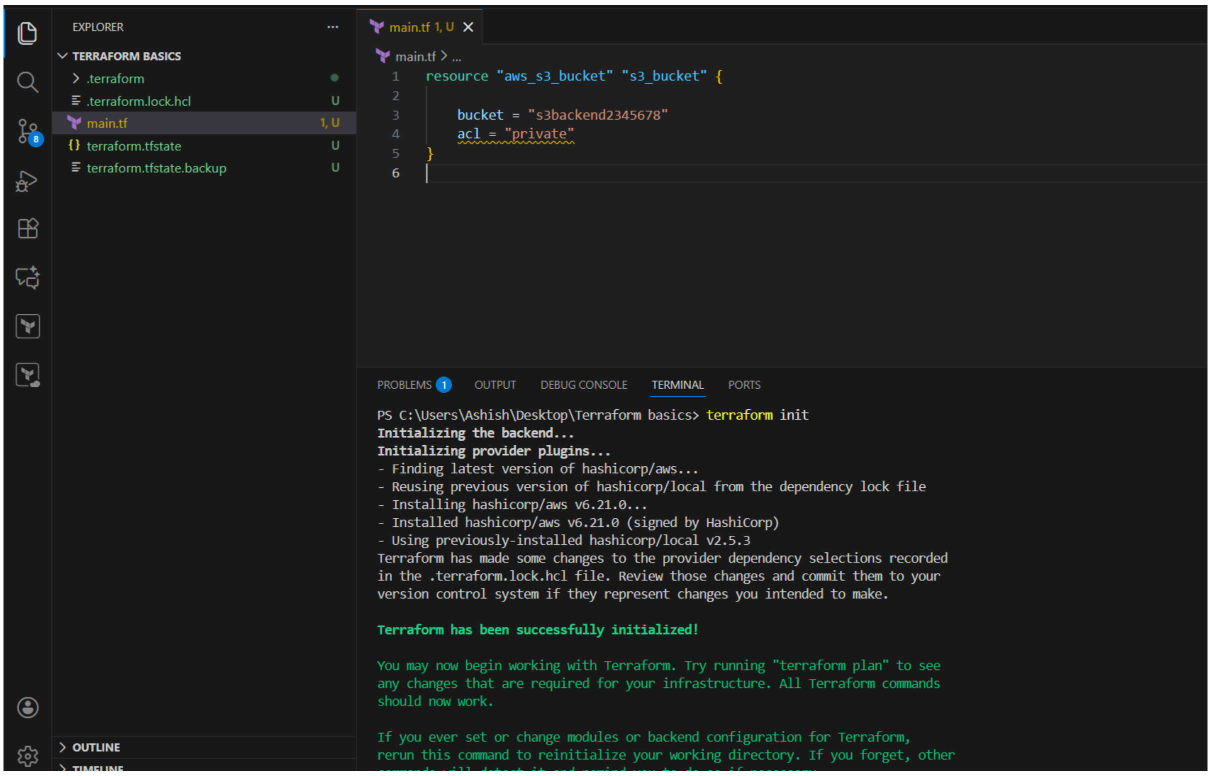
General purpose buckets (0)

Buckets are containers for data stored in S3.

Name	AWS Region	Creation date

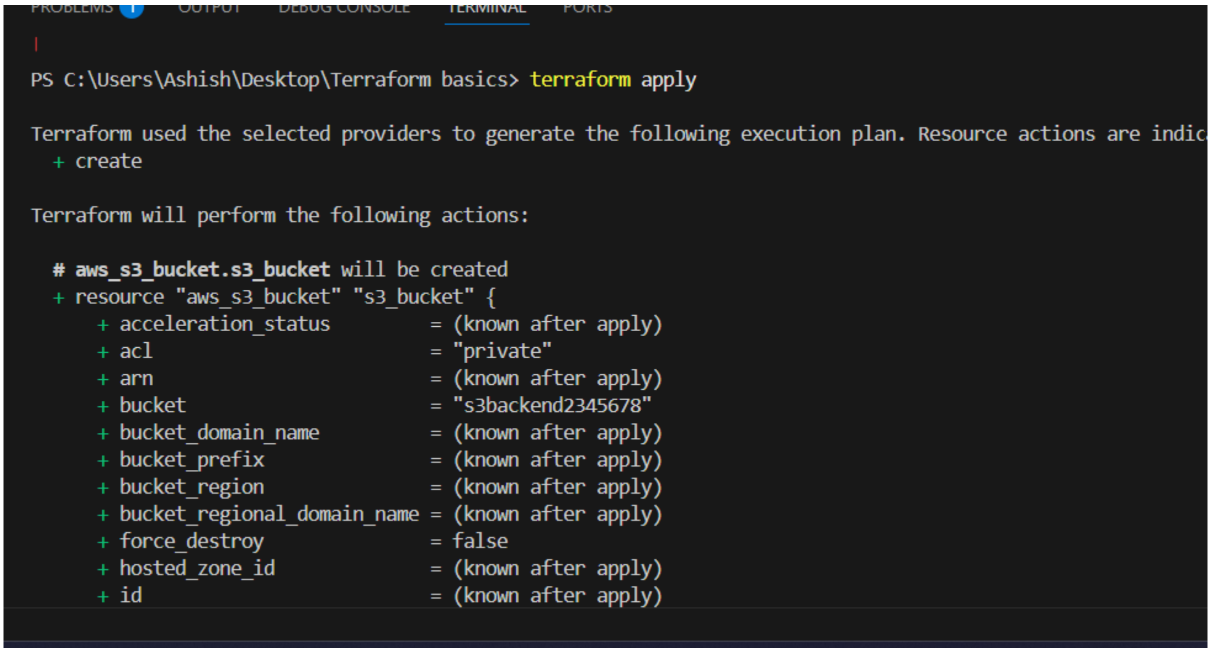
No buckets

You don't have any buckets.

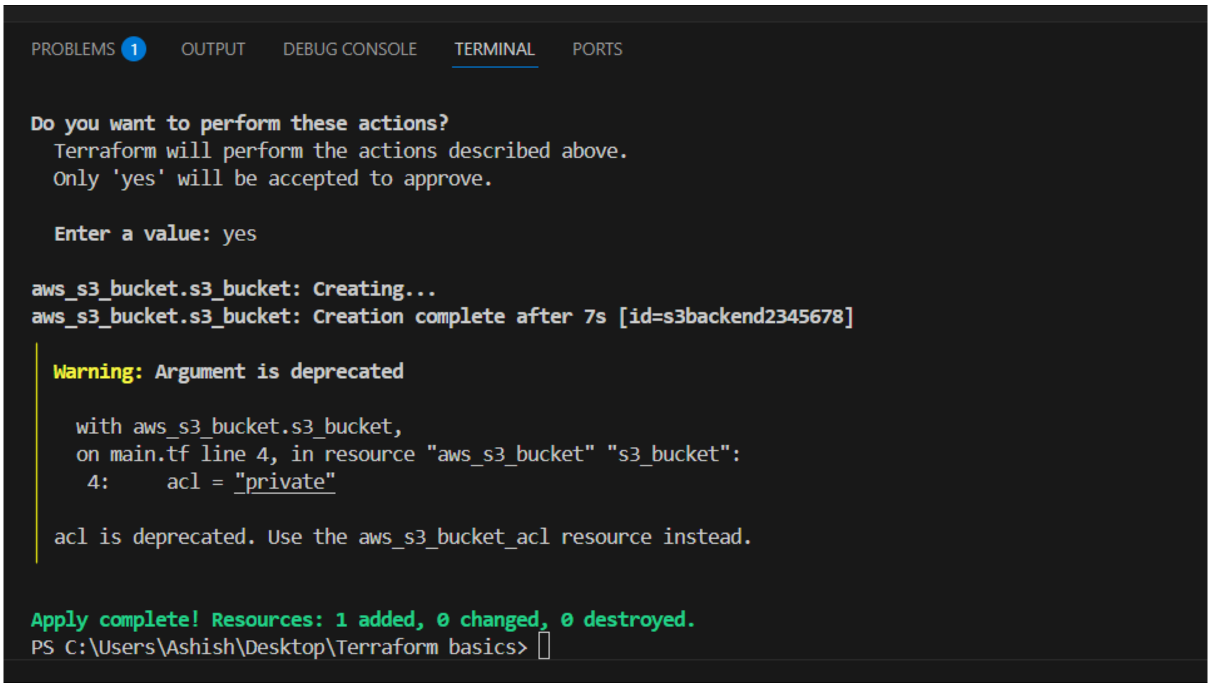


```
resource "aws_s3_bucket" "s3_bucket" {

    bucket = "s3backend2345678"
    acl = "private"
}

```

```
PS C:\Users\Ashish\Desktop\Terraform basics> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Reusing previous version of hashicorp/local from the dependency lock file
- Installing hashicorp/aws v6.21.0...
- Installed hashicorp/aws v6.21.0 (signed by HashiCorp)
- Using previously-installed hashicorp/local v2.5.3
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.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
```

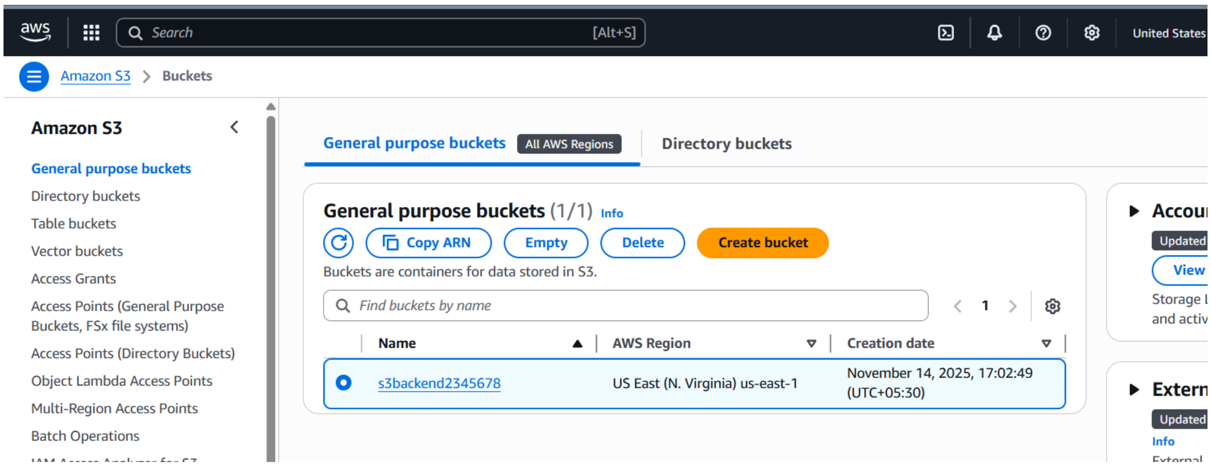


```
PS C:\Users\Ashish\Desktop\Terraform basics> terraform apply

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

Terraform will perform the following actions:

  # aws_s3_bucket.s3_bucket will be created
  + resource "aws_s3_bucket" "s3_bucket" {
      + acceleration_status         = (known after apply)
      + acl                         = "private"
      + arn                         = (known after apply)
      + bucket                      = "s3backend2345678"
      + bucket_domain_name          = (known after apply)
      + bucket_prefix               = (known after apply)
      + bucket_region               = (known after apply)
      + bucket_regional_domain_name = (known after apply)
      + force_destroy               = false
      + hosted_zone_id              = (known after apply)
      + id                          = (known after apply)
```



```
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: Creating...
aws_s3_bucket.s3_bucket: Creation complete after 7s [id=s3backend2345678]

  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:\Users\Ashish\Desktop\Terraform basics>
```

A bucket has been created.



General purpose buckets (1/1)

Buckets are containers for data stored in S3.

Name	AWS Region	Creation date
s3backend2345678	US East (N. Virginia) us-east-1	November 14, 2025, 17:02:49 (UTC+05:30)

Create a dynamodb with this script.

resource "aws_dynamodb_table" "dynamodb-terraform-state-lock" {

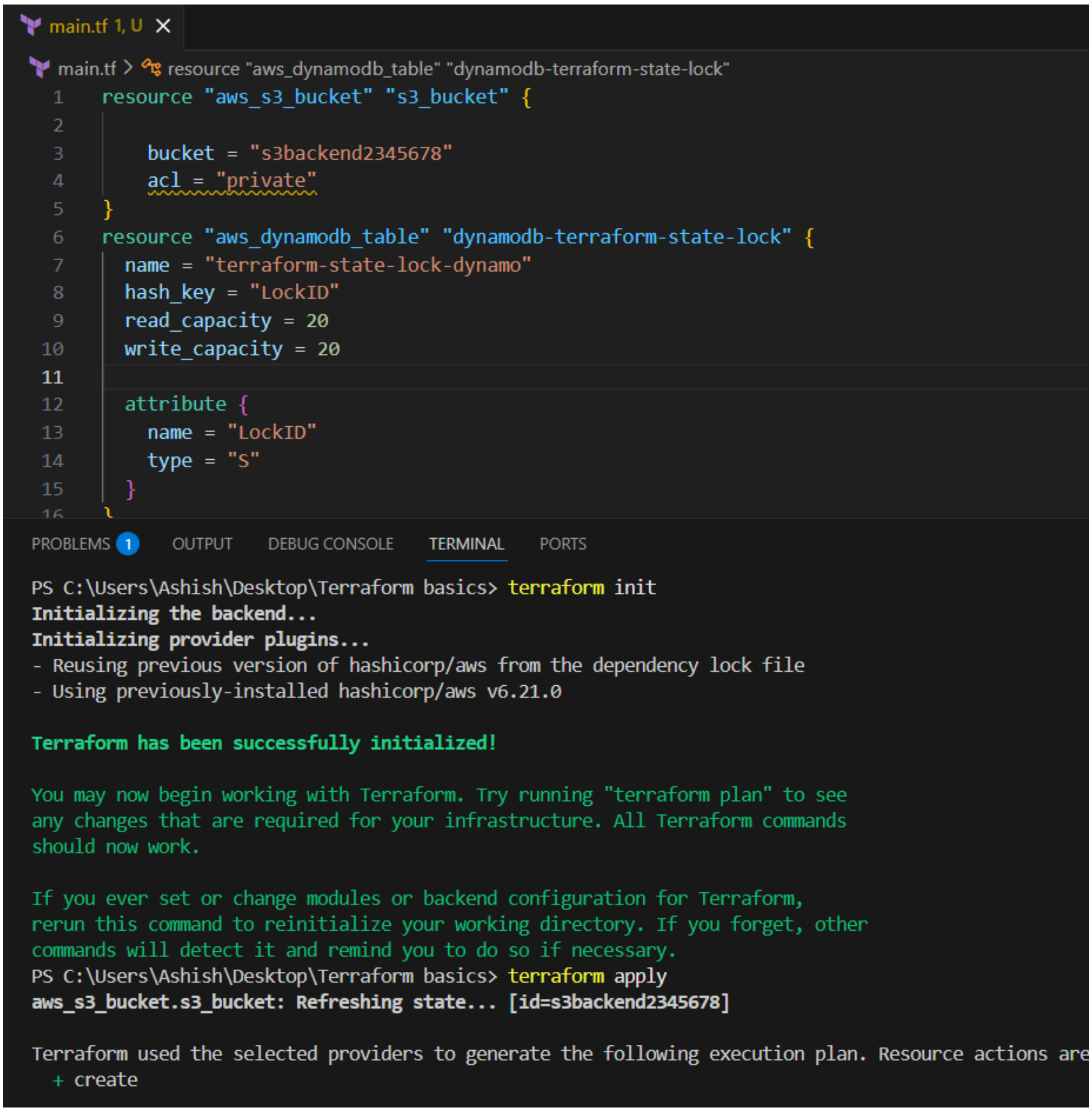
name = "terraform-state-lock-dynamo"

hash_key = "LockID"

read_capacity = 20

write_capacity = 20

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



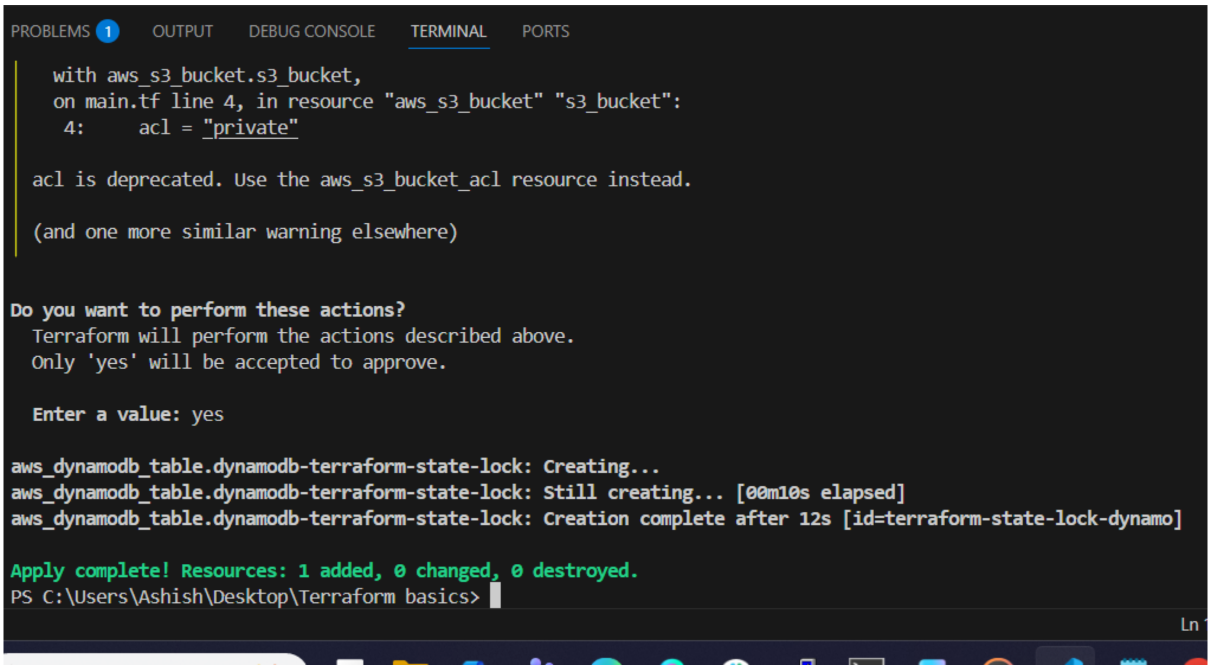
main.tf 1, U

main.tf > resource "aws_dynamodb_table" "dynamodb-terraform-state-lock"

```
resource "aws_s3_bucket" "s3_bucket" {

    bucket = "s3backend2345678"
    acl = "private"
}
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"
  }
}
```

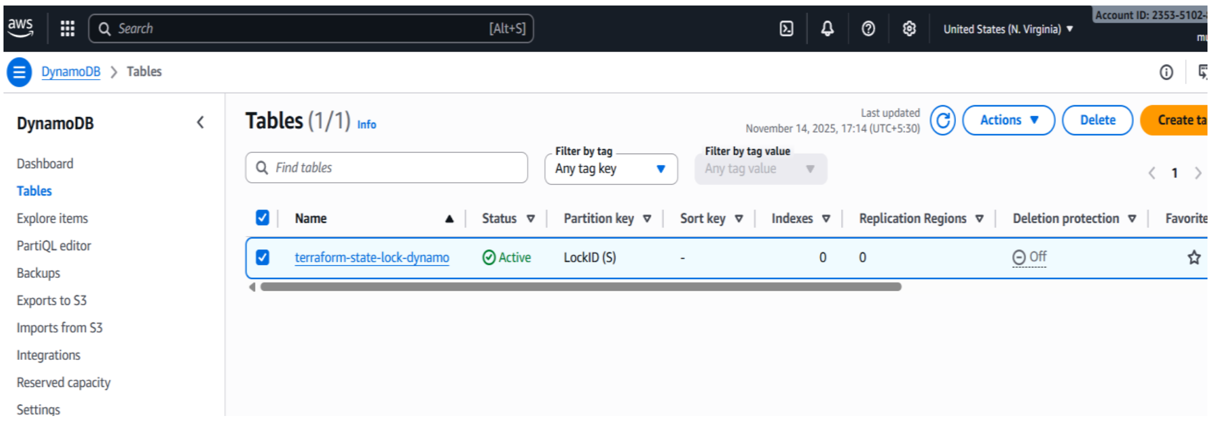
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Ashish\Desktop\Terraform basics> terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v6.21.0

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.
PS C:\Users\Ashish\Desktop\Terraform basics> terraform apply
aws_s3_bucket.s3_bucket: Refreshing state... [id=s3backend2345678]

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



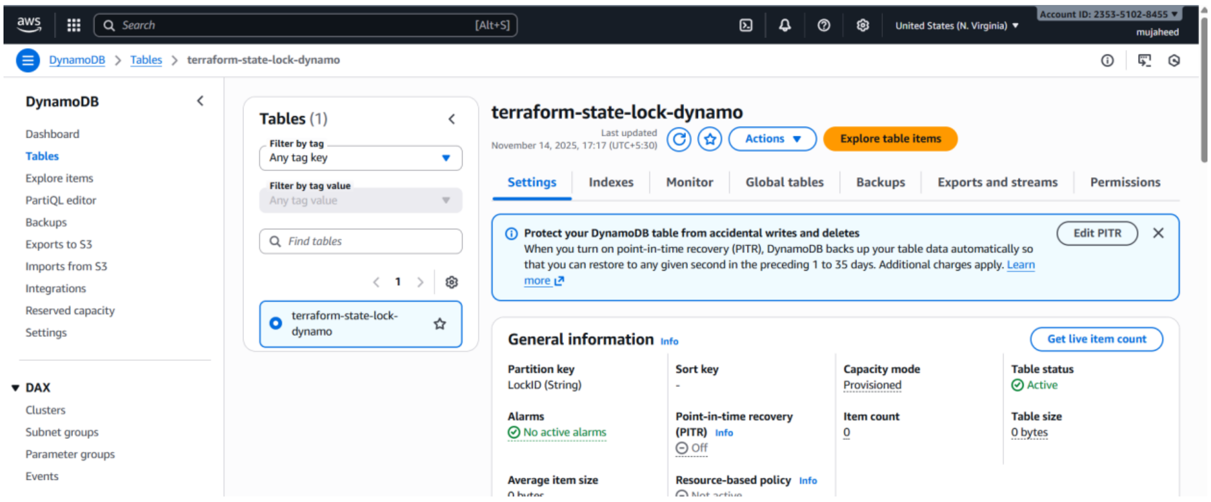
```
    with aws_s3_bucket.s3_bucket,
    on main.tf line 4, in resource "aws_s3_bucket" "s3_bucket":
     4:     acl = "private"

  acl is deprecated. Use the aws_s3_bucket_acl resource instead.

  (and one more similar warning elsewhere)


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... [00m10s elapsed]
aws_dynamodb_table.dynamodb-terraform-state-lock: Creation complete after 12s [id=terraform-state-lock-dynamo]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\Users\Ashish\Desktop\Terraform basics>
```

You can see a table has been created in dynamodb



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

An empty table has been created.



terraform-state-lock-dynamo

Protect your DynamoDB table from accidental writes and deletes
When you turn on point-in-time recovery (PITR), DynamoDB backs up your table data automatically so that you can restore to any given second in the preceding 1 to 35 days. Additional charges apply. Learn more

General information Info

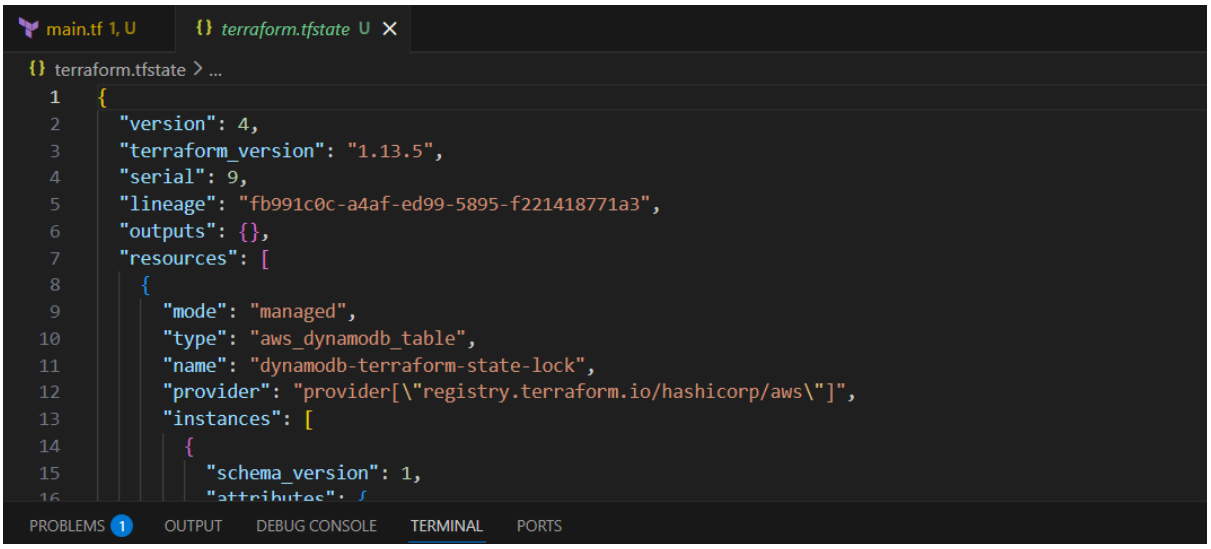
Partition key: LockID (String)
Sort key: -
Capacity mode: Provisioned
Table status: Active
Alarms: No active alarms
Point-in-time recovery (PITR) Info: Off
Item count: 0
Table size: 0 bytes
Average item size: 0 bytes
Resource-based policy Info: Not active

Giving s3 as backend for terraform.tf state file.

Giving this script.

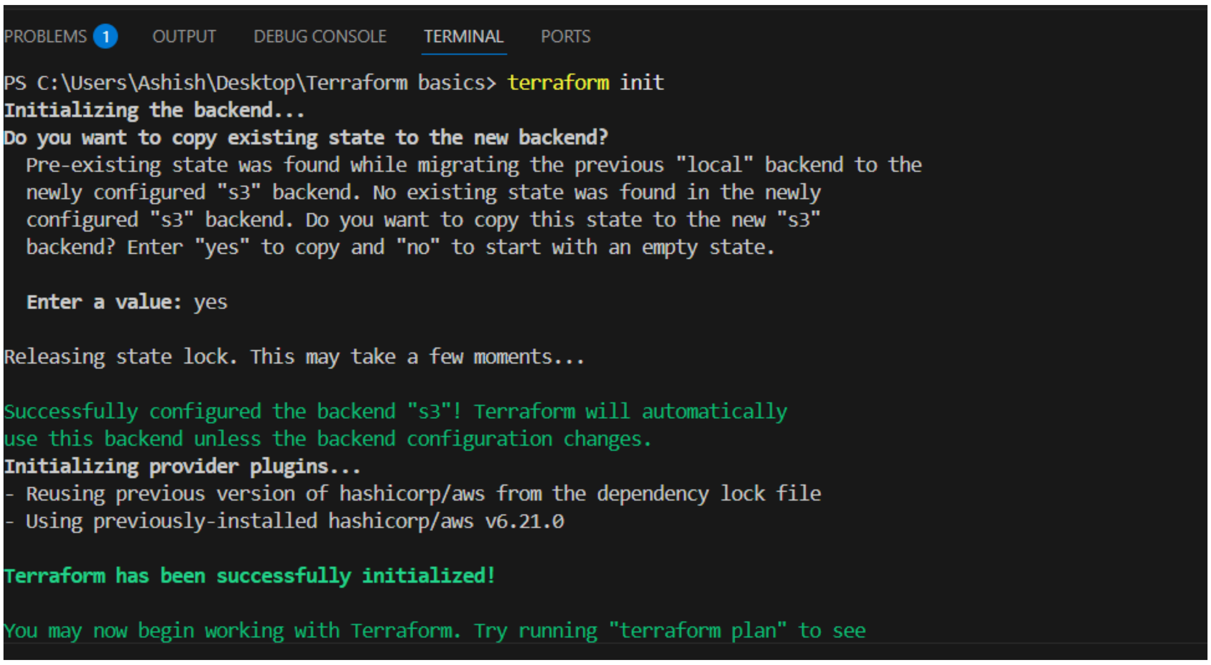
```
terraform {
  backend "s3" {
    bucket = "s3backend2345678"
    dynamodb_table = "terraform-state-lock-dynamo"
    key    = "terraform.tfstate"
    region = "us-east-1"
  }
}
```

If you open terraform state file you can see your content.



```
{
  "version": 4,
  "terraform_version": "1.13.5",
  "serial": 9,
  "lineage": "fb991c0c-a4af-ed99-5895-f221418771a3",
  "outputs": {},
  "resources": [
    {
      "mode": "managed",
      "type": "aws_dynamodb_table",
      "name": "dynamodb-terraform-state-lock",
      "provider": "provider[\"registry.terraform.io/hashicorp/aws\"]",
      "instances": [
        {
          "schema_version": 1,
          "attributes": {
```

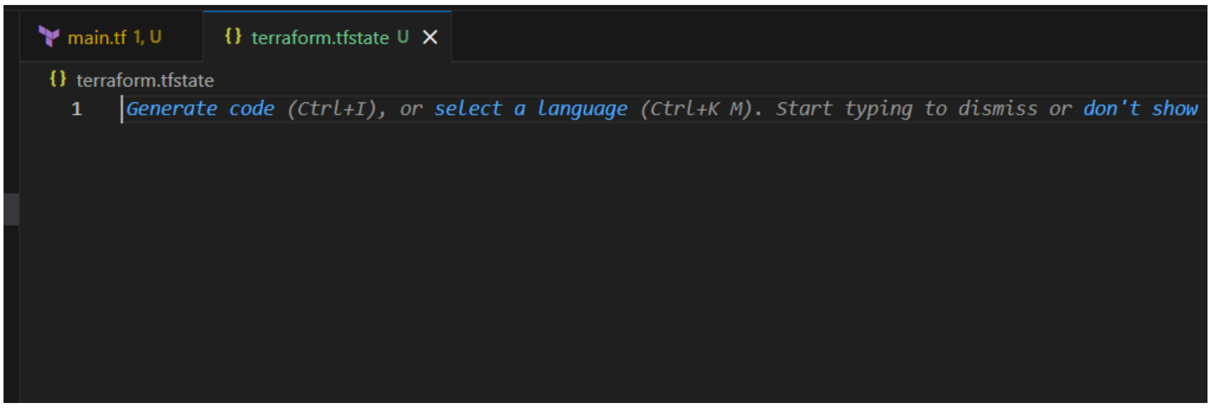
After initializing you can't see your content.



```
PS C:\Users\Ashish\Desktop\Terraform basics> terraform init
Initializing the backend...
Do you want to copy existing state to the new backend?
  Pre-existing state was found while migrating the previous "local" backend to the
  newly configured "s3" backend. No existing state was found in the newly
  configured "s3" backend. Do you want to copy this state to the new "s3"
  backend? Enter "yes" to copy and "no" to start with an empty state.

  Enter a value: yes

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

Successfully configured the backend "s3"! Terraform will automatically
use this backend unless the backend configuration changes.
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v6.21.0

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
```

If you open statefile.

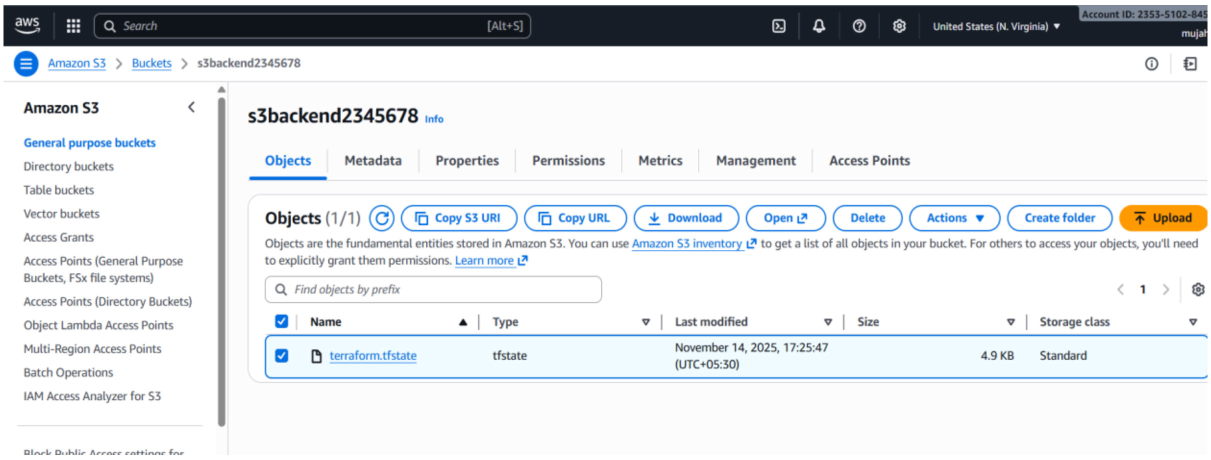
It will be empty.



```
Generate code (Ctrl+I), or select a language (Ctrl+K M). Start typing to dismiss or don't show
```

It will be stored in s3bucket go to your s3bucket and click on objects.

Our terraform statefile will be stored in s3.


Amazon S3 > Buckets > s3backend2345678

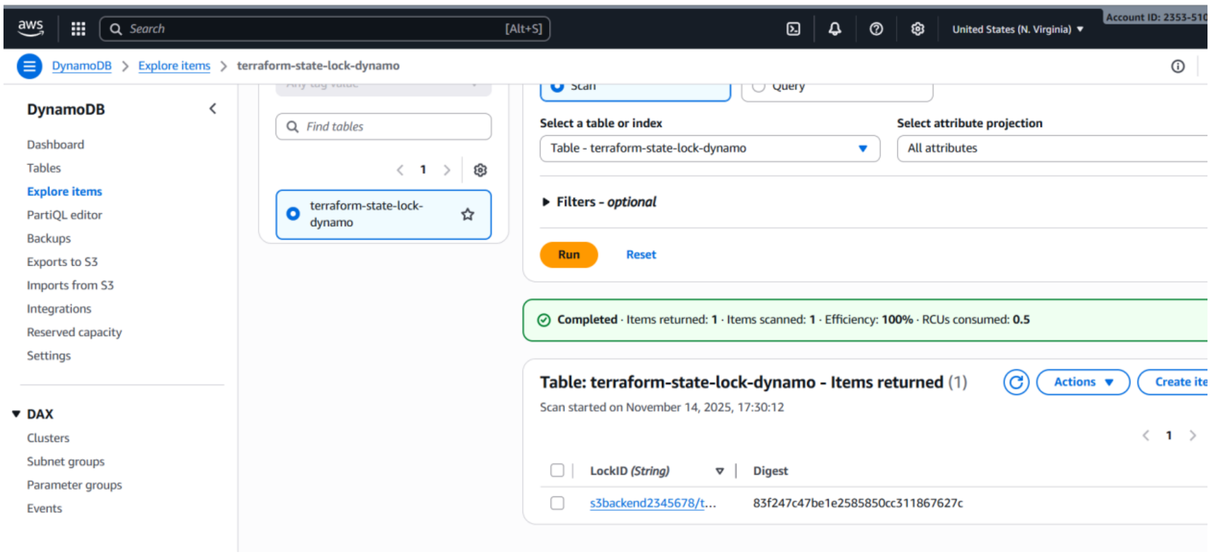
s3backend2345678 Info

Objects (1/1)

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	November 14, 2025, 17:25:47 (UTC+05:30)	4.9 KB	Standard

If you go to dynamodb,explore items you willsee terraform statefile will be stored in your table.


DynamoDB > Explore items > terraform-state-lock-dynamo

Select a table or index: Table - terraform-state-lock-dynamo

Select attribute projection: All attributes

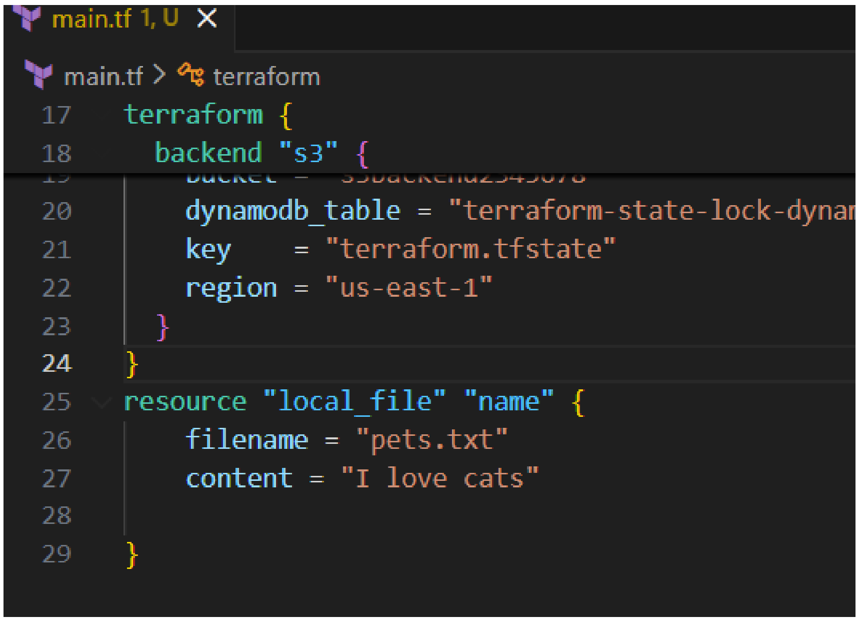
Completed · Items returned: 1 · Items scanned: 1 · Efficiency: 100% · RCUs consumed: 0.5

Table: terraform-state-lock-dynamo - Items returned (1)

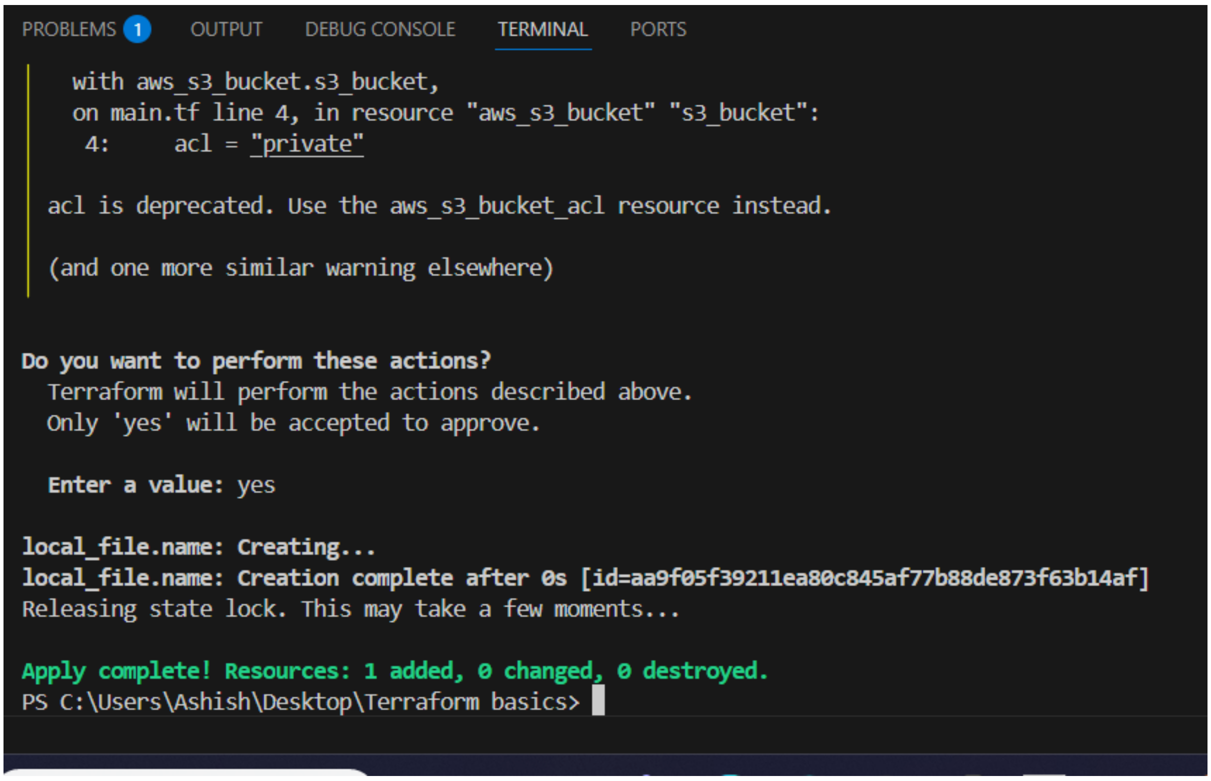
Scan started on November 14, 2025, 17:30:12

LockID (String)	Digest
s3backend2345678/t...	83f247c47be1e2585850cc311867627c

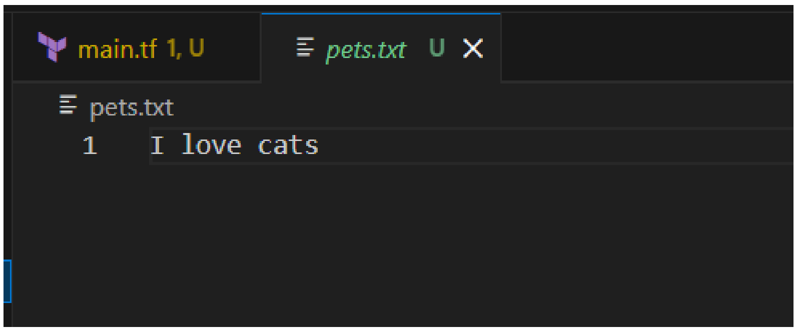
If you add another resource



```
terraform {
  backend "s3" {
    dynamodb_table = "terraform-state-lock-dynar
    key     = "terraform.tfstate"
    region = "us-east-1"
  }
}
resource "local_file" "name" {
    filename = "pets.txt"
    content = "I love cats"

}
```

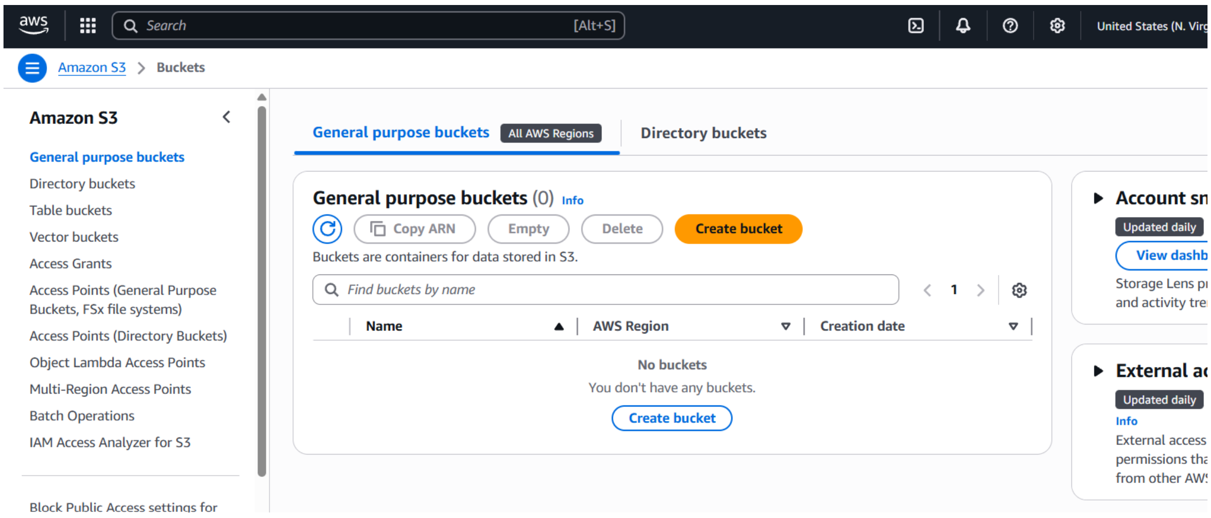


```
    with aws_s3_bucket.s3_bucket,
    on main.tf line 4, in resource "aws_s3_bucket" "s3_bucket":
     4:     acl = "private"

  acl is deprecated. Use the aws_s3_bucket_acl resource instead.

  (and one more similar warning elsewhere)


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

local_file.name: Creating...
local_file.name: Creation complete after 0s [id=aa9f05f39211ea80c845af77b88de873f63b14af]
Releasing state lock. This may take a few moments...

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\Users\Ashish\Desktop\Terraform basics>
```



```
I love cats
```

terraform destroy it will delete all s3bucket,dynamodb.



Buckets are containers for data stored in S3.

No buckets

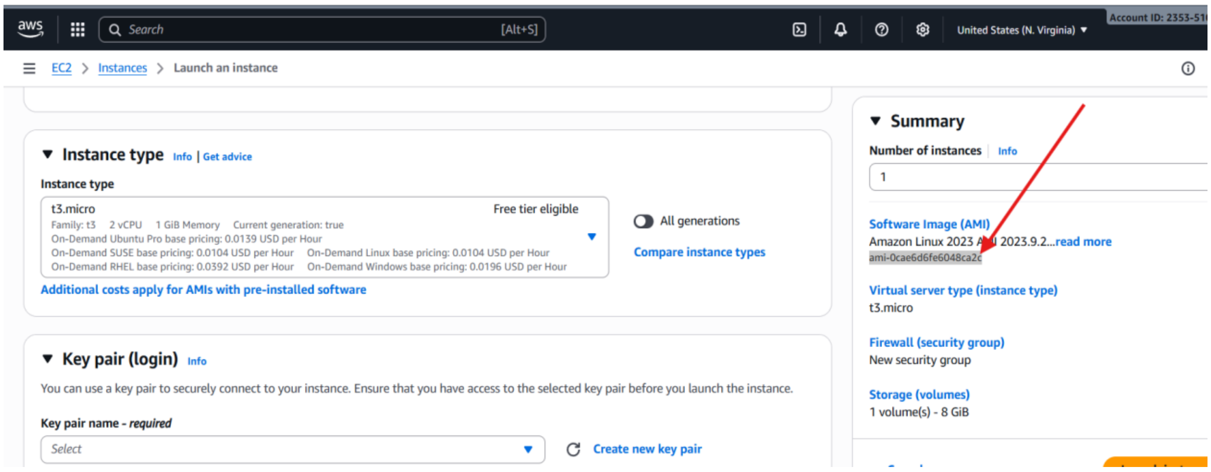
You don't have any buckets.

Storage Lens pr and activity tre

External access permissions tha from other AWS

Creating an ec2 instance with terraform.

Go to e2-instance copy the ami id



t3.micro Free tier eligible
Family: t3 2 vCPU 1 GiB Memory Current generation: true
On-Demand Ubuntu Pro base pricing: 0.0139 USD per Hour
On-Demand SUSE base pricing: 0.0104 USD per Hour On-Demand Linux base pricing: 0.0104 USD per Hour
On-Demand RHEL base pricing: 0.0392 USD per Hour On-Demand Windows base pricing: 0.0196 USD per Hour

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Number of instances: 1

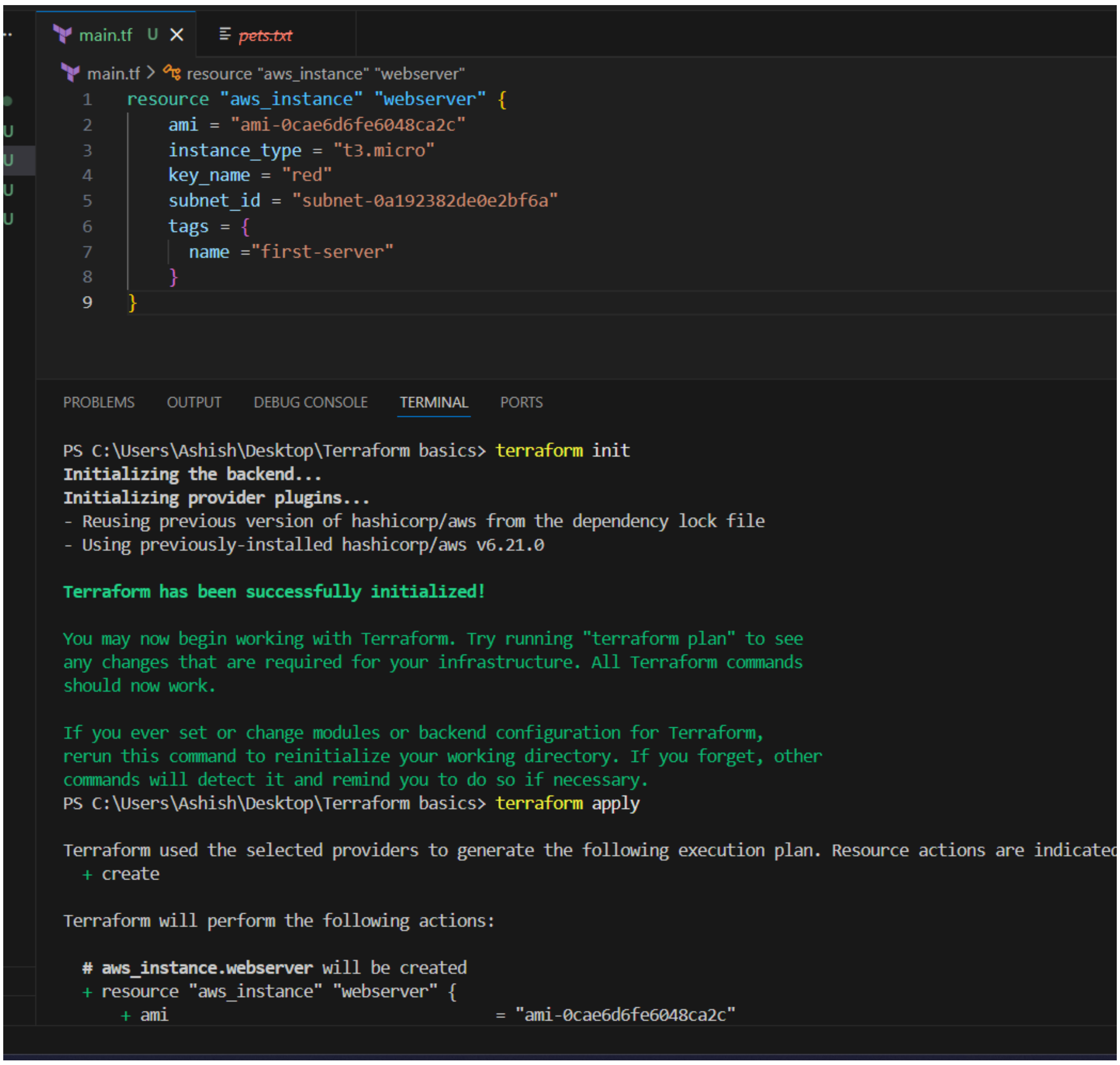
Software Image (AMI): Amazon Linux 2023 AMI 2023.9.2...read more
ami-0cae6d6fe6048ca2c

Virtual server type (instance type): t3.micro

Firewall (security group): New security group

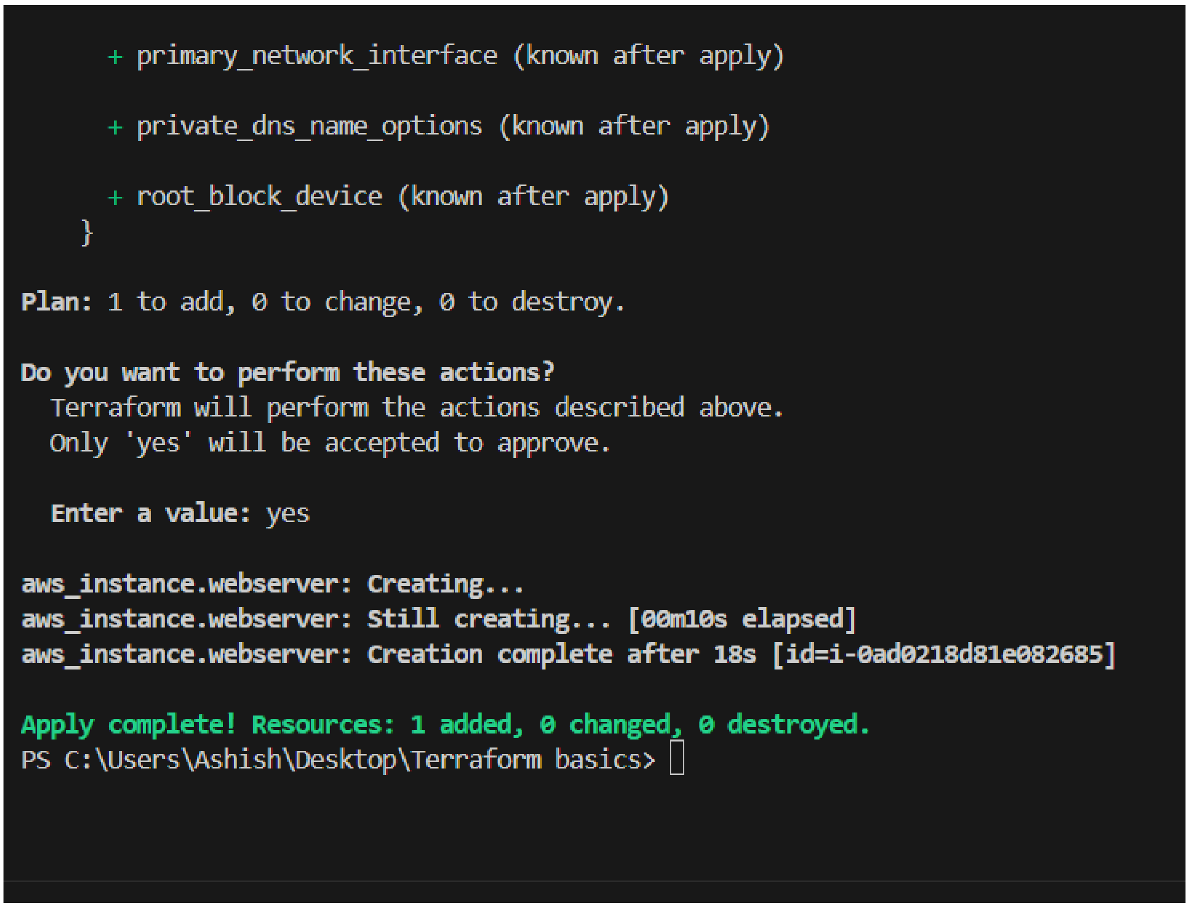
Storage (volumes): 1 volume(s) - 8 GiB

```
resource "aws_instance" "webserver" {

  ami = "ami-0cae6d6fe6048ca2c"

  instance_type = "t3.micro"

  key_name = "red"

  subnet_id = "subnet-0a192382de0e2bf6a"

  tags = {

    name ="first-server"
    }
}
```



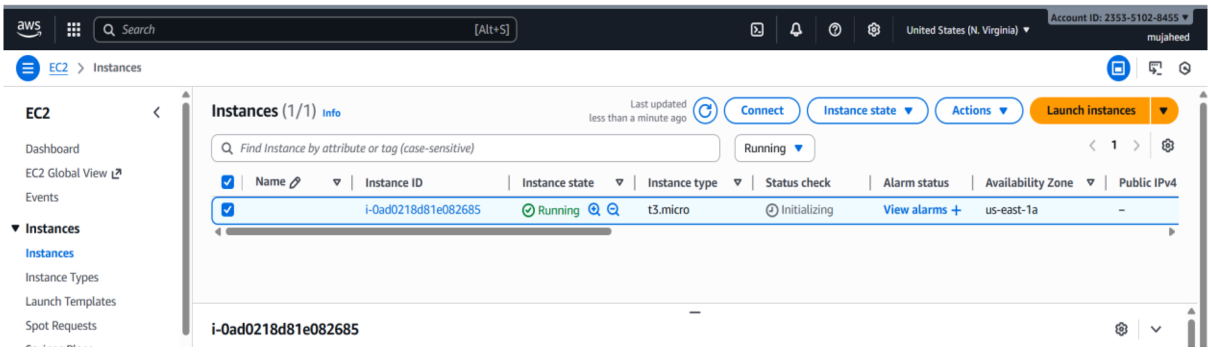
```
resource "aws_instance" "webserver" {
    ami = "ami-0cae6d6fe6048ca2c"
    instance_type = "t3.micro"
    key_name = "red"
    subnet_id = "subnet-0a192382de0e2bf6a"
    tags = {
      name ="first-server"
    }
}
```

```
PS C:\Users\Ashish\Desktop\Terraform basics> terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v6.21.0

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.
PS C:\Users\Ashish\Desktop\Terraform basics> terraform apply

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

Terraform will perform the following actions:

  # aws_instance.webserver will be created
  + resource "aws_instance" "webserver" {
      + ami                                   = "ami-0cae6d6fe6048ca2c"
```



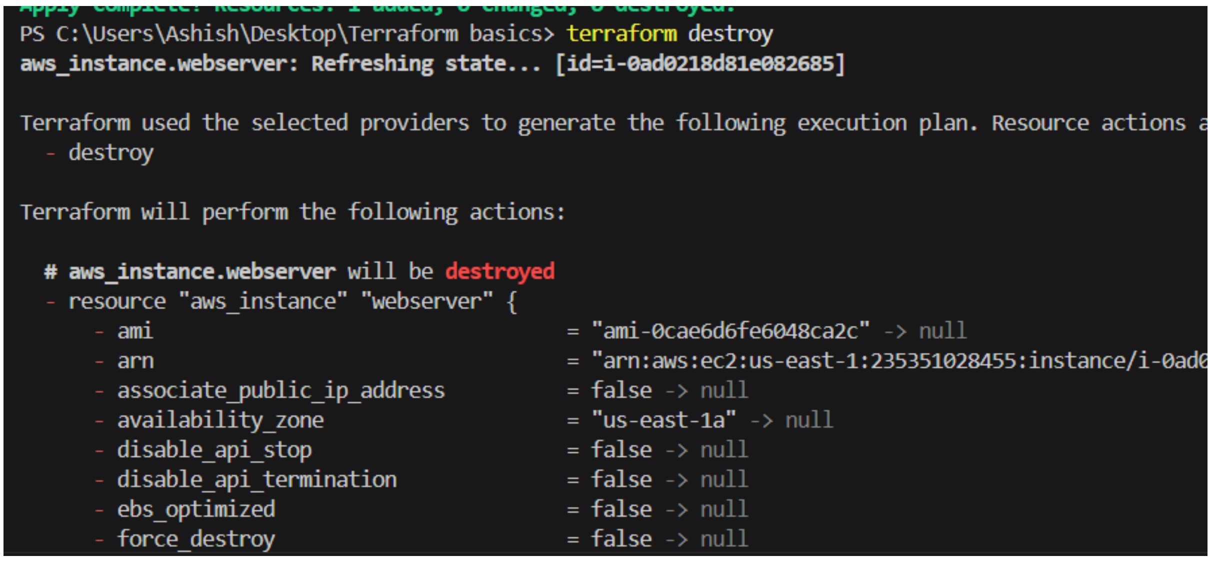
```
      + primary_network_interface (known after apply)

      + 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?
  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... [00m10s elapsed]
aws_instance.webserver: Creation complete after 18s [id=i-0ad0218d81e082685]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\Users\Ashish\Desktop\Terraform basics>
```

An ec2 instance has been created here.

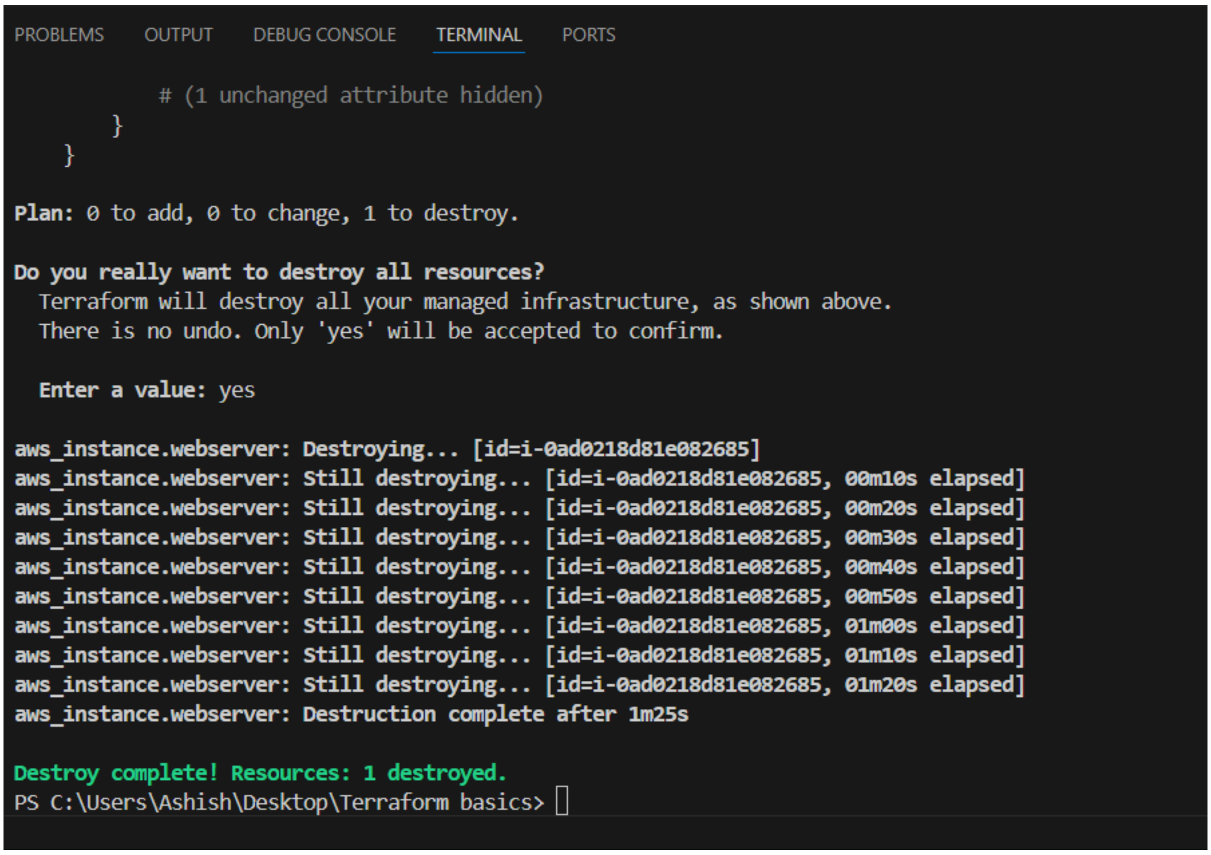


	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
☑		i-0ad0218d81e082685	Running	t3.micro	Initializing	View alarms +	us-east-1a	–

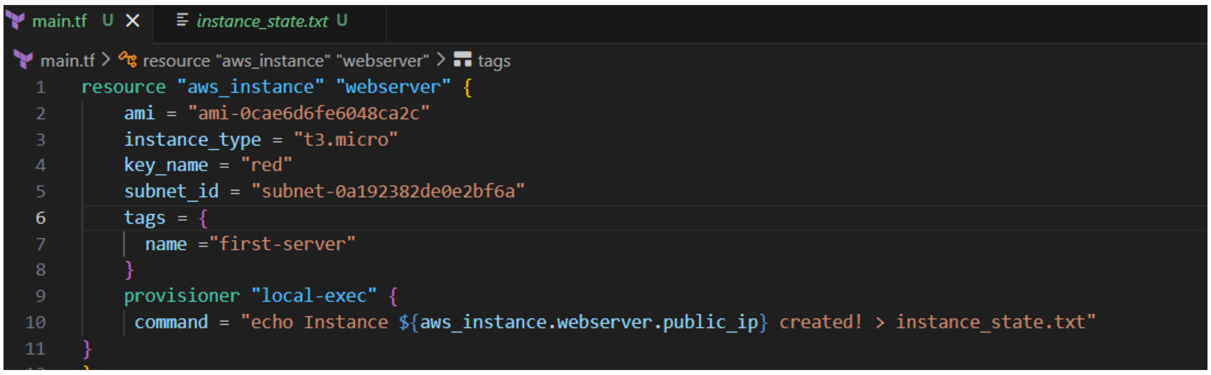
i-0ad0218d81e082685



```
PS C:\Users\Ashish\Desktop\Terraform basics> terraform destroy
aws_instance.webserver: Refreshing state... [id=i-0ad0218d81e082685]

Terraform used the selected providers to generate the following execution plan. Resource actions a
  - destroy

Terraform will perform the following actions:

  # aws_instance.webserver will be destroyed
  - resource "aws_instance" "webserver" {
      - ami                                  = "ami-0cae6d6fe6048ca2c" -> null
      - arn                                  = "arn:aws:ec2:us-east-1:235351028455:instance/i-0ad0
      - associate_public_ip_address          = false -> null
      - availability_zone                    = "us-east-1a" -> null
      - disable_api_stop                     = false -> null
      - disable_api_termination              = false -> null
      - ebs_optimized                        = false -> null
      - force_destroy                        = false -> null
```

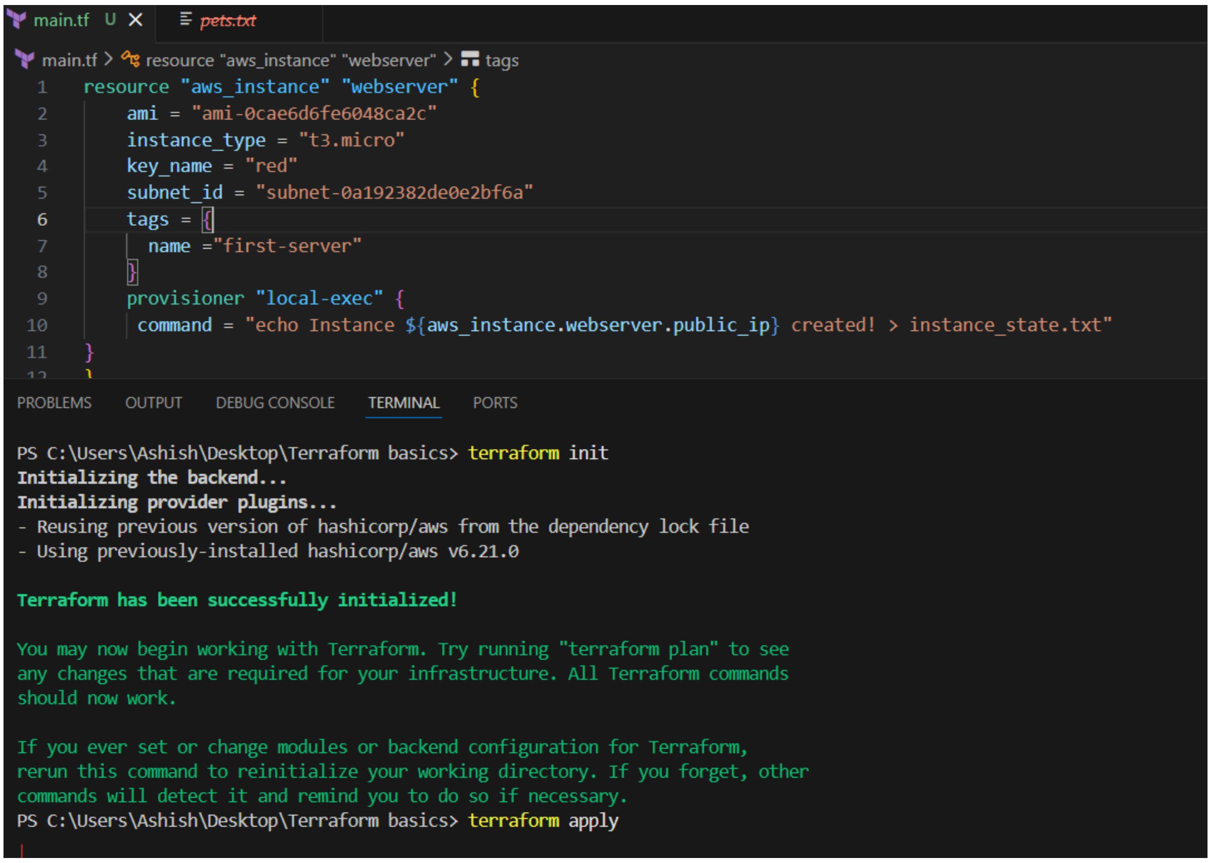


```
          # (1 unchanged attribute hidden)
        }
    }

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.webserver: Destroying... [id=i-0ad0218d81e082685]
aws_instance.webserver: Still destroying... [id=i-0ad0218d81e082685, 00m10s elapsed]
aws_instance.webserver: Still destroying... [id=i-0ad0218d81e082685, 00m20s elapsed]
aws_instance.webserver: Still destroying... [id=i-0ad0218d81e082685, 00m30s elapsed]
aws_instance.webserver: Still destroying... [id=i-0ad0218d81e082685, 00m40s elapsed]
aws_instance.webserver: Still destroying... [id=i-0ad0218d81e082685, 00m50s elapsed]
aws_instance.webserver: Still destroying... [id=i-0ad0218d81e082685, 01m00s elapsed]
aws_instance.webserver: Still destroying... [id=i-0ad0218d81e082685, 01m10s elapsed]
aws_instance.webserver: Still destroying... [id=i-0ad0218d81e082685, 01m20s elapsed]
aws_instance.webserver: Destruction complete after 1m25s

Destroy complete! Resources: 1 destroyed.
PS C:\Users\Ashish\Desktop\Terraform basics>
```



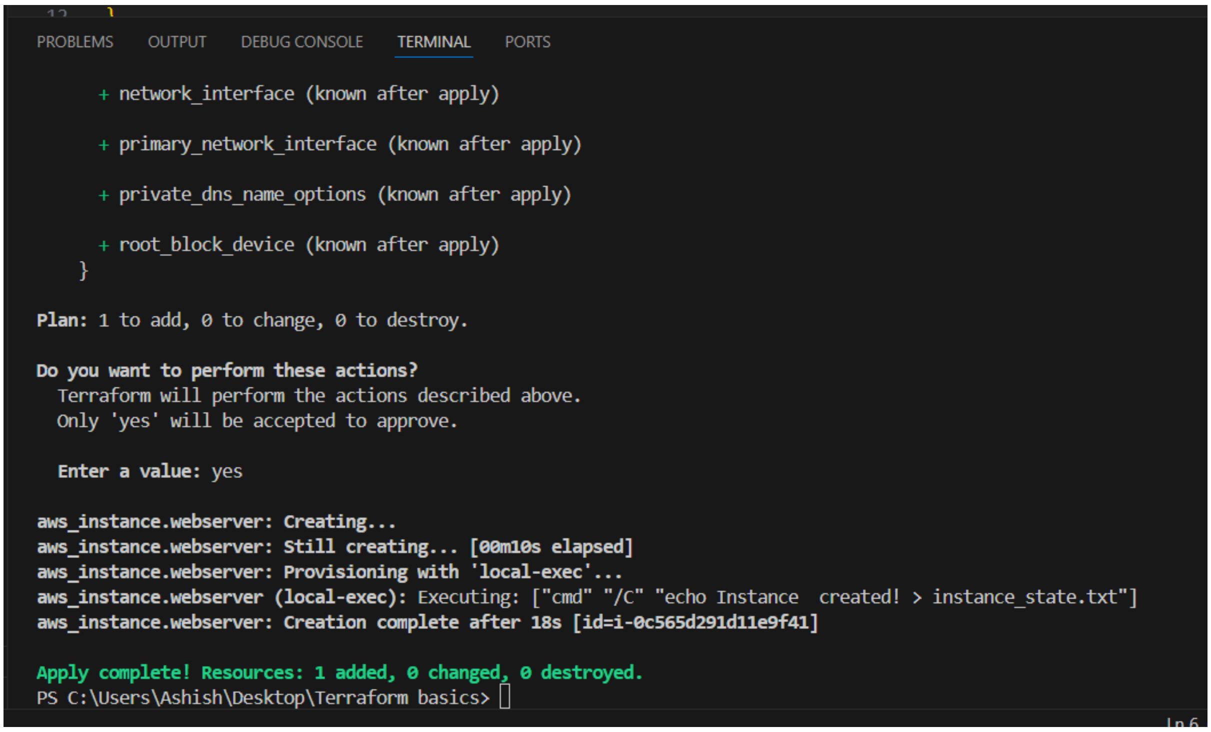
```
resource "aws_instance" "webserver" {
    ami = "ami-0cae6d6fe6048ca2c"
    instance_type = "t3.micro"
    key_name = "red"
    subnet_id = "subnet-0a192382de0e2bf6a"
    tags = {
      name ="first-server"
    }
    provisioner "local-exec" {
     command = "echo Instance ${aws_instance.webserver.public_ip} created! > instance_state.txt"
}
```

```
resource "aws_instance" "webserver" {
    ami = "ami-0cae6d6fe6048ca2c"
    instance_type = "t3.micro"
    key_name = "red"
    subnet_id = "subnet-0a192382de0e2bf6a"
    tags = {
      name ="first-server"
    }
    provisioner "local-exec" {
     command = "echo Instance
${aws_instance.webserver.public_ip} created! >
instance_state.txt"
}
}
```


main.tf U ×
pets.txt
main.tf > resource "aws_instance" "webserver" > tags
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

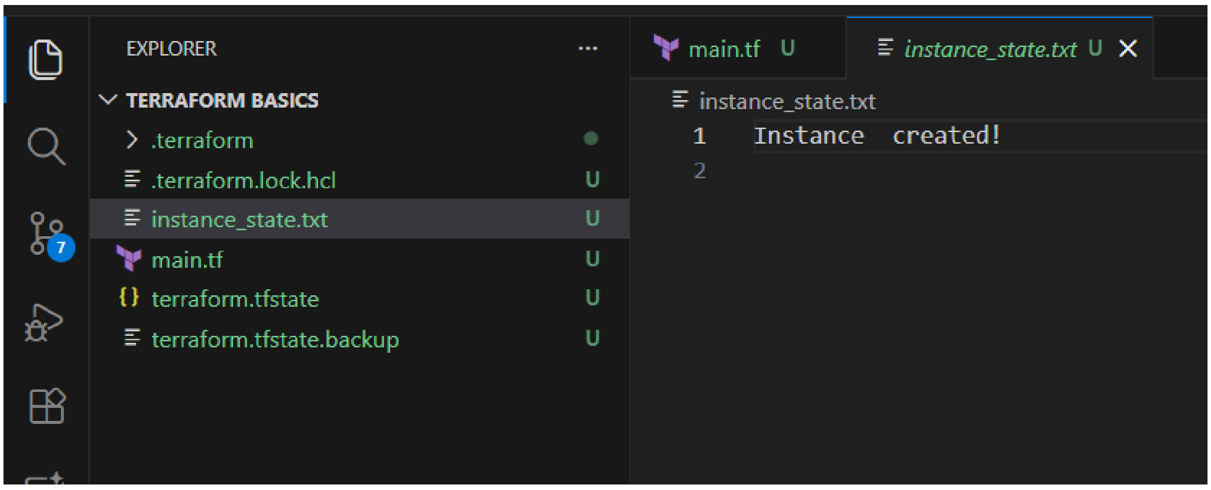
```
resource "aws_instance" "webserver" {
    ami = "ami-0cae6d6fe6048ca2c"
    instance_type = "t3.micro"
    key_name = "red"
    subnet_id = "subnet-0a192382de0e2bf6a"
    tags = {
      name ="first-server"
    }
    provisioner "local-exec" {
     command = "echo Instance ${aws_instance.webserver.public_ip} created! > instance_state.txt"
  }
}
```

```
PS C:\Users\Ashish\Desktop\Terraform basics> terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v6.21.0

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.
PS C:\Users\Ashish\Desktop\Terraform basics> terraform apply
```



```
      + network_interface (known after apply)

      + primary_network_interface (known after apply)

      + 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?
  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... [00m10s elapsed]
aws_instance.webserver: Provisioning with 'local-exec'...
aws_instance.webserver (local-exec): Executing: ["cmd" "/C" "echo Instance  created! > instance_state.txt"]
aws_instance.webserver: Creation complete after 18s [id=i-0c565d291d11e9f41]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\Users\Ashish\Desktop\Terraform basics>
```

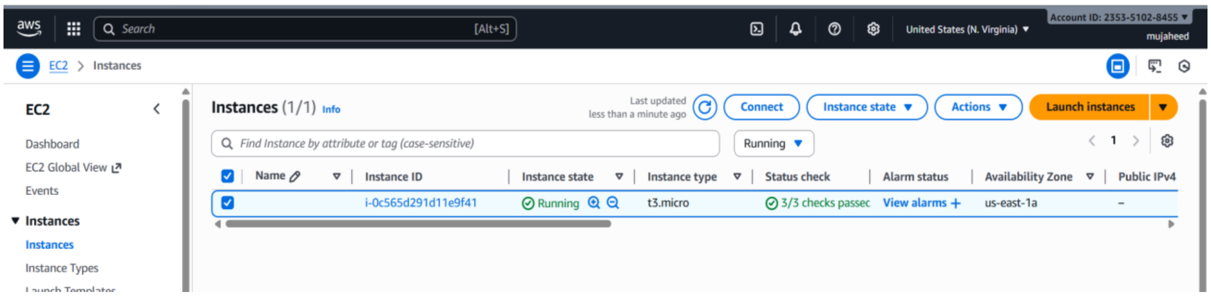
You will get a message in the file that you given in the script.


EXPLORER

TERRAFORM BASICS
- .terraform
- .terraform.lock.hcl U
- instance_state.txt U
- main.tf U
- terraform.tfstate U
- terraform.tfstate.backup U

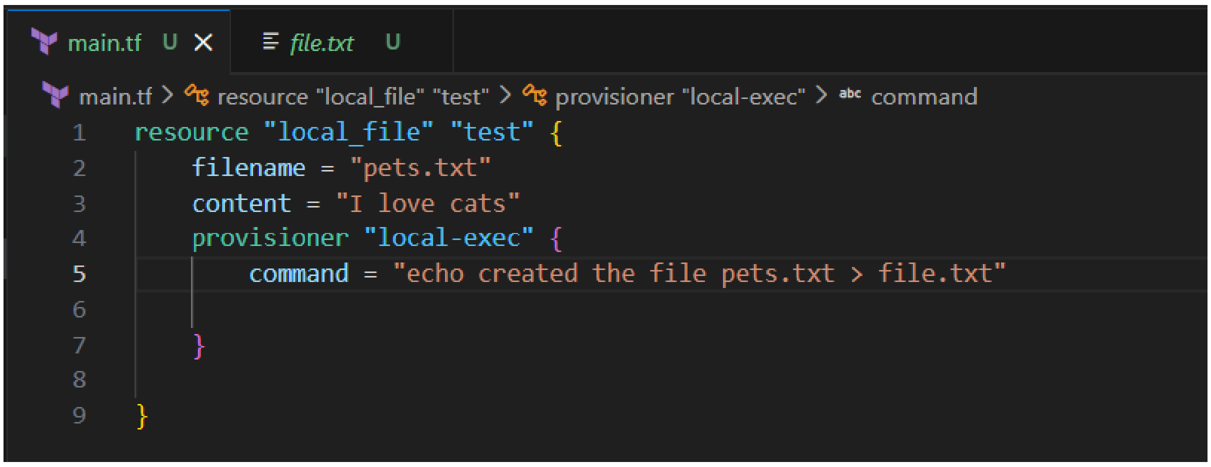
instance_state.txt

```
Instance  created!
```

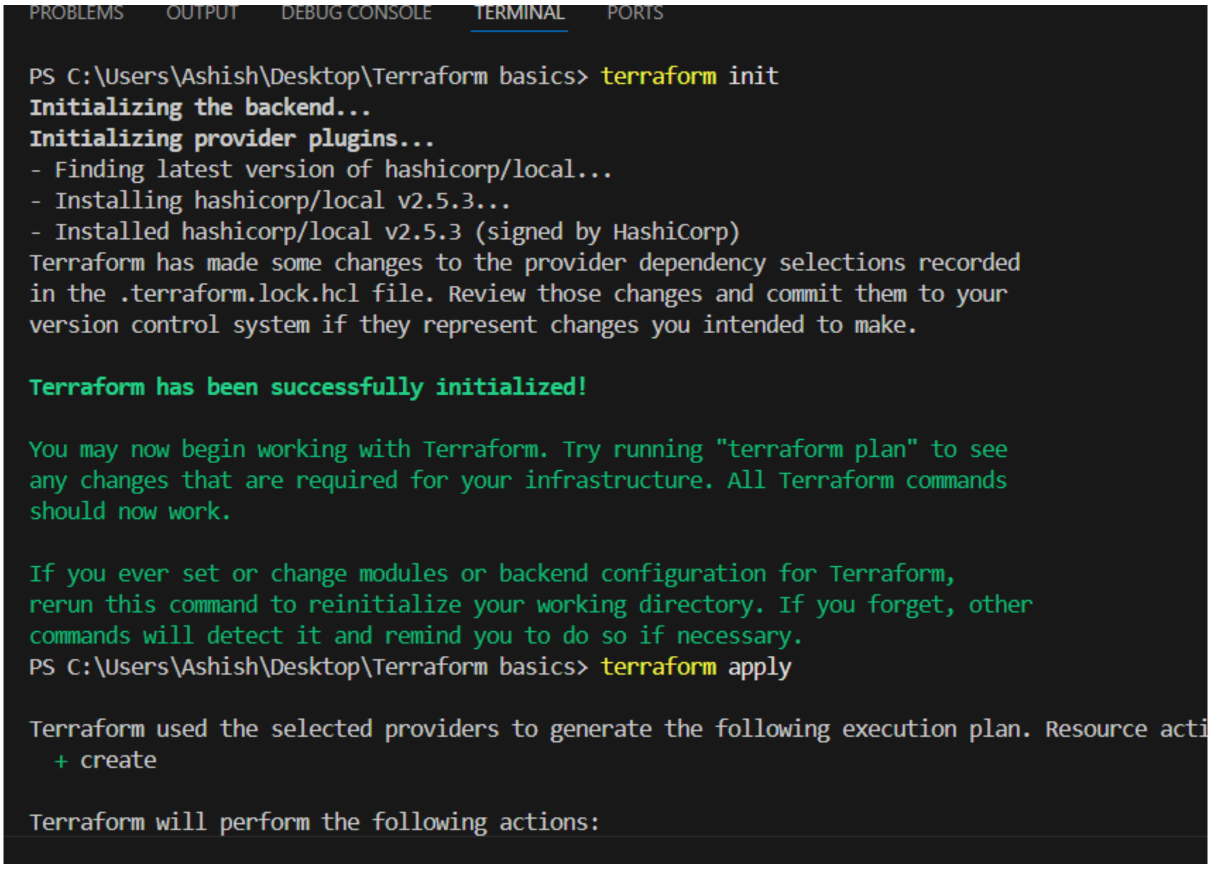

EC2 > Instances

Instances (1/1) Info — Last updated less than a minute ago

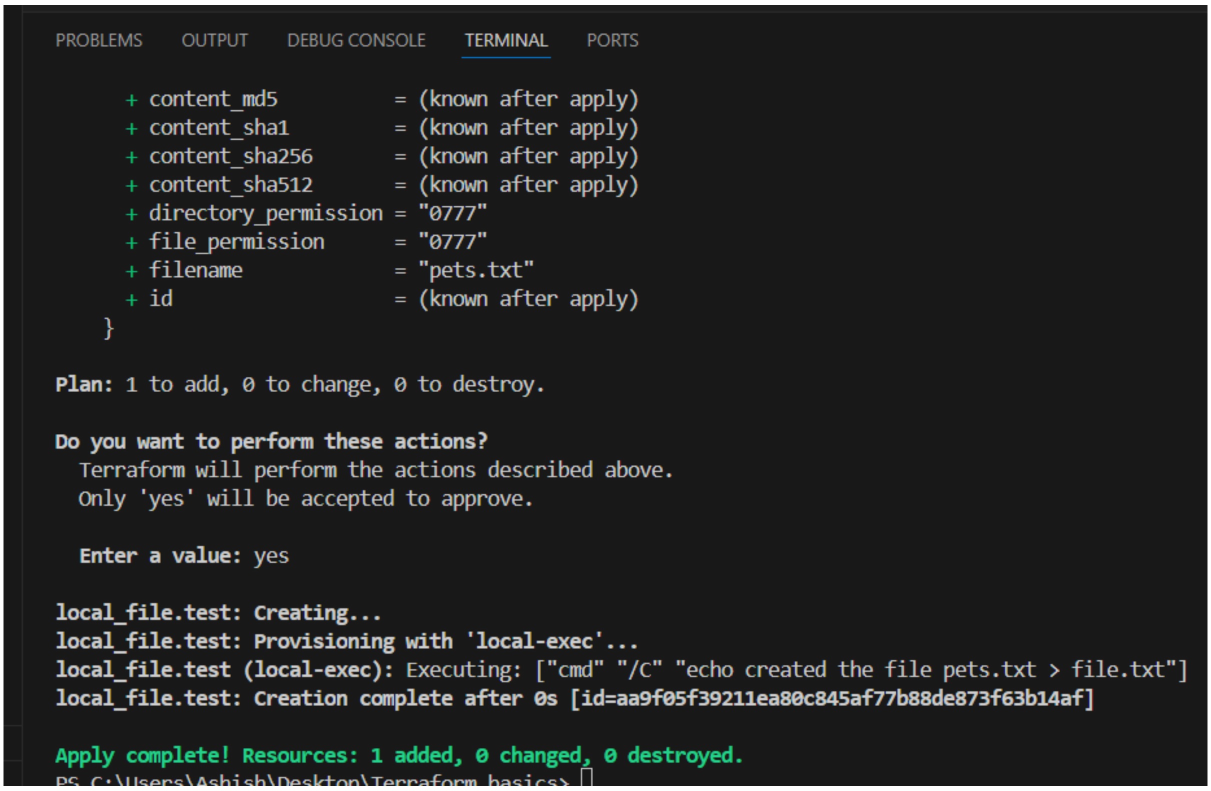
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
✓		i-0c565d291d11e9f41	Running	t3.micro	3/3 checks passec	View alarms +	us-east-1a	–


main.tf > resource "local_file" "test" > provisioner "local-exec" > command

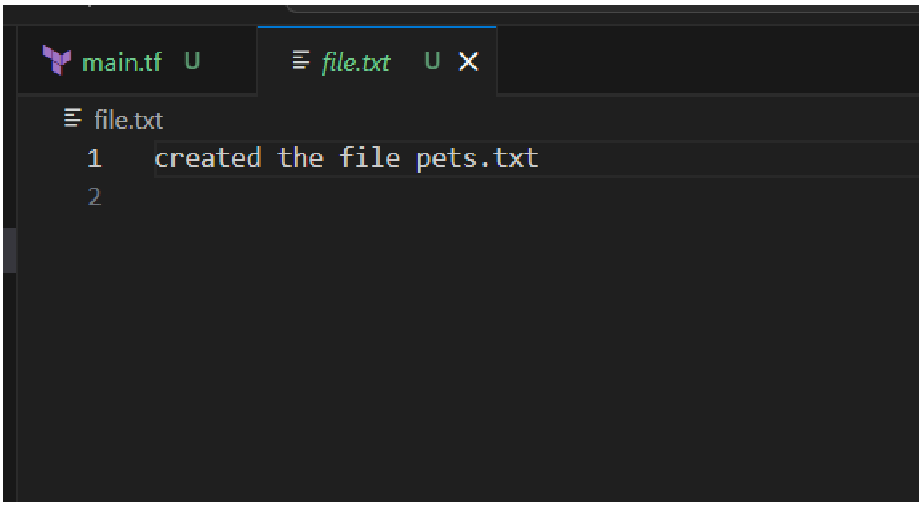
```
resource "local_file" "test" {
    filename = "pets.txt"
    content = "I love cats"
    provisioner "local-exec" {
        command = "echo created the file pets.txt > file.txt"

    }

}
```



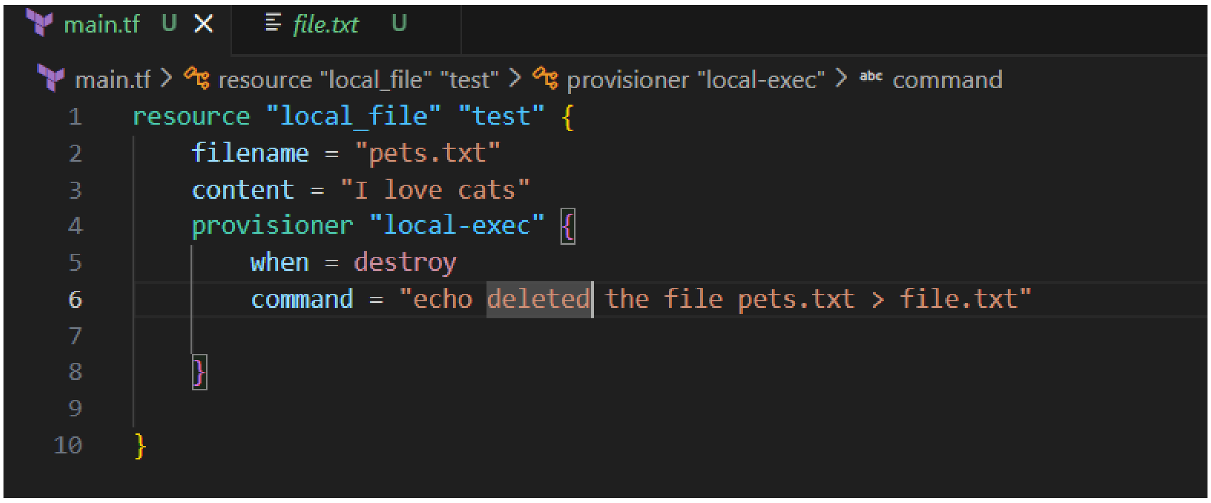
```
PS C:\Users\Ashish\Desktop\Terraform basics> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/local...
- Installing hashicorp/local v2.5.3...
- Installed hashicorp/local v2.5.3 (signed by HashiCorp)
Terraform has made some changes to 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.

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.
PS C:\Users\Ashish\Desktop\Terraform basics> terraform apply

Terraform used the selected providers to generate the following execution plan. Resource acti
  + create

Terraform will perform the following actions:
```



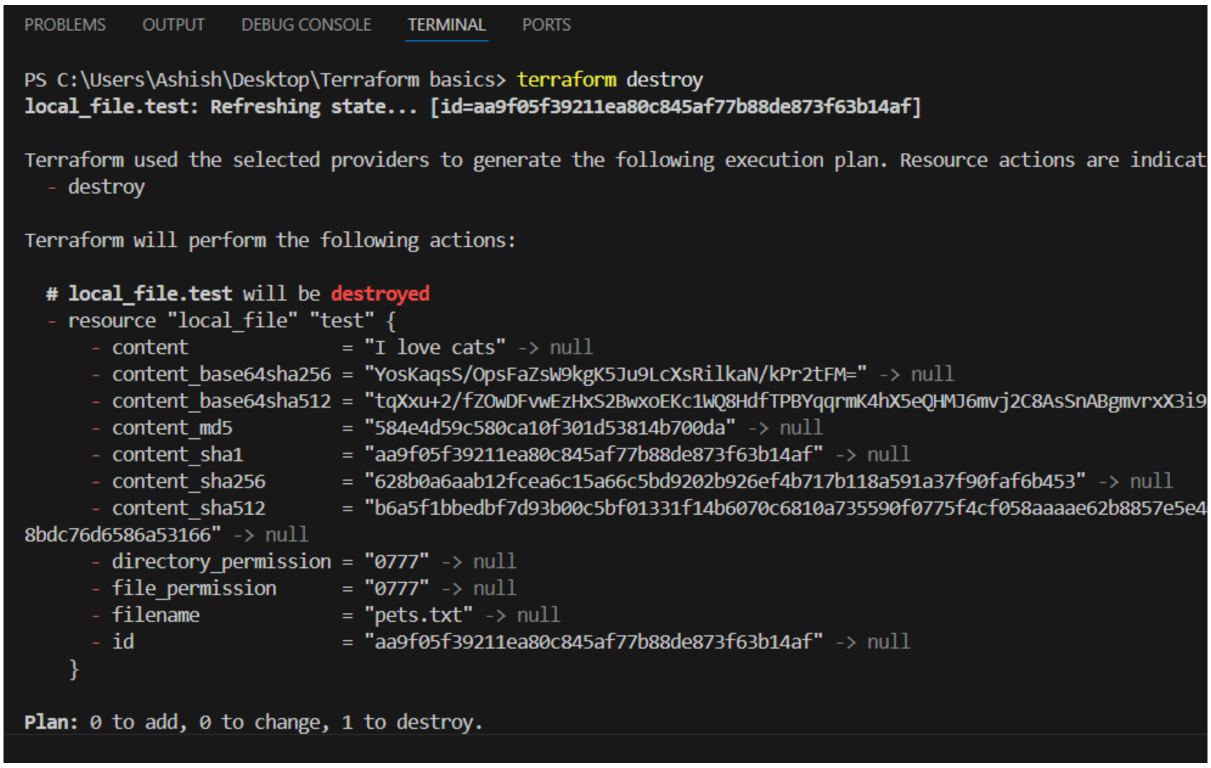
```
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "pets.txt"
      + id                   = (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

local_file.test: Creating...
local_file.test: Provisioning with 'local-exec'...
local_file.test (local-exec): Executing: ["cmd" "/C" "echo created the file pets.txt > file.txt"]
local_file.test: Creation complete after 0s [id=aa9f05f39211ea80c845af77b88de873f63b14af]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\Users\Ashish\Desktop\Terraform basics>
```



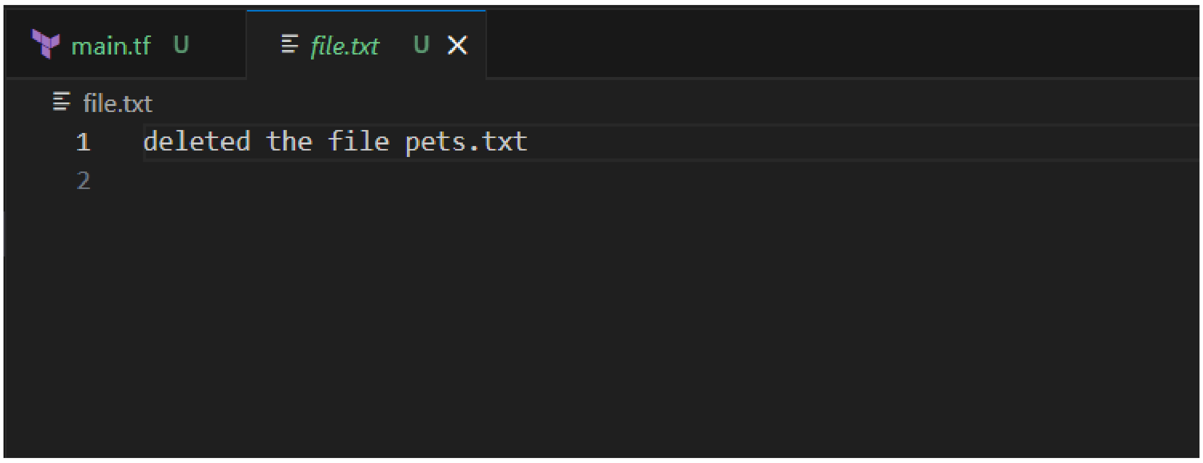
```
created the file pets.txt
```


main.tf U
file.txt U
main.tf > resource "local_file" "test" > provisioner "local-exec" > command

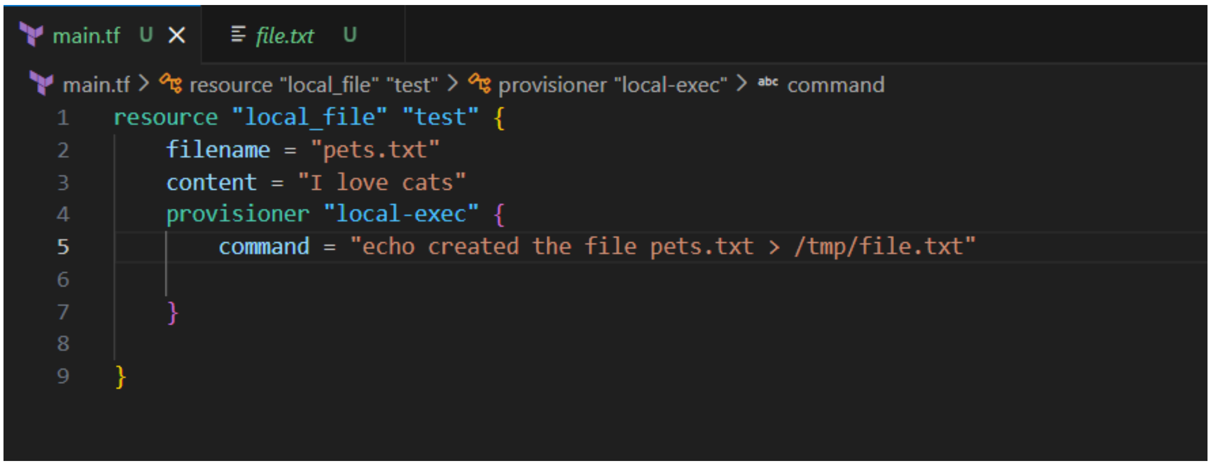
```
resource "local_file" "test" {
    filename = "pets.txt"
    content = "I love cats"
    provisioner "local-exec" {
        when = destroy
        command = "echo deleted the file pets.txt > file.txt"

    }

}
```



```
PS C:\Users\Ashish\Desktop\Terraform basics> terraform destroy
local_file.test: Refreshing state... [id=aa9f05f39211ea80c845af77b88de873f63b14af]

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

Terraform will perform the following actions:

  # local_file.test will be destroyed
  - resource "local_file" "test" {
      - content              = "I love cats" -> null
      - content_base64sha256 = "YosKaqsS/OpsFaZsW9kgK5Ju9LcXsRilkaN/kPr2tFM=" -> null
      - content_base64sha512 = "tqXxu+2/fZOwDFvwEzHxS2BwxoEKc1WQ8HdfTPBYqqrmK4hX5eQHMJ6mvj2C8AsSnABgmvrxX3i9
      - content_md5          = "584e4d59c580ca10f301d53814b700da" -> null
      - content_sha1         = "aa9f05f39211ea80c845af77b88de873f63b14af" -> null
      - content_sha256       = "628b0a6aab12fcea6c15a66c5bd9202b926ef4b717b118a591a37f90faf6b453" -> null
      - content_sha512       = "b6a5f1bbedbf7d93b00c5bf01331f14b6070c6810a735590f0775f4cf058aaaae62b8857e5e4
8bdc76d6586a53166" -> null
      - directory_permission = "0777" -> null
      - file_permission      = "0777" -> null
      - filename             = "pets.txt" -> null
      - id                   = "aa9f05f39211ea80c845af77b88de873f63b14af" -> null
    }

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

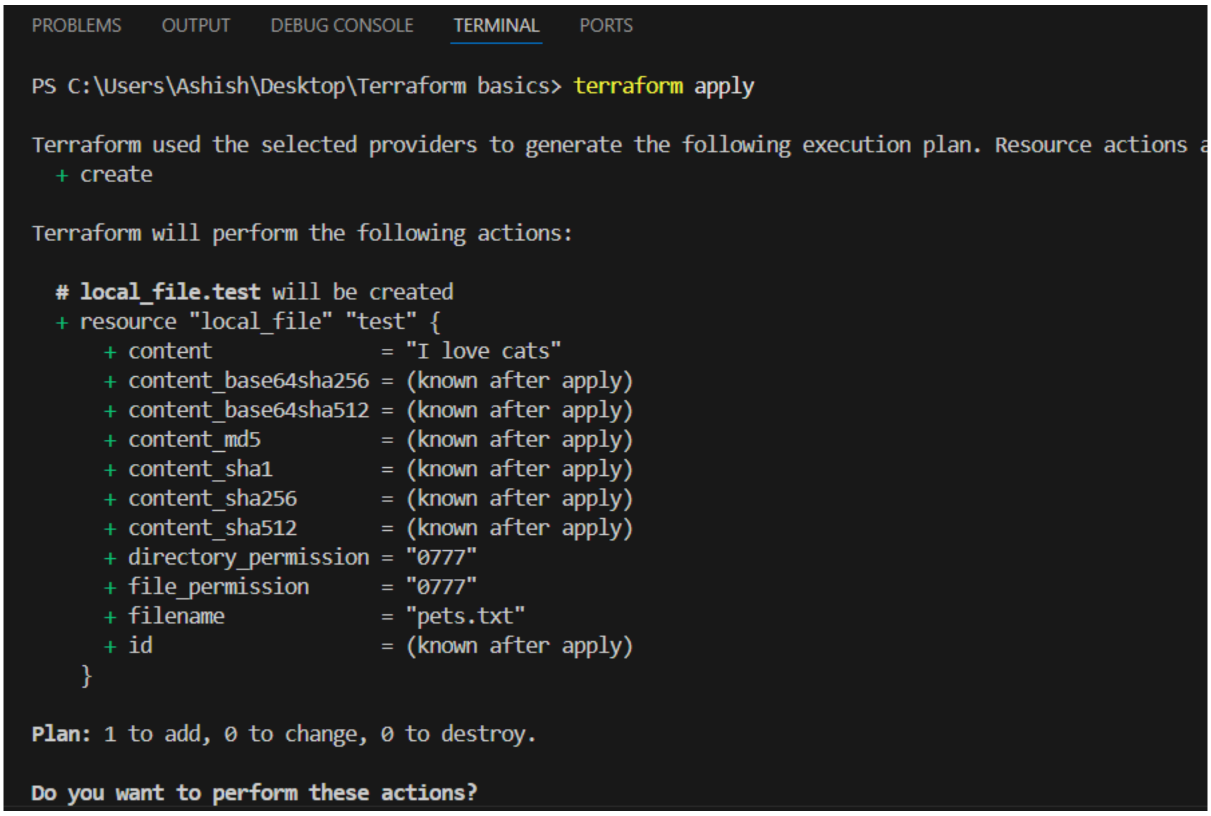


```
deleted the file pets.txt
```

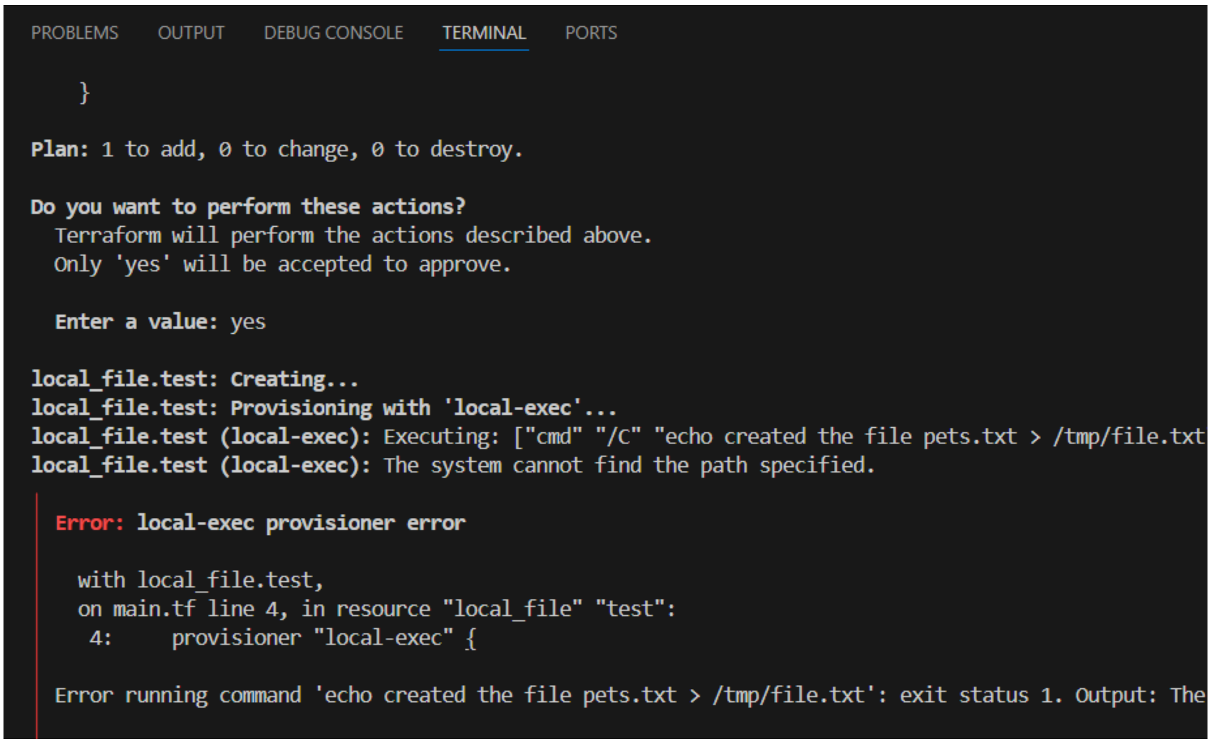

main.tf U
file.txt U
main.tf > resource "local_file" "test" > provisioner "local-exec" > command

```
resource "local_file" "test" {
    filename = "pets.txt"
    content = "I love cats"
    provisioner "local-exec" {
        command = "echo created the file pets.txt > /tmp/file.txt"

    }

}
```

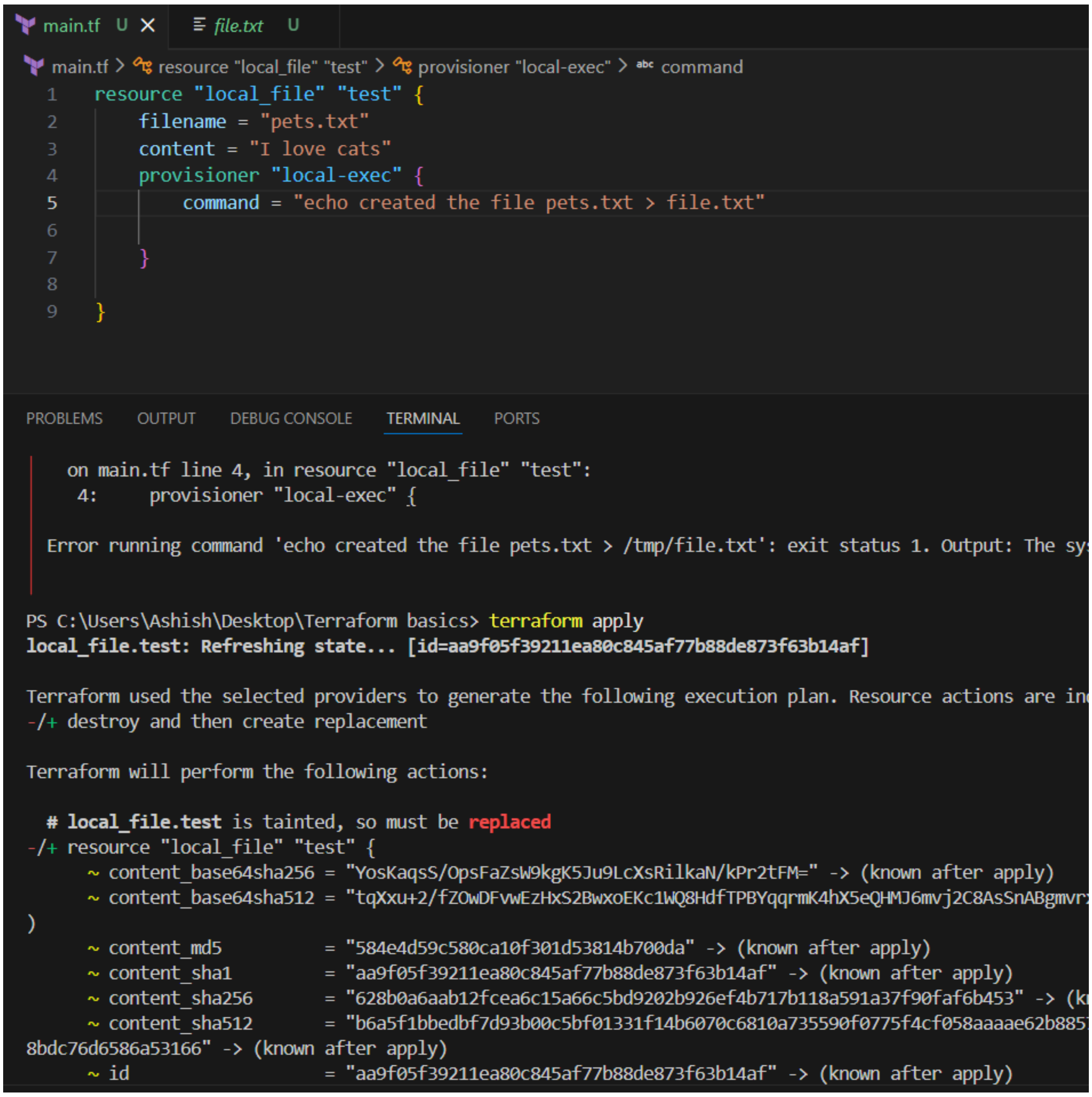
It is not created because in my pc there is no such path is available.



```
PS C:\Users\Ashish\Desktop\Terraform basics> terraform apply

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

Terraform will perform the following actions:

  # local_file.test will be created
  + resource "local_file" "test" {
      + content              = "I love cats"
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "pets.txt"
      + id                   = (known after apply)
    }

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

Do you want to perform these actions?
```



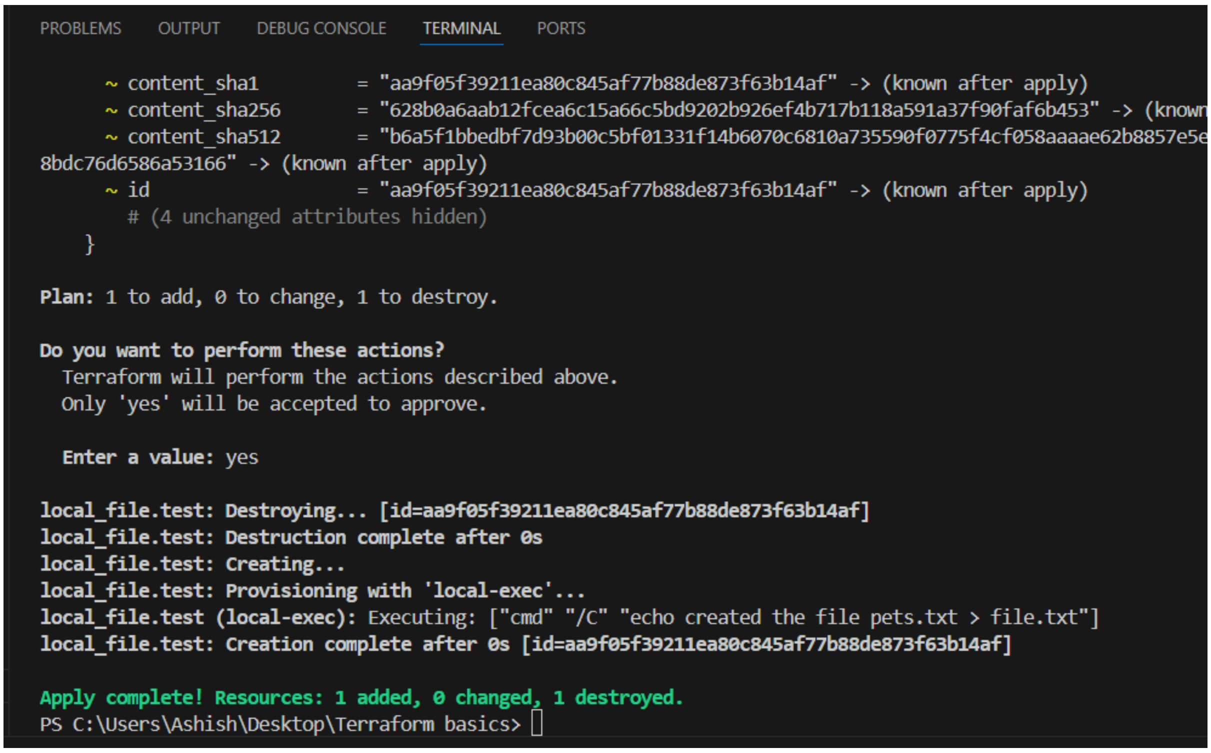
```
    }

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

local_file.test: Creating...
local_file.test: Provisioning with 'local-exec'...
local_file.test (local-exec): Executing: ["cmd" "/C" "echo created the file pets.txt > /tmp/file.txt
local_file.test (local-exec): The system cannot find the path specified.

  Error: local-exec provisioner error

    with local_file.test,
    on main.tf line 4, in resource "local_file" "test":
     4:     provisioner "local-exec" {

  Error running command 'echo created the file pets.txt > /tmp/file.txt': exit status 1. Output: The
```



main.tf U
file.txt U
main.tf > resource "local_file" "test" > provisioner "local-exec" > command
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS

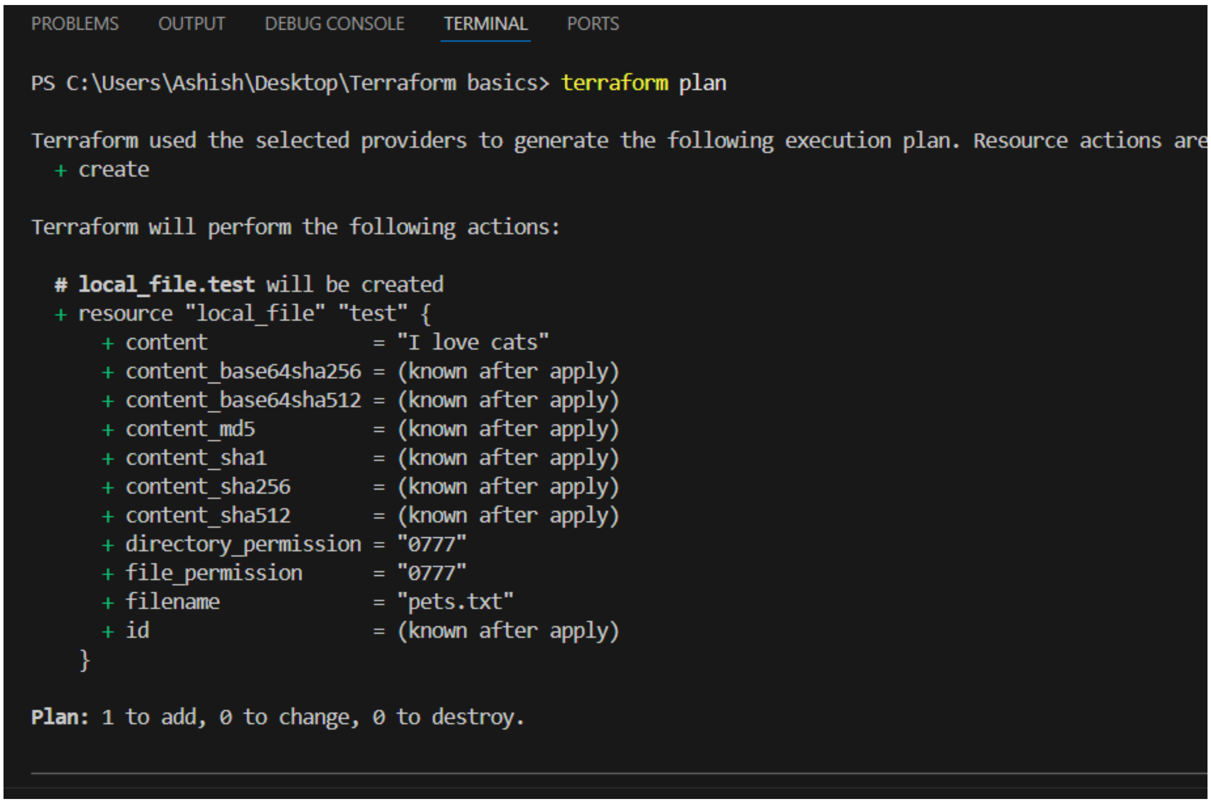
```
resource "local_file" "test" {
    filename = "pets.txt"
    content = "I love cats"
    provisioner "local-exec" {
        command = "echo created the file pets.txt > file.txt"

    }

}
```

```
   on main.tf line 4, in resource "local_file" "test":
    4:     provisioner "local-exec" {

 Error running command 'echo created the file pets.txt > /tmp/file.txt': exit status 1. Output: The sy

PS C:\Users\Ashish\Desktop\Terraform basics> terraform apply
local_file.test: Refreshing state... [id=aa9f05f39211ea80c845af77b88de873f63b14af]

Terraform used the selected providers to generate the following execution plan. Resource actions are in
-/+ destroy and then create replacement

Terraform will perform the following actions:

  # local_file.test is tainted, so must be replaced
-/+ resource "local_file" "test" {
      ~ content_base64sha256 = "YosKaqsS/OpsFaZsW9kgK5Ju9LcXsRilkaN/kPr2tFM=" -> (known after apply)
      ~ content_base64sha512 = "tqXxu+2/fZOwDFvwEzHxS2BwxoEKc1WQ8HdfTPBYqqrmK4hX5eQHMJ6mvj2C8AsSnABgmvr
)
      ~ content_md5          = "584e4d59c580ca10f301d53814b700da" -> (known after apply)
      ~ content_sha1         = "aa9f05f39211ea80c845af77b88de873f63b14af" -> (known after apply)
      ~ content_sha256       = "628b0a6aab12fcea6c15a66c5bd9202b926ef4b717b118a591a37f90faf6b453" -> (k
      ~ content_sha512       = "b6a5f1bbedbf7d93b00c5bf01331f14b6070c6810a735590f0775f4cf058aaaae62b885
8bdc76d6586a53166" -> (known after apply)
      ~ id                   = "aa9f05f39211ea80c845af77b88de873f63b14af" -> (known after apply)
```


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
      ~ content_sha1         = "aa9f05f39211ea80c845af77b88de873f63b14af" -> (known after apply)
      ~ content_sha256       = "628b0a6aab12fcea6c15a66c5bd9202b926ef4b717b118a591a37f90faf6b453" -> (known
      ~ content_sha512       = "b6a5f1bbedbf7d93b00c5bf01331f14b6070c6810a735590f0775f4cf058aaaae62b8857e5e
8bdc76d6586a53166" -> (known after apply)
      ~ id                   = "aa9f05f39211ea80c845af77b88de873f63b14af" -> (known after apply)
        # (4 unchanged attributes hidden)
    }

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

local_file.test: Destroying... [id=aa9f05f39211ea80c845af77b88de873f63b14af]
local_file.test: Destruction complete after 0s
local_file.test: Creating...
local_file.test: Provisioning with 'local-exec'...
local_file.test (local-exec): Executing: ["cmd" "/C" "echo created the file pets.txt > file.txt"]
local_file.test: Creation complete after 0s [id=aa9f05f39211ea80c845af77b88de873f63b14af]

Apply complete! Resources: 1 added, 0 changed, 1 destroyed.
PS C:\Users\Ashish\Desktop\Terraform basics>
```

If you do terraform you will see some logs.

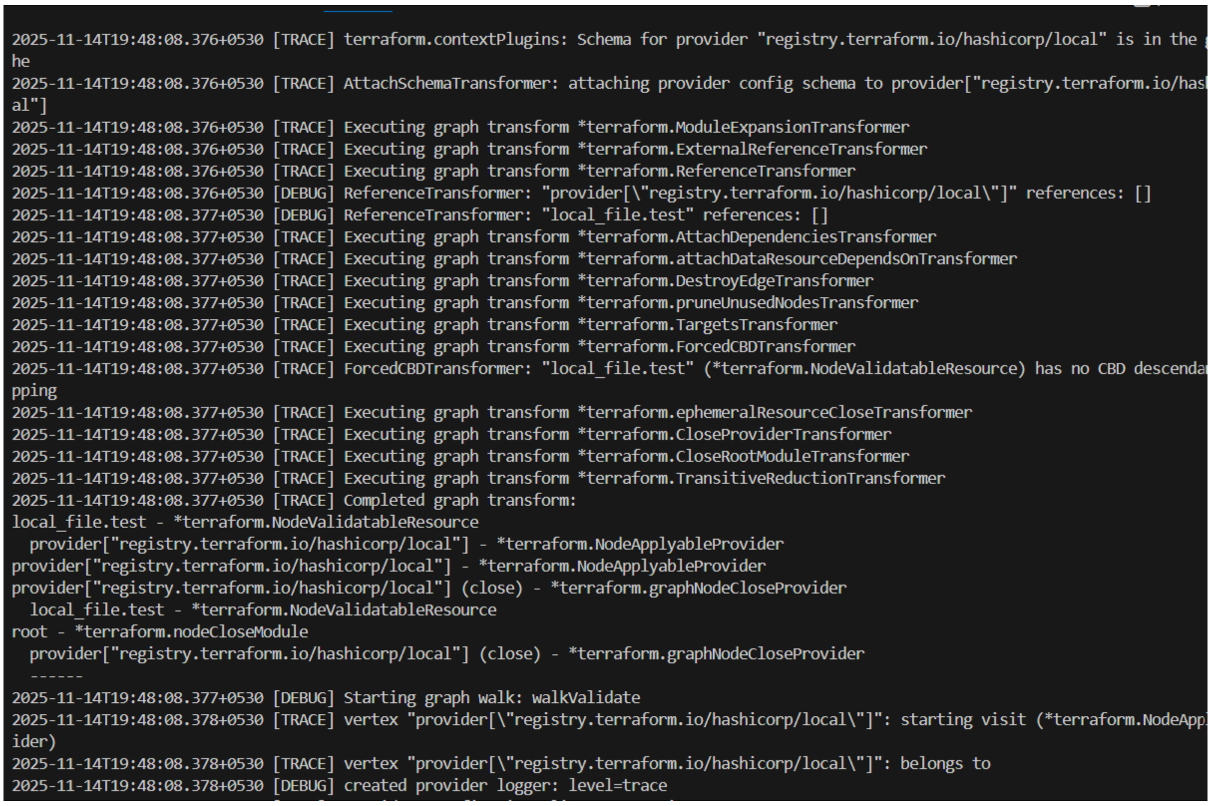

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Ashish\Desktop\Terraform basics> terraform plan

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

Terraform will perform the following actions:

  # local_file.test will be created
  + resource "local_file" "test" {
      + content              = "I love cats"
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "pets.txt"
      + id                   = (known after apply)
    }

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

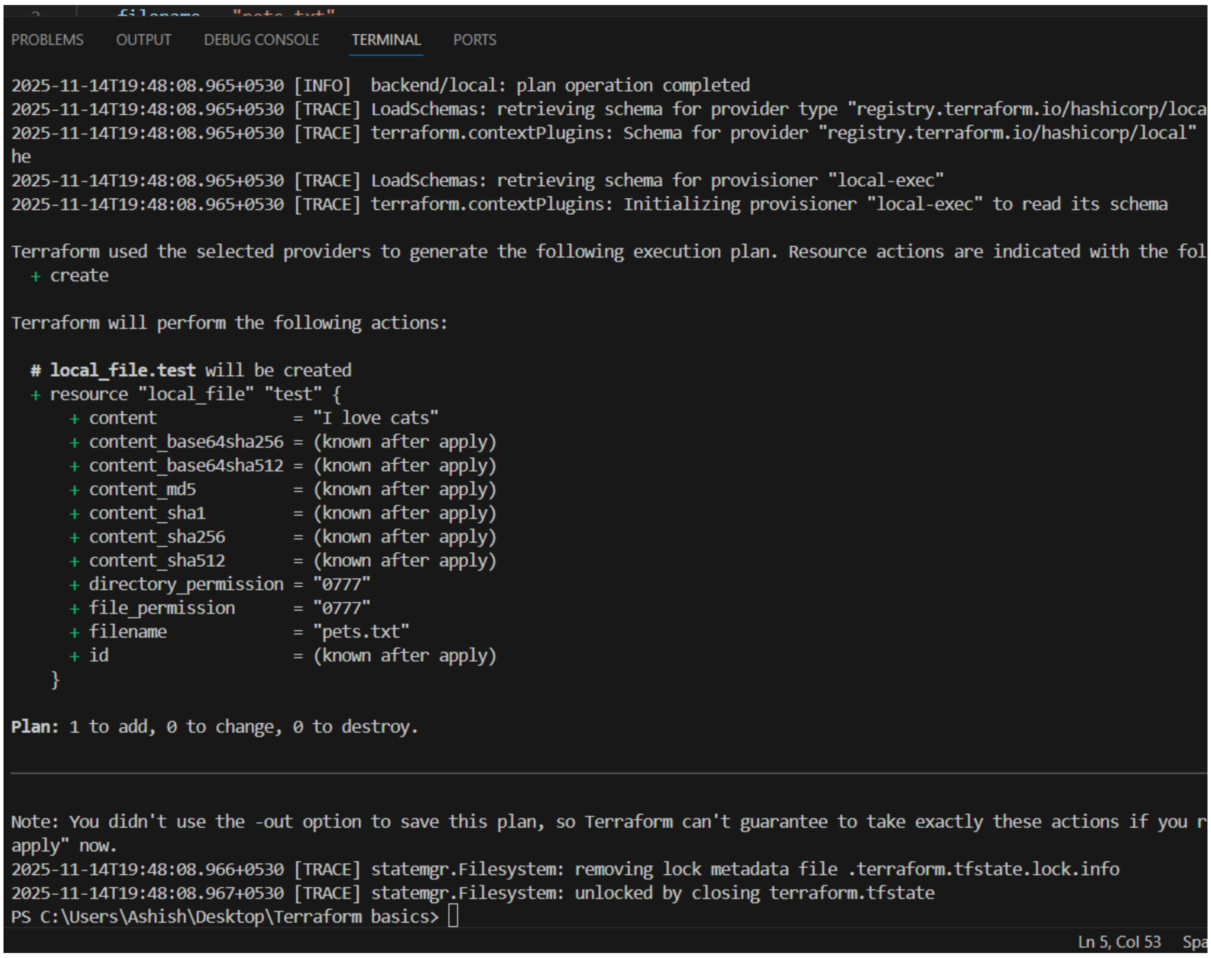
By passing this

- Set-Item -Path env:TF_LOG -value "TRACE"

After you will do terraform plan you will see more number of logs.



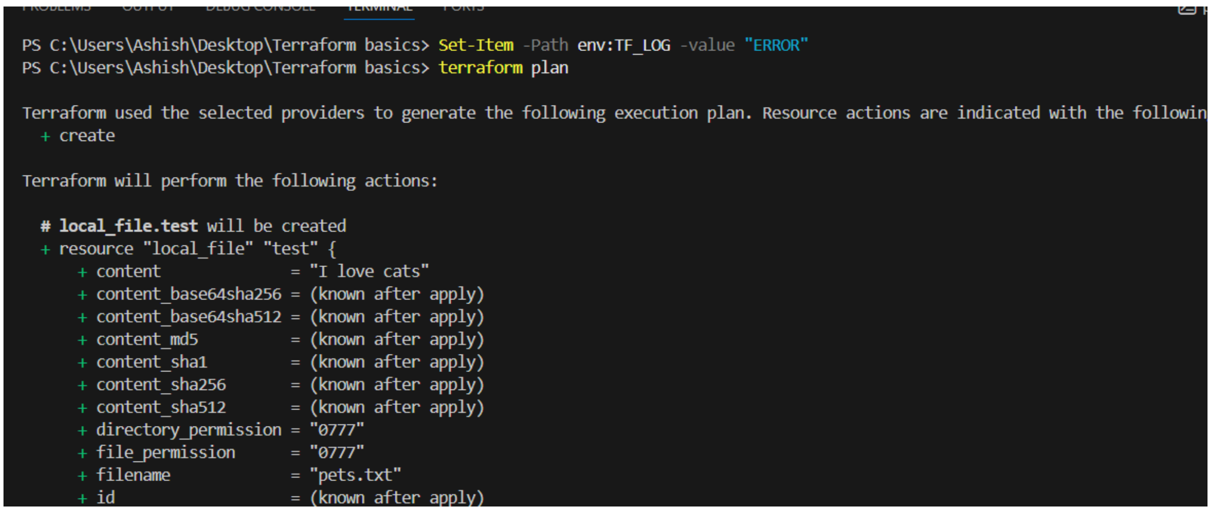
```
2025-11-14T19:48:08.376+0530 [TRACE] terraform.contextPlugins: Schema for provider "registry.terraform.io/hashicorp/local" is in the g
he
2025-11-14T19:48:08.376+0530 [TRACE] AttachSchemaTransformer: attaching provider config schema to provider["registry.terraform.io/hash
al"]
2025-11-14T19:48:08.376+0530 [TRACE] Executing graph transform *terraform.ModuleExpansionTransformer
2025-11-14T19:48:08.376+0530 [TRACE] Executing graph transform *terraform.ExternalReferenceTransformer
2025-11-14T19:48:08.376+0530 [TRACE] Executing graph transform *terraform.ReferenceTransformer
2025-11-14T19:48:08.376+0530 [DEBUG] ReferenceTransformer: "provider[\"registry.terraform.io/hashicorp/local\"]" references: []
2025-11-14T19:48:08.377+0530 [DEBUG] ReferenceTransformer: "local_file.test" references: []
2025-11-14T19:48:08.377+0530 [TRACE] Executing graph transform *terraform.AttachDependenciesTransformer
2025-11-14T19:48:08.377+0530 [TRACE] Executing graph transform *terraform.attachDataResourceDependsOnTransformer
2025-11-14T19:48:08.377+0530 [TRACE] Executing graph transform *terraform.DestroyEdgeTransformer
2025-11-14T19:48:08.377+0530 [TRACE] Executing graph transform *terraform.pruneUnusedNodesTransformer
2025-11-14T19:48:08.377+0530 [TRACE] Executing graph transform *terraform.TargetsTransformer
2025-11-14T19:48:08.377+0530 [TRACE] Executing graph transform *terraform.ForcedCBDTransformer
2025-11-14T19:48:08.377+0530 [TRACE] ForcedCBDTransformer: "local_file.test" (*terraform.NodeValidatableResource) has no CBD descenda
pping
2025-11-14T19:48:08.377+0530 [TRACE] Executing graph transform *terraform.ephemeralResourceCloseTransformer
2025-11-14T19:48:08.377+0530 [TRACE] Executing graph transform *terraform.CloseProviderTransformer
2025-11-14T19:48:08.377+0530 [TRACE] Executing graph transform *terraform.CloseRootModuleTransformer
2025-11-14T19:48:08.377+0530 [TRACE] Executing graph transform *terraform.TransitiveReductionTransformer
2025-11-14T19:48:08.377+0530 [TRACE] Completed graph transform:
local_file.test - *terraform.NodeValidatableResource
  provider["registry.terraform.io/hashicorp/local"] - *terraform.NodeApplyableProvider
provider["registry.terraform.io/hashicorp/local"] - *terraform.NodeApplyableProvider
provider["registry.terraform.io/hashicorp/local"] (close) - *terraform.graphNodeCloseProvider
  local_file.test - *terraform.NodeValidatableResource
root - *terraform.nodeCloseModule
  provider["registry.terraform.io/hashicorp/local"] (close) - *terraform.graphNodeCloseProvider
  ------
2025-11-14T19:48:08.377+0530 [DEBUG] Starting graph walk: walkValidate
2025-11-14T19:48:08.378+0530 [TRACE] vertex "provider[\"registry.terraform.io/hashicorp/local\"]": starting visit (*terraform.NodeAppl
ider)
2025-11-14T19:48:08.378+0530 [TRACE] vertex "provider[\"registry.terraform.io/hashicorp/local\"]": belongs to
2025-11-14T19:48:08.378+0530 [DEBUG] created provider logger: level=trace
```



```
2025-11-14T19:48:08.965+0530 [INFO]  backend/local: plan operation completed
2025-11-14T19:48:08.965+0530 [TRACE] LoadSchemas: retrieving schema for provider type "registry.terraform.io/hashicorp/loca
2025-11-14T19:48:08.965+0530 [TRACE] terraform.contextPlugins: Schema for provider "registry.terraform.io/hashicorp/local"
he
2025-11-14T19:48:08.965+0530 [TRACE] LoadSchemas: retrieving schema for provisioner "local-exec"
2025-11-14T19:48:08.965+0530 [TRACE] terraform.contextPlugins: Initializing provisioner "local-exec" to read its schema

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

Terraform will perform the following actions:

  # local_file.test will be created
  + resource "local_file" "test" {
      + content              = "I love cats"
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "pets.txt"
      + id                   = (known after apply)
    }

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

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you r
apply" now.
2025-11-14T19:48:08.966+0530 [TRACE] statemgr.Filesystem: removing lock metadata file .terraform.tfstate.lock.info
2025-11-14T19:48:08.967+0530 [TRACE] statemgr.Filesystem: unlocked by closing terraform.tfstate
PS C:\Users\Ashish\Desktop\Terraform basics>
```

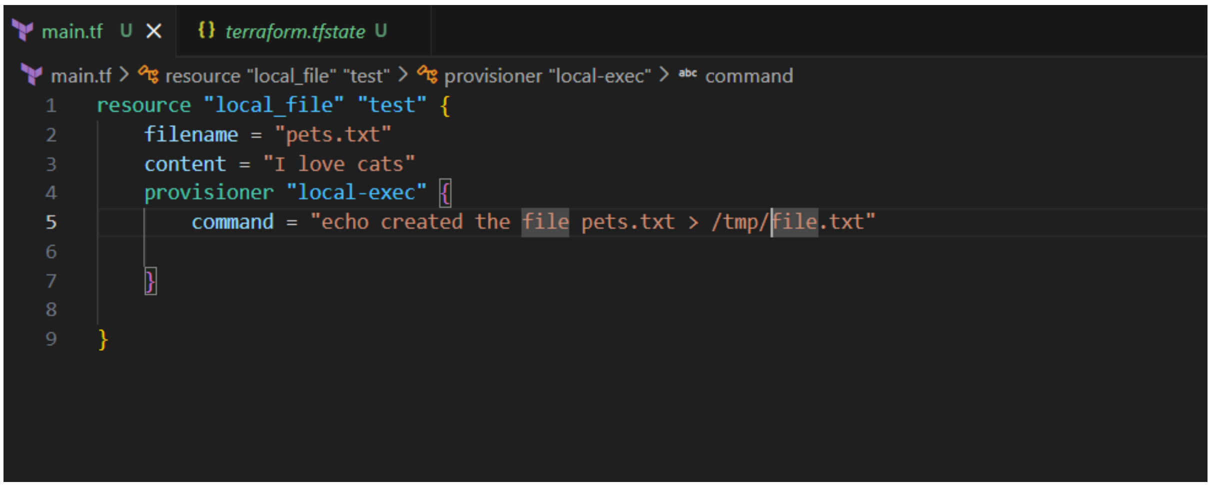
- Set-Item -Path env:TF_LOG -value "ERROR"
- terraform plan

you don’t see any errors because there are no errors.



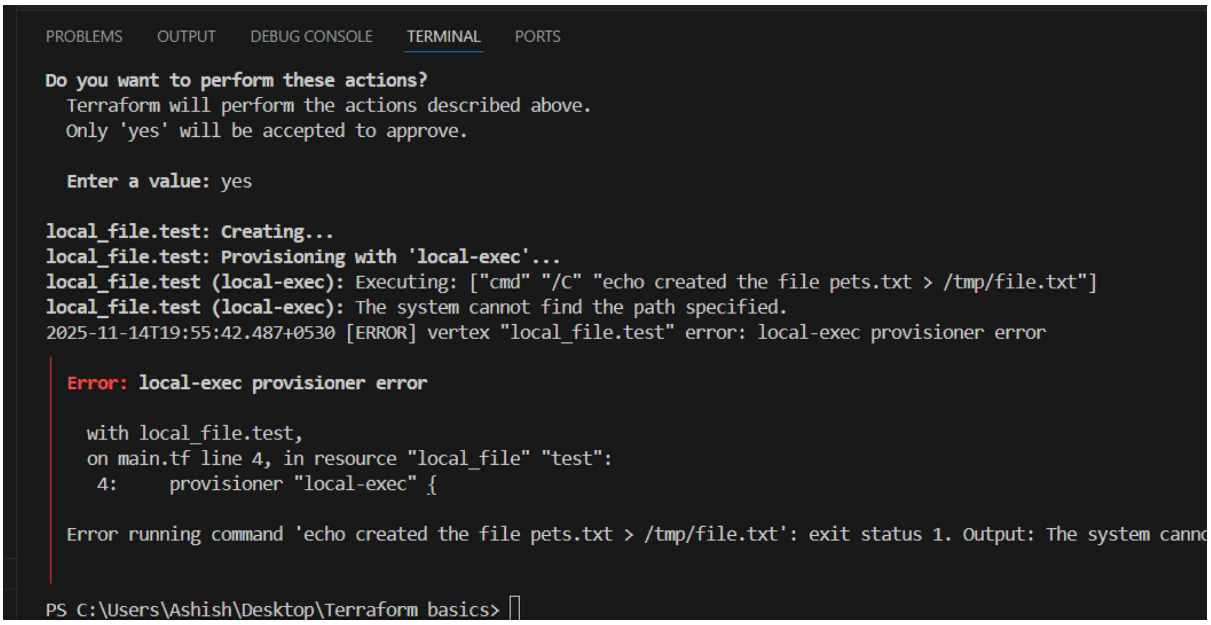
```
PS C:\Users\Ashish\Desktop\Terraform basics> Set-Item -Path env:TF_LOG -value "ERROR"
PS C:\Users\Ashish\Desktop\Terraform basics> terraform plan

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

Terraform will perform the following actions:

  # local_file.test will be created
  + resource "local_file" "test" {
      + content              = "I love cats"
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "pets.txt"
      + id                   = (known after apply)
```

If you provide error then it will show error.


main.tf U
terraform.tfstate U
main.tf > resource "local_file" "test" > provisioner "local-exec" > command

```
resource "local_file" "test" {
    filename = "pets.txt"
    content = "I love cats"
    provisioner "local-exec" {
        command = "echo created the file pets.txt > /tmp/file.txt"

    }

}
```

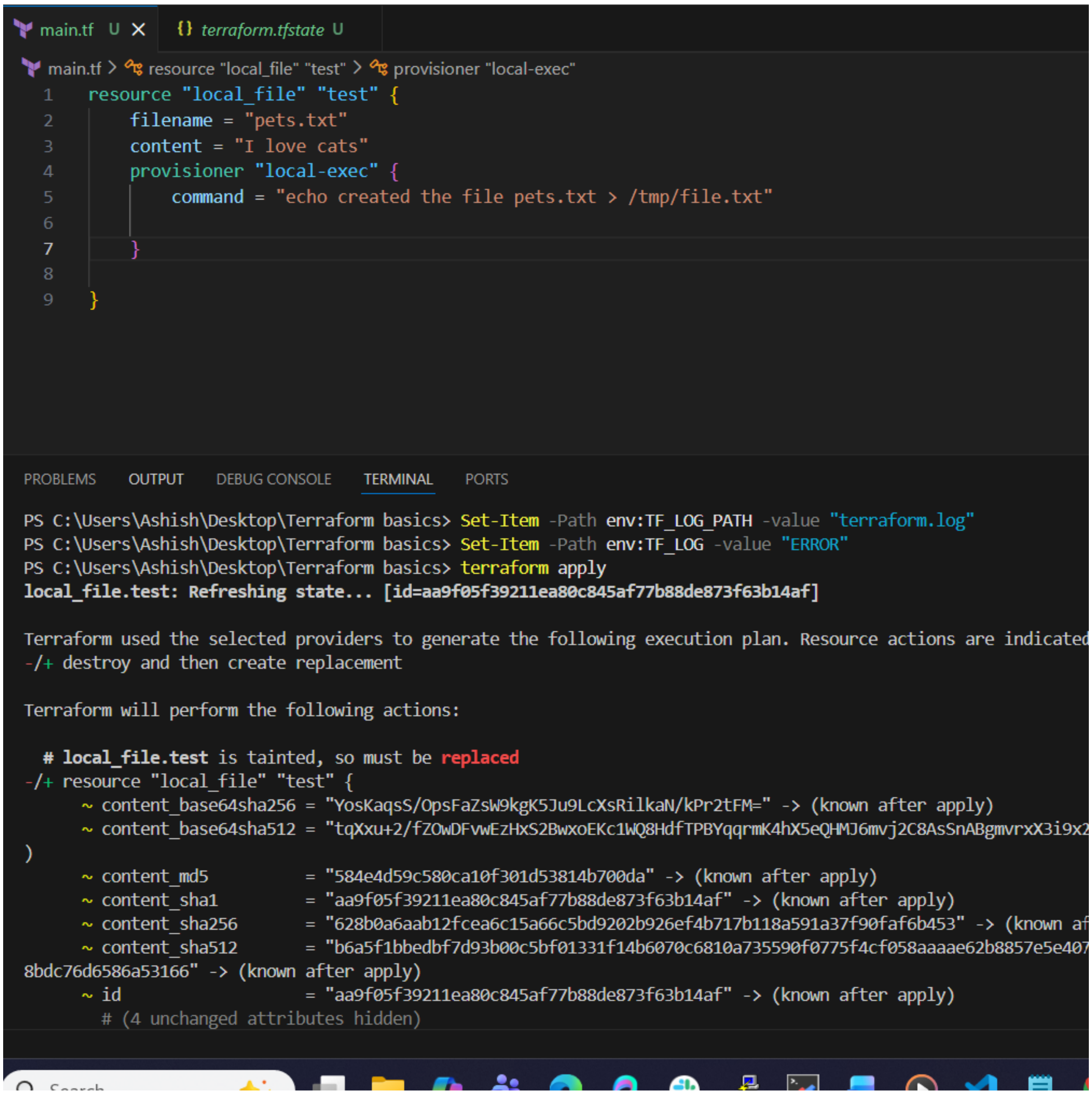
Click on terraform apply you will see errors.



```
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

local_file.test: Creating...
local_file.test: Provisioning with 'local-exec'...
local_file.test (local-exec): Executing: ["cmd" "/C" "echo created the file pets.txt > /tmp/file.txt"]
local_file.test (local-exec): The system cannot find the path specified.
2025-11-14T19:55:42.487+0530 [ERROR] vertex "local_file.test" error: local-exec provisioner error

  Error: local-exec provisioner error

    with local_file.test,
    on main.tf line 4, in resource "local_file" "test":
     4:     provisioner "local-exec" {

  Error running command 'echo created the file pets.txt > /tmp/file.txt': exit status 1. Output: The system canno

PS C:\Users\Ashish\Desktop\Terraform basics>
```

- Set-Item -Path env:TF_LOG_PATH -value "terraform.log"
- Set-Item -Path env:TF_LOG -value "ERROR"
- terraform apply

if we have errors the it will be stored In permanent file.



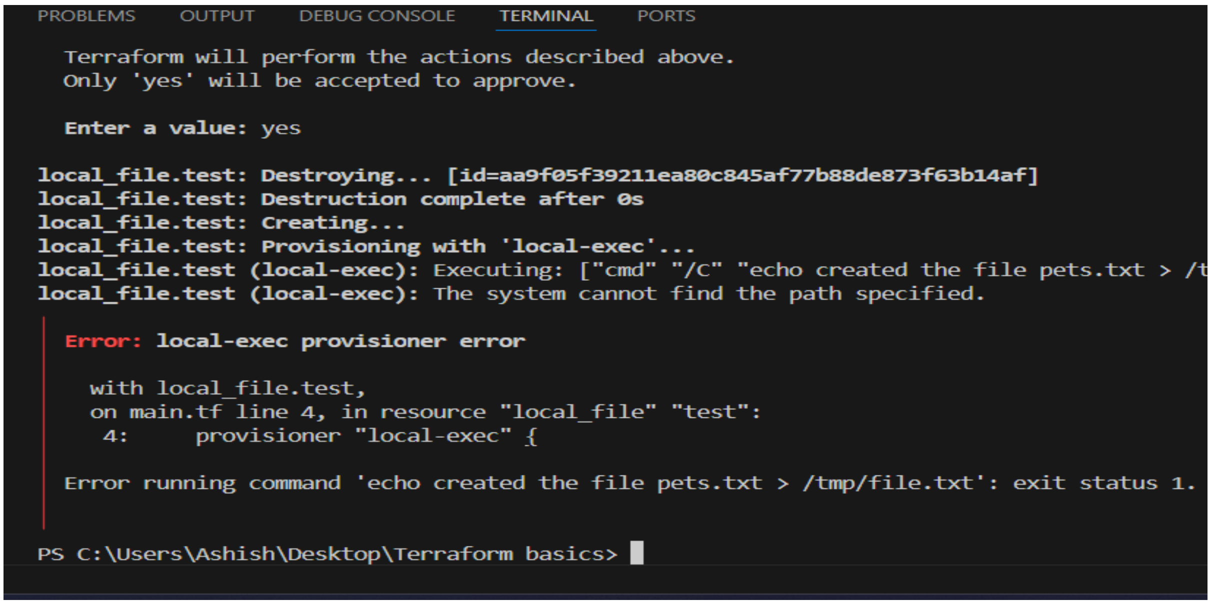
main.tf U × {} terraform.tfstate U

main.tf > resource "local_file" "test" > provisioner "local-exec"

```
resource "local_file" "test" {
    filename = "pets.txt"
    content = "I love cats"
    provisioner "local-exec" {
        command = "echo created the file pets.txt > /tmp/file.txt"

    }

}
```

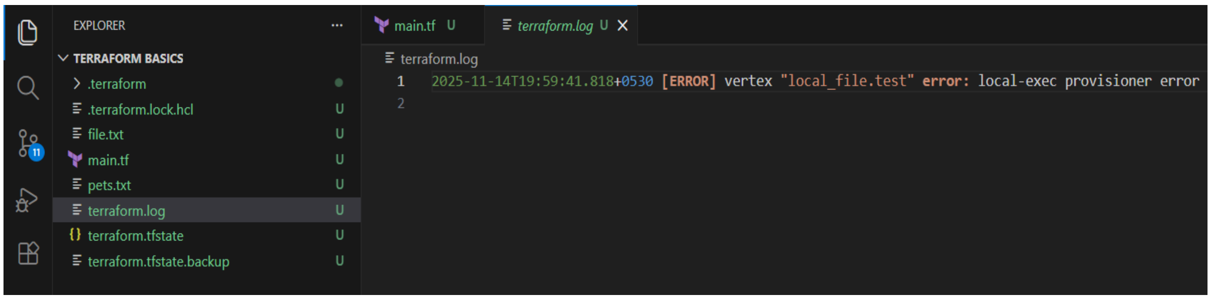
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Ashish\Desktop\Terraform basics> Set-Item -Path env:TF_LOG_PATH -value "terraform.log"
PS C:\Users\Ashish\Desktop\Terraform basics> Set-Item -Path env:TF_LOG -value "ERROR"
PS C:\Users\Ashish\Desktop\Terraform basics> terraform apply
local_file.test: Refreshing state... [id=aa9f05f39211ea80c845af77b88de873f63b14af]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated
-/+ destroy and then create replacement

Terraform will perform the following actions:

  # local_file.test is tainted, so must be replaced
-/+ resource "local_file" "test" {
      ~ content_base64sha256 = "YosKaqsS/OpsFaZsW9kgK5Ju9LcXsRilkaN/kPr2tFM=" -> (known after apply)
      ~ content_base64sha512 = "tqXxu+2/fZOwDFvwEzHxS2BwxoEKc1WQ8HdfTPBYqqrmK4hX5eQHMJ6mvj2C8AsSnABgmvrxX3i9x2
)
      ~ content_md5          = "584e4d59c580ca10f301d53814b700da" -> (known after apply)
      ~ content_sha1         = "aa9f05f39211ea80c845af77b88de873f63b14af" -> (known after apply)
      ~ content_sha256       = "628b0a6aab12fcea6c15a66c5bd9202b926ef4b717b118a591a37f90faf6b453" -> (known af
      ~ content_sha512       = "b6a5f1bbedbf7d93b00c5bf01331f14b6070c6810a735590f0775f4cf058aaaae62b8857e5e407
8bdc76d6586a53166" -> (known after apply)
      ~ id                   = "aa9f05f39211ea80c845af77b88de873f63b14af" -> (known after apply)
        # (4 unchanged attributes hidden)
```

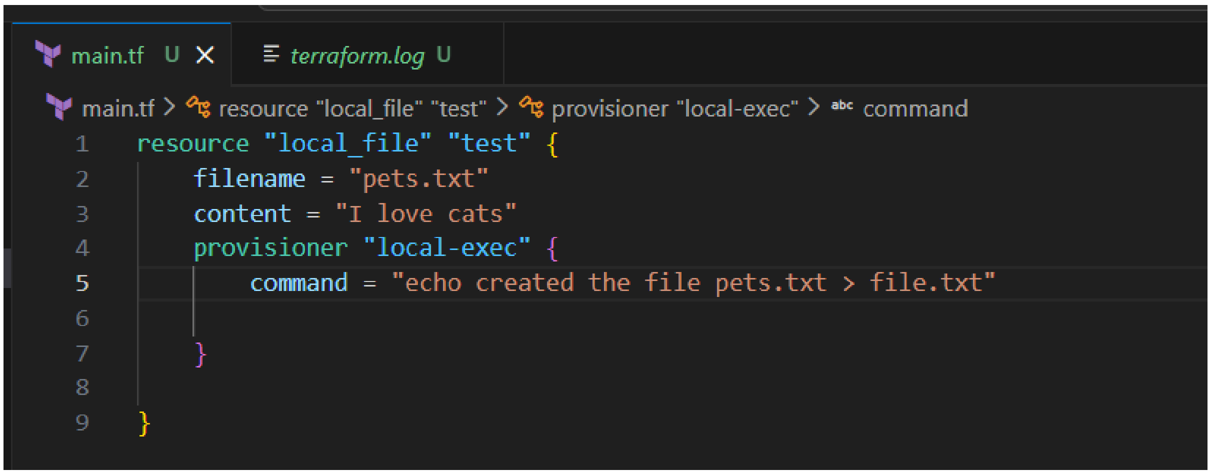


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

local_file.test: Destroying... [id=aa9f05f39211ea80c845af77b88de873f63b14af]
local_file.test: Destruction complete after 0s
local_file.test: Creating...
local_file.test: Provisioning with 'local-exec'...
local_file.test (local-exec): Executing: ["cmd" "/C" "echo created the file pets.txt > /t
local_file.test (local-exec): The system cannot find the path specified.
│
│ Error: local-exec provisioner error
│
│   with local_file.test,
│   on main.tf line 4, in resource "local_file" "test":
│    4:     provisioner "local-exec" {
│
│ Error running command 'echo created the file pets.txt > /tmp/file.txt': exit status 1.
│

PS C:\Users\Ashish\Desktop\Terraform basics> █
```

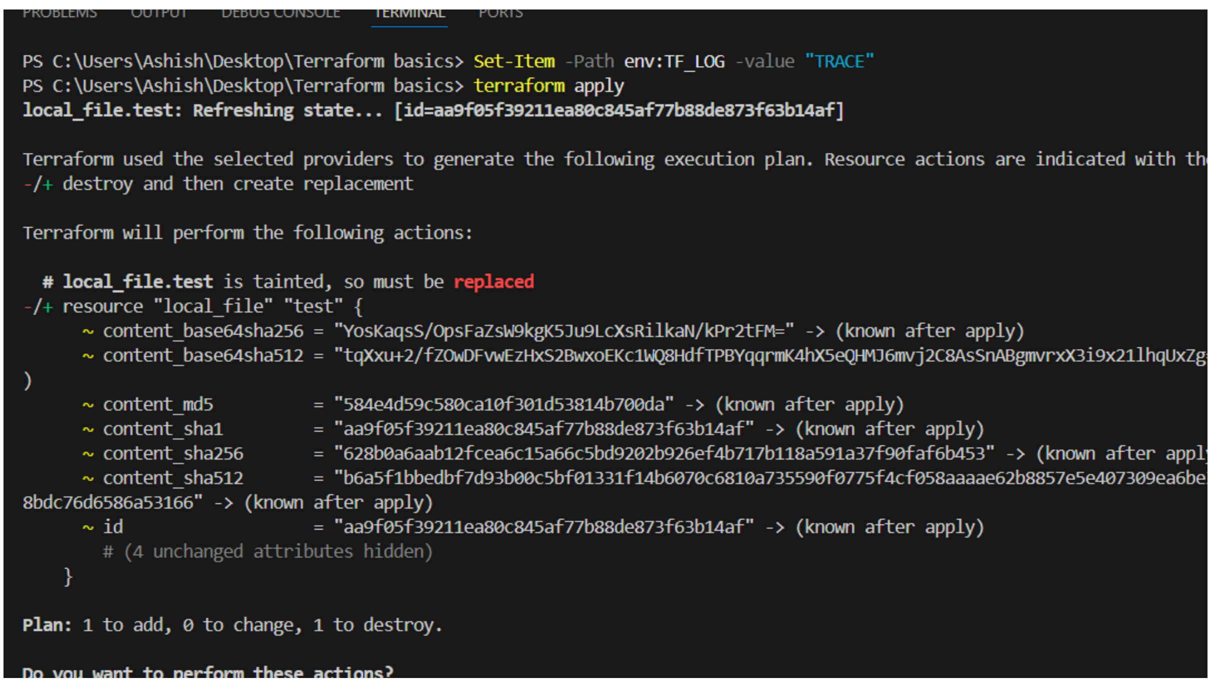


```
2025-11-14T19:59:41.818+0530 [ERROR] vertex "local_file.test" error: local-exec provisioner error
```

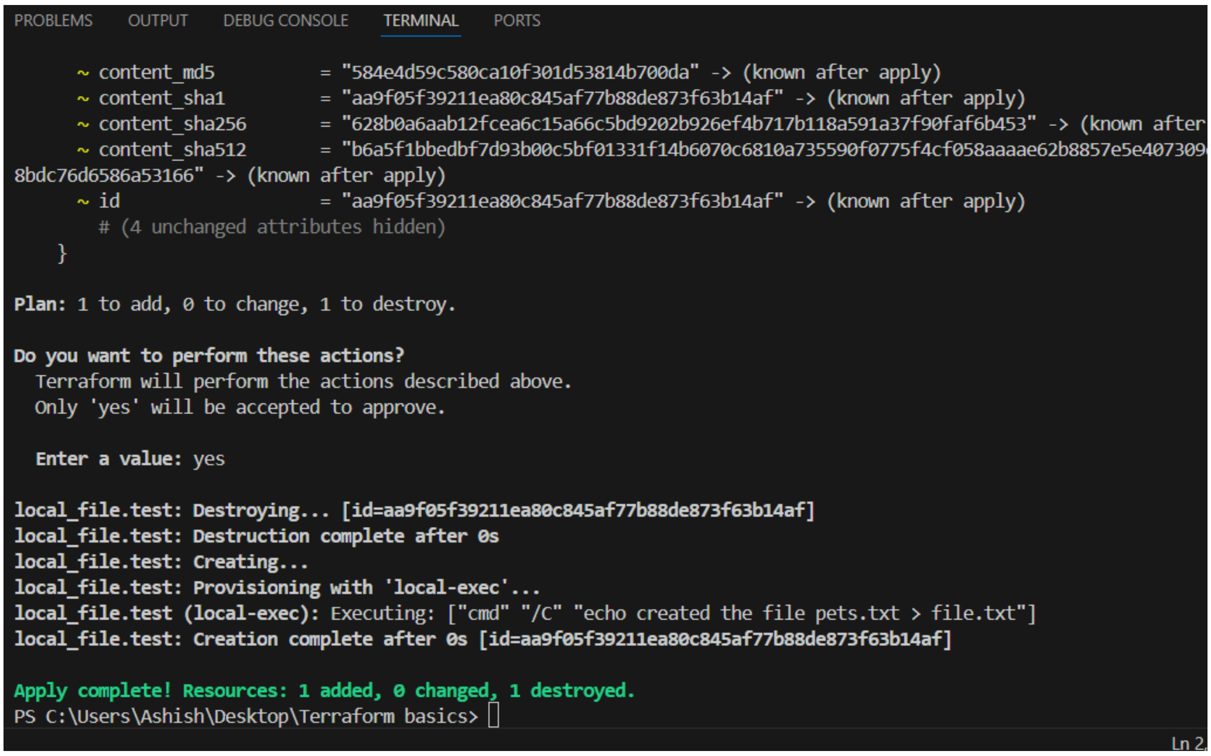

main.tf
terraform.log
main.tf > resource "local_file" "test" > provisioner "local-exec" > command

```
resource "local_file" "test" {
    filename = "pets.txt"
    content = "I love cats"
    provisioner "local-exec" {
        command = "echo created the file pets.txt > file.txt"

    }

}
```

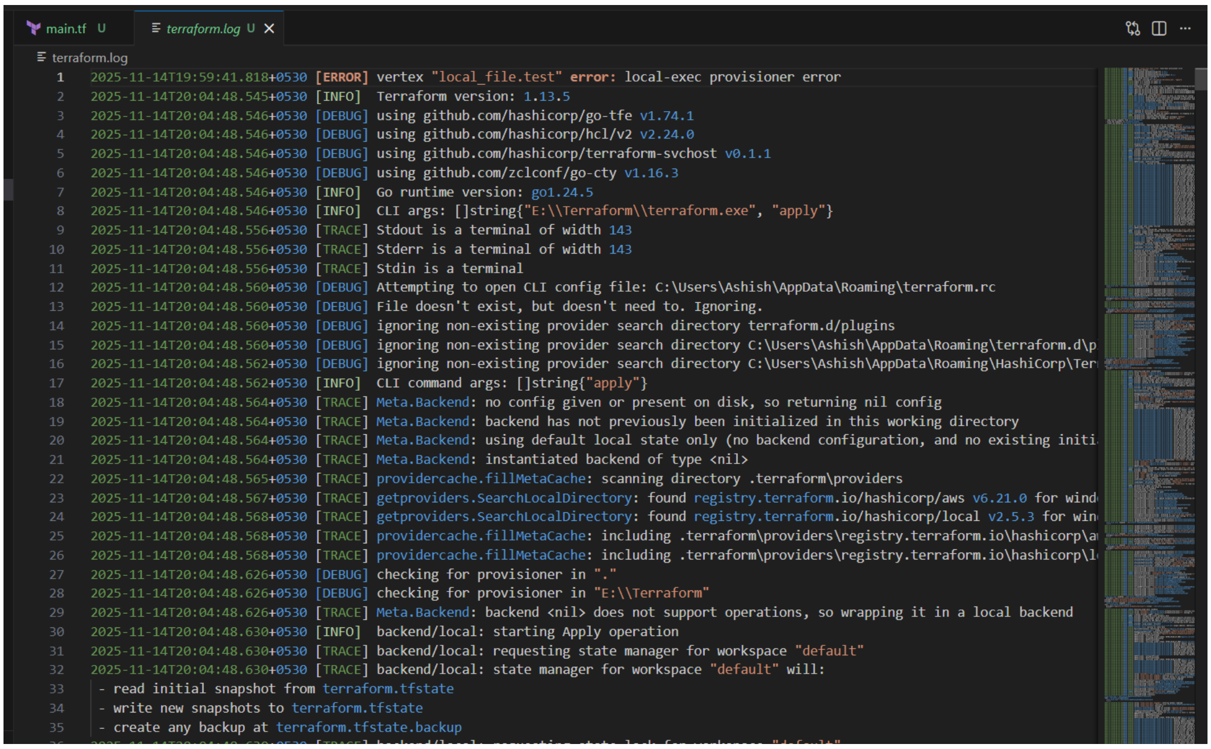
- Set-Item -Path env:TF_LOG -value "TRACE"
- terraform apply



```
PS C:\Users\Ashish\Desktop\Terraform basics> Set-Item -Path env:TF_LOG -value "TRACE"
PS C:\Users\Ashish\Desktop\Terraform basics> terraform apply
local_file.test: Refreshing state... [id=aa9f05f39211ea80c845af77b88de873f63b14af]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with th
-/+ destroy and then create replacement

Terraform will perform the following actions:

  # local_file.test is tainted, so must be replaced
-/+ resource "local_file" "test" {
      ~ content_base64sha256 = "YosKaqsS/OpsFaZsW9kgK5Ju9LcXsRilkaN/kPr2tFM=" -> (known after apply)
      ~ content_base64sha512 = "tqXxu+2/fZOwDFvwEzHxS2BwxoEKc1WQ8HdfTPBYqqrmK4hX5eQHMJ6mvj2C8AsSnABgmvrxX3i9x21lhqUxZg
)
      ~ content_md5          = "584e4d59c580ca10f301d53814b700da" -> (known after apply)
      ~ content_sha1         = "aa9f05f39211ea80c845af77b88de873f63b14af" -> (known after apply)
      ~ content_sha256       = "628b0a6aab12fcea6c15a66c5bd9202b926ef4b717b118a591a37f90faf6b453" -> (known after appl
      ~ content_sha512       = "b6a5f1bbedbf7d93b00c5bf01331f14b6070c6810a735590f0775f4cf058aaaae62b8857e5e407309ea6be
8bdc76d6586a53166" -> (known after apply)
      ~ id                   = "aa9f05f39211ea80c845af77b88de873f63b14af" -> (known after apply)
        # (4 unchanged attributes hidden)
    }

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

Do you want to perform these actions?
```

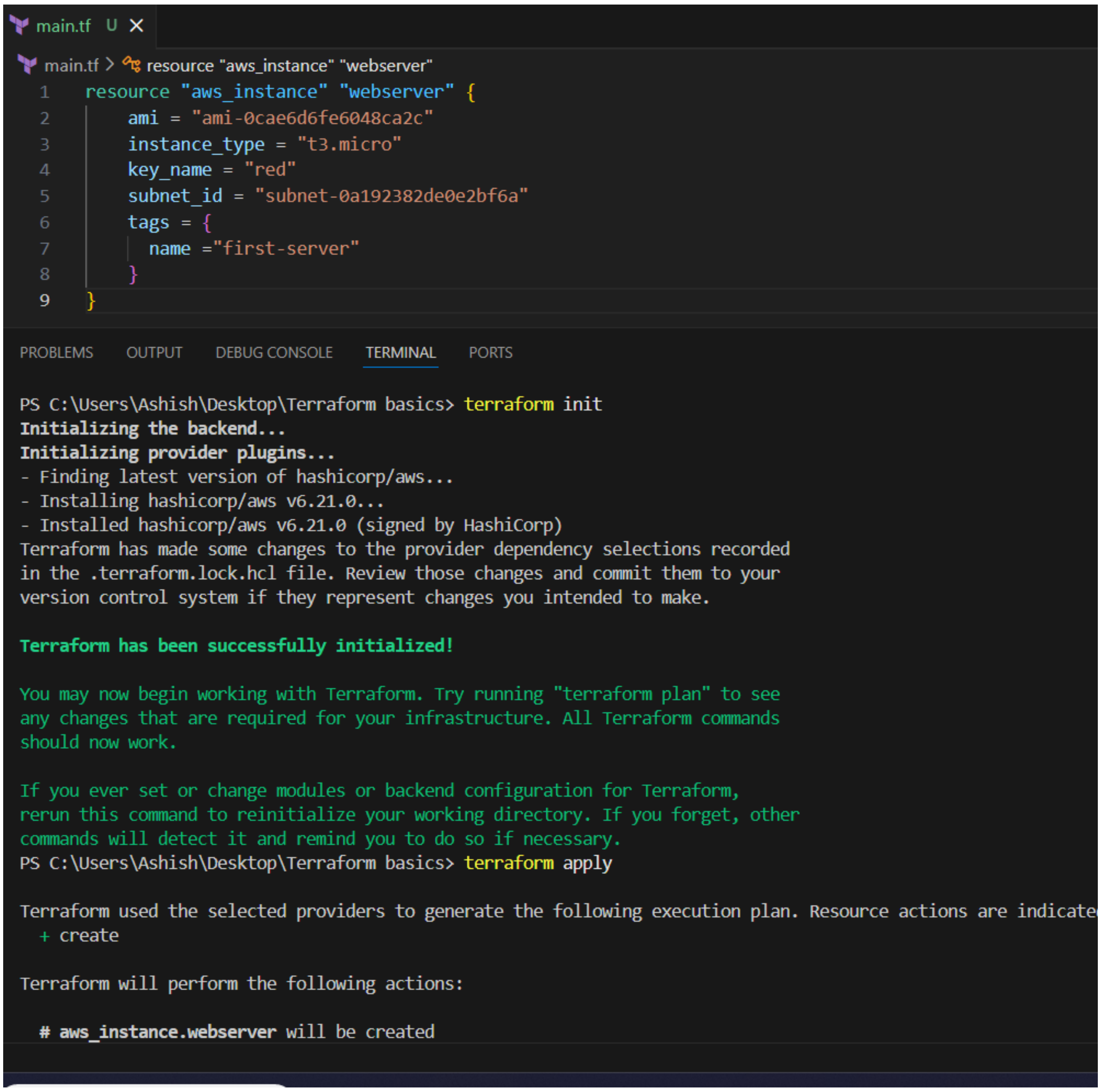


```
      ~ content_md5          = "584e4d59c580ca10f301d53814b700da" -> (known after apply)
      ~ content_sha1         = "aa9f05f39211ea80c845af77b88de873f63b14af" -> (known after apply)
      ~ content_sha256       = "628b0a6aab12fcea6c15a66c5bd9202b926ef4b717b118a591a37f90faf6b453" -> (known after
      ~ content_sha512       = "b6a5f1bbedbf7d93b00c5bf01331f14b6070c6810a735590f0775f4cf058aaaae62b8857e5e407309
8bdc76d6586a53166" -> (known after apply)
      ~ id                   = "aa9f05f39211ea80c845af77b88de873f63b14af" -> (known after apply)
        # (4 unchanged attributes hidden)
    }

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

local_file.test: Destroying... [id=aa9f05f39211ea80c845af77b88de873f63b14af]
local_file.test: Destruction complete after 0s
local_file.test: Creating...
local_file.test: Provisioning with 'local-exec'...
local_file.test (local-exec): Executing: ["cmd" "/C" "echo created the file pets.txt > file.txt"]
local_file.test: Creation complete after 0s [id=aa9f05f39211ea80c845af77b88de873f63b14af]

Apply complete! Resources: 1 added, 0 changed, 1 destroyed.
PS C:\Users\Ashish\Desktop\Terraform basics> 
```



```
2025-11-14T19:59:41.818+0530 [ERROR] vertex "local_file.test" error: local-exec provisioner error
2025-11-14T20:04:48.545+0530 [INFO]  Terraform version: 1.13.5
2025-11-14T20:04:48.546+0530 [DEBUG] using github.com/hashicorp/go-tfe v1.74.1
2025-11-14T20:04:48.546+0530 [DEBUG] using github.com/hashicorp/hcl/v2 v2.24.0
2025-11-14T20:04:48.546+0530 [DEBUG] using github.com/hashicorp/terraform-svchost v0.1.1
2025-11-14T20:04:48.546+0530 [DEBUG] using github.com/zclconf/go-cty v1.16.3
2025-11-14T20:04:48.546+0530 [INFO]  Go runtime version: go1.24.5
2025-11-14T20:04:48.546+0530 [INFO]  CLI args: []string{"E:\\Terraform\\terraform.exe", "apply"}
2025-11-14T20:04:48.556+0530 [TRACE] Stdout is a terminal of width 143
2025-11-14T20:04:48.556+0530 [TRACE] Stderr is a terminal of width 143
2025-11-14T20:04:48.556+0530 [TRACE] Stdin is a terminal
2025-11-14T20:04:48.560+0530 [DEBUG] Attempting to open CLI config file: C:\Users\Ashish\AppData\Roaming\terraform.rc
2025-11-14T20:04:48.560+0530 [DEBUG] File doesn't exist, but doesn't need to. Ignoring.
2025-11-14T20:04:48.560+0530 [DEBUG] ignoring non-existing provider search directory terraform.d/plugins
2025-11-14T20:04:48.560+0530 [DEBUG] ignoring non-existing provider search directory C:\Users\Ashish\AppData\Roaming\terraform.d\p
2025-11-14T20:04:48.562+0530 [DEBUG] ignoring non-existing provider search directory C:\Users\Ashish\AppData\Roaming\HashiCorp\Ter
2025-11-14T20:04:48.562+0530 [INFO]  CLI command args: []string{"apply"}
2025-11-14T20:04:48.564+0530 [TRACE] Meta.Backend: no config given or present on disk, so returning nil config
2025-11-14T20:04:48.564+0530 [TRACE] Meta.Backend: backend has not previously been initialized in this working directory
2025-11-14T20:04:48.564+0530 [TRACE] Meta.Backend: using default local state only (no backend configuration, and no existing initi
2025-11-14T20:04:48.564+0530 [TRACE] Meta.Backend: instantiated backend of type <nil>
2025-11-14T20:04:48.565+0530 [TRACE] providercache.fillMetaCache: scanning directory .terraform\providers
2025-11-14T20:04:48.567+0530 [TRACE] getproviders.SearchLocalDirectory: found registry.terraform.io/hashicorp/aws v6.21.0 for wind
2025-11-14T20:04:48.568+0530 [TRACE] getproviders.SearchLocalDirectory: found registry.terraform.io/hashicorp/local v2.5.3 for win
2025-11-14T20:04:48.568+0530 [TRACE] providercache.fillMetaCache: including .terraform\providers\registry.terraform.io\hashicorp\a
2025-11-14T20:04:48.568+0530 [TRACE] providercache.fillMetaCache: including .terraform\providers\registry.terraform.io\hashicorp\l
2025-11-14T20:04:48.626+0530 [DEBUG] checking for provisioner in "."
2025-11-14T20:04:48.626+0530 [DEBUG] checking for provisioner in "E:\\Terraform"
2025-11-14T20:04:48.626+0530 [TRACE] Meta.Backend: backend <nil> does not support operations, so wrapping it in a local backend
2025-11-14T20:04:48.630+0530 [INFO]  backend/local: starting Apply operation
2025-11-14T20:04:48.630+0530 [TRACE] backend/local: requesting state manager for workspace "default"
2025-11-14T20:04:48.630+0530 [TRACE] backend/local: state manager for workspace "default" will:
  - read initial snapshot from terraform.tfstate
  - write new snapshots to terraform.tfstate
  - create any backup at terraform.tfstate.backup
```

Create an instance.



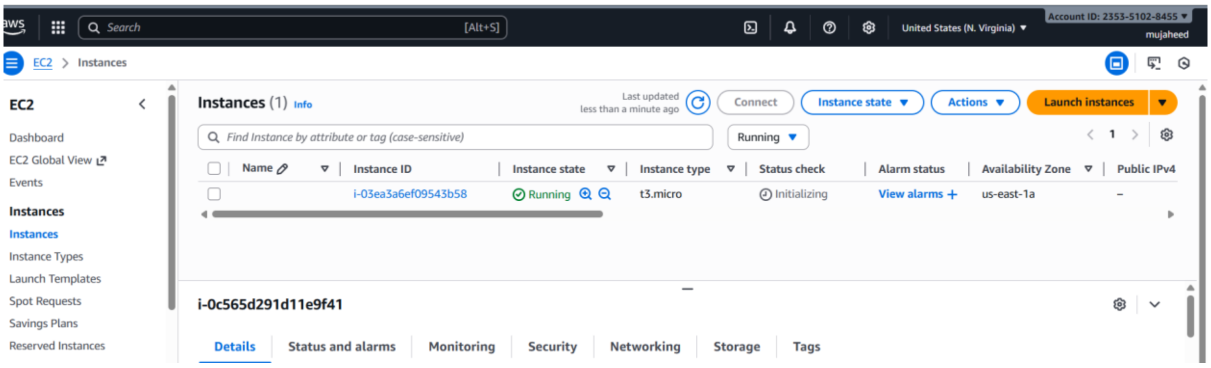
main.tf U ×

main.tf > resource "aws_instance" "webserver"

```
resource "aws_instance" "webserver" {
    ami = "ami-0cae6d6fe6048ca2c"
    instance_type = "t3.micro"
    key_name = "red"
    subnet_id = "subnet-0a192382de0e2bf6a"
    tags = {
      name ="first-server"
    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Ashish\Desktop\Terraform basics> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v6.21.0...
- Installed hashicorp/aws v6.21.0 (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.

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.
PS C:\Users\Ashish\Desktop\Terraform basics> terraform apply

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

Terraform will perform the following actions:

  # aws_instance.webserver will be created
```



aws Search [Alt+S] United States (N. Virginia) Account ID: 2353-5102-8455 mujaheed

EC2 > Instances

EC2: Dashboard, EC2 Global View, Events, Instances, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances

Instances (1) Info — Last updated less than a minute ago — Connect — Instance state — Actions — Launch instances

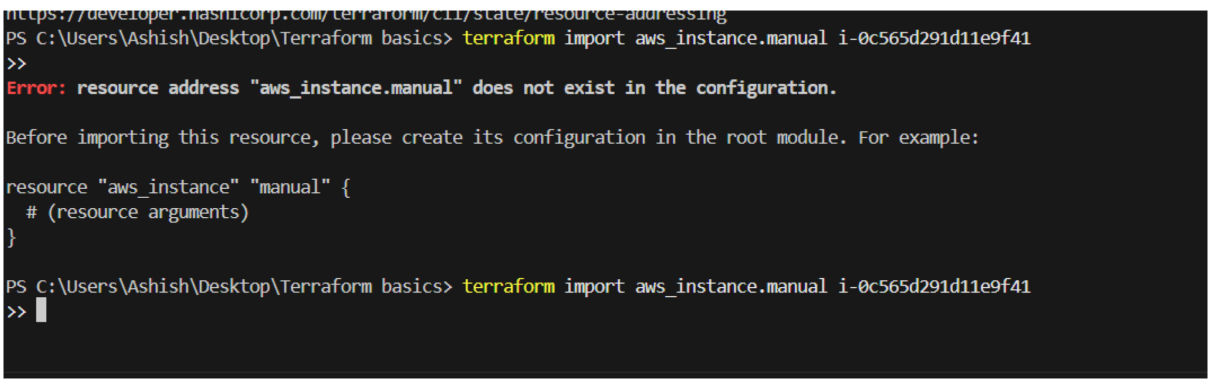
Find Instance by attribute or tag (case-sensitive) — Running

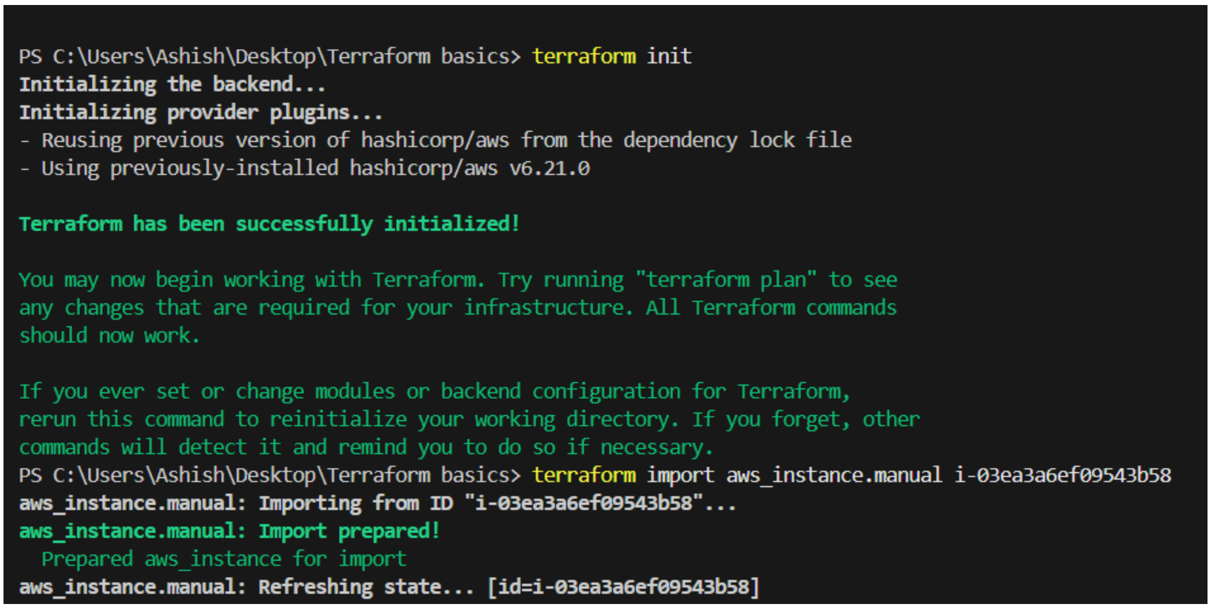
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
		i-03ea3a6ef09543b58	Running	t3.micro	Initializing	View alarms +	us-east-1a	–

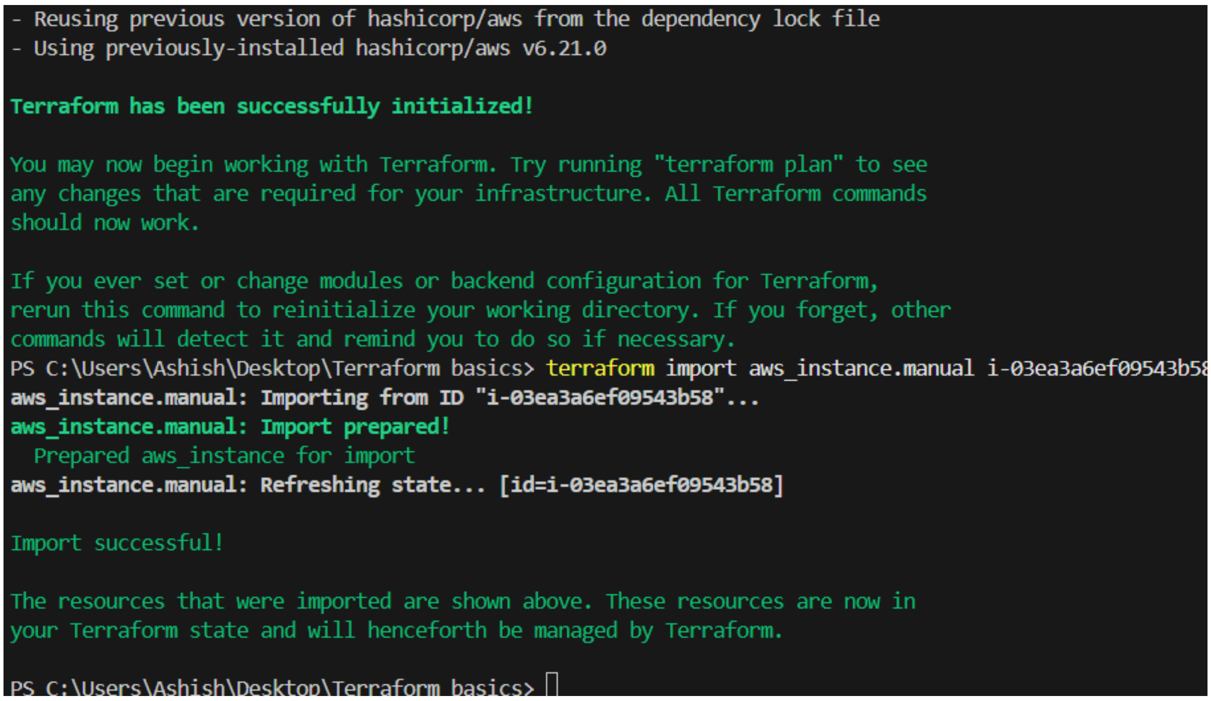
i-0c565d291d11e9f41

Details | Status and alarms | Monitoring | Security | Networking | Storage | Tags

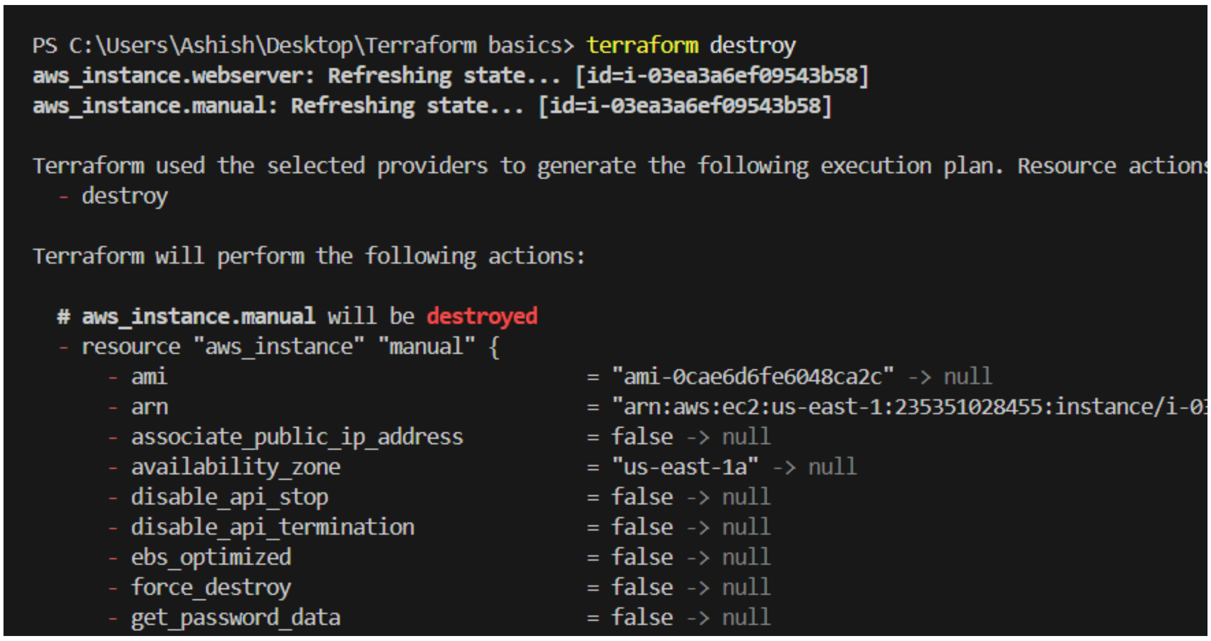
- terraform import aws_instance.<name> instance_id


```
https://developer.hashicorp.com/terraform/cli/state/resource-addressing
PS C:\Users\Ashish\Desktop\Terraform basics> terraform import aws_instance.manual i-0c565d291d11e9f41
>>
Error: resource address "aws_instance.manual" does not exist in the configuration.

Before importing this resource, please create its configuration in the root module. For example:

resource "aws_instance" "manual" {
  # (resource arguments)
}

PS C:\Users\Ashish\Desktop\Terraform basics> terraform import aws_instance.manual i-0c565d291d11e9f41
>>
```

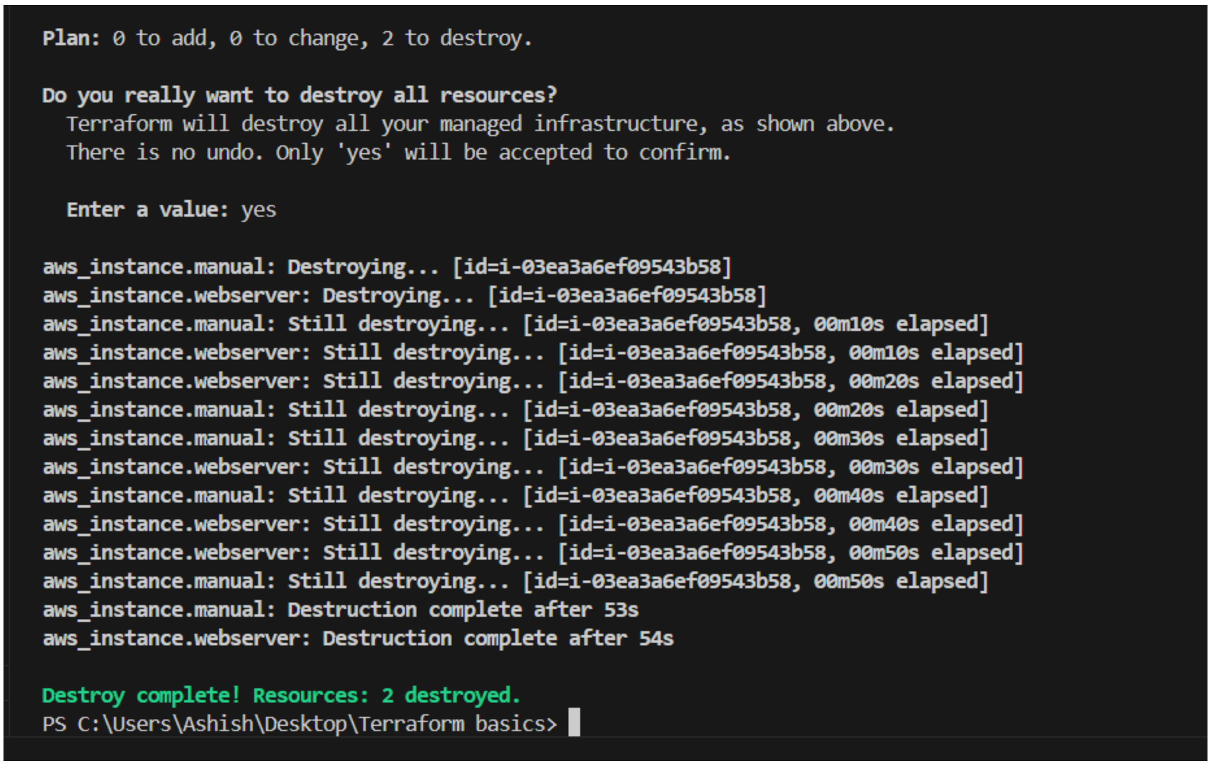

```
PS C:\Users\Ashish\Desktop\Terraform basics> terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v6.21.0

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.
PS C:\Users\Ashish\Desktop\Terraform basics> terraform import aws_instance.manual i-03ea3a6ef09543b58
aws_instance.manual: Importing from ID "i-03ea3a6ef09543b58"...
aws_instance.manual: Import prepared!
  Prepared aws_instance for import
aws_instance.manual: Refreshing state... [id=i-03ea3a6ef09543b58]
```


```
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v6.21.0

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.
PS C:\Users\Ashish\Desktop\Terraform basics> terraform import aws_instance.manual i-03ea3a6ef09543b58
aws_instance.manual: Importing from ID "i-03ea3a6ef09543b58"...
aws_instance.manual: Import prepared!
  Prepared aws_instance for import
aws_instance.manual: Refreshing state... [id=i-03ea3a6ef09543b58]

Import successful!

The resources that were imported are shown above. These resources are now in
your Terraform state and will henceforth be managed by Terraform.

PS C:\Users\Ashish\Desktop\Terraform basics>
```

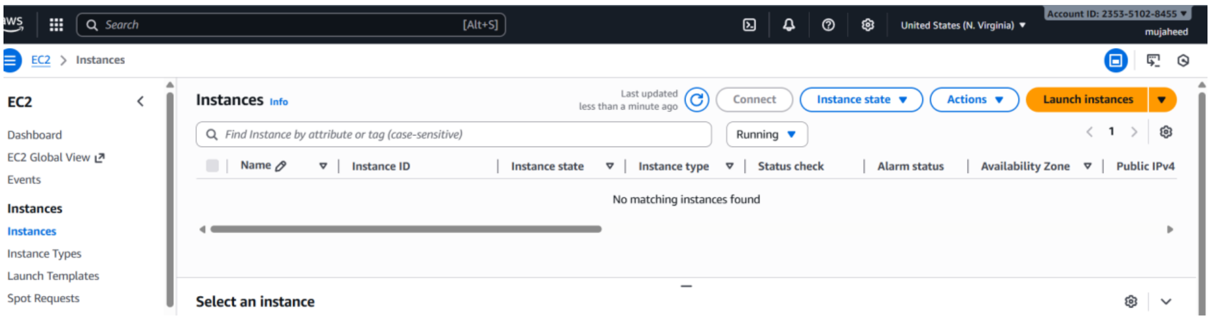
If you do terraform destroy instance will be deleted.



```
PS C:\Users\Ashish\Desktop\Terraform basics> terraform destroy
aws_instance.webserver: Refreshing state... [id=i-03ea3a6ef09543b58]
aws_instance.manual: Refreshing state... [id=i-03ea3a6ef09543b58]

Terraform used the selected providers to generate the following execution plan. Resource action
  - destroy

Terraform will perform the following actions:

  # aws_instance.manual will be destroyed
  - resource "aws_instance" "manual" {
      - ami                                  = "ami-0cae6d6fe6048ca2c" -> null
      - arn                                  = "arn:aws:ec2:us-east-1:235351028455:instance/i-0
      - associate_public_ip_address          = false -> null
      - availability_zone                    = "us-east-1a" -> null
      - disable_api_stop                     = false -> null
      - disable_api_termination              = false -> null
      - ebs_optimized                        = false -> null
      - force_destroy                        = false -> null
      - get_password_data                    = false -> null
```



```
Plan: 0 to add, 0 to change, 2 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.manual: Destroying... [id=i-03ea3a6ef09543b58]
aws_instance.webserver: Destroying... [id=i-03ea3a6ef09543b58]
aws_instance.manual: Still destroying... [id=i-03ea3a6ef09543b58, 00m10s elapsed]
aws_instance.webserver: Still destroying... [id=i-03ea3a6ef09543b58, 00m10s elapsed]
aws_instance.webserver: Still destroying... [id=i-03ea3a6ef09543b58, 00m20s elapsed]
aws_instance.manual: Still destroying... [id=i-03ea3a6ef09543b58, 00m20s elapsed]
aws_instance.manual: Still destroying... [id=i-03ea3a6ef09543b58, 00m30s elapsed]
aws_instance.webserver: Still destroying... [id=i-03ea3a6ef09543b58, 00m30s elapsed]
aws_instance.manual: Still destroying... [id=i-03ea3a6ef09543b58, 00m40s elapsed]
aws_instance.webserver: Still destroying... [id=i-03ea3a6ef09543b58, 00m40s elapsed]
aws_instance.webserver: Still destroying... [id=i-03ea3a6ef09543b58, 00m50s elapsed]
aws_instance.manual: Still destroying... [id=i-03ea3a6ef09543b58, 00m50s elapsed]
aws_instance.manual: Destruction complete after 53s
aws_instance.webserver: Destruction complete after 54s

Destroy complete! Resources: 2 destroyed.
PS C:\Users\Ashish\Desktop\Terraform basics>
```



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4

No matching instances found

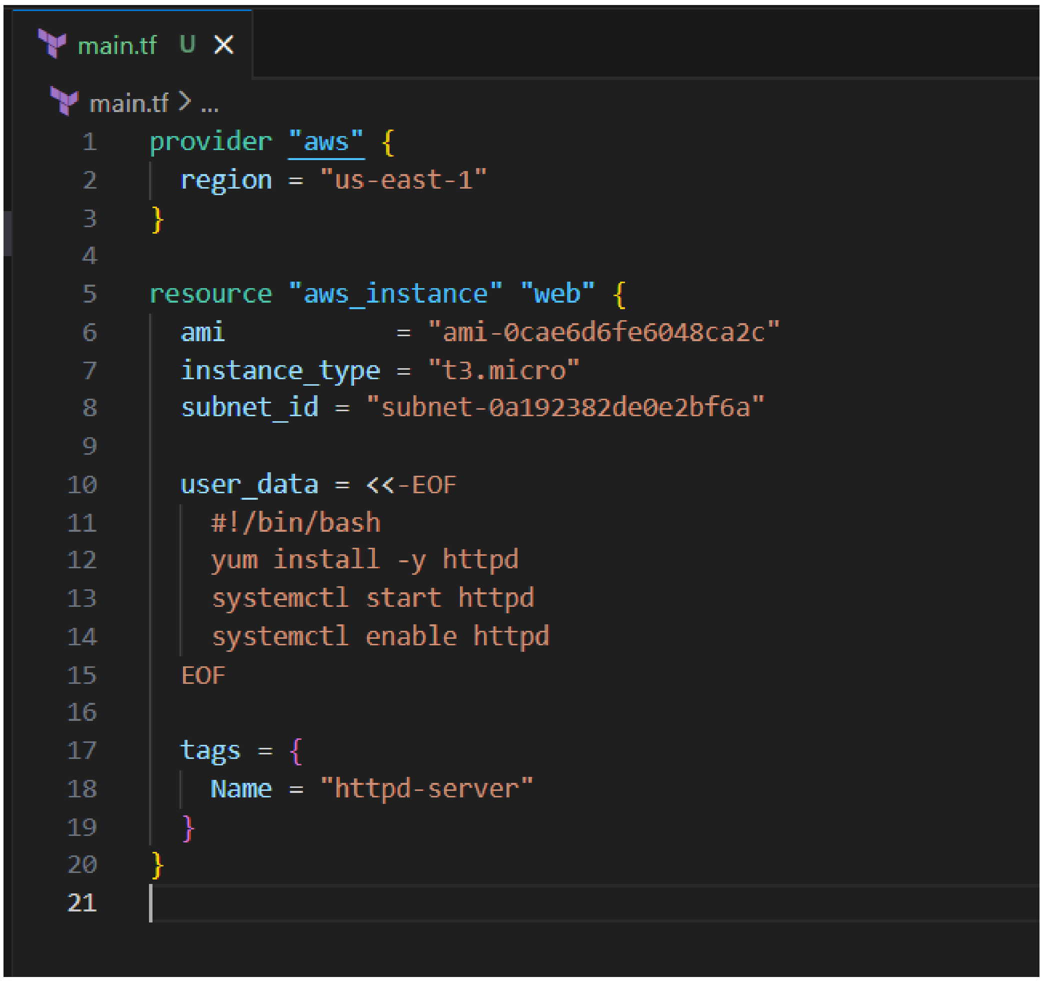
Select an instance

3. Create one EC2 instance with httpd installed using a Terraform script.

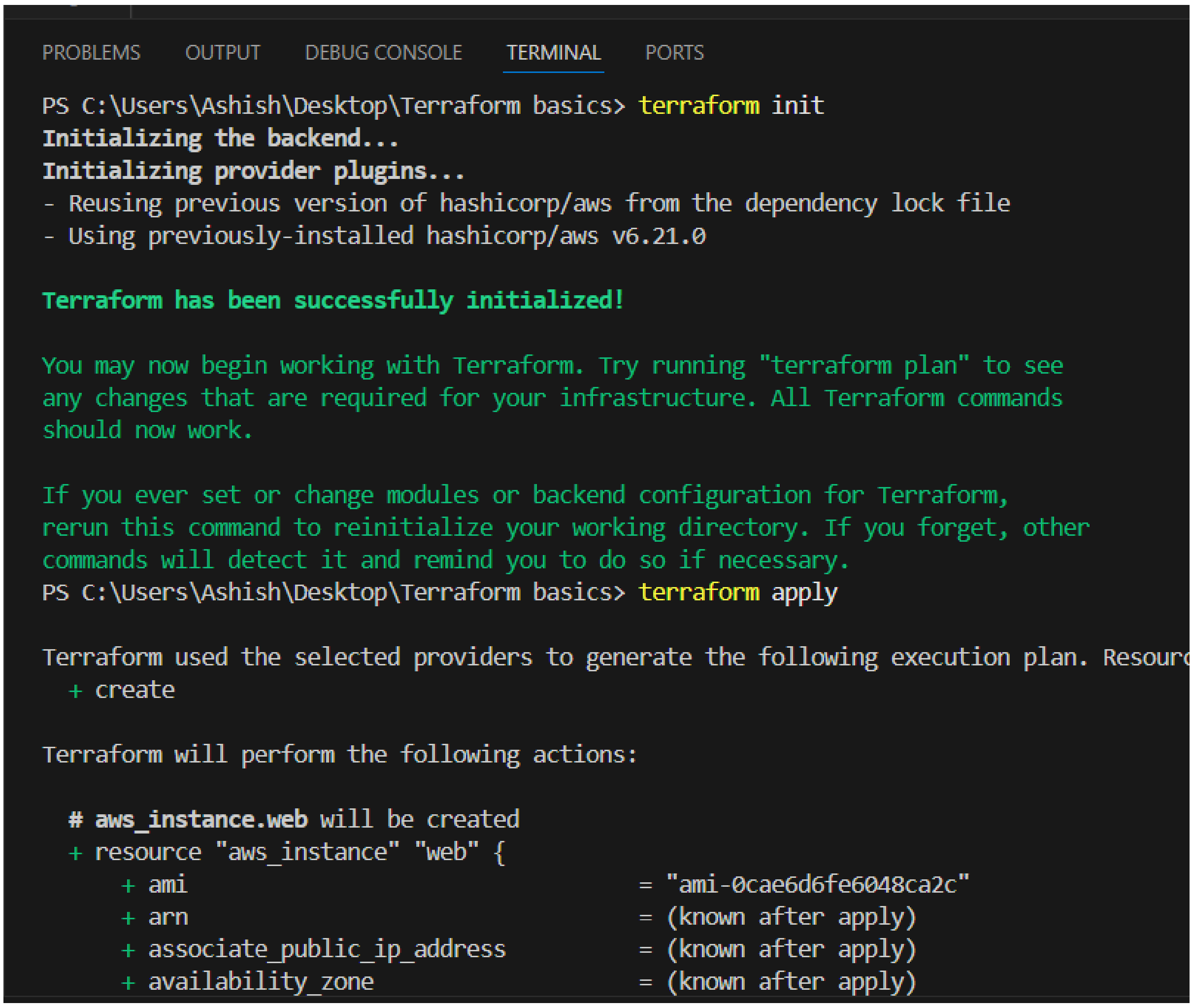
```
provider "aws" {
  region = "us-east-1"
}

resource "aws_instance" "web" {
  ami           = "ami-0cae6d6fe6048ca2c"
  instance_type = "t3.micro"
  subnet_id = "subnet-0a192382de0e2bf6a"

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

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

}

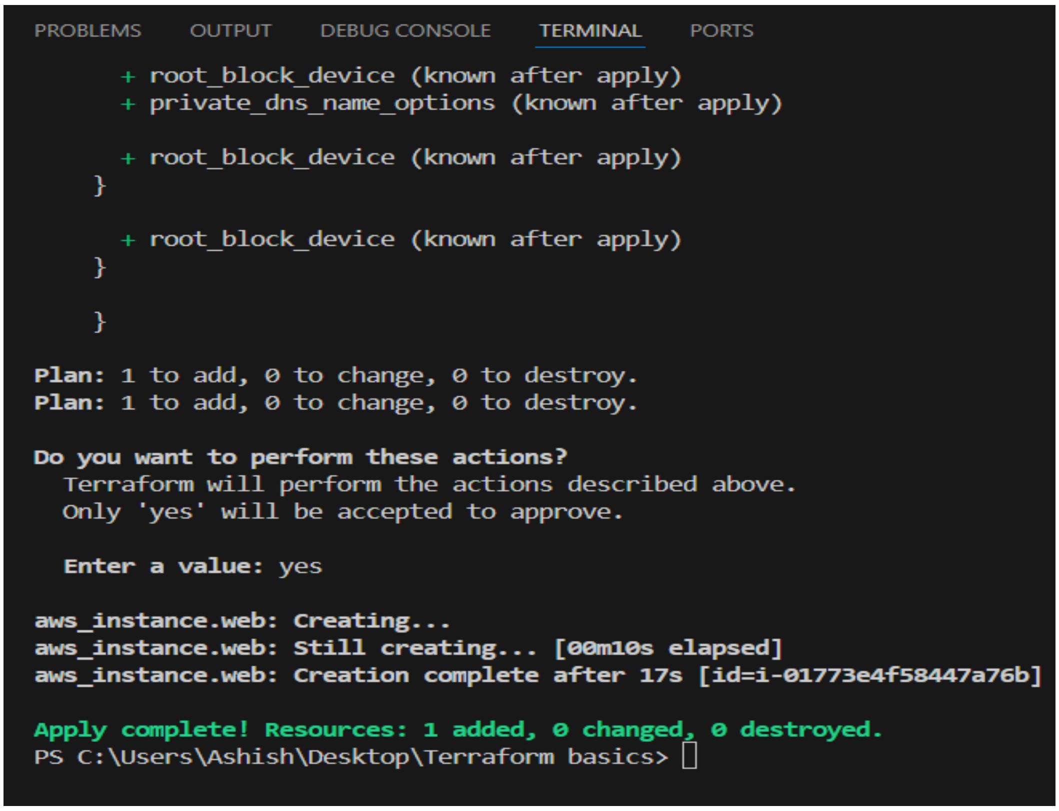
- terraform init
- terraform apply



```
provider "aws" {
  region = "us-east-1"
}

resource "aws_instance" "web" {
  ami           = "ami-0cae6d6fe6048ca2c"
  instance_type = "t3.micro"
  subnet_id = "subnet-0a192382de0e2bf6a"

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

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

```

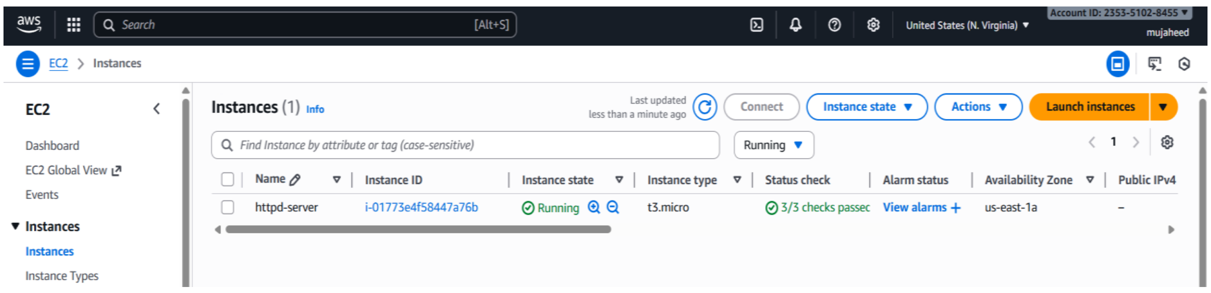


```
PS C:\Users\Ashish\Desktop\Terraform basics> terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v6.21.0

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.
PS C:\Users\Ashish\Desktop\Terraform basics> terraform apply

Terraform used the selected providers to generate the following execution plan. Resourc
  + create

Terraform will perform the following actions:

  # aws_instance.web will be created
  + resource "aws_instance" "web" {
      + ami                                  = "ami-0cae6d6fe6048ca2c"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
```



```
      + root_block_device (known after apply)
      + private_dns_name_options (known after apply)

      + root_block_device (known after apply)
    }

      + root_block_device (known after apply)
    }

    }

Plan: 1 to add, 0 to change, 0 to destroy.
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... [00m10s elapsed]
aws_instance.web: Creation complete after 17s [id=i-01773e4f58447a76b]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\Users\Ashish\Desktop\Terraform basics> 
```

An instance will be created by using terraform with the userdata httpd init.

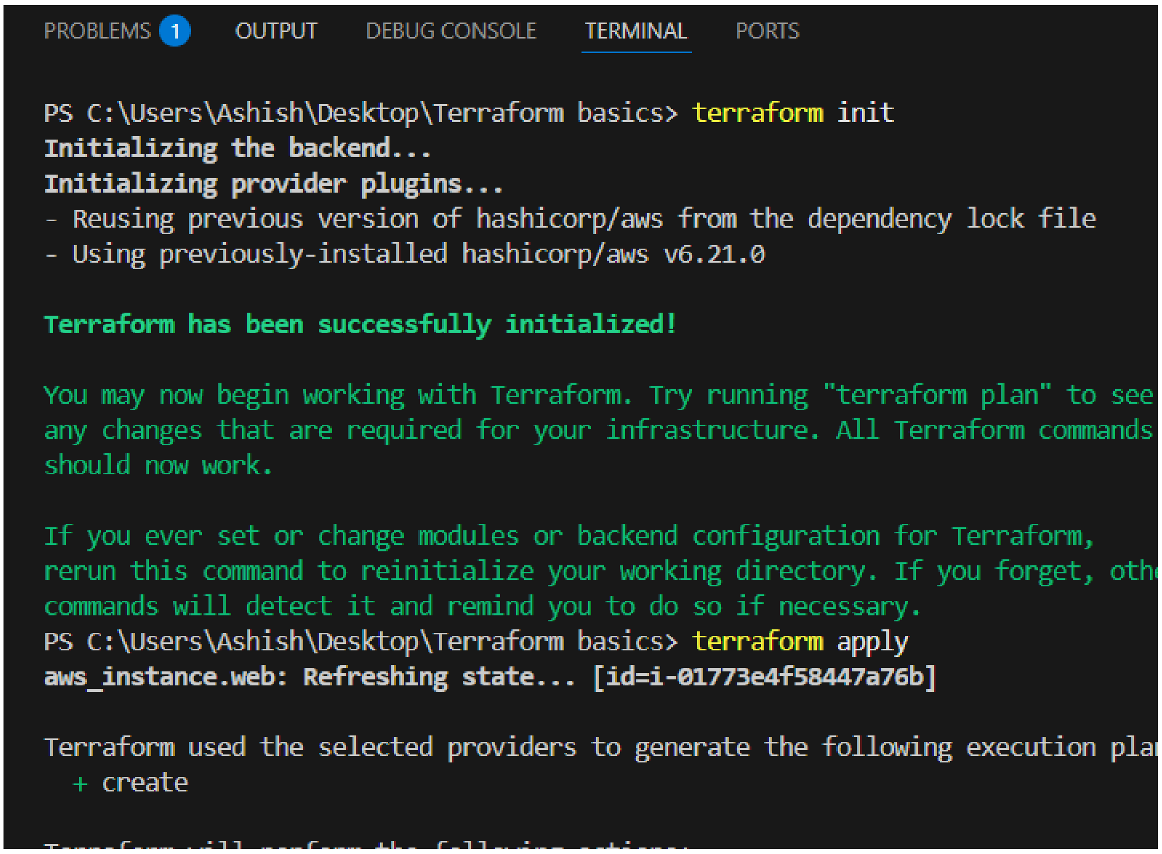


Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
httpd-server	i-01773e4f58447a76b	Running	t3.micro	3/3 checks passec	View alarms +	us-east-1a	–

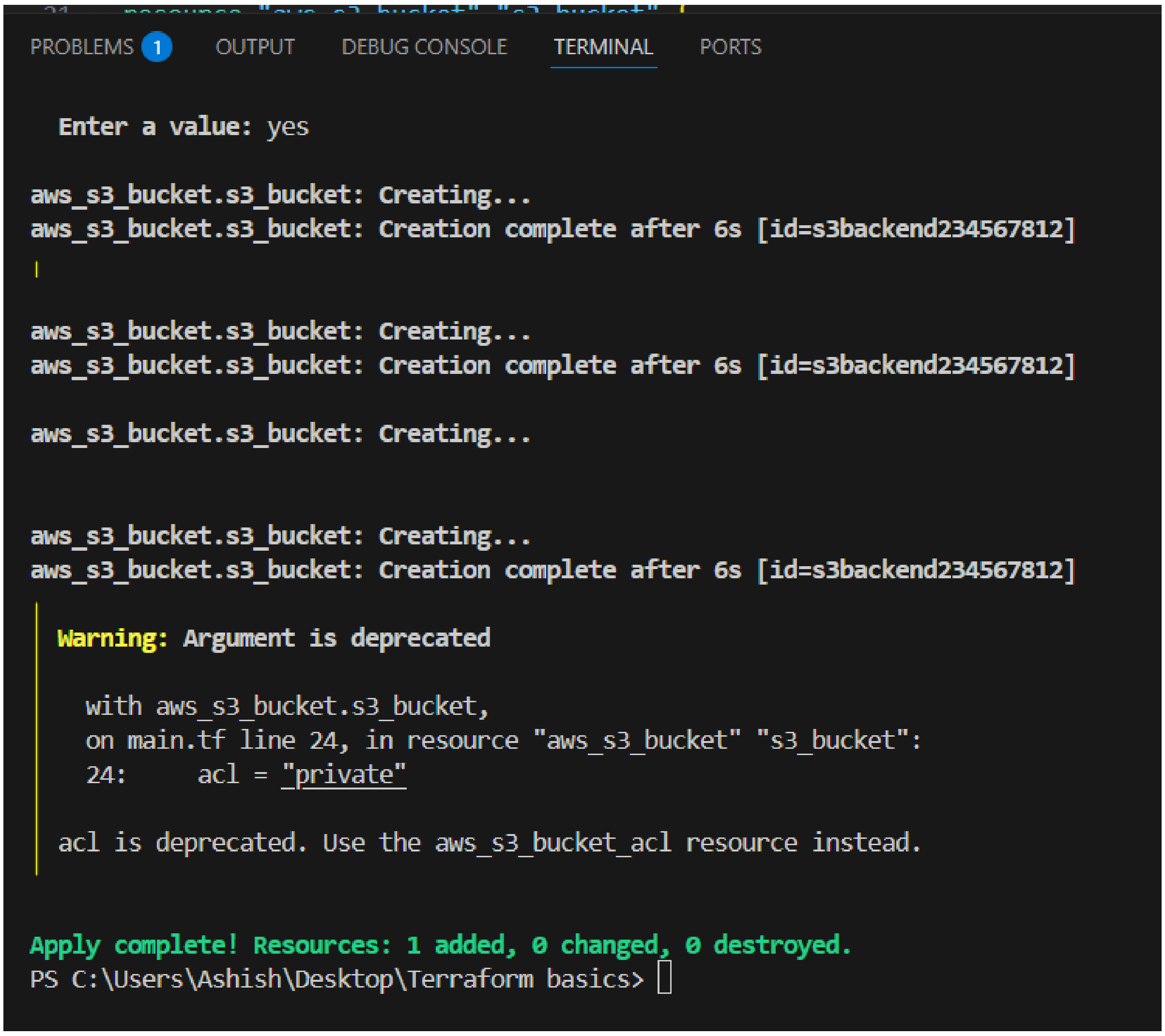
4. Set up S3 as backend for task 3.

By adding resource to that script we can create a s3 bucket as backend.

```
provider "aws" {
  region = "us-east-1"
}


resource "aws_instance" "web" {
  ami           = "ami-0cae6d6fe6048ca2c"
  instance_type = "t3.micro"
  subnet_id = "subnet-0a192382de0e2bf6a"


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

    systemctl enable httpd

  EOF


  tags = {

    Name = "httpd-server"

  }

}

resource "aws_s3_bucket" "s3_bucket" {


    bucket = "s3backend234567812"

    acl = "private"

}
```

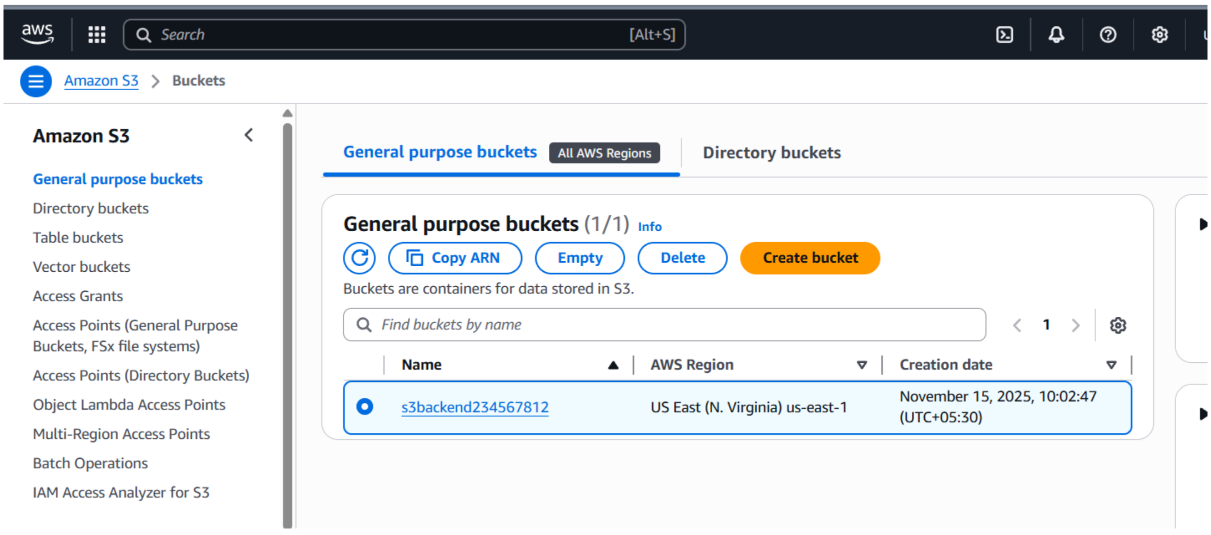


```
PS C:\Users\Ashish\Desktop\Terraform basics> terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v6.21.0

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, othe
commands will detect it and remind you to do so if necessary.
PS C:\Users\Ashish\Desktop\Terraform basics> terraform apply
aws_instance.web: Refreshing state... [id=i-01773e4f58447a76b]

Terraform used the selected providers to generate the following execution plan
  + create
```



```
  Enter a value: yes

aws_s3_bucket.s3_bucket: Creating...
aws_s3_bucket.s3_bucket: Creation complete after 6s [id=s3backend234567812]


aws_s3_bucket.s3_bucket: Creating...
aws_s3_bucket.s3_bucket: Creation complete after 6s [id=s3backend234567812]

aws_s3_bucket.s3_bucket: Creating...


aws_s3_bucket.s3_bucket: Creating...
aws_s3_bucket.s3_bucket: Creation complete after 6s [id=s3backend234567812]

  Warning: Argument is deprecated

    with aws_s3_bucket.s3_bucket,
    on main.tf line 24, in resource "aws_s3_bucket" "s3_bucket":
    24:     acl = "private"

  acl is deprecated. Use the aws_s3_bucket_acl resource instead.


Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\Users\Ashish\Desktop\Terraform basics>
```

A bucket has been created as backend for that httpd server.



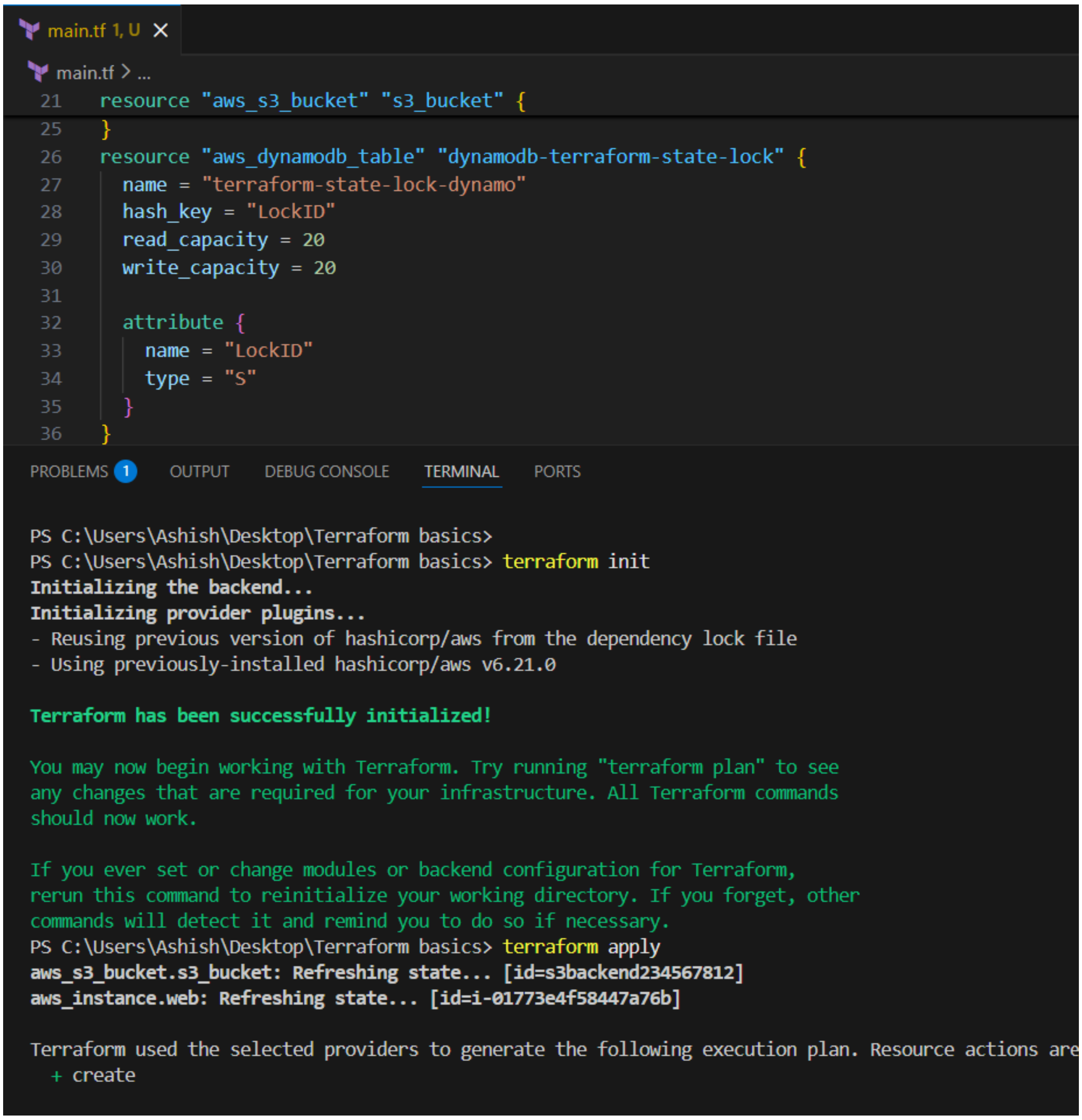
Buckets are containers for data stored in S3.

Name	AWS Region	Creation date
s3backend234567812	US East (N. Virginia) us-east-1	November 15, 2025, 10:02:47 (UTC+05:30)

5. Set up DynamoDB locking for task 3.

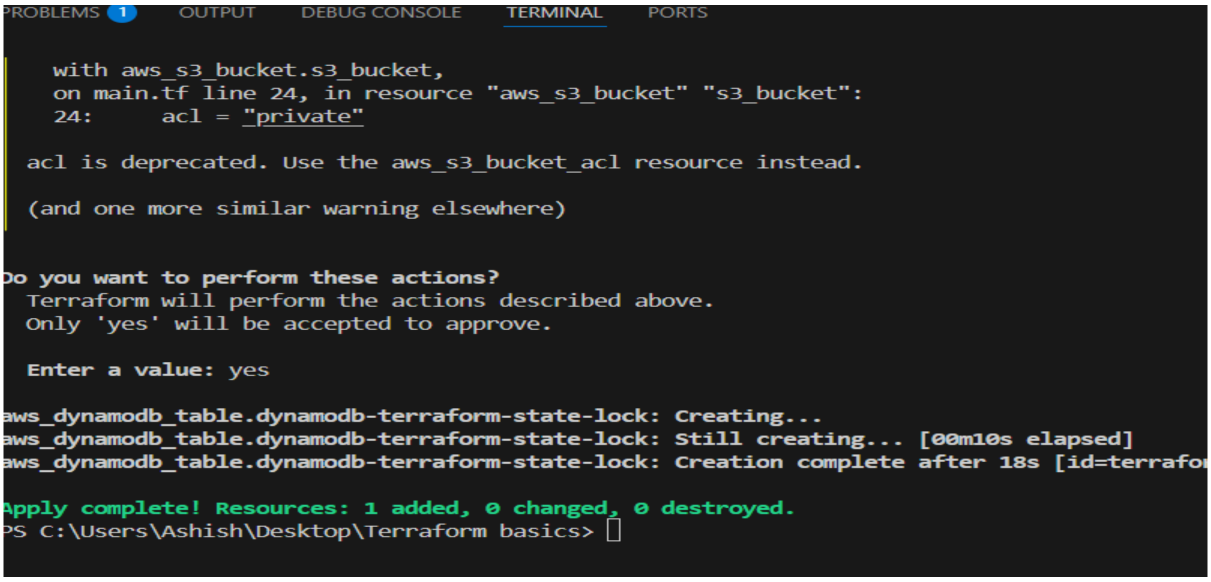
Creating dynamodb for the httpd and s3 backend.

```
provider "aws" {
  region = "us-east-1"
}


resource "aws_instance" "web" {
  ami           = "ami-0cae6d6fe6048ca2c"
  instance_type = "t3.micro"
  subnet_id = "subnet-0a192382de0e2bf6a"


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


  tags = {
    Name = "httpd-server"
  }
}
resource "aws_s3_bucket" "s3_bucket" {

   bucket = "s3backend234567812"
   acl = "private"
}
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"
  }
}
```



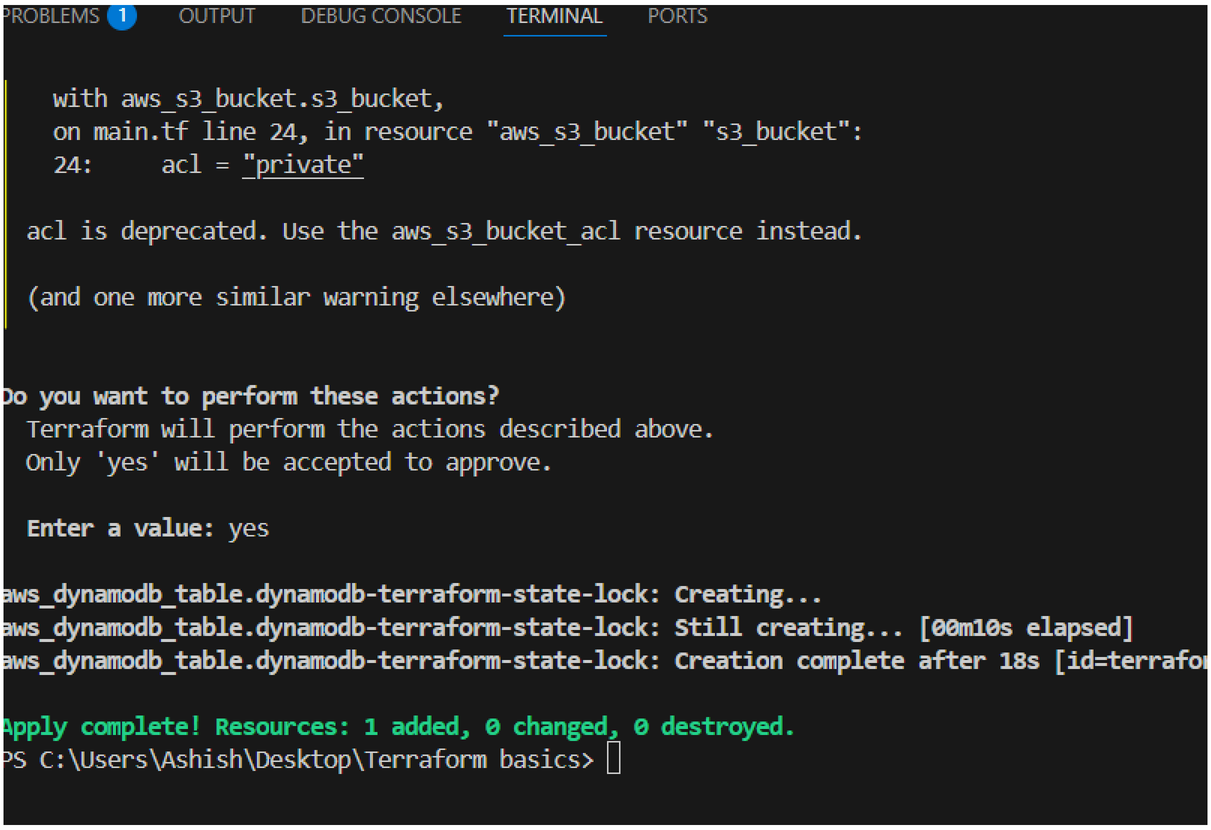
```
resource "aws_s3_bucket" "s3_bucket" {
}
resource "aws_dynamodb_table" "dynamodb-terraform-state-lock" {
  name = "terraform-state-lock-dynamo"
  hash_key = "LockID"
  read_capacity = 20
  write_capacity = 20

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

```
PS C:\Users\Ashish\Desktop\Terraform basics>
PS C:\Users\Ashish\Desktop\Terraform basics> terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v6.21.0

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.
PS C:\Users\Ashish\Desktop\Terraform basics> terraform apply
aws_s3_bucket.s3_bucket: Refreshing state... [id=s3backend234567812]
aws_instance.web: Refreshing state... [id=i-01773e4f58447a76b]

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

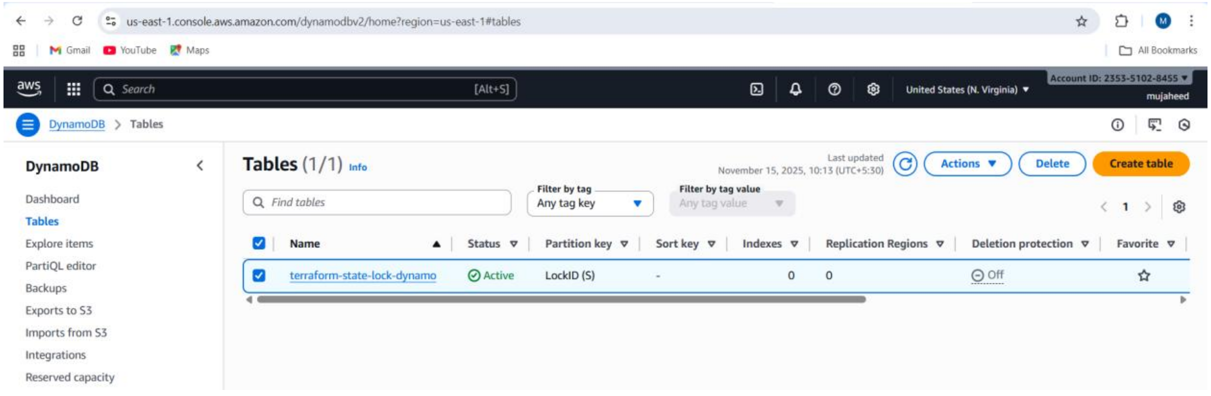


```
    with aws_s3_bucket.s3_bucket,
    on main.tf line 24, in resource "aws_s3_bucket" "s3_bucket":
    24:      acl = "private"

  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... [00m10s elapsed]
aws_dynamodb_table.dynamodb-terraform-state-lock: Creation complete after 18s [id=terrafo

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\Users\Ashish\Desktop\Terraform basics>
```

A dynamodb has been created.



```
    with aws_s3_bucket.s3_bucket,
    on main.tf line 24, in resource "aws_s3_bucket" "s3_bucket":
    24:     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... [00m10s elapsed]
aws_dynamodb_table.dynamodb-terraform-state-lock: Creation complete after 18s [id=terrafo

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\Users\Ashish\Desktop\Terraform basics>
```



Tables (1/1)

Last updated November 15, 2025, 10:13 (UTC+5:30)

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

Giving s3 as backend for terraform statefile.tf

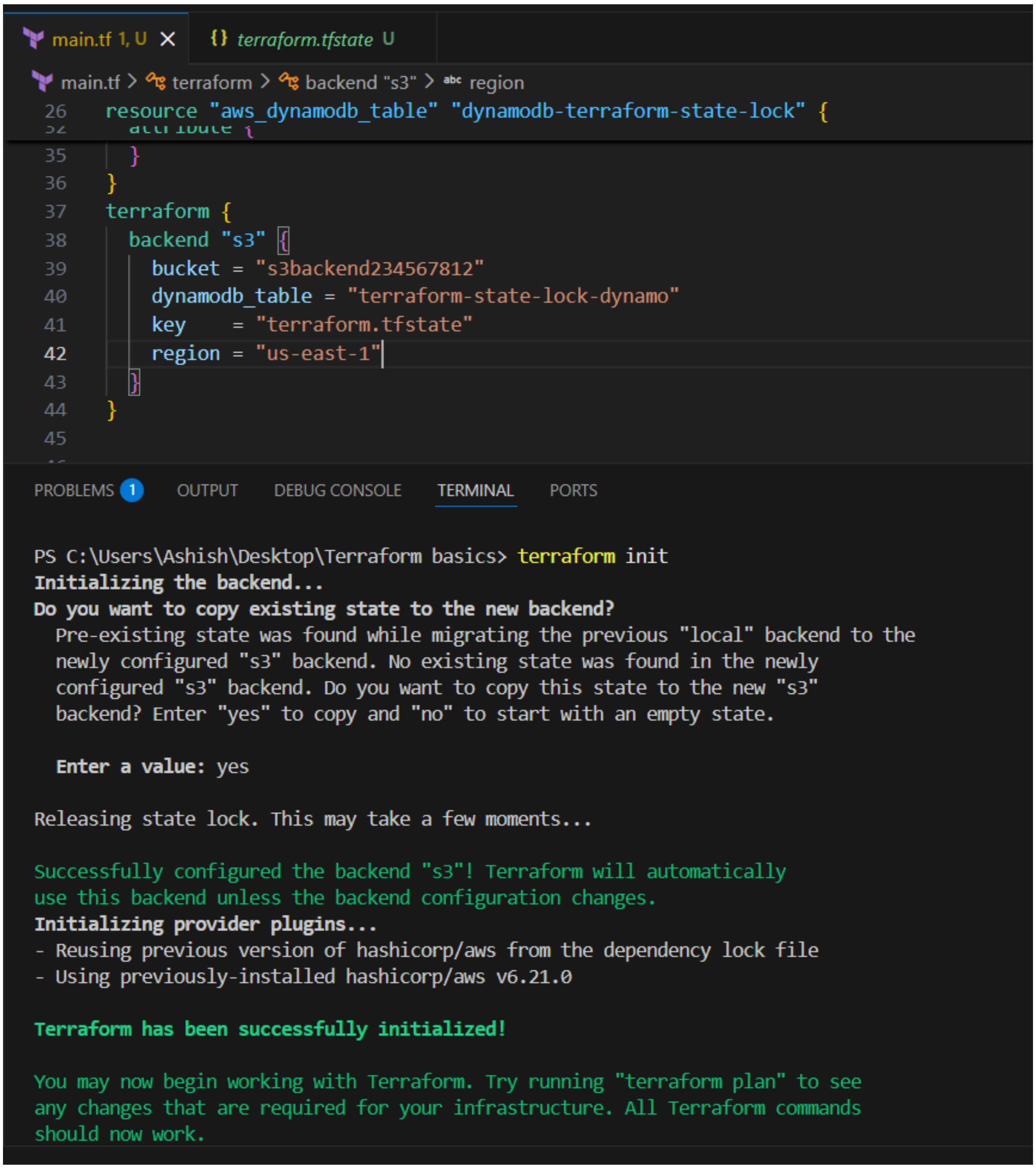
provider "aws" {

  region = "us-east-1"

}

```
resource "aws_instance" "web" {
  ami           = "ami-0cae6d6fe6048ca2c"
  instance_type = "t3.micro"
  subnet_id = "subnet-0a192382de0e2bf6a"

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

  tags = {
    Name = "httpd-server"
  }
}
resource "aws_s3_bucket" "s3_bucket" {

    bucket = "s3backend234567812"
    acl = "private"
}
```

```
}

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"
  }
}
terraform {
  backend "s3" {
    bucket = "s3backend234567812"
    dynamodb_table = "terraform-state-lock-dynamo"
    key    = "terraform.tfstate"
    region = "us-east-1"
  }
}
```



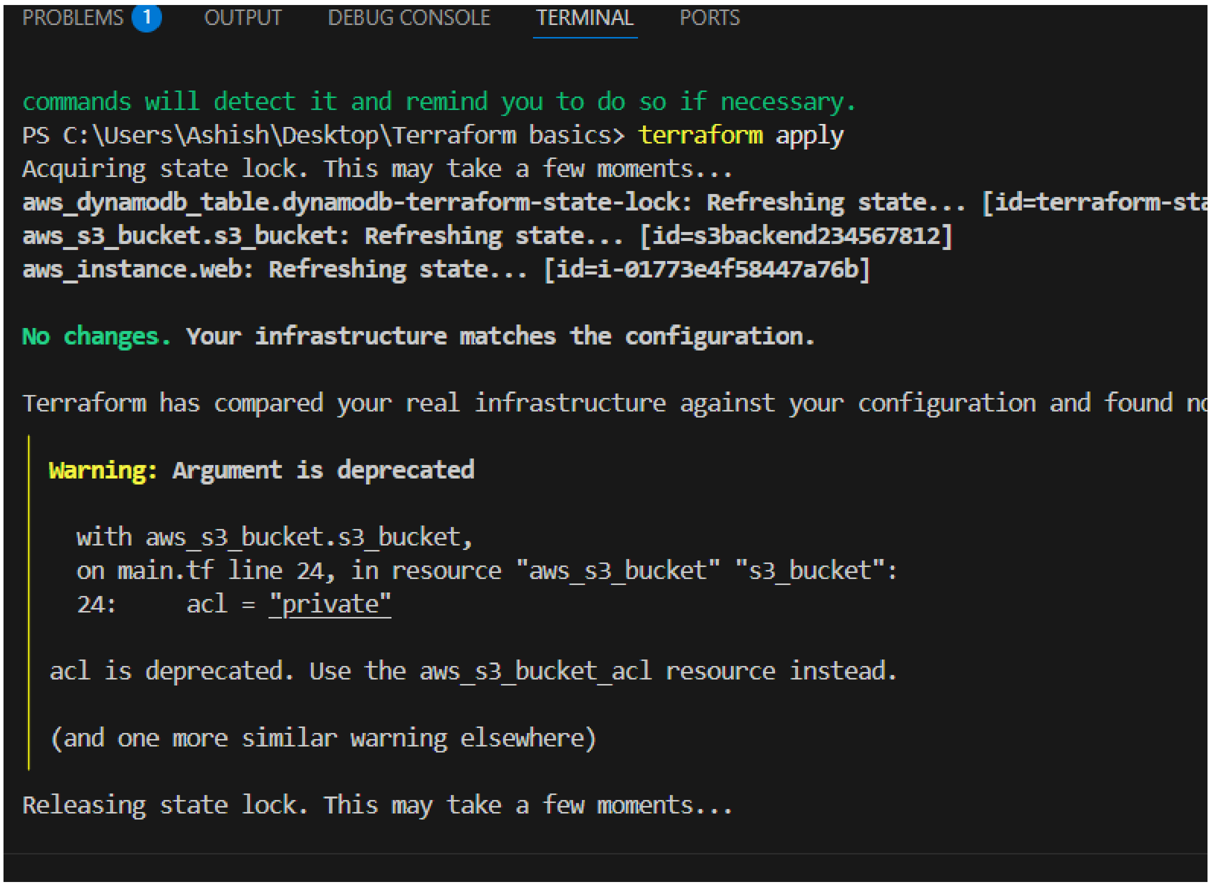
main.tf 1, U   terraform.tfstate U

main.tf > terraform > backend "s3" > region

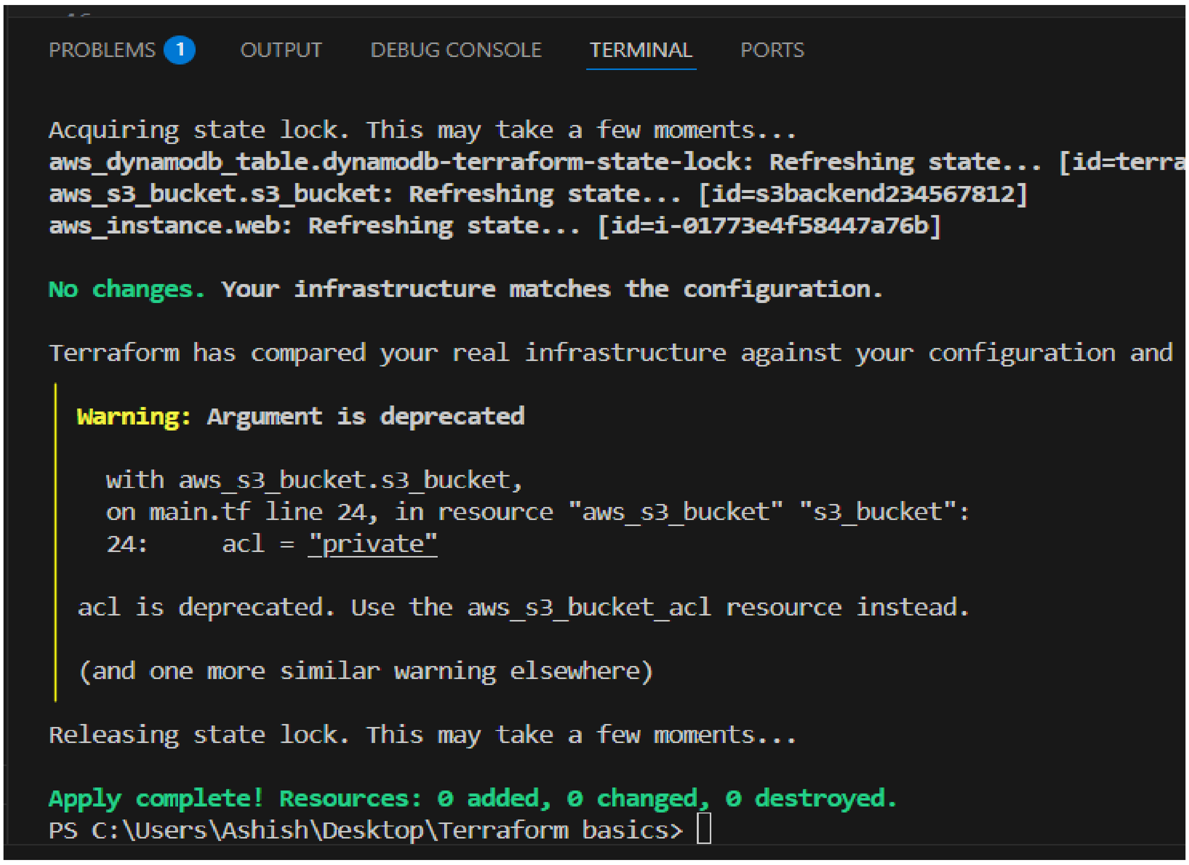
```
resource "aws_dynamodb_table" "dynamodb-terraform-state-lock" {
    attribute {
    }
}
terraform {
  backend "s3" {
    bucket = "s3backend234567812"
    dynamodb_table = "terraform-state-lock-dynamo"
    key    = "terraform.tfstate"
    region = "us-east-1"
  }
}
```

PROBLEMS 1   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```
PS C:\Users\Ashish\Desktop\Terraform basics> terraform init
Initializing the backend...
Do you want to copy existing state to the new backend?
  Pre-existing state was found while migrating the previous "local" backend to the
  newly configured "s3" backend. No existing state was found in the newly
  configured "s3" backend. Do you want to copy this state to the new "s3"
  backend? Enter "yes" to copy and "no" to start with an empty state.

  Enter a value: yes

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

Successfully configured the backend "s3"! Terraform will automatically
use this backend unless the backend configuration changes.
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v6.21.0

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.
```

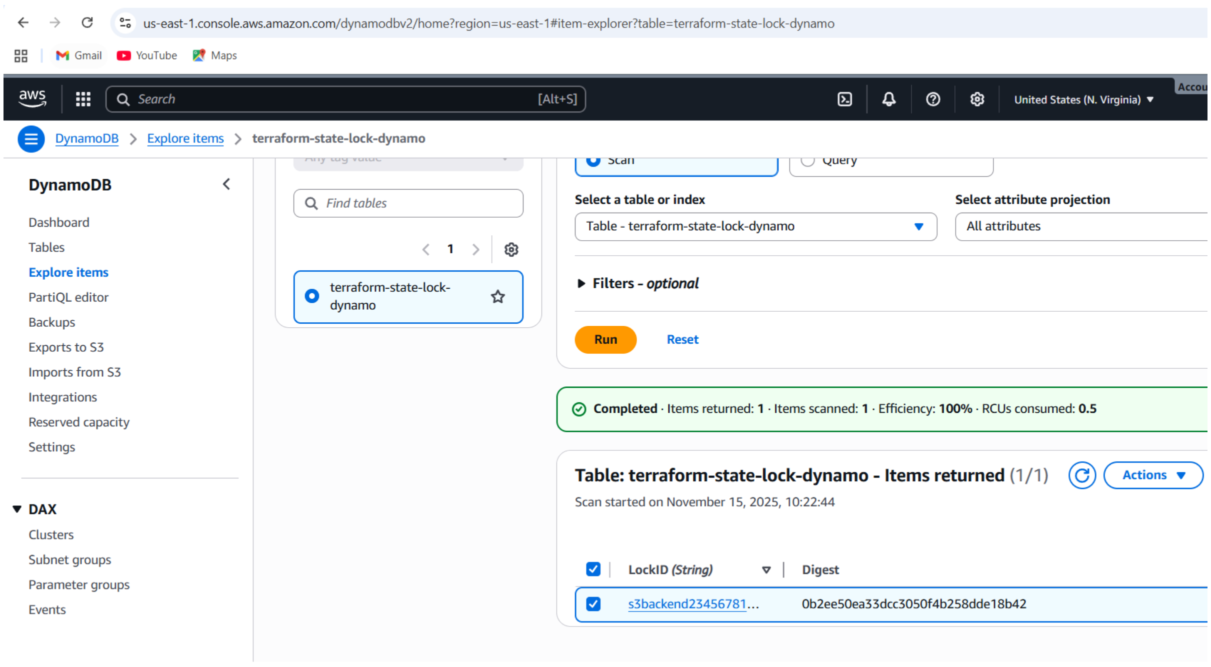


```
commands will detect it and remind you to do so if necessary.
PS C:\Users\Ashish\Desktop\Terraform basics> terraform apply
Acquiring state lock. This may take a few moments...
aws_dynamodb_table.dynamodb-terraform-state-lock: Refreshing state... [id=terraform-sta
aws_s3_bucket.s3_bucket: Refreshing state... [id=s3backend234567812]
aws_instance.web: Refreshing state... [id=i-01773e4f58447a76b]

No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and found no

│ Warning: Argument is deprecated
│
│   with aws_s3_bucket.s3_bucket,
│   on main.tf line 24, in resource "aws_s3_bucket" "s3_bucket":
│   24:     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...
```



```
Acquiring state lock. This may take a few moments...
aws_dynamodb_table.dynamodb-terraform-state-lock: Refreshing state... [id=terra
aws_s3_bucket.s3_bucket: Refreshing state... [id=s3backend234567812]
aws_instance.web: Refreshing state... [id=i-01773e4f58447a76b]

No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and

│ Warning: Argument is deprecated
│
│   with aws_s3_bucket.s3_bucket,
│   on main.tf line 24, in resource "aws_s3_bucket" "s3_bucket":
│   24:     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:\Users\Ashish\Desktop\Terraform basics> 
```

A table has been created in the dynamodb.



Select a table or index: Table - terraform-state-lock-dynamo

Select attribute projection: All attributes

Filters - *optional*

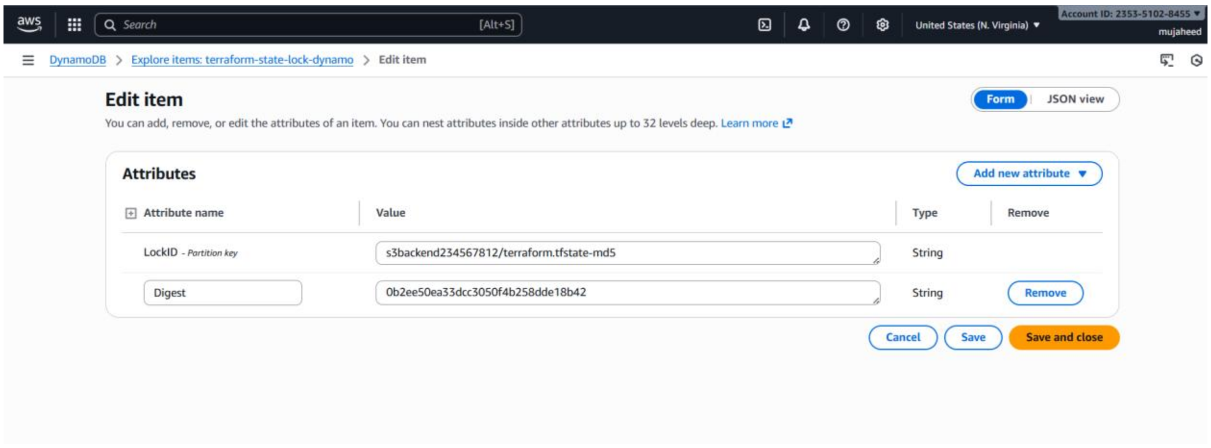
Completed · Items returned: 1 · Items scanned: 1 · Efficiency: 100% · RCUs consumed: 0.5

Table: terraform-state-lock-dynamo - Items returned (1/1)

Scan started on November 15, 2025, 10:22:44

LockID (String)	Digest
s3backend23456781…	0b2ee50ea33dcc3050f4b258dde18b42

Our state file has been stored in dynamodb.

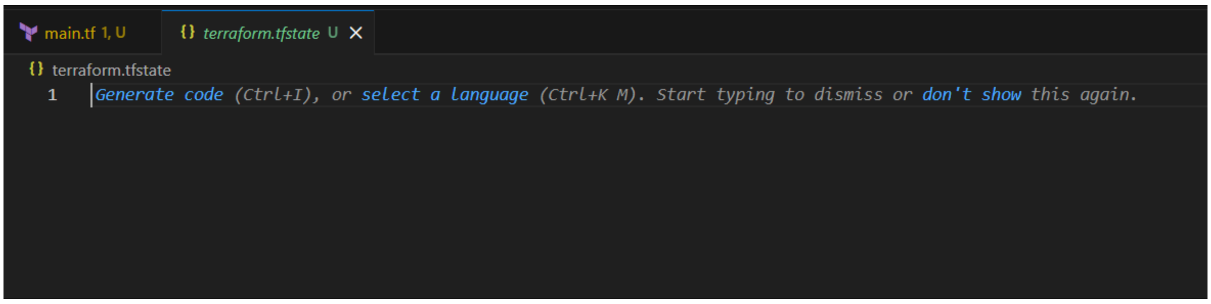

DynamoDB > Explore items: terraform-state-lock-dynamo > Edit item
Form
JSON view
Add new attribute
Cancel
Save
Save and close

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	Remove
LockID - *Partition key*	s3backend234567812/terraform.tfstate-md5	String	
Digest	0b2ee50ea33dcc3050f4b258dde18b42	String	Remove



```
Generate code (Ctrl+I), or select a language (Ctrl+K M). Start typing to dismiss or don't show this again.
```