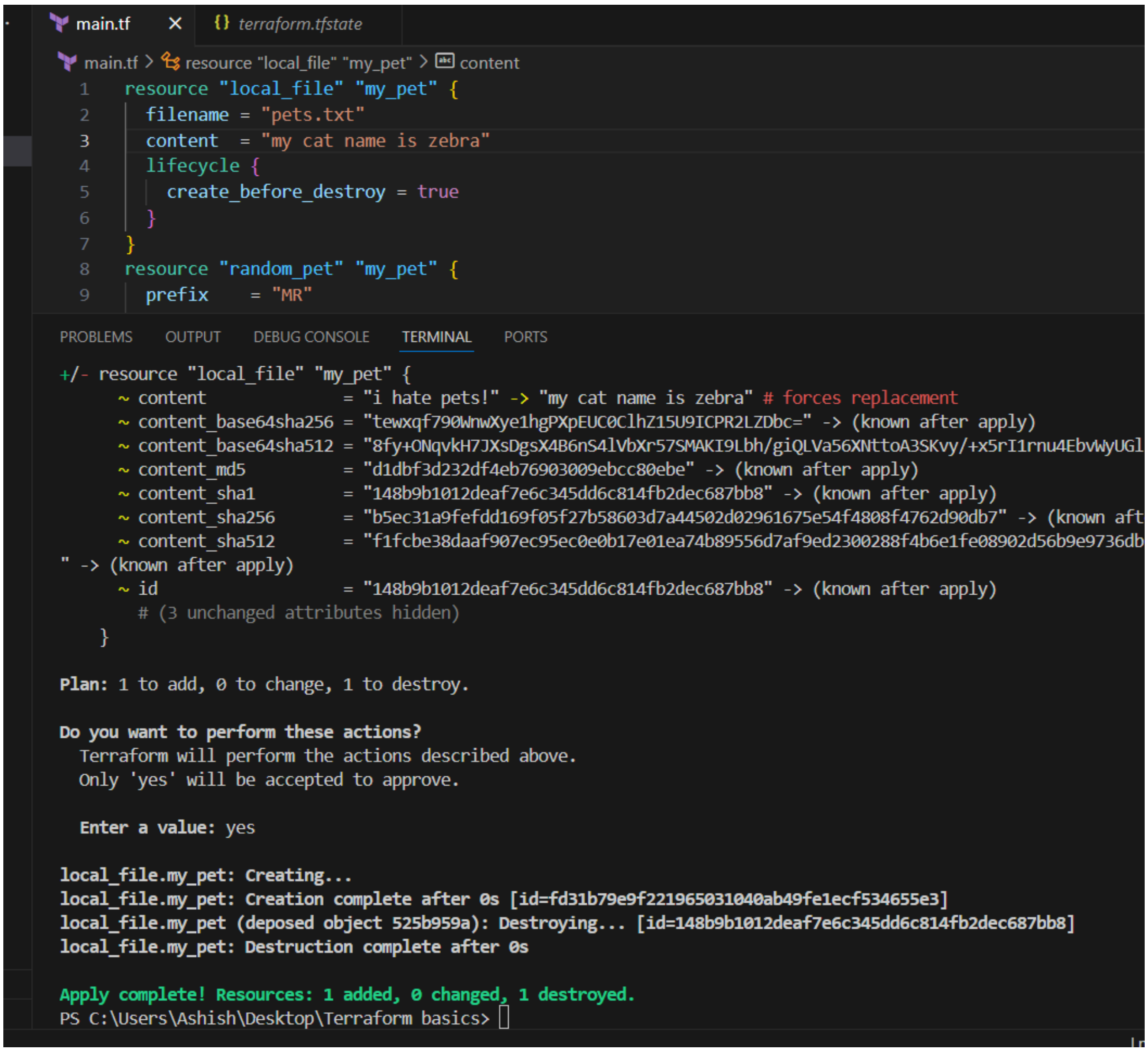
1. Watch the Terraform-02 video.

2. Execute all the templates shown in the video.

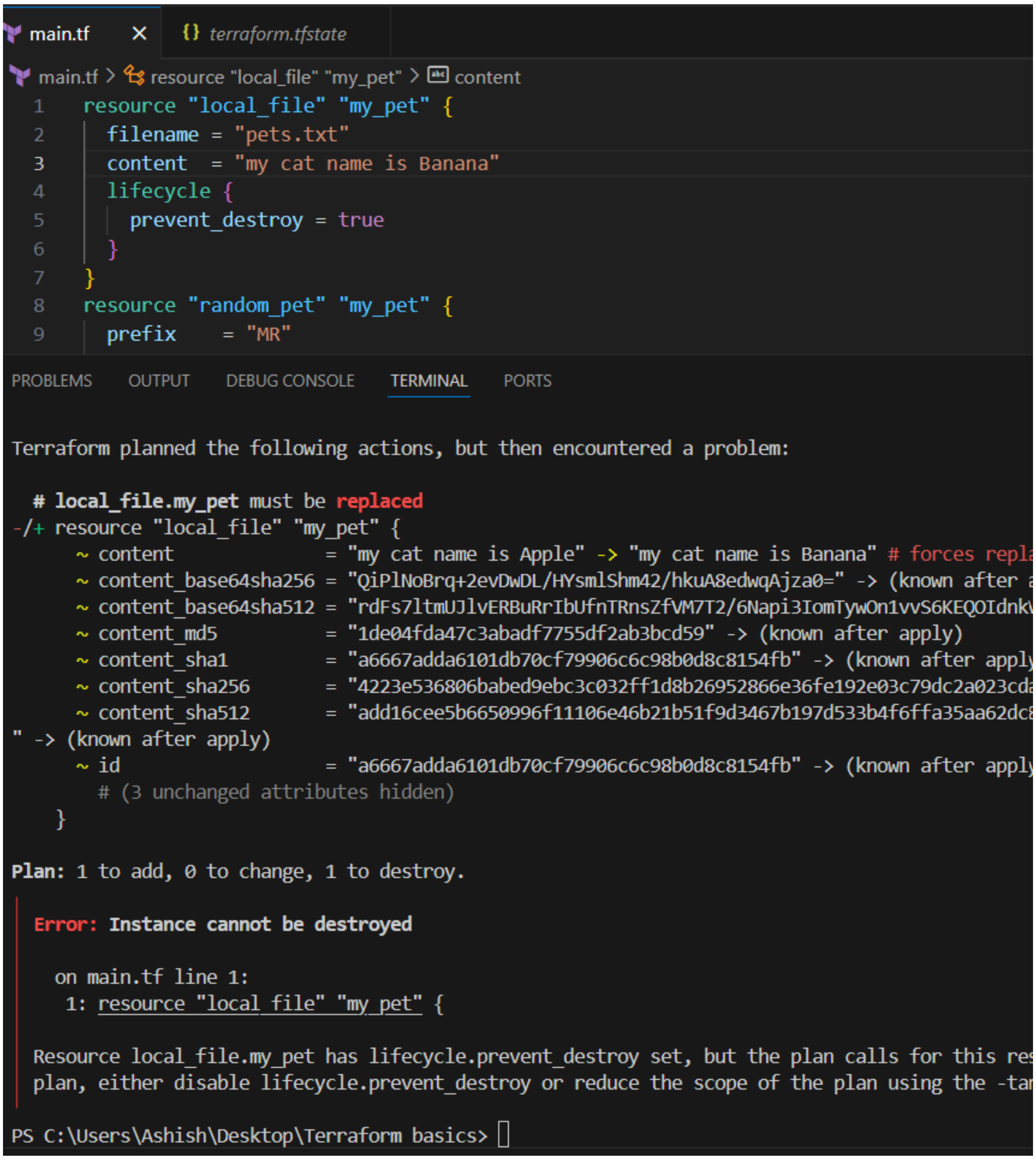
- Using lifecycle rule create_before_destroy



```hcl
resource "local_file" "my_pet" {
  filename = "pets.txt"
  content  = "my cat name is zebra"
  lifecycle {
    create_before_destroy = true
  }
}
resource "random_pet" "my_pet" {
  prefix    = "MR"
```

```
+/- resource "local_file" "my_pet" {
      ~ content              = "i hate pets!" -> "my cat name is zebra" # forces replacement
      ~ content_base64sha256 = "tewxqf790WnwXye1hgPXpEUC0ClhZ15U9ICPR2LZDbc=" -> (known after apply)
      ~ content_base64sha512 = "8fy+ONqvkH7JXsDgsX4B6nS4lVbXr57SMAKI9Lbh/giQLVa56XNttoA3SKvy/+x5rI1rnu4EbvWyUGl
      ~ content_md5          = "d1dbf3d232df4eb76903009ebcc80ebe" -> (known after apply)
      ~ content_sha1         = "148b9b1012deaf7e6c345dd6c814fb2dec687bb8" -> (known after apply)
      ~ content_sha256       = "b5ec31a9fefdd169f05f27b58603d7a44502d02961675e54f4808f4762d90db7" -> (known aft
      ~ content_sha512       = "f1fcbe38daaf907ec95ec0e0b17e01ea74b89556d7af9ed2300288f4b6e1fe08902d56b9e9736db
" -> (known after apply)
      ~ id                   = "148b9b1012deaf7e6c345dd6c814fb2dec687bb8" -> (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: Creating...
local_file.my_pet: Creation complete after 0s [id=fd31b79e9f221965031040ab49fe1ecf534655e3]
local_file.my_pet (deposed object 525b959a): Destroying... [id=148b9b1012deaf7e6c345dd6c814fb2dec687bb8]
local_file.my_pet: Destruction complete after 0s

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

- Using lifecycle rule prevent_destory

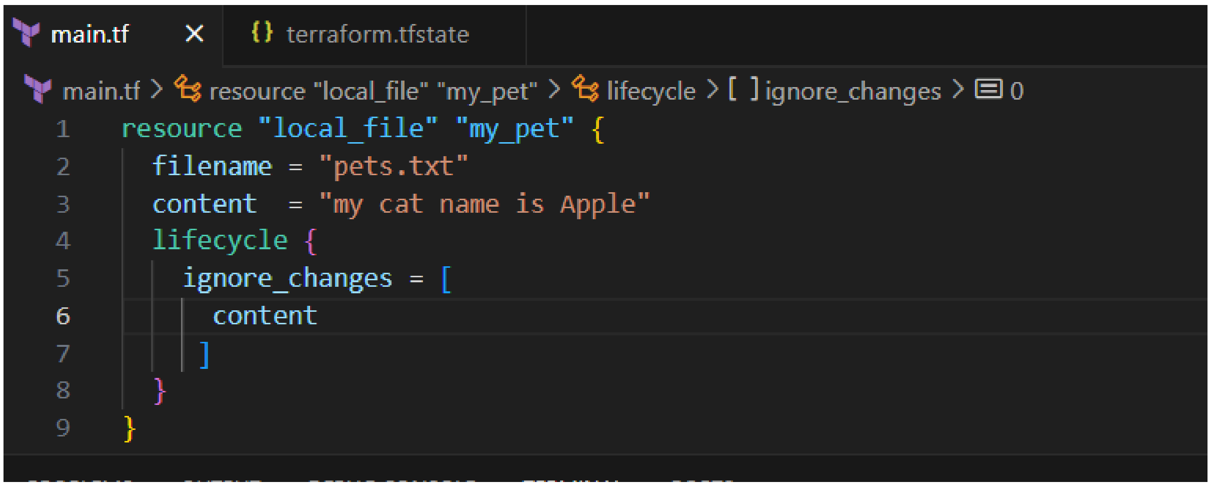


```
resource "local_file" "my_pet" {
  filename = "pets.txt"
  content  = "my cat name is Banana"
  lifecycle {
    prevent_destroy = true
  }
}
resource "random_pet" "my_pet" {
  prefix     = "MR"
```

```
Terraform planned the following actions, but then encountered a problem:

  # local_file.my_pet must be replaced
-/+ resource "local_file" "my_pet" {
      ~ content              = "my cat name is Apple" -> "my cat name is Banana" # forces repla
      ~ content_base64sha256 = "QiPlNoBrq+2evDwDL/HYsmlShm42/hkuA8edwqAjza0=" -> (known after a
      ~ content_base64sha512 = "rdFs7ltmUJlvERBuRrIbUfnTRnsZfVM7T2/6Napi3IomTywOn1vvS6KEQOIdnkV
      ~ content_md5          = "1de04fda47c3abadf7755df2ab3bcd59" -> (known after apply)
      ~ content_sha1         = "a6667adda6101db70cf79906c6c98b0d8c8154fb" -> (known after apply
      ~ content_sha256       = "4223e536806babed9ebc3c032ff1d8b26952866e36fe192e03c79dc2a023cda
      ~ content_sha512       = "add16cee5b6650996f11106e46b21b51f9d3467b197d533b4f6ffa35aa62dc8
" -> (known after apply)
      ~ id                   = "a6667adda6101db70cf79906c6c98b0d8c8154fb" -> (known after apply
        # (3 unchanged attributes hidden)
    }

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

  Error: Instance cannot be destroyed

    on main.tf line 1:
     1: resource "local_file" "my_pet" {

  Resource local_file.my_pet has lifecycle.prevent_destroy set, but the plan calls for this res
  plan, either disable lifecycle.prevent_destroy or reduce the scope of the plan using the -tar

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

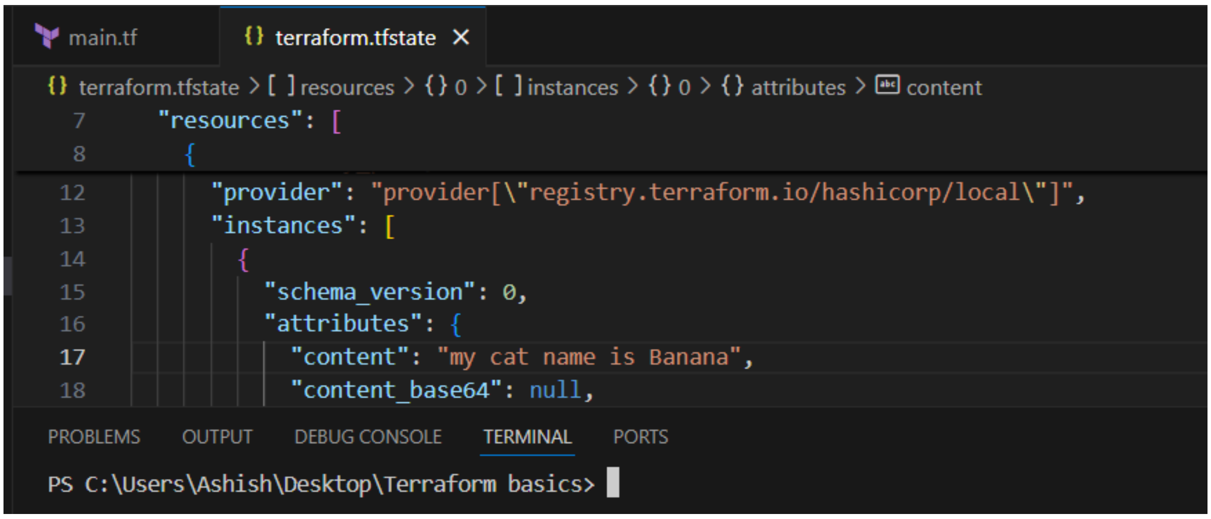
- Using lifecycle rule ignore_changes

In main.tf


main.tf
terraform.tfstate
main.tf > resource "local_file" "my_pet" > lifecycle > [ ] ignore_changes > 0

```
resource "local_file" "my_pet" {
  filename = "pets.txt"
  content  = "my cat name is Apple"
  lifecycle {
    ignore_changes = [
      content
    ]
  }
}
```

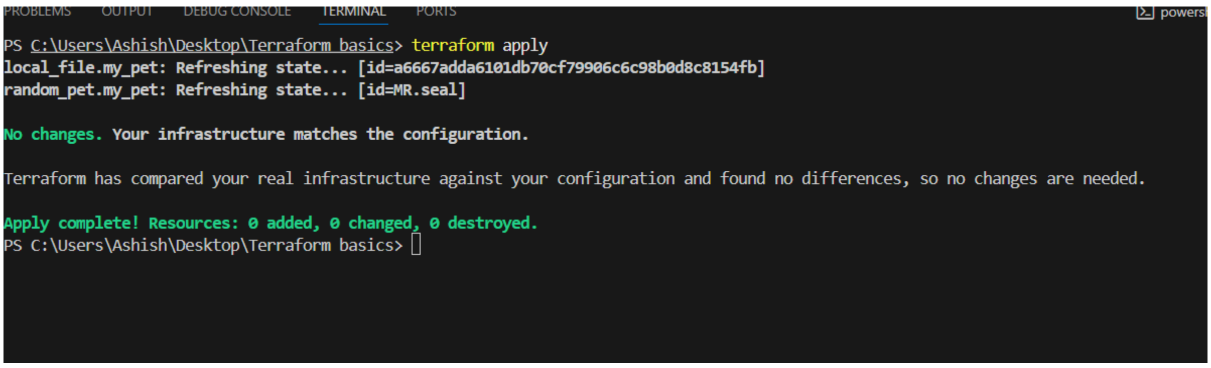
In .tf state


main.tf
terraform.tfstate
terraform.tfstate > [ ] resources > { } 0 > [ ] instances > { } 0 > { } attributes > content
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS

```
    "resources": [
      {
        "provider": "provider[\"registry.terraform.io/hashicorp/local\"]",
        "instances": [
          {
            "schema_version": 0,
            "attributes": {
              "content": "my cat name is Banana",
              "content_base64": null,
```

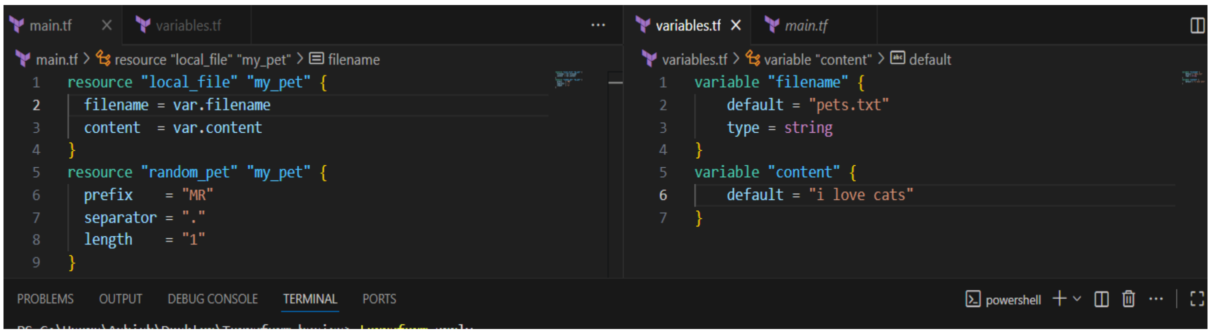
```
PS C:\Users\Ashish\Desktop\Terraform basics>
```

Changes are ignored



```
PS C:\Users\Ashish\Desktop\Terraform basics> terraform apply
local_file.my_pet: Refreshing state... [id=a6667adda6101db70cf79906c6c98b0d8c8154fb]
random_pet.my_pet: Refreshing state... [id=MR.seal]

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\Ashish\Desktop\Terraform basics>
```

- Create a file named as variable.tf

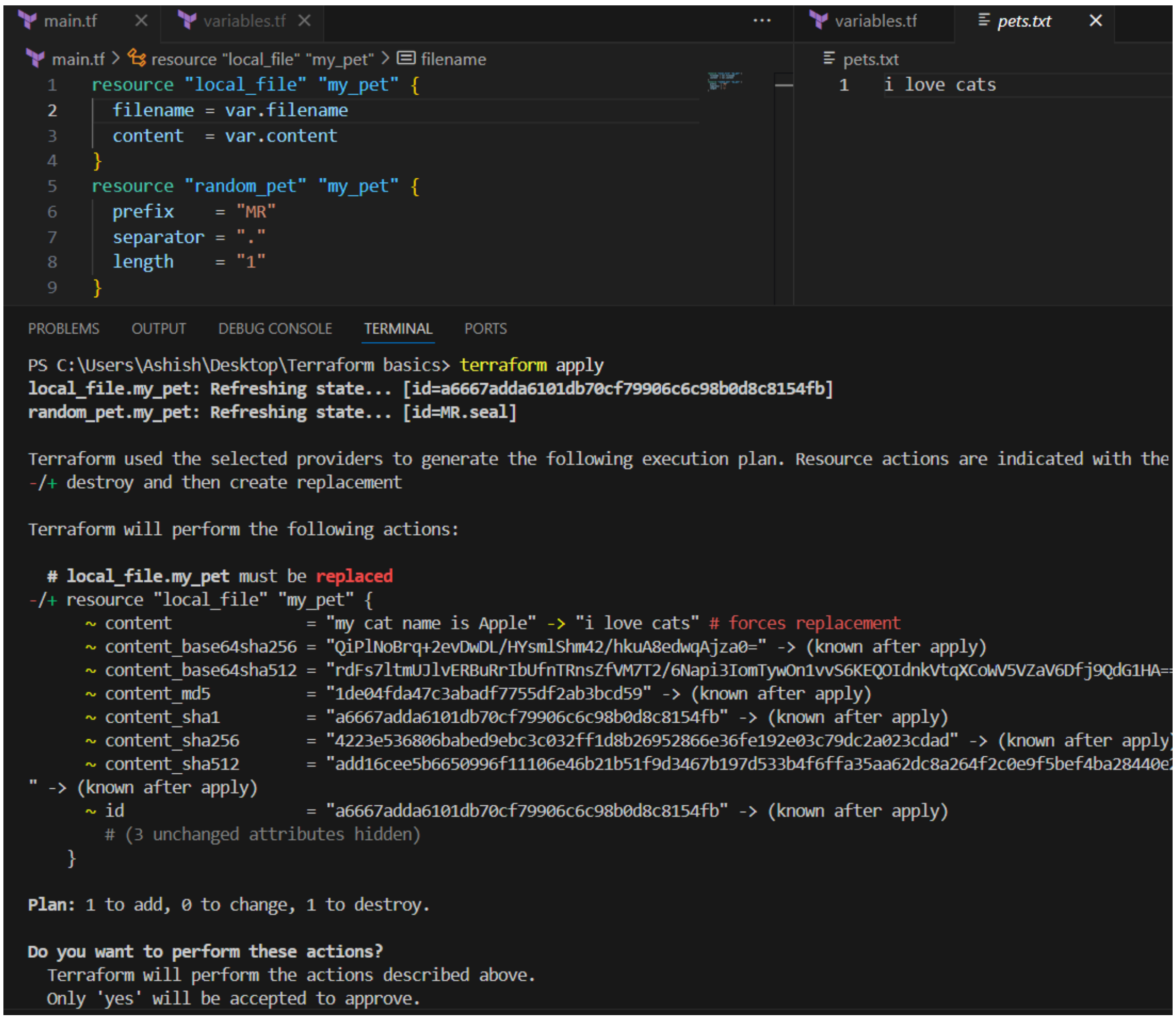


main.tf > resource "local_file" "my_pet" > filename

```
resource "local_file" "my_pet" {
  filename = var.filename
  content  = var.content
}
resource "random_pet" "my_pet" {
  prefix    = "MR"
  separator = "."
  length    = "1"
}
```

variables.tf > variable "content" > default

```
variable "filename" {
    default = "pets.txt"
    type = string
}
variable "content" {
    default = "i love cats"
}
```



main.tf > resource "local_file" "my_pet" > filename

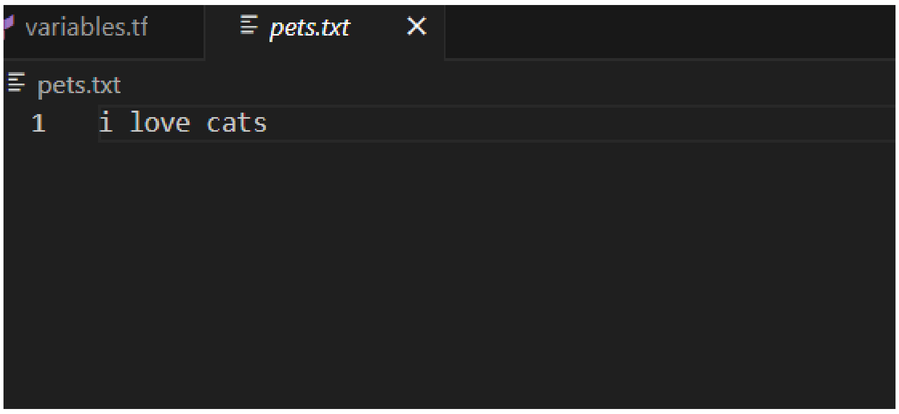
```
resource "local_file" "my_pet" {
  filename = var.filename
  content  = var.content
}
resource "random_pet" "my_pet" {
  prefix    = "MR"
  separator = "."
  length    = "1"
}
```

pets.txt

```
i love cats
```

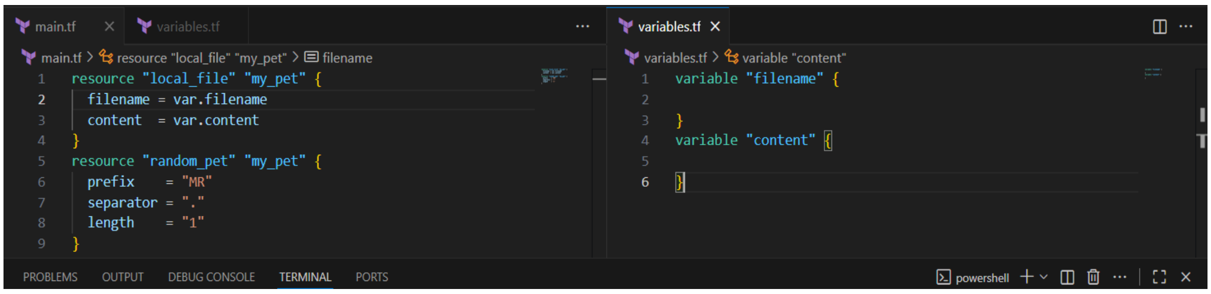
```
PS C:\Users\Ashish\Desktop\Terraform basics> terraform apply
local_file.my_pet: Refreshing state... [id=a6667adda6101db70cf79906c6c98b0d8c8154fb]
random_pet.my_pet: Refreshing state... [id=MR.seal]

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

Terraform will perform the following actions:

  # local_file.my_pet must be replaced
-/+ resource "local_file" "my_pet" {
      ~ content              = "my cat name is Apple" -> "i love cats" # forces replacement
      ~ content_base64sha256 = "QiPlNoBrq+2evDwDL/HYsmlShm42/hkuA8edwqAjza0=" -> (known after apply)
      ~ content_base64sha512 = "rdFs7ltmUJlvERBuRrIbUfnTRnsZfVM7T2/6Napi3IomTywOn1vvS6KEQOIdnkVtqXCoWV5VZaV6Dfj9QdG1HA==
      ~ content_md5          = "1de04fda47c3abadf7755df2ab3bcd59" -> (known after apply)
      ~ content_sha1         = "a6667adda6101db70cf79906c6c98b0d8c8154fb" -> (known after apply)
      ~ content_sha256       = "4223e536806babed9ebc3c032ff1d8b26952866e36fe192e03c79dc2a023cdad" -> (known after apply)
      ~ content_sha512       = "add16cee5b6650996f11106e46b21b51f9d3467b197d533b4f6ffa35aa62dc8a264f2c0e9f5bef4ba28440e
" -> (known after apply)
      ~ id                   = "a6667adda6101db70cf79906c6c98b0d8c8154fb" -> (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.
```

If you see the pets.txt this will be executed.



```
pets.txt
i love cats
```

- Another type variable



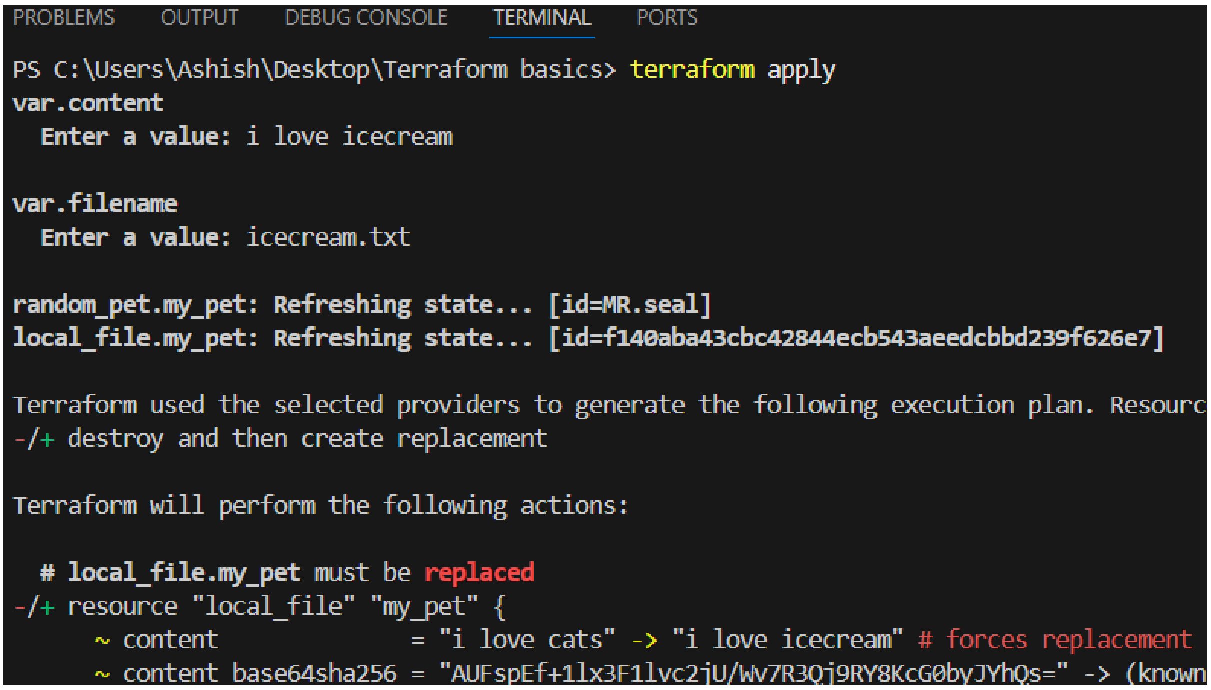
main.tf:

```
resource "local_file" "my_pet" {
  filename = var.filename
  content  = var.content
}
resource "random_pet" "my_pet" {
  prefix    = "MR"
  separator = "."
  length    = "1"
}
```

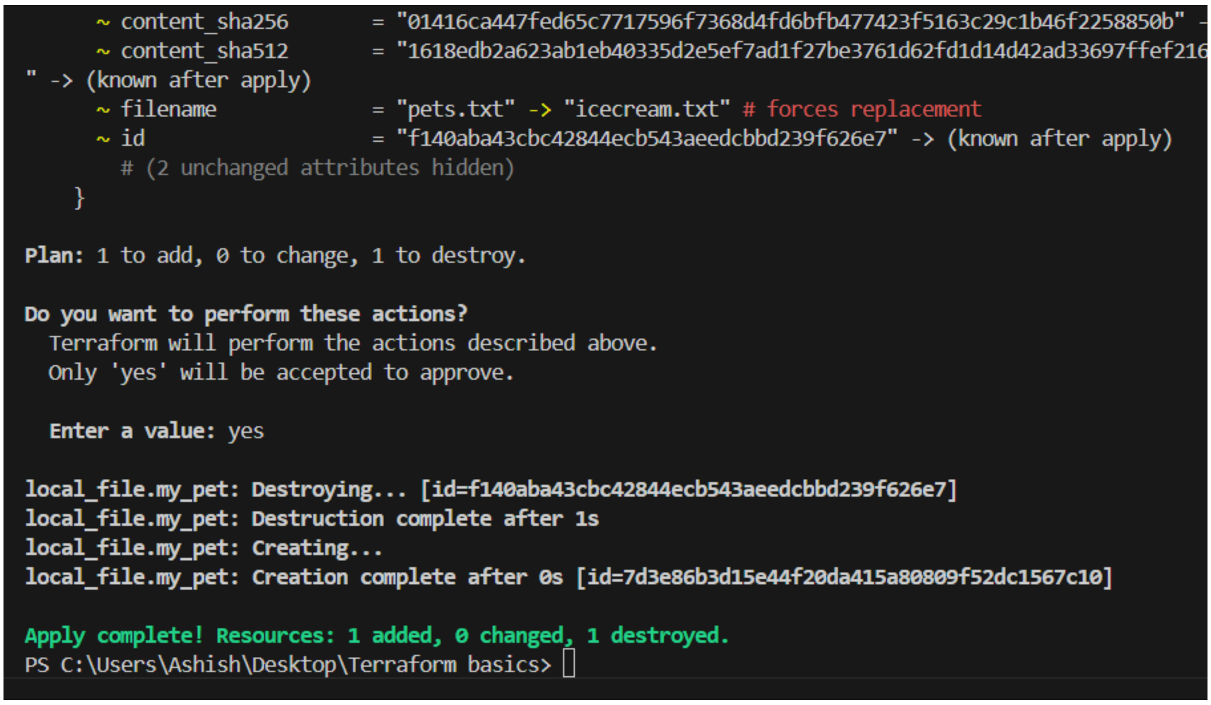
variables.tf:

```
variable "filename" {

}
variable "content" {

}
```

We need to give our content and file name in terminal

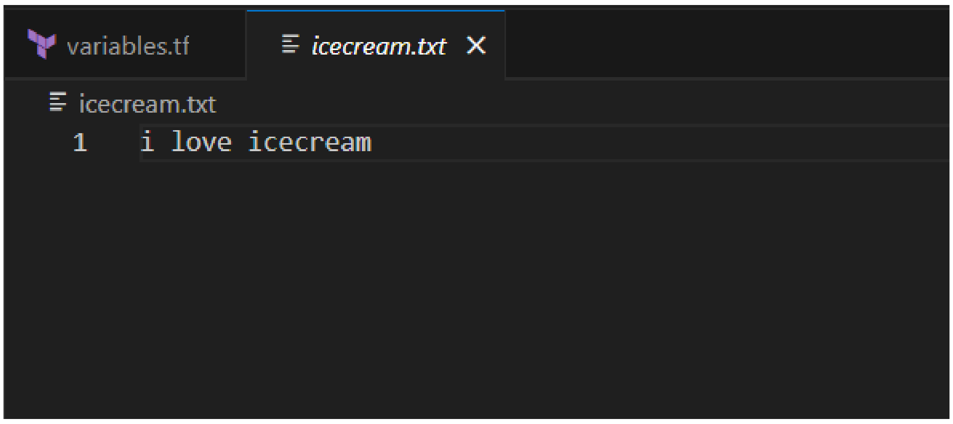


```
PS C:\Users\Ashish\Desktop\Terraform basics> terraform apply
var.content
  Enter a value: i love icecream

var.filename
  Enter a value: icecream.txt

random_pet.my_pet: Refreshing state... [id=MR.seal]
local_file.my_pet: Refreshing state... [id=f140aba43cbc42844ecb543aeedcbbd239f626e7]

Terraform used the selected providers to generate the following execution plan. Resourc
-/+ 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 cats" -> "i love icecream" # forces replacement
      ~ content_base64sha256 = "AUFspEf+1lx3F1lvc2jU/Wv7R3Qj9RY8KcG0byJYhQs=" -> (known
```



```
      ~ content_sha256       = "01416ca447fed65c7717596f7368d4fd6bfb477423f5163c29c1b46f2258850b" -
      ~ content_sha512       = "1618edb2a623ab1eb40335d2e5ef7ad1f27be3761d62fd1d14d42ad33697ffef216
" -> (known after apply)
      ~ filename             = "pets.txt" -> "icecream.txt" # forces replacement
      ~ id                   = "f140aba43cbc42844ecb543aeedcbbd239f626e7" -> (known after apply)
        # (2 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=f140aba43cbc42844ecb543aeedcbbd239f626e7]
local_file.my_pet: Destruction complete after 1s
local_file.my_pet: Creating...
local_file.my_pet: Creation complete after 0s [id=7d3e86b3d15e44f20da415a80809f52dc1567c10]

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

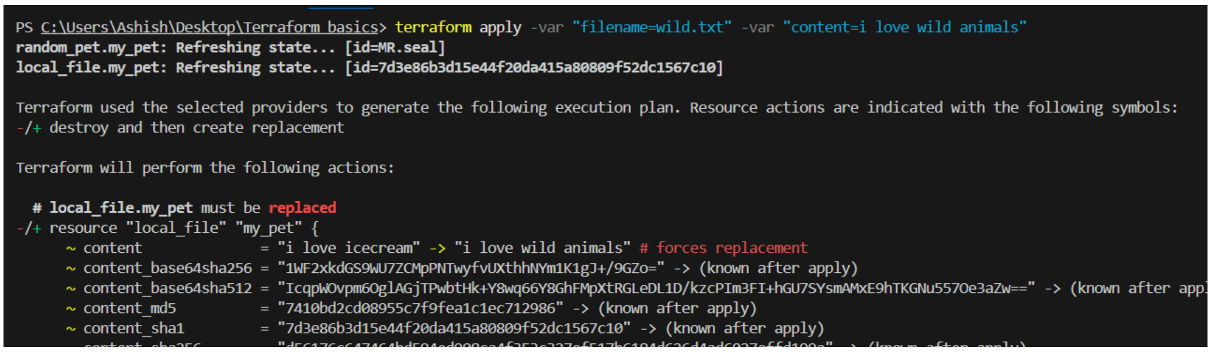
If I open the file content has been created.



```
icecream.txt
1    i love icecream
```

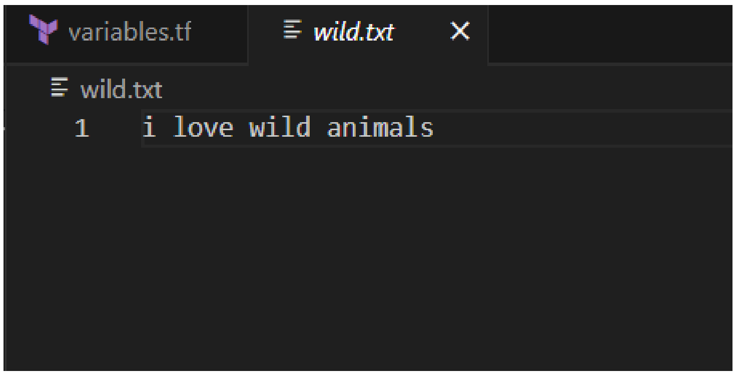
- Another type for variables

terraform apply -var “filename=wild.txt” -var ”content=I love wild animals”



```
PS C:\Users\Ashish\Desktop\Terraform basics> terraform apply -var "filename=wild.txt" -var "content=i love wild animals"
random_pet.my_pet: Refreshing state... [id=MR.seal]
local_file.my_pet: Refreshing state... [id=7d3e86b3d15e44f20da415a80809f52dc1567c10]

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 icecream" -> "i love wild animals" # forces replacement
      ~ content_base64sha256 = "1WF2xkdGS9WU7ZCMpPNTwyfvUXthhNYm1K1gJ+/9GZo=" -> (known after apply)
      ~ content_base64sha512 = "IcqpWOvpm6OglAGjTPwbtHk+Y8wq66Y8GhFMpXtRGLeDL1D/kzcPIm3FI+hGU7SYsmAMxE9hTKGNu557Oe3aZw==" -> (known after appl
      ~ content_md5          = "7410bd2cd08955c7f9fea1c1ec712986" -> (known after apply)
      ~ content_sha1         = "7d3e86b3d15e44f20da415a80809f52dc1567c10" -> (known after apply)
```

If I open wild.txt

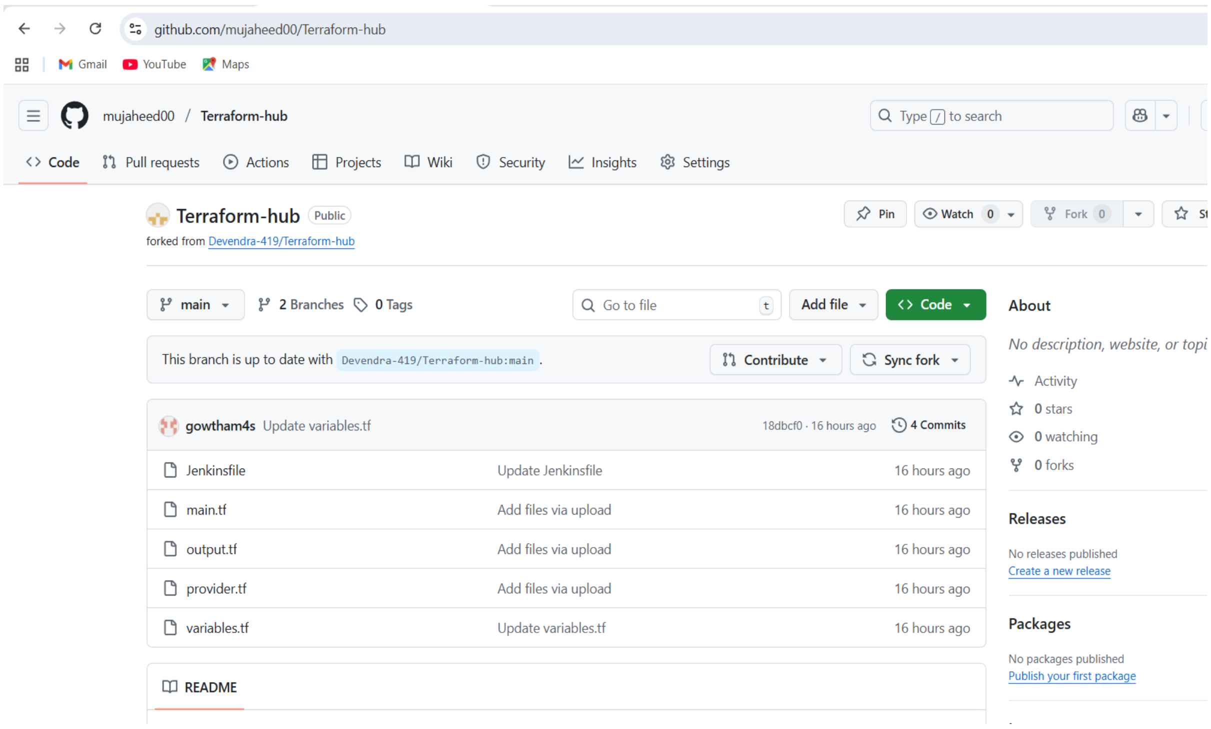


```
i love wild animals
```

3.Integrate Terraform in Jenkins using the Terraform plugin.

Keep all your files in an repository in github.

https://github.com/mujaheed00/Terraform-hub.git



forked from Devendra-419/Terraform-hub

This branch is up to date with `Devendra-419/Terraform-hub:main`.

gowtham4s Update variables.tf — 18dbcf0 · 16 hours ago — 4 Commits

Jenkinsfile	Update Jenkinsfile	16 hours ago
main.tf	Add files via upload	16 hours ago
output.tf	Add files via upload	16 hours ago
provider.tf	Add files via upload	16 hours ago
variables.tf	Update variables.tf	16 hours ago

About

No description, website, or topi

Activity
0 stars
0 watching
0 forks

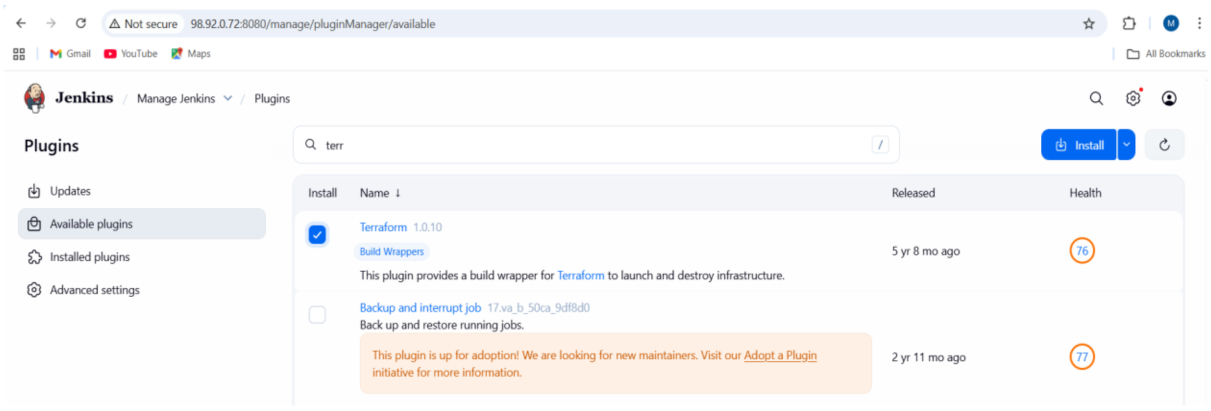
Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Go to manage Jenkins and click on plugins, install terraform plugin.


98.92.0.72:8080/manage/pluginManager/available

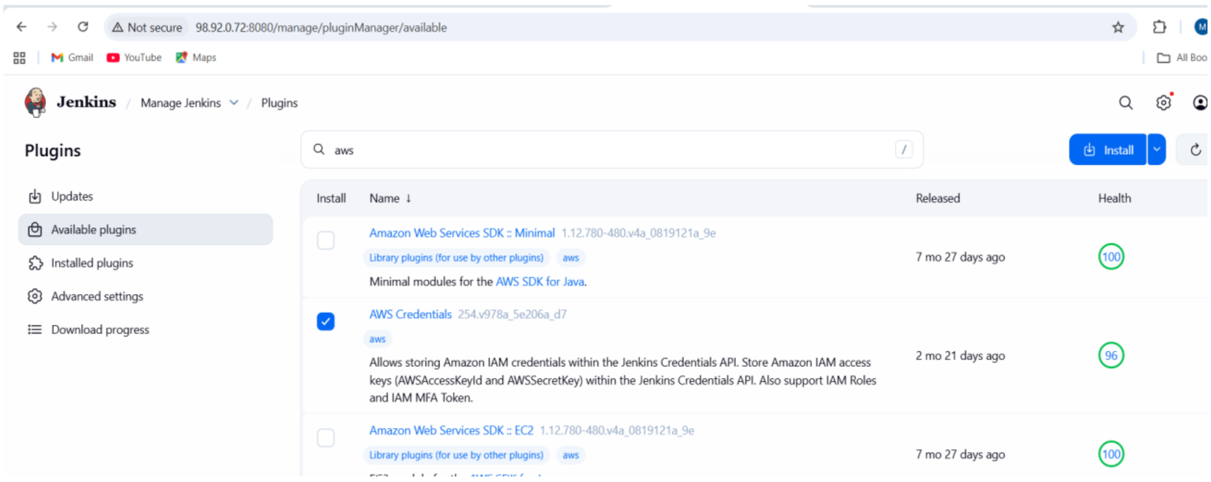
Jenkins / Manage Jenkins / Plugins

Plugins

terr

Install	Name ↓	Released	Health
✓	Terraform 1.0.10 · Build Wrappers · This plugin provides a build wrapper for Terraform to launch and destroy infrastructure.	5 yr 8 mo ago	76
	Backup and interrupt job 17.va_b_50ca_9df8d0 · Back up and restore running jobs. · This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.	2 yr 11 mo ago	77

Install another plugin called aws credentials.


98.92.0.72:8080/manage/pluginManager/available

Jenkins / Manage Jenkins / Plugins

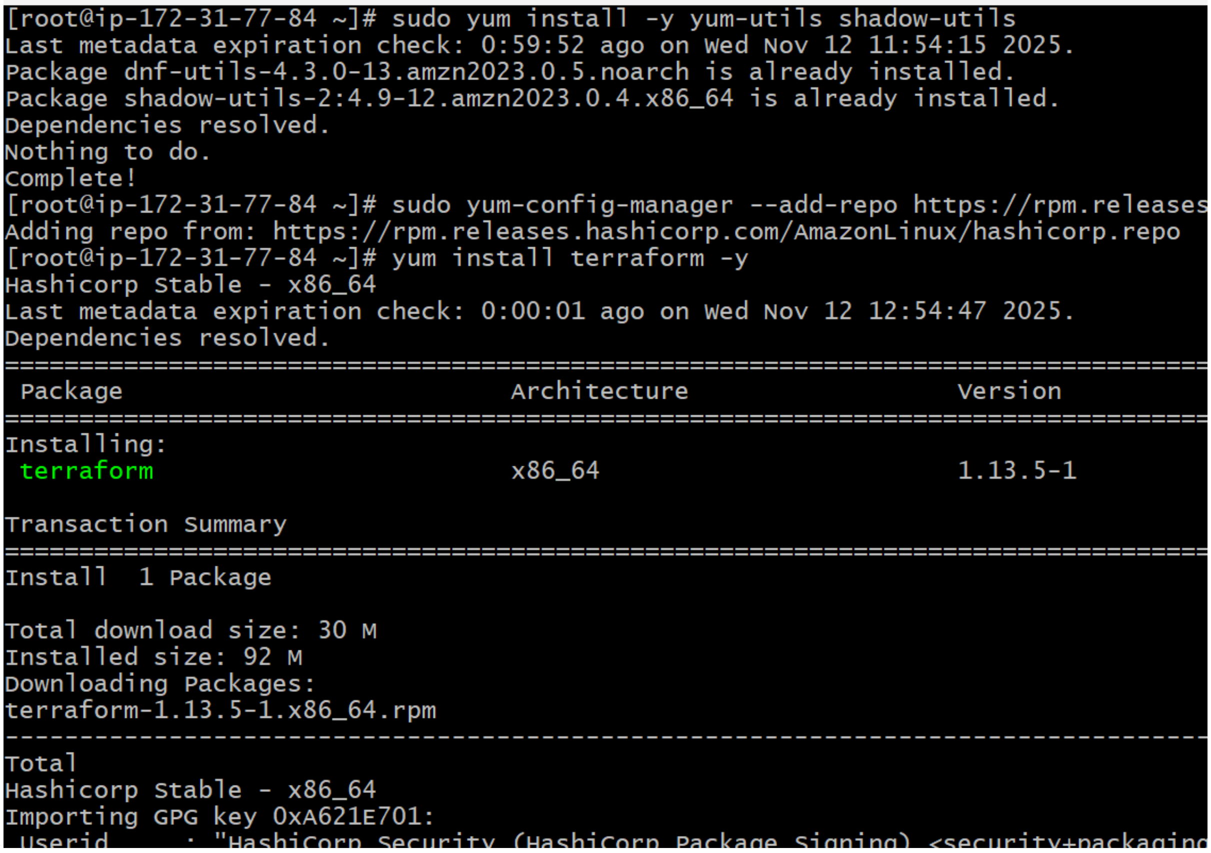
Plugins

aws

Install	Name ↓	Released	Health
	Amazon Web Services SDK :: Minimal 1.12.780-480.v4a_0819121a_9e · Library plugins (for use by other plugins) · aws · Minimal modules for the AWS SDK for Java.	7 mo 27 days ago	100
✓	AWS Credentials 254.v978a_5e206a_d7 · aws · Allows storing Amazon IAM credentials within the Jenkins Credentials API. Store Amazon IAM access keys (AWSAccessKeyId and AWSSecretKey) within the Jenkins Credentials API. Also support IAM Roles and IAM MFA Token.	2 mo 21 days ago	96
	Amazon Web Services SDK :: EC2 1.12.780-480.v4a_0819121a_9e · Library plugins (for use by other plugins) · aws	7 mo 27 days ago	100

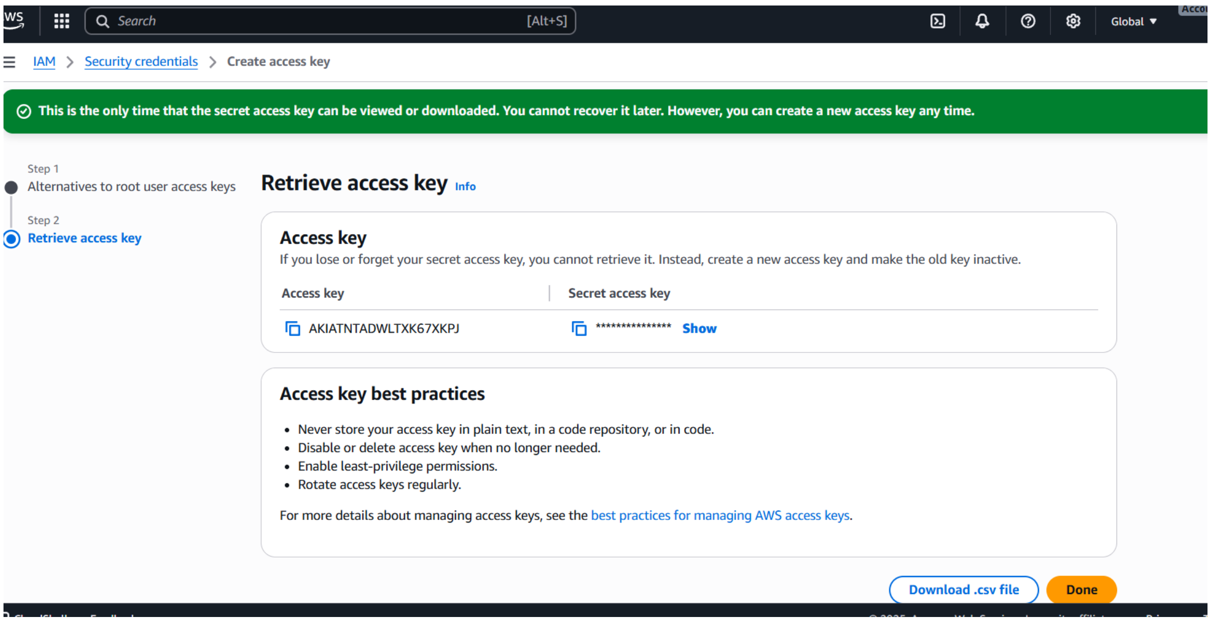
Go to Jenkins server and install terraform by using this commands.

- yum install -y yum-utils shadow-utils
- yum-config-manager --add-repo https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo
- yum install terraform



```
[root@ip-172-31-77-84 ~]# sudo yum install -y yum-utils shadow-utils
Last metadata expiration check: 0:59:52 ago on Wed Nov 12 11:54:15 2025.
Package dnf-utils-4.3.0-13.amzn2023.0.5.noarch is already installed.
Package shadow-utils-2:4.9-12.amzn2023.0.4.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-77-84 ~]# sudo yum-config-manager --add-repo https://rpm.releases
Adding repo from: https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo
[root@ip-172-31-77-84 ~]# yum install terraform -y
Hashicorp Stable - x86_64
Last metadata expiration check: 0:00:01 ago on Wed Nov 12 12:54:47 2025.
Dependencies resolved.
================================================================================
 Package                           Architecture                Version
================================================================================
Installing:
 terraform                         x86_64                      1.13.5-1

Transaction Summary
================================================================================
Install  1 Package

Total download size: 30 M
Installed size: 92 M
Downloading Packages:
terraform-1.13.5-1.x86_64.rpm
--------------------------------------------------------------------------------
Total
Hashicorp Stable - x86_64
Importing GPG key 0xA621E701:
 Userid     : "HashiCorp Security (HashiCorp Package Signing) <security+packaging
```

Create access key and secret key in aws credentials



IAM > Security credentials > Create access key

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

Step 1
Alternatives to root user access keys

Step 2
Retrieve access key

Retrieve access key Info

Access key

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

Access key	Secret access key
AKIATNTADWLTXK67XKPJ	*************** Show

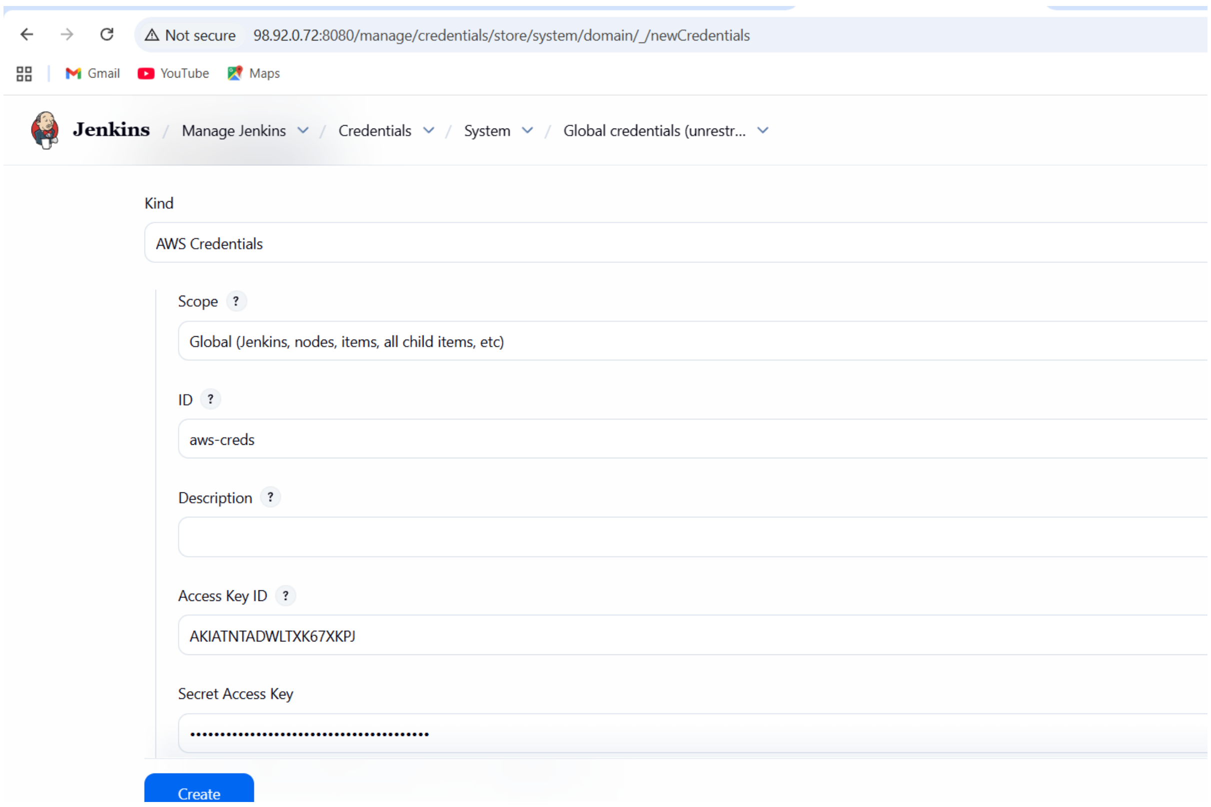
Access key best practices

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

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

Give the credentials in Jenkins

Go to manage Jenkins , credentials,global credentials select aws credentials and paste access key and secret key.



Not secure 98.92.0.72:8080/manage/credentials/store/system/domain/_/newCredentials

Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestr...

Kind

AWS Credentials

Scope

Global (Jenkins, nodes, items, all child items, etc)

ID

aws-creds

Description

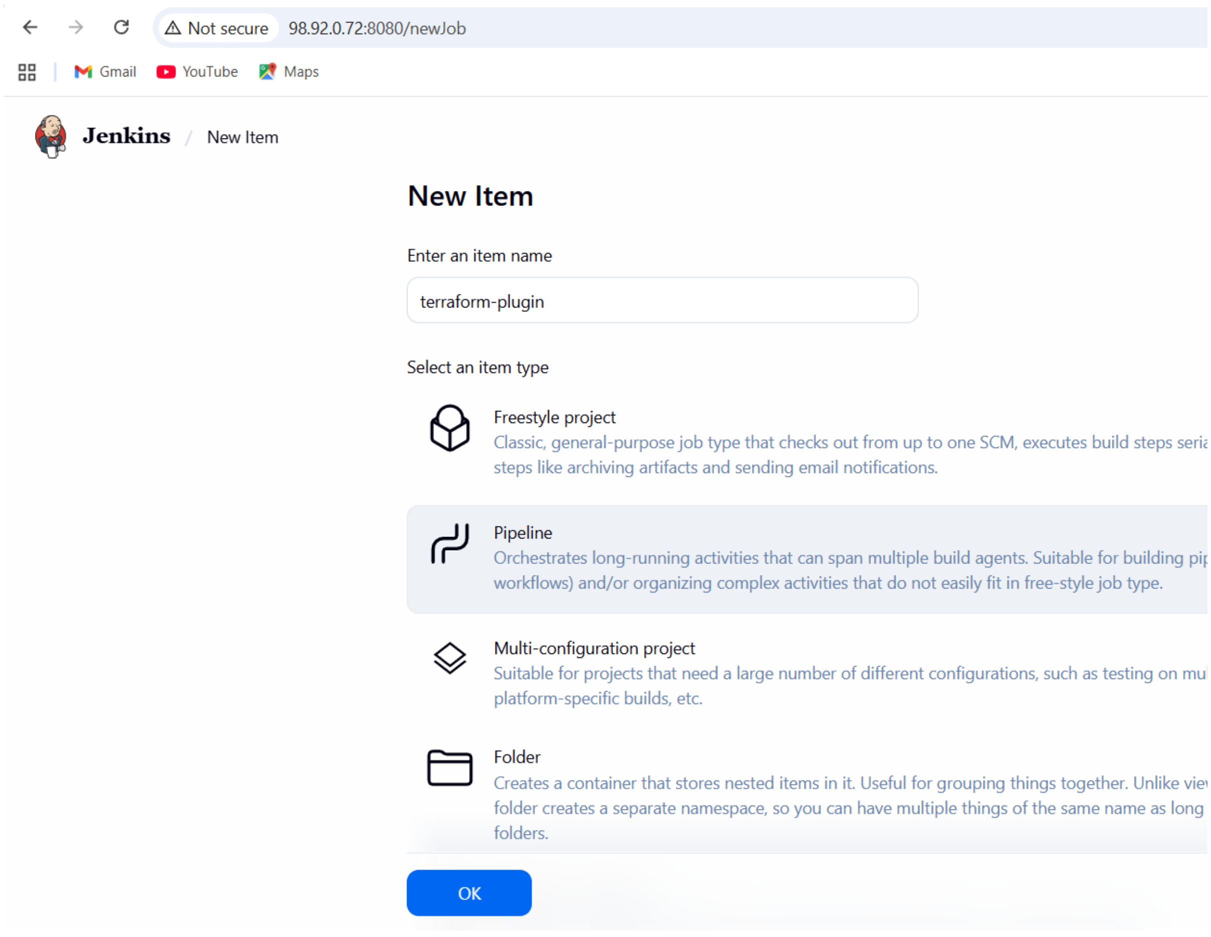
Access Key ID

AKIATNTADWLTXK67XKPJ

Secret Access Key

••

Create a new item and select pipeline.



Not secure 98.92.0.72:8080/newJob

Jenkins / New Item

New Item

Enter an item name

terraform-plugin

Select an item type

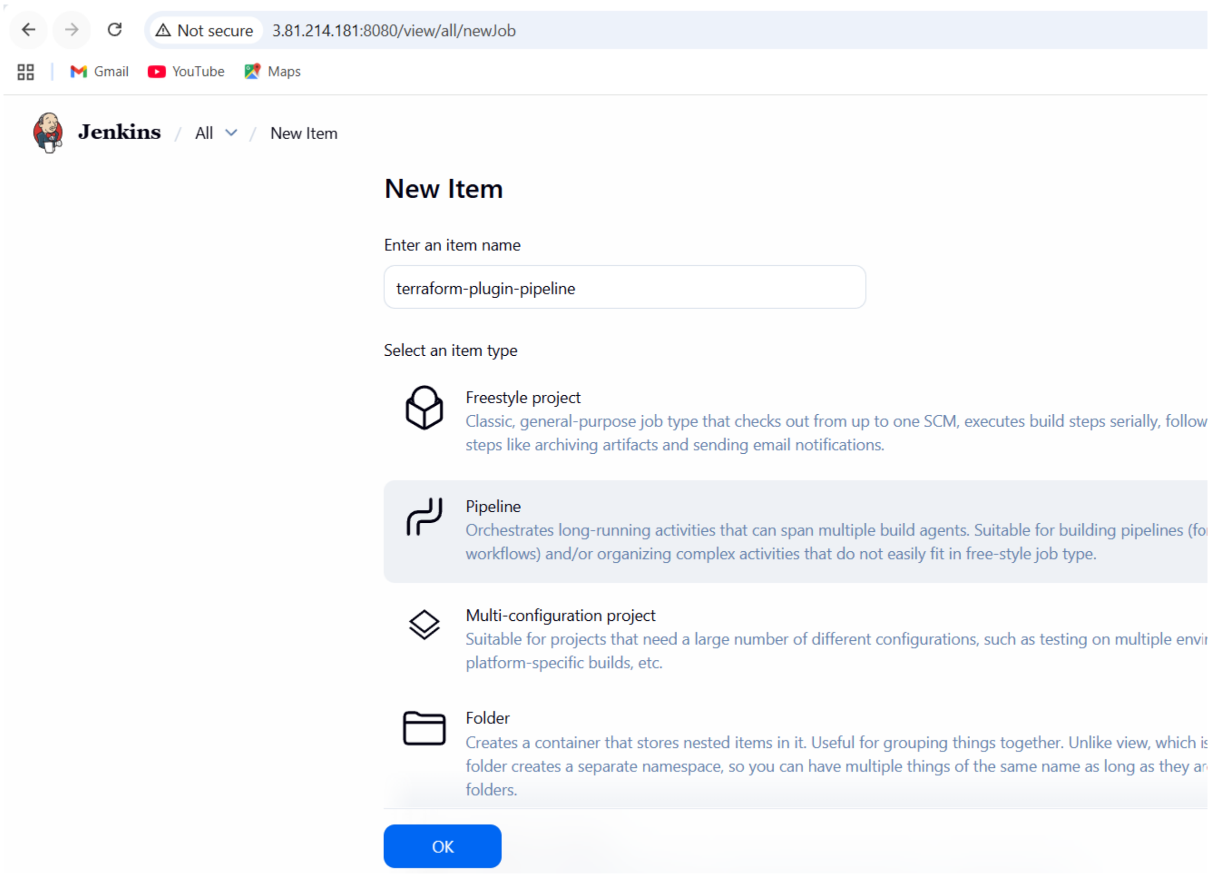
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps seria
steps like archiving artifacts and sending email notifications.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pi
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 mu
platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike vie
folder creates a separate namespace, so you can have multiple things of the same name as long
folders.

Select git in configure and give repository URL and branch click on build now.



Not secure 3.81.214.181:8080/view/all/newJob

Gmail YouTube Maps

Jenkins / All / New Item

New Item

Enter an item name

terraform-plugin-pipeline

Select an item type

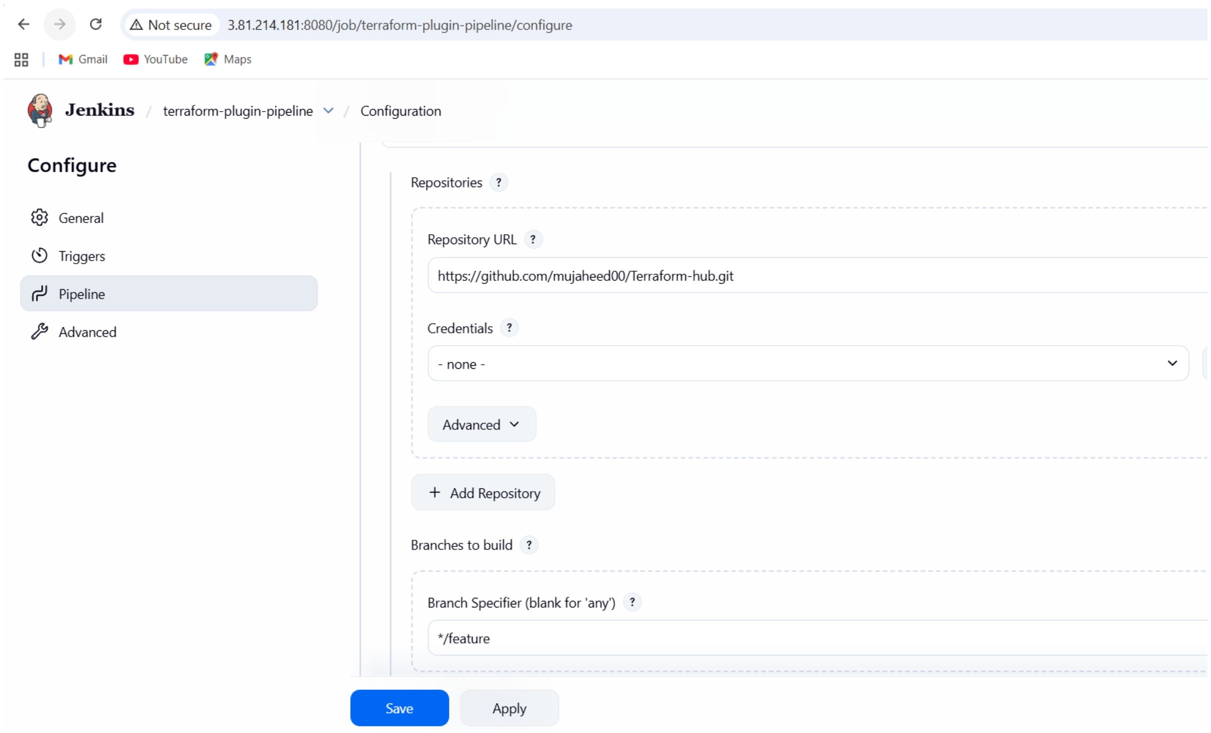
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, follow
steps like archiving artifacts and sending email notifications.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (fo
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 envi
platform-specific builds, etc.

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

OK



Not secure 3.81.214.181:8080/job/terraform-plugin-pipeline/configure

Gmail YouTube Maps

Jenkins / terraform-plugin-pipeline / Configuration

Configure

General
Triggers
Pipeline
Advanced

Repositories

Repository URL

https://github.com/mujaheed00/Terraform-hub.git

Credentials

- none -

Advanced

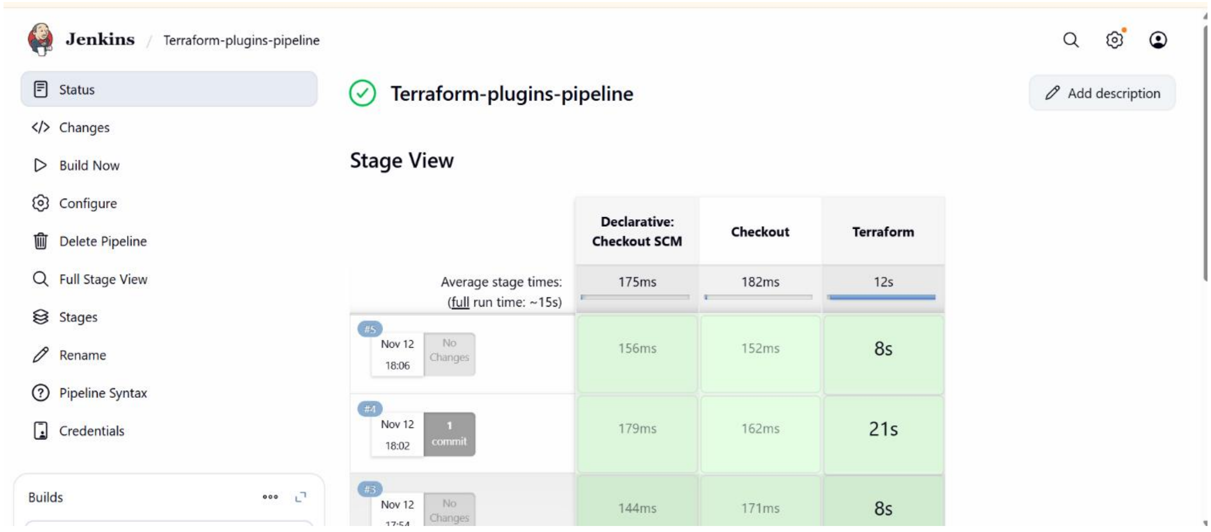
+ Add Repository

Branches to build

Branch Specifier (blank for 'any')

*/feature

Save Apply

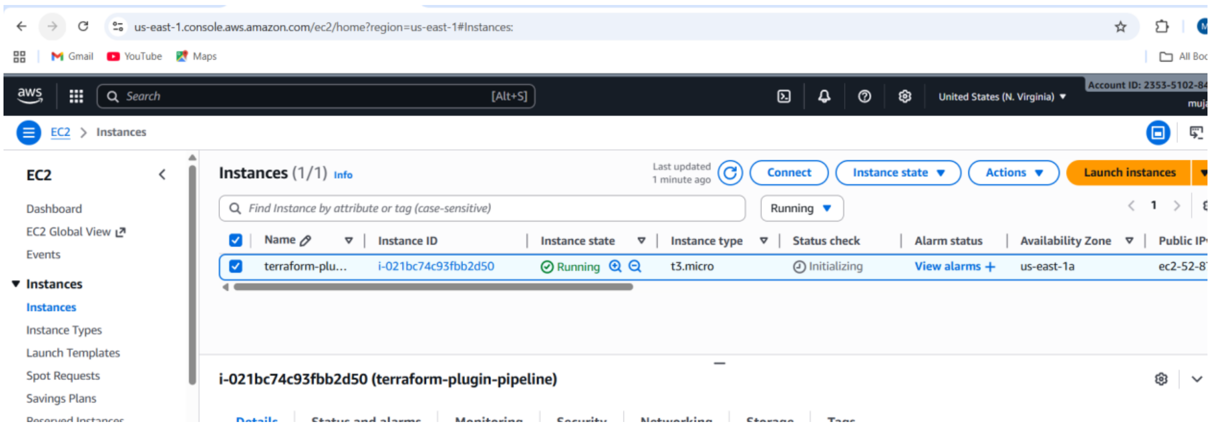

Jenkins / Terraform-plugins-pipeline

Terraform-plugins-pipeline

Stage View

	Declarative: Checkout SCM	Checkout	Terraform
Average stage times: (full run time: ~15s)	175ms	182ms	12s
#5 Nov 12 18:06 No Changes	156ms	152ms	8s
#4 Nov 12 18:02 1 commit	179ms	162ms	21s
#3 Nov 12 17:54 No Changes	144ms	171ms	8s

It will automatically create an instance.


us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:

EC2 > Instances

Instances (1/1)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
terraform-plu...	i-021bc74c93fbb2d50	Running	t3.micro	Initializing	View alarms +	us-east-1a	ec2-52-8

i-021bc74c93fbb2d50 (terraform-plugin-pipeline)