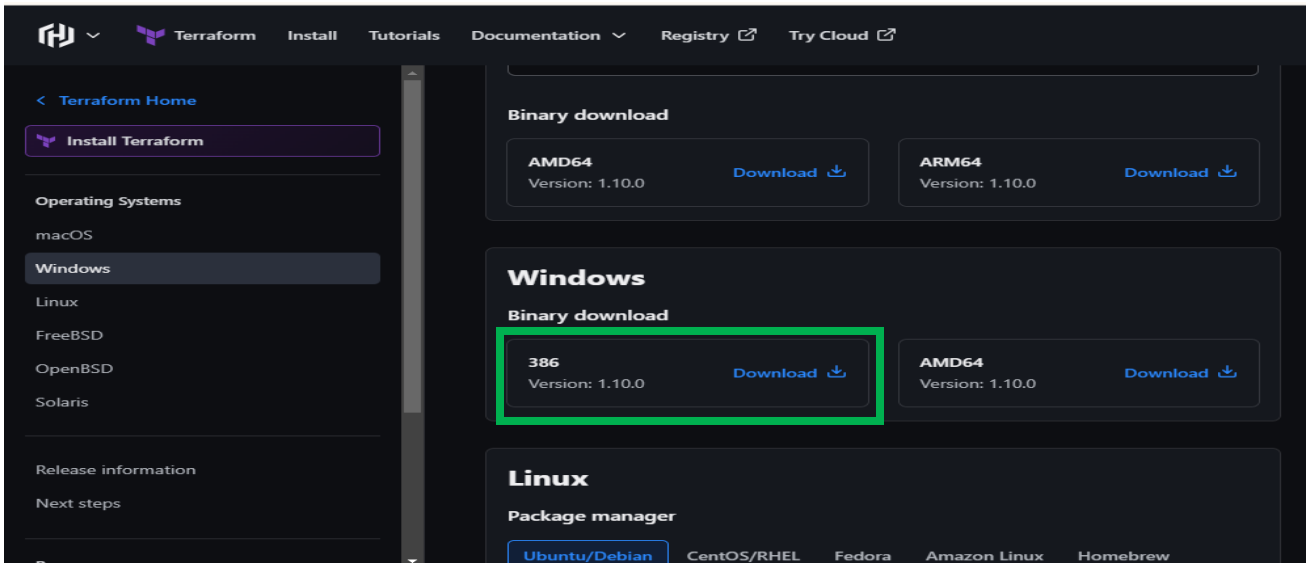
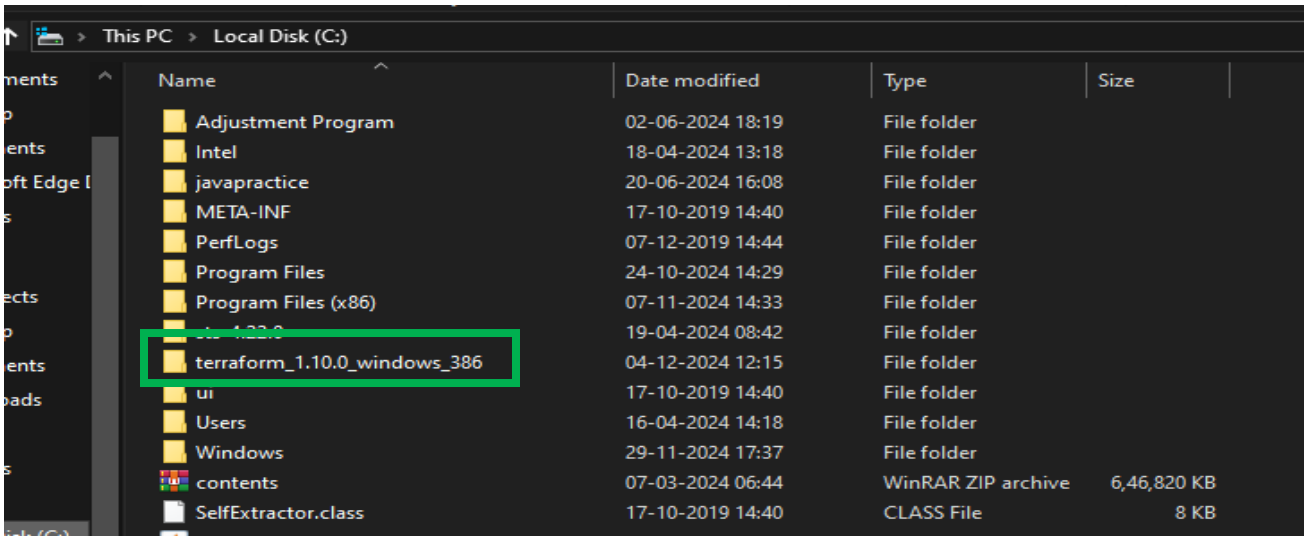
Terraform

1) Install Terraform on your PC

- Steps to Install Terraform on Your PC
  - Download Terraform
    Visit the https://developer.hashicorp.com/terraform/install

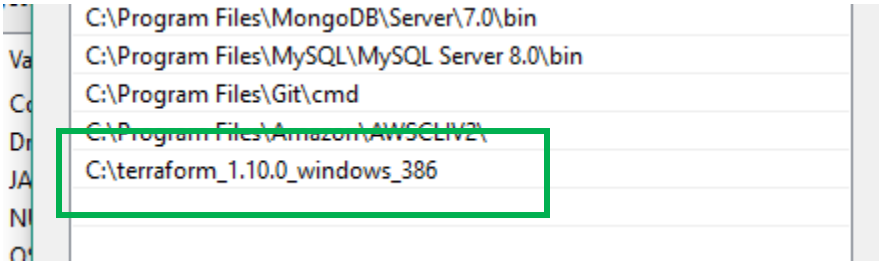


  - Extract the Executable



  - Add Terraform to Your System PATH

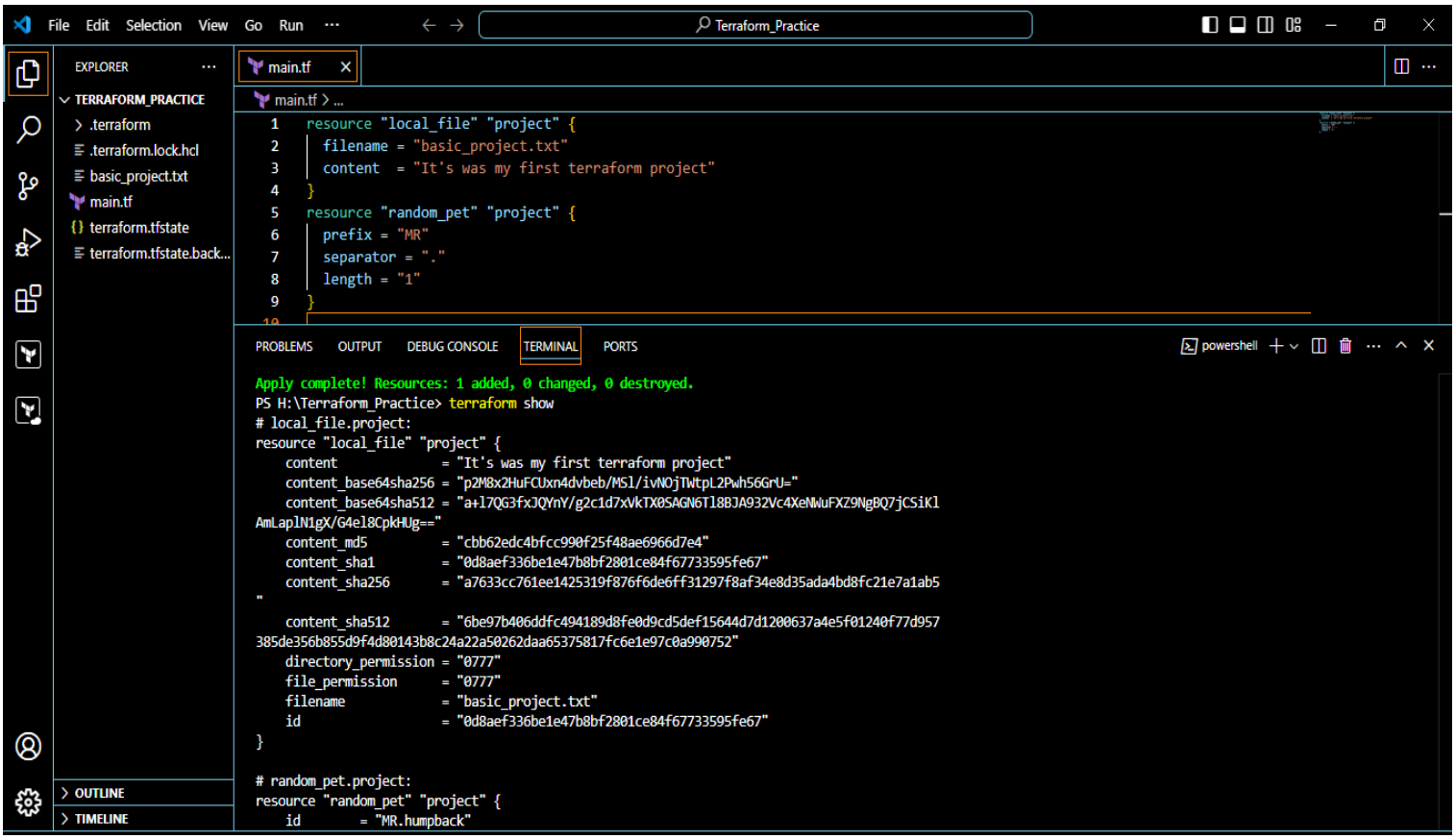
    Go to → Checkout **Edit the system environment variables** and set the path



- Verify the Installation

```
PS C:\Users\DELL> terraform -version
Terraform v1.10.0
on windows_386
PS C:\Users\DELL>
```

2) Execute all the templates shown in video.



3) Note down below points,

Terraform Init

```
main.tf
main.tf > ...
1  resource "local_file" "project" {
2  |  filename = "basic_project.txt"
3  |  content  = "It's was my first terraform project"
4  }
5  resource "random_pet" "project" {
6  |  prefix = "MR"
7  |  separator = "."
8  |  length = "1"
9  }

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS H:\Terraform_Practice> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/local...
- Installing hashicorp/local v2.5.2...
- Installed hashicorp/local v2.5.2 (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.


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.

rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

Terraform Plan

```
main.tf
main.tf > ...
1  resource "local_file" "project" {
2  |  filename = "basic_project.txt"
3  |  content  = "It's was my first terraform project"
4  }
5  resource "random_pet" "project" {
6  |  prefix = "MR"
7  |  separator = "."
8  |  length = "1"
9  }

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS H:\Terraform_Practice> terraform plan

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

Terraform will perform the following actions:

  # local_file.project will be created
  + resource "local_file" "project" {
      + content              = "It's was my first terraform project"
      + 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             = "basic_project.txt"
    }

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

Terraform Apply

```
main.tf
main.tf > ...
1  resource "local_file" "project" {
2  |  filename = "basic_project.txt"
3  |  content  = "It's was my first terraform project"
4  }
5  resource "random_pet" "project" {
6  |  prefix = "MR"
7  |  separator = "."
8  |  length = "1"
9  }

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS H:\Terraform_Practice> terraform apply
local_file.project: Refreshing state... [id=0d8aef336be1e47b8bf2801ce84f67733595fe67]

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

Terraform will perform the following actions:

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

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

Do you want to perform these actions?
  Only 'yes' will be accepted to approve.

  Enter a value: yes
```

Terraform Provider.

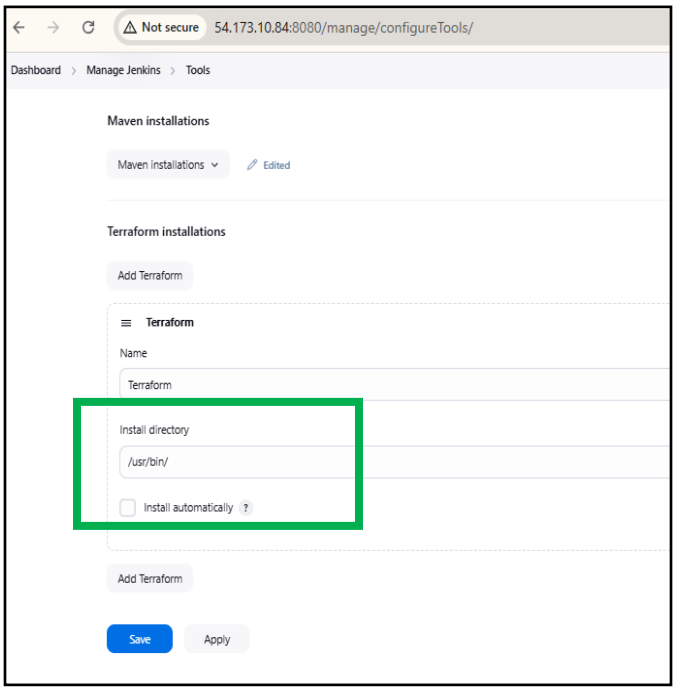
```
main.tf > ...
1  resource "local_file" "project" {
2  |  filename = "basic_project.txt"
3  |  content  = "It's was my first terraform project"
4  }
5  resource "random_pet" "project" {
6  |  prefix = "MR"
7  |  separator = "."
8  |  length = "1"
9  }

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

}

# random_pet.project:
resource "random_pet" "project" {
    id        = "MR.humpback"
    length    = 1
    prefix    = "MR"
    separator = "."
}
PS H:\Terraform_Practice> terraform providers

Providers required by configuration:
.
├── provider[registry.terraform.io/hashicorp/local]
└── provider[registry.terraform.io/hashicorp/random]

Providers required by state:

    provider[registry.terraform.io/hashicorp/local]

    provider[registry.terraform.io/hashicorp/random]
```

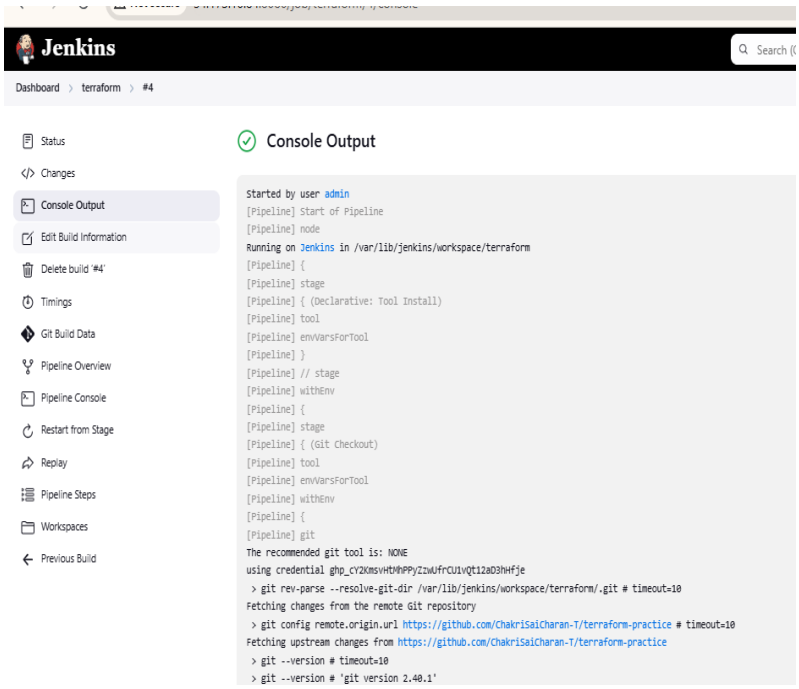
4) Integrate a sample Terraform template in jenkins.

CLI

```
306  sudo yum install -y yum-utils shadow-utils
307  sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo
308  sudo yum -y install terraform
309  which terraform
310  find / -name terraform
311  terraform -version
312  cd /usr/bin/terraform
313  cd /usr/bin/
314  ls
315  cd /bin/
```

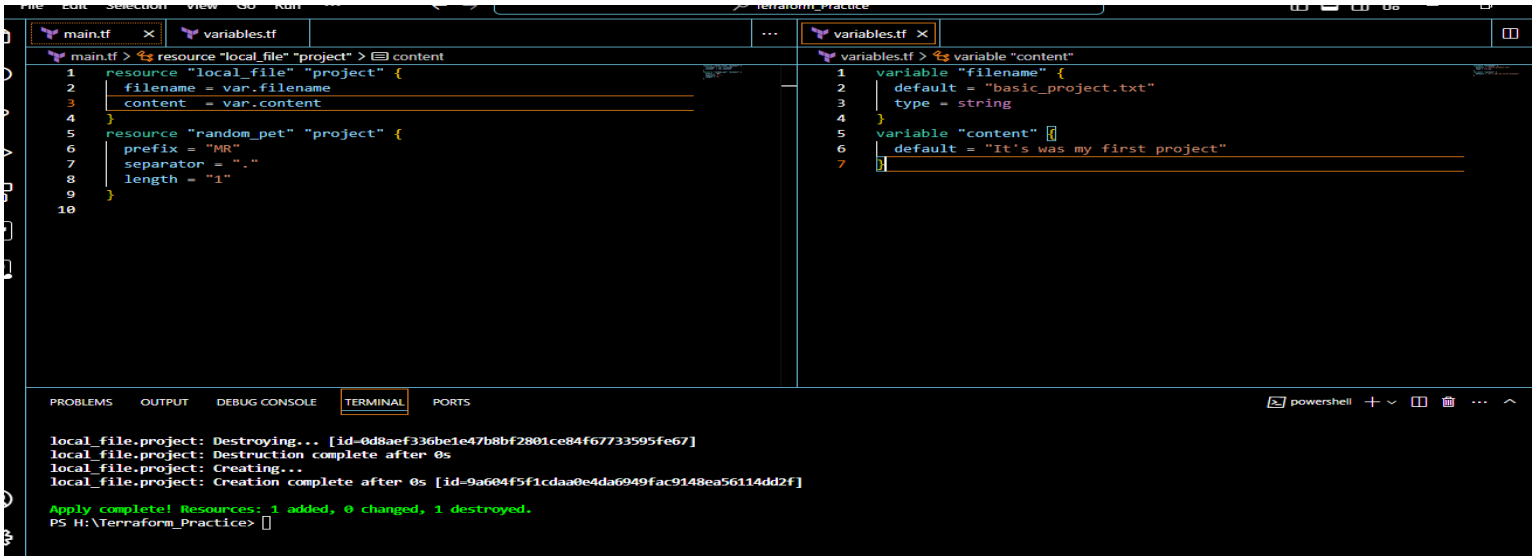
GUI



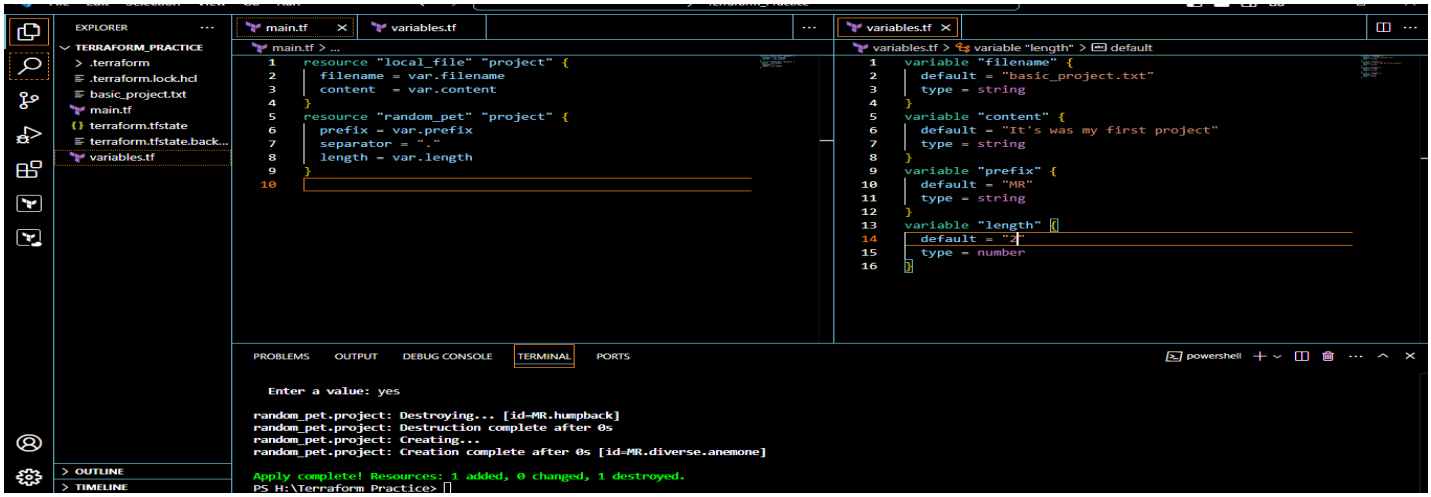


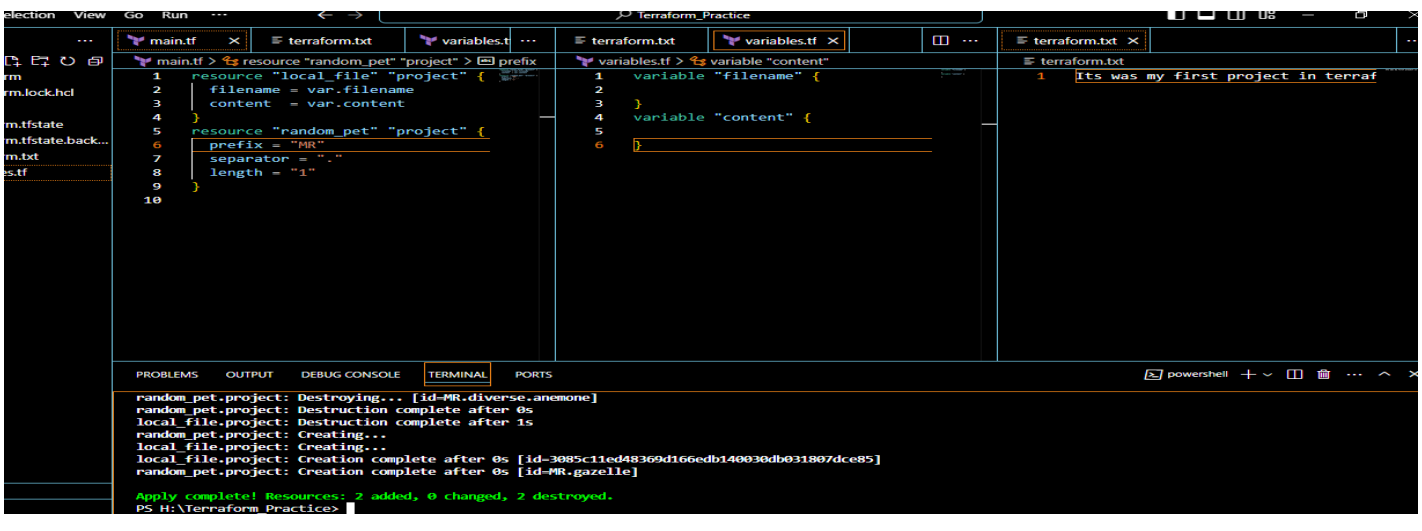
```
[root@ip-172-31-19-218 workspace]# cd terraform
[root@ip-172-31-19-218 terraform]# ll
total 12
-rw-r--r-- 1 jenkins jenkins  124 Dec  5 09:42 main.tf
-rwxr-xr-x 1 jenkins jenkins   45 Dec  5 09:42 project.txt
-rw-r--r-- 1 jenkins jenkins 1620 Dec  5 09:42 terraform.tfstate
[root@ip-172-31-19-218 terraform]#
```

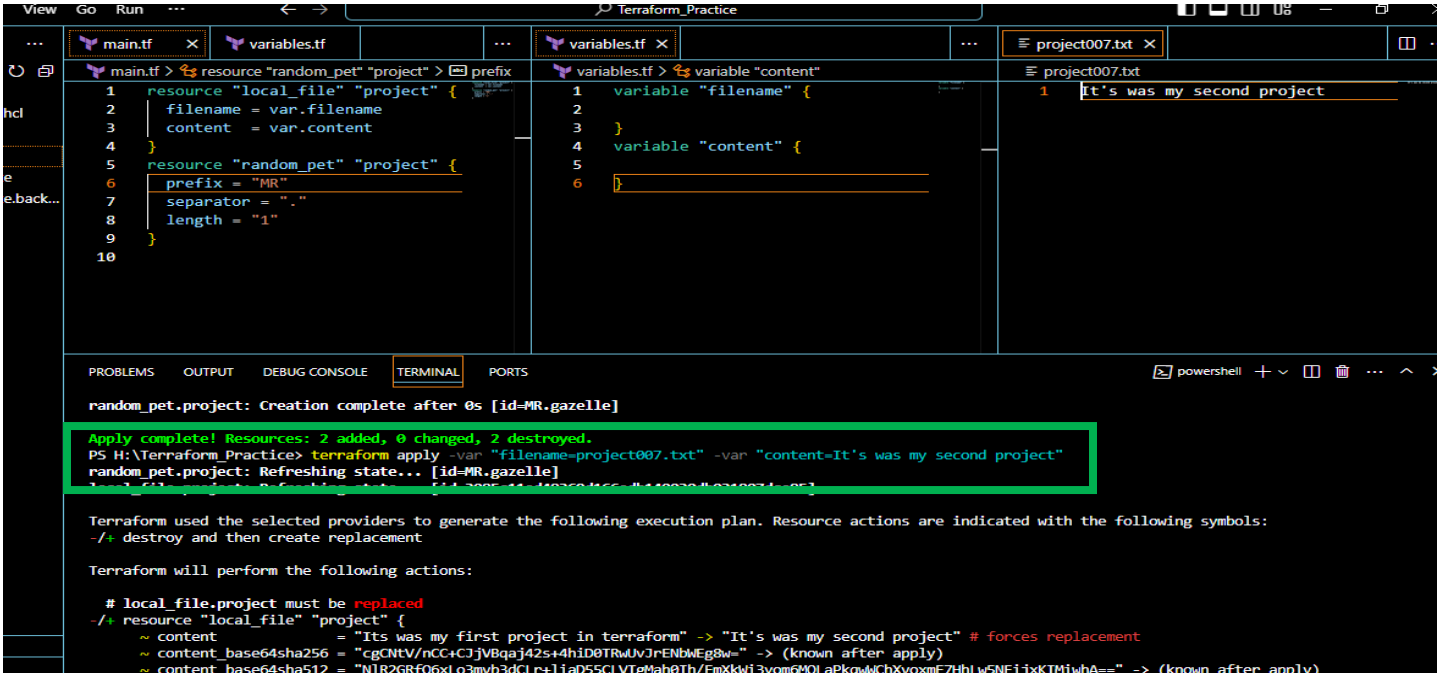
5) Watch the terraform-02 video.

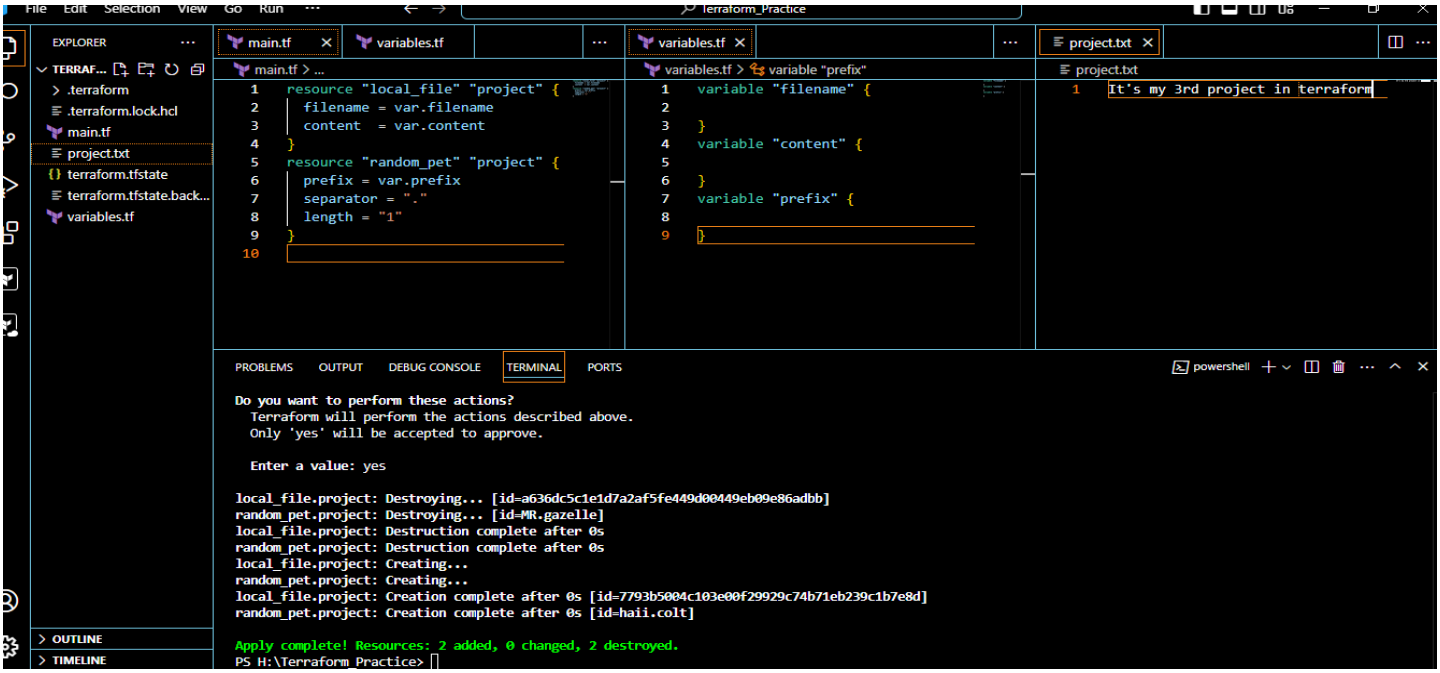


6) Execute all the templates shown in video.

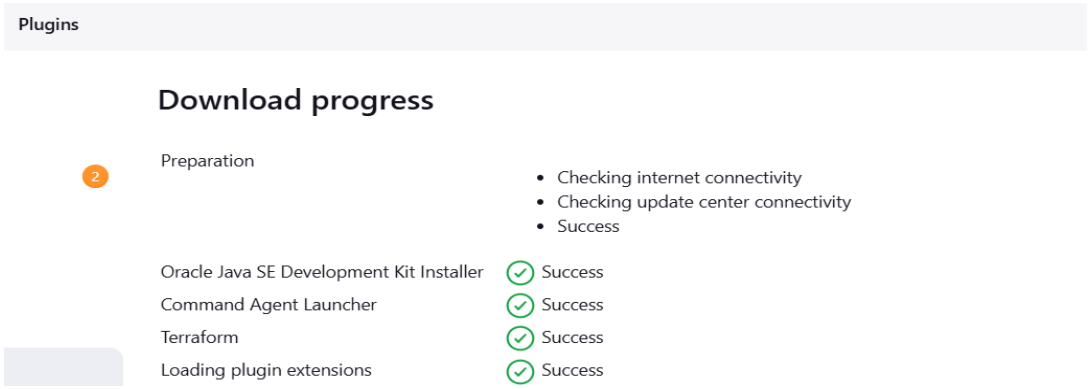


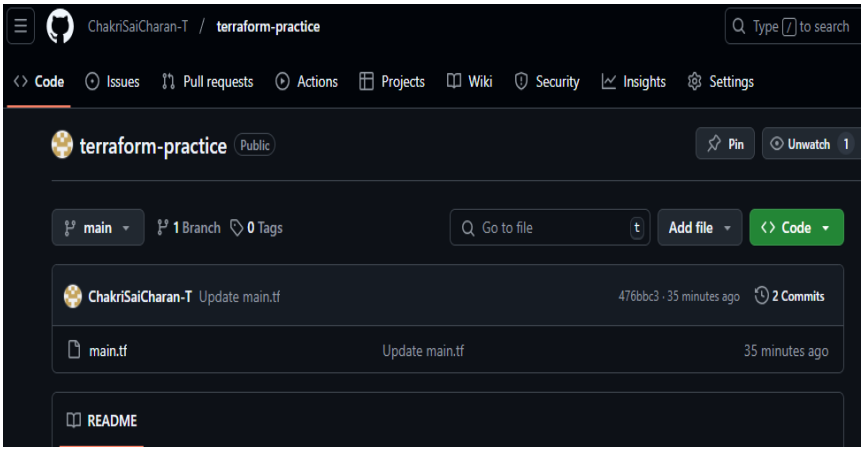


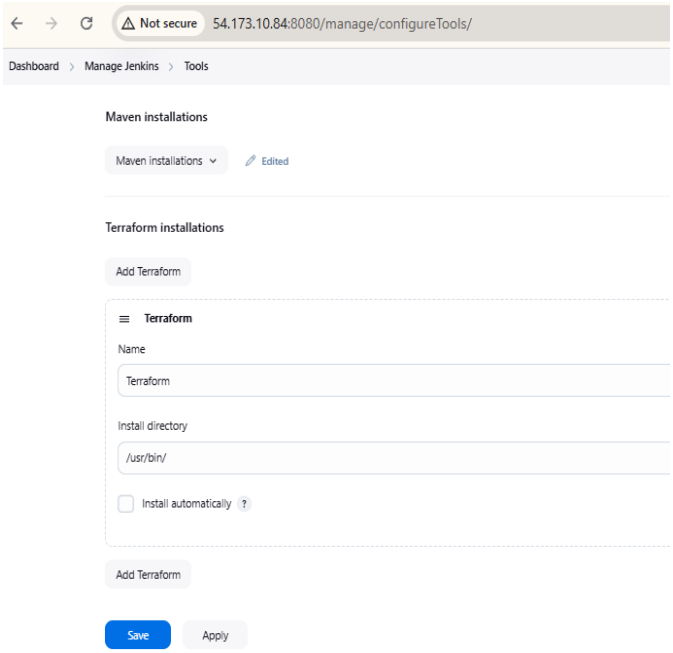

main.tf
variables.tf
project007.txt
resource "local_file" "project" {
filename = var.filename
content = var.content
}
resource "random_pet" "project" {
prefix = "MR"
separator = "."
length = "1"
}
variable "filename" {
}
variable "content" {
}
It's was my second project
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
random_pet.project: Creation complete after 0s [id=MR.gazelle]
Apply complete! Resources: 2 added, 0 changed, 2 destroyed.
PS H:\Terraform_Practice> terraform apply -var "filename=project007.txt" -var "content=It's was my second project"
random_pet.project: Refreshing state... [id=MR.gazelle]
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.project must be replaced
-/+ resource "local_file" "project" {
~ content = "Its was my first project in terraform" -> "It's was my second project" # forces replacement
~ content_base64sha256 = "cgCNtV/nCC+CJjVBqaj42s+4hiD0TRwUvJrENbWEg8w=" -> (known after apply)

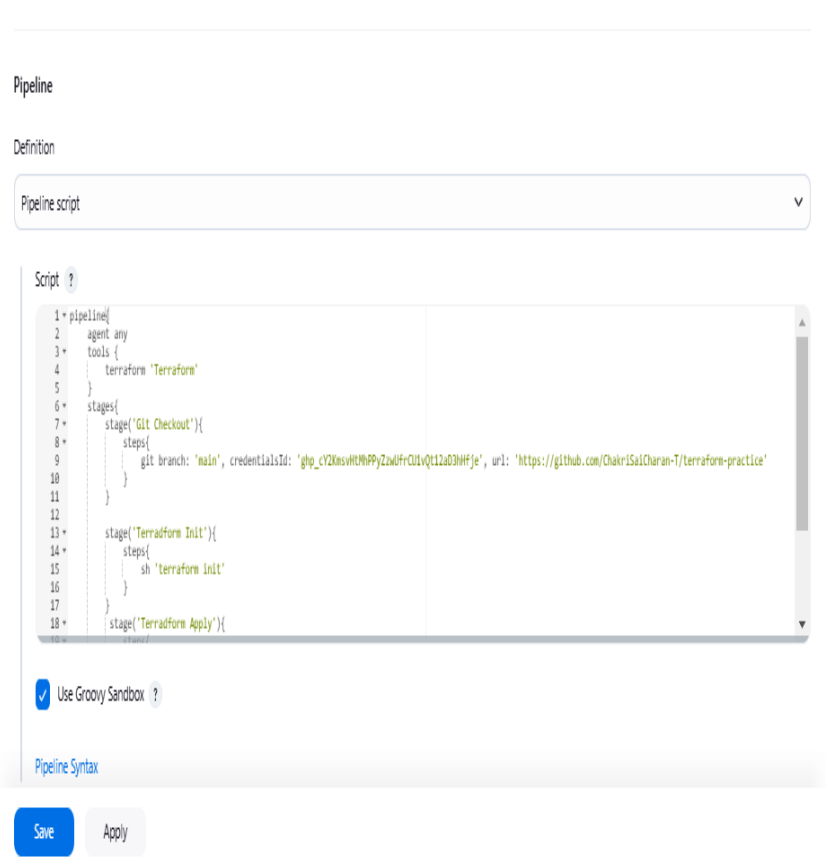

main.tf
variables.tf
project.txt
resource "local_file" "project" {
filename = var.filename
content = var.content
}
resource "random_pet" "project" {
prefix = var.prefix
separator = "."
length = "1"
}
variable "filename" {
}
variable "content" {
}
variable "prefix" {
}
It's my 3rd project in terraform
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
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.project: Destroying... [id=a636dc5c1e1d7a2af5fe449d00449eb09e86adbb]
random_pet.project: Destroying... [id=MR.gazelle]
local_file.project: Destruction complete after 0s
random_pet.project: Destruction complete after 0s
local_file.project: Creating...
random_pet.project: Creating...
local_file.project: Creation complete after 0s [id=7793b5004c103e00f29929c74b71eb239c1b7e8d]
random_pet.project: Creation complete after 0s [id=haii.colt]
Apply complete! Resources: 2 added, 0 changed, 2 destroyed.
PS H:\Terraform_Practice>

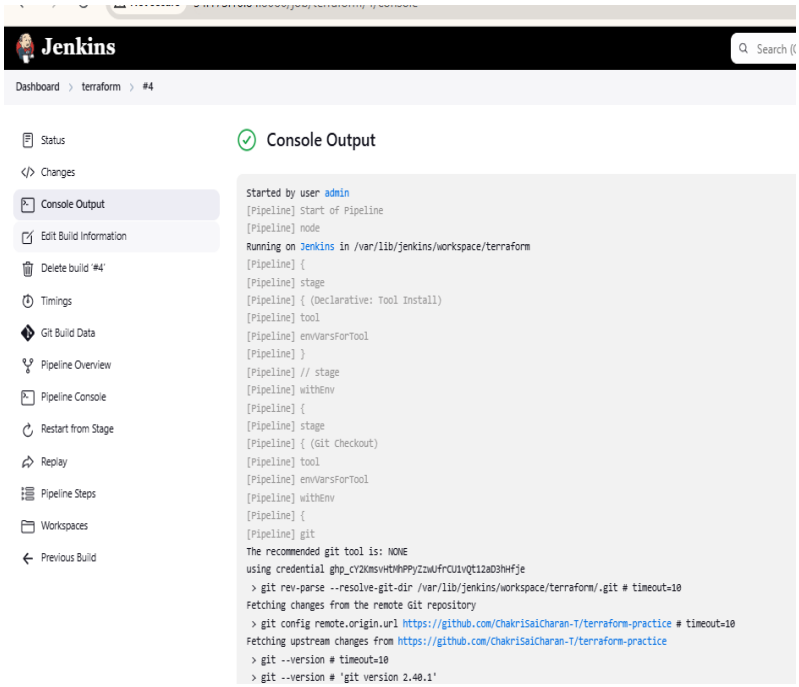
7) Integrate terraform in jenkins using Terraform plugin.










Jenkins
Search (
Dashboard › terraform › #4
Status
Changes
Console Output
Edit Build Information
Delete build '#4'
Timings
Git Build Data
Pipeline Overview
Pipeline Console
Restart from Stage
Replay
Pipeline Steps
Workspaces
Previous Build
Console Output
Started by user admin
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/terraform
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Tool Install)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Git Checkout)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] git
The recommended git tool is: NONE
using credential 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
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/terraform/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/ChakriSaiCharan-T/terraform-practice # timeout=10
Fetching upstream changes from https://github.com/ChakriSaiCharan-T/terraform-practice
> git --version # timeout=10
> git --version # 'git version 2.40.1'