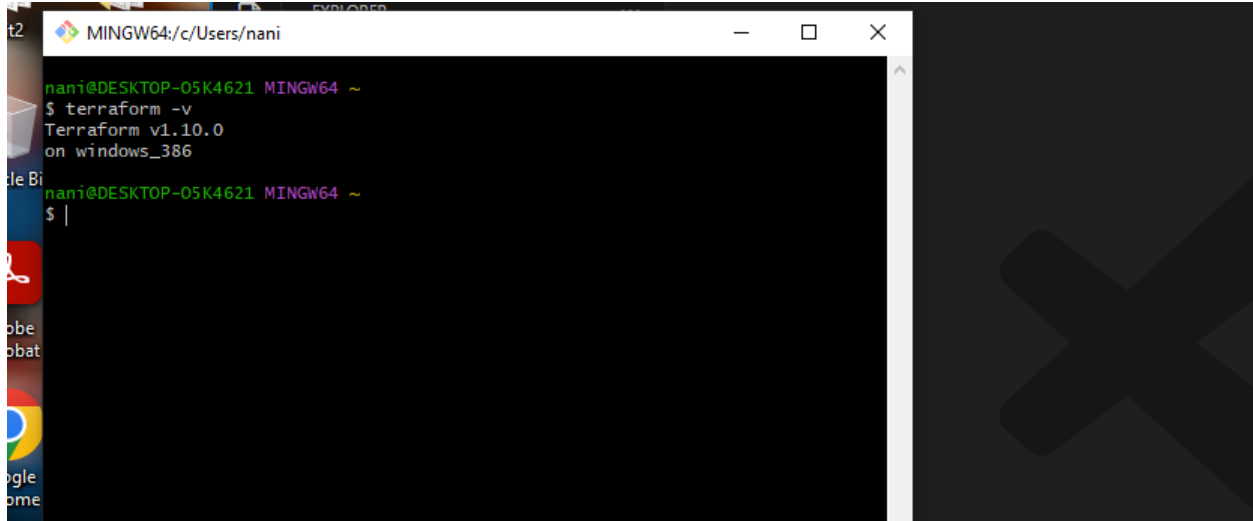


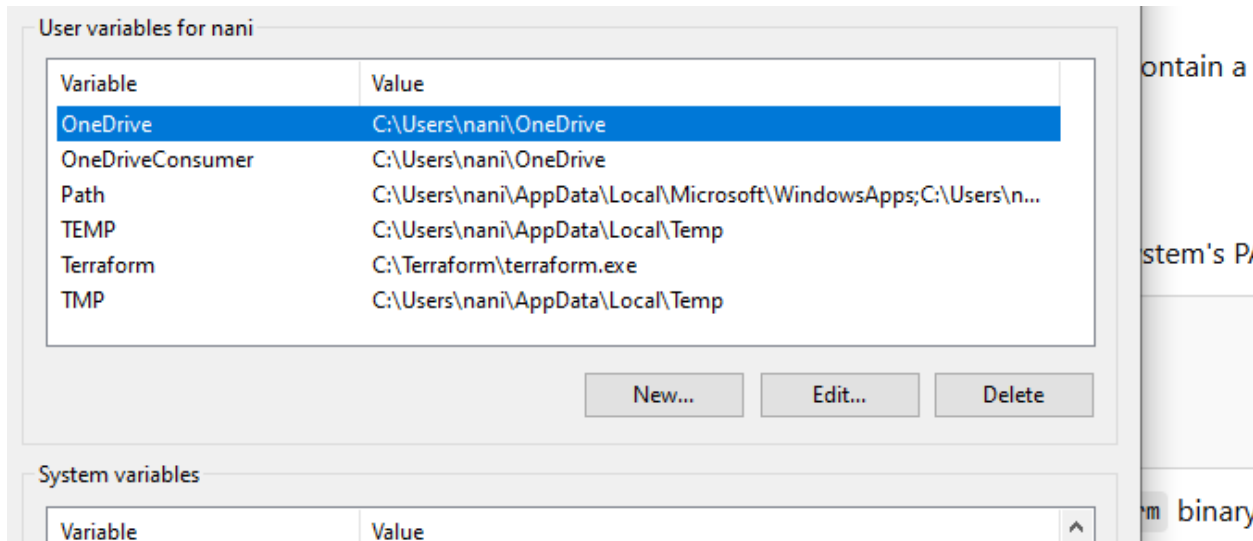
# Terraform 1 and 2

1) Install Terraform on your PC

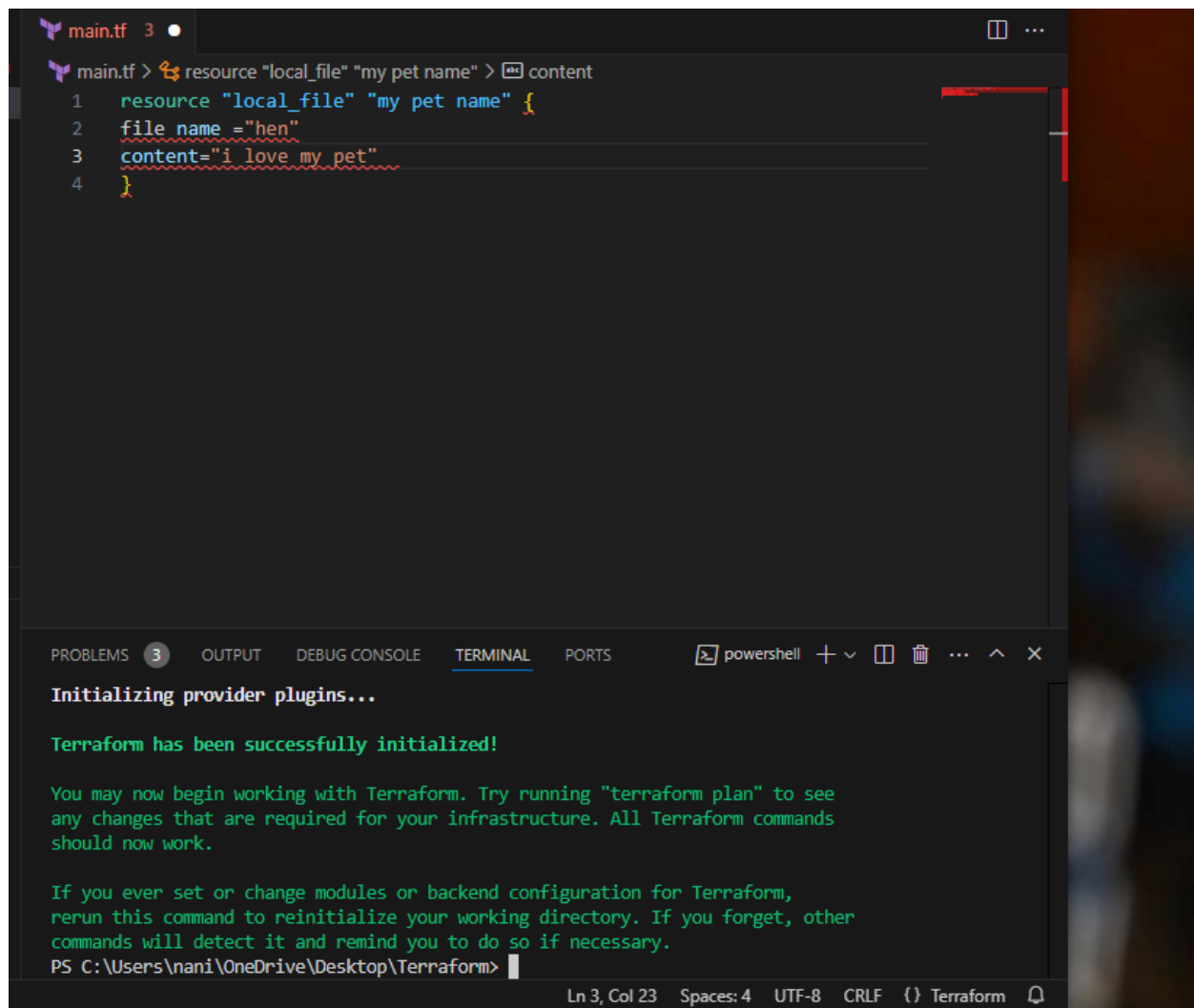


```
MINGW64/c/Users/nani
nani@DESKTOP-05K4621 MINGW64 ~
$ terraform -v
Terraform v1.10.0
on windows_386
nani@DESKTOP-05K4621 MINGW64 ~
$ |
```

2) Execute all the templates shown in video.



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



The image shows a Visual Studio Code editor window with a Terraform configuration file named `main.tf`. The file contains a `resource "local_file" "my pet name"` block with the following properties:

```
1 resource "local_file" "my pet name" {  
2   file_name = "hen"  
3   content = "i love my pet"  
4 }
```

Below the editor, the **TERMINAL** panel is active, displaying the output of the `terraform init` command. The output indicates that Terraform has been successfully initialized and provides instructions on how to proceed with the workflow.

```
Initializing provider plugins...  
  
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\nani\OneDrive\Desktop\Terraform>
```

The status bar at the bottom of the editor shows the current cursor position as **Ln 3, Col 23**, the file encoding as **UTF-8**, and the line ending as **CRLF**. The file is identified as a **Terraform** file.

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.  
PS C:\Users\nani\OneDrive\Desktop\Terraform> terraform apply

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.my_dog will be created
+ resource "local_file" "my_dog" {
  + content           = "I love dogs."
  + 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          = "dog.txt"
  + id                = (known after apply)
}
```

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\nani\OneDrive\Desktop\Terraform> 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.my_dog will be created
+ resource "local_file" "my_dog" {
  + content           = "I love dogs."
  + 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          = "dog.txt"
  + id                = (known after apply)
}
```

Resource actions are indicated with the following symbols:

-/+ destroy and then create replacement

Terraform will perform the following actions:

```
# local_file.mypetname must be replaced
-/+ resource "local_file" "mypetname" {
  ~ content          = "i hate my pet." -> "i love my pet." # forces replacement
  ~ content_base64sha256 = "ZIRB53ZFnCVUaypt23tySrEjY7dr2MqVvBz3aFpBT/8=" ->
    (known after apply)
  ~ content_base64sha512 = "2FN46nr3NU18woZd9u7hyh2wBdwZ3uU/tKmQXV5iRn1kd63T
    ZeKMXzuA0S/pGyZ4B92W1lBGFxZEB7L3nXgVXg==" -> (known after apply)
  .. content_md5         = "f3c5a76ba42dca422cd6a9fed3c1635a" -> (known after apply)
}
```

Only 'yes' will be accepted to approve.

Enter a value: yes

local\_file.mypetname: Destroying... [id=e1668968cd739f189d262613cd7a7bc52ea7c2f3]

local\_file.mypetname: Destruction complete after 0s

local\_file.mypetname: Creating...

local\_file.mypetname: Creation complete after 0s [id=fa7b1a38e4be8da46cd85e757aa8336b05dac07]

Apply complete! Resources: 1 added, 0 changed, 1 destroyed.

D5: C:\Users\pani\OneDrive\Desktop\Terraform\

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Apply complete! Resources: 1 added, 0 changed, 1 destroyed.
PS C:\Users\nani\OneDrive\Desktop\Terraform> terraform destroy
local_file.my_pet_name: Refreshing state... [id=4e164582a728ad937dd78d13507baef647bb7c85]
local_file.my_dog: Refreshing state... [id=6533175f2bfc72c962e46efd7dc7cba270dc0f03]
local_file.mypetname: Refreshing state... [id=fa7b1a38e4be8da46cd85e757aaa8336b05dac07]

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

Terraform will perform the following actions:

# local_file.my_dog will be destroyed
- resource "local_file" "my_dog" {
  - content          = "I love dogs." -> null
  - content_base64sha256 = "gmRl7D79Teh1S0VCCe99gpohfUdDKsTCu2/sIh70KpQ=" -> null
  - content_base64sha512 = "v4GJ8nhgeUamuV1i7rBxtkw90DjLH91EltFy5Wk55UuhHGxFdFR+ywRqaZ3ma0RiPd+kvWRv2t0ci412MoIbBw==" -> null
  - content_md5         = "0ed5241cc7f9182a8c9500e9aad89370" -> null
  - content_sha1        = "6533175f2bfc72c962e46efd7dc7cba270dc0f03" -> null
  - content_sha256      = "826465ec3efd4de8754b454209ef7d829a217d47432ac4c2bb6fec221ef42a94" -> null
}
```

```
main.tf x pets.txt
main.tf > resource "random_pet" "my-pet" > prefix
1 resource "local_file" "my_pet" {
2   filename = "pets.txt"
3   content = "I hate pets!"
4 }
5 resource "random_pet" "my-pet" {
6   prefix = "Miss"
7   separator = "."
8   length = "1"
9 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\0075\Desktop\Terraform_basic>
```

```
local_file.my_dog: Destruction complete after 0s

Destroy complete! Resources: 3 destroyed.
PS C:\Users\nani\OneDrive\Desktop\Terraform> terraform fmt
chief.tf
main.tf
main2.tf
PS C:\Users\nani\OneDrive\Desktop\Terraform> |
```

4) Integrate a sample Terraform template in jenkins.

The top part of the image shows a GitHub repository named 'terraform\_jenkins' by user 'Rakeshpagidimarri'. The repository has 1 branch and 0 tags. The commit history shows three recent commits: 'Update Jenkinsfile' (18 minutes ago), 'Create README.me' (18 minutes ago), and 'Rename terraform to main.tf' (2 minutes ago). The README file is currently empty.

The bottom part of the image shows the Jenkins console output for a build named 'sample\_terraform1' with ID '#2'. The output shows the pipeline starting by user 'rakesh', obtaining the Jenkinsfile from the GitHub repository, and running on Jenkins in the workspace '/var/lib/jenkins/workspace/sample\_terraform1'. The pipeline is currently at the 'checkout' stage, and the console output shows the command 'git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/sample\_terraform1/.git # timeout=10' being executed.

5) Watch the terraform-02 video.

6) Execute all the templates shown in video.

```
main1.tf  x  ...  variable.tf  x  pets.txt  ...
main1.tf > resource "random_pet" "petname" > length
1  resource "local_file" "mypetname" {
2    filename = var.filename
3    content = var.content
4  }
5
6  resource "random_pet" "petname" {
7    prefix = "miss"
8    separator = "."
9    length = var.length
10 }

variable.tf > variable "content" > default
1  variable "filename" {
2    default = "pets.txt"
3    type = string
4  }
5
6  variable "content" {
7    default = "i love dogs"
8    type = string
9  }
10 variable "prefix" {
11   default = "MR"
12   type = string
13 }
14 variable "length" {
15   default = "1"
16 }
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

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.mypetname: Destroying... [id=f140aba43cbc42844ecb543aeedcbbd239f626e7]  
local\_file.mypetname: Destruction complete after 0s  
local\_file.mypetname: Creating...  
local\_file.mypetname: Creation complete after 0s [id=c524a39c02f142ba0b81da289f2e11332d59b4dd]

Apply complete! Resources: 1 added, 0 changed, 1 destroyed.  
PS C:\Users\nani\OneDrive\Desktop\Terraform>

Ln 7, Col 25 Spaces: 4 UTF-8 CRLF {} Terraform

main1.tf terraform.tfstate

main1.tf > resource "random\_pet" "petname" > separator

```
1 resource "local_file" "mypetname" {
2     filename = "pets.txt"
3     content = "i love my zebra"
4     lifecycle {
5         create_before_destroy = true
6     }
7 }
8
9 resource "random_pet" "petname" {
10     prefix = "miss"
11     separator = "."
12     length = "1"
13 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Only 'yes' will be accepted to approve.

Enter a value: yes

random\_pet.petname: Destroying... [id=mr.penguin]  
random\_pet.petname: Destruction complete after 0s  
local\_file.mypetname: Creating...  
local\_file.mypetname: Creation complete after 0s [id=fe5079e178b52a30c9cbc77f047329cf7f63c173]  
random\_pet.petname: Creating...  
random\_pet.petname: Creation complete after 0s [id=miss.orca]  
local\_file.mypetname (deposed object b0fd656a): Destroying... [id=ab5698382c31cff07c6adfed0c7167d1c348806]  
local\_file.mypetname: Destruction complete after 0s

Apply complete! Resources: 2 added, 0 changed, 2 destroyed.

PS C:\Users\nani\OneDrive\Desktop\Terraform>

Ln 11, Col 18, Spaces: 4, UTF-8, CRLF, {} Terraform



main1.tf terraform.tfstate

main1.tf > resource "local\_file" "mypetname" > lifecycle > prevent\_destroy

```
1 resource "local_file" "mypetname" {
2     filename = "pets.txt"
3     content = "i love my zebra"
4     lifecycle {
5         prevent_destroy = true
6     }
7 }
8
9 resource "random_pet" "petname" {
10     prefix = "miss"
11     separator = "."
12     length = "1"
13 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

+ 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.mypetname: Creating...  
local\_file.mypetname: Creation complete after 0s [id=fe5079e178b52a30c9cbc77f047329cf7f63c173]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

PS C:\Users\nani\OneDrive\Desktop\Terraform>

powershell powershell

main1.tf × terraform.tfstate

main1.tf > resource "local\_file" "mypetname" > lifecycle > [ ] ignore\_changes > 0

```
1 resource "local_file" "mypetname" {
2   filename = "pets.txt"
3   content = "i love my zebra11"
4   lifecycle {
5     ignore_changes = [
6     contant
7   ]
8 }
9 }
10
11 resource "random_pet" "petname" {
12   prefix = "miss"
13   separator = "."
14   length = "1"
15 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

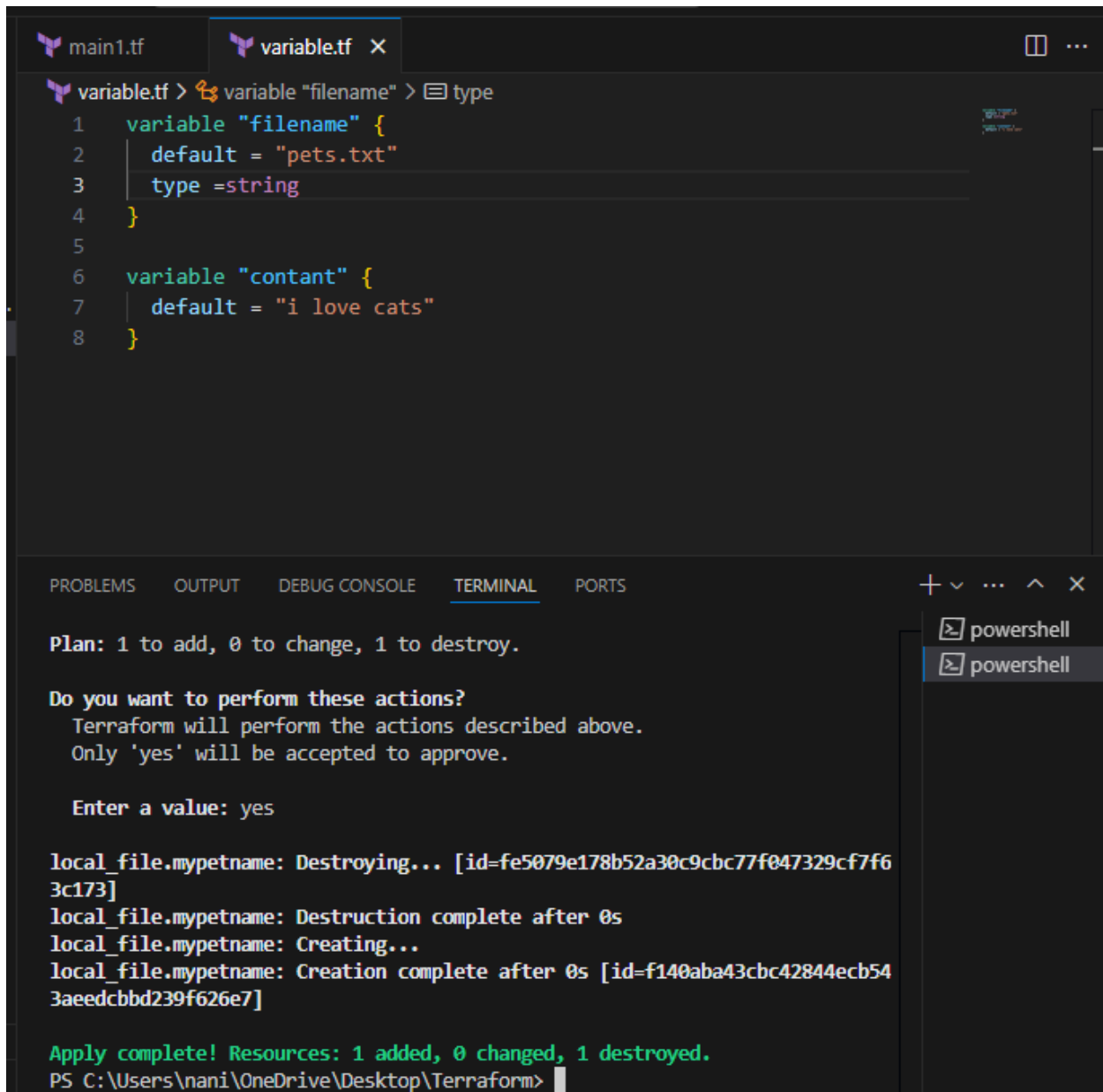
on main1.tf line 6, in resource "local\_file" "mypetname":  
6: contant

This object has no argument, nested block, or exported attribute named "contant". Did you mean "content"?

PS C:\Users\nani\OneDrive\Desktop\Terraform> terraform apply

+ ^ ... ^ ×

powershell  
powershell



The screenshot shows the Visual Studio Code editor with two tabs: `main1.tf` and `variable.tf`. The `variable.tf` file is open, showing the following Terraform configuration:

```
1 variable "filename" {
2   default = "pets.txt"
3   type = string
4 }
5
6 variable "contant" {
7   default = "i love cats"
8 }
```

Below the editor, the **TERMINAL** panel is active, displaying the output of a Terraform plan and apply command. The output shows the destruction of a resource and the creation of a new one.

```
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.mypetname: Destroying... [id=fe5079e178b52a30c9cbc77f047329cf7f63c173]
local_file.mypetname: Destruction complete after 0s
local_file.mypetname: Creating...
local_file.mypetname: Creation complete after 0s [id=f140aba43cbc42844ecb543aeedcbdd239f626e7]

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

7) Integrate terraform in jenkins using Terraform plugin.

terraform-sample7 / jenkinsfile

Rakeshpagidimarri Update jenkinsfile

ff2ff1d · now History

Code Blame 21 lines (20 loc) · 485 Bytes Code 55% faster with GitHub Copilot

Raw Copy Download Edit View

```
1 pipeline {
2   agent any
3   stages {
4     stage('Clone') {
5       steps {
6         git branch: 'main', url: 'https://github.com/Rakeshpagidimarri/terraform-sample7.git '
7       }
8     }
9   }
10  stage('Initialize Terraform') {
11    steps {
12      sh 'terraform init'
13    }
14  }
15  stage('Execute Terraform Apply') {
16    steps {
17      sh 'terraform apply -auto-approve'
18    }
19  }
20 }
```

Rakeshpagidimarri Create main.tf

Code Blame 12 lines (12 loc) · 221 Bytes Code 55% faster with GitHub Copilot

```
1 resource "local_file" "mypetname" {
2   filename = "pets.txt"
3   content = "i love my pets"
4   lifecycle {
5     create_before_destroy = true
6   }
7 }
8 resource "random_pet" "petname" {
9   prefix = "Mr"
10  separator = "."
11  length = "1"
12 }
```

## Pipeline

### Definition

Pipeline script from SCM

### SCM ?

Git

### Repositories ?

#### Repository URL ?

https://github.com/Rakeshpagidimarri/terraform-sample7.git

#### Credentials ?

- none -

+ Add

Dashboard > sample\_7 > #2

Status

Changes

Console Output

Edit Build Information

Delete build '#2'

Timings

Git Build Data

Pipeline Overview

Pipeline Console

## Console Output

Download

Copy

View as plain text

```
Started by user rakesh
Obtained Jenkinsfile from git https://github.com/Rakeshpagidimarri/terraform-sample7.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/sample_7
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
```