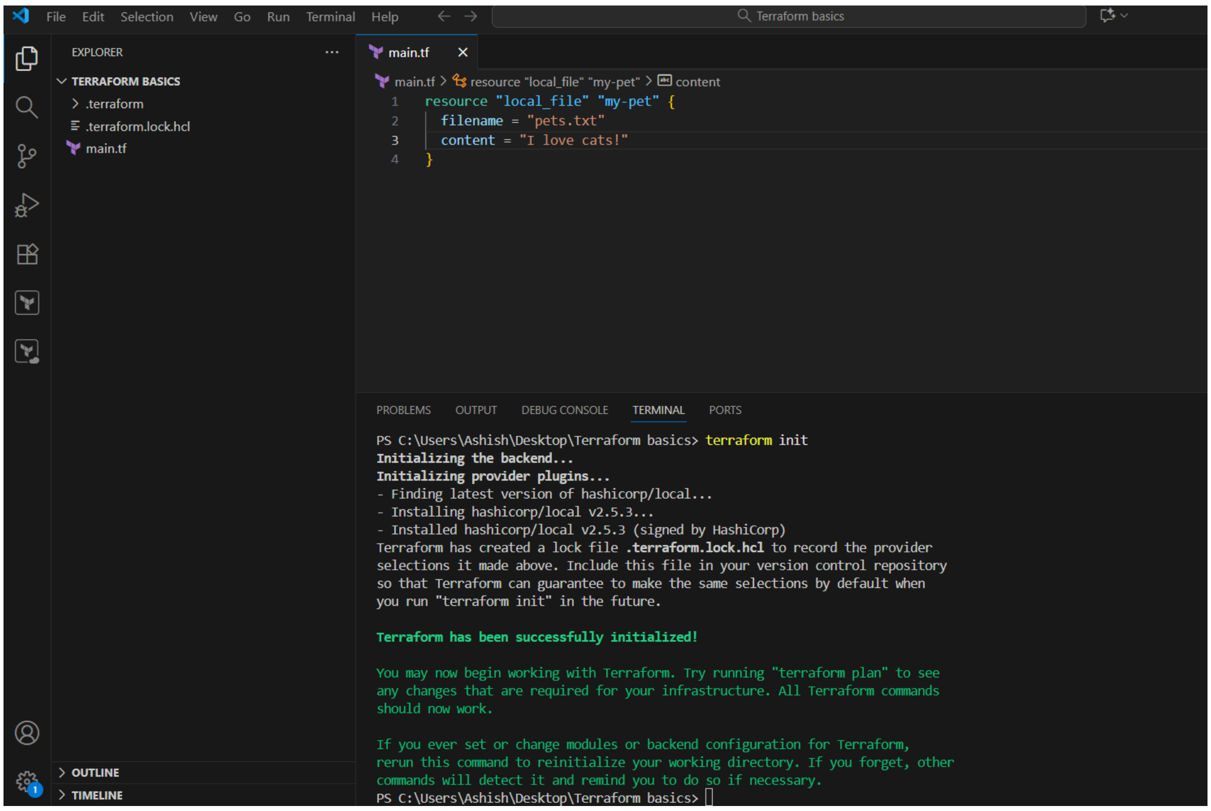
1. Watch Terraform-04 video.

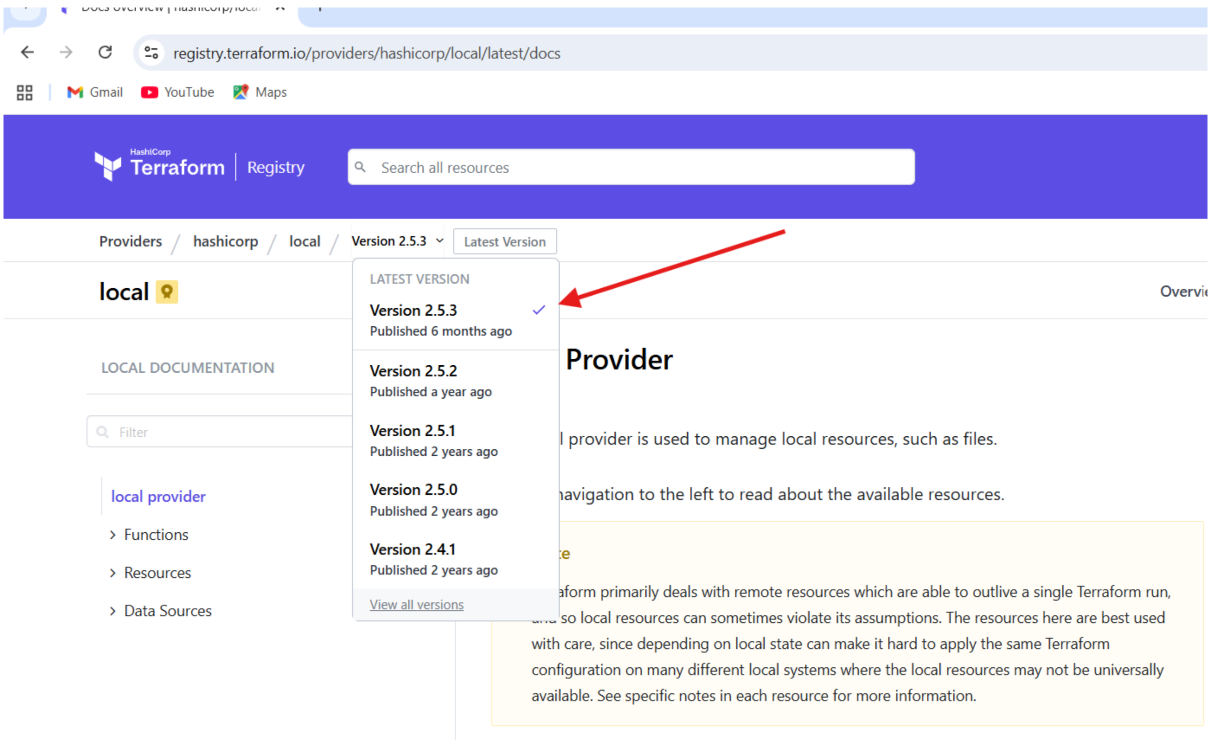
2. Execute the Script Shown in the video.


File Edit Selection View Go Run Terminal Help
Terraform basics
EXPLORER
TERRAFORM BASICS
.terraform
.terraform.lock.hcl
main.tf
main.tf > resource "local_file" "my-pet" > content
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
OUTLINE
TIMELINE

```
resource "local_file" "my-pet" {
  filename = "pets.txt"
  content = "I love cats!"
}
```

```
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>
```

it will create latest version.



Version 2.5.3
Published 6 months ago

Version 2.5.2
Published a year ago

Version 2.5.1
Published 2 years ago

Version 2.5.0
Published 2 years ago

Version 2.4.1
Published 2 years ago

Provider

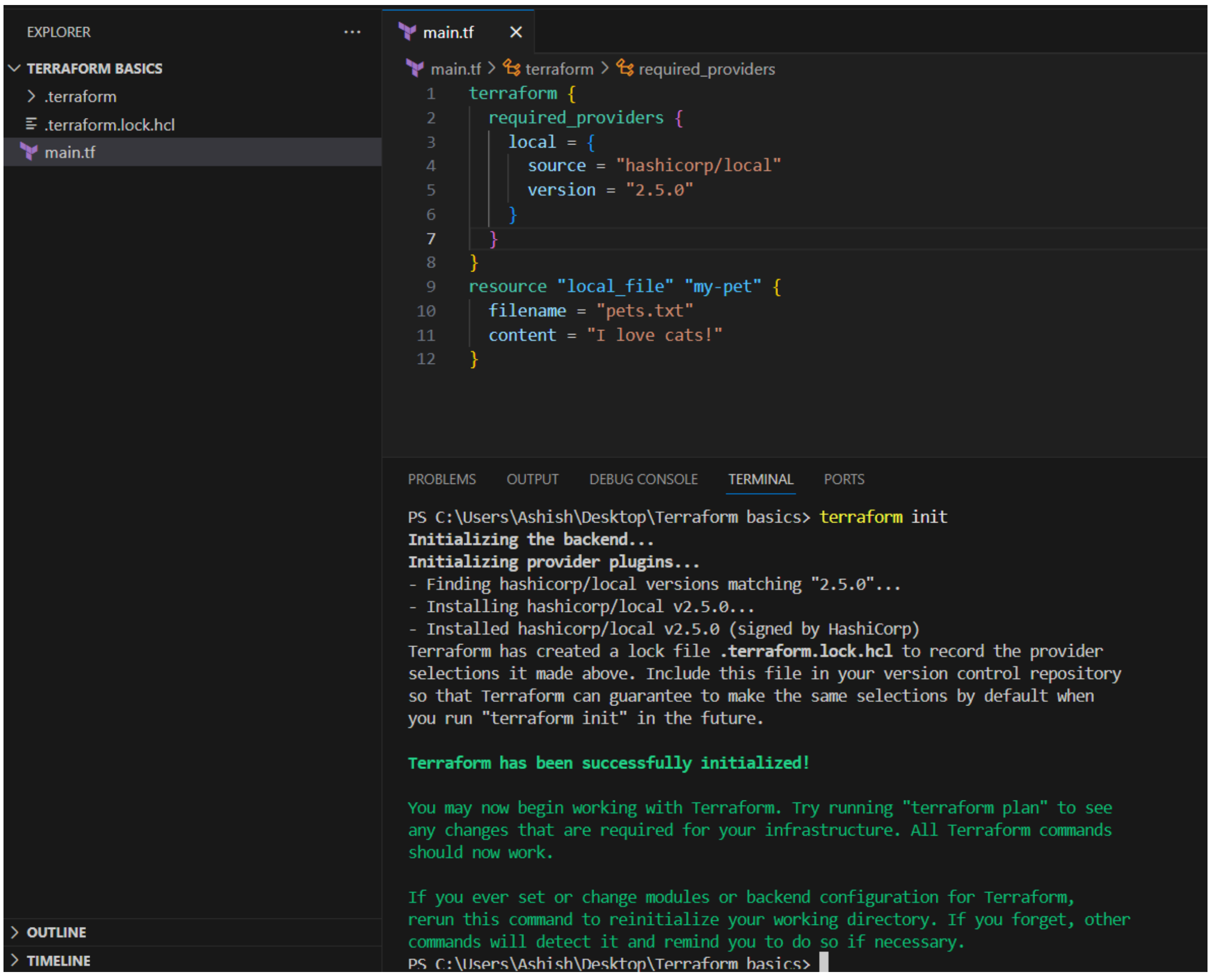
l provider is used to manage local resources, such as files.

navigation to the left to read about the available resources.

aform primarily deals with remote resources which are able to outlive a single Terraform run, so local resources can sometimes violate its assumptions. The resources here are best used with care, since depending on local state can make it hard to apply the same Terraform configuration on many different local systems where the local resources may not be universally available. See specific notes in each resource for more information.

If we provide particular version like

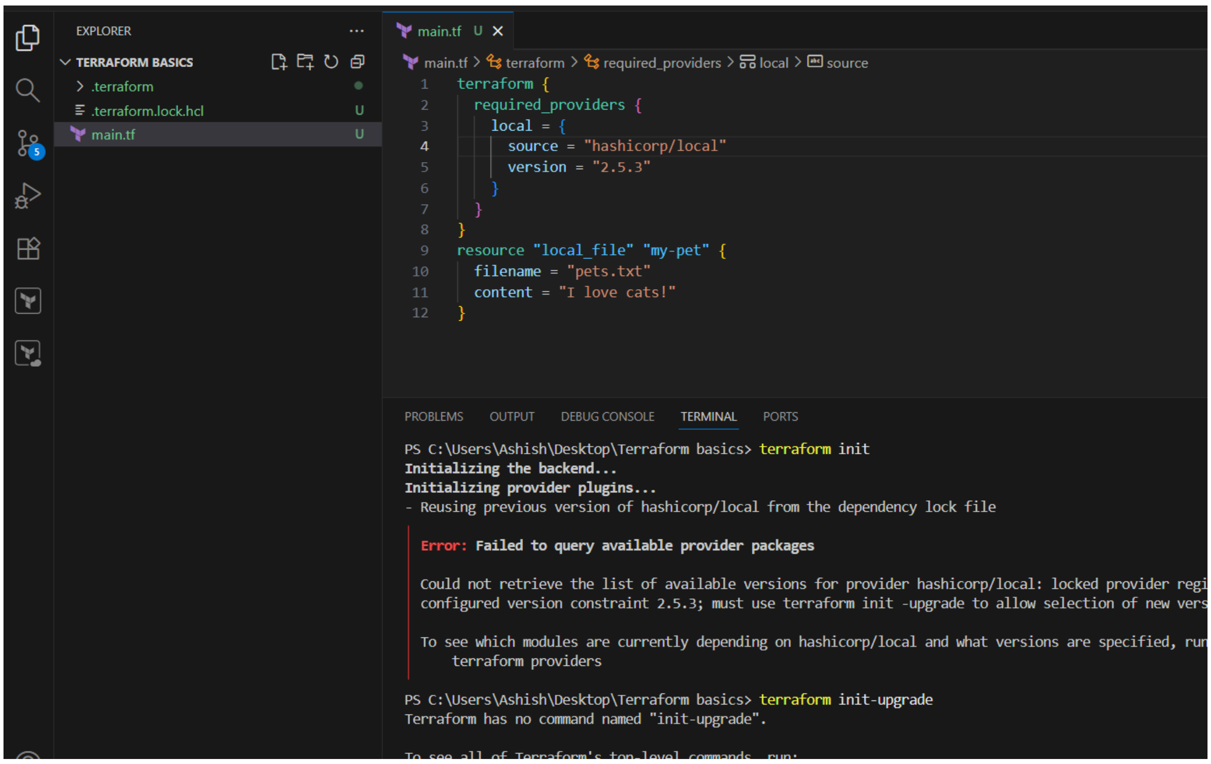
```
terraform {
  required_providers {
    local = {
      source = "hashicorp/local"
      version = "2.5.0"
    }
  }
}
```



```
terraform {
  required_providers {
    local = {
      source = "hashicorp/local"
      version = "2.5.0"
    }
  }
}
resource "local_file" "my-pet" {
  filename = "pets.txt"
  content = "I love cats!"
}
```

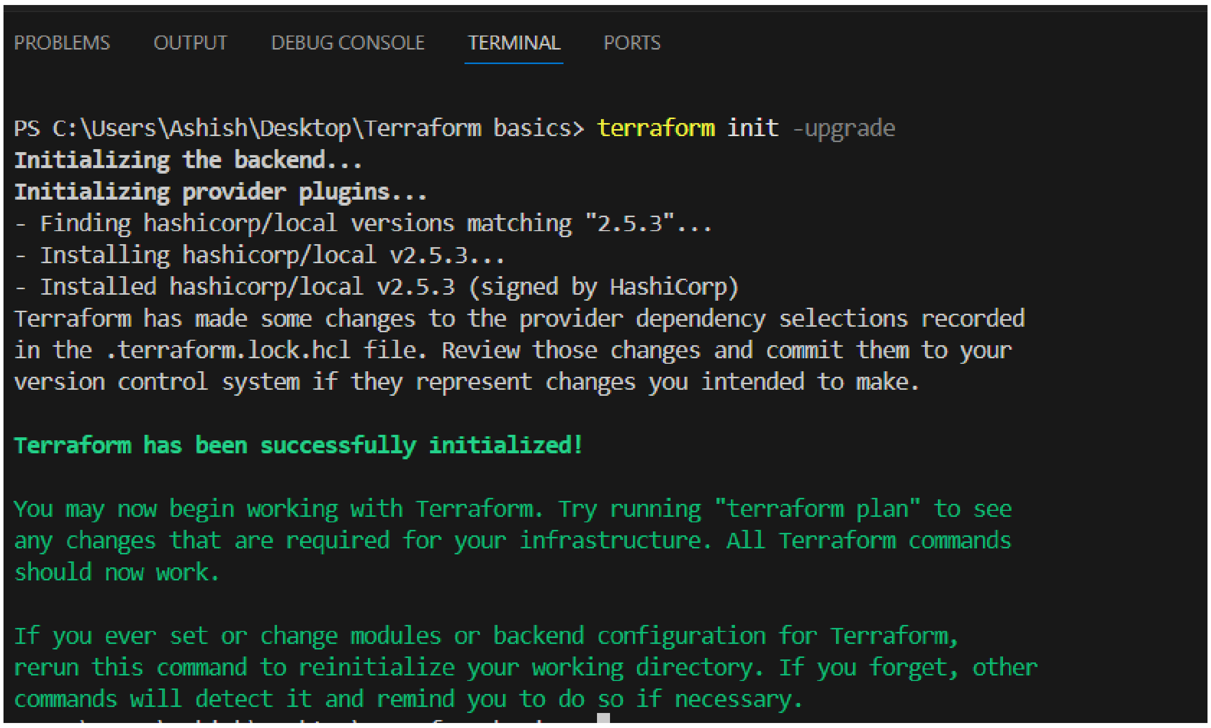
```
PS C:\Users\Ashish\Desktop\Terraform basics> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding hashicorp/local versions matching "2.5.0"...
- Installing hashicorp/local v2.5.0...
- Installed hashicorp/local v2.5.0 (signed by HashiCorp)
Terraform has created a lock 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>
```

If you need to keep the another version without deleting you will use terraform init -upgrade



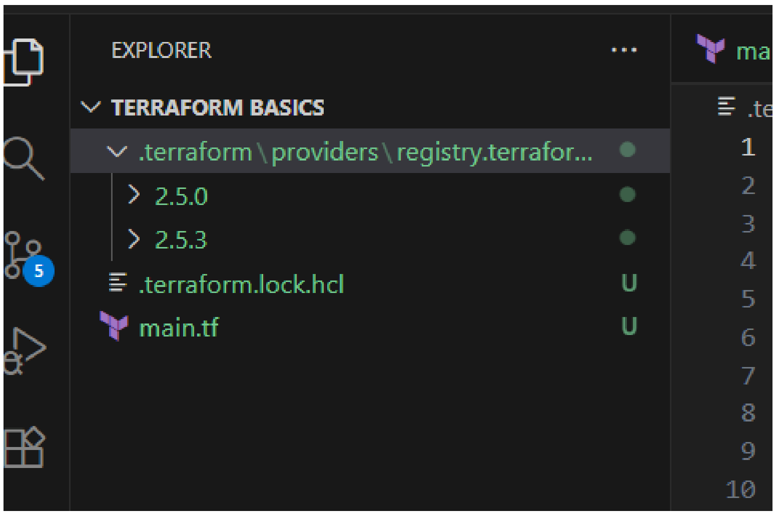
```
terraform {
  required_providers {
    local = {
      source = "hashicorp/local"
      version = "2.5.3"
    }
  }
}
resource "local_file" "my-pet" {
  filename = "pets.txt"
  content = "I love cats!"
}
```

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

  Error: Failed to query available provider packages

  Could not retrieve the list of available versions for provider hashicorp/local: locked provider regi
  configured version constraint 2.5.3; must use terraform init -upgrade to allow selection of new vers

  To see which modules are currently depending on hashicorp/local and what versions are specified, run
      terraform providers

PS C:\Users\Ashish\Desktop\Terraform basics> terraform init-upgrade
Terraform has no command named "init-upgrade".

To see all of Terraform's top-level commands, run:
```



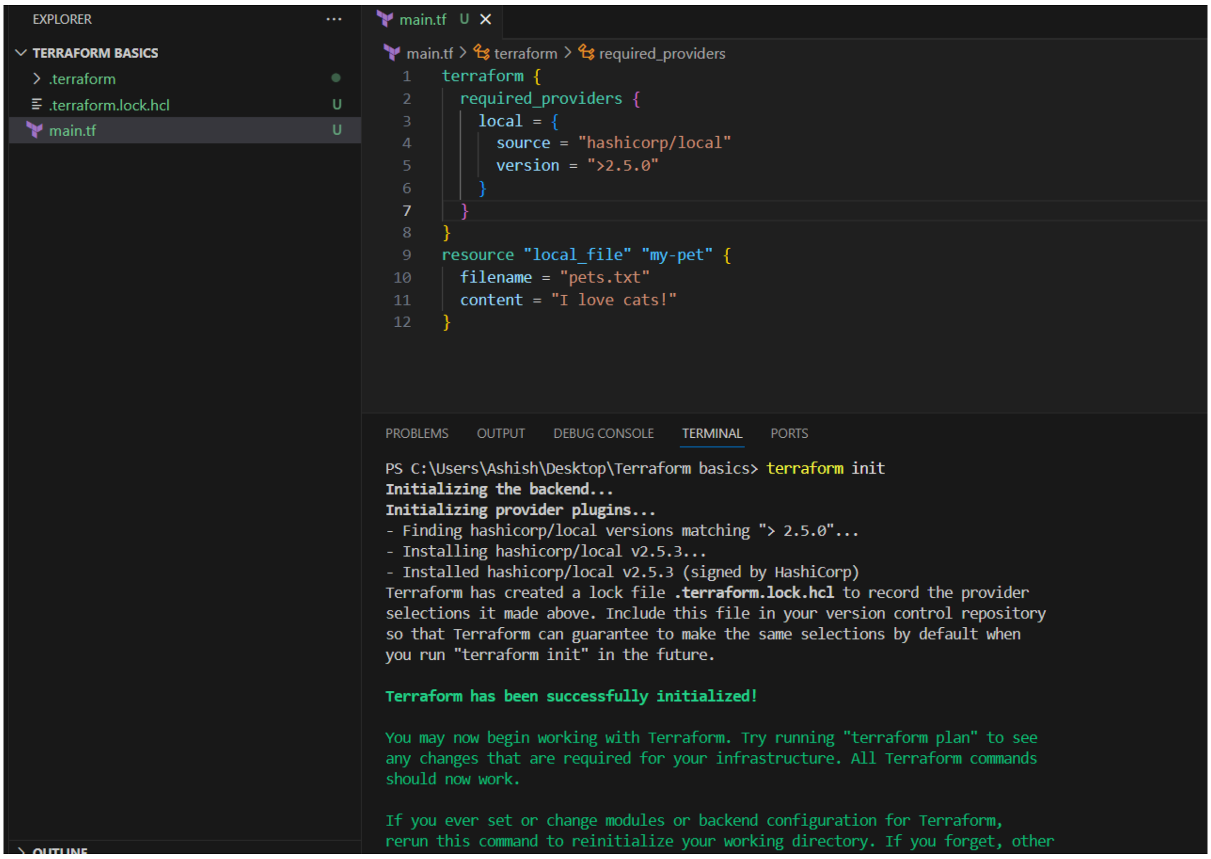
```
PS C:\Users\Ashish\Desktop\Terraform basics> terraform init -upgrade
Initializing the backend...
Initializing provider plugins...
- Finding hashicorp/local versions matching "2.5.3"...
- Installing hashicorp/local v2.5.3...
- Installed hashicorp/local v2.5.3 (signed by HashiCorp)
Terraform has made 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.
```

We will see both the versions are here.



If we need specific version we can provide like >, !=, <

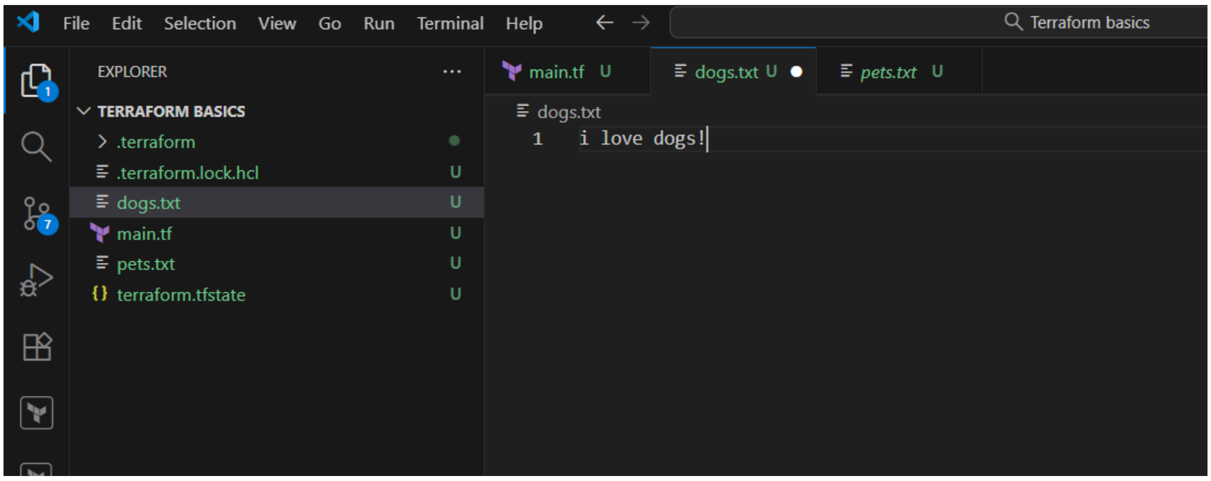
If I provide here like >2.5.0 it will download 2.5.3 because in terraform registry available versions only will be downloaded.



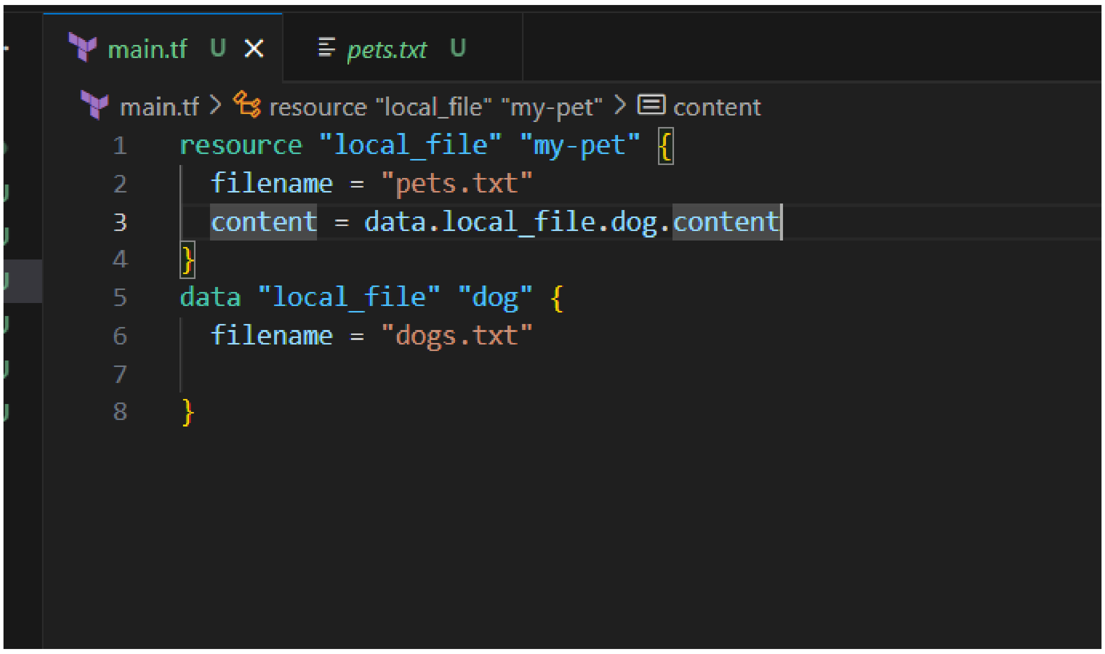
```
terraform {
  required_providers {
    local = {
      source = "hashicorp/local"
      version = ">2.5.0"
    }
  }
}
resource "local_file" "my-pet" {
  filename = "pets.txt"
  content = "I love cats!"
}
```

```
PS C:\Users\Ashish\Desktop\Terraform basics> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding hashicorp/local versions matching "> 2.5.0"...
- 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
```

Create a file named as dogs.txt



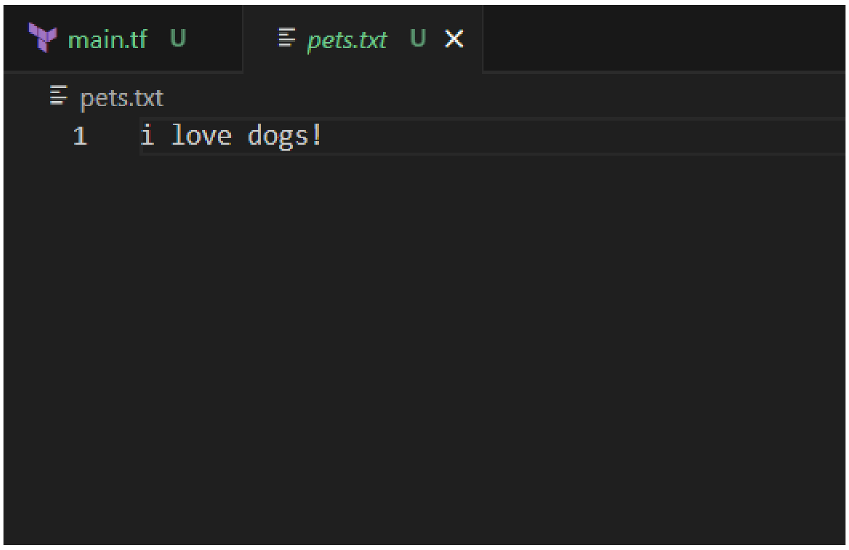
```
i love dogs!
```



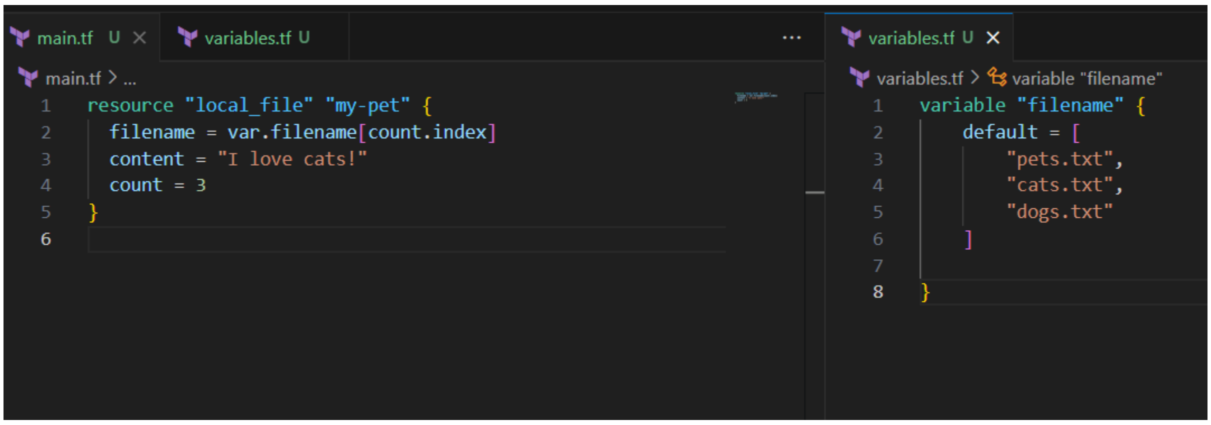
```
resource "local_file" "my-pet" {
  filename = "pets.txt"
  content = data.local_file.dog.content
}
data "local_file" "dog" {
  filename = "dogs.txt"

}
```

If I do terraform apply it will create a pets.txt

Data block is used to read the content.

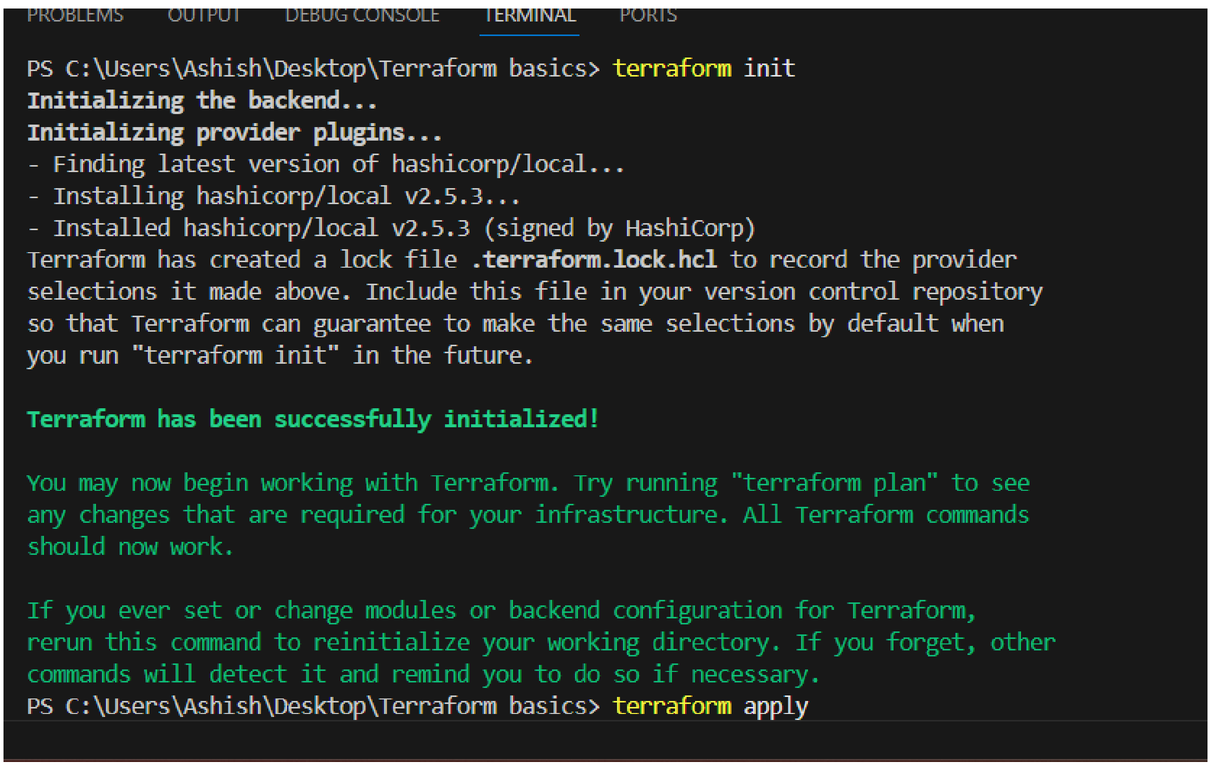


```
i love dogs!
```

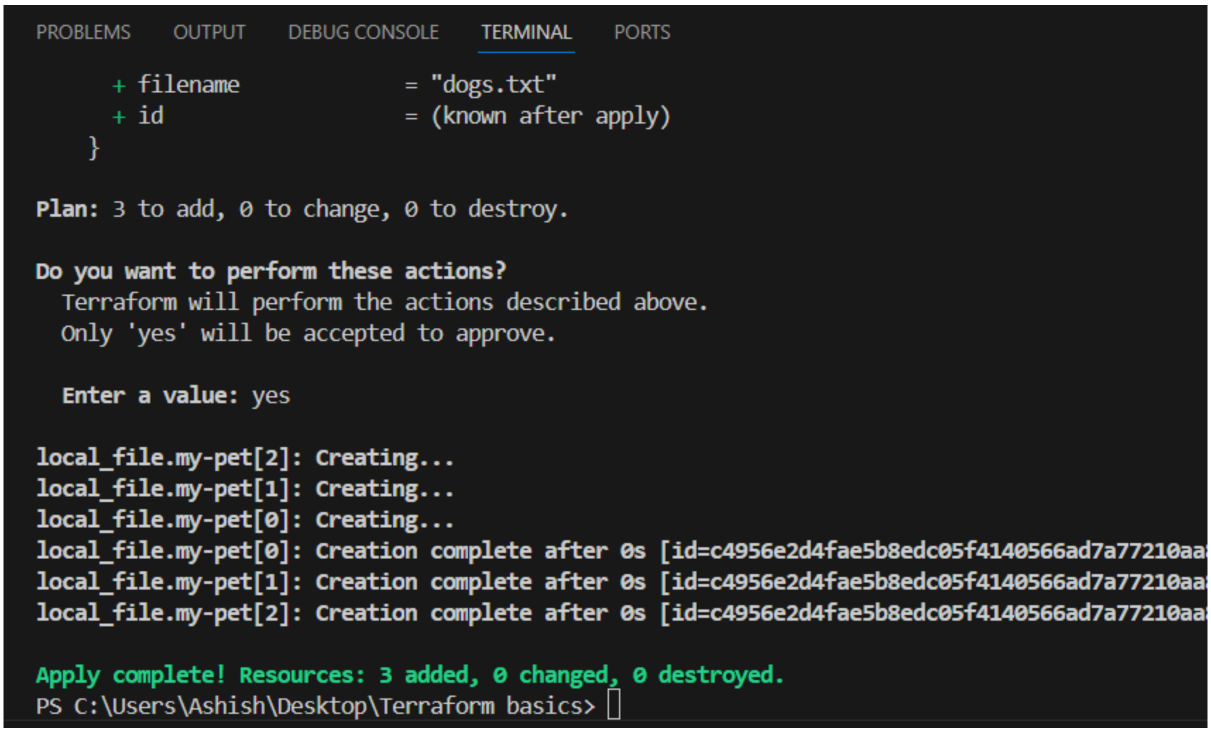


```
resource "local_file" "my-pet" {
  filename = var.filename[count.index]
  content = "I love cats!"
  count = 3
}
```

```
variable "filename" {
    default = [
        "pets.txt",
        "cats.txt",
        "dogs.txt"
    ]

}
```

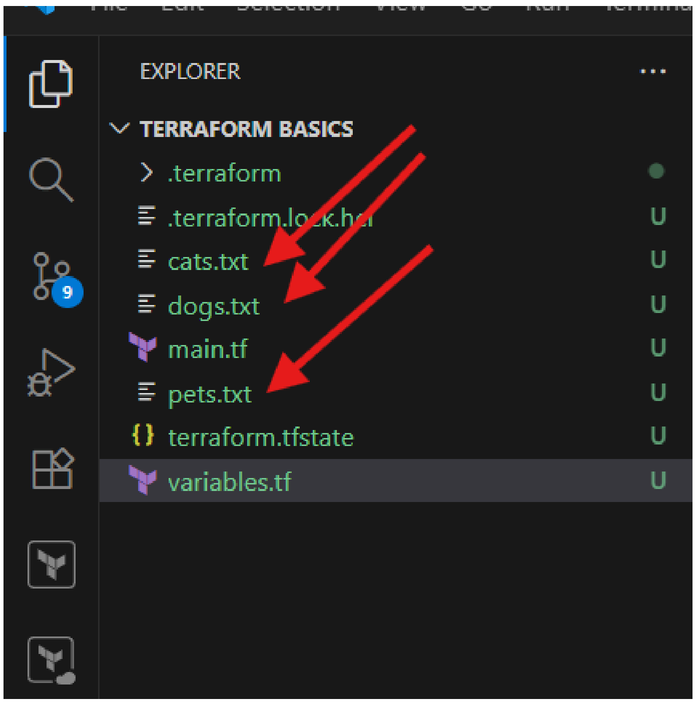


```
PS C:\Users\Ashish\Desktop\Terraform basics> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/local...
- Installing hashicorp/local v2.5.3...
- Installed hashicorp/local v2.5.3 (signed by HashiCorp)
Terraform has created a lock file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

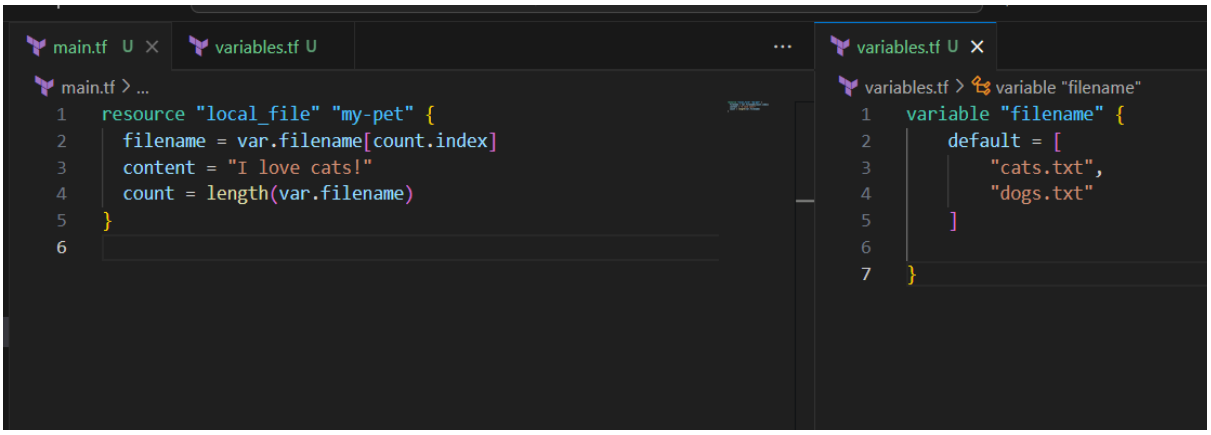


```
      + filename             = "dogs.txt"
      + id                   = (known after apply)
    }

Plan: 3 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[2]: Creating...
local_file.my-pet[1]: Creating...
local_file.my-pet[0]: Creating...
local_file.my-pet[0]: Creation complete after 0s [id=c4956e2d4fae5b8edc05f4140566ad7a77210aa
local_file.my-pet[1]: Creation complete after 0s [id=c4956e2d4fae5b8edc05f4140566ad7a77210aa
local_file.my-pet[2]: Creation complete after 0s [id=c4956e2d4fae5b8edc05f4140566ad7a77210aa

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
PS C:\Users\Ashish\Desktop\Terraform basics>
```

3 files will be created.

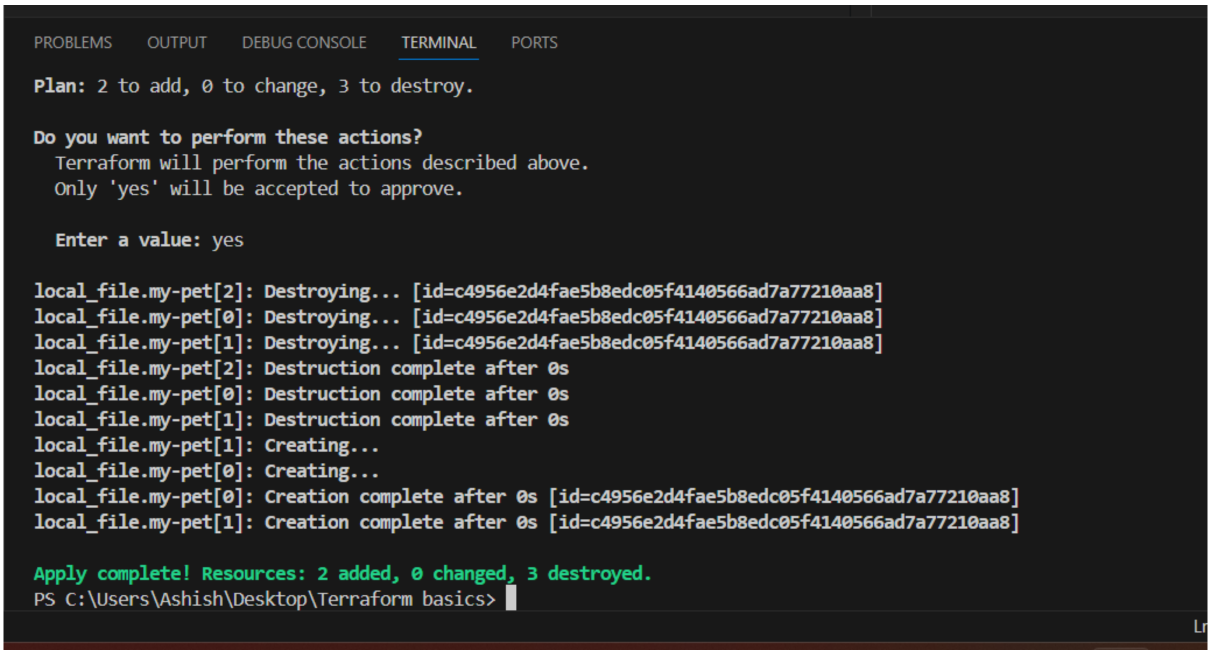


We can give count based on length



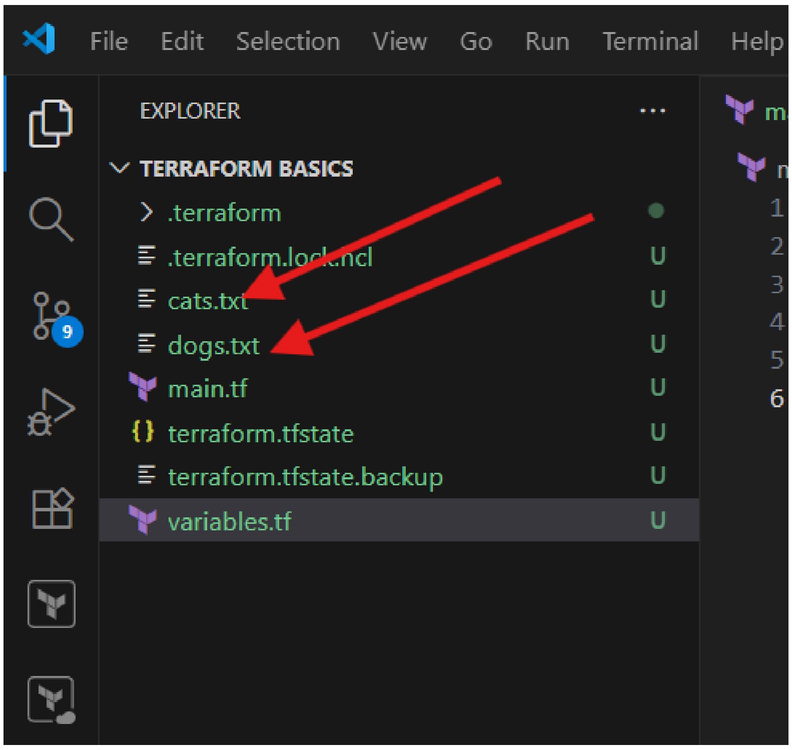
```
resource "local_file" "my-pet" {
  filename = var.filename[count.index]
  content = "I love cats!"
  count = length(var.filename)
}
```

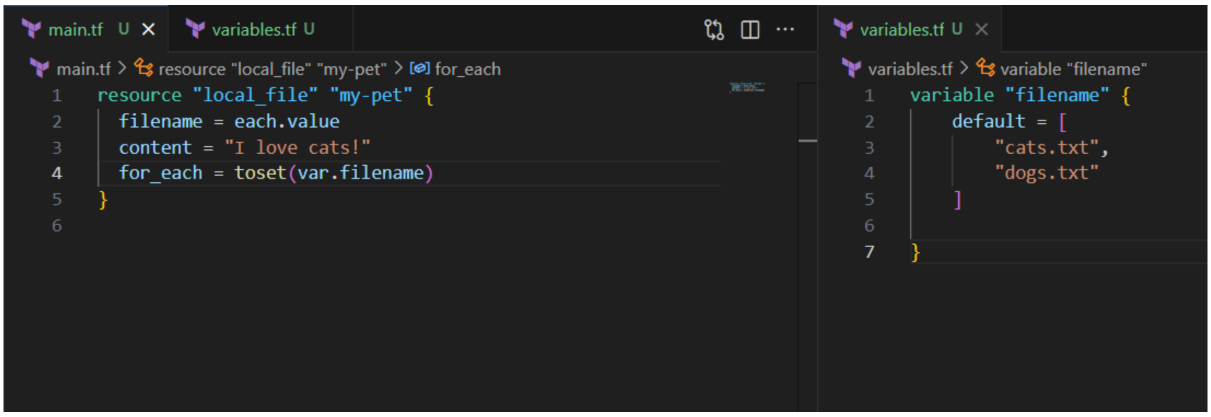
```
variable "filename" {
    default = [
        "cats.txt",
        "dogs.txt"
    ]

}
```



```
Plan: 2 to add, 0 to change, 3 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[2]: Destroying... [id=c4956e2d4fae5b8edc05f4140566ad7a77210aa8]
local_file.my-pet[0]: Destroying... [id=c4956e2d4fae5b8edc05f4140566ad7a77210aa8]
local_file.my-pet[1]: Destroying... [id=c4956e2d4fae5b8edc05f4140566ad7a77210aa8]
local_file.my-pet[2]: Destruction complete after 0s
local_file.my-pet[0]: Destruction complete after 0s
local_file.my-pet[1]: Destruction complete after 0s
local_file.my-pet[1]: Creating...
local_file.my-pet[0]: Creating...
local_file.my-pet[0]: Creation complete after 0s [id=c4956e2d4fae5b8edc05f4140566ad7a77210aa8]
local_file.my-pet[1]: Creation complete after 0s [id=c4956e2d4fae5b8edc05f4140566ad7a77210aa8]

Apply complete! Resources: 2 added, 0 changed, 3 destroyed.
PS C:\Users\Ashish\Desktop\Terraform basics>
```

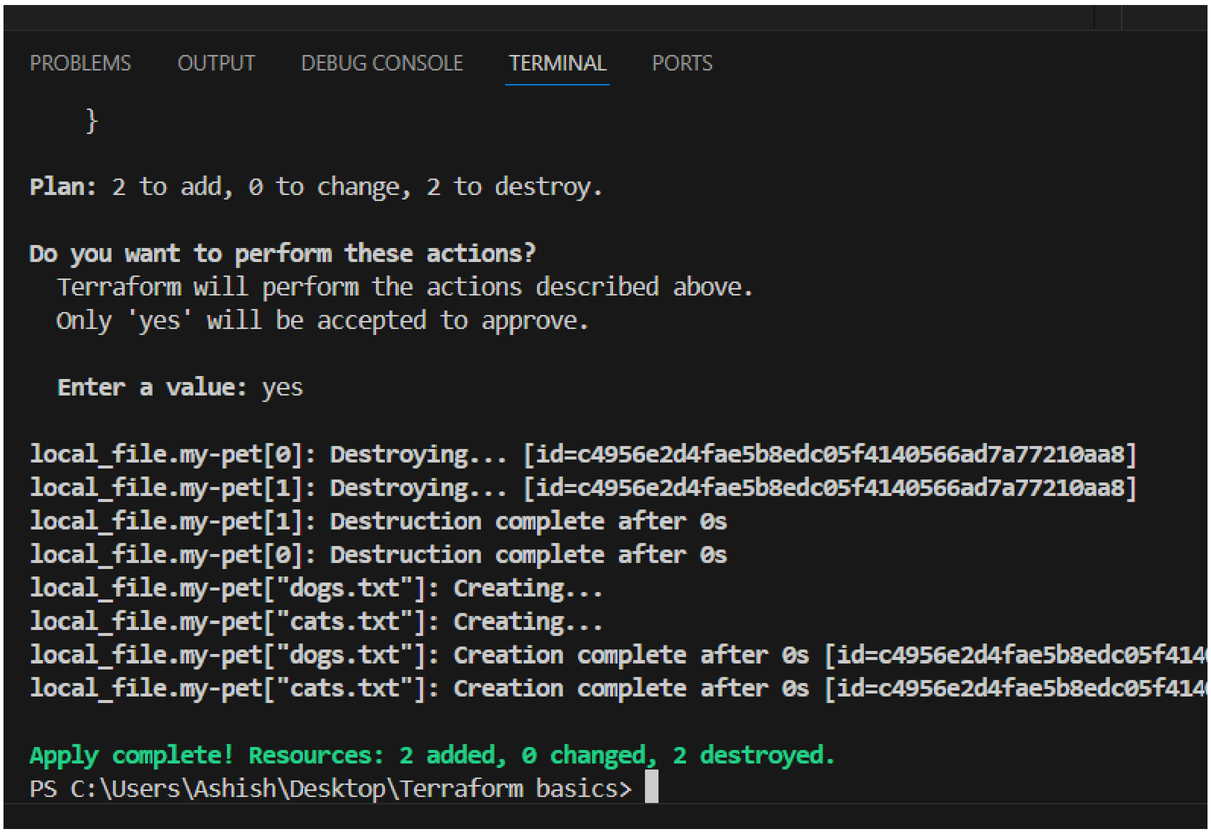
Only 2 only created.



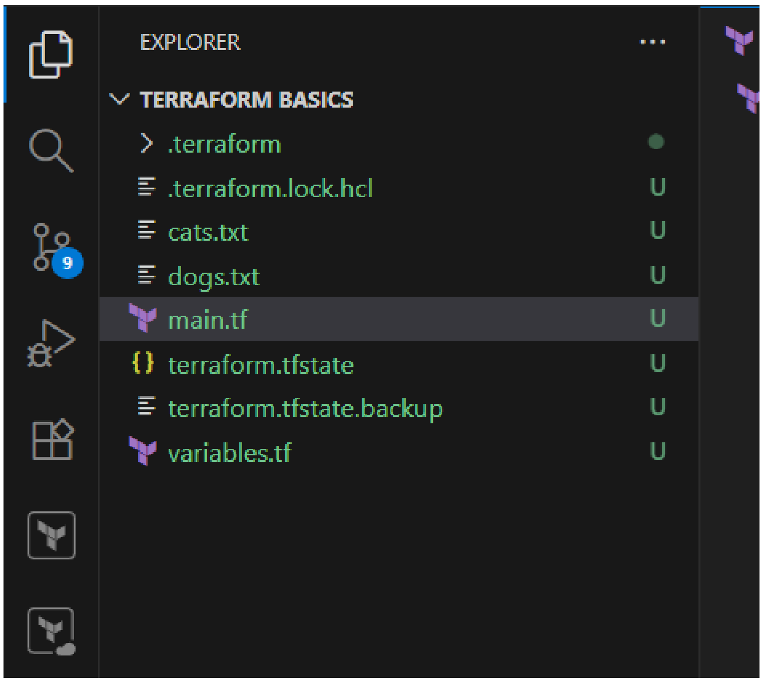

main.tf U
variables.tf U
main.tf > resource "local_file" "my-pet" > for_each
variables.tf > variable "filename"

```
resource "local_file" "my-pet" {
  filename = each.value
  content = "I love cats!"
  for_each = toset(var.filename)
}
```

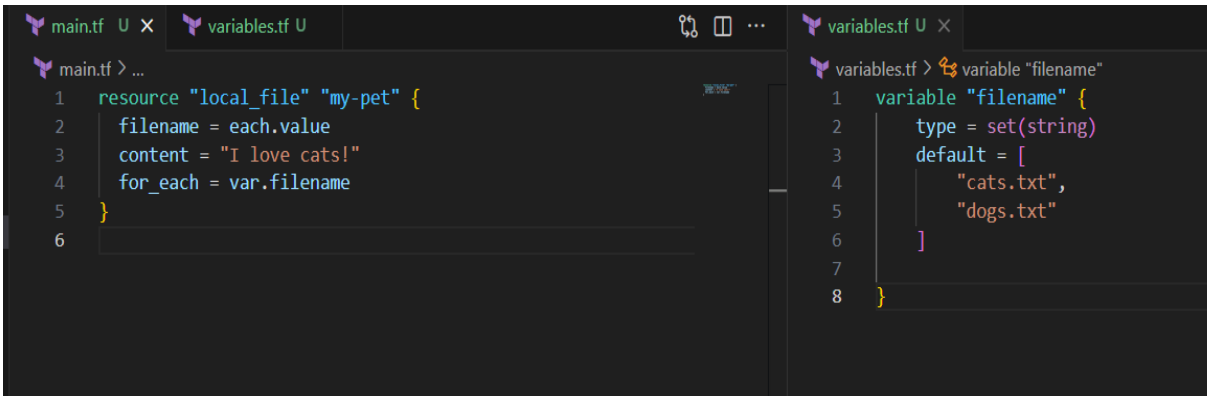
```
variable "filename" {
    default = [
        "cats.txt",
        "dogs.txt"
    ]

}
```



```
    }

Plan: 2 to add, 0 to change, 2 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[0]: Destroying... [id=c4956e2d4fae5b8edc05f4140566ad7a77210aa8]
local_file.my-pet[1]: Destroying... [id=c4956e2d4fae5b8edc05f4140566ad7a77210aa8]
local_file.my-pet[1]: Destruction complete after 0s
local_file.my-pet[0]: Destruction complete after 0s
local_file.my-pet["dogs.txt"]: Creating...
local_file.my-pet["cats.txt"]: Creating...
local_file.my-pet["dogs.txt"]: Creation complete after 0s [id=c4956e2d4fae5b8edc05f414
local_file.my-pet["cats.txt"]: Creation complete after 0s [id=c4956e2d4fae5b8edc05f414

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

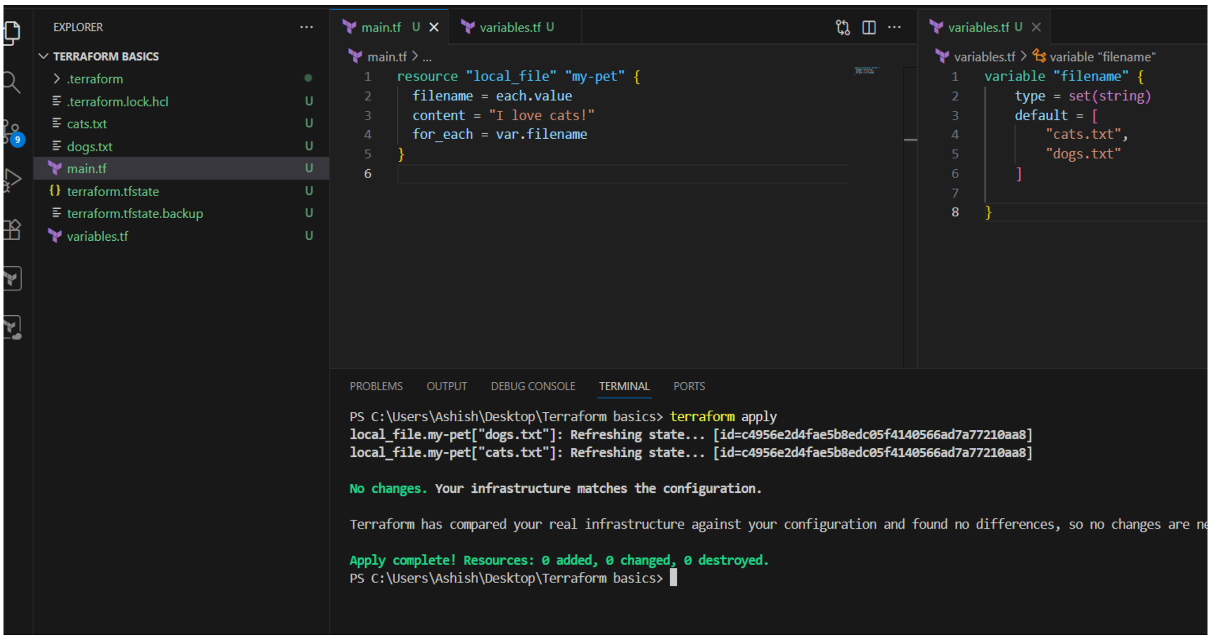


```
> .terraform
.terraform.lock.hcl       U
cats.txt                  U
dogs.txt                  U
main.tf                   U
terraform.tfstate         U
terraform.tfstate.backup  U
variables.tf              U
```



```
main.tf
resource "local_file" "my-pet" {
  filename = each.value
  content = "I love cats!"
  for_each = var.filename
}
```

```
variables.tf > variable "filename"
variable "filename" {
  type = set(string)
  default = [
    "cats.txt",
    "dogs.txt"
  ]

}
```



```
EXPLORER
TERRAFORM BASICS
> .terraform
  .terraform.lock.hcl        U
  cats.txt                   U
  dogs.txt                   U
  main.tf                    U
  terraform.tfstate          U
  terraform.tfstate.backup   U
  variables.tf               U
```

```
main.tf
resource "local_file" "my-pet" {
  filename = each.value
  content = "I love cats!"
  for_each = var.filename
}
```

```
variables.tf > variable "filename"
variable "filename" {
  type = set(string)
  default = [
    "cats.txt",
    "dogs.txt"
  ]

}
```

```
PS C:\Users\Ashish\Desktop\Terraform basics> terraform apply
local_file.my-pet["dogs.txt"]: Refreshing state... [id=c4956e2d4fae5b8edc05f4140566ad7a77210aa8]
local_file.my-pet["cats.txt"]: Refreshing state... [id=c4956e2d4fae5b8edc05f4140566ad7a77210aa8]

No changes. Your infrastructure matches the configuration.

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

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

Go to IAM there is no users in the IAM we can create ec2 resoures with terraform.

resource "aws_iam_user" "Admin-user" {

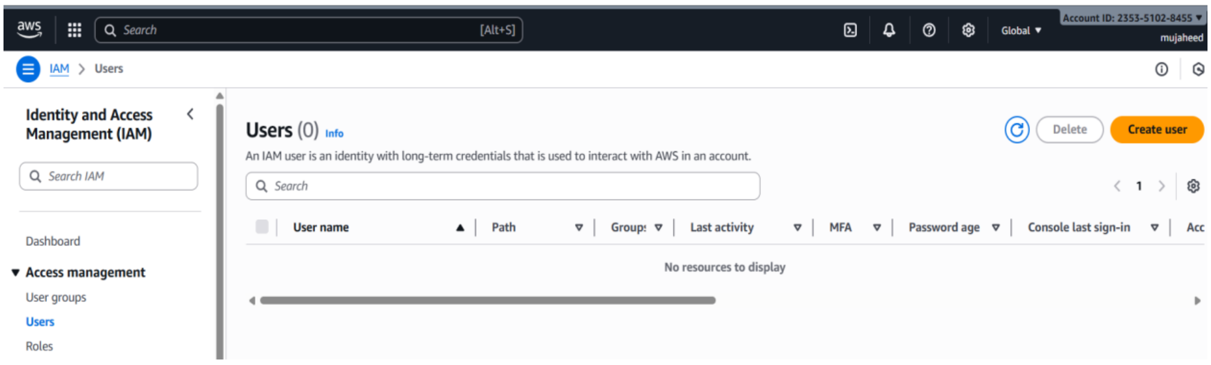
name = "lucy"

tags = {

  "description" = "Technical Team Lead"

}

}



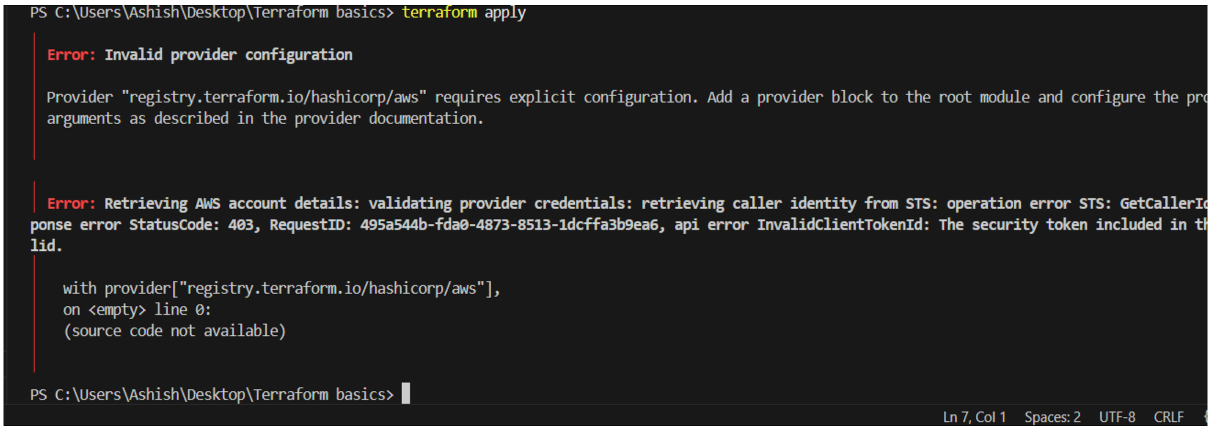
Users (0) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

User name	Path	Groups	Last activity	MFA	Password age	Console last sign-in	Acc

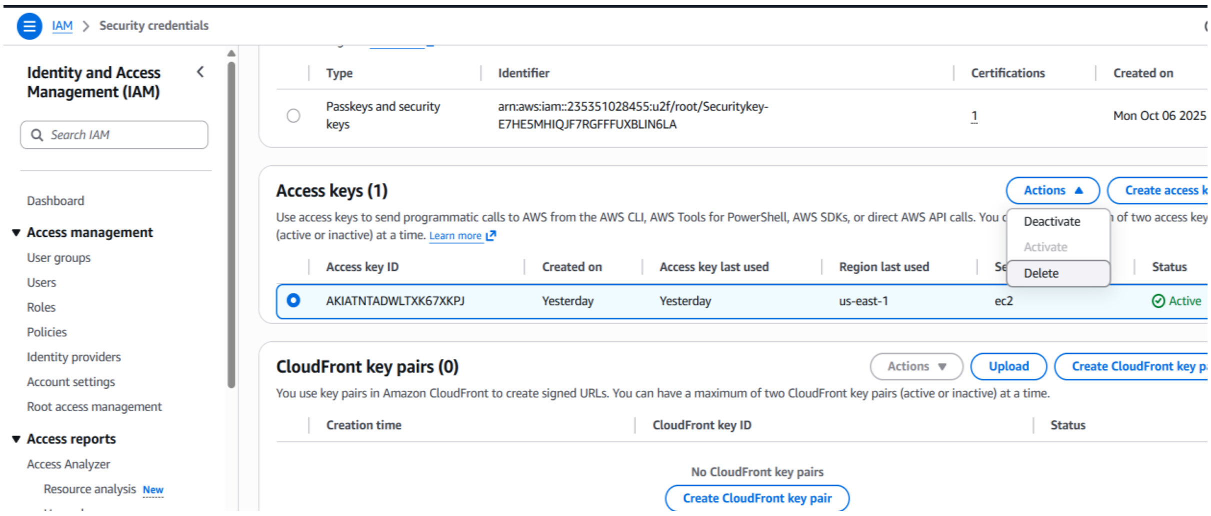
No resources to display

- First we need to delete our aws credentials other wise it will show an error like this.



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

  Error: Invalid provider configuration

  Provider "registry.terraform.io/hashicorp/aws" requires explicit configuration. Add a provider block to the root module and configure the pro
  arguments as described in the provider documentation.


  Error: Retrieving AWS account details: validating provider credentials: retrieving caller identity from STS: operation error STS: GetCallerIc
ponse error StatusCode: 403, RequestID: 495a544b-fda0-4873-8513-1dcffa3b9ea6, api error InvalidClientTokenId: The security token included in th
lid.

    with provider["registry.terraform.io/hashicorp/aws"],
    on <empty> line 0:
    (source code not available)

PS C:\Users\Ashish\Desktop\Terraform basics> █
                                                                          Ln 7, Col 1   Spaces: 2   UTF-8   CRLF
```

Delete this key.



Type	Identifier	Certifications	Created on
Passkeys and security keys	arn:aws:iam::235351028455:u2f/root/Securitykey-E7HE5MHIQJF7RGFFFUXBLIN6LA	1	Mon Oct 06 2025

Access keys (1)

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You c … of two access key (active or inactive) at a time. Learn more

Access key ID	Created on	Access key last used	Region last used	Se	Status
AKIATNTADWLTXK67XKPJ	Yesterday	Yesterday	us-east-1	ec2	Active

CloudFront key pairs (0)

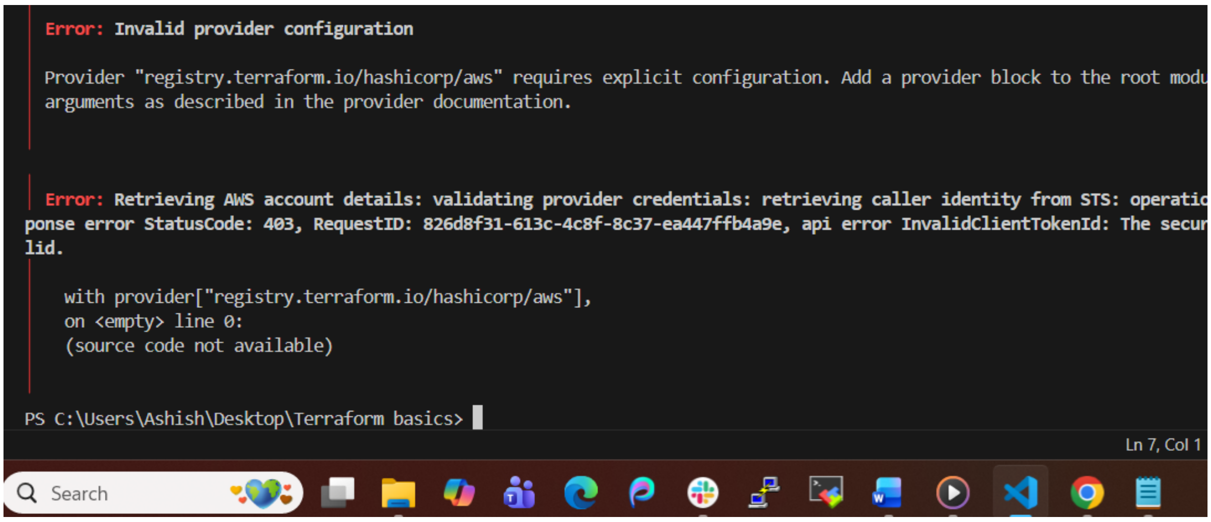
You use key pairs in Amazon CloudFront to create signed URLs. You can have a maximum of two CloudFront key pairs (active or inactive) at a time.

Creation time	CloudFront key ID	Status

No CloudFront key pairs

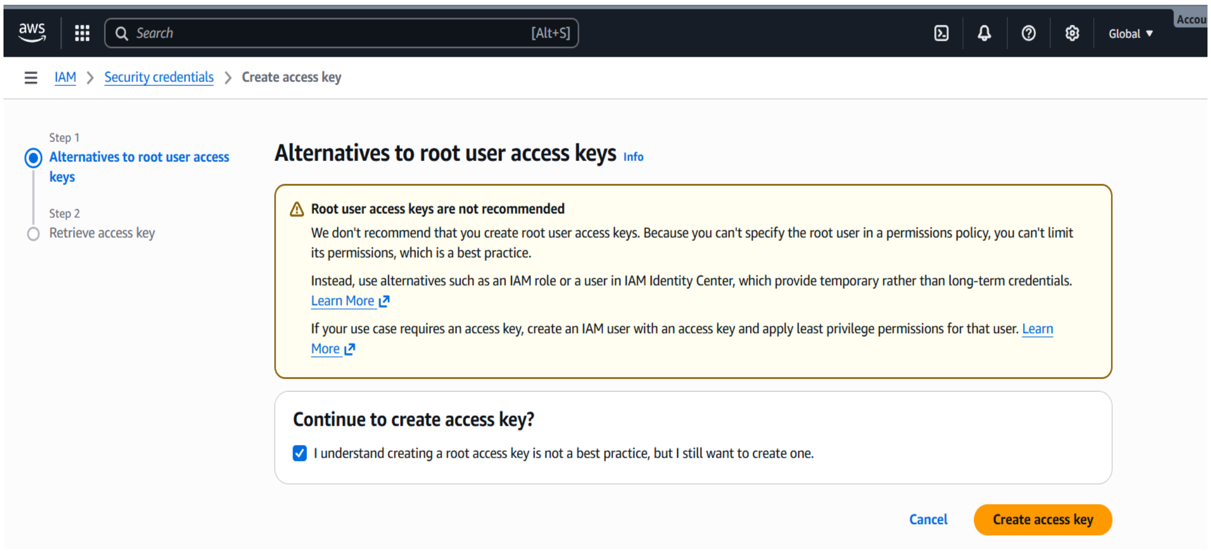
Create CloudFront key pair

Again it shown error.



```
Error: Invalid provider configuration

Provider "registry.terraform.io/hashicorp/aws" requires explicit configuration. Add a provider block to the root modu
arguments as described in the provider documentation.


Error: Retrieving AWS account details: validating provider credentials: retrieving caller identity from STS: operatio
ponse error StatusCode: 403, RequestID: 826d8f31-613c-4c8f-8c37-ea447ffb4a9e, api error InvalidClientTokenId: The secur
lid.

  with provider["registry.terraform.io/hashicorp/aws"],
  on <empty> line 0:
  (source code not available)

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

Here we need to give credentials so that we need to create access key and secret key.



IAM > Security credentials > Create access key

Step 1 Alternatives to root user access keys
Step 2 Retrieve access key

Alternatives to root user access keys Info

Root user access keys are not recommended

We don't recommend that you create root user access keys. Because you can't specify the root user in a permissions policy, you can't limit its permissions, which is a best practice.

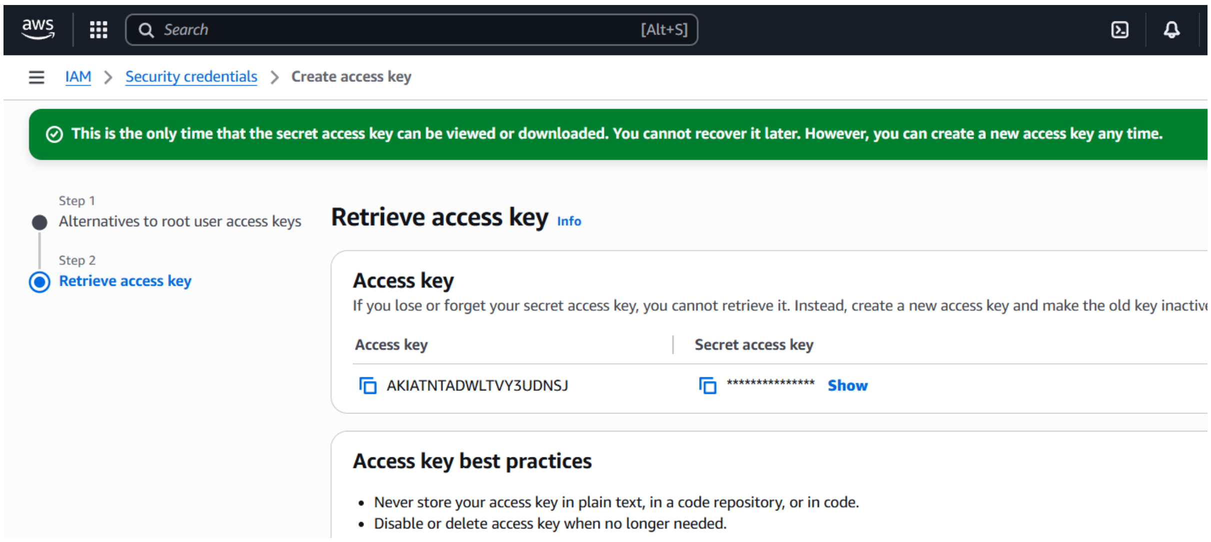
Instead, use alternatives such as an IAM role or a user in IAM Identity Center, which provide temporary rather than long-term credentials. Learn More

If your use case requires an access key, create an IAM user with an access key and apply least privilege permissions for that user. Learn More

Continue to create access key?

[✓] I understand creating a root access key is not a best practice, but I still want to create one.

Cancel | Create access key



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

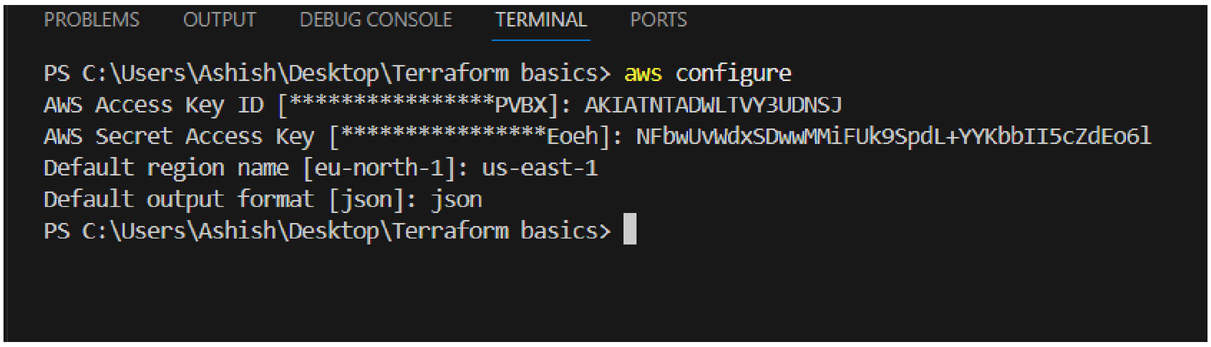
If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactiv

Access key	Secret access key
AKIATNTADWLTVY3UDNSJ	*************** Show

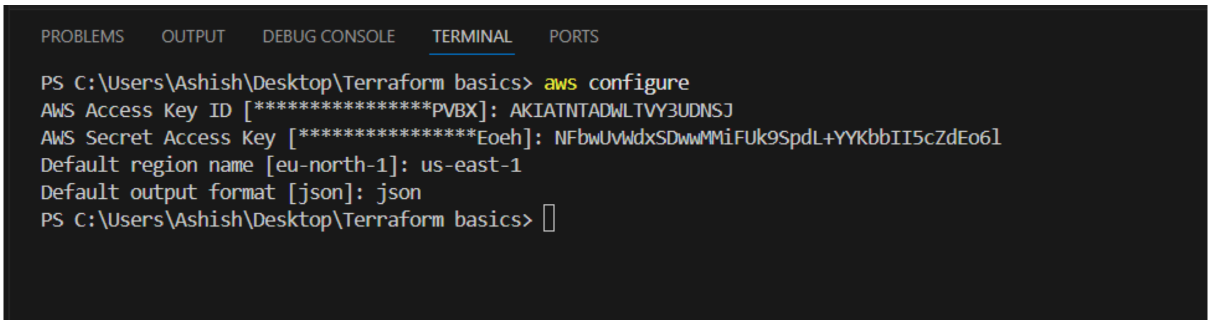
Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.

- aws configure

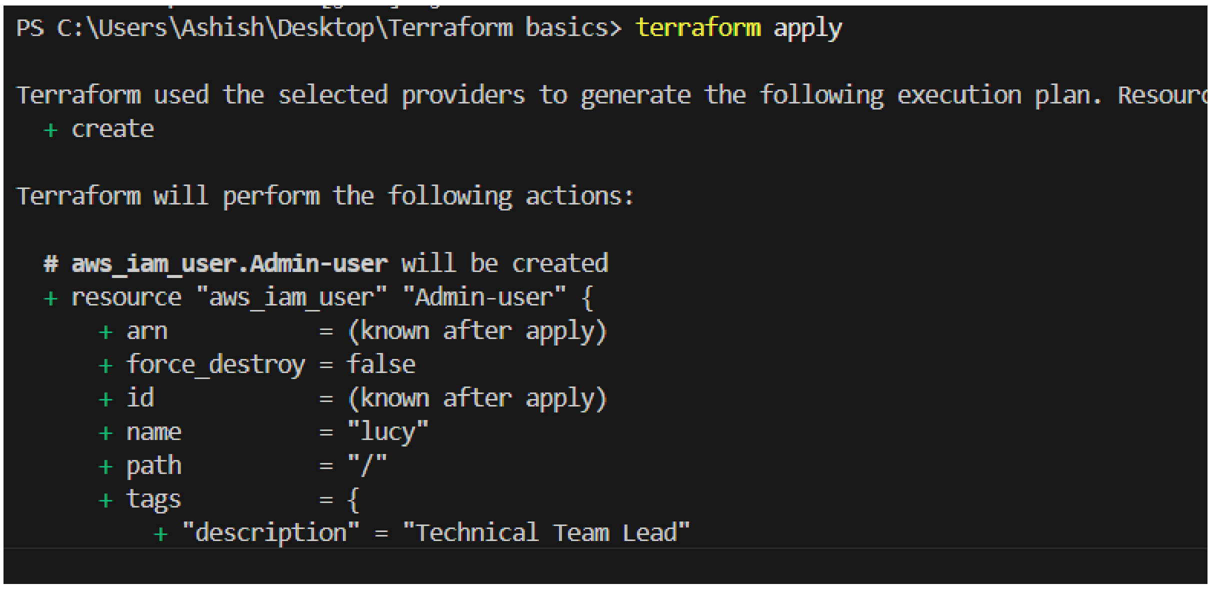


```
PS C:\Users\Ashish\Desktop\Terraform basics> aws configure
AWS Access Key ID [****************PVBX]: AKIATNTADWLTVY3UDNSJ
AWS Secret Access Key [****************Eoeh]: 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
Default region name [eu-north-1]: us-east-1
Default output format [json]: json
PS C:\Users\Ashish\Desktop\Terraform basics>
```

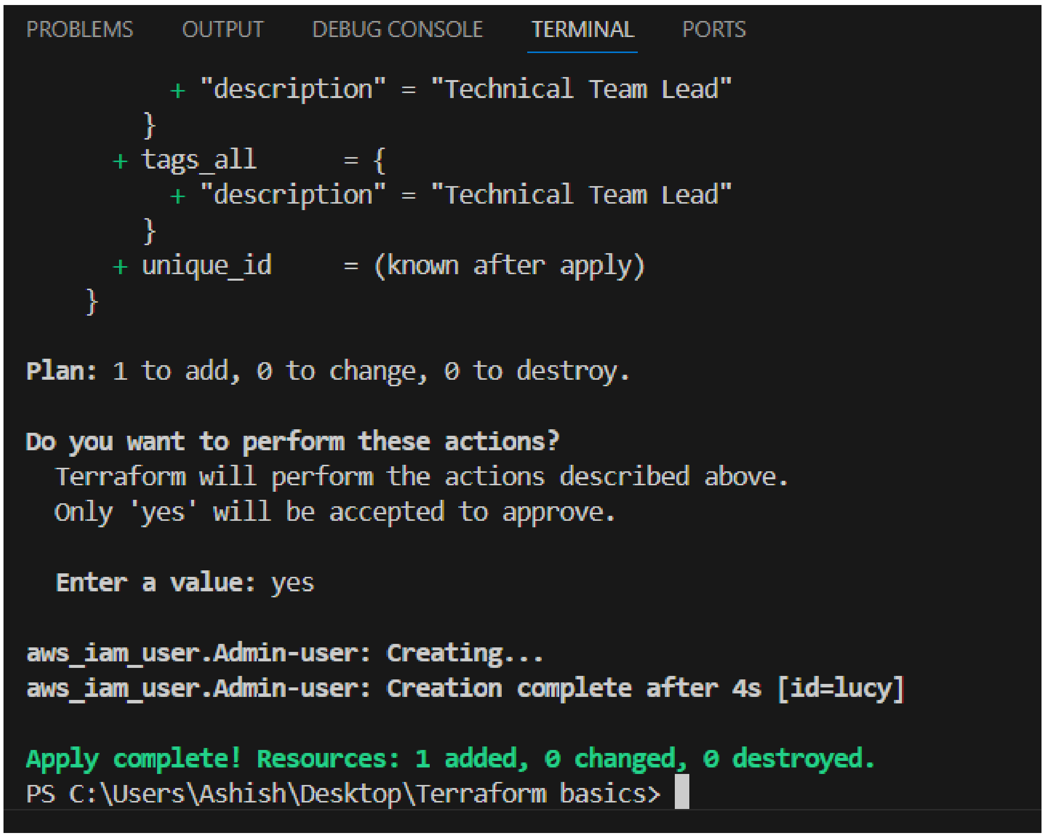


```
PS C:\Users\Ashish\Desktop\Terraform basics> aws configure
AWS Access Key ID [****************PVBX]: AKIATNTADWLTVY3UDNSJ
AWS Secret Access Key [****************Eoeh]: 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
Default region name [eu-north-1]: us-east-1
Default output format [json]: json
PS C:\Users\Ashish\Desktop\Terraform basics>
```

Then if you do terraform apply

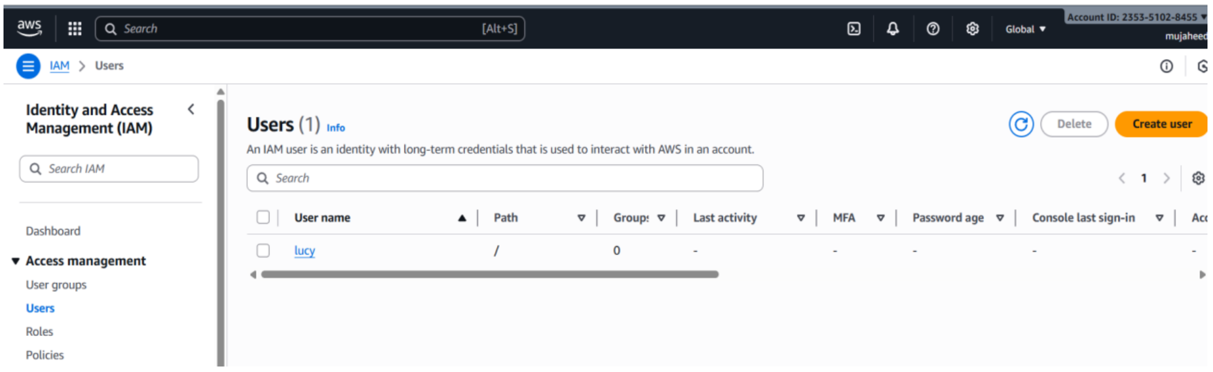


```
PS C:\Users\Ashish\Desktop\Terraform basics> terraform apply

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

Terraform will perform the following actions:

  # aws_iam_user.Admin-user will be created
  + resource "aws_iam_user" "Admin-user" {
      + arn           = (known after apply)
      + force_destroy = false
      + id            = (known after apply)
      + name          = "lucy"
      + path          = "/"
      + tags          = {
          + "description" = "Technical Team Lead"
```



```
          + "description" = "Technical Team Lead"
        }
      + tags_all      = {
          + "description" = "Technical Team Lead"
        }
      + unique_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

aws_iam_user.Admin-user: Creating...
aws_iam_user.Admin-user: Creation complete after 4s [id=lucy]

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

If you refresh your aws iam users page you will see a user has been created.



Users (1)

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

User name	Path	Groups	Last activity	MFA	Password age	Console last sign-in	Acc
lucy	/	0	-	-	-	-	-

- if you want to attach a admin policy to the user we can do that by this script.

resource "aws_iam_user" "Admin-user" {

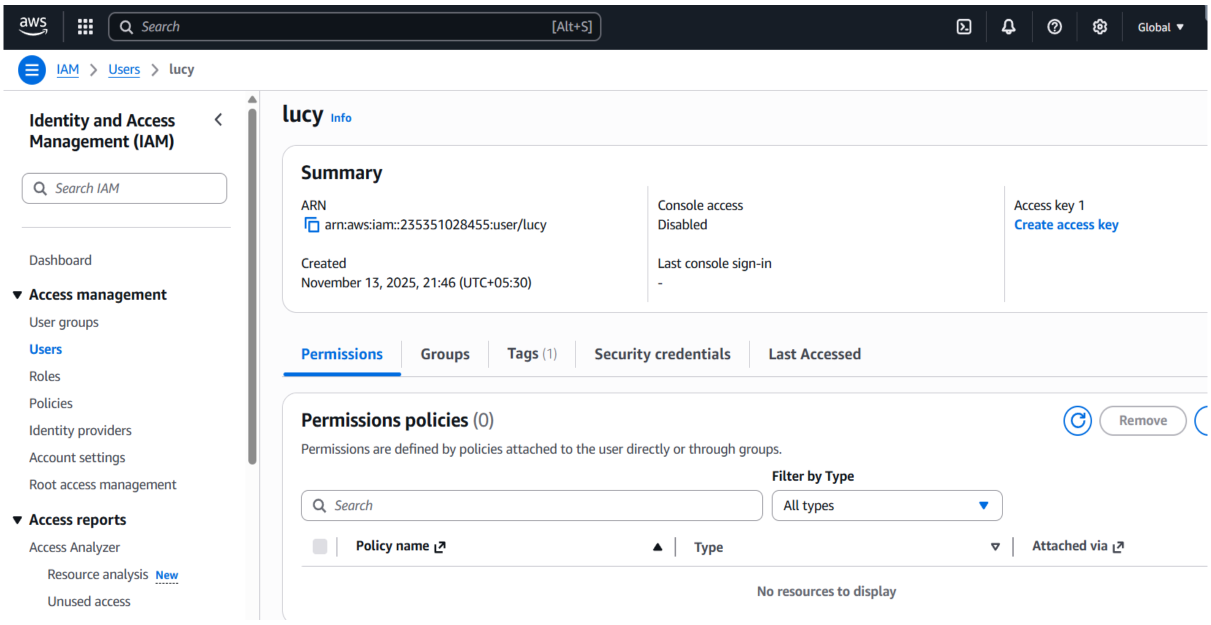
 name = "lucy"

 tags = {

```
    "description" = "Technical Team Lead"
  }
}
resource "aws_iam_policy" "adminuser" {
  name   = "AdminUsers"
  policy = <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "1234567890",
            "Effect": "Allow",
            "Action": "*",
            "Resource": "*"
        }
    ]
}
EOF
}
```

```
resource "aws_iam_user_policy_attachment" "lucy-admin-access" {
  user       = aws_iam_user.Admin-user.name
  policy_arn = aws_iam_policy.adminuser.arn
}
```

There is no policy attached to the user.



lucy

Summary

ARN
arn:aws:iam::235351028455:user/lucy

Created
November 13, 2025, 21:46 (UTC+05:30)

Console access
Disabled

Last console sign-in
-

Access key 1
Create access key

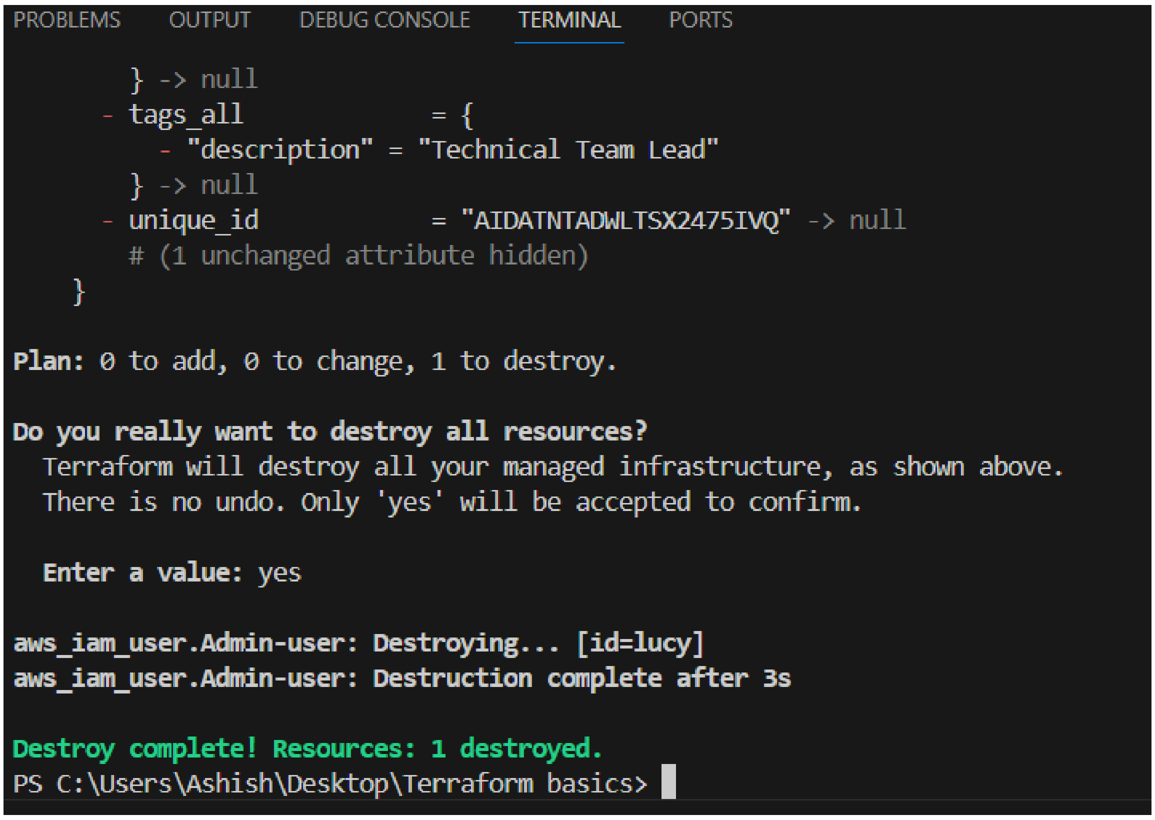
Permissions policies (0)

Permissions are defined by policies attached to the user directly or through groups.

Policy name	Type	Attached via

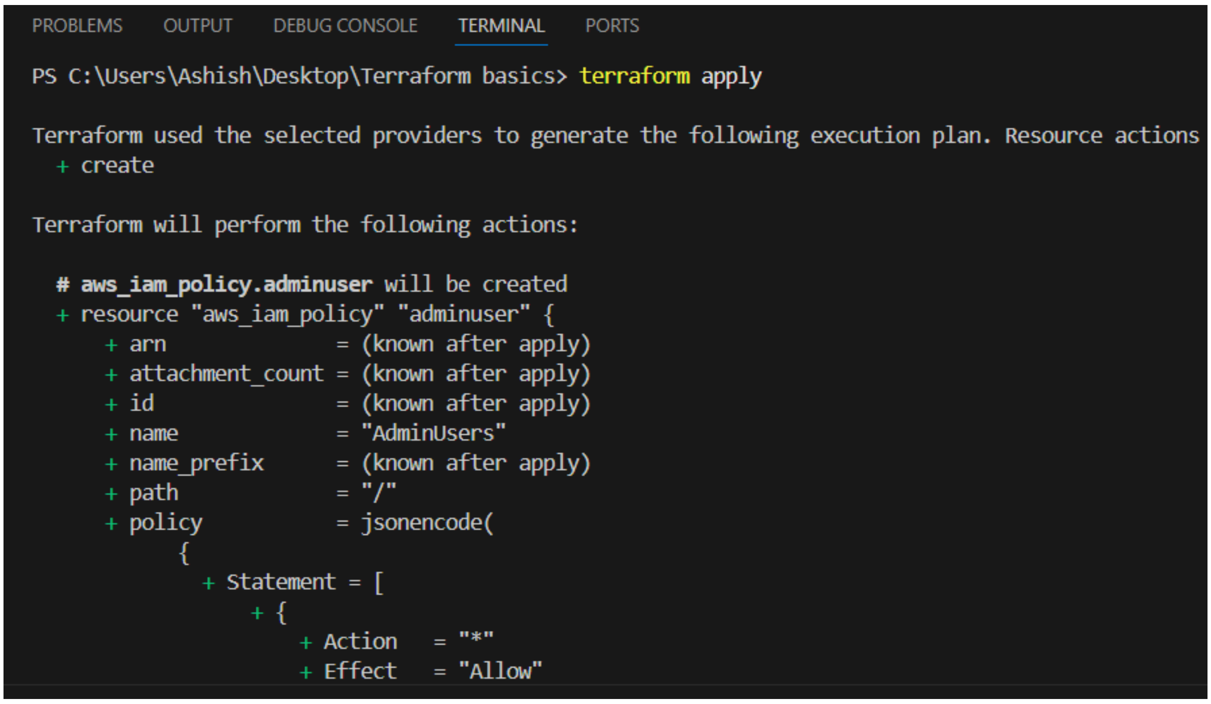
No resources to display

Do terraform destroy after that do terraform apply.



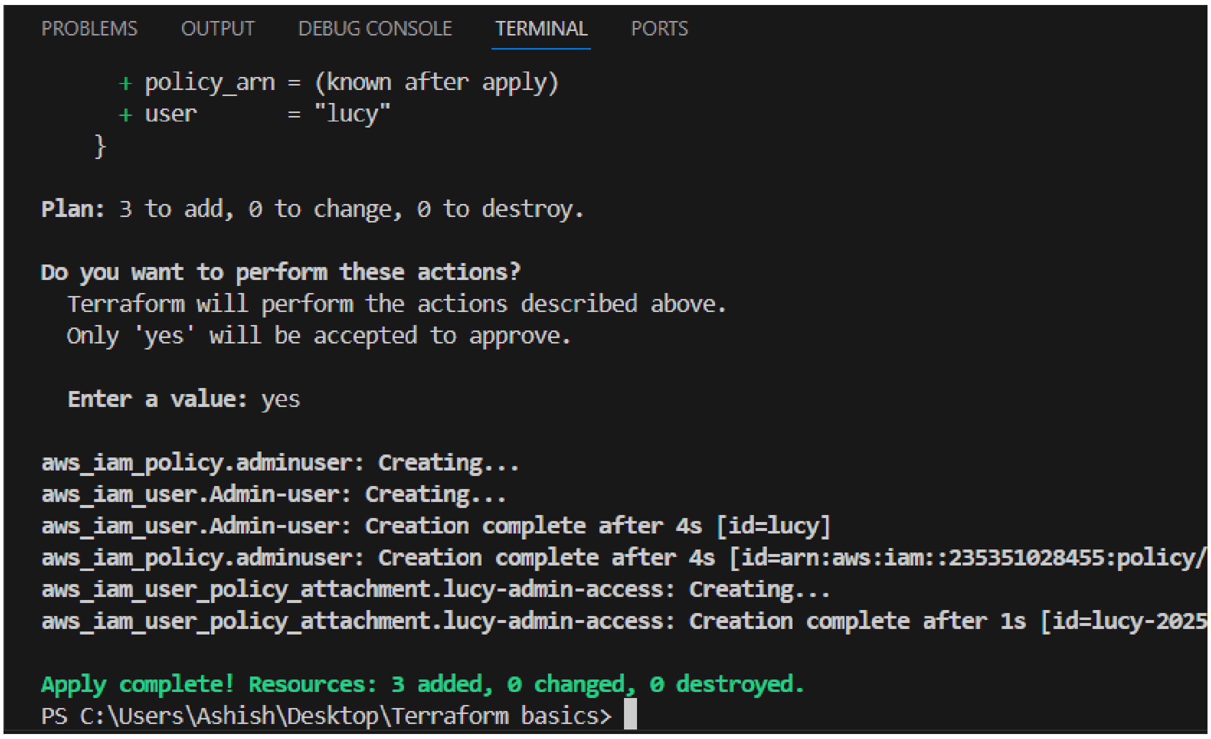
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
        } -> null
      - tags_all             = {
          - "description" = "Technical Team Lead"
        } -> null
      - unique_id            = "AIDATNTADWLTSX2475IVQ" -> 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_iam_user.Admin-user: Destroying... [id=lucy]
aws_iam_user.Admin-user: Destruction complete after 3s

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



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

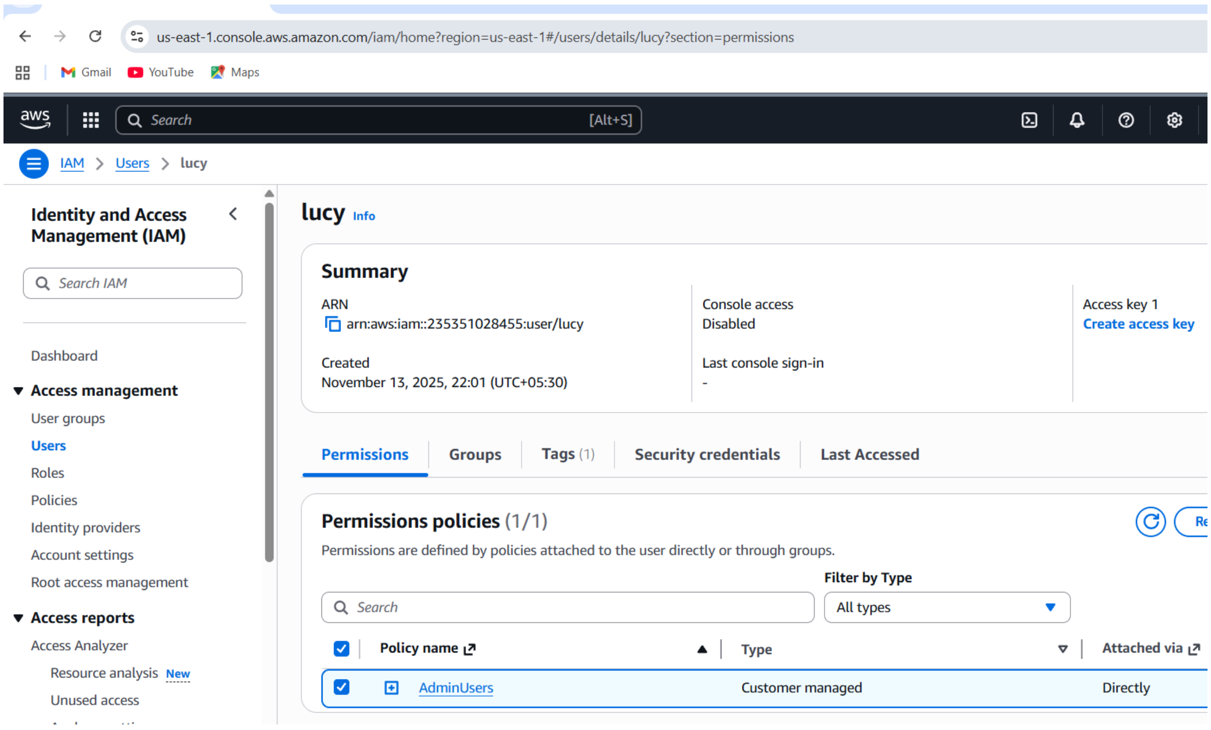
```
PS C:\Users\Ashish\Desktop\Terraform basics> terraform apply

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

Terraform will perform the following actions:

  # aws_iam_policy.adminuser will be created
  + resource "aws_iam_policy" "adminuser" {
      + arn              = (known after apply)
      + attachment_count = (known after apply)
      + id               = (known after apply)
      + name             = "AdminUsers"
      + name_prefix      = (known after apply)
      + path             = "/"
      + policy           = jsonencode(
            {
              + Statement = [
                  + {
                      + Action   = "*"
                      + Effect   = "Allow"
```



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
      + policy_arn = (known after apply)
      + user       = "lucy"
    }

Plan: 3 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_iam_policy.adminuser: Creating...
aws_iam_user.Admin-user: Creating...
aws_iam_user.Admin-user: Creation complete after 4s [id=lucy]
aws_iam_policy.adminuser: Creation complete after 4s [id=arn:aws:iam::235351028455:policy/
aws_iam_user_policy_attachment.lucy-admin-access: Creating...
aws_iam_user_policy_attachment.lucy-admin-access: Creation complete after 1s [id=lucy-2025

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
PS C:\Users\Ashish\Desktop\Terraform basics>
```

If you refresh this IAM user page a policy created to that lucy user.



us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/users/details/lucy?section=permissions

IAM > Users > lucy

lucy Info

Summary

ARN arn:aws:iam::235351028455:user/lucy	Console access Disabled	Access key 1 Create access key
Created November 13, 2025, 22:01 (UTC+05:30)	Last console sign-in -	

Permissions | Groups | Tags (1) | Security credentials | Last Accessed

Permissions policies (1/1)

Permissions are defined by policies attached to the user directly or through groups.

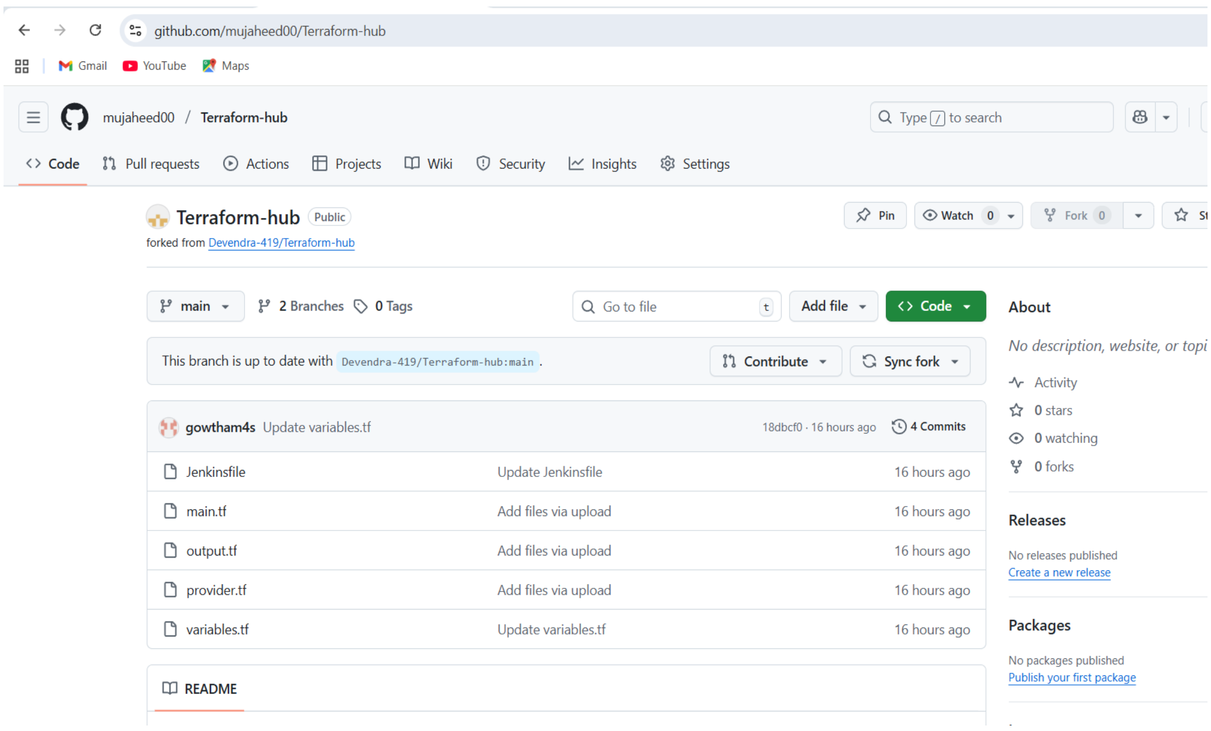
Filter by Type: All types

Policy name	Type	Attached via
AdminUsers	Customer managed	Directly

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



mujaheed00 / Terraform-hub

Terraform-hub Public
forked from Devendra-419/Terraform-hub

main · 2 Branches · 0 Tags

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

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

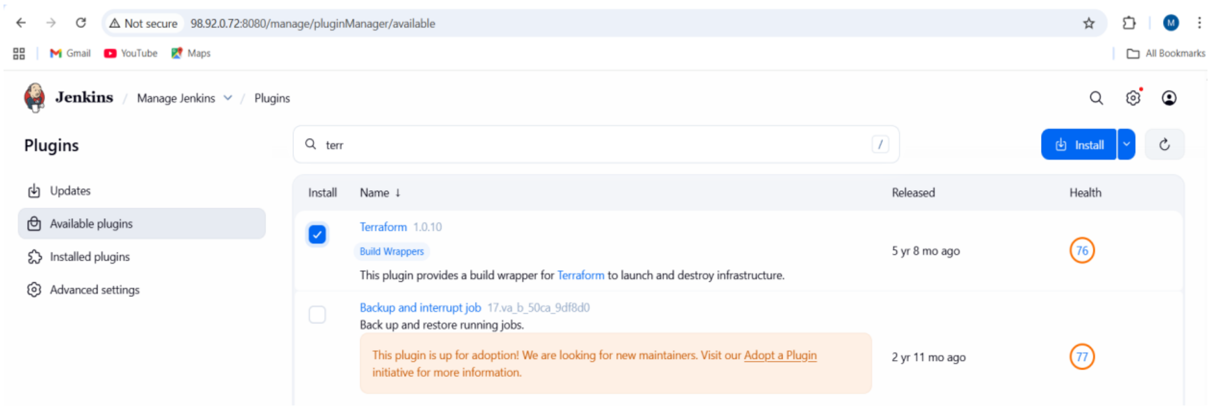
Jenkinsfile	Update Jenkinsfile	16 hours ago
main.tf	Add files via upload	16 hours ago
output.tf	Add files via upload	16 hours ago
provider.tf	Add files via upload	16 hours ago
variables.tf	Update variables.tf	16 hours ago

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.



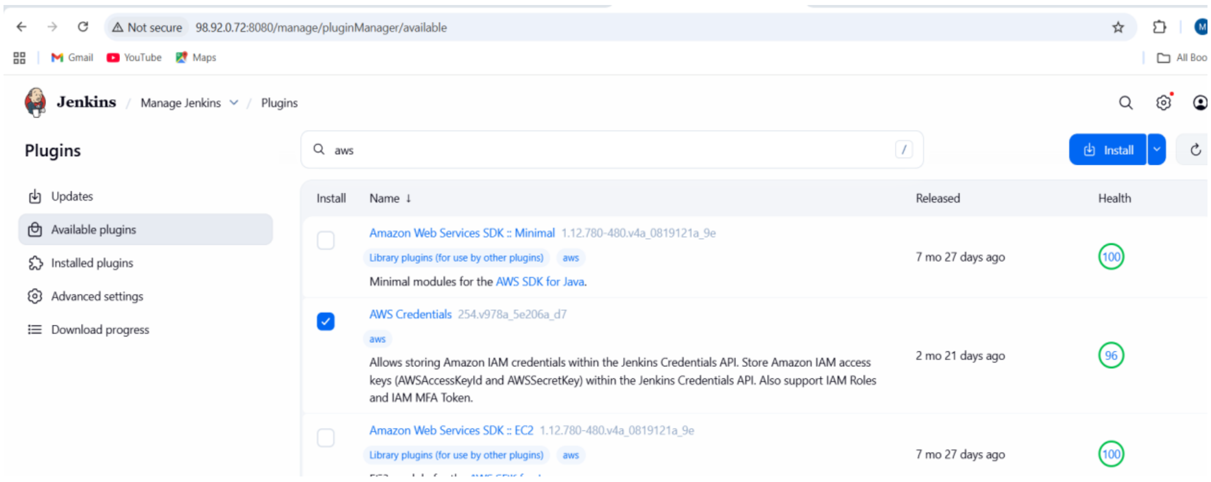
98.92.0.72:8080/manage/pluginManager/available

Jenkins / Manage Jenkins / Plugins

Plugins — search: 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

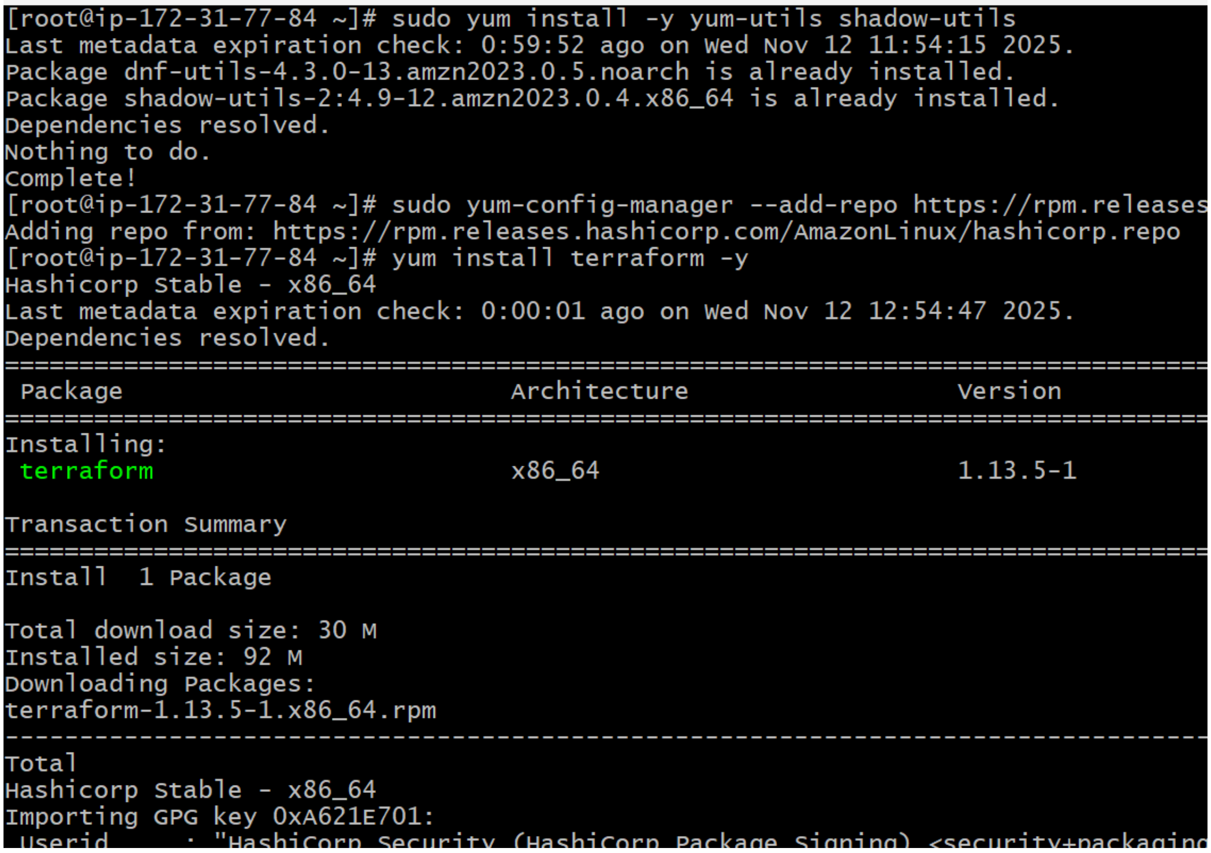
Install another plugin called aws credentials.



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

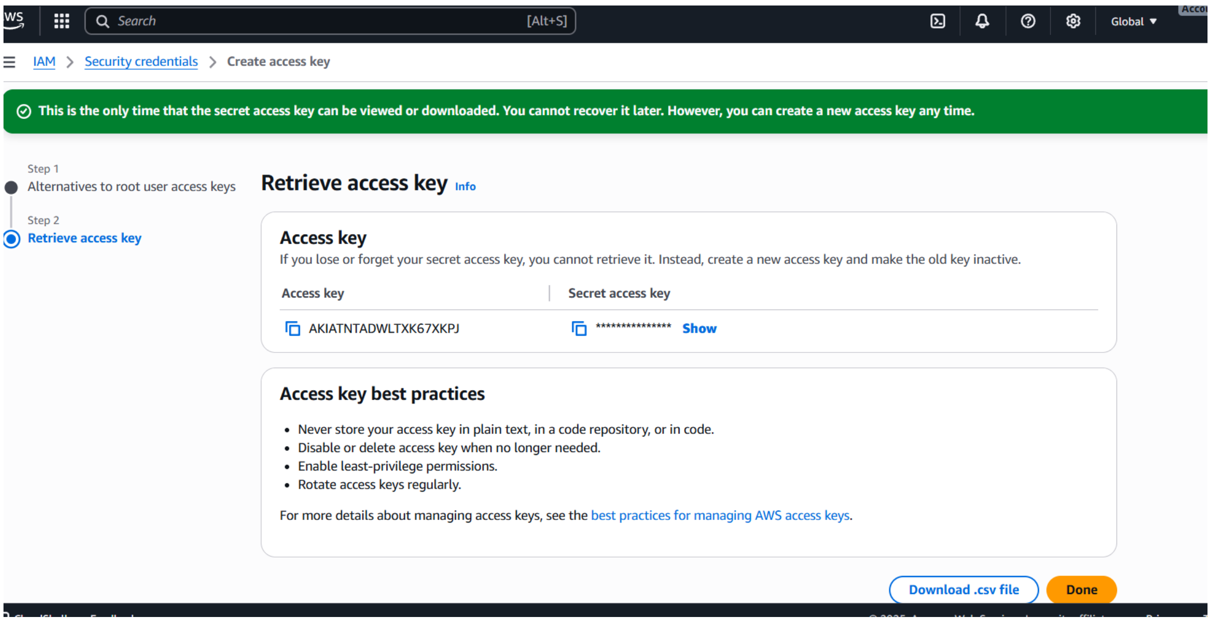
Go to Jenkins server and install terraform by using this commands.

- 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.

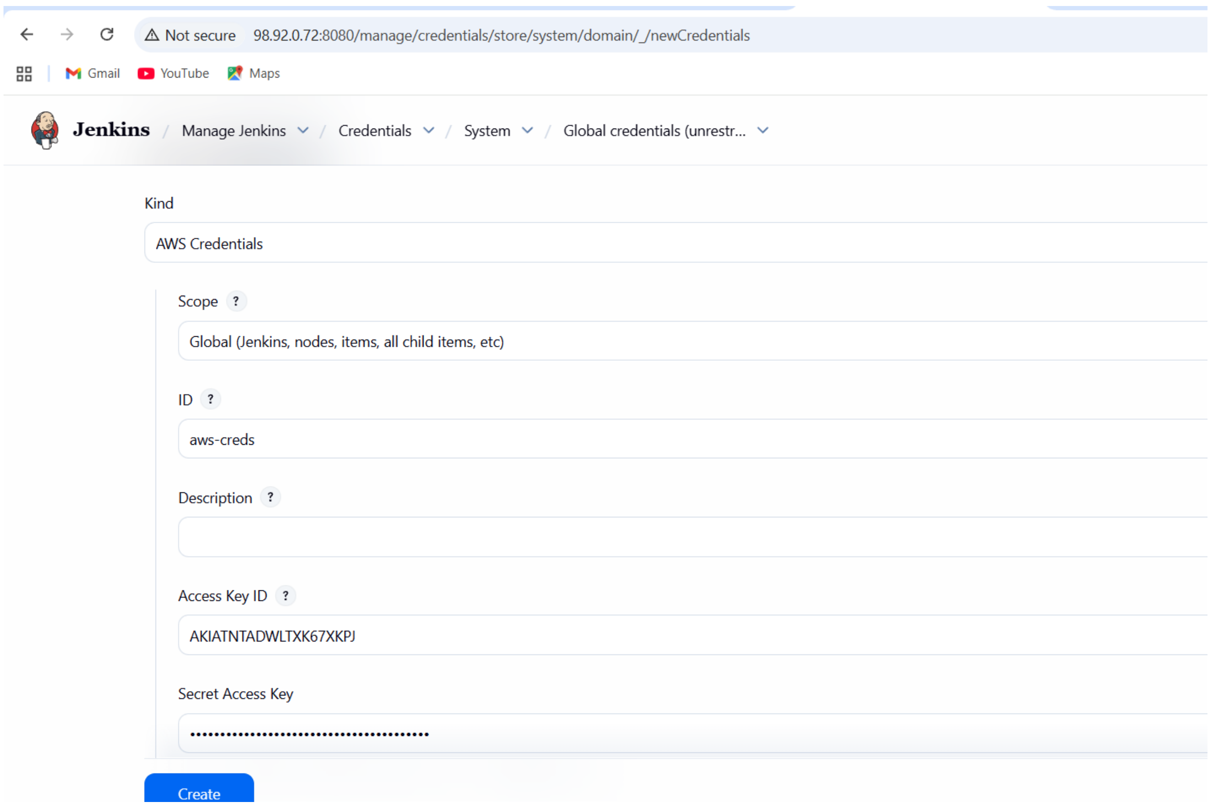
Access key	Secret access key
AKIATNTADWLTXK67XKPJ	*************** Show

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

Gmail YouTube Maps

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 ?

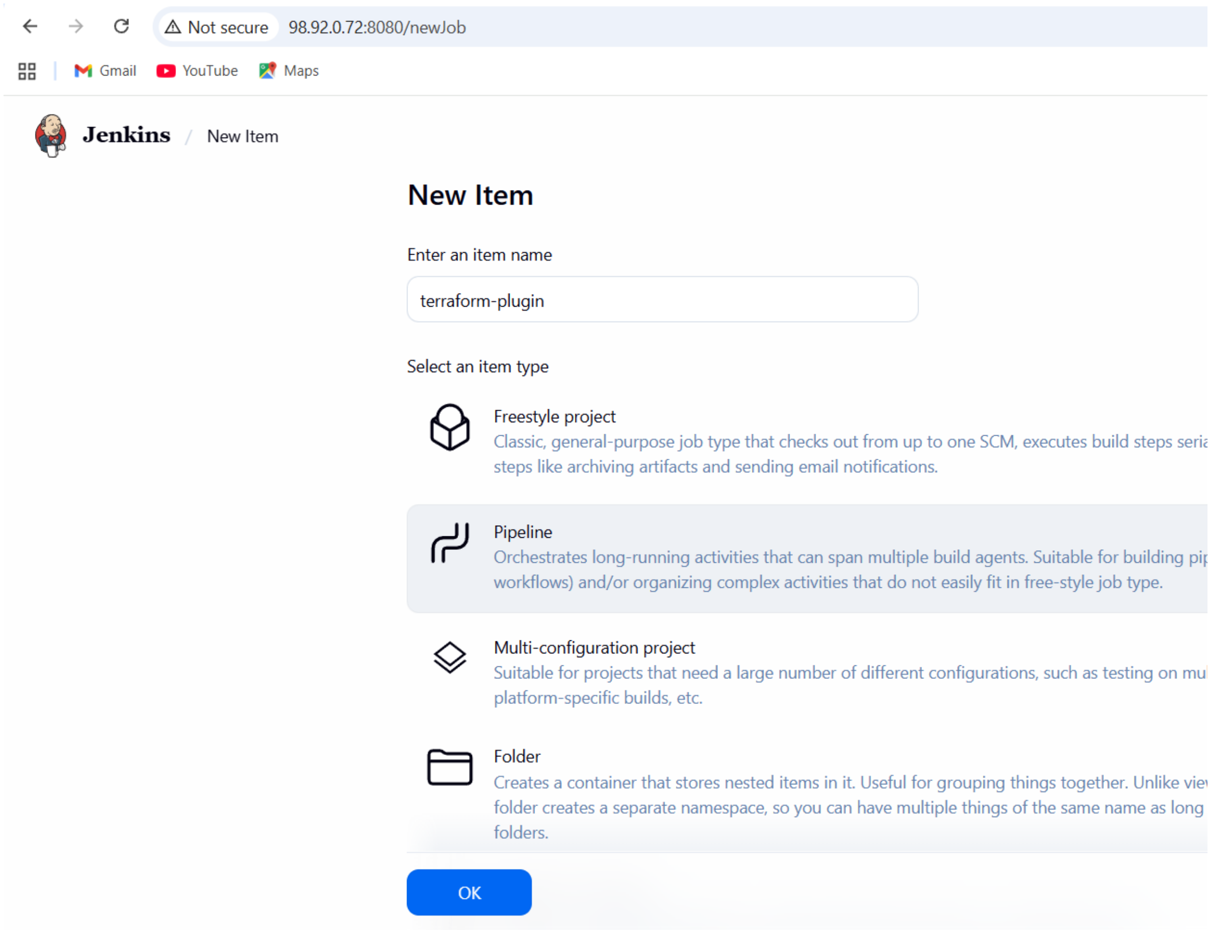
AKIATNTADWLTXK67XKPJ

Secret Access Key

••

Create

Create a new item and select pipeline.


Not secure 98.92.0.72:8080/newJob

Gmail YouTube Maps

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.

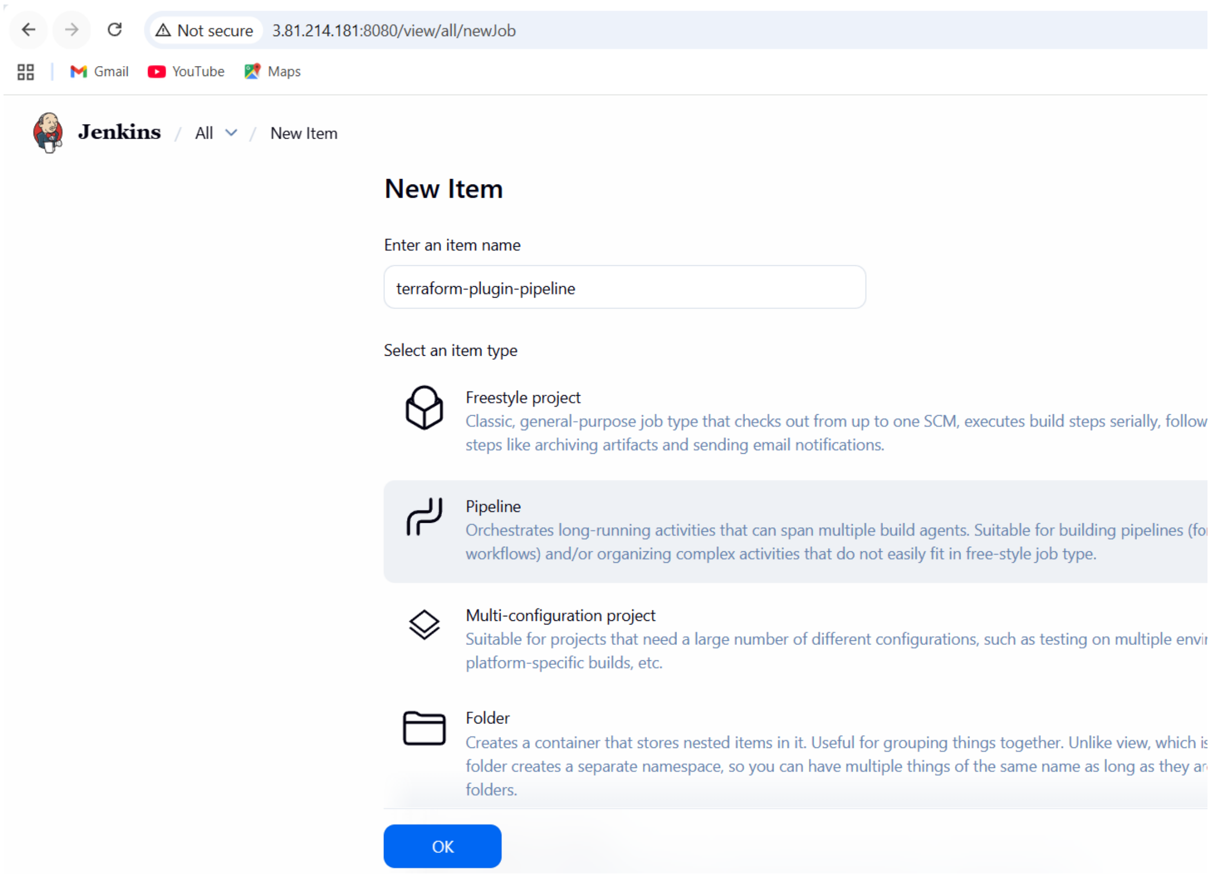
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pi
workflows) and/or organizing complex activities that do not easily fit in free-style job type.

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.

OK

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


Not secure 3.81.214.181:8080/view/all/newJob

Jenkins / All / New Item

New Item

Enter an item name

terraform-plugin-pipeline

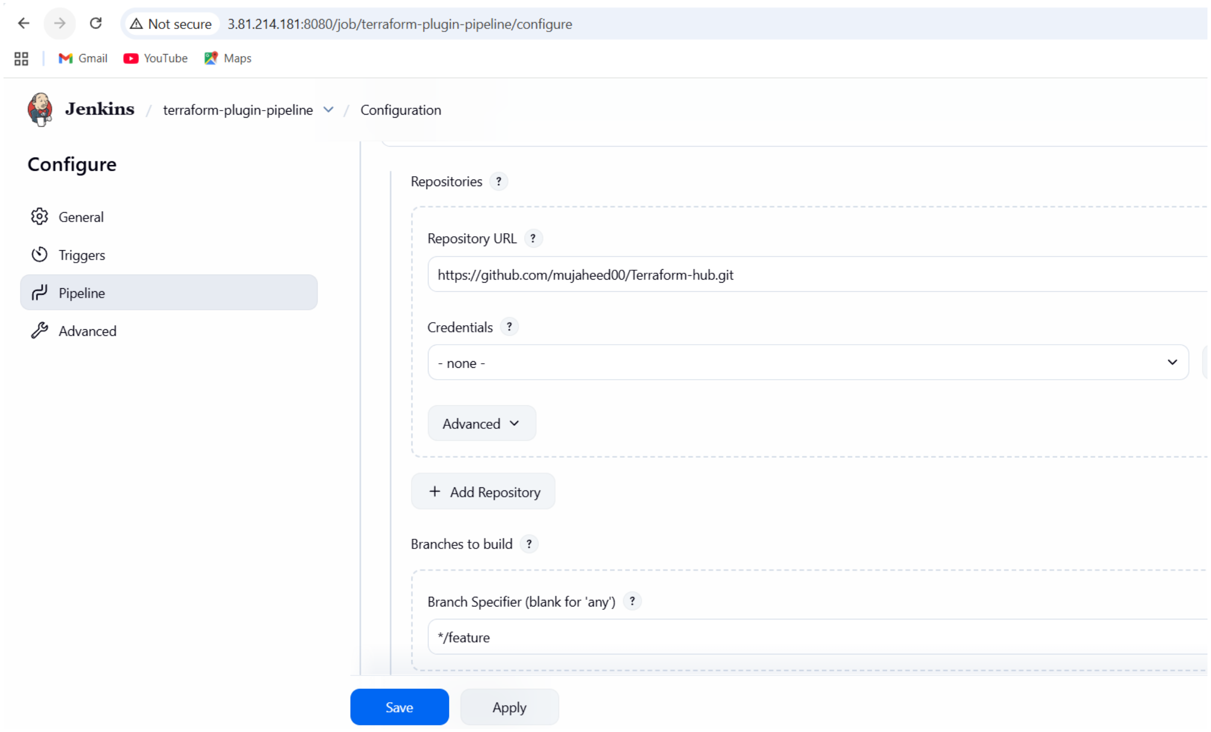
Select an item type

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 is
folder creates a separate namespace, so you can have multiple things of the same name as long as they a
folders.


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

Jenkins / terraform-plugin-pipeline / Configuration

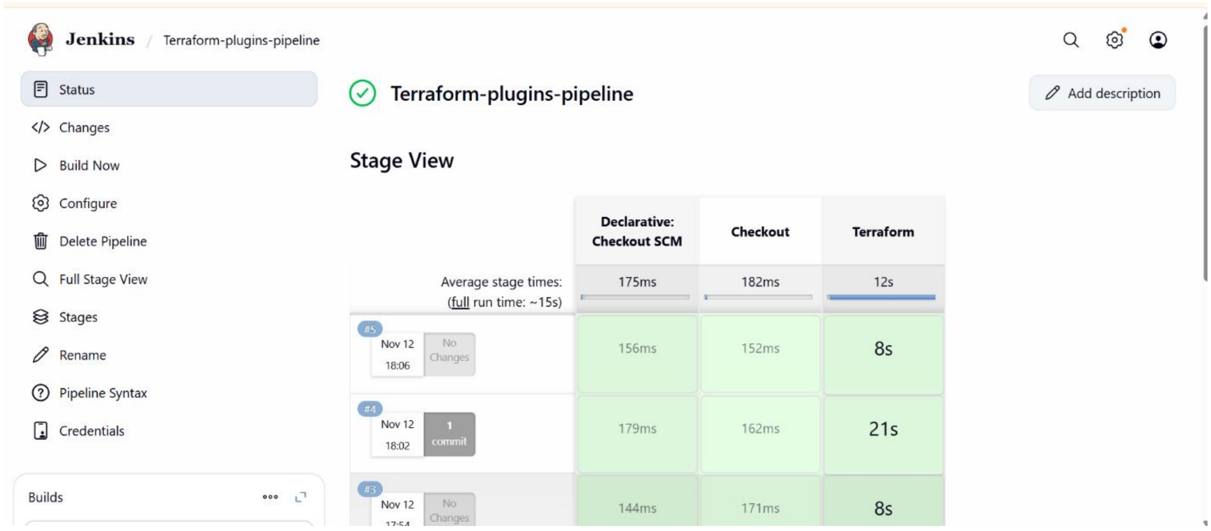
Repositories

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

Credentials
- none -

Branches to build

Branch Specifier (blank for 'any')
*/feature

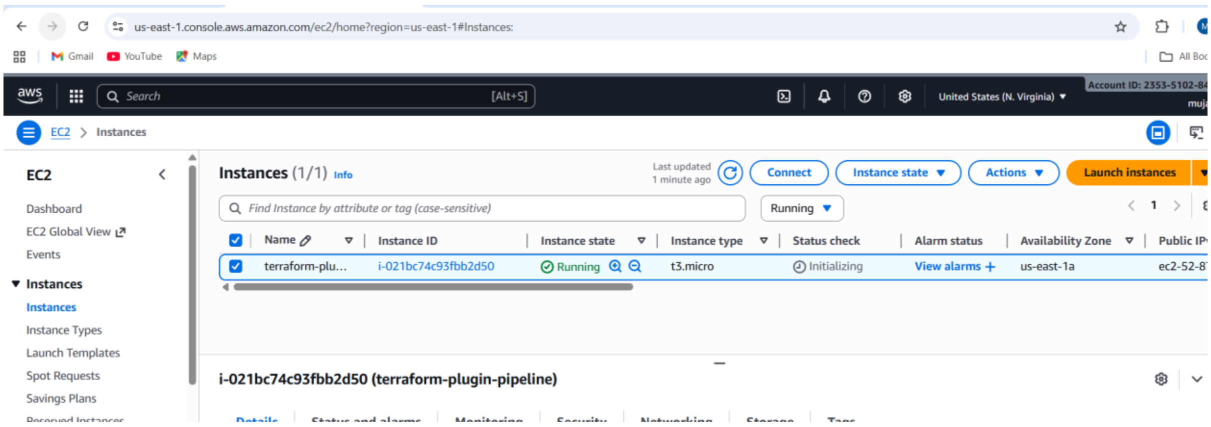


Terraform-plugins-pipeline

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.



Instances (1/1) Info

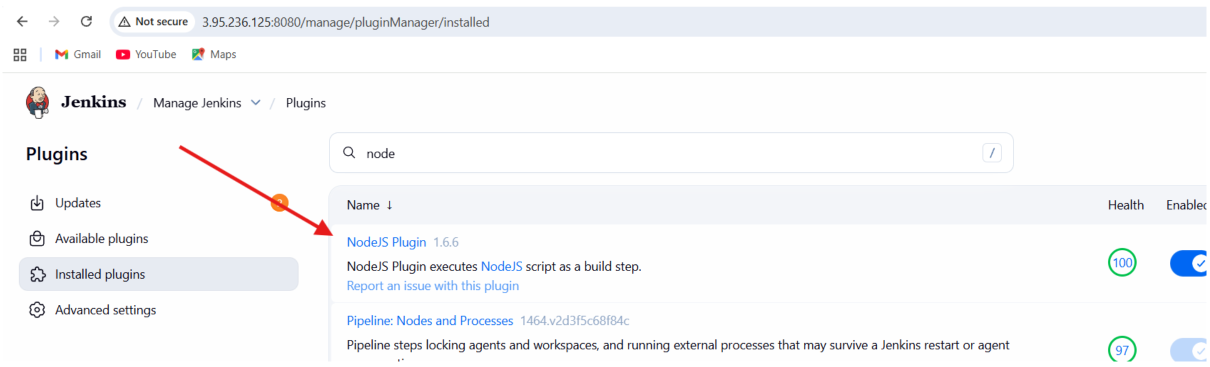
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
terraform-plu...	i-021bc74c93fbb2d50	Running	t3.micro	Initializing	View alarms +	us-east-1a	ec2-52-8

i-021bc74c93fbb2d50 (terraform-plugin-pipeline)

4. Create a CI/CD pipeline for a Nodejs Application: https://github.com/betawins/Trading-UI.git

* https://github.com/mujaheed00/Trading-UI.git

Install nodejs plugin.


Not secure 3.95.236.125:8080/manage/pluginManager/installed

Jenkins / Manage Jenkins / Plugins

Plugins

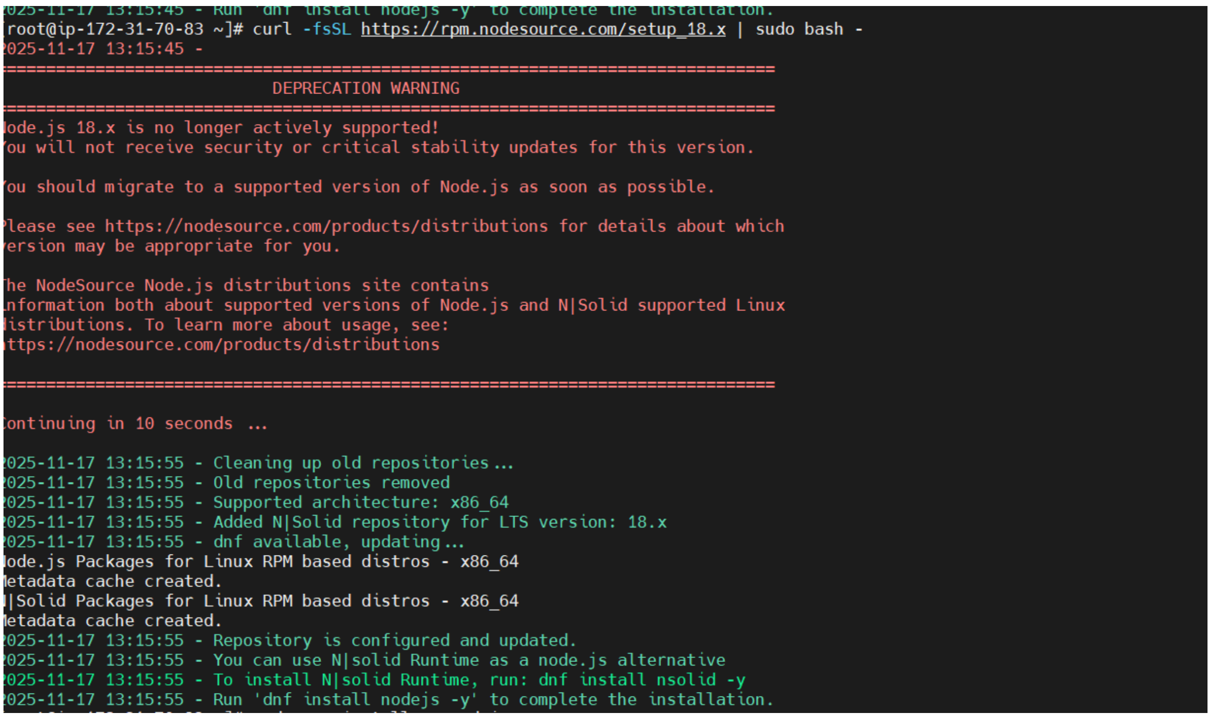
- Updates
- Available plugins
- Installed plugins
- Advanced settings

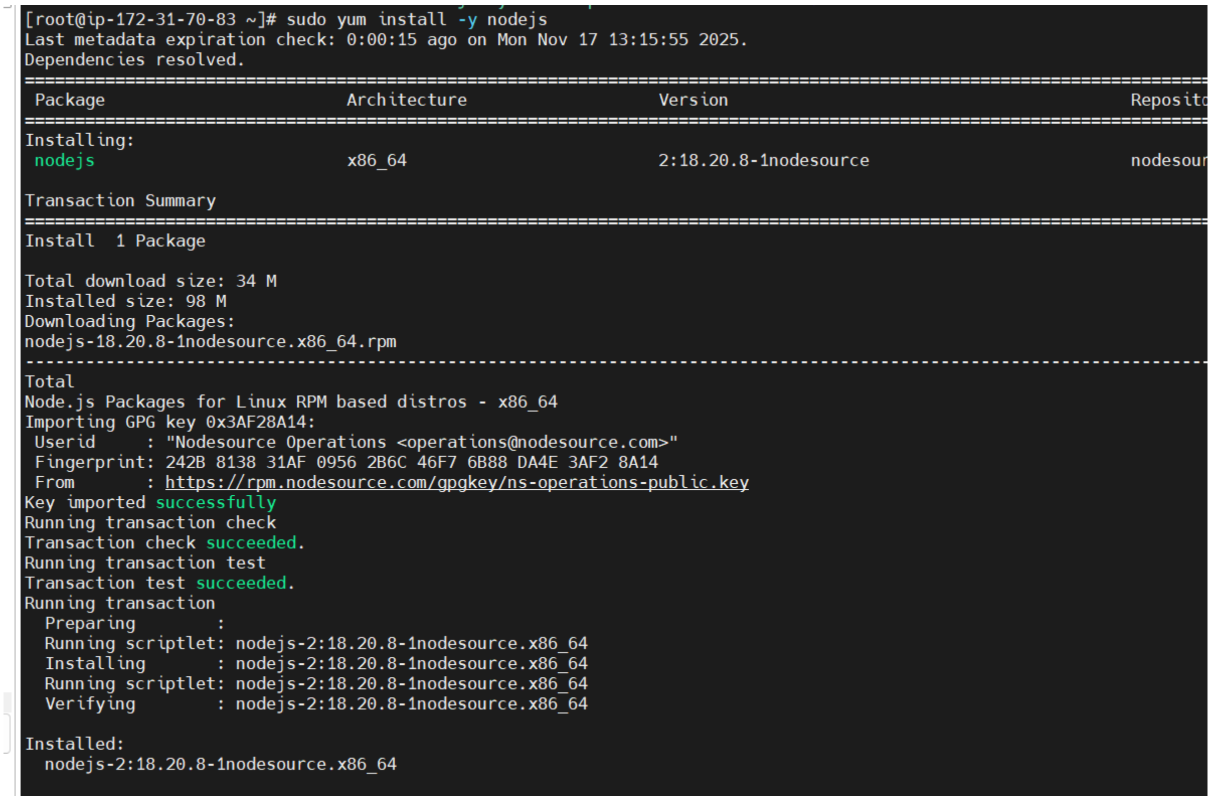
node

Name ↓	Health	Enabled
NodeJS Plugin 1.6.6 NodeJS Plugin executes NodeJS script as a build step. Report an issue with this plugin	100	✓
Pipeline: Nodes and Processes 1464.v2d3f5c68f84c Pipeline steps locking agents and workspaces, and running external processes that may survive a Jenkins restart or agent	97	✓

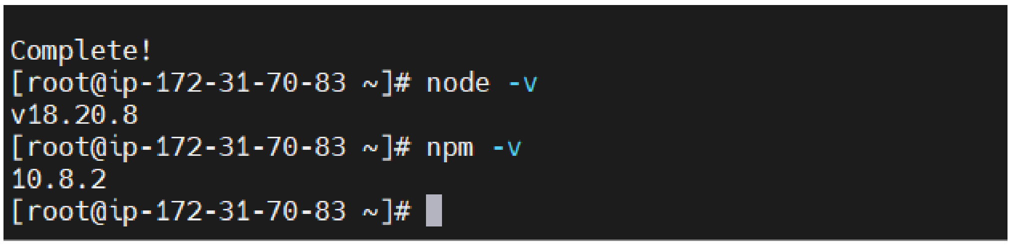
Intall nodejs and npm in Jenkins server.

- curl -fsSL https://rpm.nodesource.com/setup_18.x | sudo bash -
- sudo yum install -y nodejs
- node -v
- npm -v


```
025-11-17 13:15:45 - Run 'dnf install nodejs -y' to complete the installation.
root@ip-172-31-70-83 ~]# curl -fsSL https://rpm.nodesource.com/setup_18.x | sudo bash -
025-11-17 13:15:45 -
===============================================================================
                           DEPRECATION WARNING
===============================================================================
ode.js 18.x is no longer actively supported!
ou will not receive security or critical stability updates for this version.

ou should migrate to a supported version of Node.js as soon as possible.

lease see https://nodesource.com/products/distributions for details about which
ersion may be appropriate for you.

he NodeSource Node.js distributions site contains
nformation both about supported versions of Node.js and N|Solid supported Linux
istributions. To learn more about usage, see:
ttps://nodesource.com/products/distributions

===============================================================================

ontinuing in 10 seconds ...

025-11-17 13:15:55 - Cleaning up old repositories...
025-11-17 13:15:55 - Old repositories removed
025-11-17 13:15:55 - Supported architecture: x86_64
025-11-17 13:15:55 - Added N|Solid repository for LTS version: 18.x
025-11-17 13:15:55 - dnf available, updating...
ode.js Packages for Linux RPM based distros - x86_64
etadata cache created.
|Solid Packages for Linux RPM based distros - x86_64
etadata cache created.
025-11-17 13:15:55 - Repository is configured and updated.
025-11-17 13:15:55 - You can use N|solid Runtime as a node.js alternative
025-11-17 13:15:55 - To install N|solid Runtime, run: dnf install nsolid -y
025-11-17 13:15:55 - Run 'dnf install nodejs -y' to complete the installation.
```

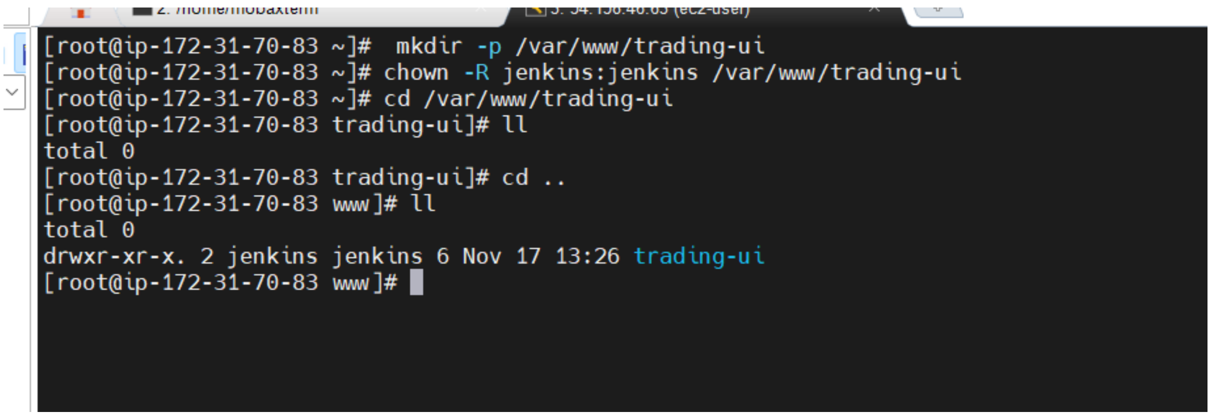


```
[root@ip-172-31-70-83 ~]# sudo yum install -y nodejs
Last metadata expiration check: 0:00:15 ago on Mon Nov 17 13:15:55 2025.
Dependencies resolved.
=====================================================================================================================
 Package                        Architecture                  Version                                       Reposito
=====================================================================================================================
Installing:
 nodejs                         x86_64                        2:18.20.8-1nodesource                         nodesour

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

Total download size: 34 M
Installed size: 98 M
Downloading Packages:
nodejs-18.20.8-1nodesource.x86_64.rpm
---------------------------------------------------------------------------------------------------------------------
Total
Node.js Packages for Linux RPM based distros - x86_64
Importing GPG key 0x3AF28A14:
 Userid     : "Nodesource Operations <operations@nodesource.com>"
 Fingerprint: 242B 8138 31AF 0956 2B6C 46F7 6B88 DA4E 3AF2 8A14
 From       : https://rpm.nodesource.com/gpgkey/ns-operations-public.key
Key imported successfully
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :
  Running scriptlet: nodejs-2:18.20.8-1nodesource.x86_64
  Installing       : nodejs-2:18.20.8-1nodesource.x86_64
  Running scriptlet: nodejs-2:18.20.8-1nodesource.x86_64
  Verifying        : nodejs-2:18.20.8-1nodesource.x86_64

Installed:
  nodejs-2:18.20.8-1nodesource.x86_64
```



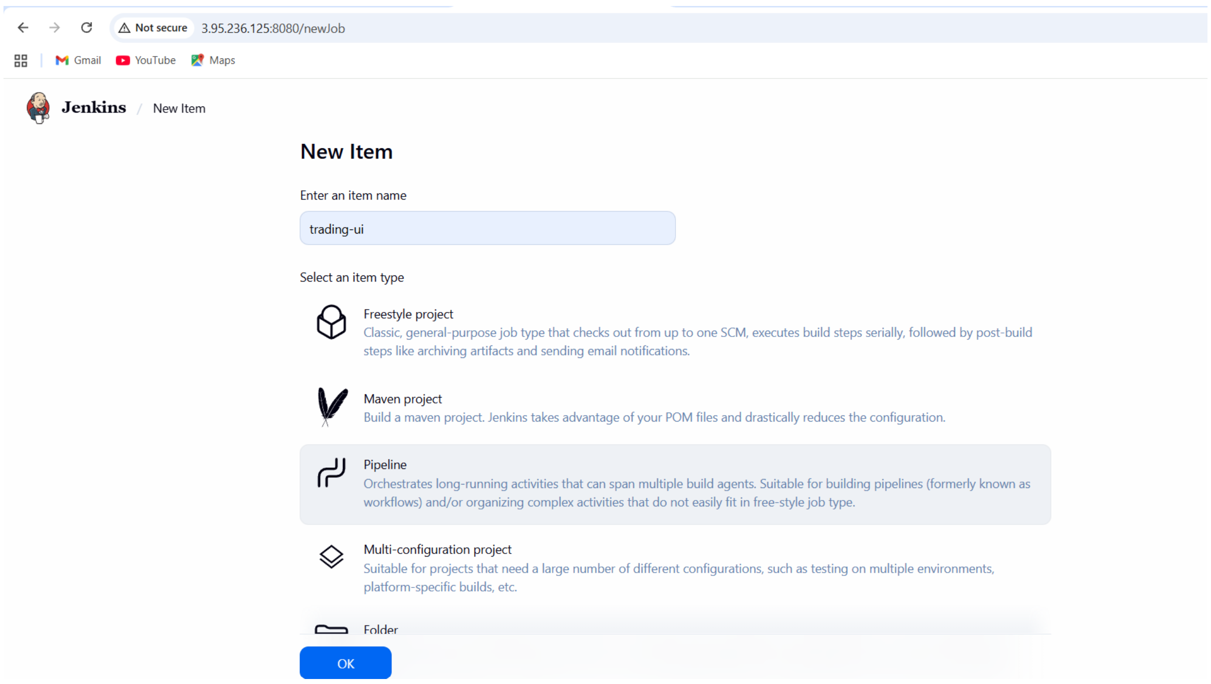
```
Complete!
[root@ip-172-31-70-83 ~]# node -v
v18.20.8
[root@ip-172-31-70-83 ~]# npm -v
10.8.2
[root@ip-172-31-70-83 ~]#
```

- mkdir -p /var/www/trading-ui
- chown -R jenkins:jenkins /var/www/trading-ui



```
[root@ip-172-31-70-83 ~]#  mkdir -p /var/www/trading-ui
[root@ip-172-31-70-83 ~]# chown -R jenkins:jenkins /var/www/trading-ui
[root@ip-172-31-70-83 ~]# cd /var/www/trading-ui
[root@ip-172-31-70-83 trading-ui]# ll
total 0
[root@ip-172-31-70-83 trading-ui]# cd ..
[root@ip-172-31-70-83 www]# ll
total 0
drwxr-xr-x. 2 jenkins jenkins 6 Nov 17 13:26 trading-ui
[root@ip-172-31-70-83 www]#
```

Go to Jenkins click on new item.



Not secure 3.95.236.125:8080/newJob

Jenkins / New Item

New Item

Enter an item name

trading-ui

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.

Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

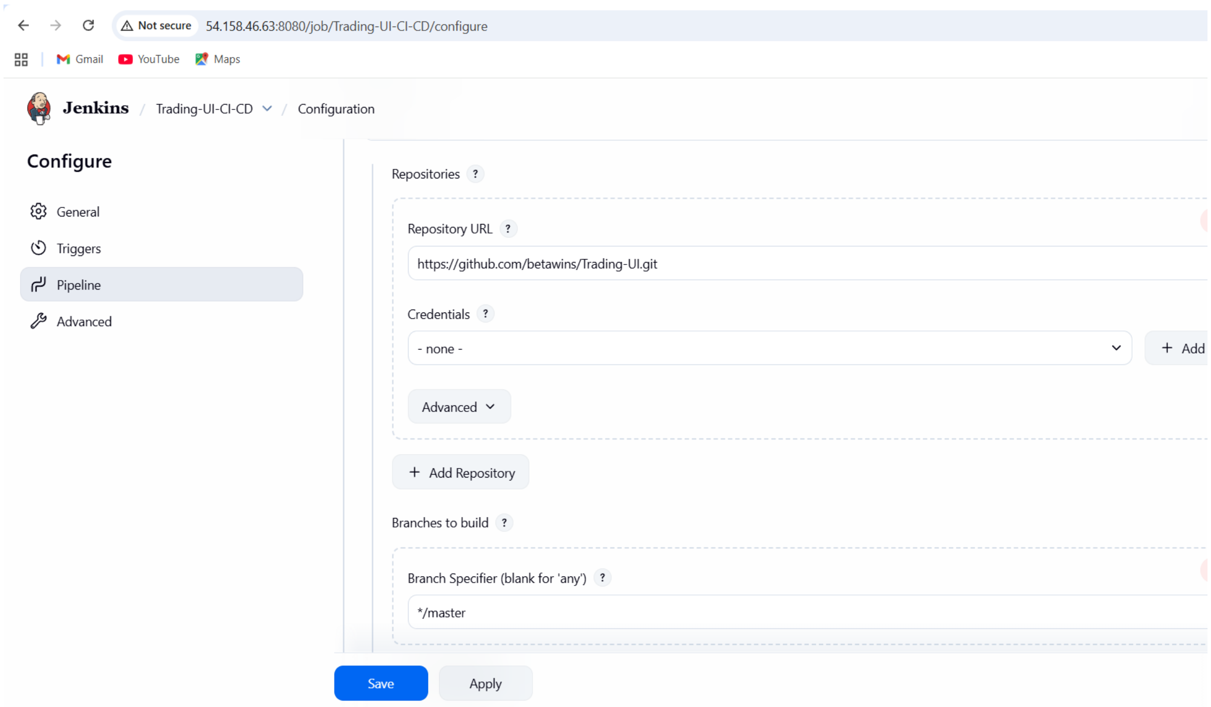
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

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

OK

Go to pipeline script scm give your git URL and branch as master.



Not secure 54.158.46.63:8080/job/Trading-UI-CI-CD/configure

Jenkins / Trading-UI-CI-CD / Configuration

Configure

General
Triggers
Pipeline
Advanced

Repositories

Repository URL

https://github.com/betawins/Trading-UI.git

Credentials

- none -

Advanced

Add Repository

Branches to build

Branch Specifier (blank for 'any')

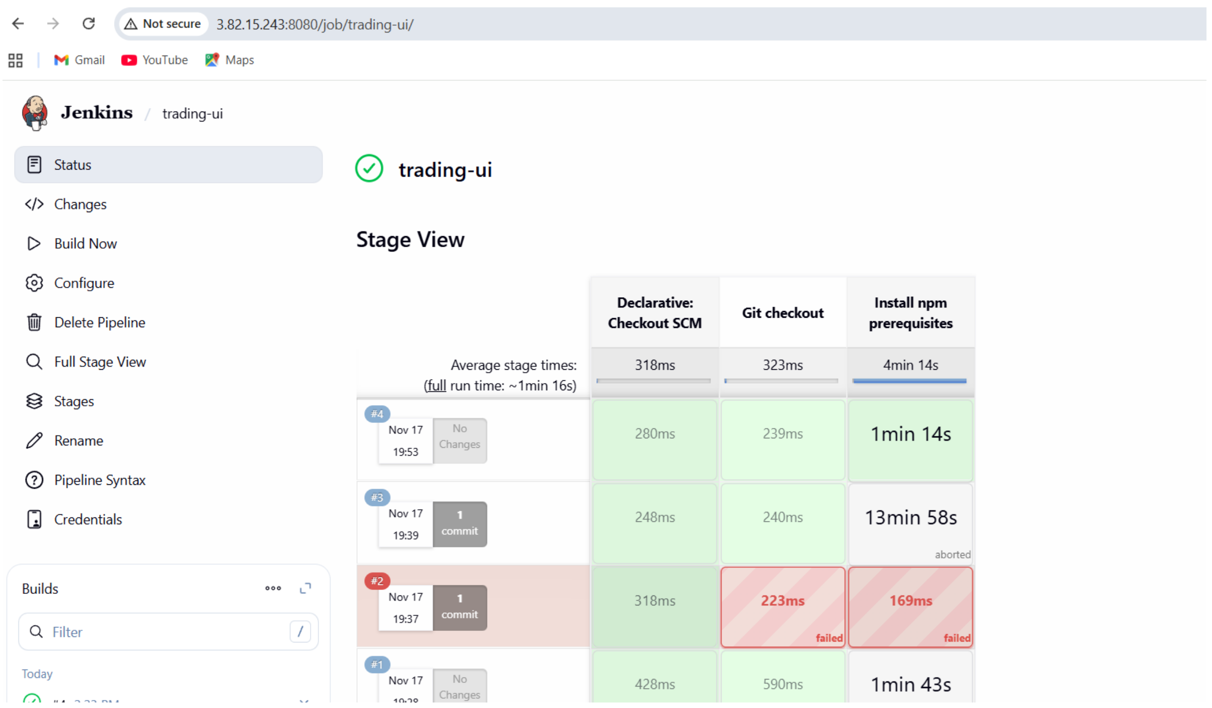
*/master

Save Apply

```
pipeline {
    agent any
    stages {
        stage('Git checkout') {
            steps {
                git 'https://github.com/mujaheed00/Trading-UI.git'
            }
        }

        stage('Install npm prerequisites') {
            steps {
                // Skip audit completely
                sh 'npm install --no-audit || true'

                // Show build logs even on failure
                sh 'npm run build || true'

                // Move into build folder
                dir('build') {
                    // PM2 often returns exit code 1 if already
running
                    sh 'pm2 --name Trading-UI start npm -- start ||
true'
```

```
                }
            }
        }
    }
}
```

Click on build now.



	Declarative: Checkout SCM	Git checkout	Install npm prerequisites
Average stage times: (full run time: ~1min 16s)	318ms	323ms	4min 14s
#4 Nov 17 19:53 No Changes	280ms	239ms	1min 14s
#3 Nov 17 19:39 1 commit	248ms	240ms	13min 58s aborted
#2 Nov 17 19:37 1 commit	318ms	223ms failed	169ms failed
#1 Nov 17 No Changes	428ms	590ms	1min 43s

5. Explain 10 Maven commands.

1.mvn clean:

Deletes the target/ directory (where compiled files, artifacts, and temporary files are stored).

2. mvn compile:

Compiles the source code under src/main/java.

- Converts .java files → .class files
- Stores output in target/classes

3. mvn test:

Runs all test cases in src/test/java.

- Executes unit tests
- Creates test reports under target/surefire-reports

4. mvn package:

Compiles code + runs tests + packages into a **JAR** or **WAR** file.

- Output file will be inside target/ directory

5. mvn install:

Installs the JAR/WAR into the **local Maven repository** (~/.m2/repository).

- Makes the artifact available for other local projects
- Used during CI/CD builds.

6. mvn deploy:

Deploys the packaged artifact to a **remote repository** (like Nexus/Artifactory).

- Uploads to remote Maven repository
- Used in enterprise CI/CD pipelines

7.mvn clean install:

Most commonly used in Jenkins pipelines.

- Clean → Compile → Test → Package → Install

- Ensures complete build from scratch.

8. mvn clean package:

Cleans the project and creates a fresh JAR/WAR.

- Used before deploying to tomcat,docker etc.

9. mvn dependency:tree

Displays all dependencies of the project.

- Helps find version conflicts
- Shows transitive dependencies
- Useful for debugging issues

10. mvn --version:

Shows Maven version and Java version.

- Quick check of environment
- Confirms correct JDK is installed.