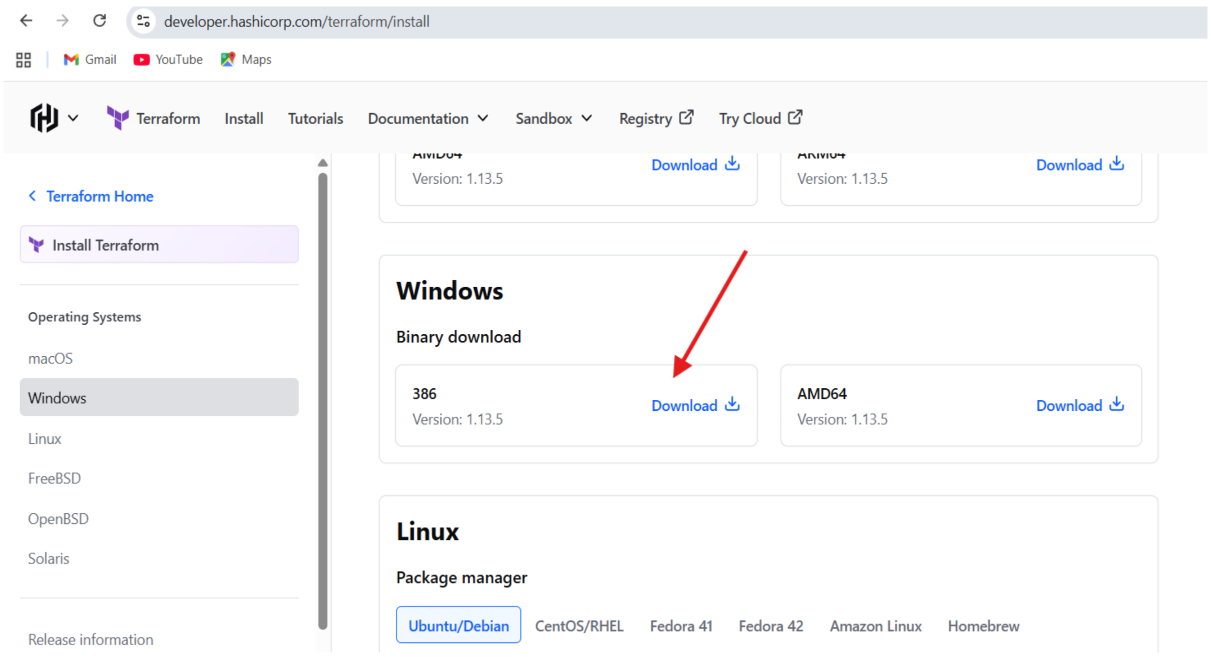
1. Install Terraform on your PC

Go to browser search as terraform download for windows and check for windows option and click on download.



AMD64
Version: 1.13.5
Download

ARM64
Version: 1.13.5
Download

Windows

Binary download

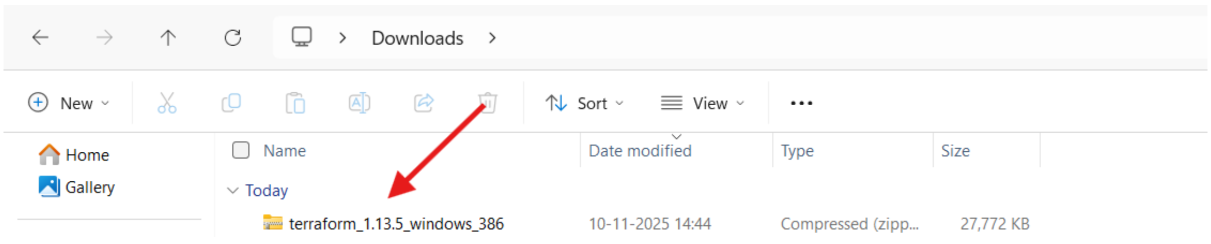
386
Version: 1.13.5
Download

AMD64
Version: 1.13.5
Download

Linux

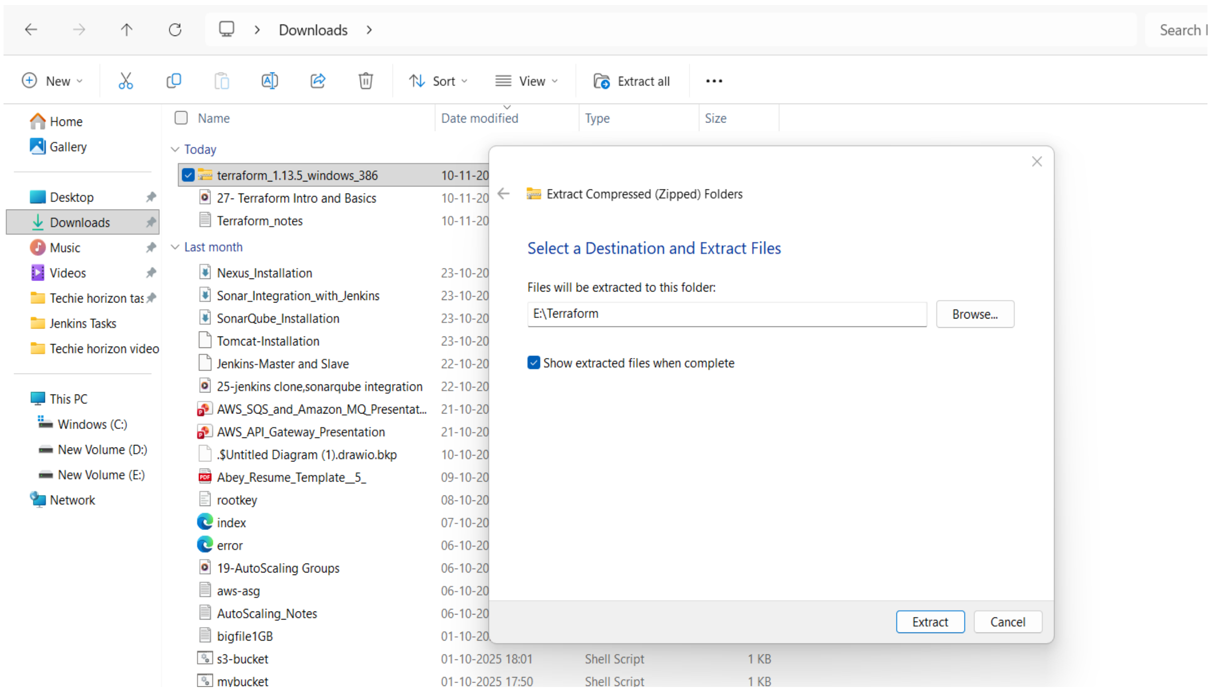
Package manager

Ubuntu/Debian CentOS/RHEL Fedora 41 Fedora 42 Amazon Linux Homebrew



Name	Date modified	Type	Size
Today			
terraform_1.13.5_windows_386	10-11-2025 14:44	Compressed (zipp...	27,772 KB

Give a right click on that select extract all and give your location.


Extract Compressed (Zipped) Folders

Select a Destination and Extract Files

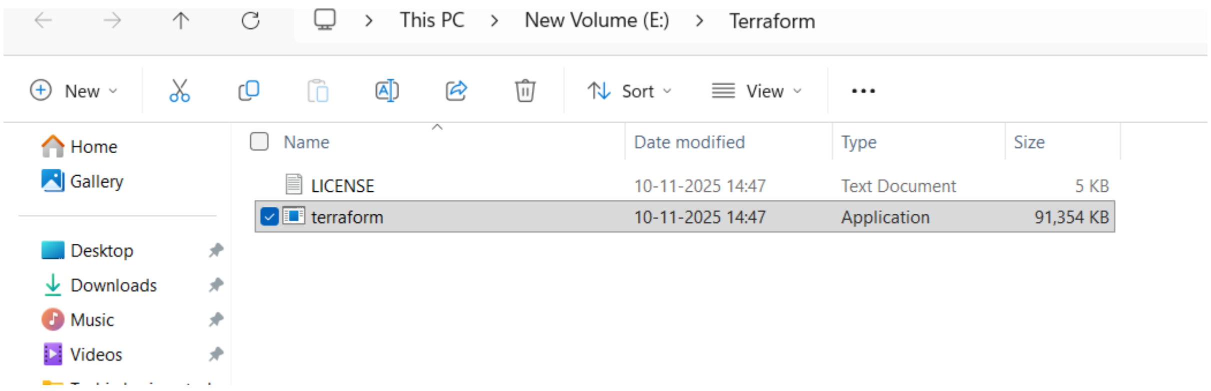
Files will be extracted to this folder:

E:\Terraform

☑ Show extracted files when complete

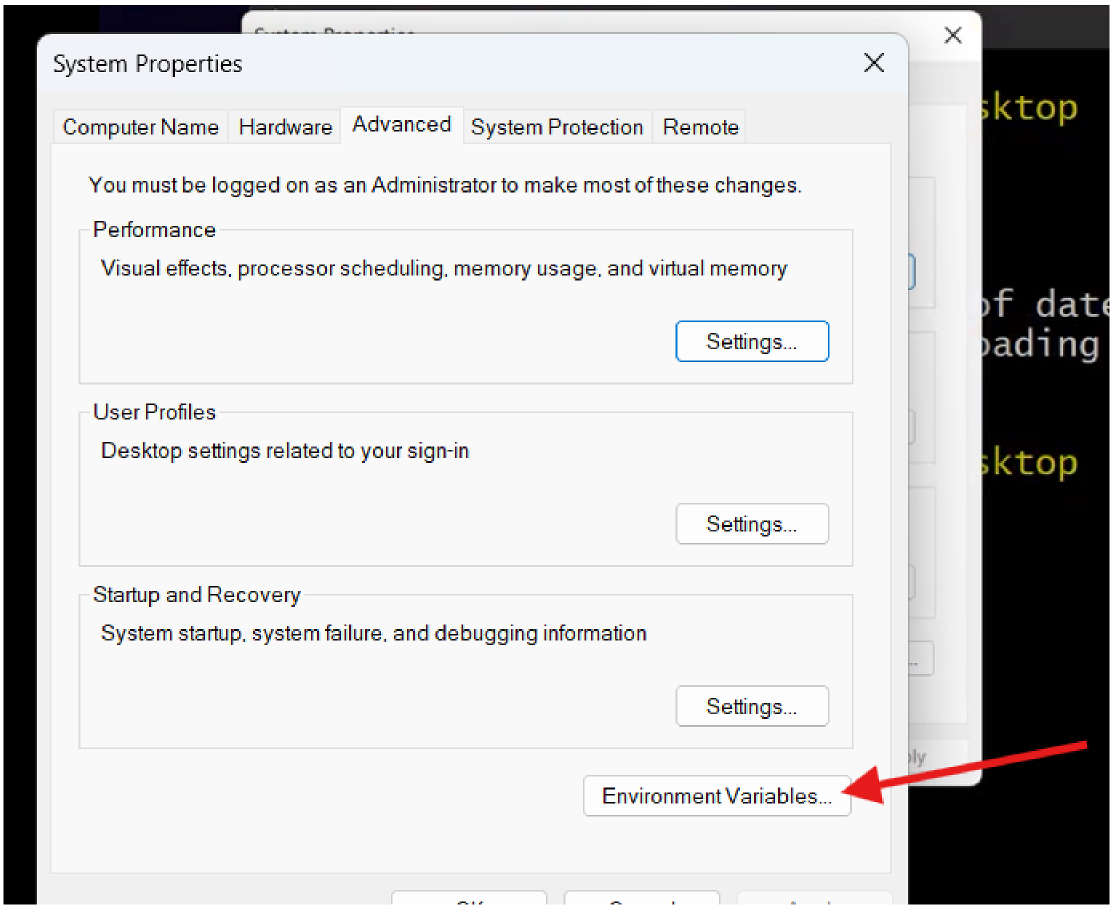
Name	Date modified	Type	Size
terraform_1.13.5_windows_386	10-11-20		
27- Terraform Intro and Basics	10-11-20		
Terraform_notes	10-11-20		
Nexus_Installation	23-10-20		
Sonar_Integration_with_Jenkins	23-10-20		
SonarQube_Installation	23-10-20		
Tomcat-Installation	23-10-20		
Jenkins-Master and Slave	22-10-20		
25-jenkins clone,sonarqube integration	22-10-20		
AWS_SQS_and_Amazon_MQ_Presentat...	21-10-20		
AWS_API_Gateway_Presentation	21-10-20		
.$Untitled Diagram (1).drawio.bkp	10-10-20		
Abey_Resume_Template__5_	09-10-20		
rootkey	08-10-20		
index	07-10-20		
error	06-10-20		
19-AutoScaling Groups	06-10-20		
aws-asg	06-10-20		
AutoScaling_Notes	06-10-20		
bigfile1GB	01-10-20		
s3-bucket	01-10-2025 18:01	Shell Script	1 KB
mybucket	01-10-2025 17:50	Shell Script	1 KB

You will see an exe file.


This PC > New Volume (E:) > Terraform

Name	Date modified	Type	Size
LICENSE	10-11-2025 14:47	Text Document	5 KB
terraform	10-11-2025 14:47	Application	91,354 KB

- Search in your pc as system variables select environmental variables.


System Properties

You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage, and virtual memory

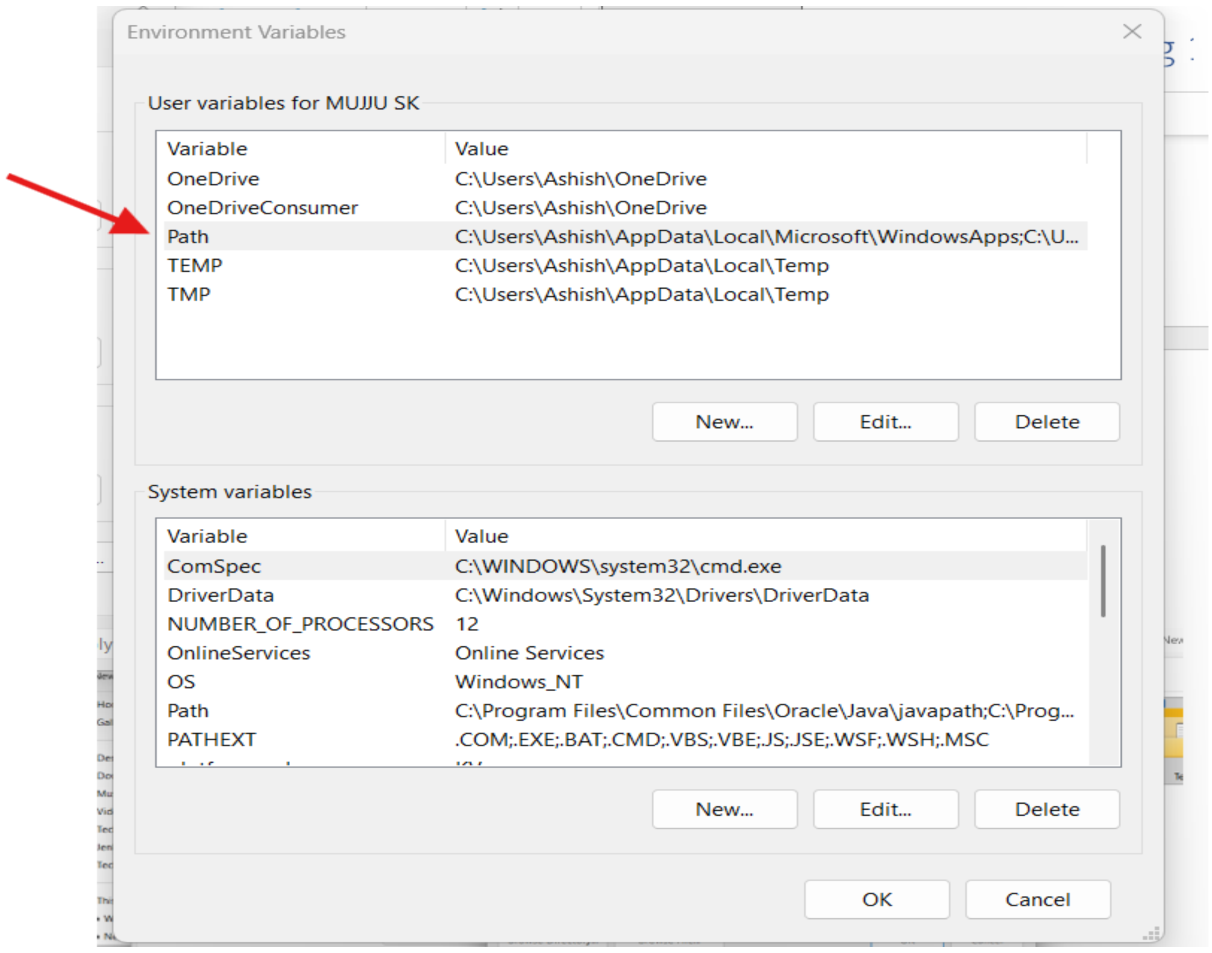
User Profiles

Desktop settings related to your sign-in

Startup and Recovery

System startup, system failure, and debugging information

- In that first table click on path and click on edit give your extracted directory path and click on save.



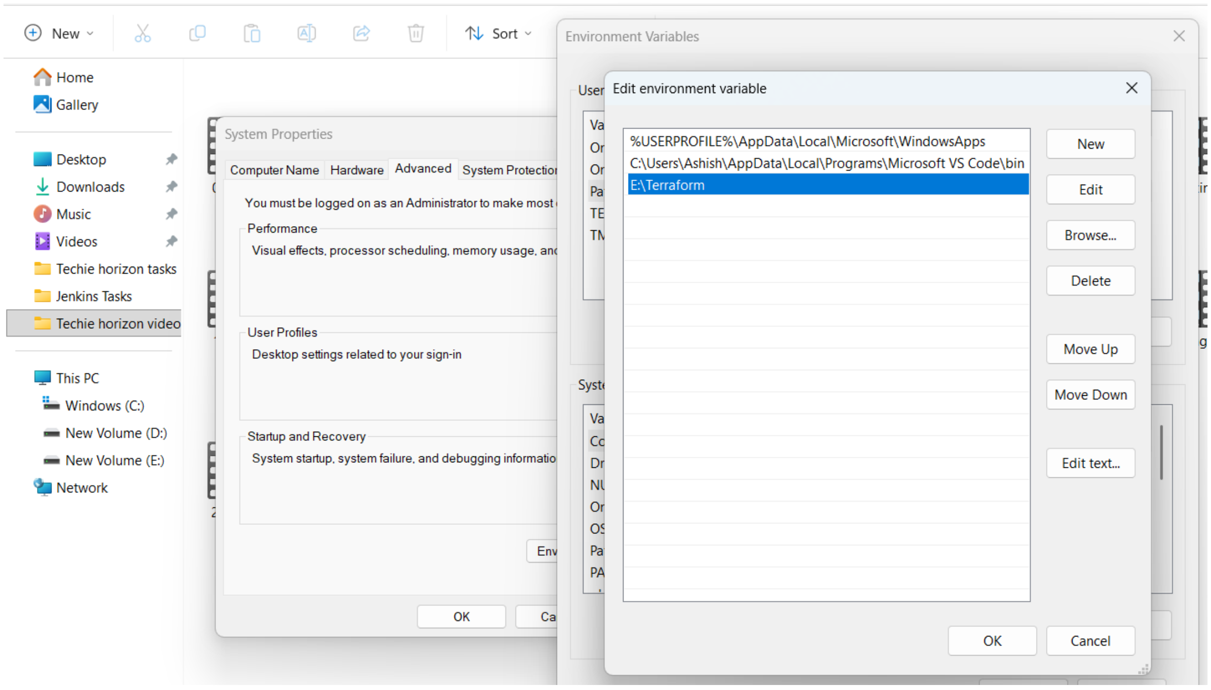
Environment Variables

User variables for MUJJU SK

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

System variables

Variable	Value
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	12
OnlineServices	Online Services
OS	Windows_NT
Path	C:\Program Files\Common Files\Oracle\Java\javapath;C:\Prog...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC



You must be logged on as an Administrator to make most

Performance
Visual effects, processor scheduling, memory usage, and

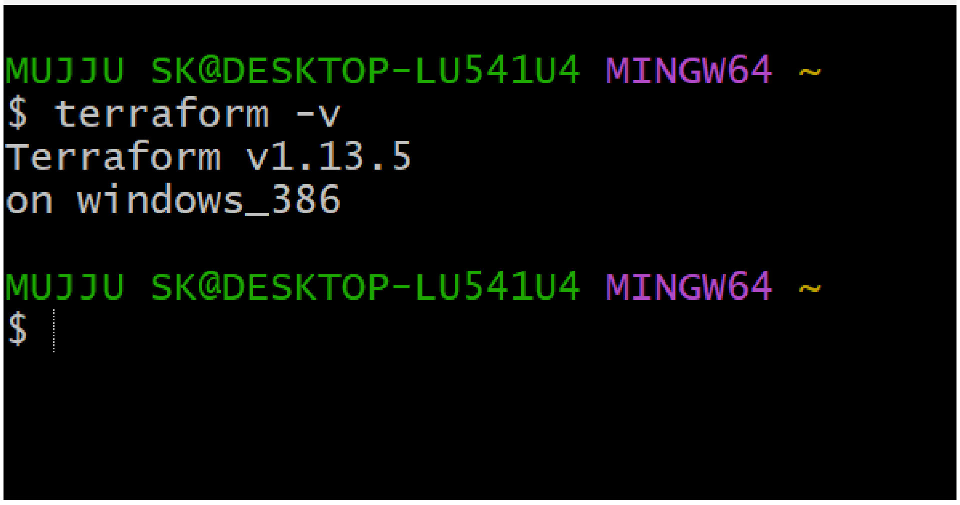
User Profiles
Desktop settings related to your sign-in

Startup and Recovery
System startup, system failure, and debugging informatio

Edit environment variable

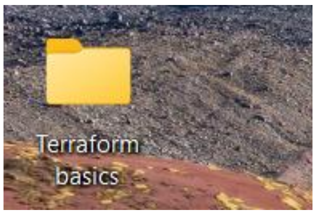
```
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\Users\Ashish\AppData\Local\Programs\Microsoft VS Code\bin
E:\Terraform
```

Check the version.

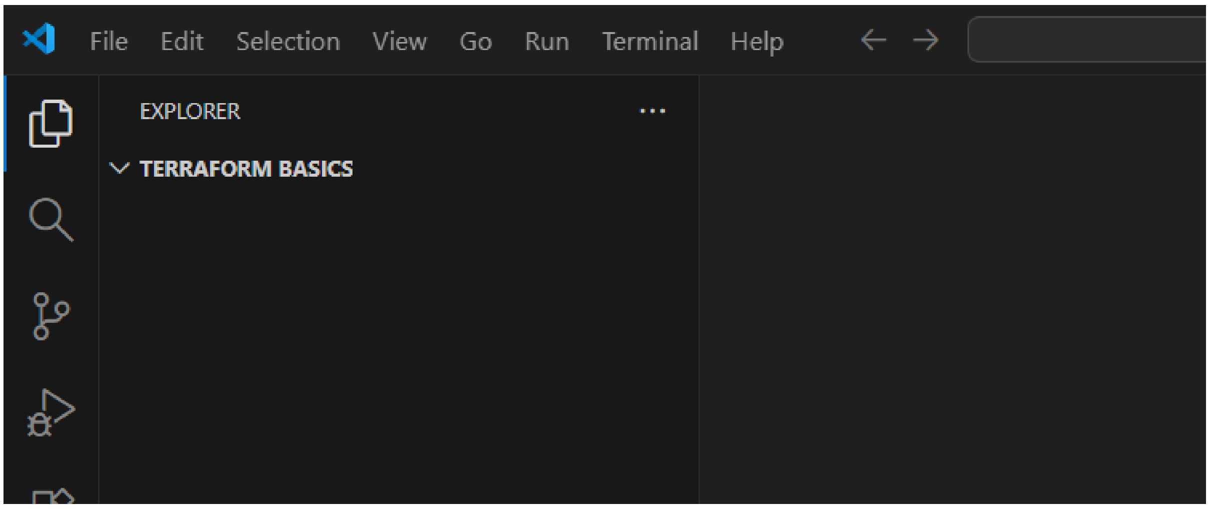


```
MUJJU SK@DESKTOP-LU541U4 MINGW64 ~
$ terraform -v
Terraform v1.13.5
on windows_386

MUJJU SK@DESKTOP-LU541U4 MINGW64 ~
$ 
```

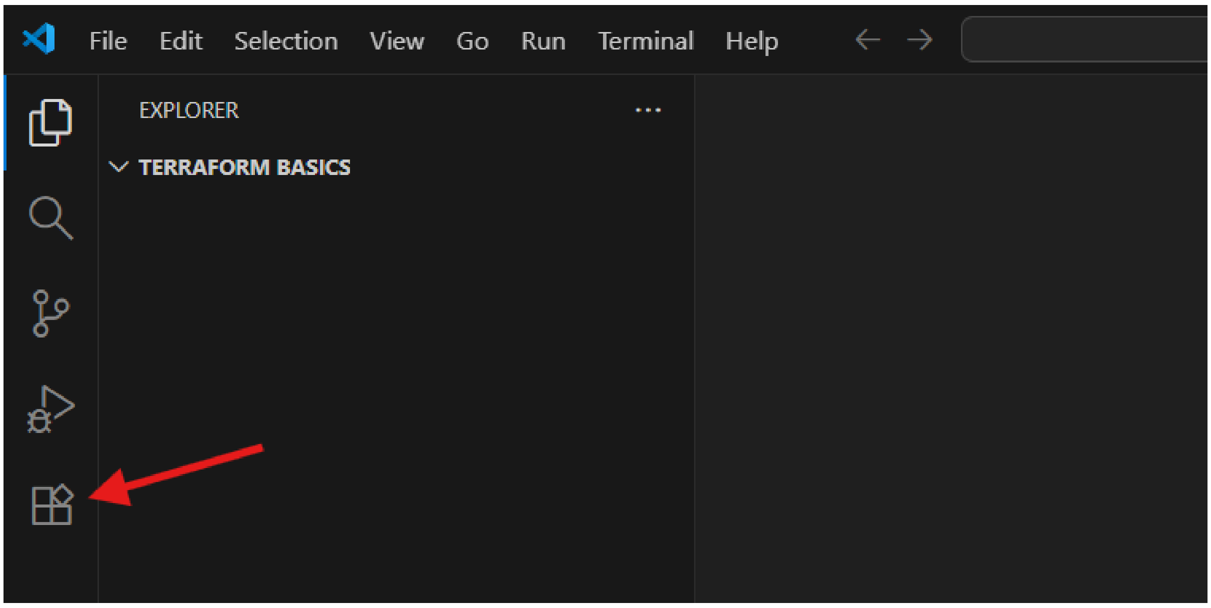
- Create a folder in the desktop as terraform basics.



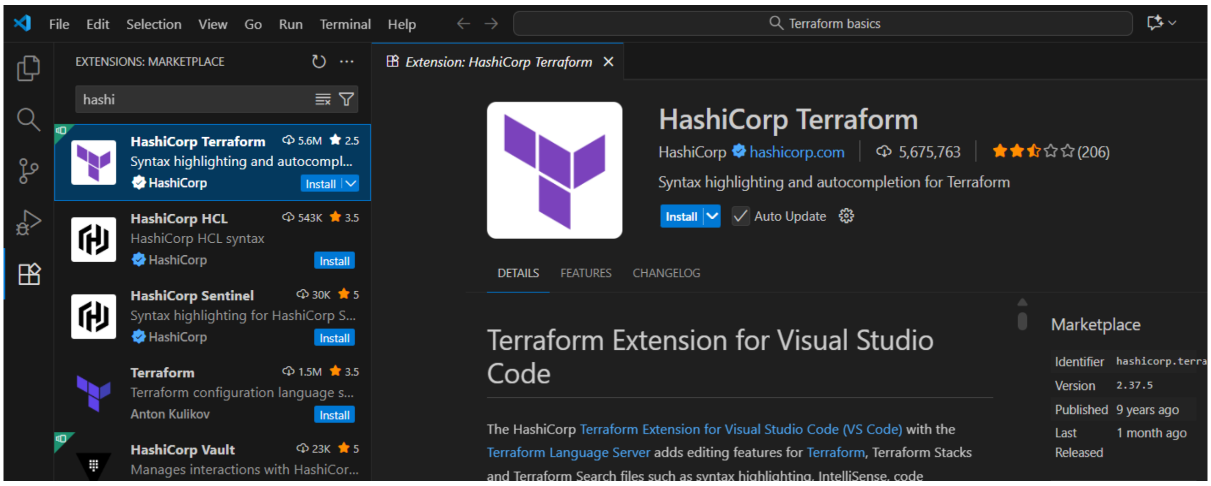
Open visual studio app and click on file, click on open folder and select that you created as terraform baics in desktop.



Click on extensions.



Search in vs studio as hashocrop terraform and install that plugin.



HashiCorp Terraform 5.6M 2.5
Syntax highlighting and autocompl...
HashiCorp Install

HashiCorp HCL 543K 3.5
HashiCorp HCL syntax
HashiCorp Install

HashiCorp Sentinel 30K 5
Syntax highlighting for HashiCorp S...
HashiCorp Install

Terraform 1.5M 3.5
Terraform configuration language s...
Anton Kulikov Install

HashiCorp Vault 23K 5
Manages interactions with HashiCor...

HashiCorp Terraform
HashiCorp hashicorp.com | 5,675,763 | (206)
Syntax highlighting and autocompletion for Terraform
Install | Auto Update

DETAILS FEATURES CHANGELOG

Terraform Extension for Visual Studio Code

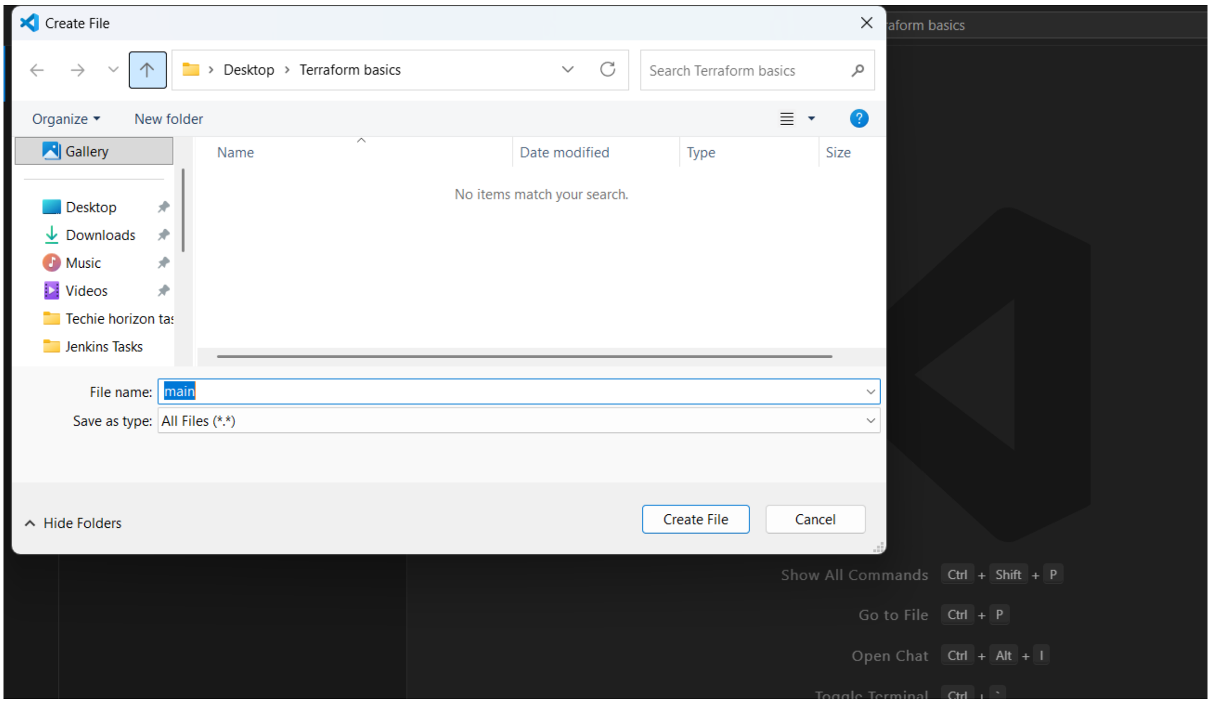
The HashiCorp Terraform Extension for Visual Studio Code (VS Code) with the Terraform Language Server adds editing features for Terraform, Terraform Stacks and Terraform Search files such as syntax highlighting, IntelliSense, code

Marketplace

Identifier	hashicorp.terra
Version	2.37.5
Published	9 years ago
Last Released	1 month ago

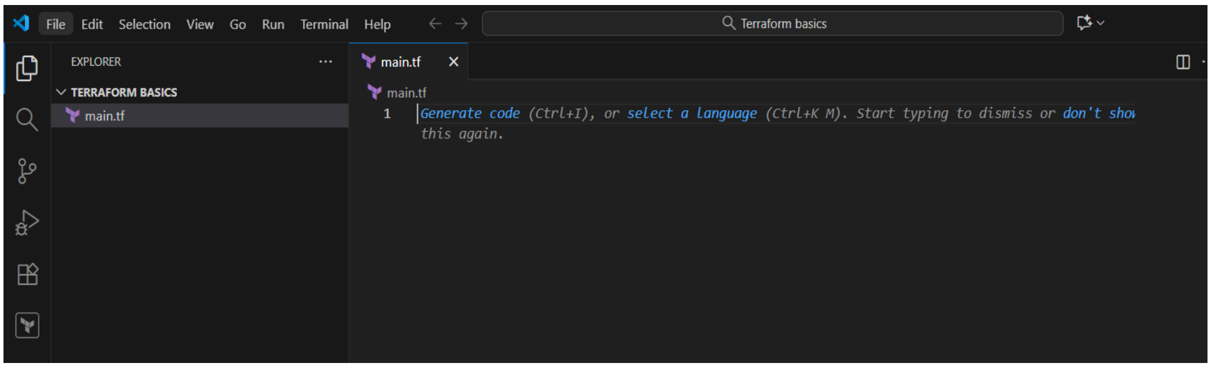
2. Execute all the templates shown in video.

Click on add new file and name it as main.tf and save it in the terraformbasics folder.



File name: main

Save as type: All Files (*.*)



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

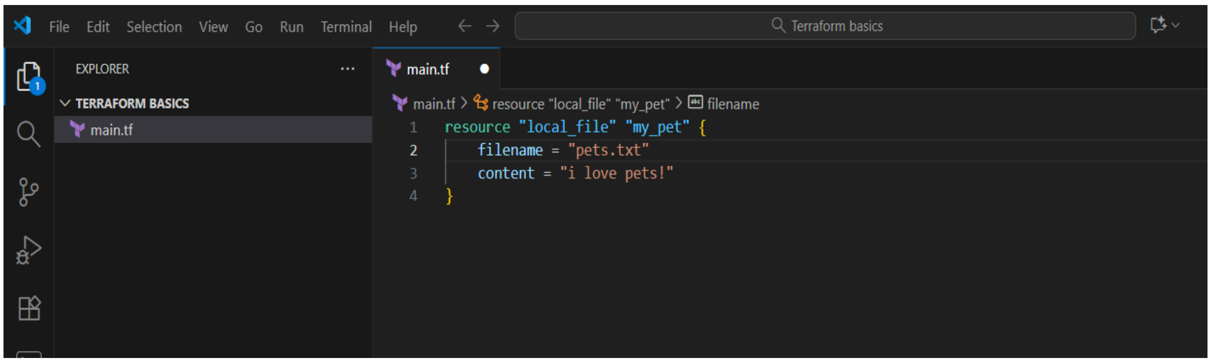
Give this code.

resource "local_file" "my_pet" {

filename = "pets.txt"

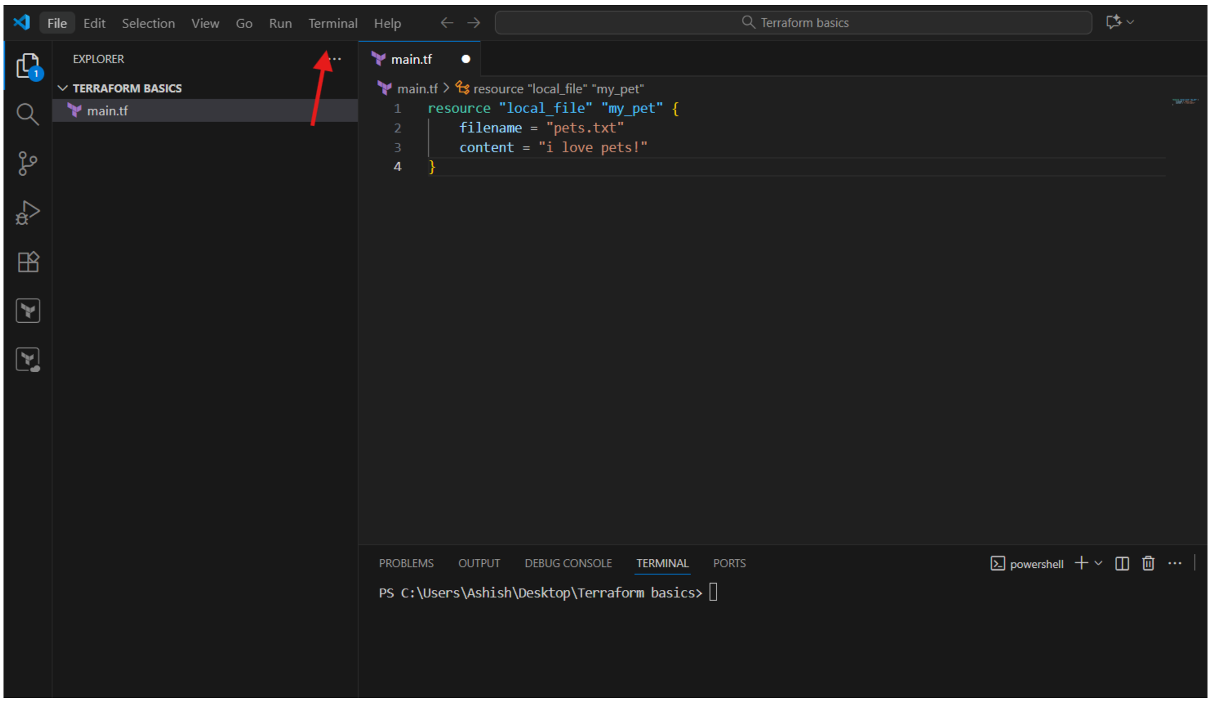
content = "i love pets!"

}



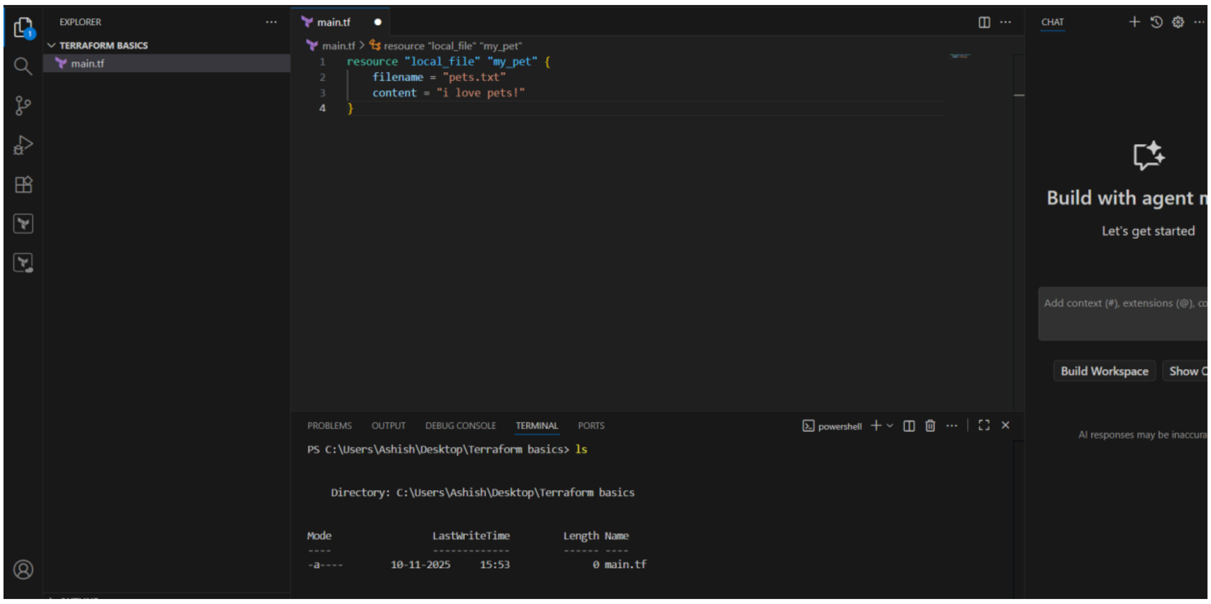
```
resource "local_file" "my_pet" {
    filename = "pets.txt"
    content = "i love pets!"
}
```

To execute this code click on terminal and select new terminal.



```
resource "local_file" "my_pet" {
    filename = "pets.txt"
    content = "i love pets!"
}
```

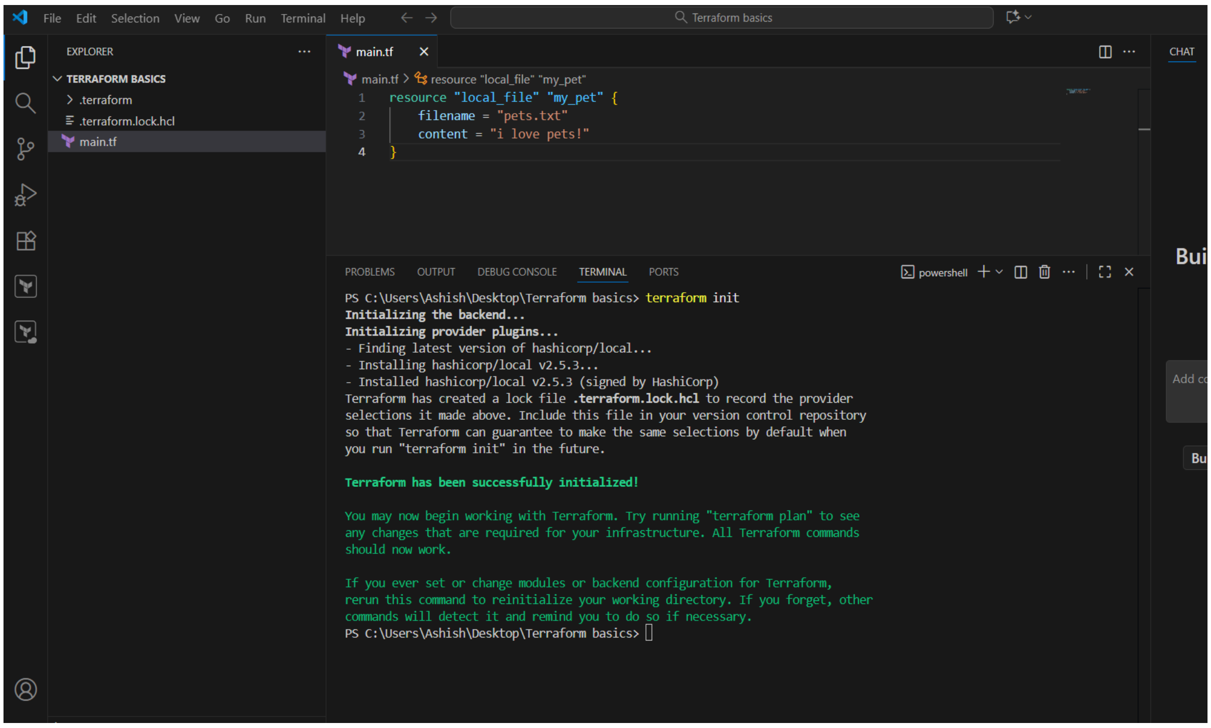
```
PS C:\Users\Ashish\Desktop\Terraform basics>
```



```
resource "local_file" "my_pet" {
    filename = "pets.txt"
    content = "i love pets!"
}
```

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


    Directory: C:\Users\Ashish\Desktop\Terraform basics


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        10-11-2025     15:53              0 main.tf
```

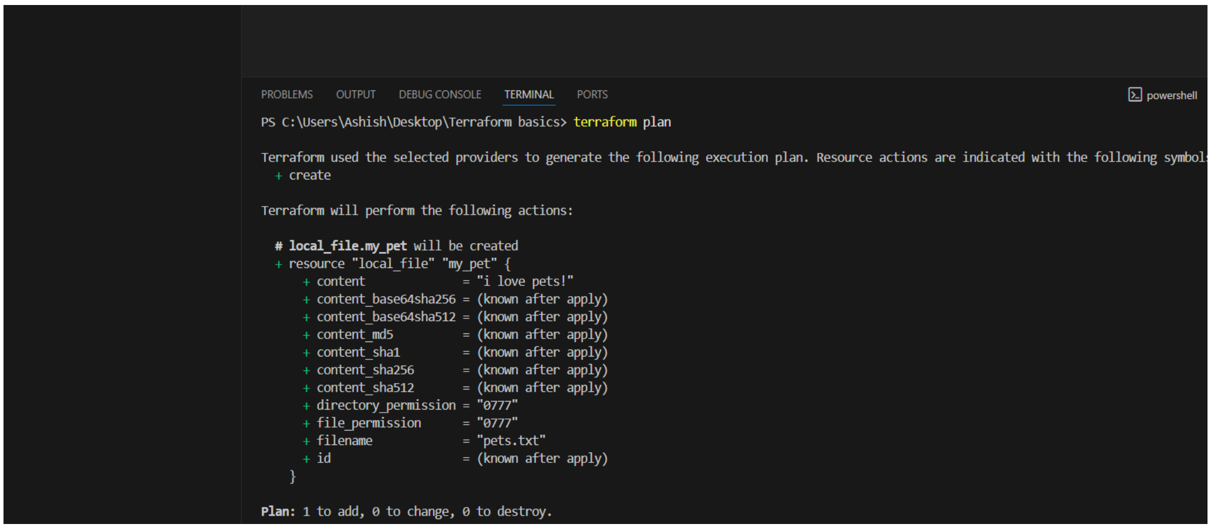
- terraform init



```
resource "local_file" "my_pet" {
    filename = "pets.txt"
    content = "i love pets!"
}
```

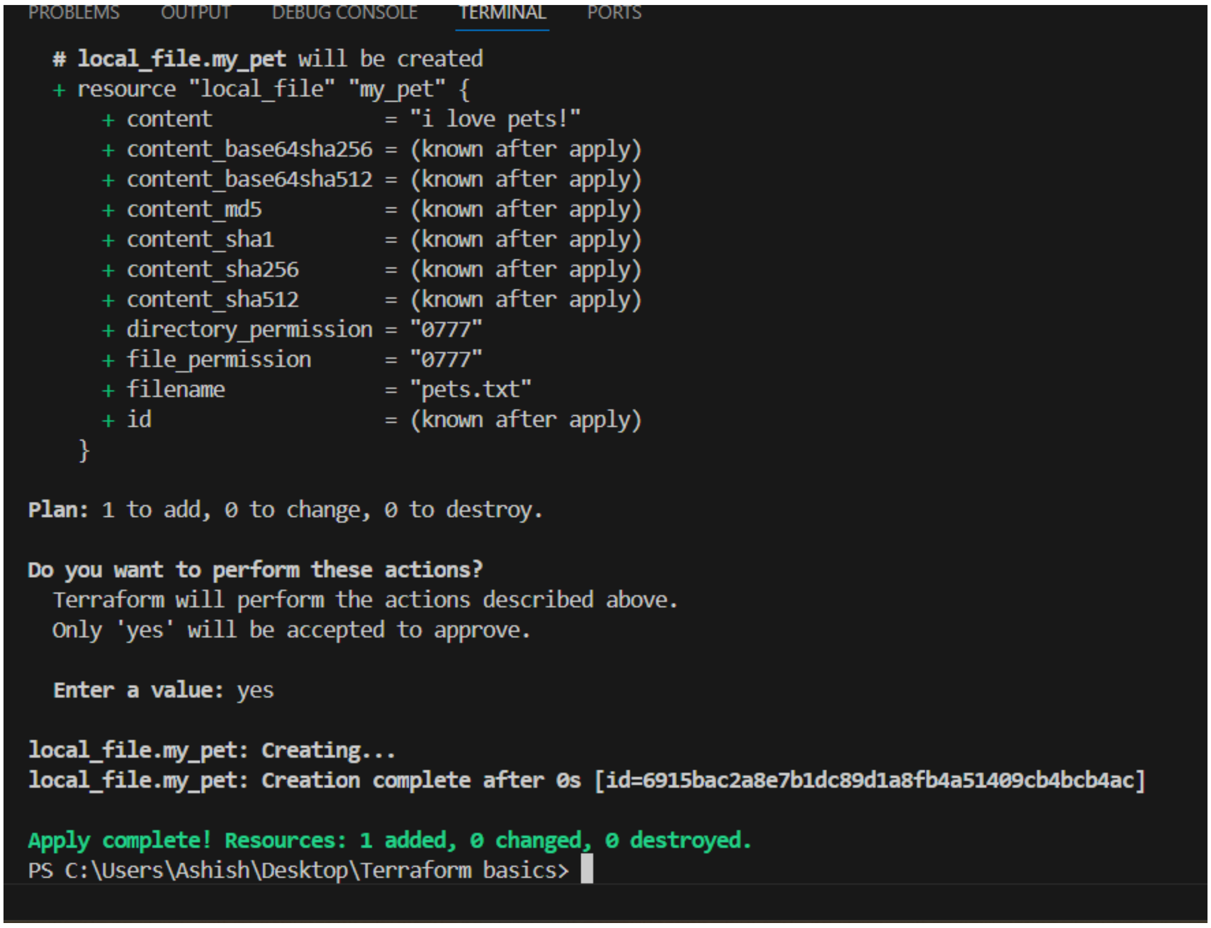
```
PS C:\Users\Ashish\Desktop\Terraform basics> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/local...
- Installing hashicorp/local v2.5.3...
- Installed hashicorp/local v2.5.3 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

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

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

- terraform plan



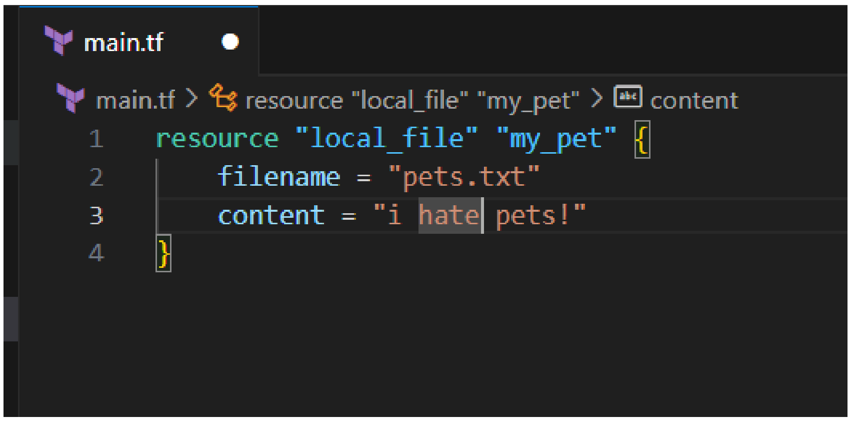
```
PS C:\Users\Ashish\Desktop\Terraform basics> terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbol
  + 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             = "pets.txt"
      + id                   = (known after apply)
    }

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

- terraform apply



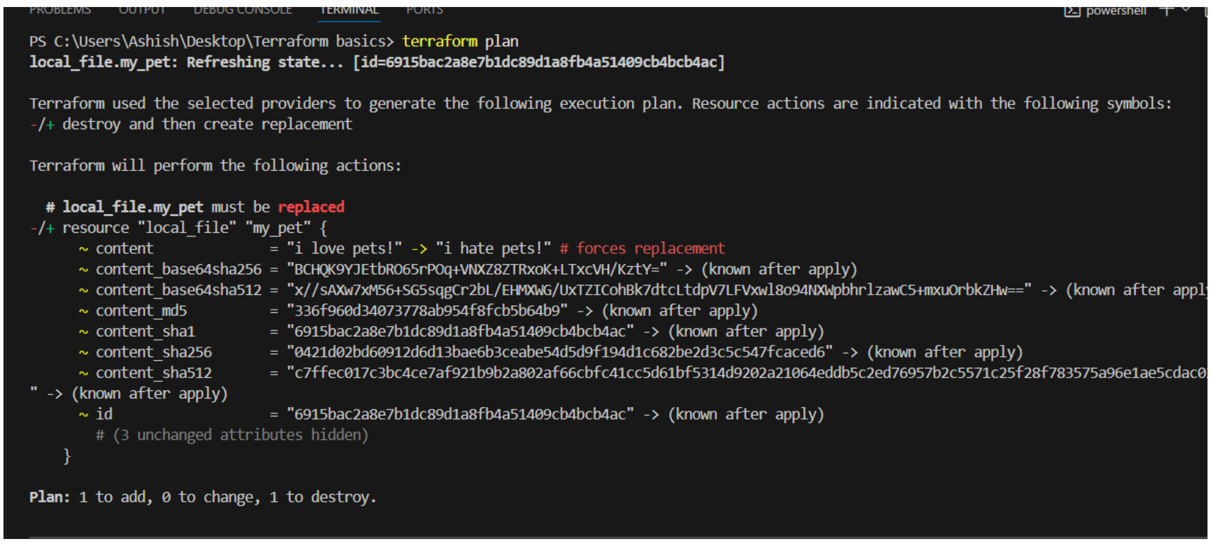
```
  # local_file.my_pet will be created
  + resource "local_file" "my_pet" {
      + content              = "i love pets!"
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "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.my_pet: Creating...
local_file.my_pet: Creation complete after 0s [id=6915bac2a8e7b1dc89d1a8fb4a51409cb4bcb4ac]

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

If I change the content love to hate.



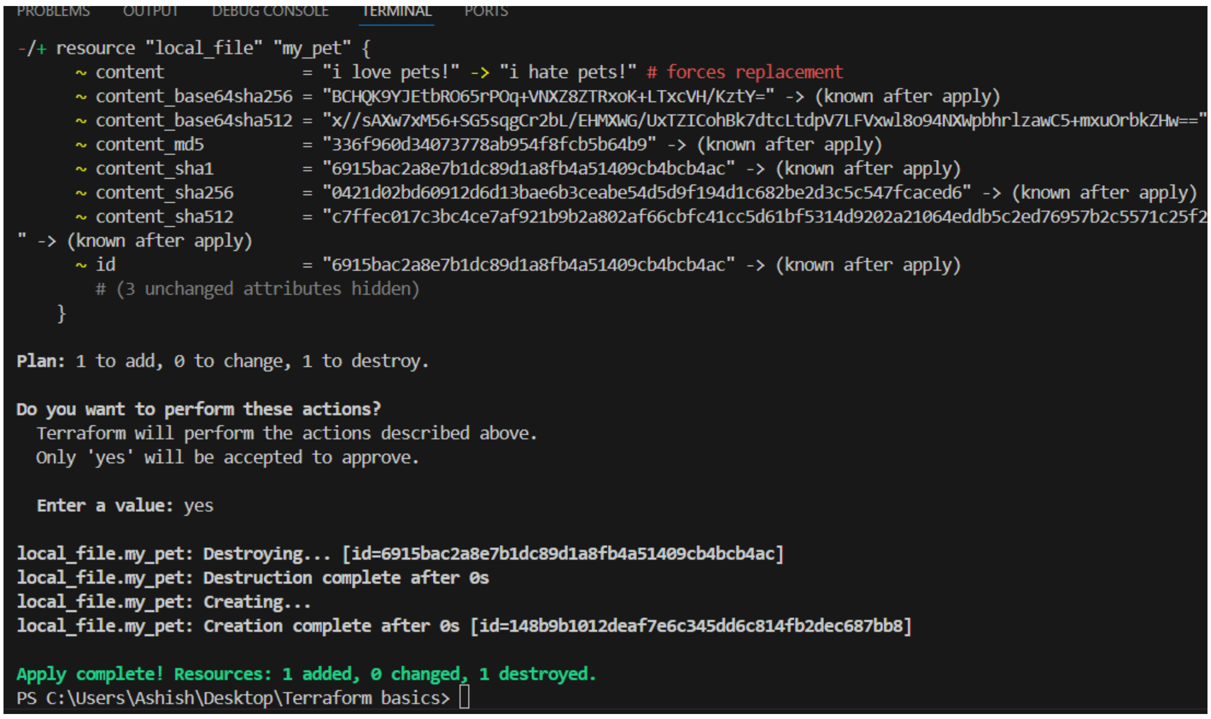
```
resource "local_file" "my_pet" {
    filename = "pets.txt"
    content = "i hate pets!"
}
```

- terraform plan



```
PS C:\Users\Ashish\Desktop\Terraform basics> terraform plan
local_file.my_pet: Refreshing state... [id=6915bac2a8e7b1dc89d1a8fb4a51409cb4bcb4ac]

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

Terraform will perform the following actions:

  # local_file.my_pet must be replaced
-/+ resource "local_file" "my_pet" {
      ~ content              = "i love pets!" -> "i hate pets!" # forces replacement
      ~ content_base64sha256 = "BCHQK9YJEtbRO65rPOq+VNXZ8ZTRxoK+LTxcVH/KztY=" -> (known after apply)
      ~ content_base64sha512 = "x//sAXw7xM56+SG5sqgCr2bL/EHMXWG/UxTZICohBk7dtcLtdpV7LFVxwl8o94NXWpbhrlzawC5+mxuOrbkZHw==" -> (known after appl
      ~ content_md5          = "336f960d34073778ab954f8fcb5b64b9" -> (known after apply)
      ~ content_sha1         = "6915bac2a8e7b1dc89d1a8fb4a51409cb4bcb4ac" -> (known after apply)
      ~ content_sha256       = "0421d02bd60912d6d13bae6b3ceabe54d5d9f194d1c682be2d3c5c547fcaced6" -> (known after apply)
      ~ content_sha512       = "c7ffec017c3bc4ce7af921b9b2a802af66cbfc41cc5d61bf5314d9202a21064eddb5c2ed76957b2c5571c25f28f783575a96e1ae5cdac0
" -> (known after apply)
      ~ id                   = "6915bac2a8e7b1dc89d1a8fb4a51409cb4bcb4ac" -> (known after apply)
        # (3 unchanged attributes hidden)
    }

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

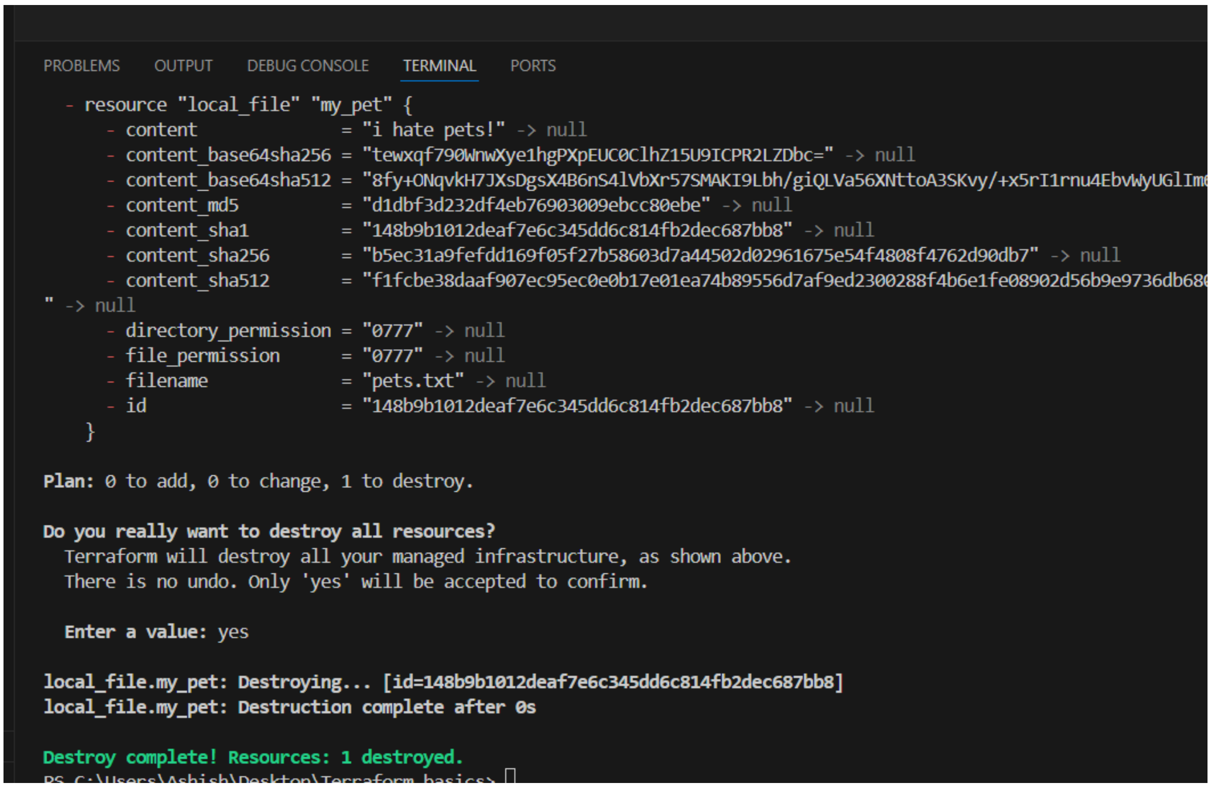
- terraform apply



```
-/+ resource "local_file" "my_pet" {
      ~ content              = "i love pets!" -> "i hate pets!" # forces replacement
      ~ content_base64sha256 = "BCHQK9YJEtbRO65rPOq+VNXZ8ZTRxoK+LTxcVH/KztY=" -> (known after apply)
      ~ content_base64sha512 = "x//sAXw7xM56+SG5sqgCr2bL/EHMXWG/UxTZICohBk7dtcLtdpV7LFVxwl8o94NXWpbhrlzawC5+mxuOrbkZHw=="
      ~ content_md5          = "336f960d34073778ab954f8fcb5b64b9" -> (known after apply)
      ~ content_sha1         = "6915bac2a8e7b1dc89d1a8fb4a51409cb4bcb4ac" -> (known after apply)
      ~ content_sha256       = "0421d02bd60912d6d13bae6b3ceabe54d5d9f194d1c682be2d3c5c547fcaced6" -> (known after apply)
      ~ content_sha512       = "c7ffec017c3bc4ce7af921b9b2a802af66cbfc41cc5d61bf5314d9202a21064eddb5c2ed76957b2c5571c25f2
" -> (known after apply)
      ~ id                   = "6915bac2a8e7b1dc89d1a8fb4a51409cb4bcb4ac" -> (known after apply)
        # (3 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.my_pet: Destroying... [id=6915bac2a8e7b1dc89d1a8fb4a51409cb4bcb4ac]
local_file.my_pet: Destruction complete after 0s
local_file.my_pet: Creating...
local_file.my_pet: Creation complete after 0s [id=148b9b1012deaf7e6c345dd6c814fb2dec687bb8]

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

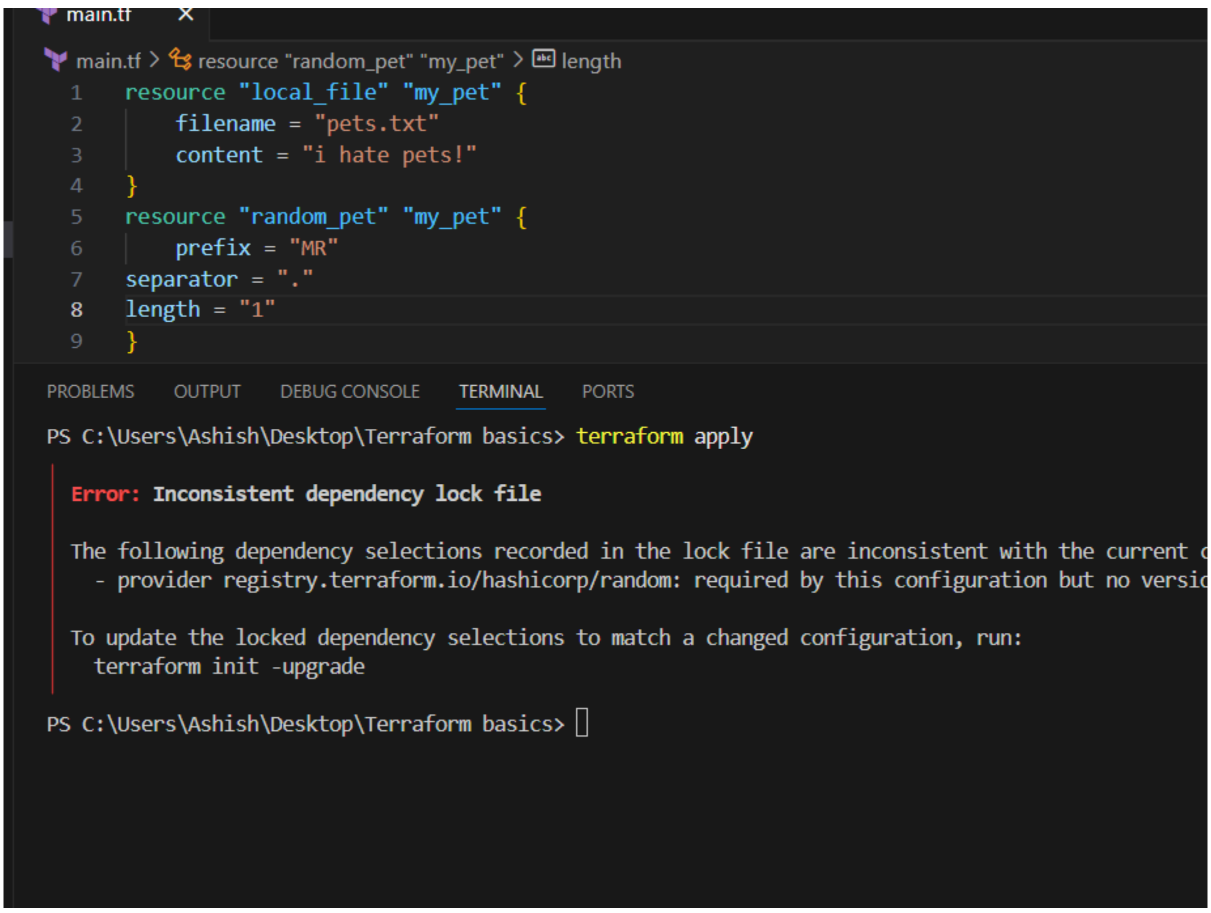
- terraform destroy

it will delete the pets.txt file.



```
  - resource "local_file" "my_pet" {
      - content              = "i hate pets!" -> null
      - content_base64sha256 = "tewxqf790WnwXye1hgPXpEUC0ClhZ15U9ICPR2LZDbc=" -> null
      - content_base64sha512 = "8fy+ONqvkH7JXsDgsX4B6nS4lVbXr57SMAKI9Lbh/giQLVa56XNttoA3SKvy/+x5rI1rnu4EbvWyUGlIm
      - content_md5          = "d1dbf3d232df4eb76903009ebcc80ebe" -> null
      - content_sha1         = "148b9b1012deaf7e6c345dd6c814fb2dec687bb8" -> null
      - content_sha256       = "b5ec31a9fefdd169f05f27b58603d7a44502d02961675e54f4808f4762d90db7" -> null
      - content_sha512       = "f1fcbe38daaf907ec95ec0e0b17e01ea74b89556d7af9ed2300288f4b6e1fe08902d56b9e9736db68
" -> null
      - directory_permission = "0777" -> null
      - file_permission      = "0777" -> null
      - filename             = "pets.txt" -> null
      - id                   = "148b9b1012deaf7e6c345dd6c814fb2dec687bb8" -> null
    }

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

Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

local_file.my_pet: Destroying... [id=148b9b1012deaf7e6c345dd6c814fb2dec687bb8]
local_file.my_pet: Destruction complete after 0s

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

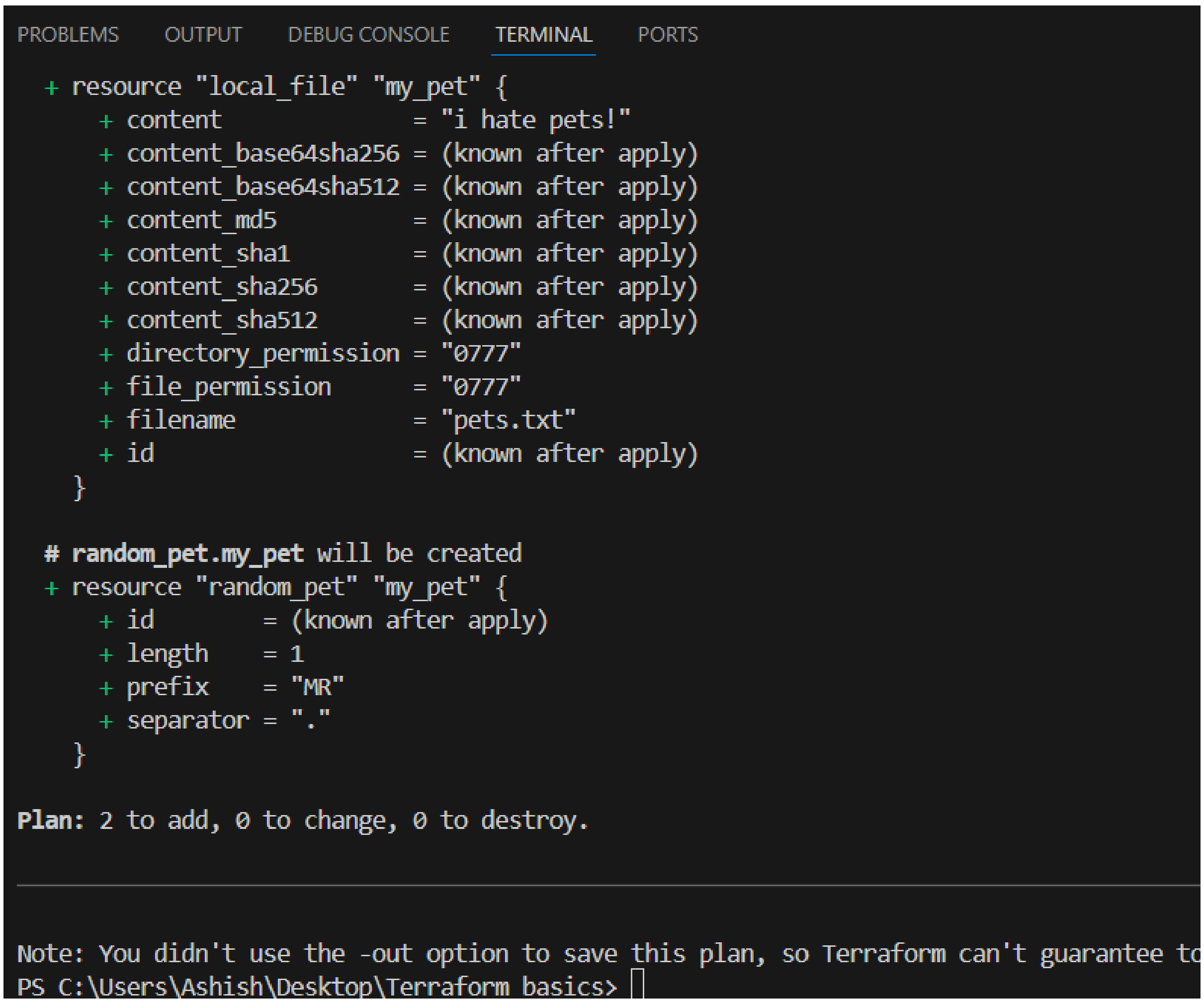
Add random.pet file to the code



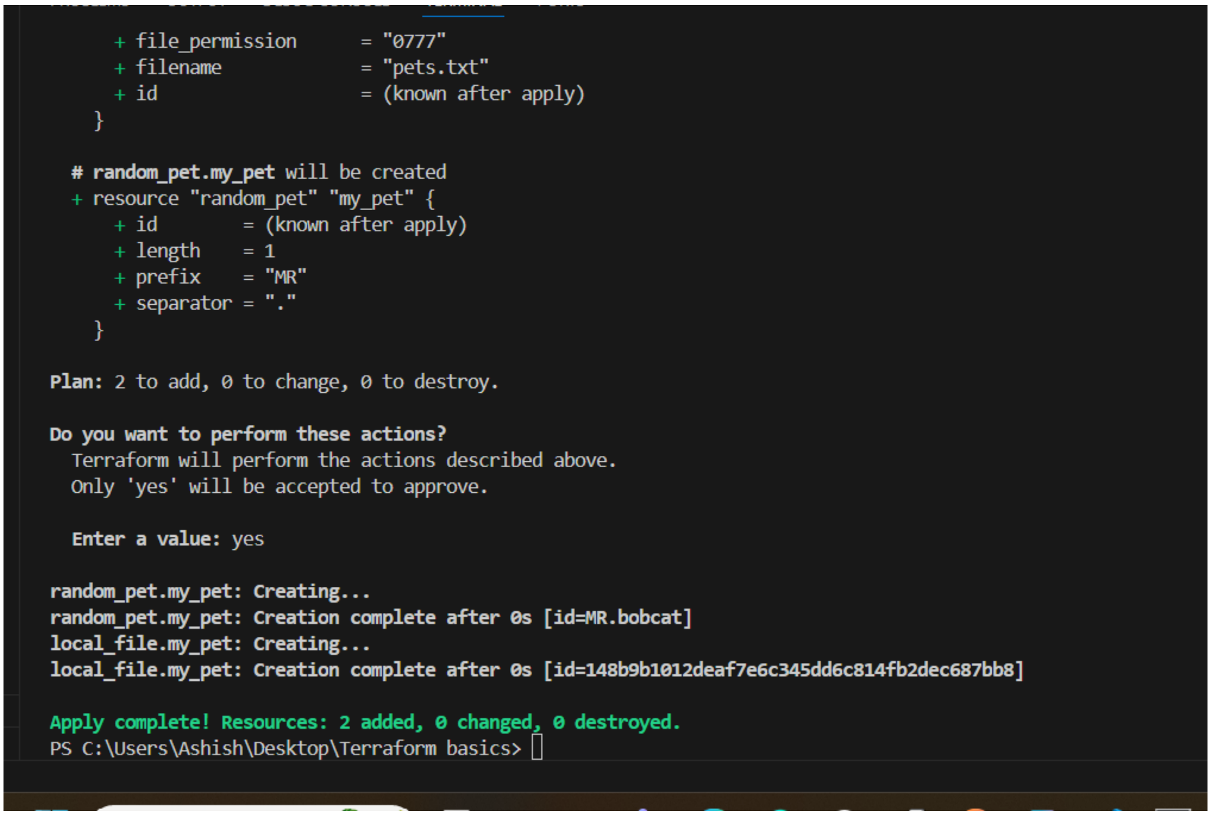
```
resource "local_file" "my_pet" {
    filename = "pets.txt"
    content = "i hate pets!"
}
resource "random_pet" "my_pet" {
    prefix = "MR"
separator = "."
length = "1"
}
```

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

  Error: Inconsistent dependency lock file

  The following dependency selections recorded in the lock file are inconsistent with the current 
    - provider registry.terraform.io/hashicorp/random: required by this configuration but no versi

  To update the locked dependency selections to match a changed configuration, run:
    terraform init -upgrade

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

This time error found then we need to do terraform init and terraform add.



```
  + resource "local_file" "my_pet" {
      + content              = "i hate 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             = "pets.txt"
      + id                   = (known after apply)
    }

  # random_pet.my_pet will be created
  + resource "random_pet" "my_pet" {
      + id        = (known after apply)
      + length    = 1
      + prefix    = "MR"
      + separator = "."
    }

Plan: 2 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
PS C:\Users\Ashish\Desktop\Terraform basics>
```



```
      + file_permission      = "0777"
      + filename             = "pets.txt"
      + id                   = (known after apply)
    }

  # random_pet.my_pet will be created
  + resource "random_pet" "my_pet" {
      + id        = (known after apply)
      + length    = 1
      + prefix    = "MR"
      + separator = "."
    }

Plan: 2 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

random_pet.my_pet: Creating...
random_pet.my_pet: Creation complete after 0s [id=MR.bobcat]
local_file.my_pet: Creating...
local_file.my_pet: Creation complete after 0s [id=148b9b1012deaf7e6c345dd6c814fb2dec687bb8]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
PS C:\Users\Ashish\Desktop\Terraform basics>
```

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

Terraform init:

The command **terraform init** is used to **initialize** a Terraform working directory before you can use Terraform commands like plan, apply, or destroy.

- It prepares your folder (with .tf files) so Terraform knows how to run there.

Terraform plan:

The command **terraform plan** is used to **preview** what Terraform will do before actually making any changes to your infrastructure.

- Which **resources will be created**
- Which **will be modified**
- Which **will be destroyed**
- Any **errors or conflicts** before applying

Terraform apply:

The command **terraform apply** is used to **create or modify real infrastructure** as defined in your Terraform configuration files (.tf).

- **Builds** new infrastructure
- **Updates** existing infrastructure
- **Destroys** resources that are no longer in your config
- Saves the new **state** to your backend.

Terraform provider:

The **terraform provider** concept refers to the **plugins** that Terraform uses to interact with different **clouds, services, and APIs** — such as AWS, Azure, Google Cloud, Kubernetes, GitHub, etc.

How terraform uses providers:

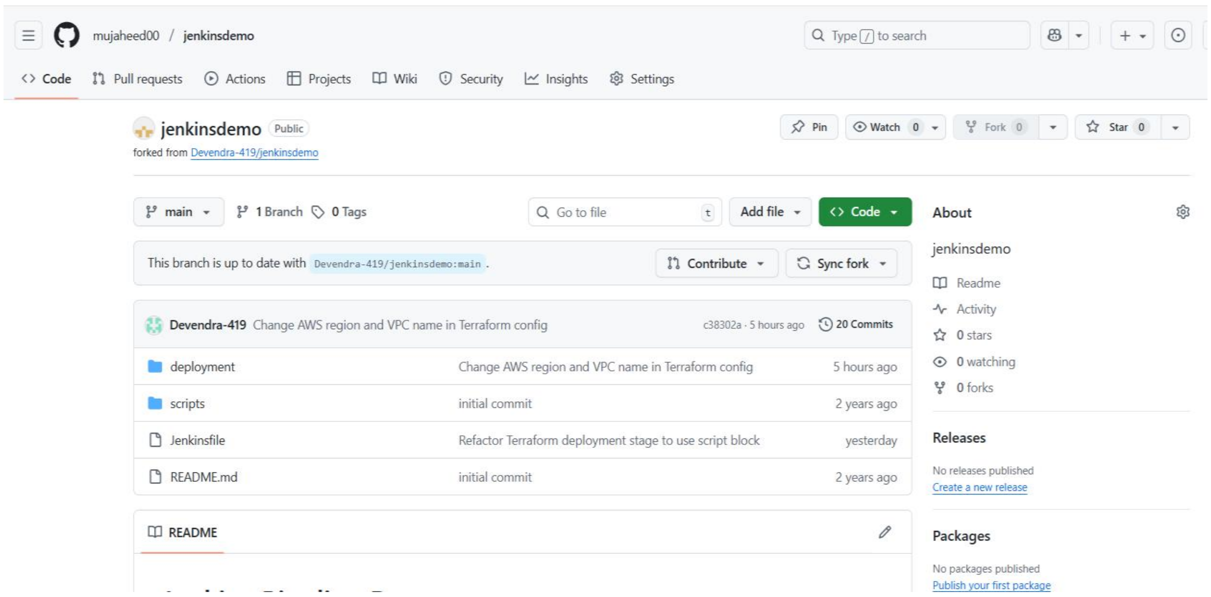
1. You define a provider in your .tf files.
2. terraform init downloads the necessary provider binaries into .terraform/providers/.

3. terraform plan and terraform apply use that provider to communicate with the cloud API.

4. Integrate a sample Terraform template in Jenkins.

Create a repository in github and add necessary documents into that like jenkinsfile,main.tf.

https://github.com/mujaheed00/jenkinsdemo.git



jenkinsdemo Public
forked from Devendra-419/jenkinsdemo

This branch is up to date with Devendra-419/jenkinsdemo:main .

Devendra-419 Change AWS region and VPC name in Terraform config c38302a · 5 hours ago 20 Commits

deployment	Change AWS region and VPC name in Terraform config	5 hours ago
scripts	initial commit	2 years ago
Jenkinsfile	Refactor Terraform deployment stage to use script block	yesterday
README.md	initial commit	2 years ago

About

jenkinsdemo

Readme
Activity
0 stars
0 watching
0 forks

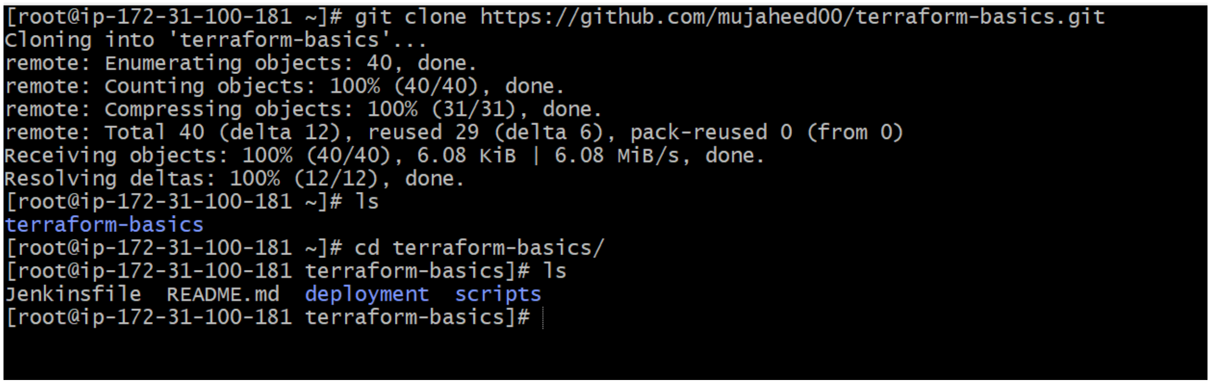
Releases

No releases published
Create a new release

Packages

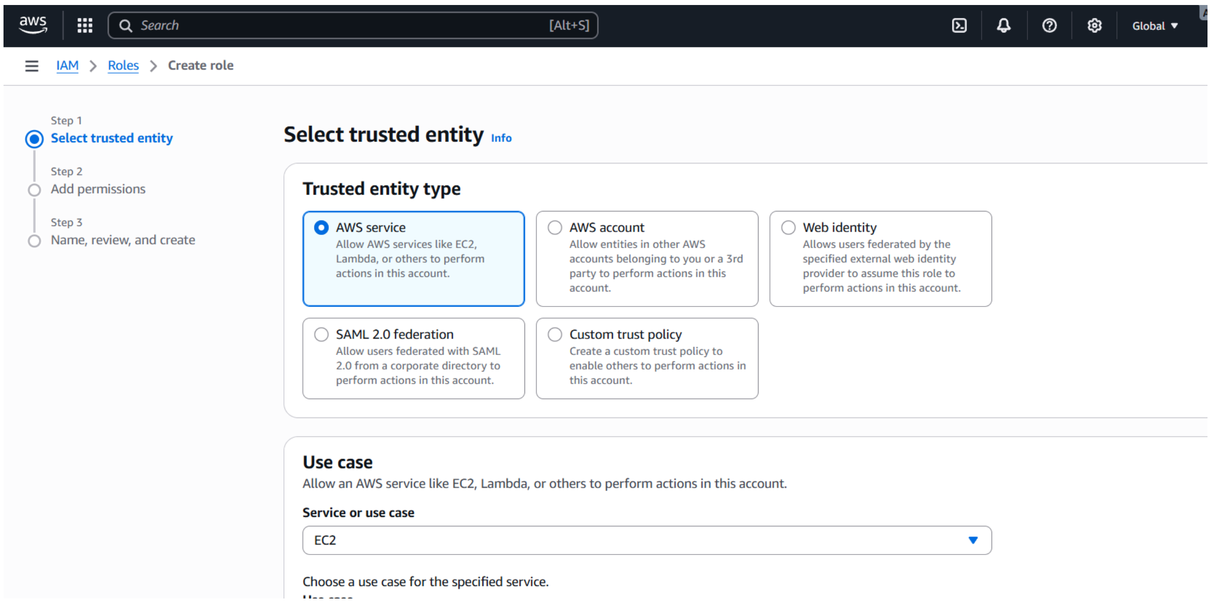
No packages published
Publish your first package

Git clone in Jenkins server



```
[root@ip-172-31-100-181 ~]# git clone https://github.com/mujaheed00/terraform-basics.git
Cloning into 'terraform-basics'...
remote: Enumerating objects: 40, done.
remote: Counting objects: 100% (40/40), done.
remote: Compressing objects: 100% (31/31), done.
remote: Total 40 (delta 12), reused 29 (delta 6), pack-reused 0 (from 0)
Receiving objects: 100% (40/40), 6.08 KiB | 6.08 MiB/s, done.
Resolving deltas: 100% (12/12), done.
[root@ip-172-31-100-181 ~]# ls
terraform-basics
[root@ip-172-31-100-181 ~]# cd terraform-basics/
[root@ip-172-31-100-181 terraform-basics]# ls
Jenkinsfile  README.md  deployment  scripts
[root@ip-172-31-100-181 terraform-basics]#
```

Create an IAM role


IAM > Roles > Create role

Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create

Select trusted entity Info

Trusted entity type

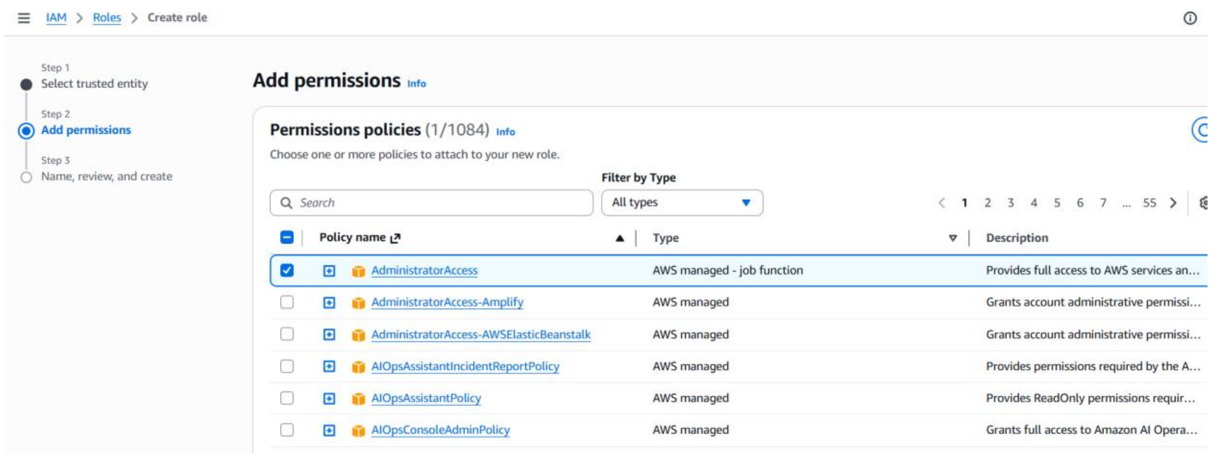
- [✓ AWS service] Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account: Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity: Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- SAML 2.0 federation: Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy: Create a custom trust policy to enable others to perform actions in this account.

Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case
EC2

Choose a use case for the specified service.

Select administrator access and save.


IAM > Roles > Create role

Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create

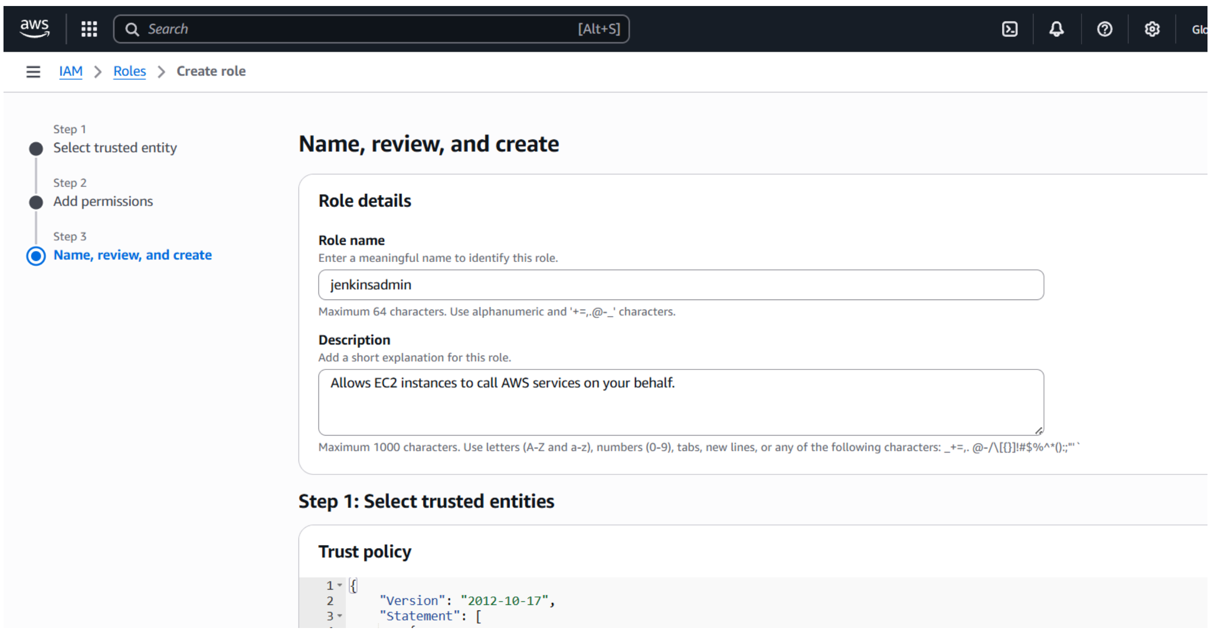
Add permissions Info

Permissions policies (1/1084) Info
Choose one or more policies to attach to your new role.

Filter by Type: All types

	Policy name	Type	Description
✓	AdministratorAccess	AWS managed - job function	Provides full access to AWS services an...
	AdministratorAccess-Amplify	AWS managed	Grants account administrative permissi...
	AdministratorAccess-AWSElasticBeanstalk	AWS managed	Grants account administrative permissi...
	AIOpsAssistantIncidentReportPolicy	AWS managed	Provides permissions required by the A...
	AIOpsAssistantPolicy	AWS managed	Provides ReadOnly permissions requir...
	AIOpsConsoleAdminPolicy	AWS managed	Grants full access to Amazon AI Opera...

Give name as jenkinsadmin and click on create role.



IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Name, review, and create

Role details

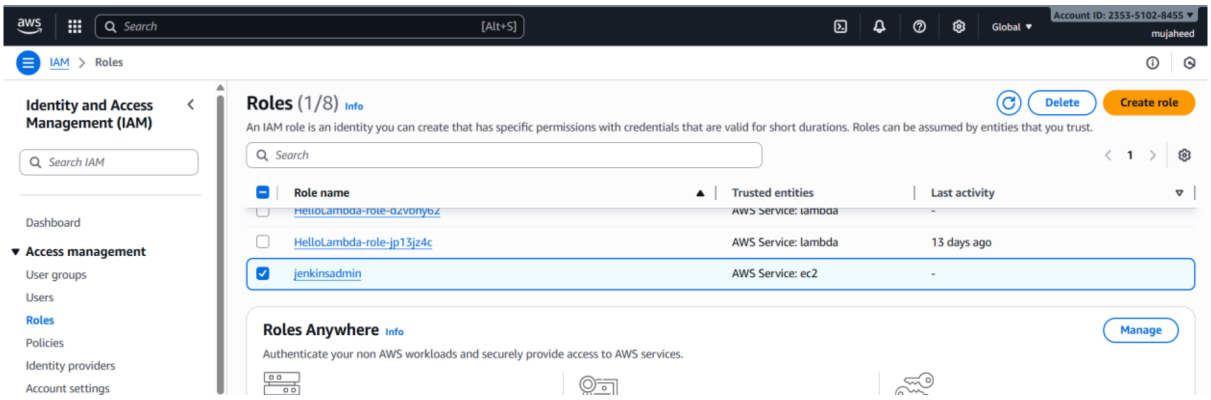
Role name
Enter a meaningful name to identify this role.
jenkinsadmin
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

Description
Add a short explanation for this role.
Allows EC2 instances to call AWS services on your behalf.
Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=,. @-/\[{}]!#$%^*();"'`

Step 1: Select trusted entities

Trust policy

```
{
    "Version": "2012-10-17",
    "Statement": [
```



IAM > Roles

Identity and Access Management (IAM)

Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings

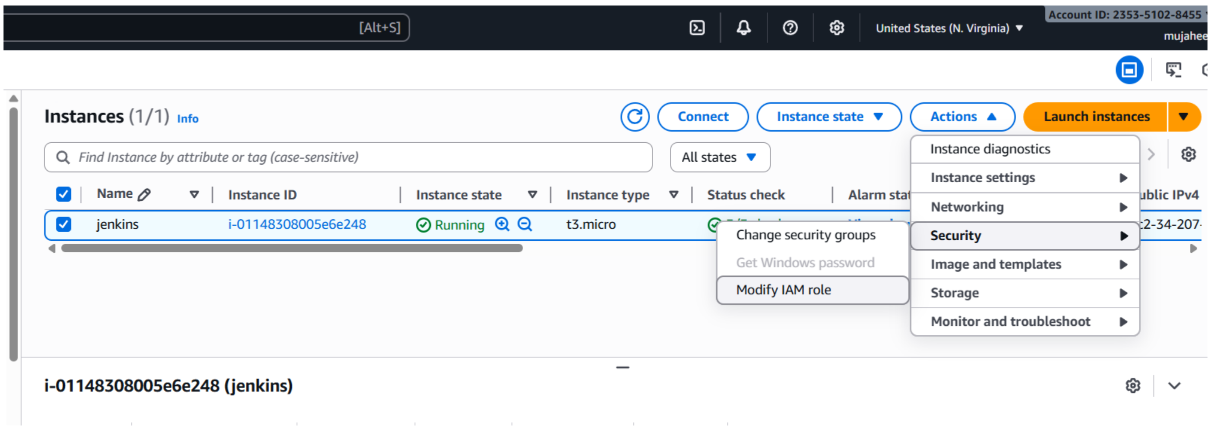
Roles (1/8) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

	Role name	Trusted entities	Last activity
☐	HelloLambda-role-dzvbny6z	AWS Service: lambda	-
☐	HelloLambda-role-jp13jz4c	AWS Service: lambda	13 days ago
☑	jenkinsadmin	AWS Service: ec2	-

Roles Anywhere Info
Authenticate your non AWS workloads and securely provide access to AWS services.

Go to your Jenkins ec2 instance and click on actions, click on security,click on modify IAM role.

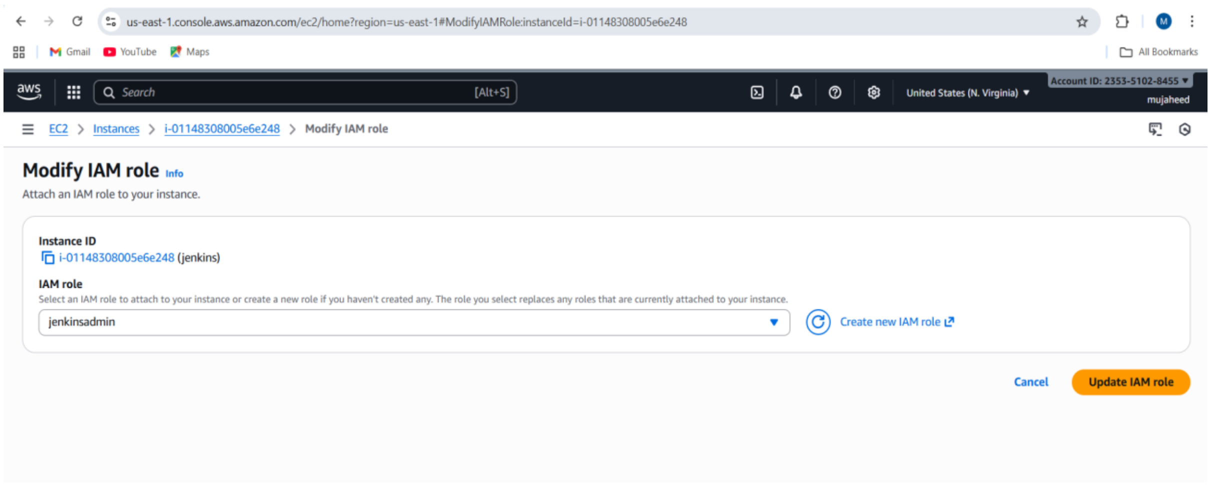


Instances (1/1) Info

	Name	Instance ID	Instance state	Instance type	Status check
☑	jenkins	i-01148308005e6e248	Running	t3.micro	

i-01148308005e6e248 (jenkins)

Select your created IAM role.



us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#ModifyIAMRole:instanceId=i-01148308005e6e248
Gmail
YouTube
Maps
All Bookmarks
Search
[Alt+S]
United States (N. Virginia) ▾
Account ID: 2353-5102-8455 ▾
mujaheed
EC2 > Instances > i-01148308005e6e248 > Modify IAM role

Modify IAM role Info

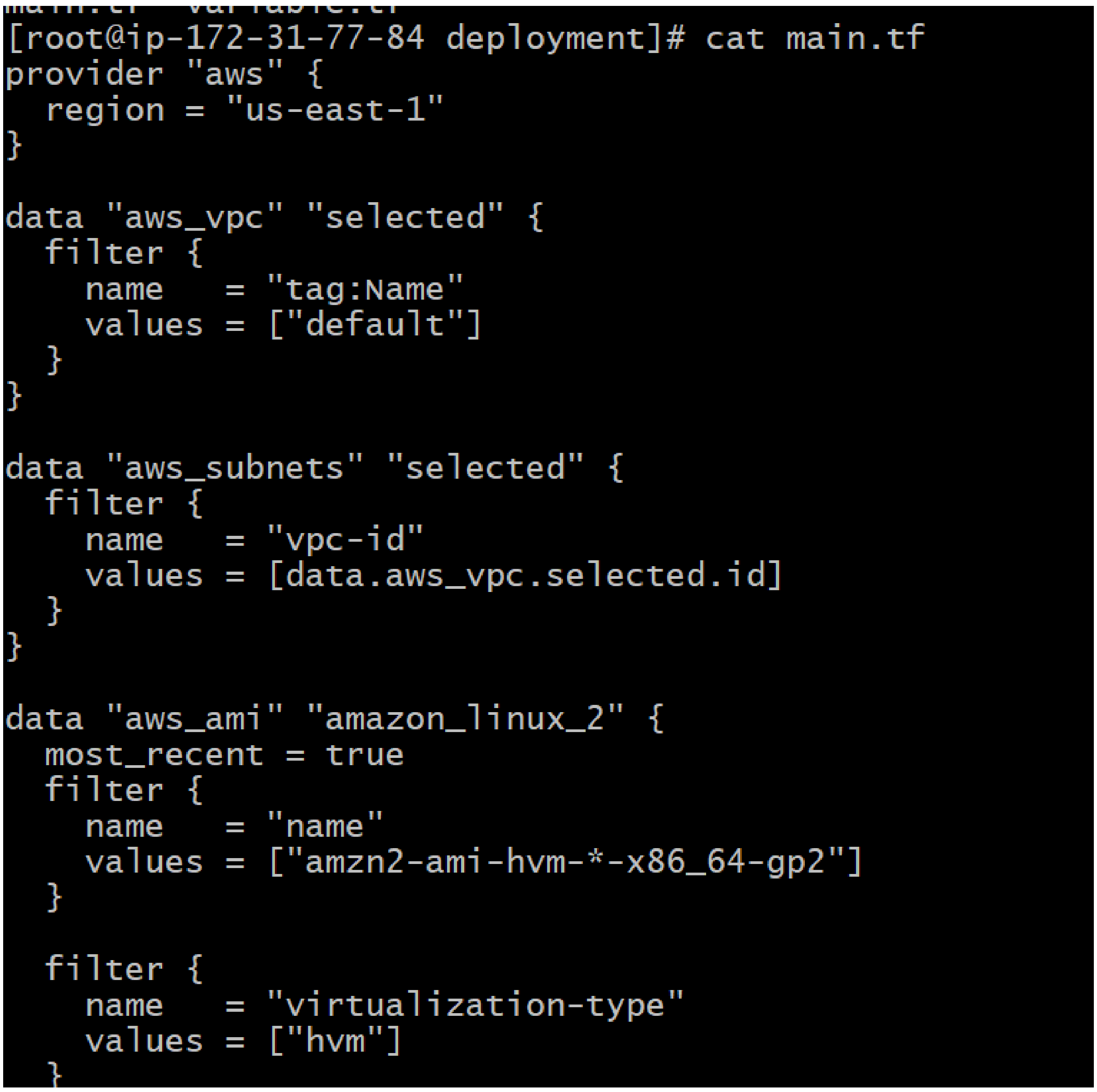
Attach an IAM role to your instance.

Instance ID
i-01148308005e6e248 (jenkins)

IAM role
Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

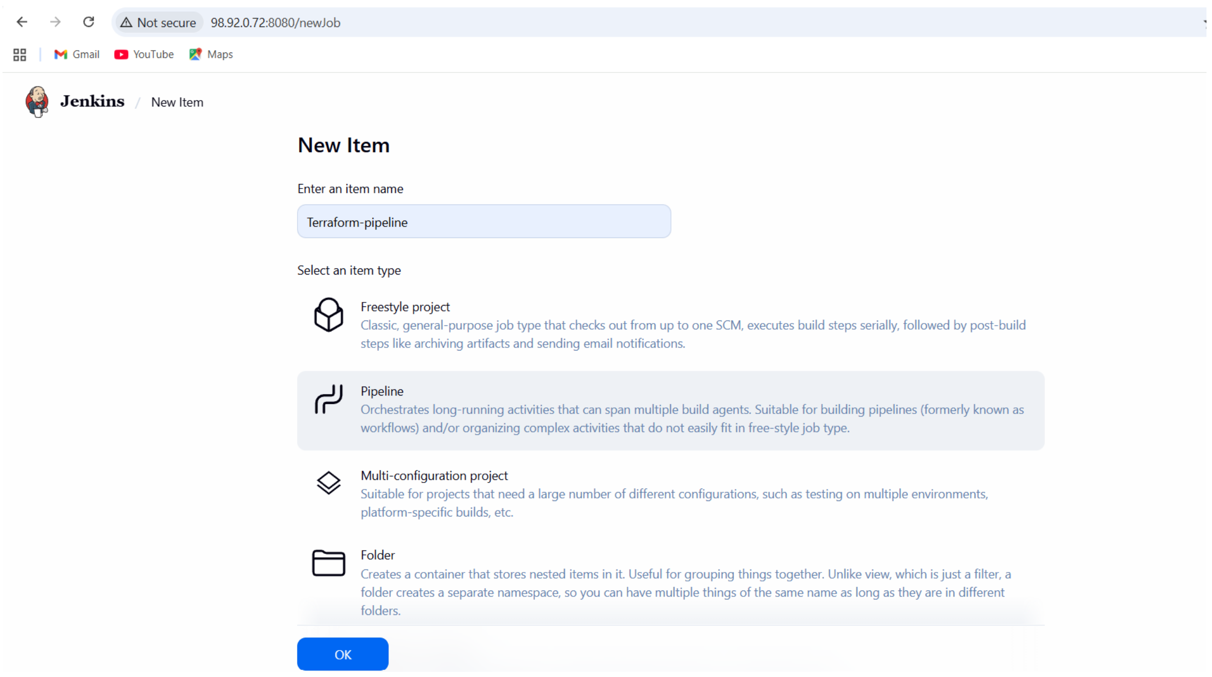
jenkinsadmin ▾ Create new IAM role

Cancel Update IAM role



```
[root@ip-172-31-77-84 deployment]# cat main.tf
provider "aws" {
  region = "us-east-1"
}

data "aws_vpc" "selected" {
  filter {
    name   = "tag:Name"
    values = ["default"]
  }
}

data "aws_subnets" "selected" {
  filter {
    name   = "vpc-id"
    values = [data.aws_vpc.selected.id]
  }
}

data "aws_ami" "amazon_linux_2" {
  most_recent = true
  filter {
    name   = "name"
    values = ["amzn2-ami-hvm-*-x86_64-gp2"]
  }

  filter {
    name   = "virtualization-type"
    values = ["hvm"]
  }
```

Create a item with name terraform-pipeline and select pipeline as type.


Not secure 98.92.0.72:8080/newJob

Gmail YouTube Maps

Jenkins / New Item

New Item

Enter an item name

Terraform-pipeline

Select an item type

Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

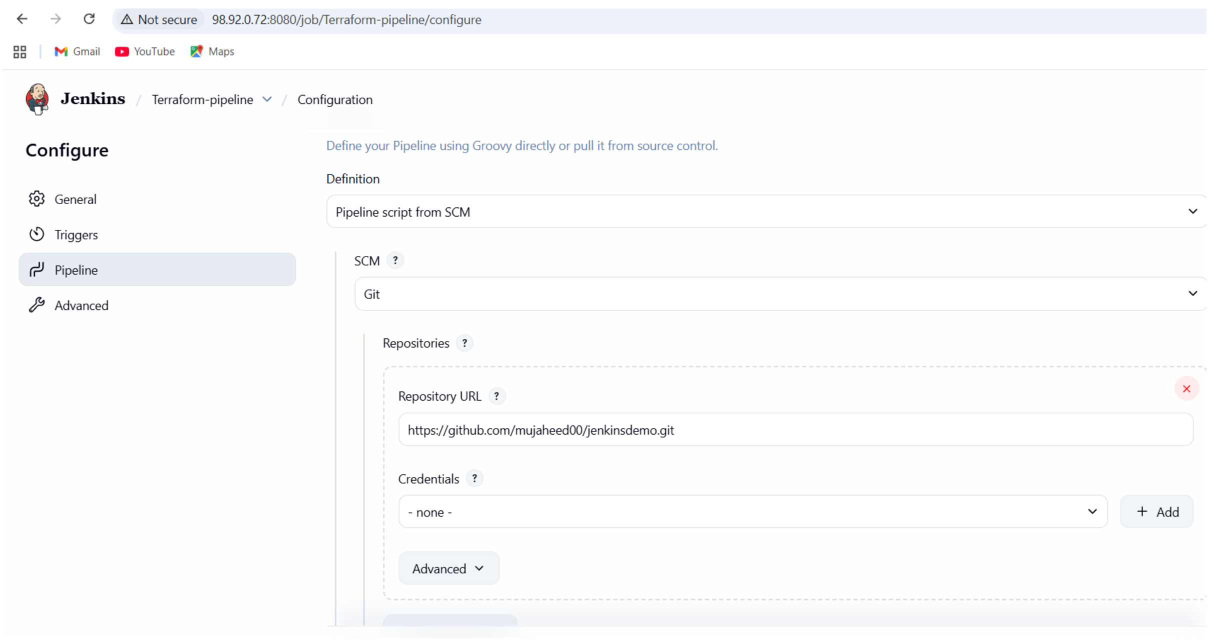
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

OK

Select as pipeline script from SCM and select git option enter your repo url and branch as main.


Not secure 98.92.0.72:8080/job/Terraform-pipeline/configure

Gmail YouTube Maps

Jenkins / Terraform-pipeline / Configuration

Configure

General
Triggers
Pipeline
Advanced

Define your Pipeline using Groovy directly or pull it from source control.

Definition: Pipeline script from SCM

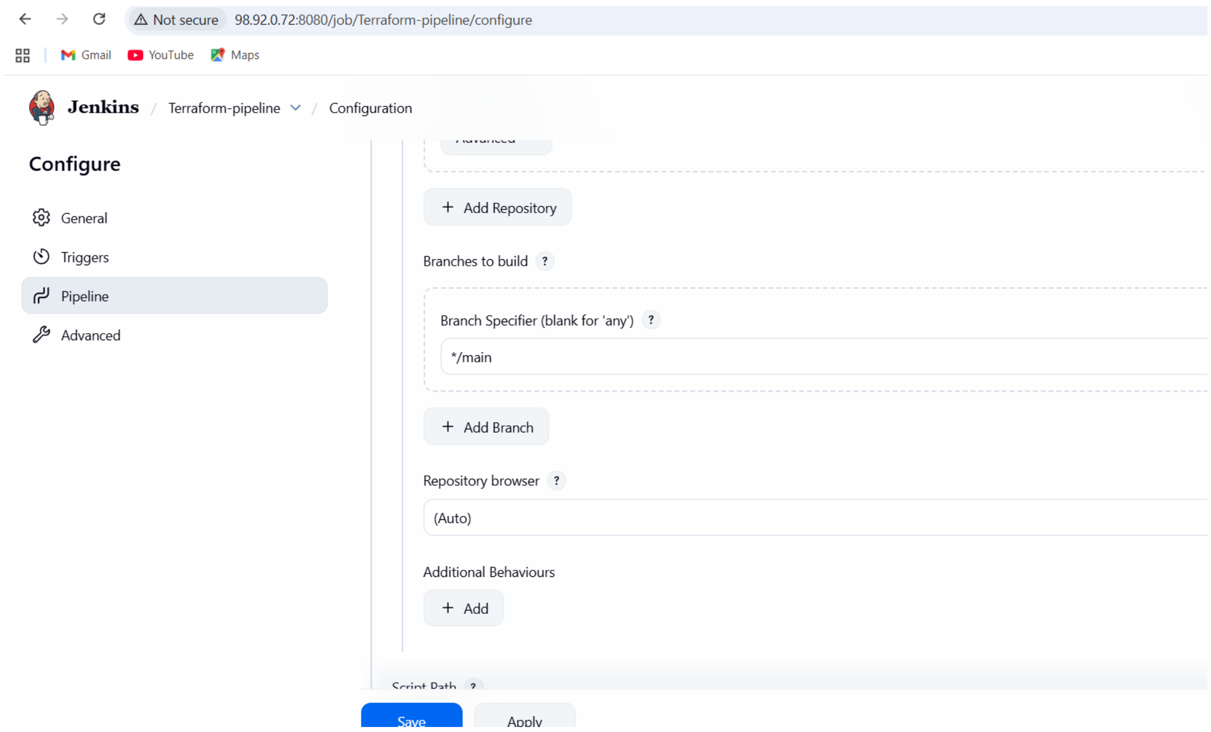
SCM: Git

Repositories

Repository URL: https://github.com/mujaheed00/jenkinsdemo.git

Credentials: - none - + Add

Advanced



98.92.0.72:8080/job/Terraform-pipeline/configure

Jenkins / Terraform-pipeline / Configuration

Configure

Branches to build

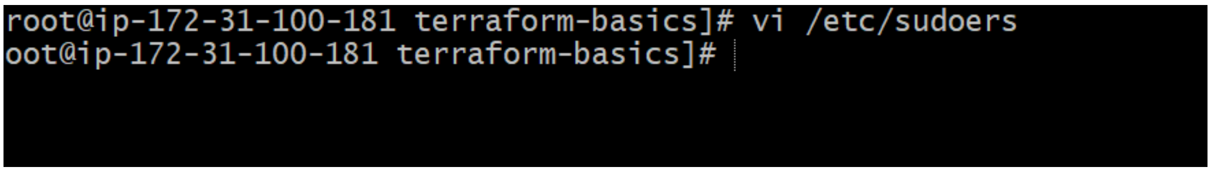
Branch Specifier (blank for 'any'): */main

Repository browser: (Auto)

Additional Behaviours

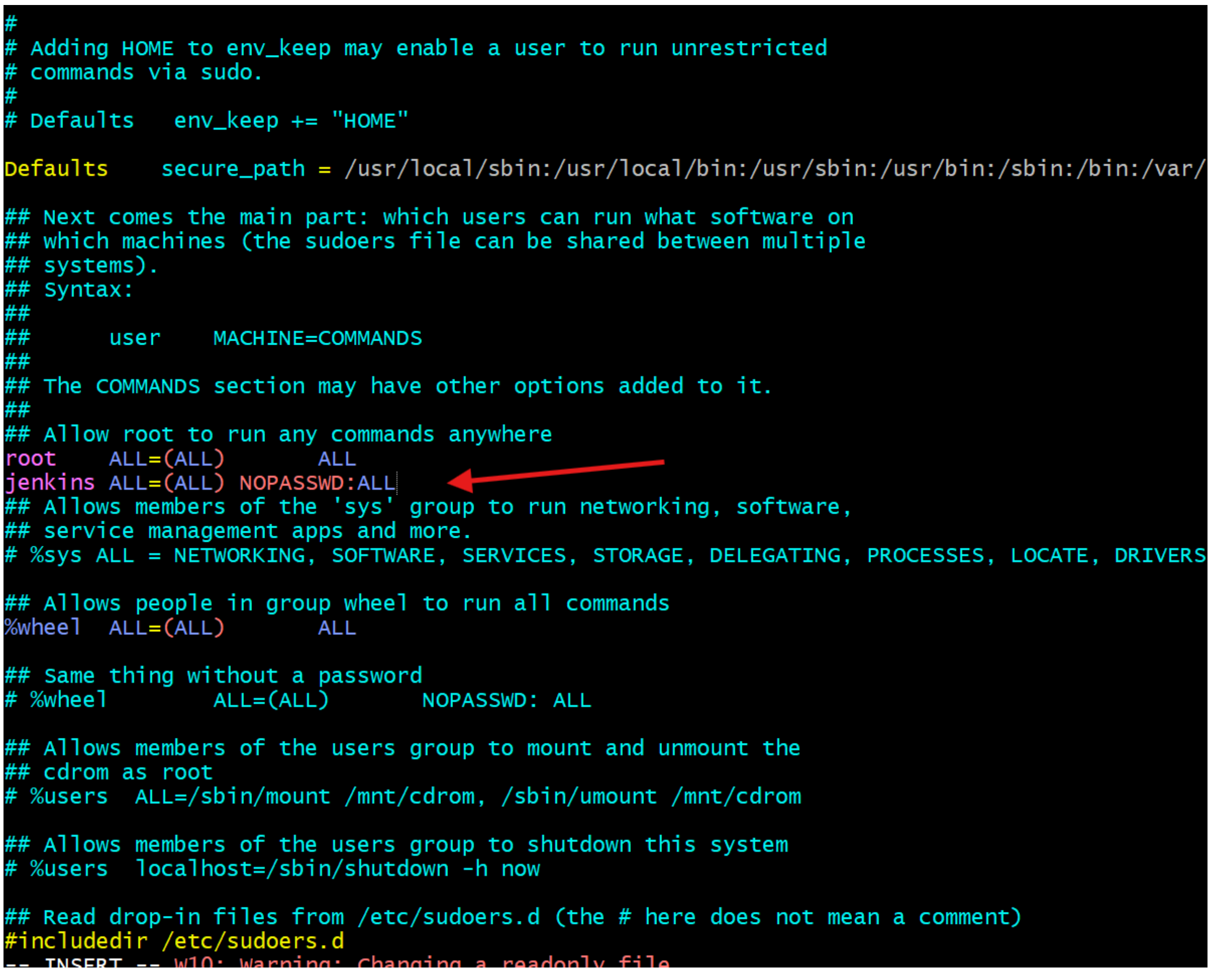
To to your cloned repository and edit sudoers file

- vi /etc/sudoers



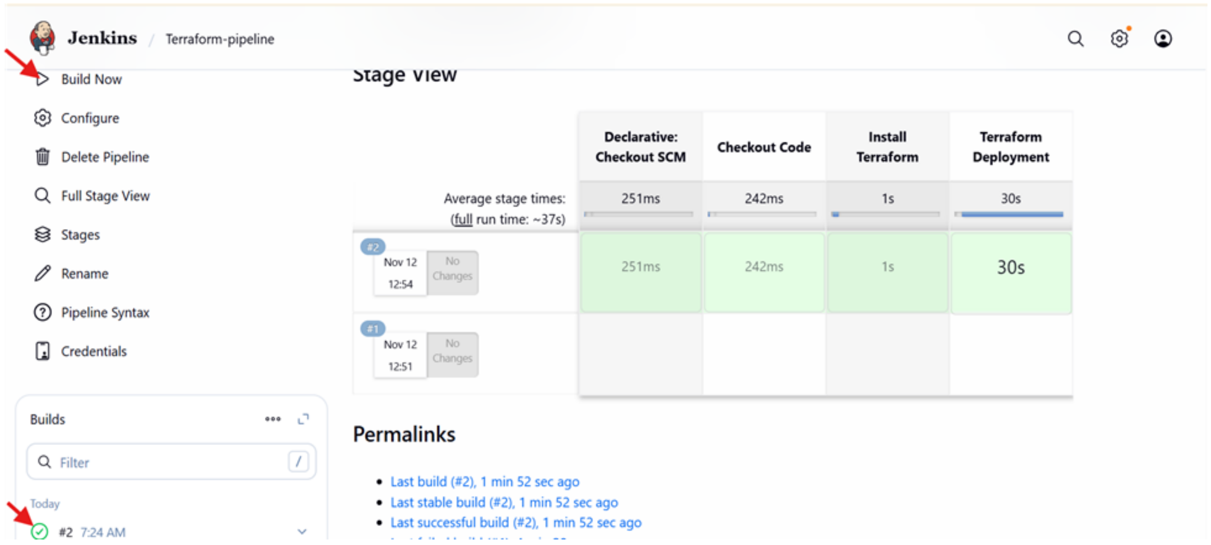
```
root@ip-172-31-100-181 terraform-basics]# vi /etc/sudoers
oot@ip-172-31-100-181 terraform-basics]#
```

Type this line.



```
#
# Adding HOME to env_keep may enable a user to run unrestricted
# commands via sudo.
#
# Defaults   env_keep += "HOME"

Defaults    secure_path = /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/var/

## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##      user    MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)       ALL
jenkins ALL=(ALL) NOPASSWD:ALL
## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS

## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)       ALL

## Same thing without a password
# %wheel        ALL=(ALL)       NOPASSWD: ALL

## Allows members of the users group to mount and unmount the
## cdrom as root
# %users  ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

## Allows members of the users group to shutdown this system
# %users  localhost=/sbin/shutdown -h now

## Read drop-in files from /etc/sudoers.d (the # here does not mean a comment)
#includedir /etc/sudoers.d
-- INSERT -- W10: Warning: Changing a readonly file
```

Go to job click on build now



Jenkins / Terraform-pipeline

- Build Now
- Configure
- Delete Pipeline
- Full Stage View
- Stages
- Rename
- Pipeline Syntax
- Credentials

Builds

Filter

Today

#2 7:24 AM

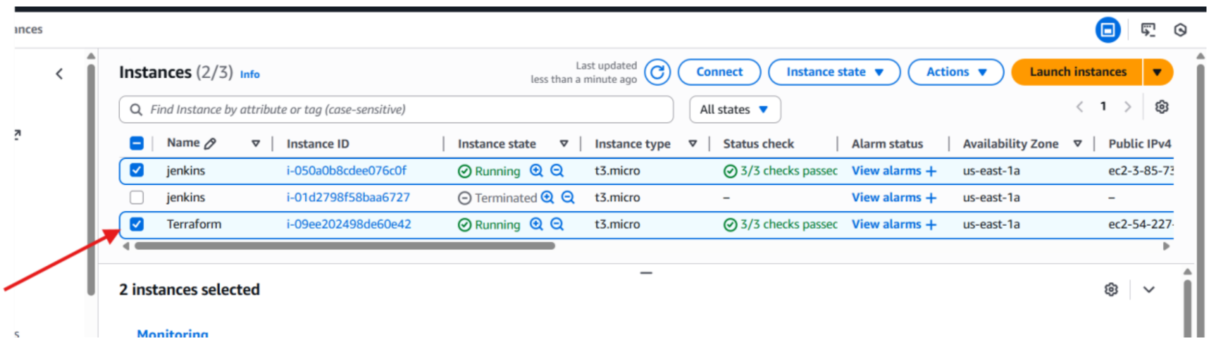
Stage View

	Declarative: Checkout SCM	Checkout Code	Install Terraform	Terraform Deployment
Average stage times: (full run time: ~37s)	251ms	242ms	1s	30s
#2 Nov 12 12:54 No Changes	251ms	242ms	1s	30s
#1 Nov 12 12:51 No Changes				

Permalinks

- Last build (#2), 1 min 52 sec ago
- Last stable build (#2), 1 min 52 sec ago
- Last successful build (#2), 1 min 52 sec ago

It will automatically create an instance with the name terraform.



	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
✓	jenkins	i-050a0b8cdee076c0f	Running	t3.micro	3/3 checks passec	View alarms +	us-east-1a	ec2-3-85-73
	jenkins	i-01d2798f58baa6727	Terminated	t3.micro	–	View alarms +	us-east-1a	–
✓	Terraform	i-09ee202498de60e42	Running	t3.micro	3/3 checks passec	View alarms +	us-east-1a	ec2-54-227-