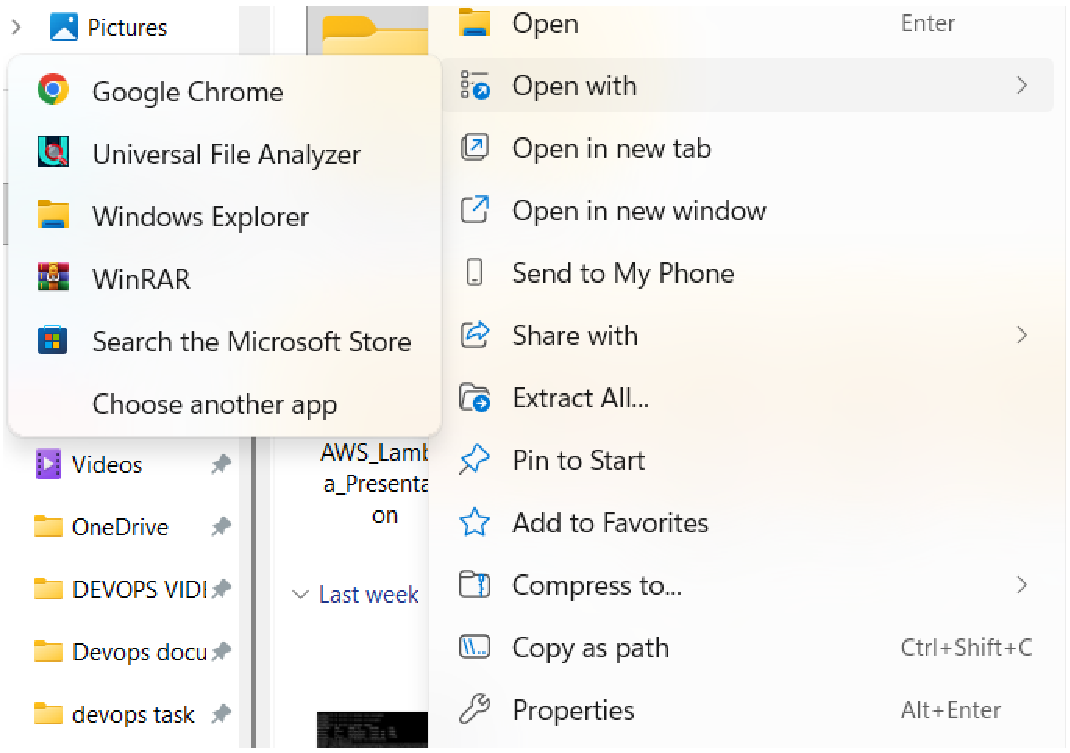
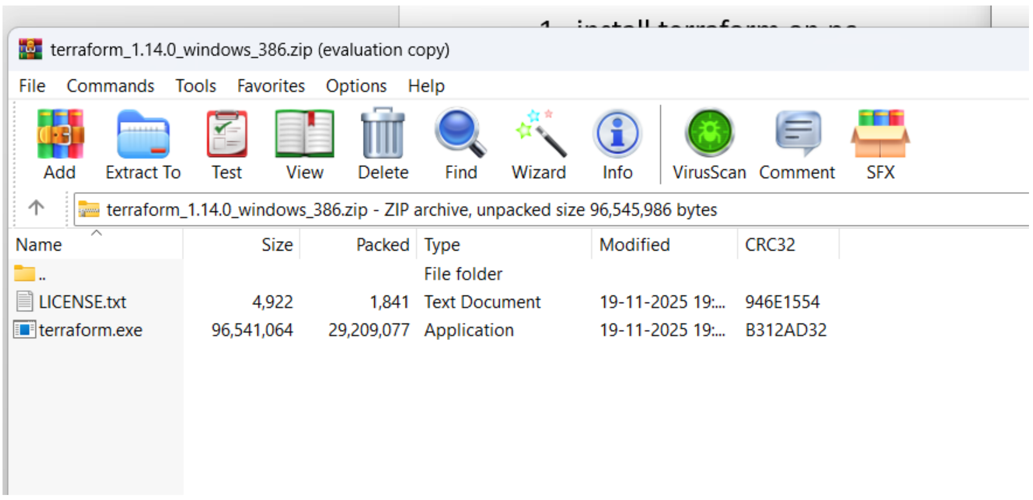
TERRAFORM – 01

1 . install terraform on pc

- Download terraform file from terraform official site
- Now, downloads >> extract   file using
- Now open it with winrar



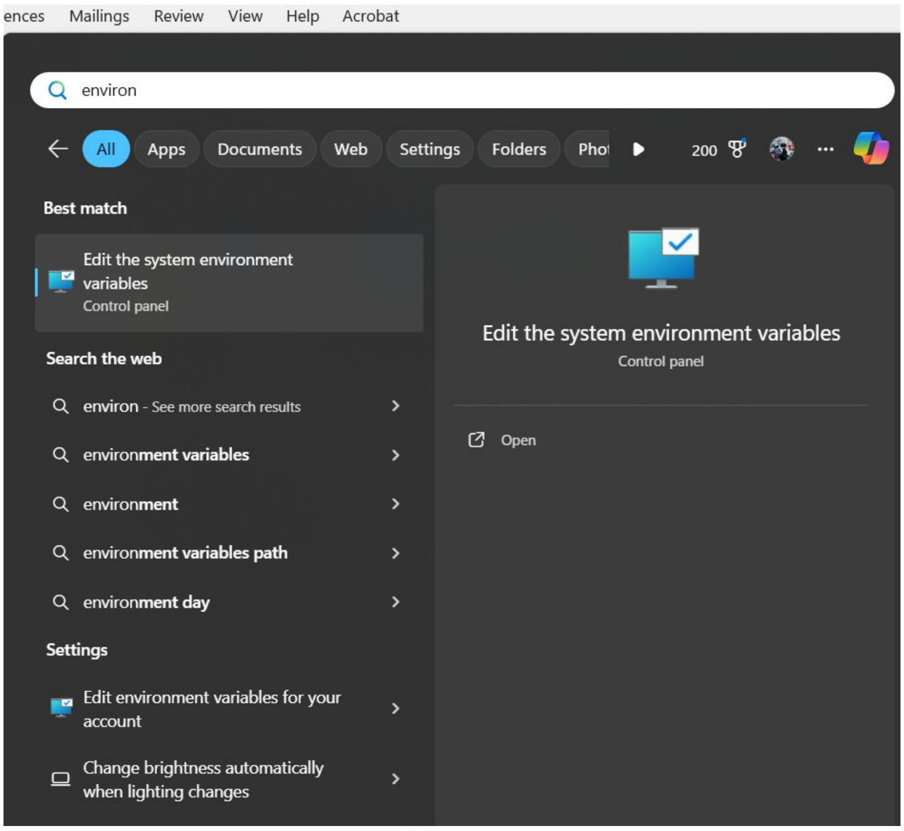
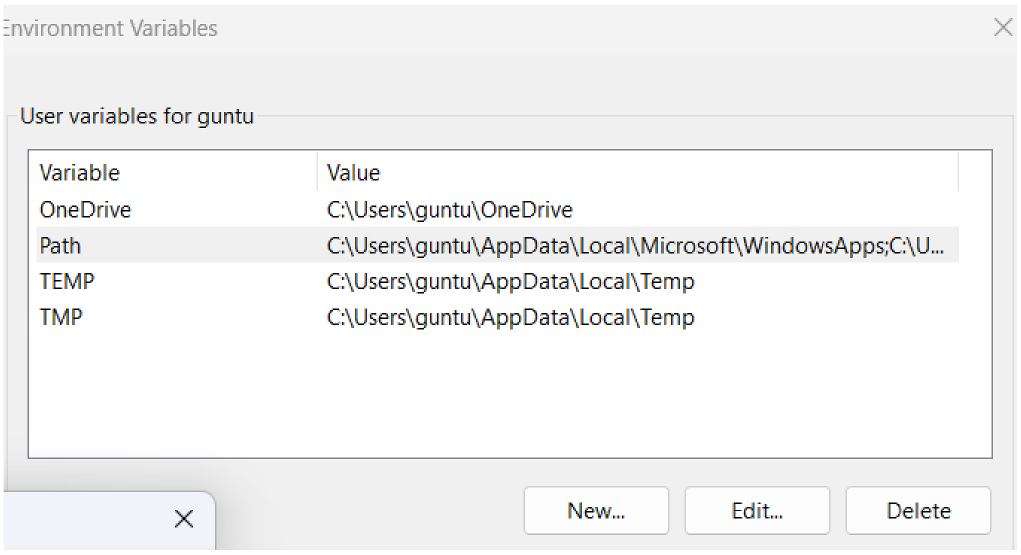
-
- Click exe file


terraform_1.14.0_windows_386.zip (evaluation copy)
File Commands Tools Favorites Options Help
Add
Extract To
Test
View
Delete
Find
Wizard
Info
VirusScan
Comment
SFX

terraform_1.14.0_windows_386.zip - ZIP archive, unpacked size 96,545,986 bytes

Name	Size	Packed	Type	Modified	CRC32
..			File folder		
LICENSE.txt	4,922	1,841	Text Document	19-11-2025 19:...	946E1554
terraform.exe	96,541,064	29,209,077	Application	19-11-2025 19:...	B312AD32

-
- Now copy the terraform file from downloads and paste it in c drive .

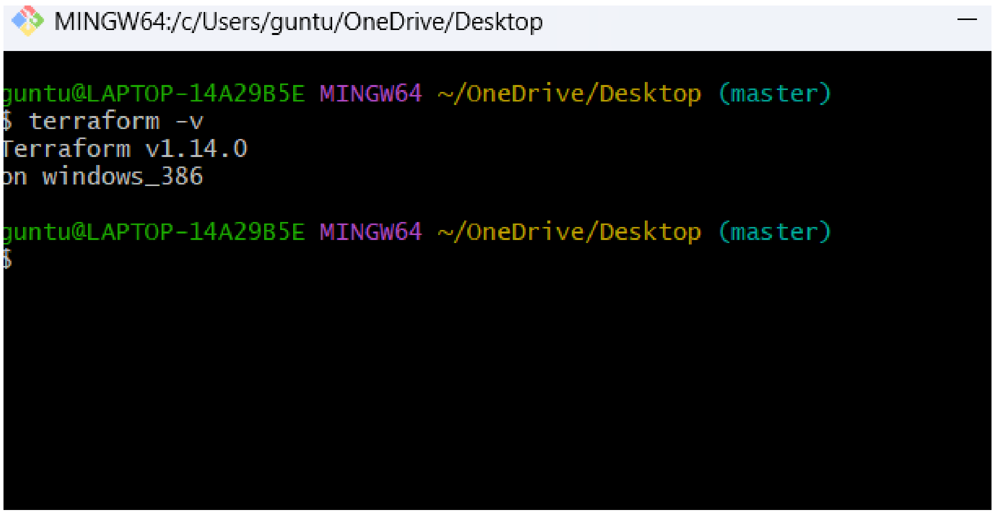
- Now add path
- Go to start button >
- Search for environmental variables
- 
ences Mailings Review View Help Acrobat
environ
All
Apps
Documents
Web
Settings
Folders
Phot
200
Best match
Edit the system environment variables
Control panel
Search the web
environ - See more search results
environment variables
environment
environment variables path
environment day
Settings
Edit environment variables for your account
Change brightness automatically when lighting changes
Edit the system environment variables
Control panel
Open
- Select path and hit edit
- 

Environment Variables

User variables for guntu

Variable	Value
OneDrive	C:\Users\guntu\OneDrive
Path	C:\Users\guntu\AppData\Local\Microsoft\WindowsApps;C:\U...
TEMP	C:\Users\guntu\AppData\Local\Temp
TMP	C:\Users\guntu\AppData\Local\Temp

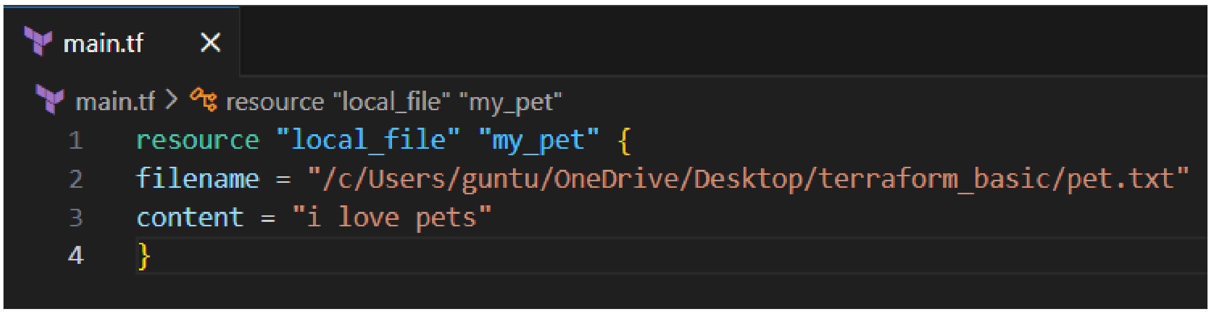
- Add the new path
- Add location of terraform

- C:\terraform_1.14.0_windows_386 (as we pasted terraform file in c drive)
- Now run terraform -v command in git from desktop location can see the version of terraform available in our local computer.
- 

```
MINGW64:/c/Users/guntu/OneDrive/Desktop

guntu@LAPTOP-14A29B5E MINGW64 ~/OneDrive/Desktop (master)
$ terraform -v
Terraform v1.14.0
on windows_386

guntu@LAPTOP-14A29B5E MINGW64 ~/OneDrive/Desktop (master)
$
```

2 . Execute all the templates shown in video.

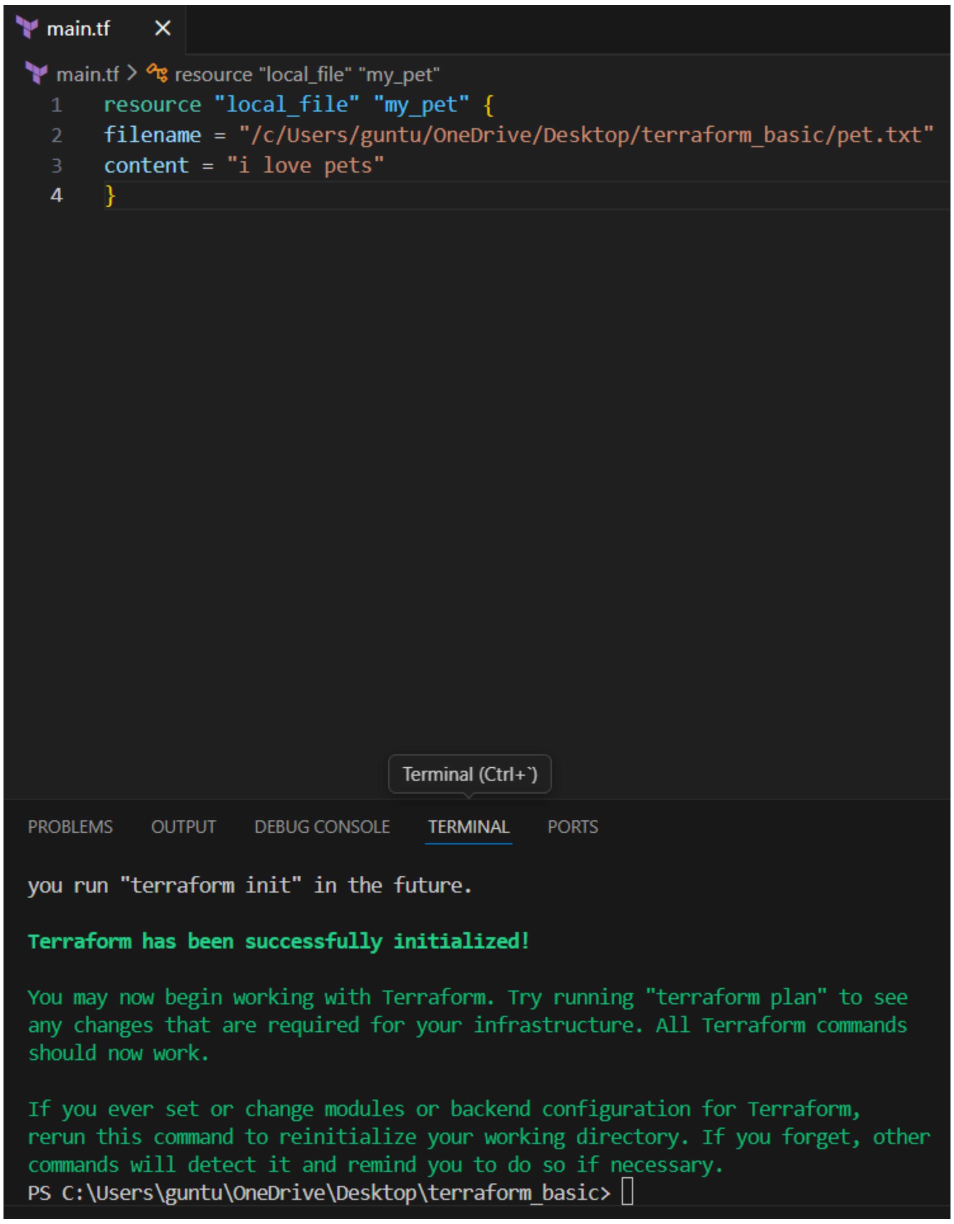
- Create a folder in desktop > local machine
- Now, open vs code
- Add a folder in vs code
- And add extension “hashi crop ”
- Now add afile to that folder
- And write a template



```
main.tf
main.tf > resource "local_file" "my_pet"
resource "local_file" "my_pet" {
filename = "/c/Users/guntu/OneDrive/Desktop/terraform_basic/pet.txt"
content = "i love pets"
}
```

- Now, on the top click terminal

- And give command -- terraform init



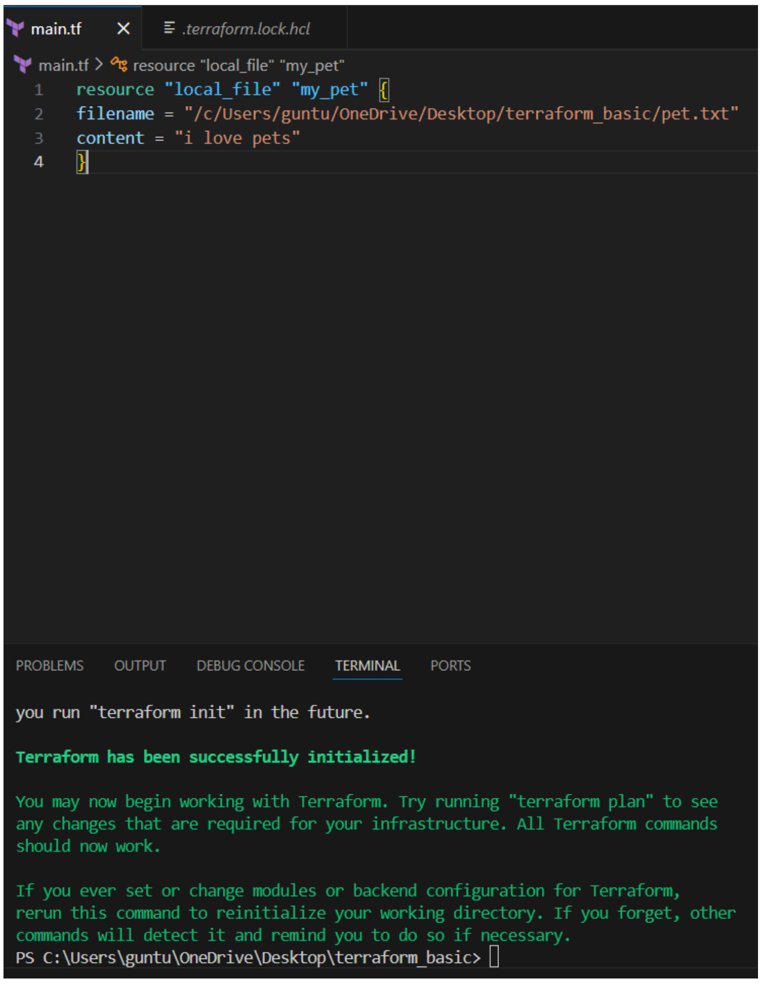
```
resource "local_file" "my_pet" {
filename = "/c/Users/guntu/OneDrive/Desktop/terraform_basic/pet.txt"
content = "i love pets"
}
```

```
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
PS C:\Users\guntu\OneDrive\Desktop\terraform_basic>
```

- Can file will be added to folder .

3 . Note down below points, Terraform Init, Terraform Plan, Terraform Apply, Terraform Provider.

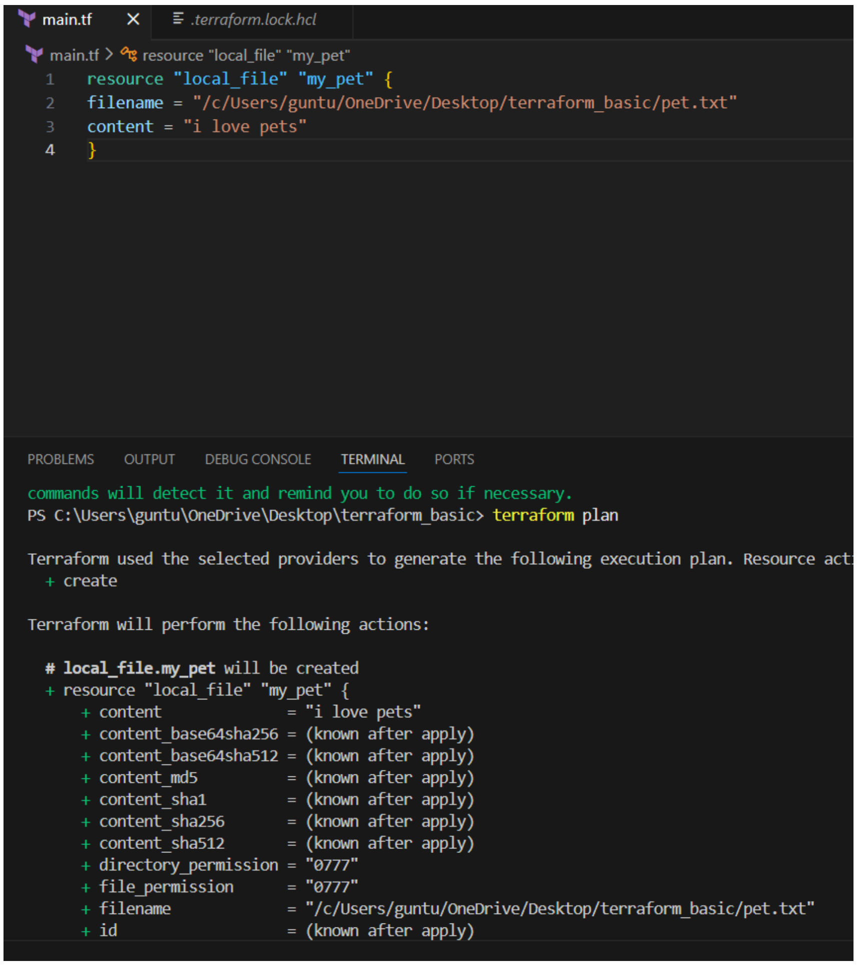
  - Terra form init



```hcl
resource "local_file" "my_pet" {
filename = "/c/Users/guntu/OneDrive/Desktop/terraform_basic/pet.txt"
content = "i love pets"
}
```

```
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
PS C:\Users\guntu\OneDrive\Desktop\terraform_basic>
```

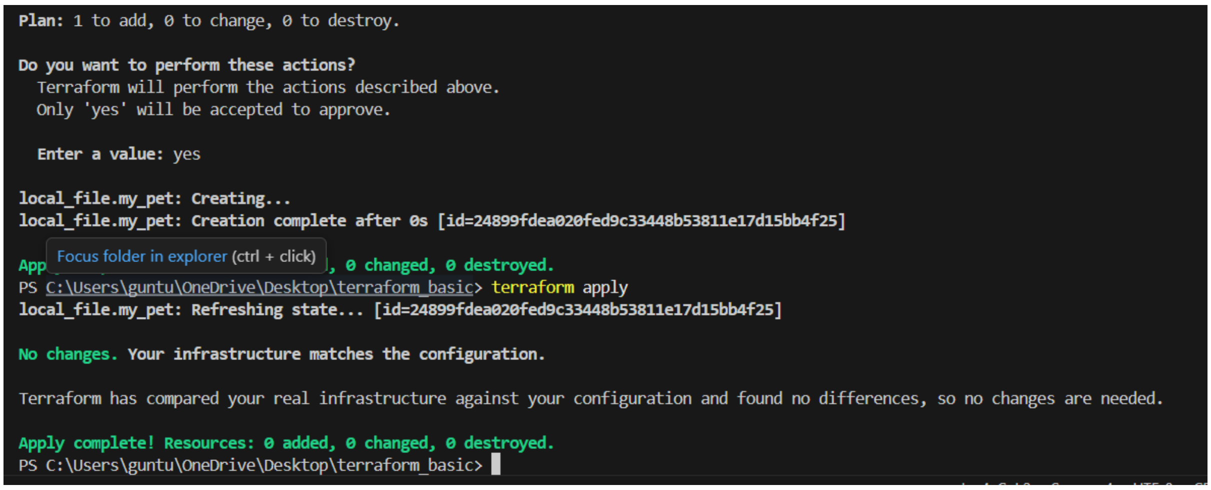
- Terraform plan



```
resource "local_file" "my_pet" {
filename = "/c/Users/guntu/OneDrive/Desktop/terraform_basic/pet.txt"
content = "i love pets"
}
```

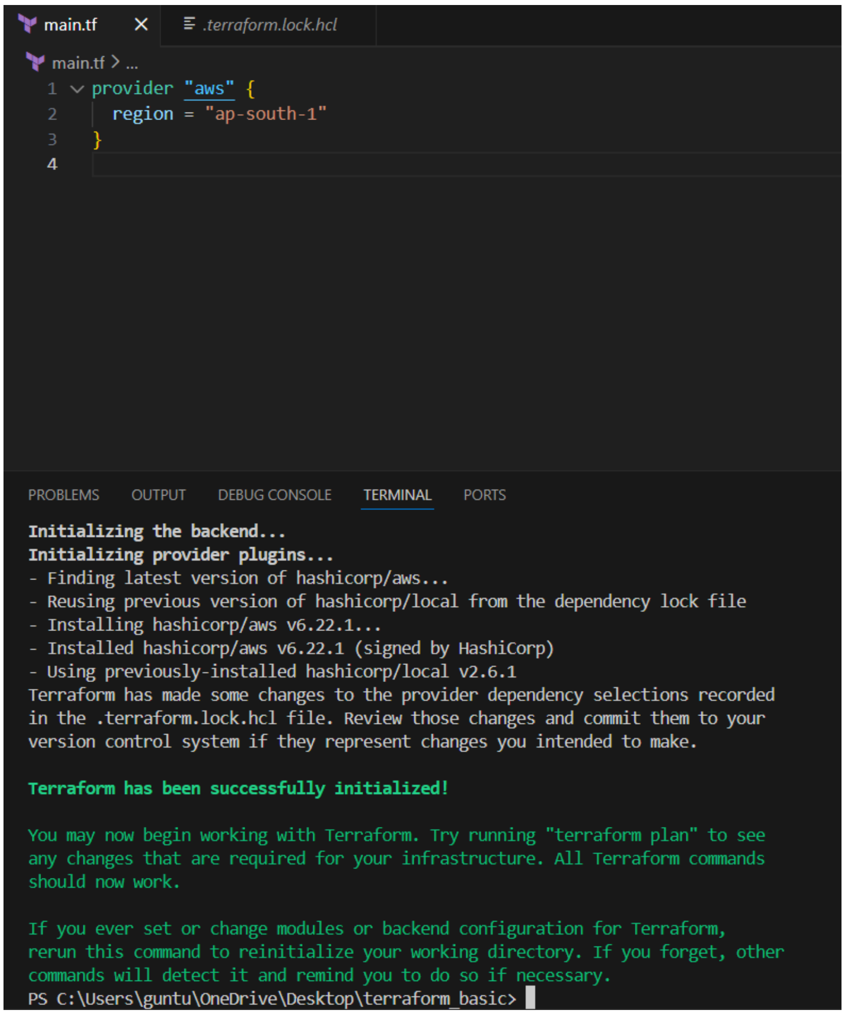
```
commands will detect it and remind you to do so if necessary.
PS C:\Users\guntu\OneDrive\Desktop\terraform_basic> terraform plan

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

Terraform will perform the following actions:

  # local_file.my_pet will be created
  + resource "local_file" "my_pet" {
      + content              = "i love pets"
      + 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             = "/c/Users/guntu/OneDrive/Desktop/terraform_basic/pet.txt"
      + id                   = (known after apply)
```

- Terraform apply now ,



```
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.my_pet: Creating...
local_file.my_pet: Creation complete after 0s [id=24899fdea020fed9c33448b53811e17d15bb4f25]

App                                          , 0 changed, 0 destroyed.
PS C:\Users\guntu\OneDrive\Desktop\terraform_basic> terraform apply
local_file.my_pet: Refreshing state... [id=24899fdea020fed9c33448b53811e17d15bb4f25]

No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and found no differences, so no changes are needed.

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
PS C:\Users\guntu\OneDrive\Desktop\terraform_basic>
```

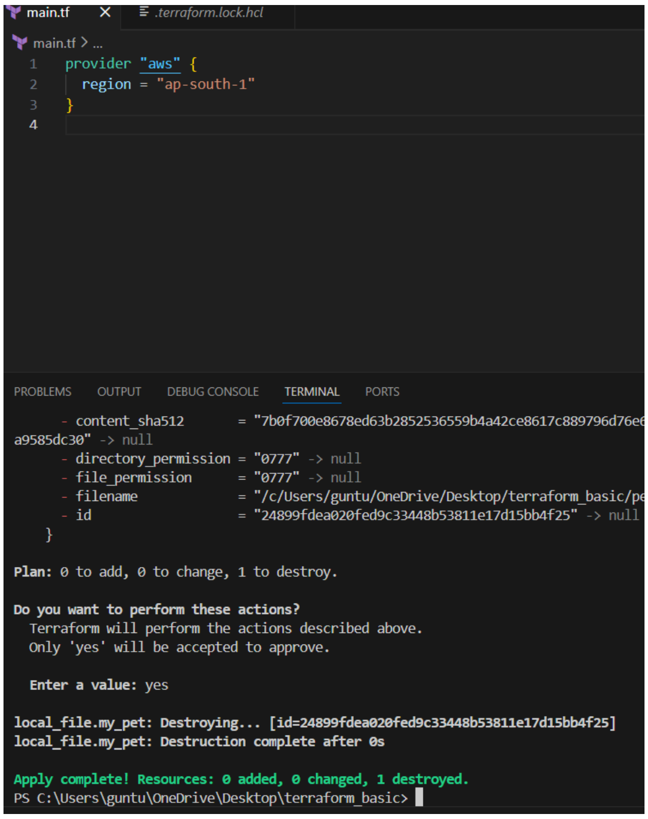
- Terraform provider.
- Add new content to main.tf
- And rthen do terraform init



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

```
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 v6.22.1...
- Installed hashicorp/aws v6.22.1 (signed by HashiCorp)
- Using previously-installed hashicorp/local v2.6.1
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\guntu\OneDrive\Desktop\terraform_basic>
```

- Terraform plan
- And then terraform apply and give yes



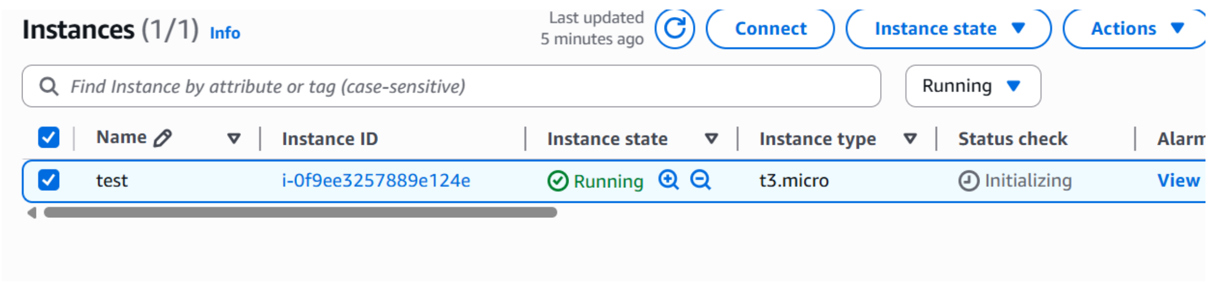
```
provider "aws" {
  region = "ap-south-1"
}
```

```
      - content_sha512       = "7b0f700e8678ed63b2852536559b4a42ce8617c889796d76e6
a9585dc30" -> null
      - directory_permission = "0777" -> null
      - file_permission      = "0777" -> null
      - filename             = "/c/Users/guntu/OneDrive/Desktop/terraform_basic/pe
      - id                   = "24899fdea020fed9c33448b53811e17d15bb4f25" -> null
    }

Plan: 0 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.my_pet: Destroying... [id=24899fdea020fed9c33448b53811e17d15bb4f25]
local_file.my_pet: Destruction complete after 0s

Apply complete! Resources: 0 added, 0 changed, 1 destroyed.
PS C:\Users\guntu\OneDrive\Desktop\terraform_basic>
```

- Terraform provider .
- Add a script
- Configure = aws configure
- Give access key and secret key

```
• resource "aws_instance" "webserver" {
•   ami           = "ami-0fa3fe0fa7920f68e"
•   subnet_id     = "subnet-022a5b6a074769fa4"
•   instance_type = "t3.micro"
•   tags = {
•     Name = "test"
•   }
• }
```

- Copy ami id from aws ec2 and give it here
- And open terminal hit terraform apply
- Ec2 instance will be launching


Instances (1/1) Info
Last updated 5 minutes ago
Connect
Instance state ▼
Actions ▼
Find Instance by attribute or tag (case-sensitive)
Running ▼

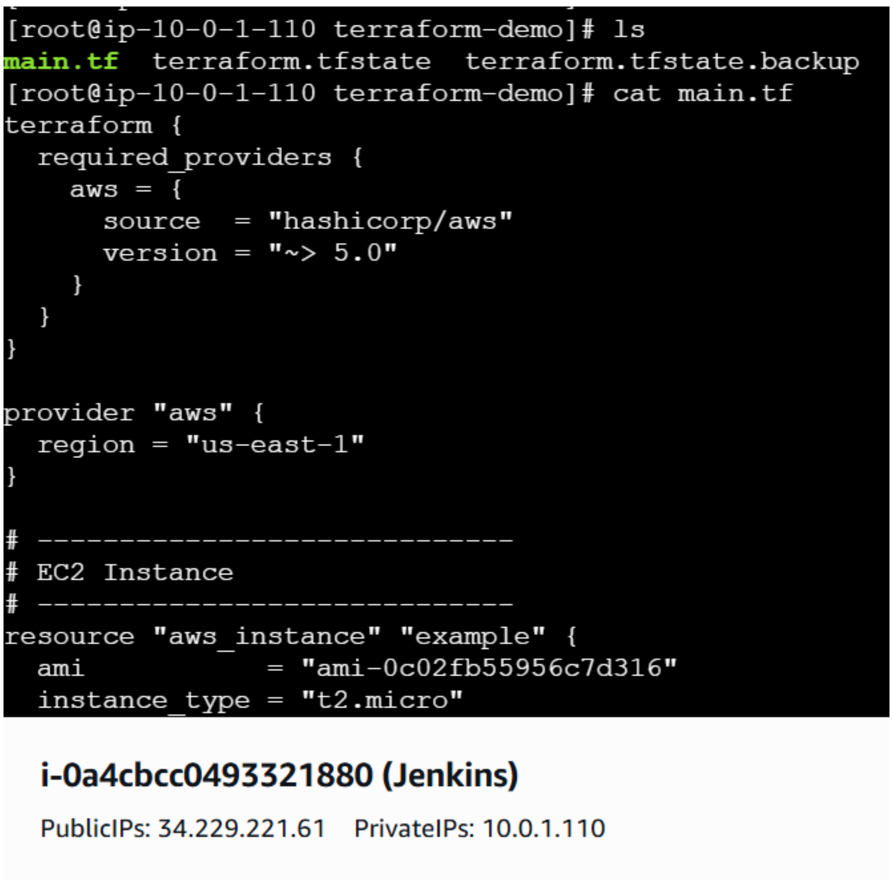
☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm
☑	test	i-0f9ee3257889e124e	Running	t3.micro	Initializing	View

-

4 . Integrate a sample Terraform template in jenkins.

- Create a file called main.tf
- /var/lib/jenkins/terraform-demo/main.tf
- Add script to main.tf (it contains vpc, subnet id, aws-cred)
- terraform {
-   required_providers {
-     aws = {
-       source = "hashicorp/aws"
-       version = "~> 5.0"
-     }
-   }
- }
-
- provider "aws" {
-   region = "us-east-1"
- }
-
- # ----------------------------
- # EC2 Instance
- # ----------------------------
- resource "aws_instance" "example" {
-   ami   = "ami-0c02fb55956c7d316"
-   instance_type = "t2.micro"
-
-   # REQUIRED → your subnet
-   subnet_id   = "subnet-022a5b6a074769fa4"
-
-   # Optional tags
-   tags = {

- Name = "jenkins-terraform-demo"
- }
- }

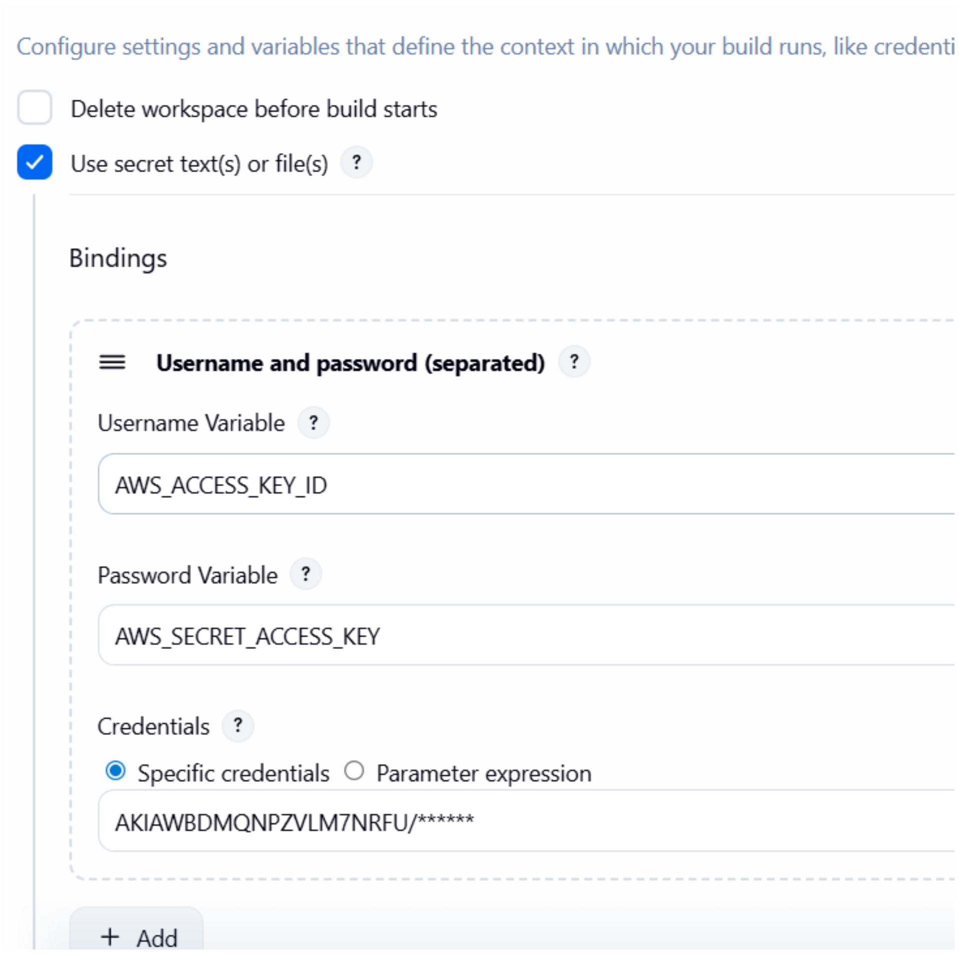


```
[root@ip-10-0-1-110 terraform-demo]# ls
main.tf  terraform.tfstate  terraform.tfstate.backup
[root@ip-10-0-1-110 terraform-demo]# cat main.tf
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 5.0"
    }
  }
}

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

# ----------------------------
# EC2 Instance
# ----------------------------
resource "aws_instance" "example" {
  ami           = "ami-0c02fb55956c7d316"
  instance_type = "t2.micro"
```

i-0a4cbcc0493321880 (Jenkins)

PublicIPs: 34.229.221.61   PrivateIPs: 10.0.1.110

- Configure aws with Jenkins and Jenkins server
- Add access key and secret to Jenkins
- Now, create job- terraform demo
- Free style job
- Go to configure
- Build environment


Configure settings and variables that define the context in which your build runs, like credenti

[ ] Delete workspace before build starts

[✓] Use secret text(s) or file(s) ?

Bindings

Username and password (separated) ?

Username Variable ?

AWS_ACCESS_KEY_ID

Password Variable ?

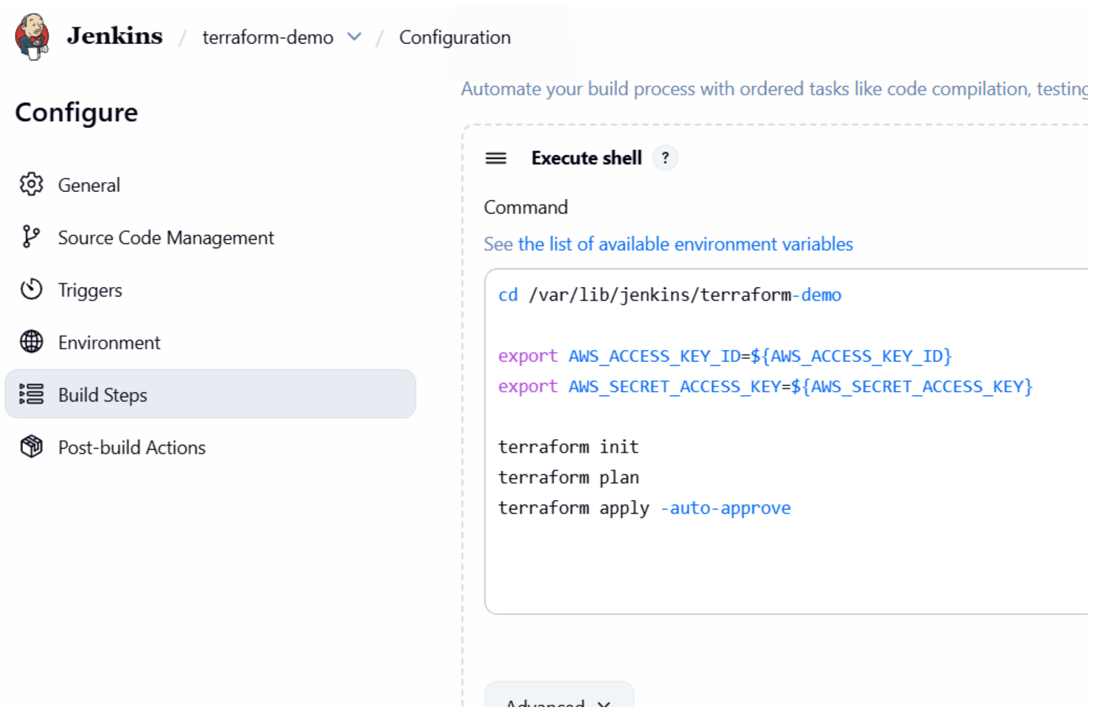
AWS_SECRET_ACCESS_KEY

Credentials ?

[✓ Specific credentials] Parameter expression

AKIAWBDMQNPZVLM7NRFU/******

-
- Select use secret test or file
- Add created aws-creds tro this
- Now ,
- Go to
- Build steps
- Select execute shell

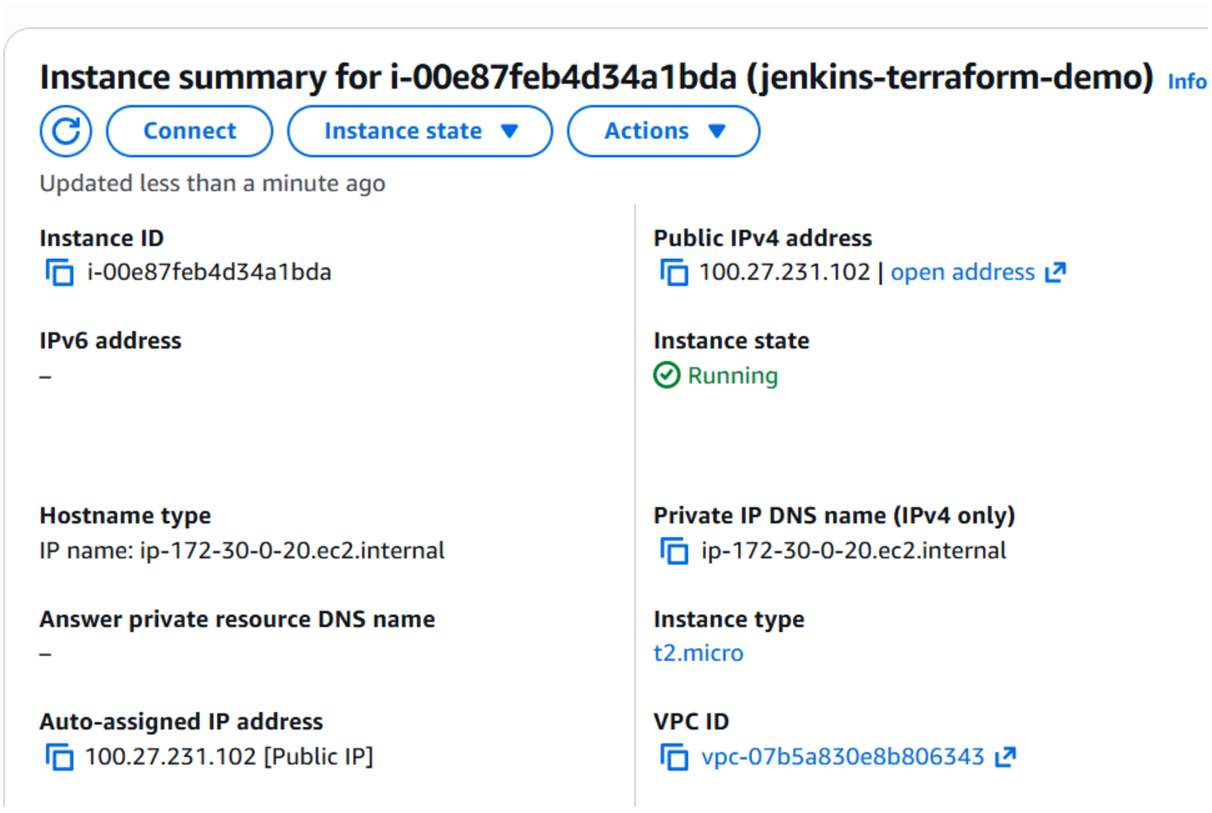


```
cd /var/lib/jenkins/terraform-demo

export AWS_ACCESS_KEY_ID=${AWS_ACCESS_KEY_ID}
export AWS_SECRET_ACCESS_KEY=${AWS_SECRET_ACCESS_KEY}

terraform init
terraform plan
terraform apply -auto-approve
```

-
- cd /var/lib/jenkins/terraform-demo
-
- export AWS_ACCESS_KEY_ID=${AWS_ACCESS_KEY_ID}
- export AWS_SECRET_ACCESS_KEY=${AWS_SECRET_ACCESS_KEY}
-
- terraform init
- terraform plan
- terraform apply -auto-approve
- now run the job

```
    }

□[1mPlan:□[0m 1 to add, 0 to change, 0 to destroy.
□[0m□[0m□[1maws_instance.example: Creating...□[0m□[0m
□[0m□[1maws_instance.example: Still creating... [10s elapsed]□[0m□[0m
□[0m□[1maws_instance.example: Creation complete after 13s [id=i-00e87feb4d34a1bda]□[0m
□[0m□[1m□[32m
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
□[0mFinished: SUCCESS
```

-
- Instanc e launched successfully



Instance summary for i-00e87feb4d34a1bda (jenkins-terraform-demo) Info

Updated less than a minute ago

Instance ID	**Public IPv4 address**
i-00e87feb4d34a1bda	100.27.231.102 | open address
IPv6 address	**Instance state**
–	Running
Hostname type	**Private IP DNS name (IPv4 only)**
IP name: ip-172-30-0-20.ec2.internal	ip-172-30-0-20.ec2.internal
Answer private resource DNS name	**Instance type**
–	t2.micro
Auto-assigned IP address	**VPC ID**
100.27.231.102 [Public IP]	vpc-07b5a830e8b806343