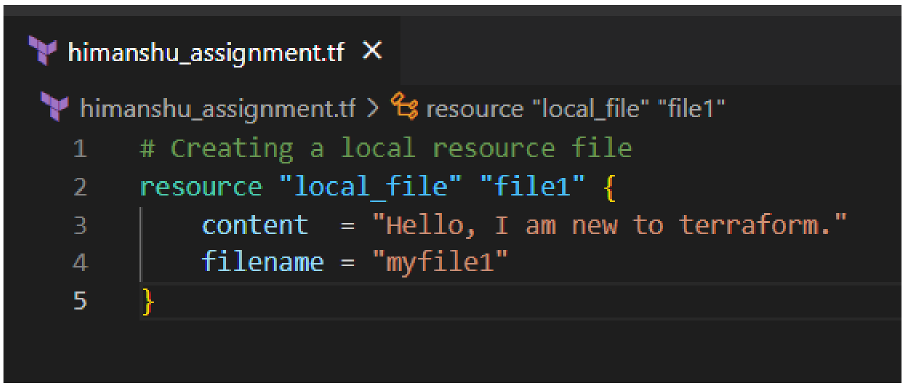
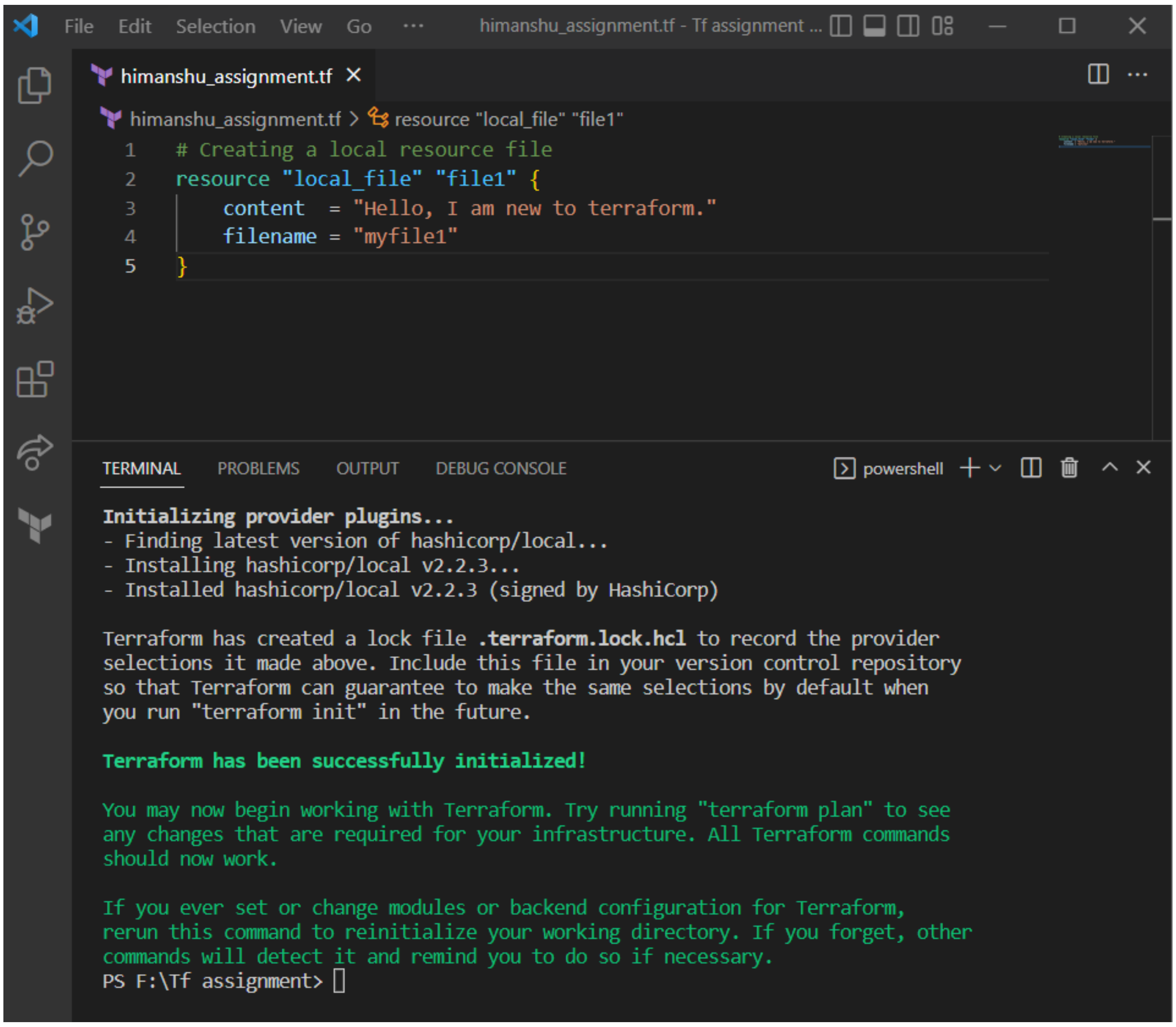
Creating a resource

Creating a local resource file with name file1



```
# Creating a local resource file
resource "local_file" "file1" {
    content  = "Hello, I am new to terraform."
    filename = "myfile1"
}
```

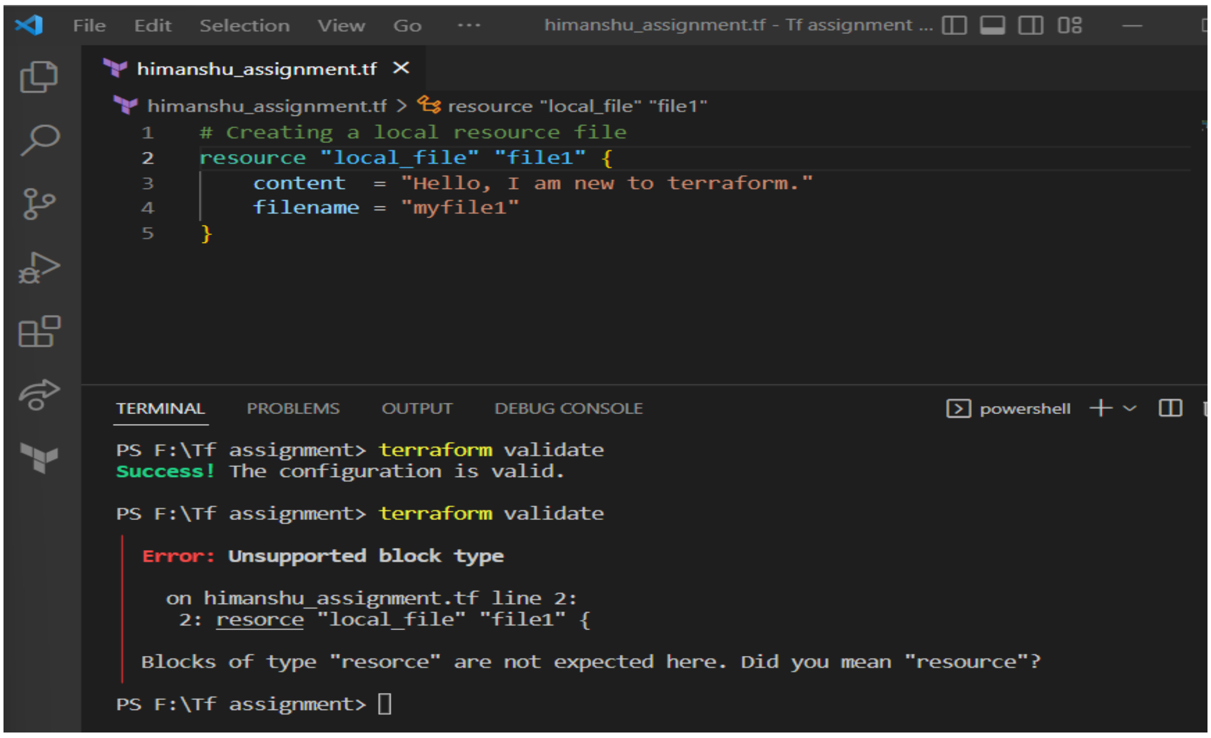
Terraform init


File Edit Selection View Go ···
himanshu_assignment.tf - Tf assignment ...
himanshu_assignment.tf
himanshu_assignment.tf > resource "local_file" "file1"
TERMINAL
PROBLEMS
OUTPUT
DEBUG CONSOLE
powershell

```
# Creating a local resource file
resource "local_file" "file1" {
    content  = "Hello, I am new to terraform."
    filename = "myfile1"
}
```

```
Initializing provider plugins...
- Finding latest version of hashicorp/local...
- Installing hashicorp/local v2.2.3...
- Installed hashicorp/local v2.2.3 (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 F:\Tf assignment>
```

Terraform validate

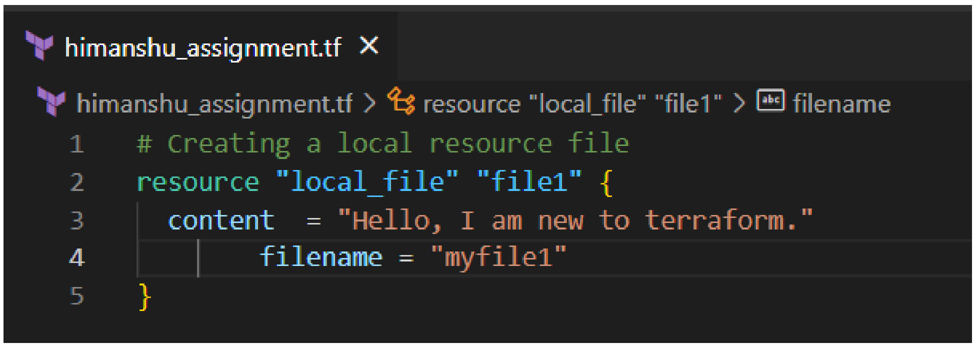


himanshu_assignment.tf > resource "local_file" "file1"

```
# Creating a local resource file
resource "local_file" "file1" {
    content  = "Hello, I am new to terraform."
    filename = "myfile1"
}
```

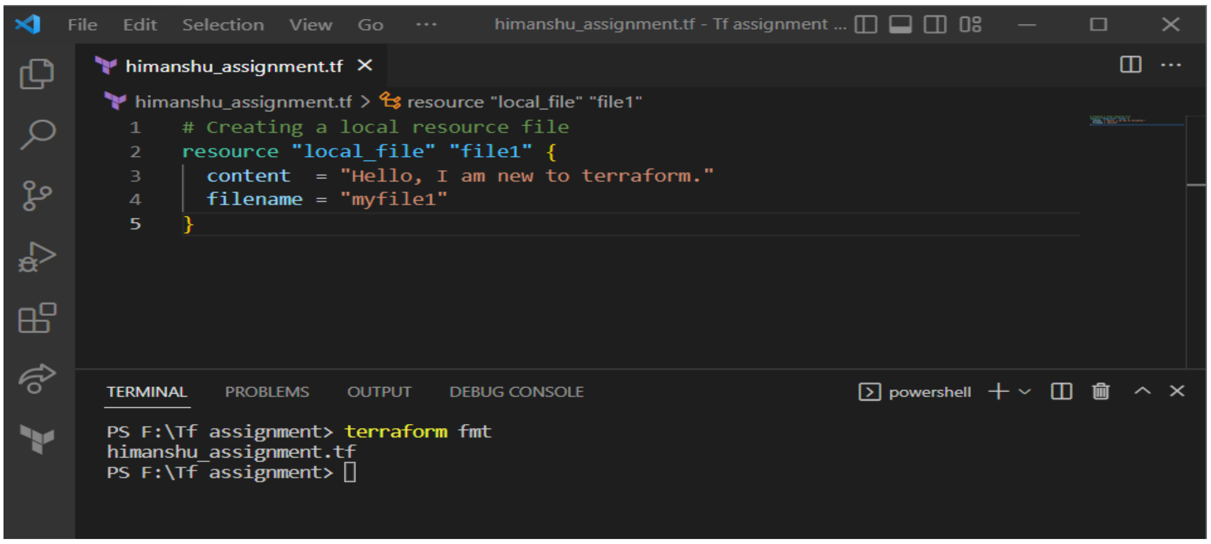
```
PS F:\Tf assignment> terraform validate
Success! The configuration is valid.

PS F:\Tf assignment> terraform validate

  Error: Unsupported block type

    on himanshu_assignment.tf line 2:
     2: resorce "local_file" "file1" {

  Blocks of type "resorce" are not expected here. Did you mean "resource"?

PS F:\Tf assignment> 
```

Terraform fmt



himanshu_assignment.tf > resource "local_file" "file1" > filename

```
# Creating a local resource file
resource "local_file" "file1" {
  content  = "Hello, I am new to terraform."
        filename = "myfile1"
}
```

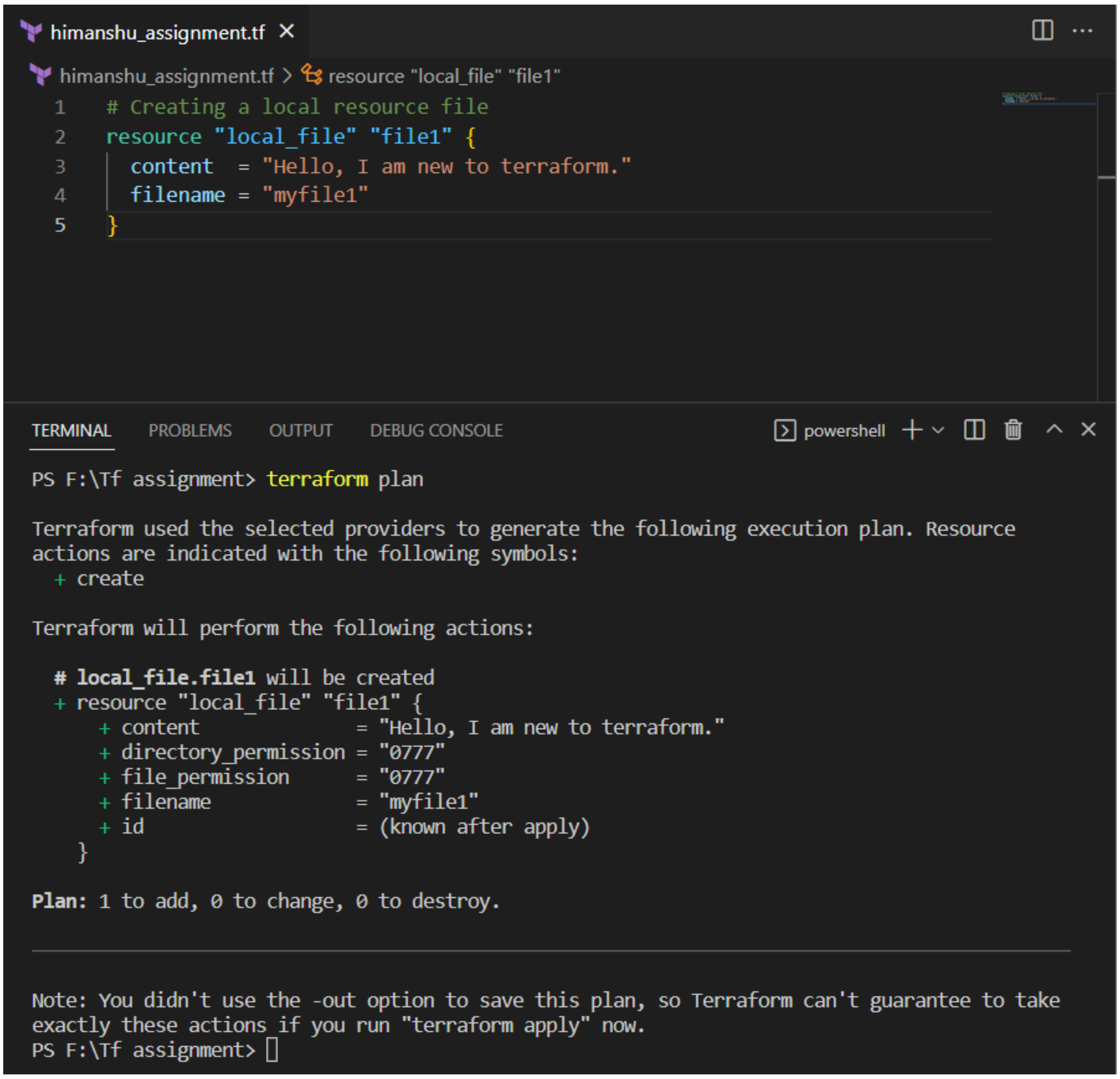


himanshu_assignment.tf > resource "local_file" "file1"

```
# Creating a local resource file
resource "local_file" "file1" {
  content  = "Hello, I am new to terraform."
  filename = "myfile1"
}
```

```
PS F:\Tf assignment> terraform fmt
himanshu_assignment.tf
PS F:\Tf assignment> 
```

Terraform plan



himanshu_assignment.tf

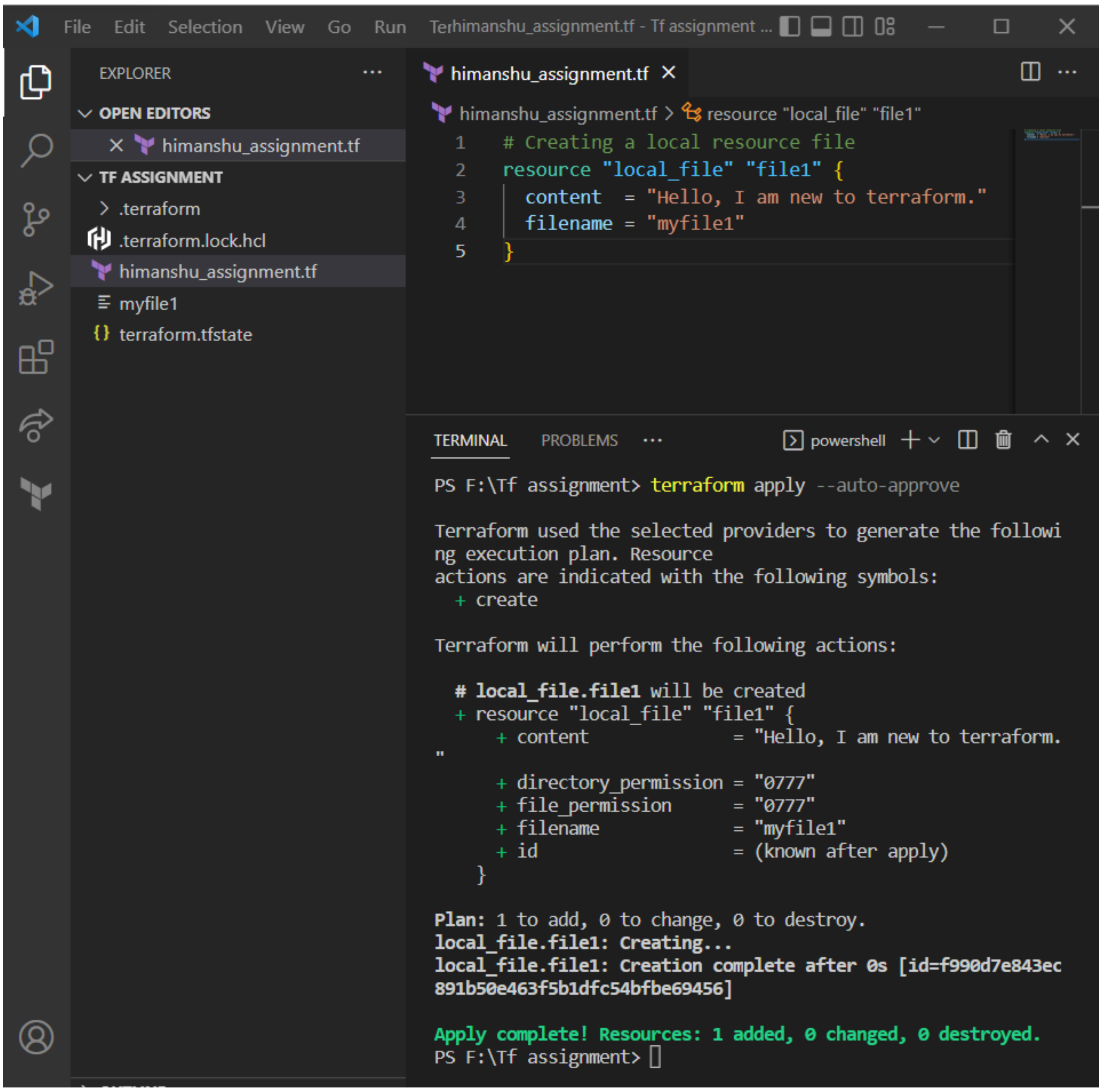
himanshu_assignment.tf > resource "local_file" "file1"

```
# Creating a local resource file
resource "local_file" "file1" {
  content  = "Hello, I am new to terraform."
  filename = "myfile1"
}
```

TERMINAL   PROBLEMS   OUTPUT   DEBUG CONSOLE   powershell

```
PS F:\Tf assignment> 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.file1 will be created
  + resource "local_file" "file1" {
      + content              = "Hello, I am new to terraform."
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "myfile1"
      + id                   = (known after apply)
    }

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

─────────────────────────────────────────────────────────────────────────────

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take
exactly these actions if you run "terraform apply" now.
PS F:\Tf assignment> 
```

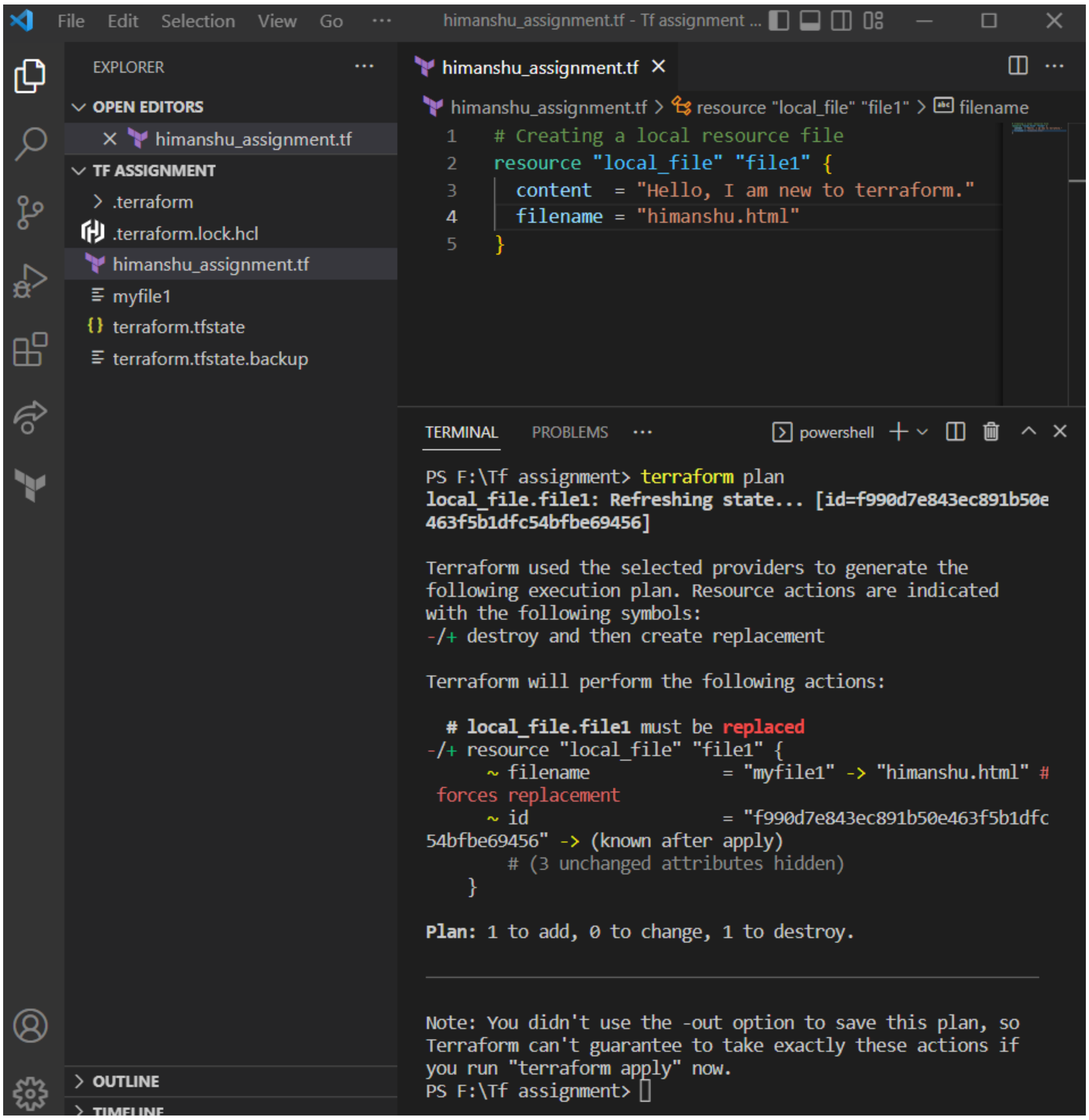
Terraform apply and auto-approve



```
# Creating a local resource file
resource "local_file" "file1" {
  content  = "Hello, I am new to terraform."
  filename = "myfile1"
}
```

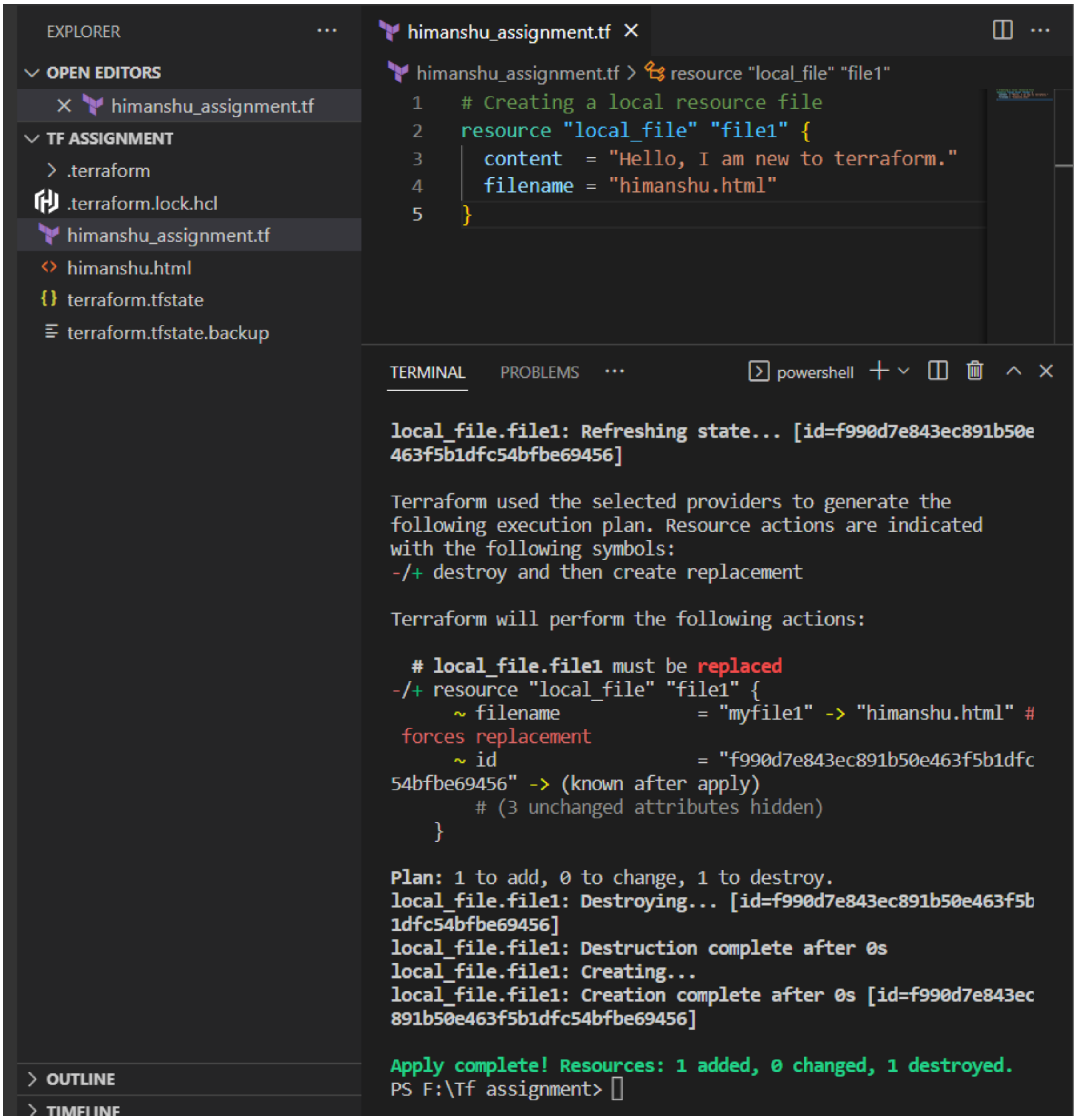
```
PS F:\Tf assignment> terraform apply --auto-approve

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

Terraform will perform the following actions:

  # local_file.file1 will be created
  + resource "local_file" "file1" {
      + content              = "Hello, I am new to terraform.
"
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "myfile1"
      + id                   = (known after apply)
    }

Plan: 1 to add, 0 to change, 0 to destroy.
local_file.file1: Creating...
local_file.file1: Creation complete after 0s [id=f990d7e843ec
891b50e463f5b1dfc54bfbe69456]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS F:\Tf assignment> 
```

Creating different file type



```
# Creating a local resource file
resource "local_file" "file1" {
  content  = "Hello, I am new to terraform."
  filename = "himanshu.html"
}
```

```
PS F:\Tf assignment> terraform plan
local_file.file1: Refreshing state... [id=f990d7e843ec891b50e
463f5b1dfc54bfbe69456]

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.file1 must be replaced
-/+ resource "local_file" "file1" {
      ~ filename             = "myfile1" -> "himanshu.html" #
 forces replacement
      ~ id                   = "f990d7e843ec891b50e463f5b1dfc
54bfbe69456" -> (known after apply)
        # (3 unchanged attributes hidden)
    }

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

─────────────────────────────────────────────────────────────

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 F:\Tf assignment> 
```

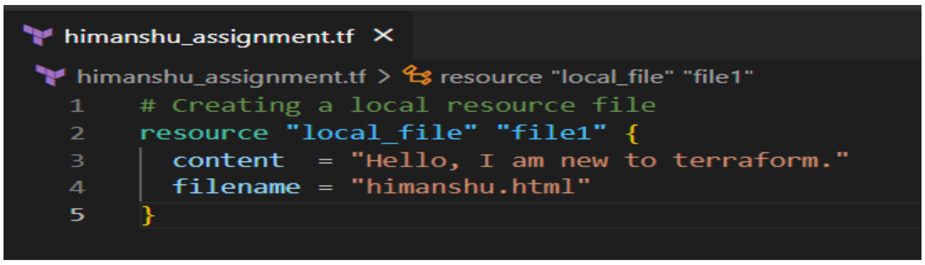


```
# Creating a local resource file
resource "local_file" "file1" {
  content  = "Hello, I am new to terraform."
  filename = "himanshu.html"
}
```

```
local_file.file1: Refreshing state... [id=f990d7e843ec891b50e
463f5b1dfc54bfbe69456]

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.file1 must be replaced
-/+ resource "local_file" "file1" {
      ~ filename             = "myfile1" -> "himanshu.html" #
 forces replacement
      ~ id                   = "f990d7e843ec891b50e463f5b1dfc
54bfbe69456" -> (known after apply)
        # (3 unchanged attributes hidden)
    }

Plan: 1 to add, 0 to change, 1 to destroy.
local_file.file1: Destroying... [id=f990d7e843ec891b50e463f5b
1dfc54bfbe69456]
local_file.file1: Destruction complete after 0s
local_file.file1: Creating...
local_file.file1: Creation complete after 0s [id=f990d7e843ec
891b50e463f5b1dfc54bfbe69456]

Apply complete! Resources: 1 added, 0 changed, 1 destroyed.
PS F:\Tf assignment>
```

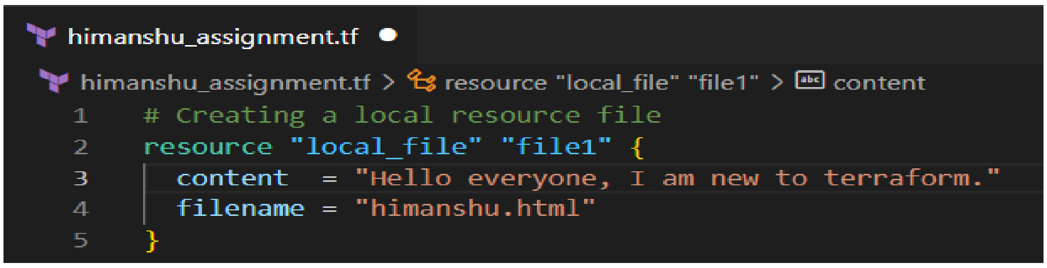
Changing the content

Old content



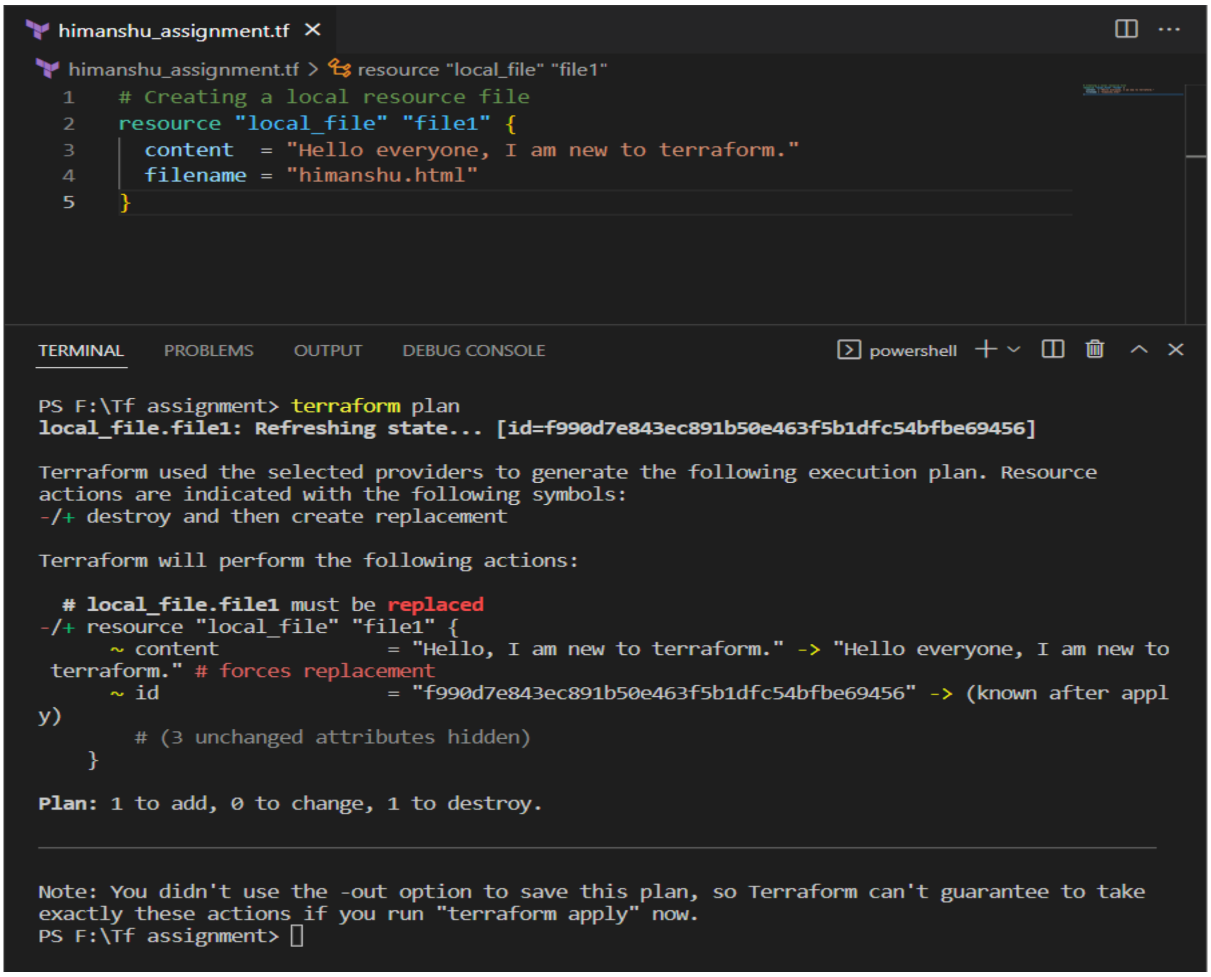
```
# Creating a local resource file
resource "local_file" "file1" {
  content  = "Hello, I am new to terraform."
  filename = "himanshu.html"
}
```

New content


himanshu_assignment.tf
himanshu_assignment.tf > resource "local_file" "file1" > content

```
# Creating a local resource file
resource "local_file" "file1" {
  content  = "Hello everyone, I am new to terraform."
  filename = "himanshu.html"
}
```

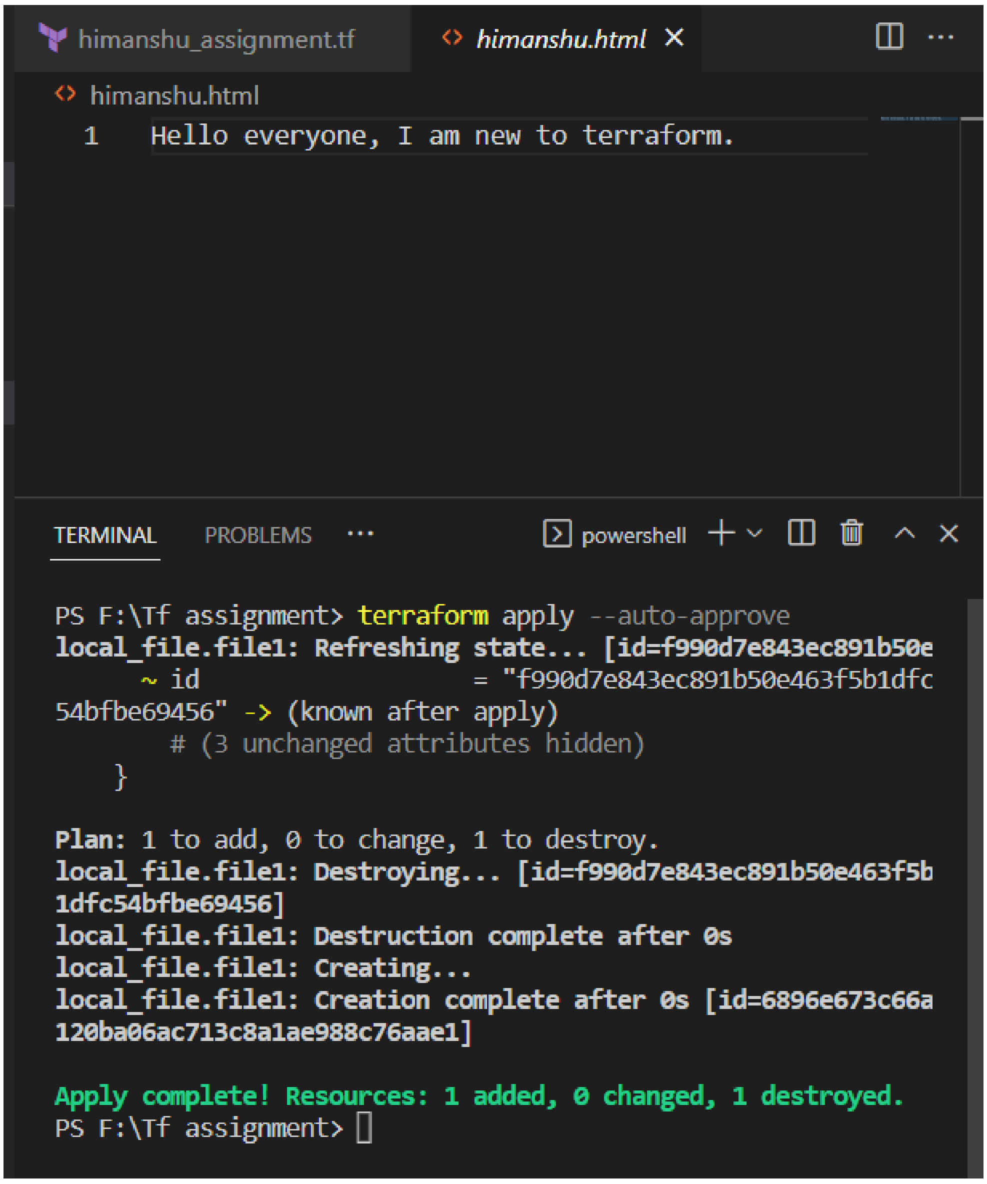
Doing terraform plan



```
# Creating a local resource file
resource "local_file" "file1" {
  content  = "Hello everyone, I am new to terraform."
  filename = "himanshu.html"
}
```

```
PS F:\Tf assignment> terraform plan
local_file.file1: Refreshing state... [id=f990d7e843ec891b50e463f5b1dfc54bfbe69456]

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.file1 must be replaced
-/+ resource "local_file" "file1" {
      ~ content              = "Hello, I am new to terraform." -> "Hello everyone, I am new to
 terraform." # forces replacement
      ~ id                   = "f990d7e843ec891b50e463f5b1dfc54bfbe69456" -> (known after appl
y)
        # (3 unchanged attributes hidden)
    }

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

─────────────────────────────────────────────────────────────────────────────

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 F:\Tf assignment> 
```

Terraform apply

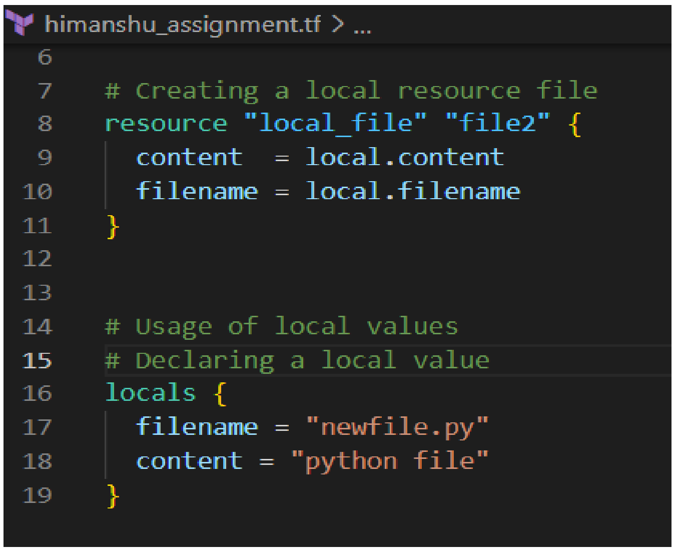


```
Hello everyone, I am new to terraform.
```

```
PS F:\Tf assignment> terraform apply --auto-approve
local_file.file1: Refreshing state... [id=f990d7e843ec891b50e
      ~ id                   = "f990d7e843ec891b50e463f5b1dfc
54bfbe69456" -> (known after apply)
        # (3 unchanged attributes hidden)
    }

Plan: 1 to add, 0 to change, 1 to destroy.
local_file.file1: Destroying... [id=f990d7e843ec891b50e463f5b
1dfc54bfbe69456]
local_file.file1: Destruction complete after 0s
local_file.file1: Creating...
local_file.file1: Creation complete after 0s [id=6896e673c66a
120ba06ac713c8a1ae988c76aae1]

Apply complete! Resources: 1 added, 0 changed, 1 destroyed.
PS F:\Tf assignment> 
```

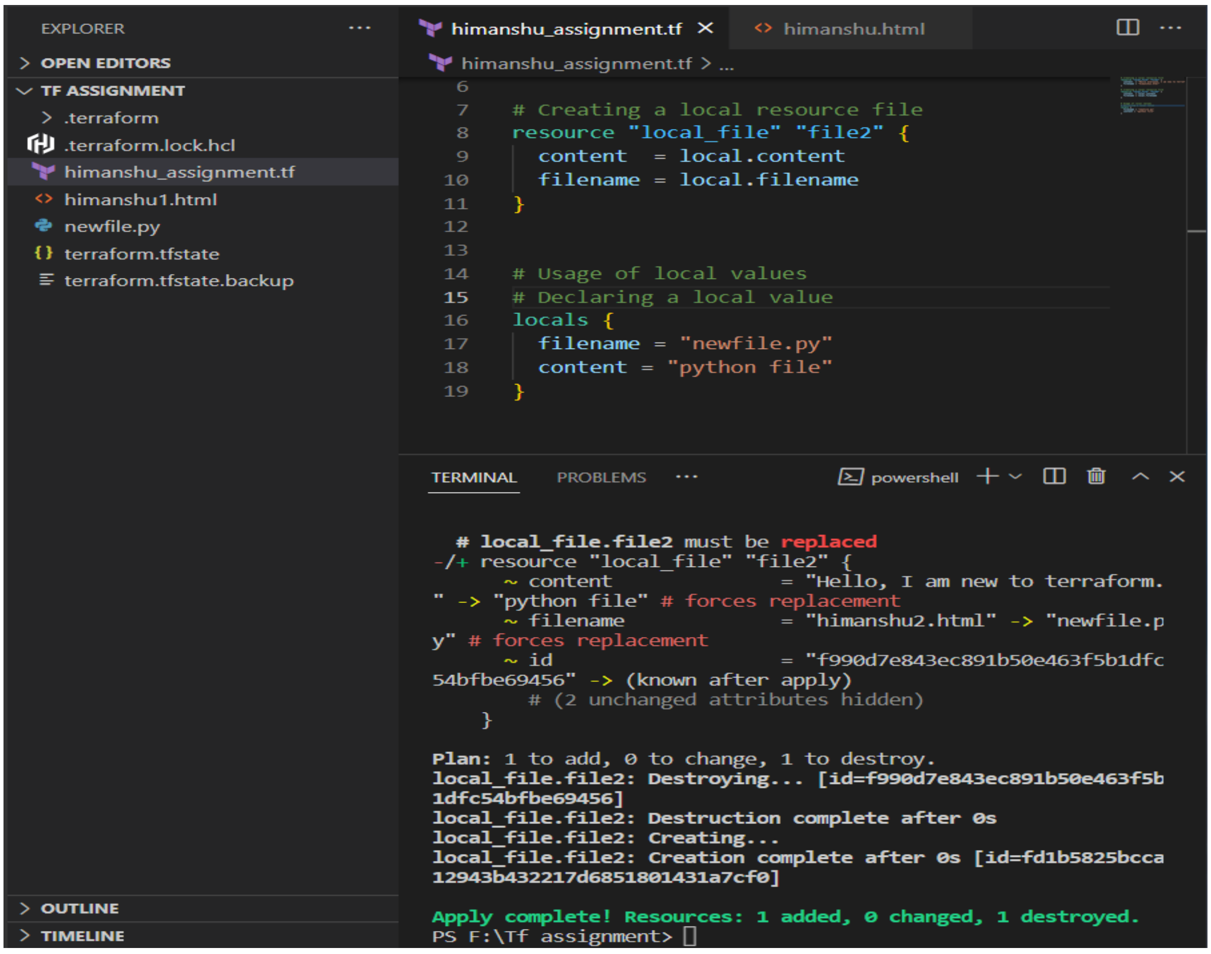
Local values

Declaring a local value



```
himanshu_assignment.tf > ...

# Creating a local resource file
resource "local_file" "file2" {
  content  = local.content
  filename = local.filename
}


# Usage of local values
# Declaring a local value
locals {
  filename = "newfile.py"
  content = "python file"
}
```

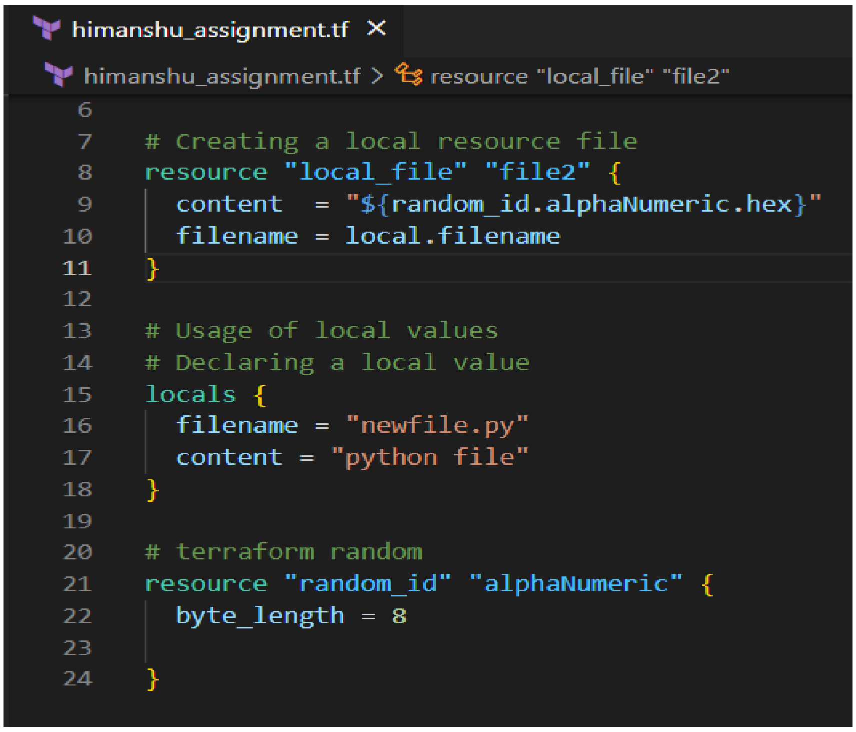
After terraform plan and terraform apply –auto-approve, himanshu2.html file will be replaced with the new file that is newfile.py.



```
himanshu_assignment.tf > ...

# Creating a local resource file
resource "local_file" "file2" {
  content  = local.content
  filename = local.filename
}


# Usage of local values
# Declaring a local value
locals {
  filename = "newfile.py"
  content = "python file"
}
```

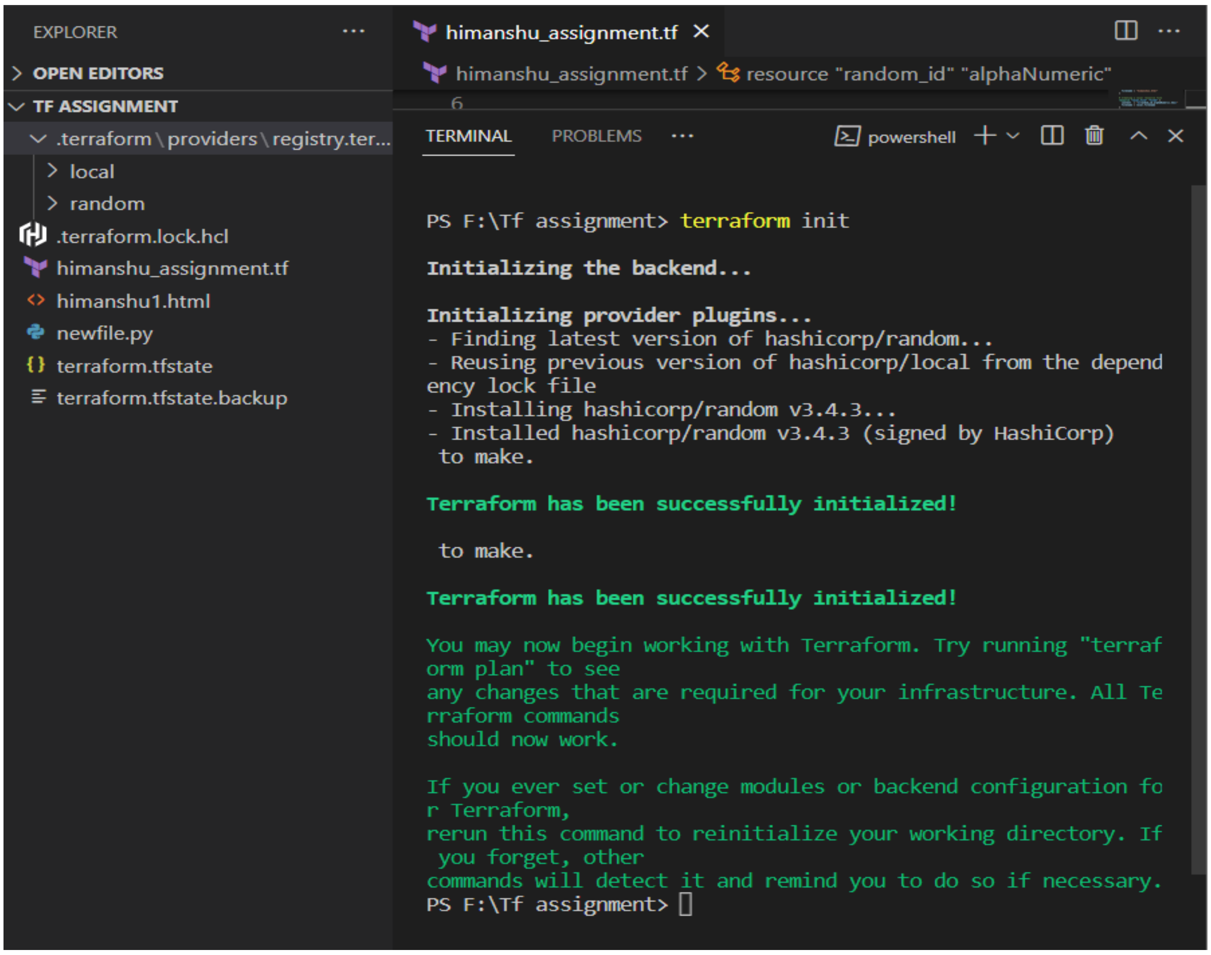
```
  # local_file.file2 must be replaced
-/+ resource "local_file" "file2" {
      ~ content              = "Hello, I am new to terraform.
" -> "python file" # forces replacement
      ~ filename             = "himanshu2.html" -> "newfile.p
y" # forces replacement
      ~ id                   = "f990d7e843ec891b50e463f5b1dfc
54bfbe69456" -> (known after apply)
        # (2 unchanged attributes hidden)
    }

Plan: 1 to add, 0 to change, 1 to destroy.
local_file.file2: Destroying... [id=f990d7e843ec891b50e463f5b
1dfc54bfbe69456]
local_file.file2: Destruction complete after 0s
local_file.file2: Creating...
local_file.file2: Creation complete after 0s [id=fd1b5825bcca
12943b432217d6851801431a7cf0]

Apply complete! Resources: 1 added, 0 changed, 1 destroyed.
PS F:\Tf assignment> 
```

Terraform random to generate random string



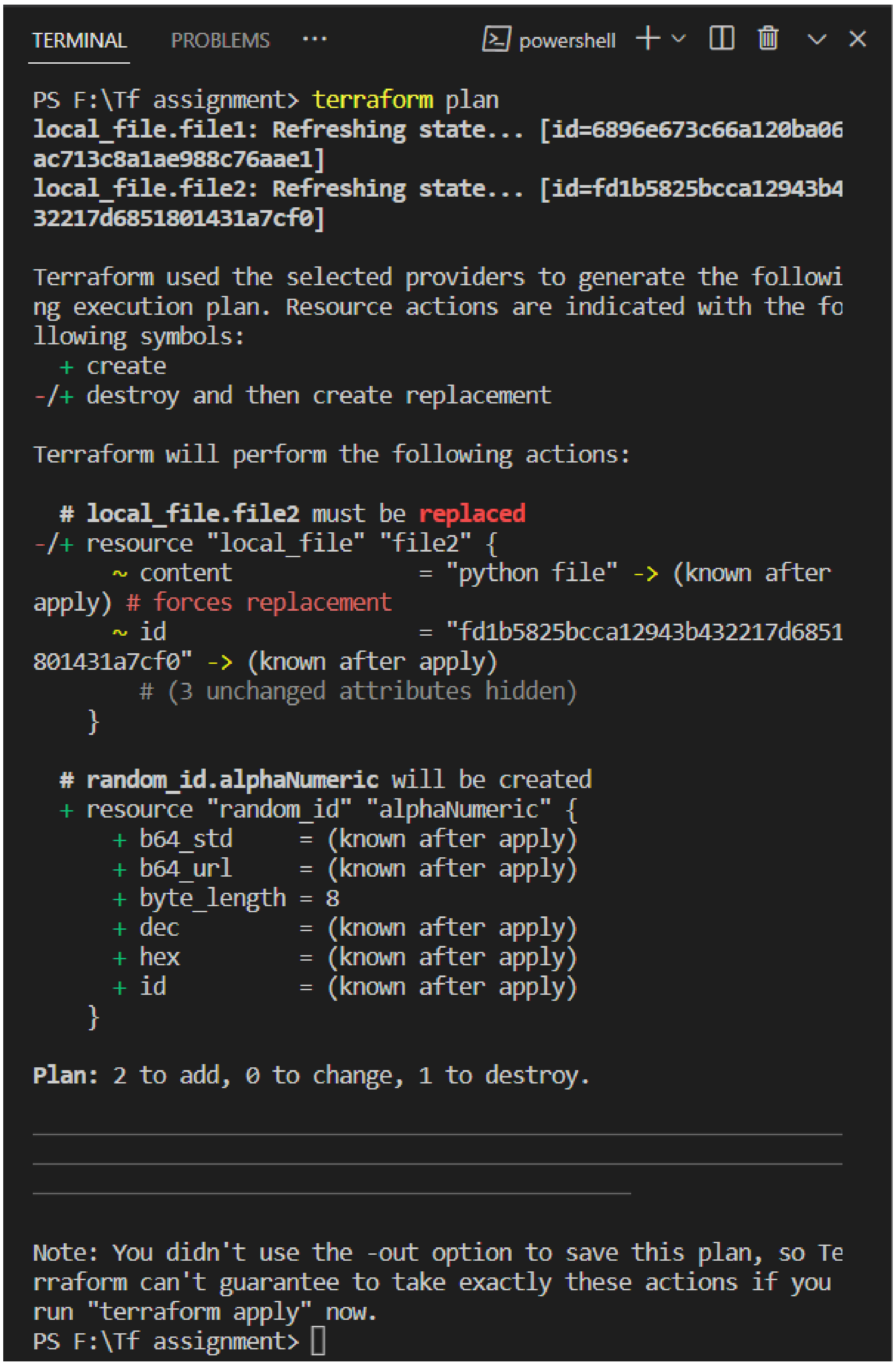
```
# Creating a local resource file
resource "local_file" "file2" {
  content  = "${random_id.alphaNumeric.hex}"
  filename = local.filename
}

# Usage of local values
# Declaring a local value
locals {
  filename = "newfile.py"
  content = "python file"
}

# terraform random
resource "random_id" "alphaNumeric" {
  byte_length = 8

}
```

Now terraform init as we are creating a new resource.



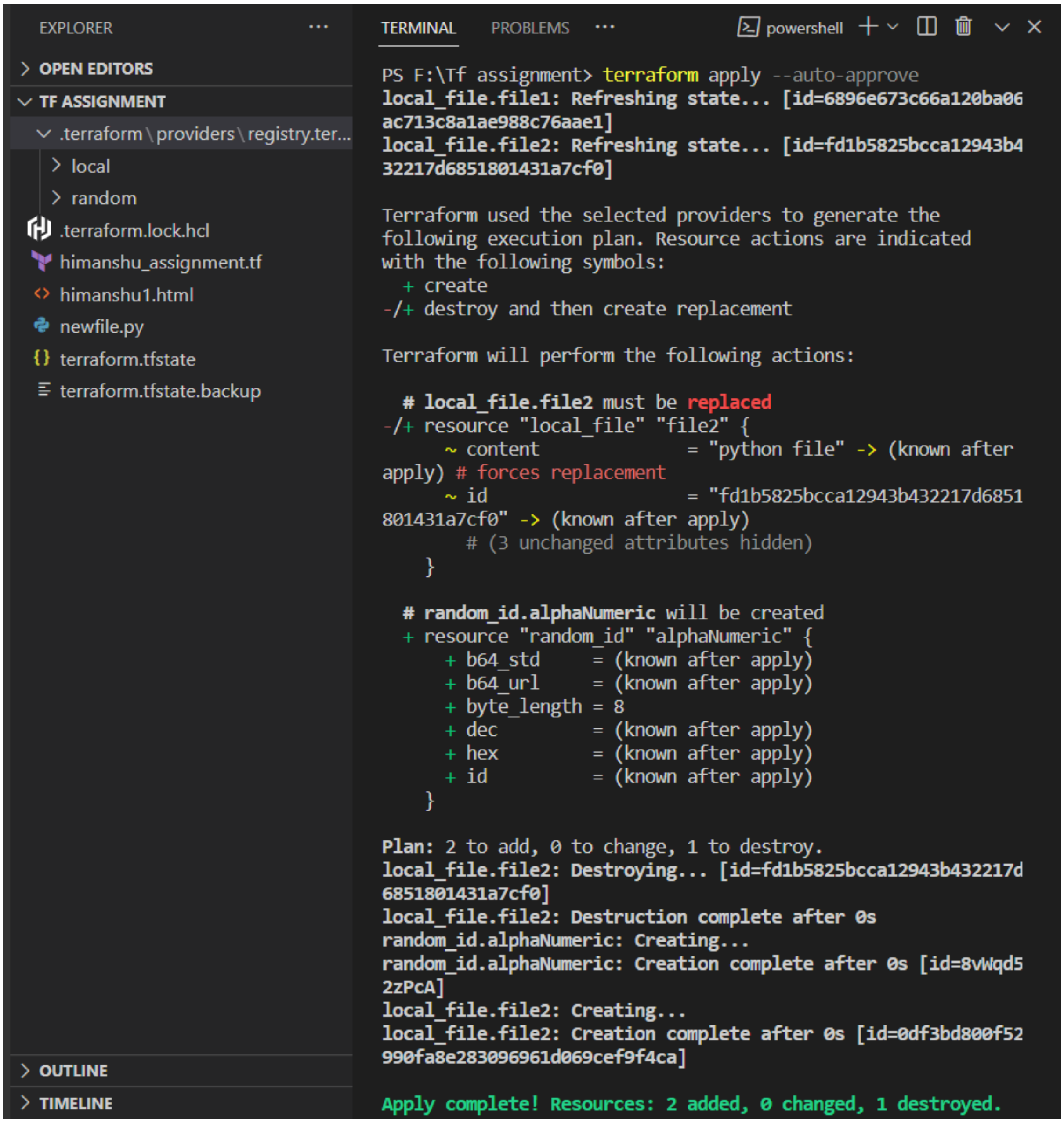
```
PS F:\Tf assignment> terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/random...
- Reusing previous version of hashicorp/local from the depend
ency lock file
- Installing hashicorp/random v3.4.3...
- Installed hashicorp/random v3.4.3 (signed by HashiCorp)
 to make.

Terraform has been successfully initialized!

 to make.

Terraform has been successfully initialized!

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

If you ever set or change modules or backend configuration fo
r 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 F:\Tf assignment> 
```

Terraform plan



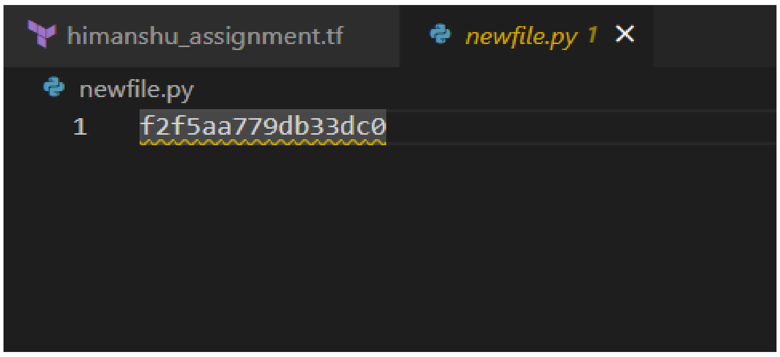
```
PS F:\Tf assignment> terraform plan
local_file.file1: Refreshing state... [id=6896e673c66a120ba06
ac713c8a1ae988c76aae1]
local_file.file2: Refreshing state... [id=fd1b5825bcca12943b4
32217d6851801431a7cf0]

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

Terraform will perform the following actions:

  # local_file.file2 must be replaced
-/+ resource "local_file" "file2" {
      ~ content              = "python file" -> (known after
apply) # forces replacement
      ~ id                   = "fd1b5825bcca12943b432217d6851
801431a7cf0" -> (known after apply)
        # (3 unchanged attributes hidden)
    }

  # random_id.alphaNumeric will be created
  + resource "random_id" "alphaNumeric" {
      + b64_std     = (known after apply)
      + b64_url     = (known after apply)
      + byte_length = 8
      + dec         = (known after apply)
      + hex         = (known after apply)
      + id          = (known after apply)
    }

Plan: 2 to add, 0 to change, 1 to destroy.

─────────────────────────────────────────────────────────────
─────────────────────────────────────────────────────────────
─────────────────────────────────────────────


Note: You didn't use the -out option to save this plan, so Te
rraform can't guarantee to take exactly these actions if you
run "terraform apply" now.
PS F:\Tf assignment> 
```

Terraform apply –auto-approve



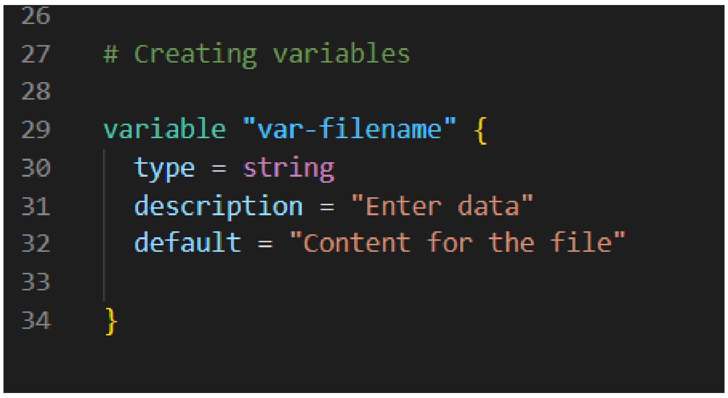
```
PS F:\Tf assignment> terraform apply --auto-approve
local_file.file1: Refreshing state... [id=6896e673c66a120ba06
ac713c8a1ae988c76aae1]
local_file.file2: Refreshing state... [id=fd1b5825bcca12943b4
32217d6851801431a7cf0]

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

Terraform will perform the following actions:

  # local_file.file2 must be replaced
-/+ resource "local_file" "file2" {
      ~ content              = "python file" -> (known after
apply) # forces replacement
      ~ id                   = "fd1b5825bcca12943b432217d6851
801431a7cf0" -> (known after apply)
        # (3 unchanged attributes hidden)
    }

  # random_id.alphaNumeric will be created
  + resource "random_id" "alphaNumeric" {
      + b64_std     = (known after apply)
      + b64_url     = (known after apply)
      + byte_length = 8
      + dec         = (known after apply)
      + hex         = (known after apply)
      + id          = (known after apply)
    }

Plan: 2 to add, 0 to change, 1 to destroy.
local_file.file2: Destroying... [id=fd1b5825bcca12943b432217d
6851801431a7cf0]
local_file.file2: Destruction complete after 0s
random_id.alphaNumeric: Creating...
random_id.alphaNumeric: Creation complete after 0s [id=8vWqd5
2zPcA]
local_file.file2: Creating...
local_file.file2: Creation complete after 0s [id=0df3bd800f52
990fa8e283096961d069cef9f4ca]

Apply complete! Resources: 2 added, 0 changed, 1 destroyed.
```

A random string is generated in the file.



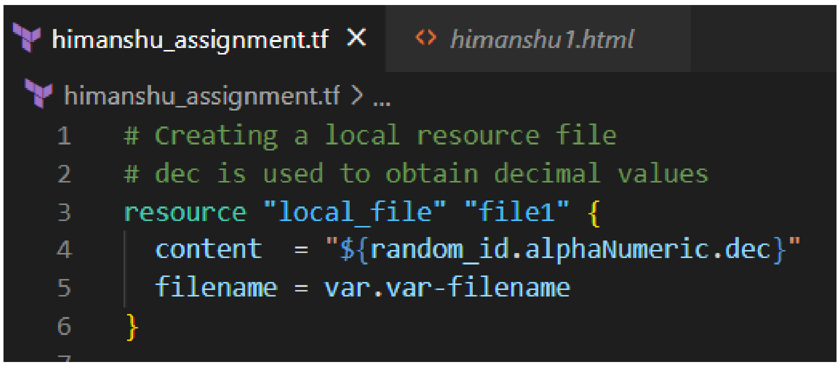
```
f2f5aa779db33dc0
```

Terraform variables



```
# Creating variables

variable "var-filename" {
  type = string
  description = "Enter data"
  default = "Content for the file"

}
```

Using a variable

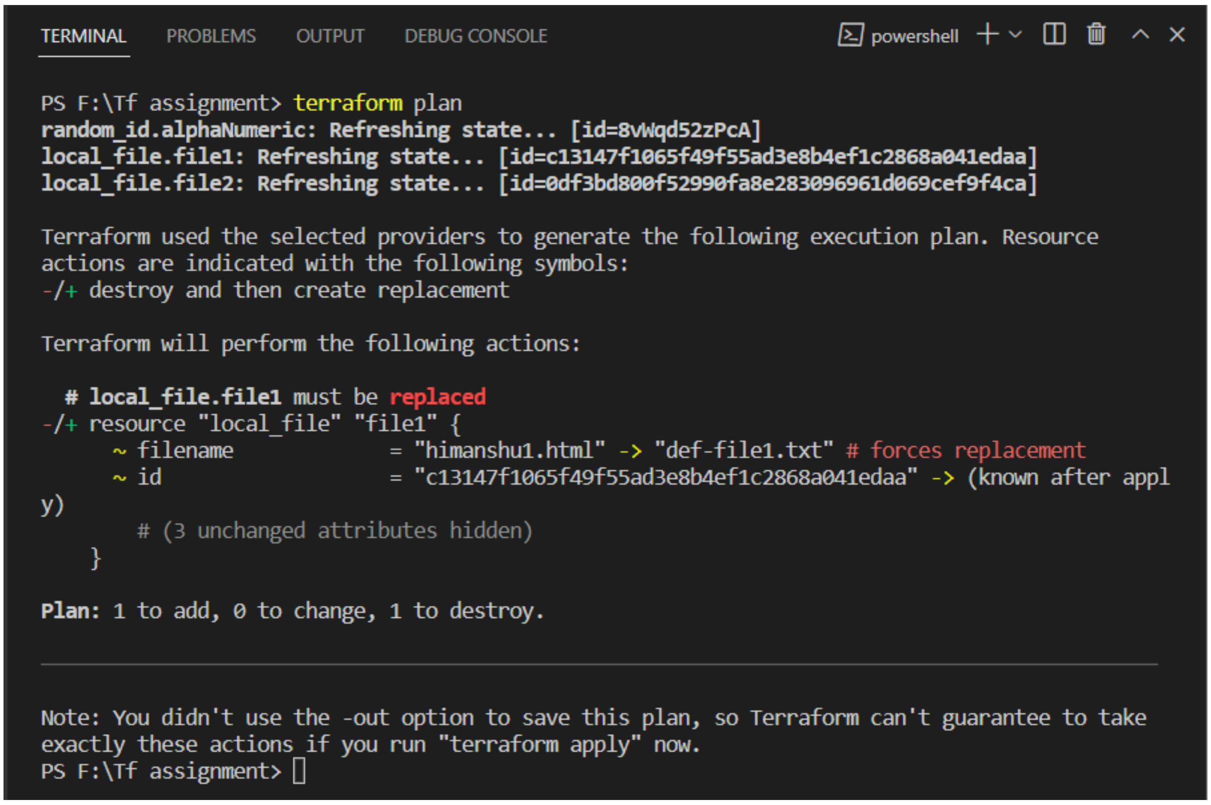


himanshu_assignment.tf ×   himanshu1.html

himanshu_assignment.tf > ...

```
# Creating a local resource file
# dec is used to obtain decimal values
resource "local_file" "file1" {
  content  = "${random_id.alphaNumeric.dec}"
  filename = var.var-filename
}
```

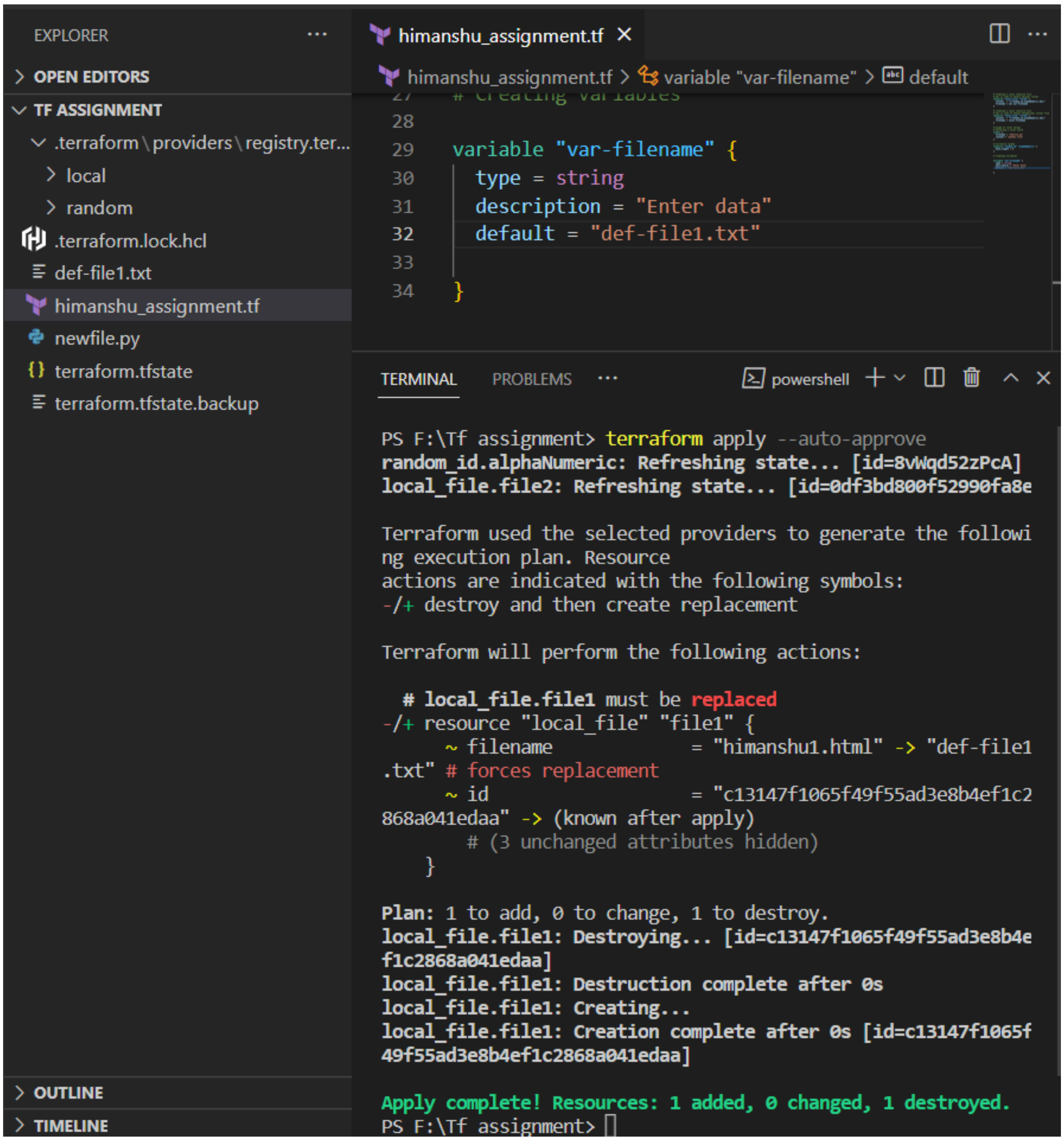
Terraform plan



TERMINAL   PROBLEMS   OUTPUT   DEBUG CONSOLE   powershell

```
PS F:\Tf assignment> terraform plan
random_id.alphaNumeric: Refreshing state... [id=8vWqd52zPcA]
local_file.file1: Refreshing state... [id=c13147f1065f49f55ad3e8b4ef1c2868a041edaa]
local_file.file2: Refreshing state... [id=0df3bd800f52990fa8e283096961d069cef9f4ca]

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.file1 must be replaced
-/+ resource "local_file" "file1" {
      ~ filename             = "himanshu1.html" -> "def-file1.txt" # forces replacement
      ~ id                   = "c13147f1065f49f55ad3e8b4ef1c2868a041edaa" -> (known after appl
y)
        # (3 unchanged attributes hidden)
    }

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

─────────────────────────────────────────────────────────────────────────────

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 F:\Tf assignment> 
```

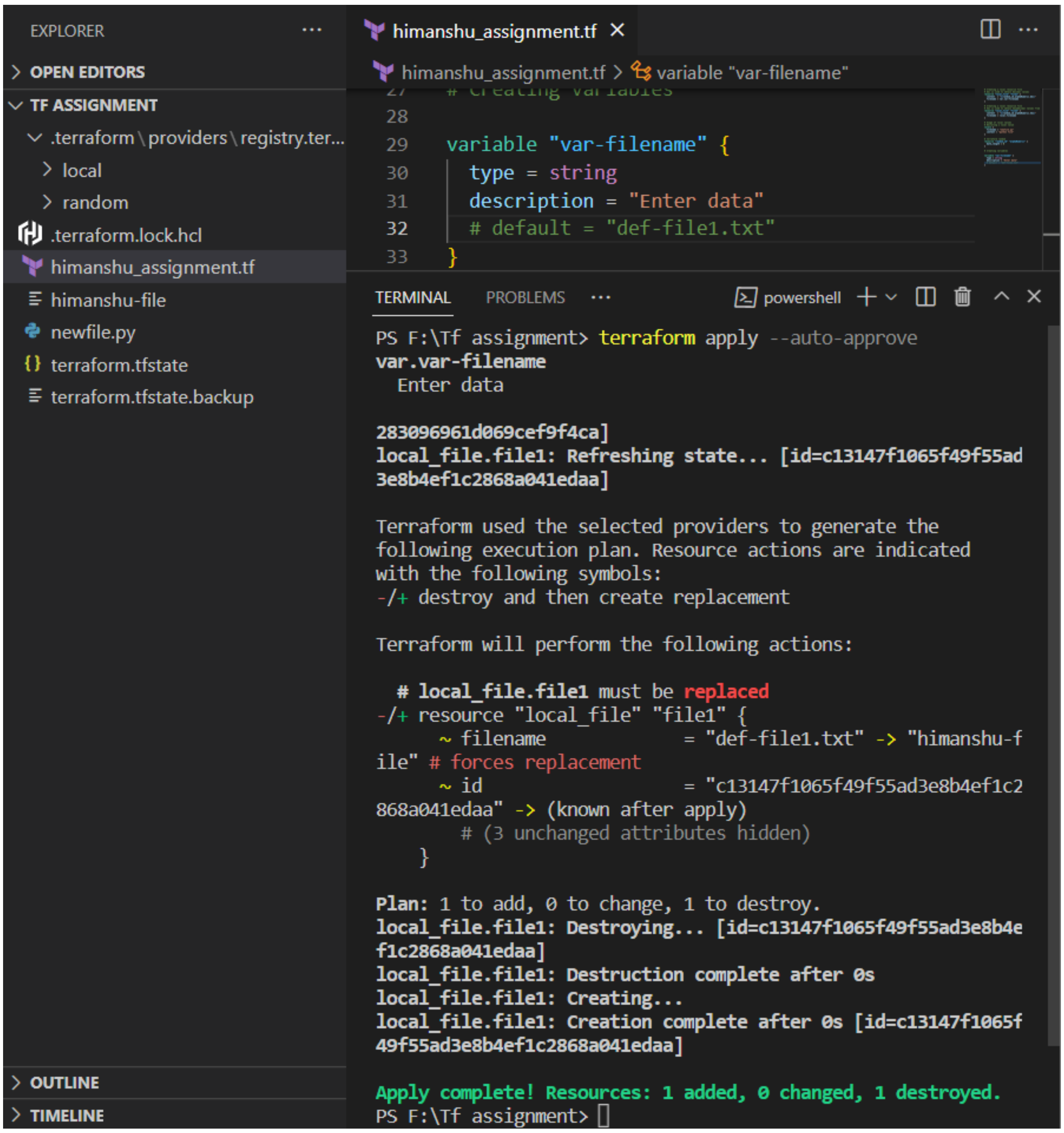
Terraform apply



```
# Creating variables

variable "var-filename" {
  type = string
  description = "Enter data"
  default = "def-file1.txt"

}
```

```
PS F:\Tf assignment> terraform apply --auto-approve
random_id.alphaNumeric: Refreshing state... [id=8vWqd52zPcA]
local_file.file2: Refreshing state... [id=0df3bd800f52990fa8e

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

Terraform will perform the following actions:

  # local_file.file1 must be replaced
-/+ resource "local_file" "file1" {
      ~ filename             = "himanshu1.html" -> "def-file1
.txt" # forces replacement
      ~ id                   = "c13147f1065f49f55ad3e8b4ef1c2
868a041edaa" -> (known after apply)
        # (3 unchanged attributes hidden)
    }

Plan: 1 to add, 0 to change, 1 to destroy.
local_file.file1: Destroying... [id=c13147f1065f49f55ad3e8b4e
f1c2868a041edaa]
local_file.file1: Destruction complete after 0s
local_file.file1: Creating...
local_file.file1: Creation complete after 0s [id=c13147f1065f
49f55ad3e8b4ef1c2868a041edaa]

Apply complete! Resources: 1 added, 0 changed, 1 destroyed.
PS F:\Tf assignment> 
```

If we remove the default from variable and then run terraform apply, then it will ask the user to enter a filename.



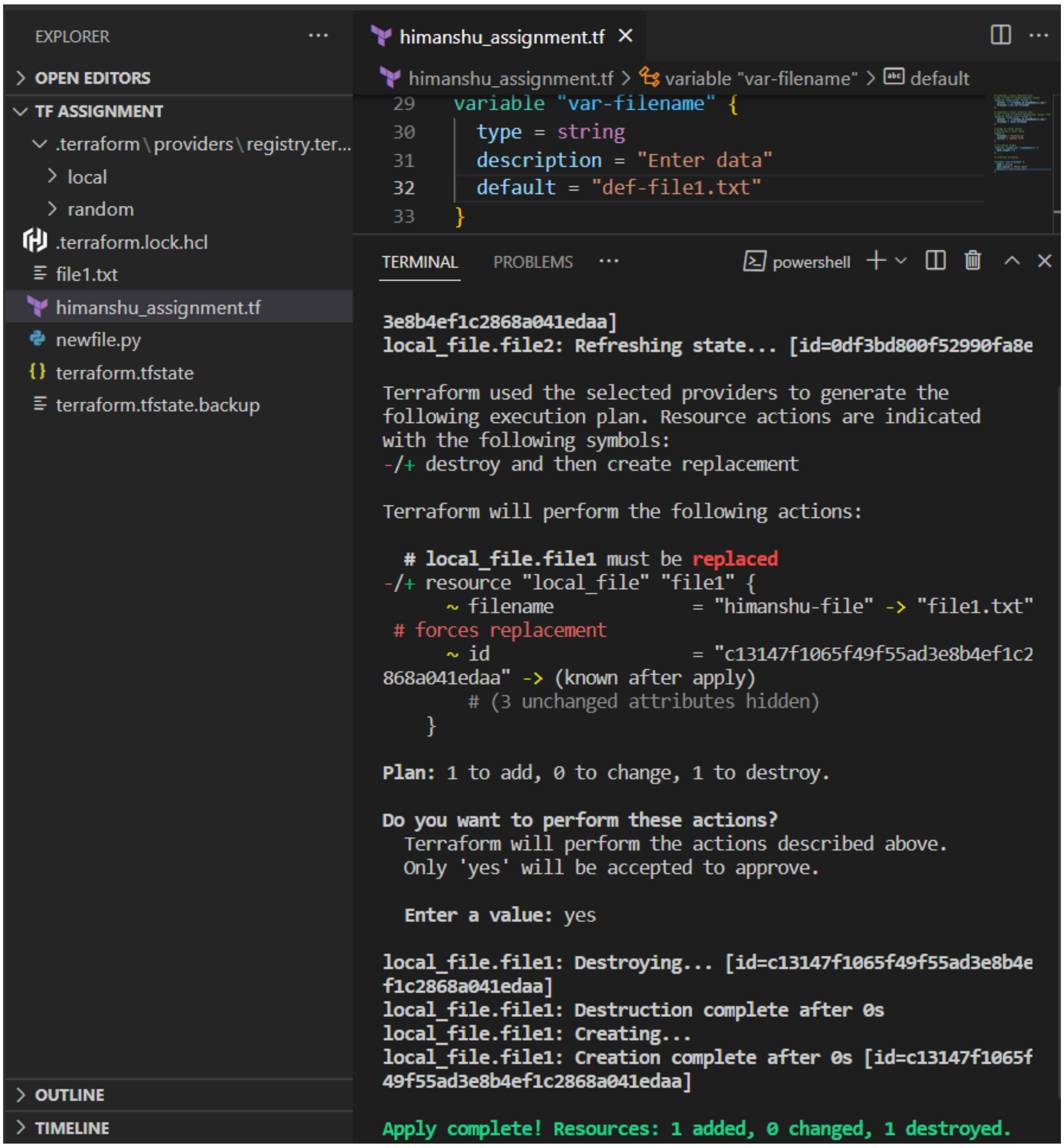
```
# creating variables

variable "var-filename" {
  type = string
  description = "Enter data"
  # default = "def-file1.txt"
}
```

```
PS F:\Tf assignment> terraform apply --auto-approve
var.var-filename
  Enter data

283096961d069cef9f4ca]
local_file.file1: Refreshing state... [id=c13147f1065f49f55ad
3e8b4ef1c2868a041edaa]

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.file1 must be replaced
-/+ resource "local_file" "file1" {
      ~ filename             = "def-file1.txt" -> "himanshu-f
ile" # forces replacement
      ~ id                   = "c13147f1065f49f55ad3e8b4ef1c2
868a041edaa" -> (known after apply)
        # (3 unchanged attributes hidden)
    }

Plan: 1 to add, 0 to change, 1 to destroy.
local_file.file1: Destroying... [id=c13147f1065f49f55ad3e8b4e
f1c2868a041edaa]
local_file.file1: Destruction complete after 0s
local_file.file1: Creating...
local_file.file1: Creation complete after 0s [id=c13147f1065f
49f55ad3e8b4ef1c2868a041edaa]

Apply complete! Resources: 1 added, 0 changed, 1 destroyed.
PS F:\Tf assignment> 
```

If user wish to supply the filename even the default is present.

Syntax: Terraform apply -var variableName=filename.extension

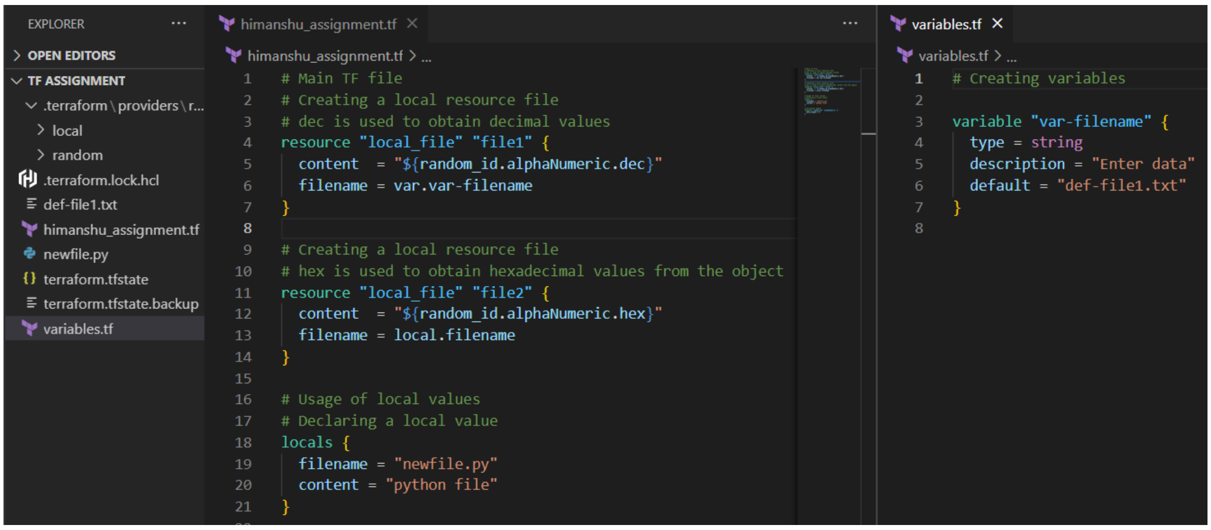
Command: Terraform apply -var var-filename=file1.txt



```
variable "var-filename" {
  type = string
  description = "Enter data"
  default = "def-file1.txt"
}
```

```
3e8b4ef1c2868a041edaa]
local_file.file2: Refreshing state... [id=0df3bd800f52990fa8e

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.file1 must be replaced
-/+ resource "local_file" "file1" {
      ~ filename             = "himanshu-file" -> "file1.txt"
 # forces replacement
      ~ id                   = "c13147f1065f49f55ad3e8b4ef1c2
868a041edaa" -> (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.file1: Destroying... [id=c13147f1065f49f55ad3e8b4e
f1c2868a041edaa]
local_file.file1: Destruction complete after 0s
local_file.file1: Creating...
local_file.file1: Creation complete after 0s [id=c13147f1065f
49f55ad3e8b4ef1c2868a041edaa]

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

Creating the variables externally in different terraform file



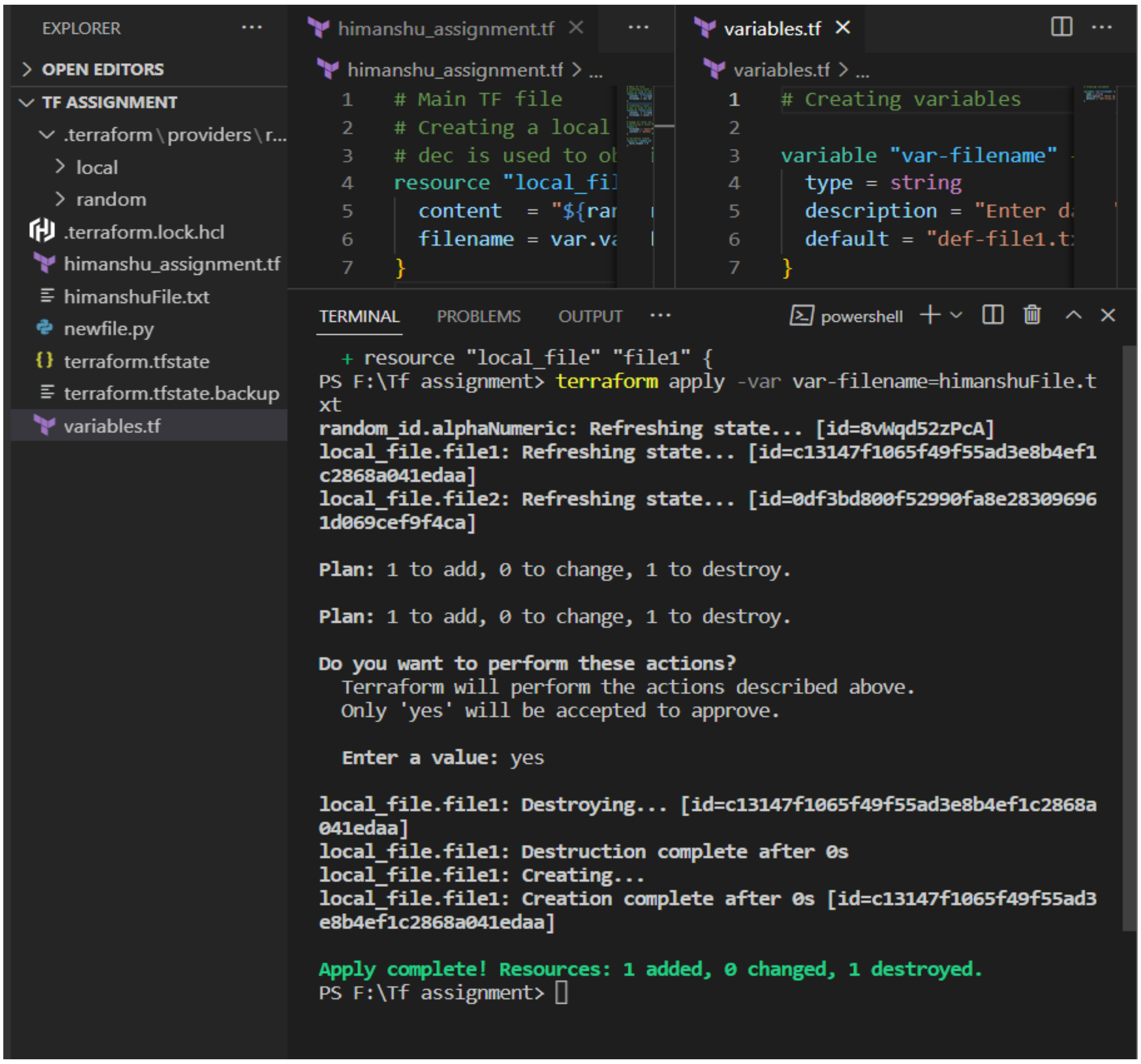
himanshu_assignment.tf

```
# Main TF file
# Creating a local resource file
# dec is used to obtain decimal values
resource "local_file" "file1" {
  content  = "${random_id.alphaNumeric.dec}"
  filename = var.var-filename
}

# Creating a local resource file
# hex is used to obtain hexadecimal values from the object
resource "local_file" "file2" {
  content  = "${random_id.alphaNumeric.hex}"
  filename = local.filename
}

# Usage of local values
# Declaring a local value
locals {
  filename = "newfile.py"
  content = "python file"
}
```

variables.tf

```
# Creating variables

variable "var-filename" {
  type = string
  description = "Enter data"
  default = "def-file1.txt"
}
```

Now doing Terraform apply -var var-filename=himanshuFile.txt



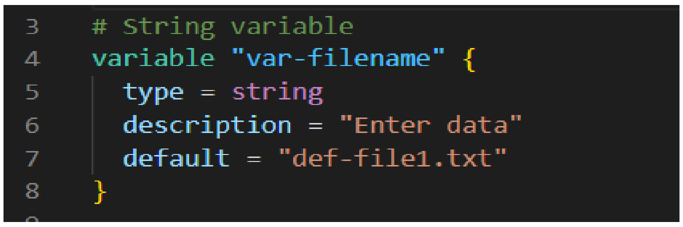
```
  + resource "local_file" "file1" {
PS F:\Tf assignment> terraform apply -var var-filename=himanshuFile.t
xt
random_id.alphaNumeric: Refreshing state... [id=8vWqd52zPcA]
local_file.file1: Refreshing state... [id=c13147f1065f49f55ad3e8b4ef1
c2868a041edaa]
local_file.file2: Refreshing state... [id=0df3bd800f52990fa8e28309696
1d069cef9f4ca]

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

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.file1: Destroying... [id=c13147f1065f49f55ad3e8b4ef1c2868a
041edaa]
local_file.file1: Destruction complete after 0s
local_file.file1: Creating...
local_file.file1: Creation complete after 0s [id=c13147f1065f49f55ad3
e8b4ef1c2868a041edaa]

Apply complete! Resources: 1 added, 0 changed, 1 destroyed.
PS F:\Tf assignment> 
```

The filename changes from def-file1.txt to himanshuFile.txt.

Types of variables in terraform.

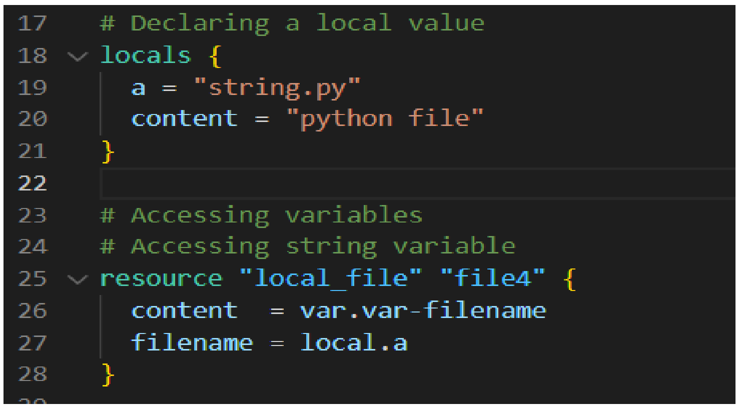
1. **String variable**

Creating a string variable



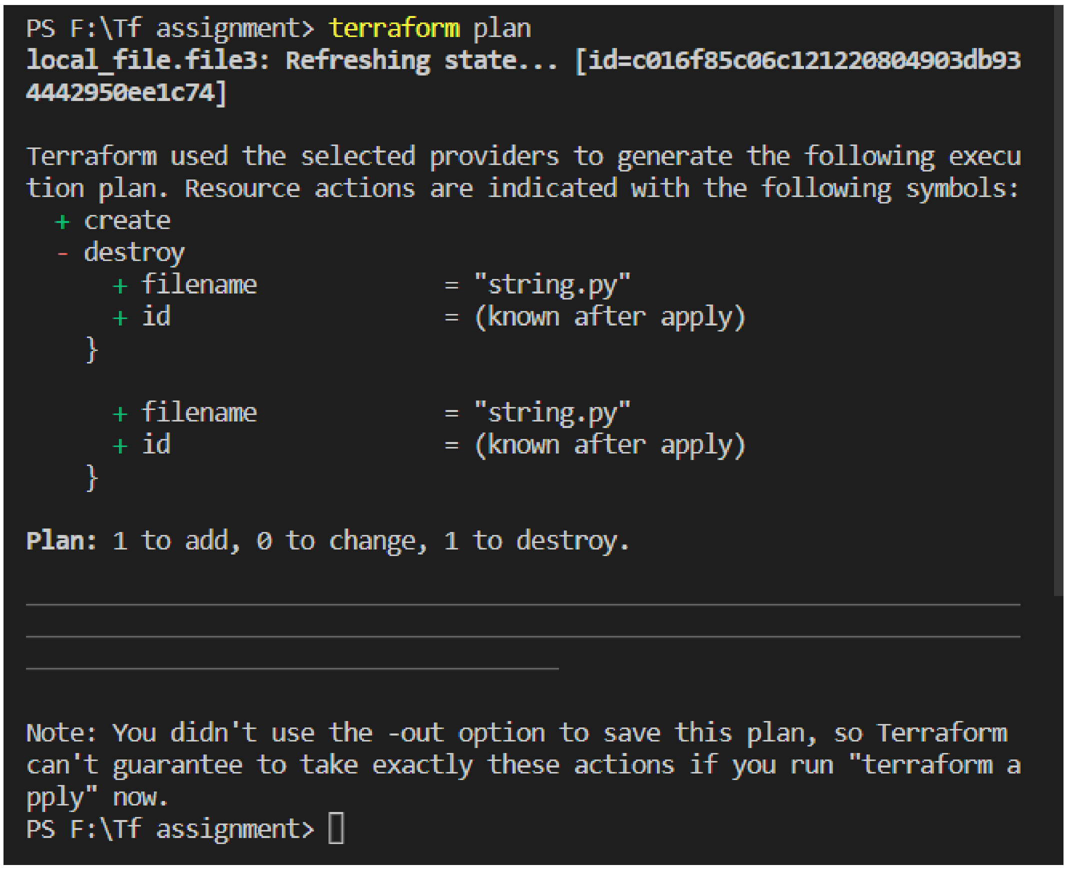
```
# String variable
variable "var-filename" {
  type = string
  description = "Enter data"
  default = "def-file1.txt"
}
```

Accessing the string variable



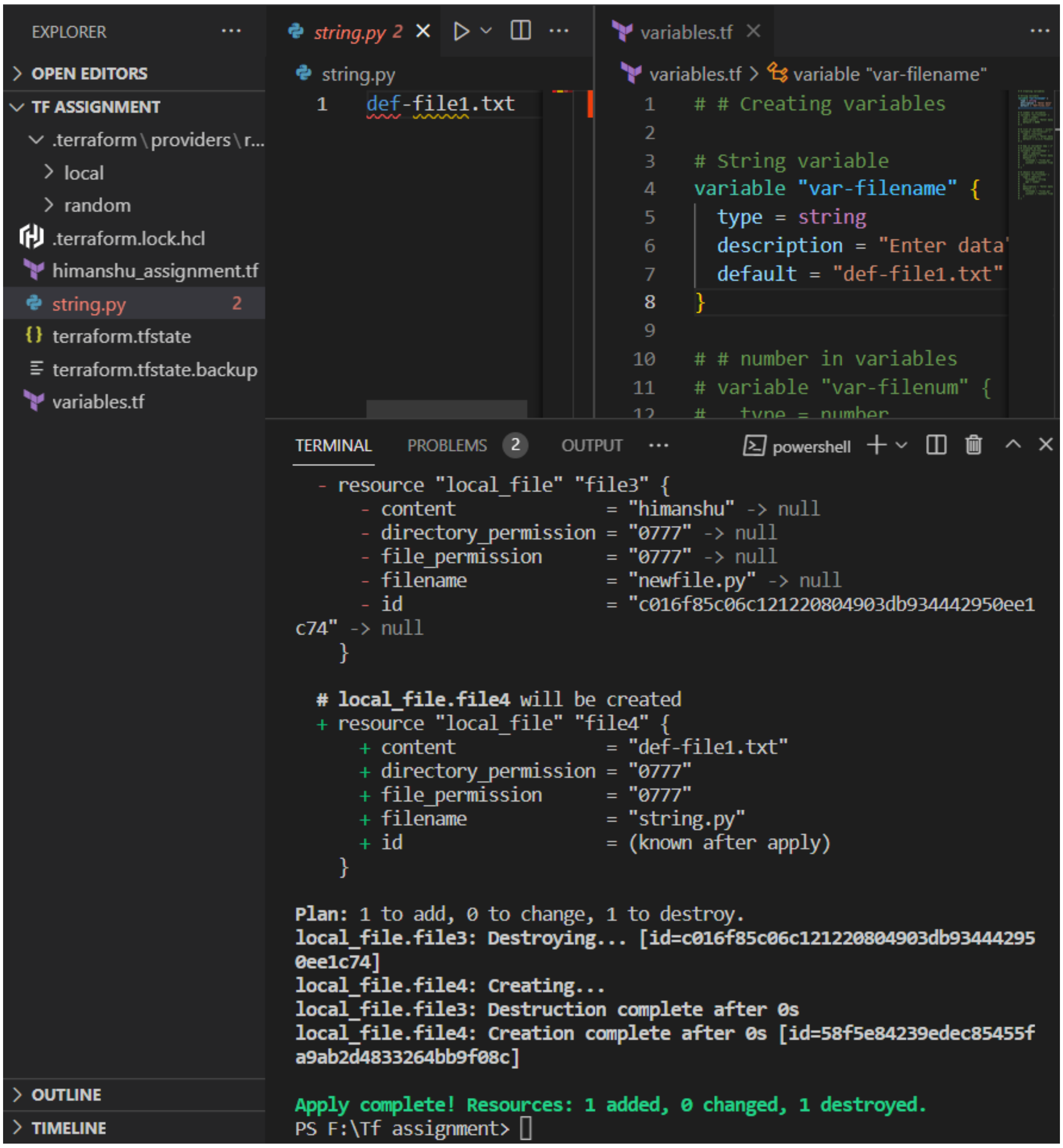
```
# Declaring a local value
locals {
  a = "string.py"
  content = "python file"
}

# Accessing variables
# Accessing string variable
resource "local_file" "file4" {
  content  = var.var-filename
  filename = local.a
}
```

After that executing terraform plan will show:



```
PS F:\Tf assignment> terraform plan
local_file.file3: Refreshing state... [id=c016f85c06c121220804903db93
4442950ee1c74]

Terraform used the selected providers to generate the following execu
tion plan. Resource actions are indicated with the following symbols:
  + create
  - destroy
      + filename             = "string.py"
      + id                   = (known after apply)
    }

      + filename             = "string.py"
      + id                   = (known after apply)
    }

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




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 a
pply" now.
PS F:\Tf assignment> 
```

Then terraform apply –auto-approve:



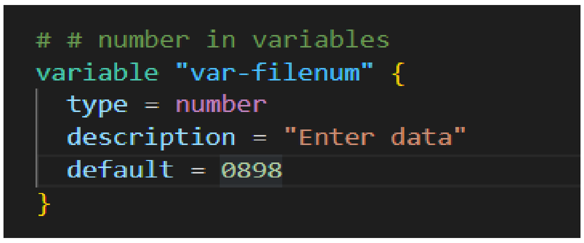
```
def-file1.txt
```

```
# # Creating variables

# String variable
variable "var-filename" {
  type = string
  description = "Enter data'
  default = "def-file1.txt"
}

# # number in variables
# variable "var-filenum" {
#   type = number
```

```
  - resource "local_file" "file3" {
      - content              = "himanshu" -> null
      - directory_permission = "0777" -> null
      - file_permission      = "0777" -> null
      - filename             = "newfile.py" -> null
      - id                   = "c016f85c06c121220804903db934442950ee1
c74" -> null
    }

  # local_file.file4 will be created
  + resource "local_file" "file4" {
      + content              = "def-file1.txt"
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "string.py"
      + id                   = (known after apply)
    }

Plan: 1 to add, 0 to change, 1 to destroy.
local_file.file3: Destroying... [id=c016f85c06c121220804903db93444295
0ee1c74]
local_file.file4: Creating...
local_file.file3: Destruction complete after 0s
local_file.file4: Creation complete after 0s [id=58f5e84239edec85455f
a9ab2d4833264bb9f08c]

Apply complete! Resources: 1 added, 0 changed, 1 destroyed.
PS F:\Tf assignment>
```

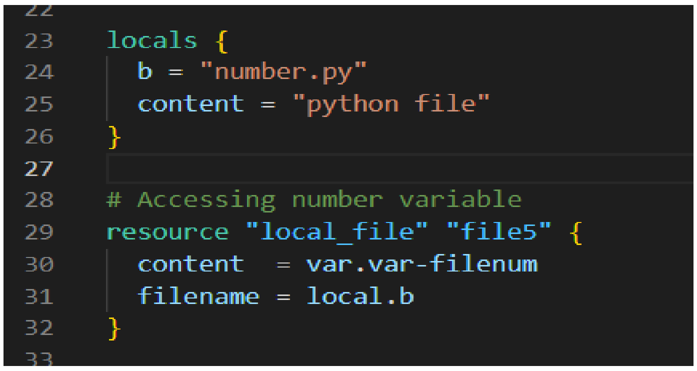
2. Integer/ number variable.

Creating a number variable



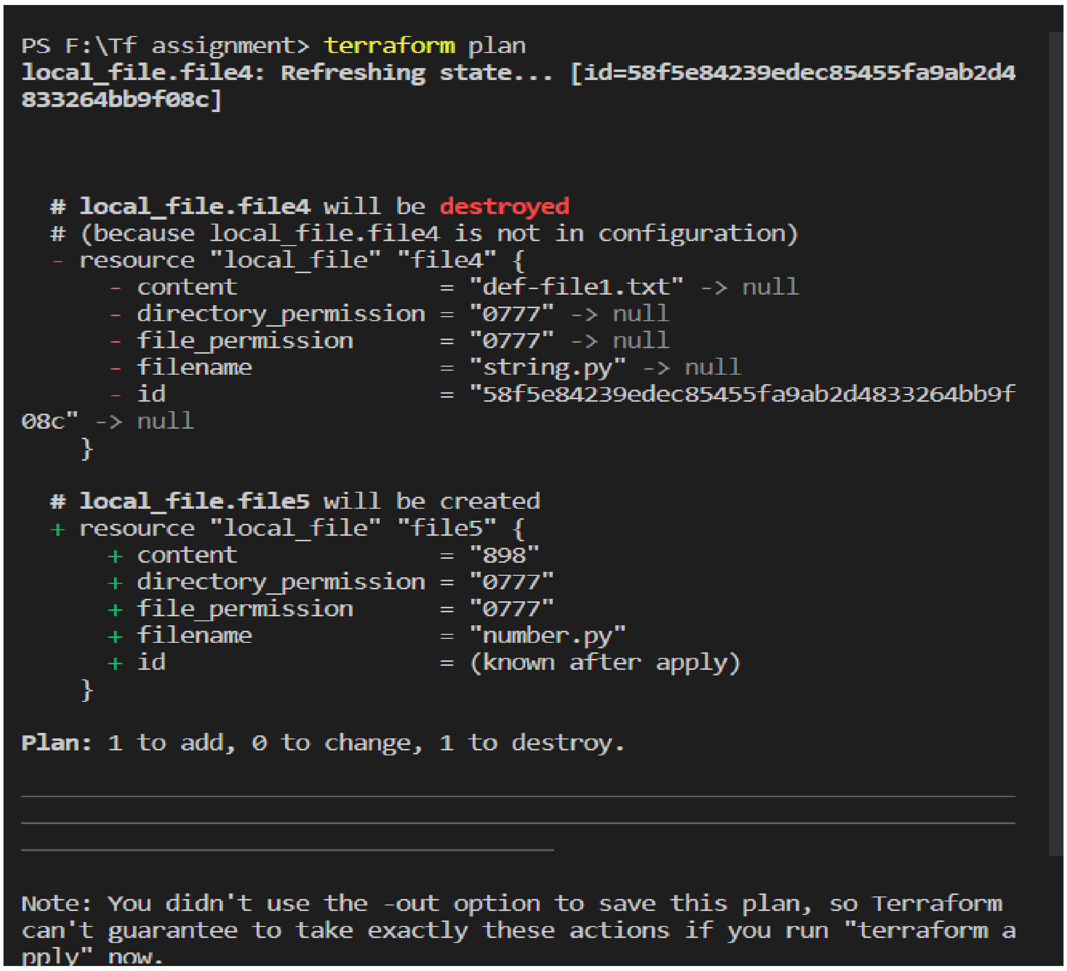
```
# # number in variables
variable "var-filenum" {
  type = number
  description = "Enter data"
  default = 0898
}
```

Accessing the number variable



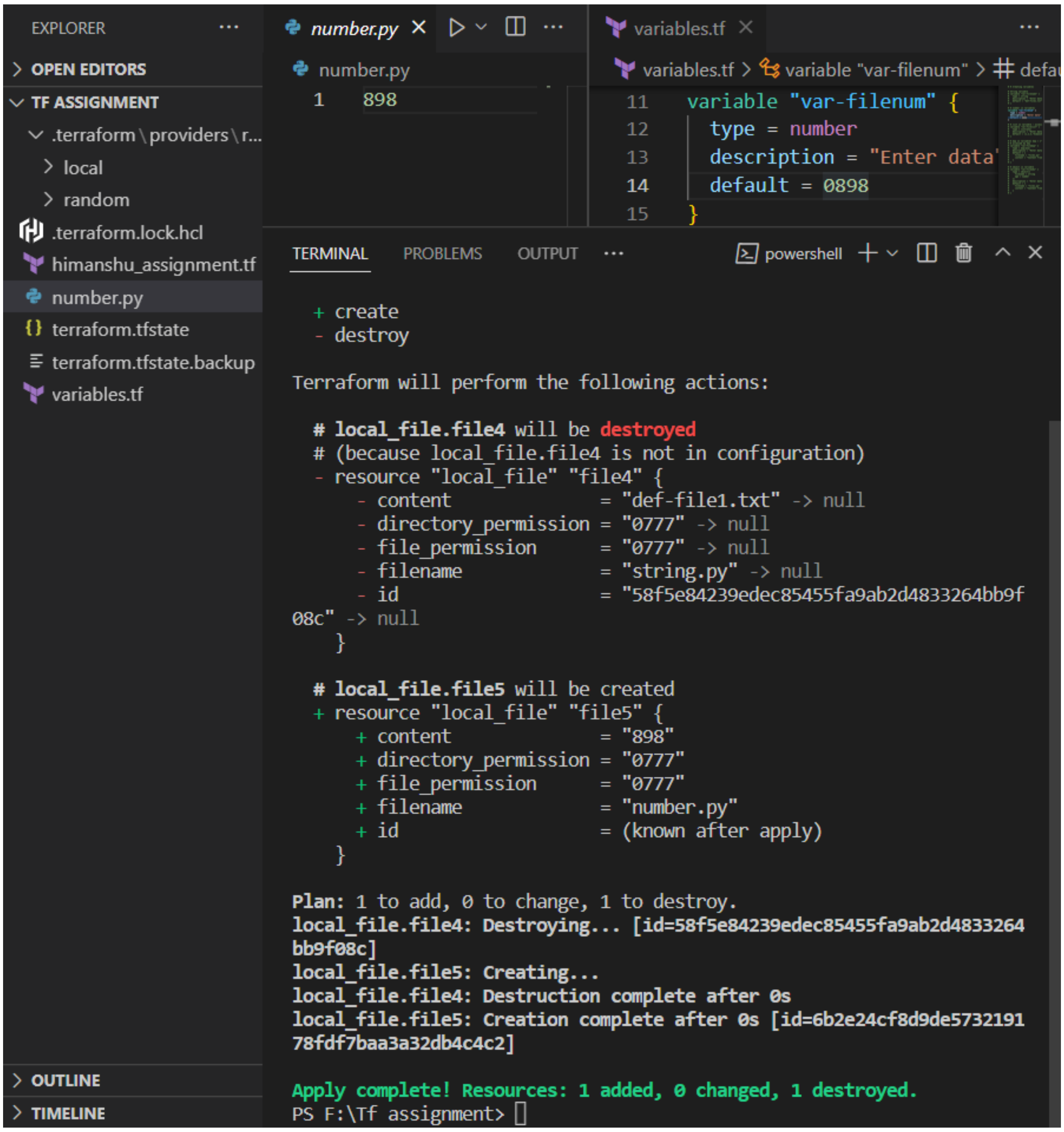
```
locals {
  b = "number.py"
  content = "python file"
}

# Accessing number variable
resource "local_file" "file5" {
  content  = var.var-filenum
  filename = local.b
}
```

After that executing terraform plan will show:



```
PS F:\Tf assignment> terraform plan
local_file.file4: Refreshing state... [id=58f5e84239edec85455fa9ab2d4
833264bb9f08c]


  # local_file.file4 will be destroyed
  # (because local_file.file4 is not in configuration)
  - resource "local_file" "file4" {
      - content              = "def-file1.txt" -> null
      - directory_permission = "0777" -> null
      - file_permission      = "0777" -> null
      - filename             = "string.py" -> null
      - id                   = "58f5e84239edec85455fa9ab2d4833264bb9f
08c" -> null
    }

  # local_file.file5 will be created
  + resource "local_file" "file5" {
      + content              = "898"
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "number.py"
      + id                   = (known after apply)
    }

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

─────────────────────────────────────────────────────────────────────

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 a
pply" now.
```

Then terraform apply –auto-approve:



number.py

```
898
```

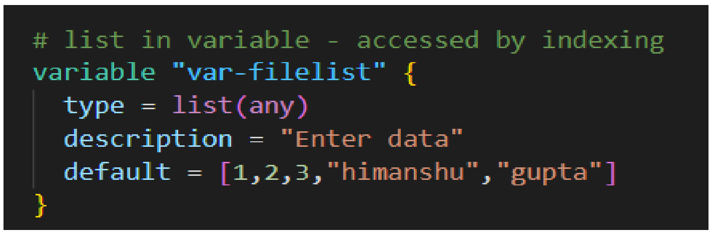
variables.tf > variable "var-filenum" > defa

```
variable "var-filenum" {
  type = number
  description = "Enter data'
  default = 0898
}
```

```
  + create
  - destroy

Terraform will perform the following actions:

  # local_file.file4 will be destroyed
  # (because local_file.file4 is not in configuration)
  - resource "local_file" "file4" {
      - content              = "def-file1.txt" -> null
      - directory_permission = "0777" -> null
      - file_permission      = "0777" -> null
      - filename             = "string.py" -> null
      - id                   = "58f5e84239edec85455fa9ab2d4833264bb9f
08c" -> null
    }

  # local_file.file5 will be created
  + resource "local_file" "file5" {
      + content              = "898"
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "number.py"
      + id                   = (known after apply)
    }

Plan: 1 to add, 0 to change, 1 to destroy.
local_file.file4: Destroying... [id=58f5e84239edec85455fa9ab2d4833264
bb9f08c]
local_file.file5: Creating...
local_file.file4: Destruction complete after 0s
local_file.file5: Creation complete after 0s [id=6b2e24cf8d9de5732191
78fdf7baa3a32db4c4c2]

Apply complete! Resources: 1 added, 0 changed, 1 destroyed.
PS F:\Tf assignment>
```

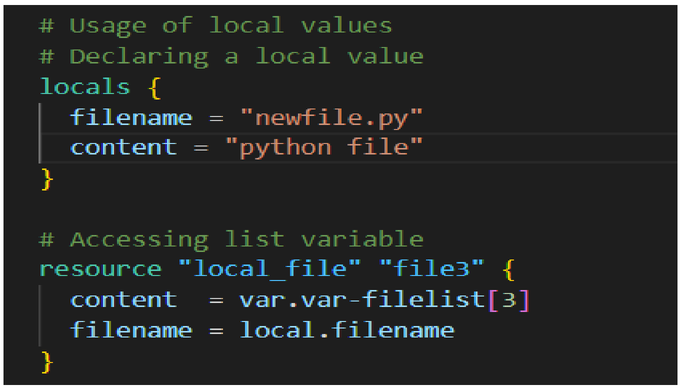
3. List variable

Creating a list variable



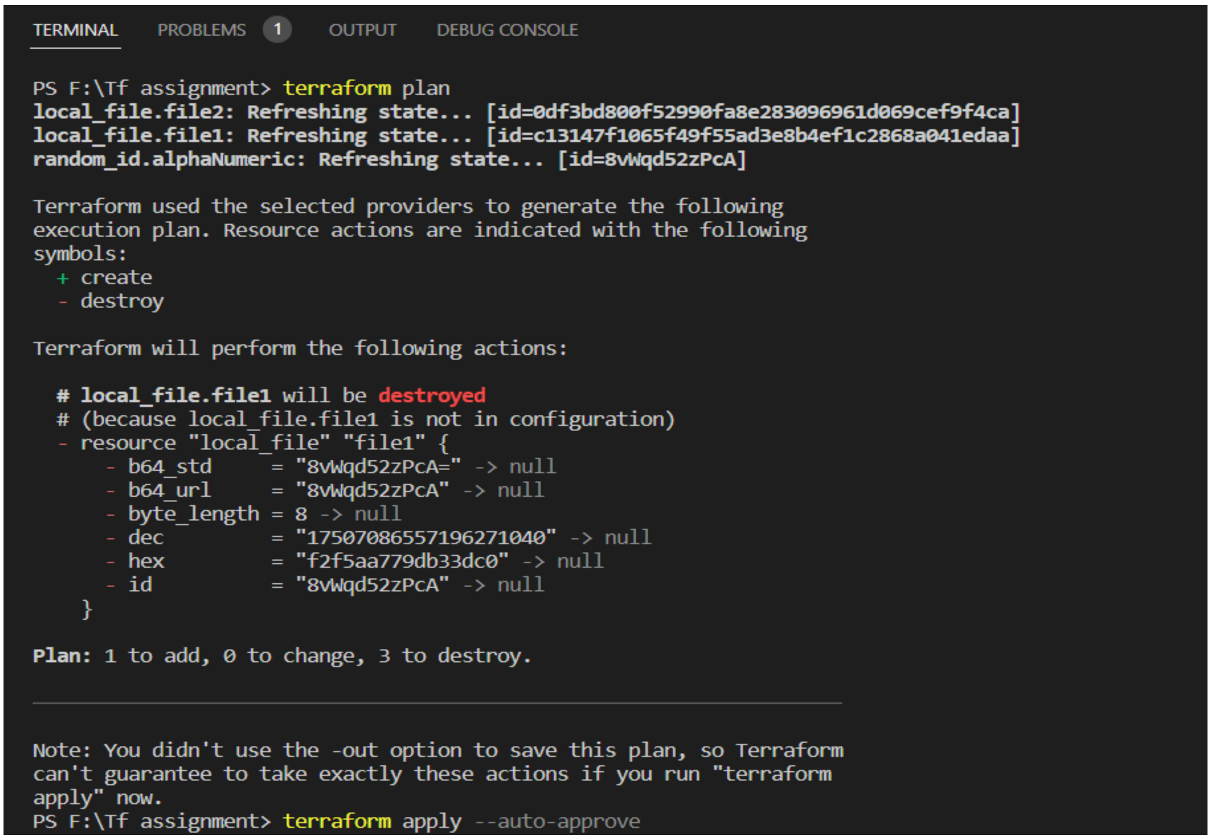
```
# list in variable - accessed by indexing
variable "var-filelist" {
  type = list(any)
  description = "Enter data"
  default = [1,2,3,"himanshu","gupta"]
}
```

Accessing the list variable



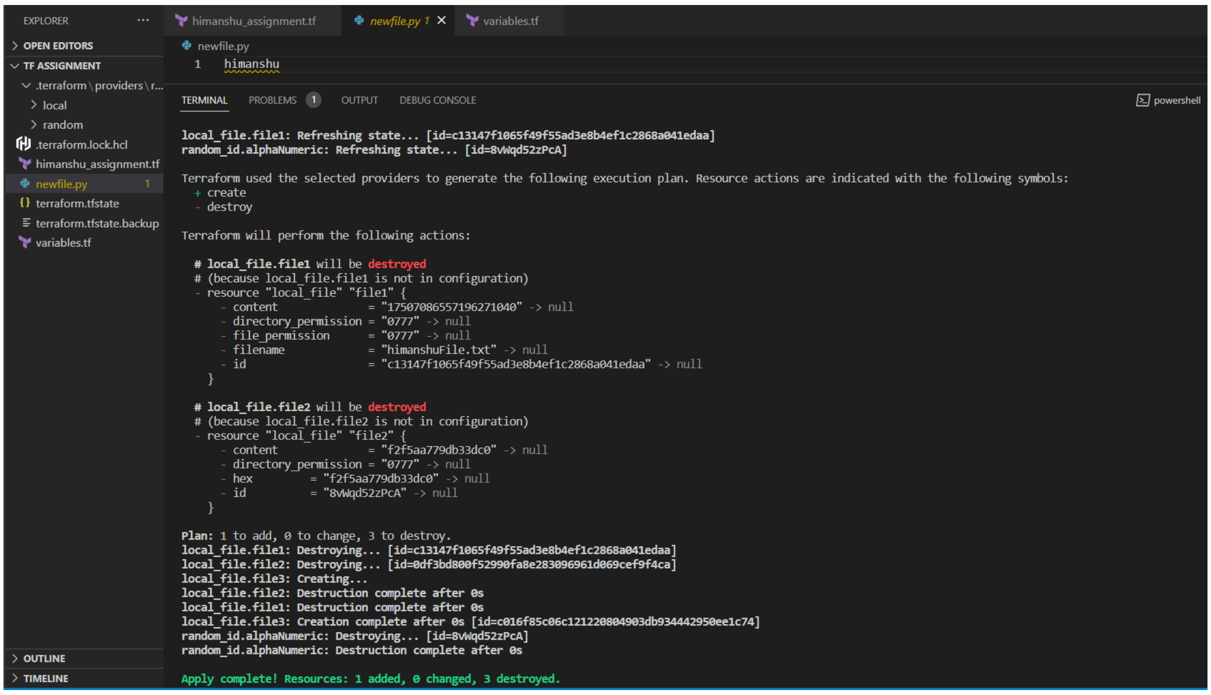
```
# Usage of local values
# Declaring a local value
locals {
  filename = "newfile.py"
  content = "python file"
}

# Accessing list variable
resource "local_file" "file3" {
  content  = var.var-filelist[3]
  filename = local.filename
}
```

After that executing terraform plan will show:



```
PS F:\Tf assignment> terraform plan
local_file.file2: Refreshing state... [id=0df3bd800f52990fa8e283096961d069cef9f4ca]
local_file.file1: Refreshing state... [id=c13147f1065f49f55ad3e8b4ef1c2868a041edaa]
random_id.alphaNumeric: Refreshing state... [id=8vWqd52zPcA]

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

Terraform will perform the following actions:

  # local_file.file1 will be destroyed
  # (because local_file.file1 is not in configuration)
  - resource "local_file" "file1" {
      - b64_std     = "8vWqd52zPcA=" -> null
      - b64_url     = "8vWqd52zPcA" -> null
      - byte_length = 8 -> null
      - dec         = "17507086557196271040" -> null
      - hex         = "f2f5aa779db33dc0" -> null
      - id          = "8vWqd52zPcA" -> null
    }

Plan: 1 to add, 0 to change, 3 to destroy.

─────────────────────────────────────────────────────────────────────

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 F:\Tf assignment> terraform apply --auto-approve
```

Then terraform apply –auto-approve:

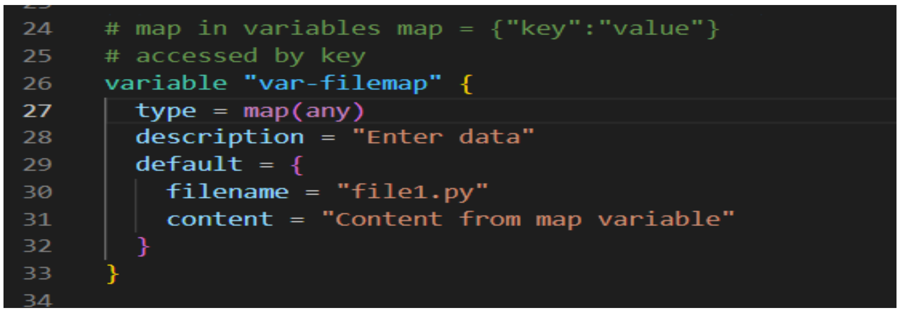


```
newfile.py
himanshu
```

```
local_file.file1: Refreshing state... [id=c13147f1065f49f55ad3e8b4ef1c2868a041edaa]
random_id.alphaNumeric: Refreshing state... [id=8vWqd52zPcA]

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

Terraform will perform the following actions:

  # local_file.file1 will be destroyed
  # (because local_file.file1 is not in configuration)
  - resource "local_file" "file1" {
      - content              = "17507086557196271040" -> null
      - directory_permission = "0777" -> null
      - file_permission      = "0777" -> null
      - filename             = "himanshuFile.txt" -> null
      - id                   = "c13147f1065f49f55ad3e8b4ef1c2868a041edaa" -> null
    }

  # local_file.file2 will be destroyed
  # (because local_file.file2 is not in configuration)
  - resource "local_file" "file2" {
      - content              = "f2f5aa779db33dc0" -> null
      - directory_permission = "0777" -> null
      - hex         = "f2f5aa779db33dc0" -> null
      - id          = "8vWqd52zPcA" -> null
    }

Plan: 1 to add, 0 to change, 3 to destroy.
local_file.file1: Destroying... [id=c13147f1065f49f55ad3e8b4ef1c2868a041edaa]
local_file.file2: Destroying... [id=0df3bd800f52990fa8e283096961d069cef9f4ca]
local_file.file3: Creating...
local_file.file2: Destruction complete after 0s
local_file.file1: Destruction complete after 0s
local_file.file3: Creation complete after 0s [id=c016f85c06c121220804903db934442950ee1c74]
random_id.alphaNumeric: Destroying... [id=8vWqd52zPcA]
random_id.alphaNumeric: Destruction complete after 0s

Apply complete! Resources: 1 added, 0 changed, 3 destroyed.
```

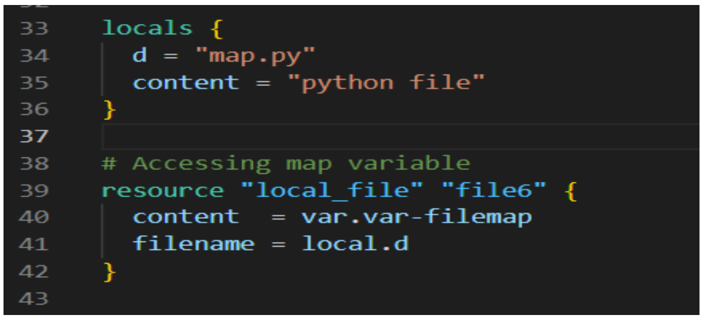
4. Map variable

Creating a map variable



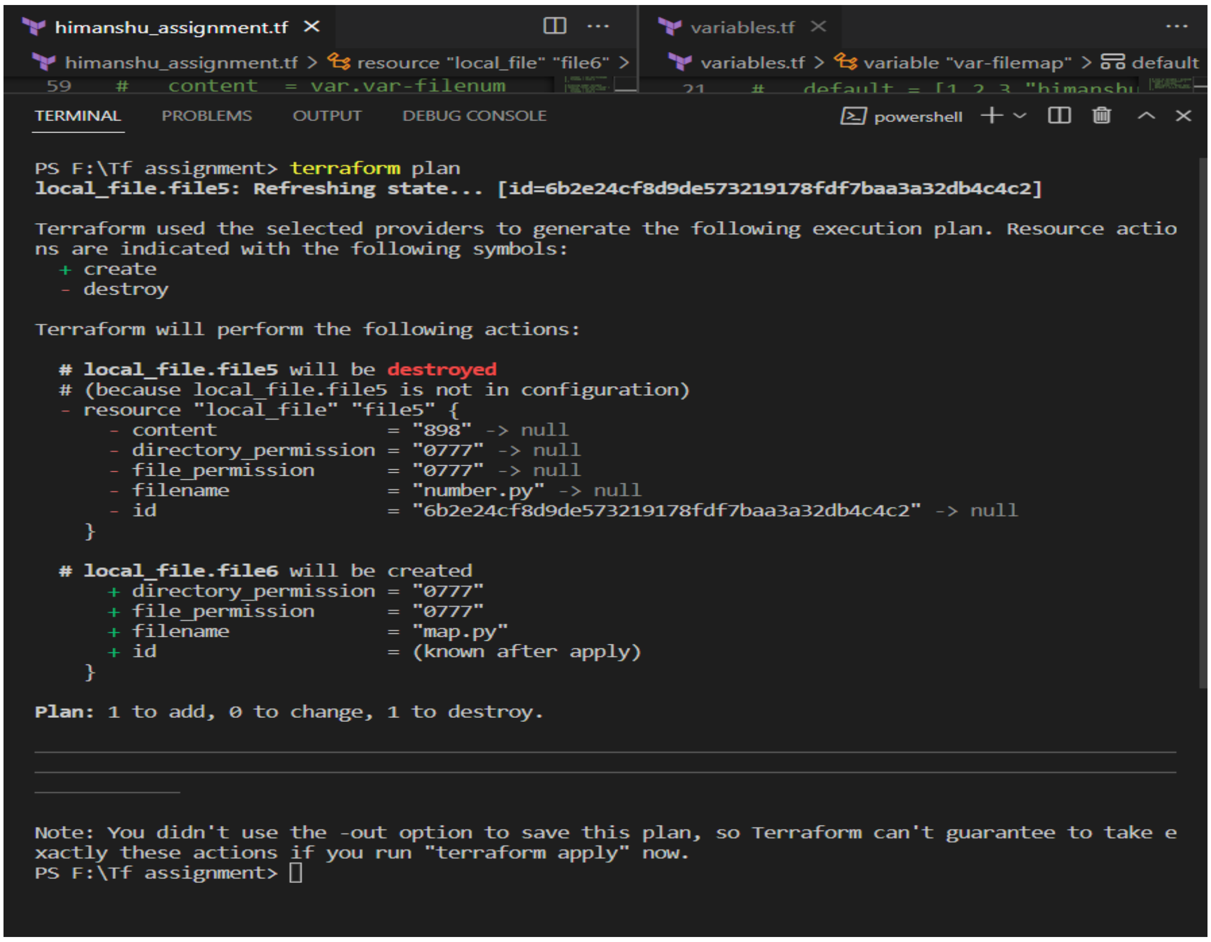
```
# map in variables map = {"key":"value"}
# accessed by key
variable "var-filemap" {
  type = map(any)
  description = "Enter data"
  default = {
    filename = "file1.py"
    content = "Content from map variable"
  }
}
```

Accessing the map variable



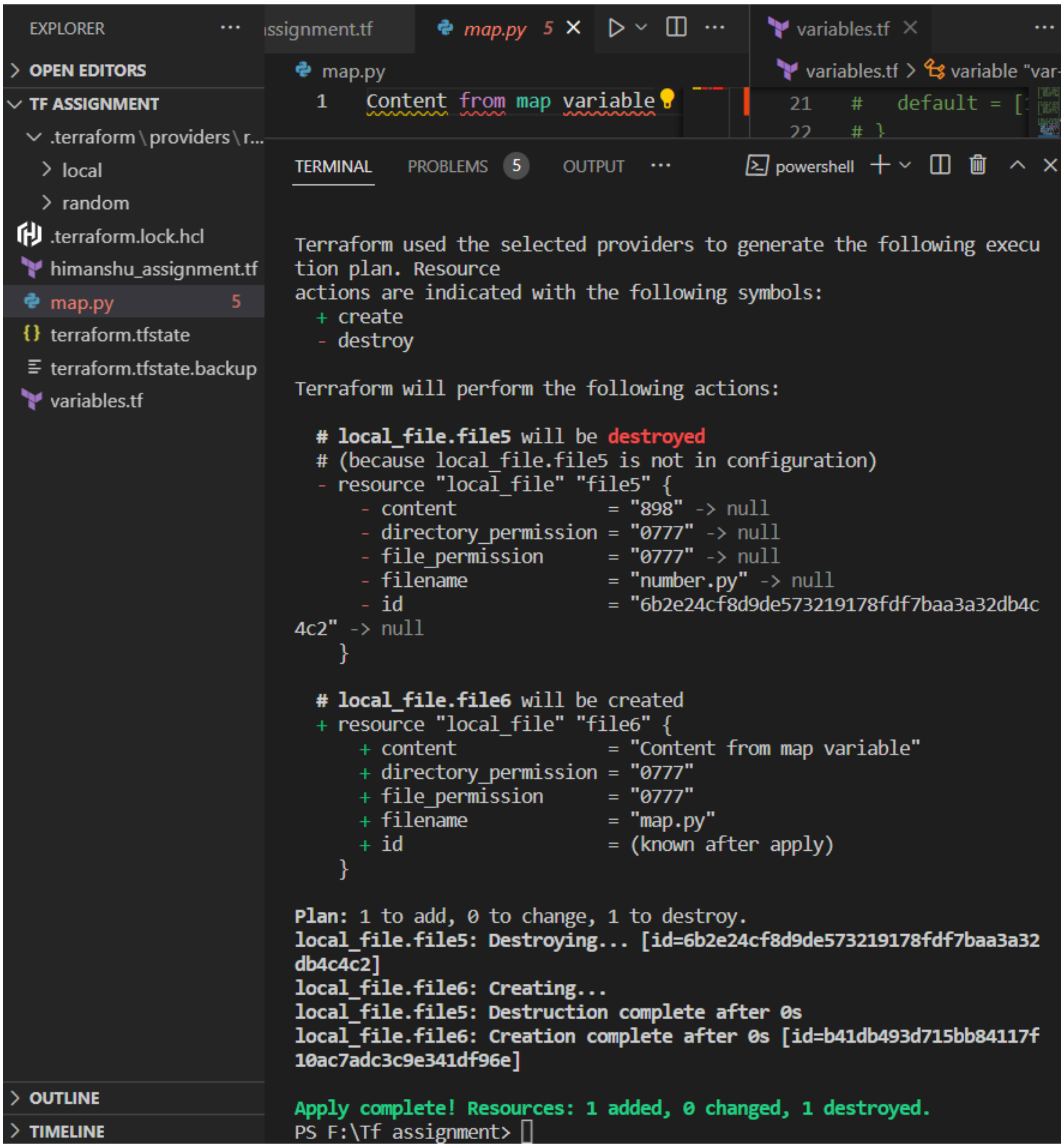
```
locals {
  d = "map.py"
  content = "python file"
}

# Accessing map variable
resource "local_file" "file6" {
  content  = var.var-filemap
  filename = local.d
}
```

After that executing terraform plan will show:


himanshu_assignment.tf
variables.tf
himanshu_assignment.tf > resource "local_file" "file6"
variables.tf > variable "var-filemap" > default
TERMINAL
PROBLEMS
OUTPUT
DEBUG CONSOLE
powershell

```
PS F:\Tf assignment> terraform plan
local_file.file5: Refreshing state... [id=6b2e24cf8d9de573219178fdf7baa3a32db4c4c2]

Terraform used the selected providers to generate the following execution plan. Resource actio
ns are indicated with the following symbols:
  + create
  - destroy

Terraform will perform the following actions:

  # local_file.file5 will be destroyed
  # (because local_file.file5 is not in configuration)
  - resource "local_file" "file5" {
      - content              = "898" -> null
      - directory_permission = "0777" -> null
      - file_permission      = "0777" -> null
      - filename             = "number.py" -> null
      - id                   = "6b2e24cf8d9de573219178fdf7baa3a32db4c4c2" -> null
    }

  # local_file.file6 will be created
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "map.py"
      + id                   = (known after apply)
    }

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

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take e
xactly these actions if you run "terraform apply" now.
PS F:\Tf assignment> 
```

Then terraform apply –auto-approve:



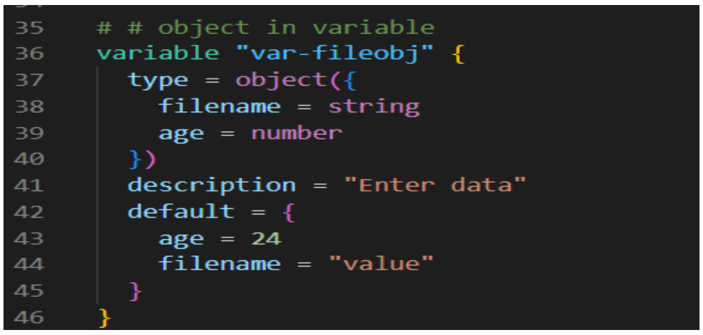
```
1  Content from map variable
```

```
21    #    default = [
22    # }
```

```
Terraform used the selected providers to generate the following execu
tion plan. Resource
actions are indicated with the following symbols:
  + create
  - destroy

Terraform will perform the following actions:

  # local_file.file5 will be destroyed
  # (because local_file.file5 is not in configuration)
  - resource "local_file" "file5" {
      - content              = "898" -> null
      - directory_permission = "0777" -> null
      - file_permission      = "0777" -> null
      - filename             = "number.py" -> null
      - id                   = "6b2e24cf8d9de573219178fdf7baa3a32db4c
4c2" -> null
    }

  # local_file.file6 will be created
  + resource "local_file" "file6" {
      + content              = "Content from map variable"
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "map.py"
      + id                   = (known after apply)
    }

Plan: 1 to add, 0 to change, 1 to destroy.
local_file.file5: Destroying... [id=6b2e24cf8d9de573219178fdf7baa3a32
db4c4c2]
local_file.file6: Creating...
local_file.file5: Destruction complete after 0s
local_file.file6: Creation complete after 0s [id=b41db493d715bb84117f
10ac7adc3c9e341df96e]

Apply complete! Resources: 1 added, 0 changed, 1 destroyed.
PS F:\Tf assignment>
```

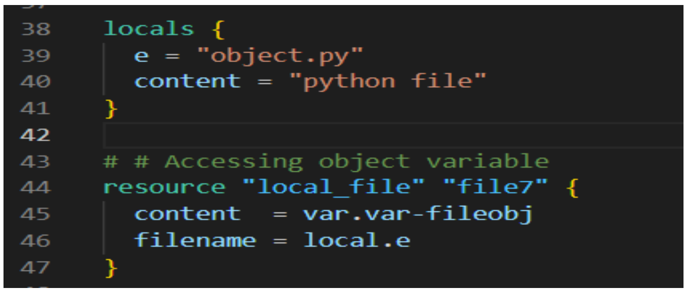
5. Object variable

Creating an object variable



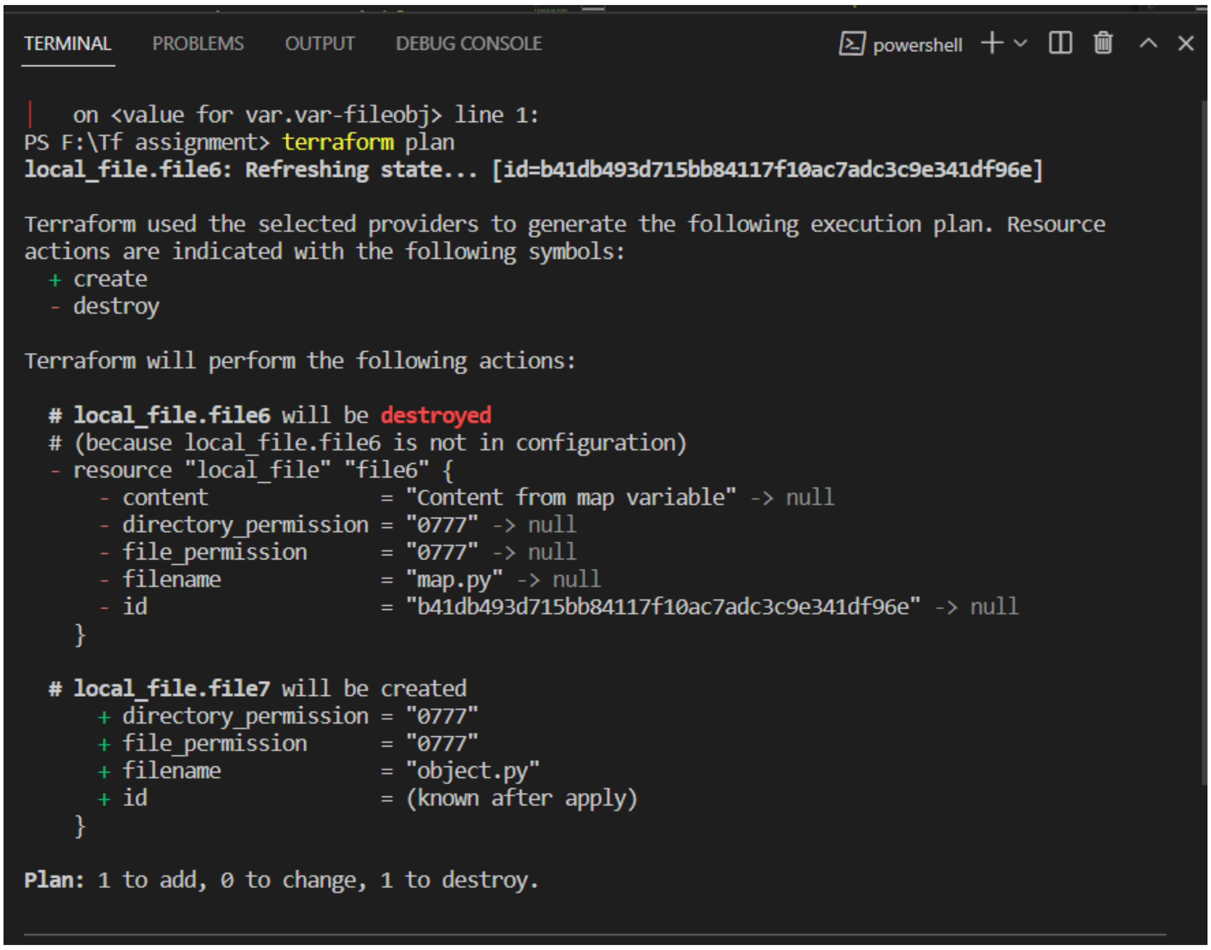
```
# # object in variable
variable "var-fileobj" {
  type = object({
    filename = string
    age = number
  })
  description = "Enter data"
  default = {
    age = 24
    filename = "value"
  }
}
```

Accessing the object variable



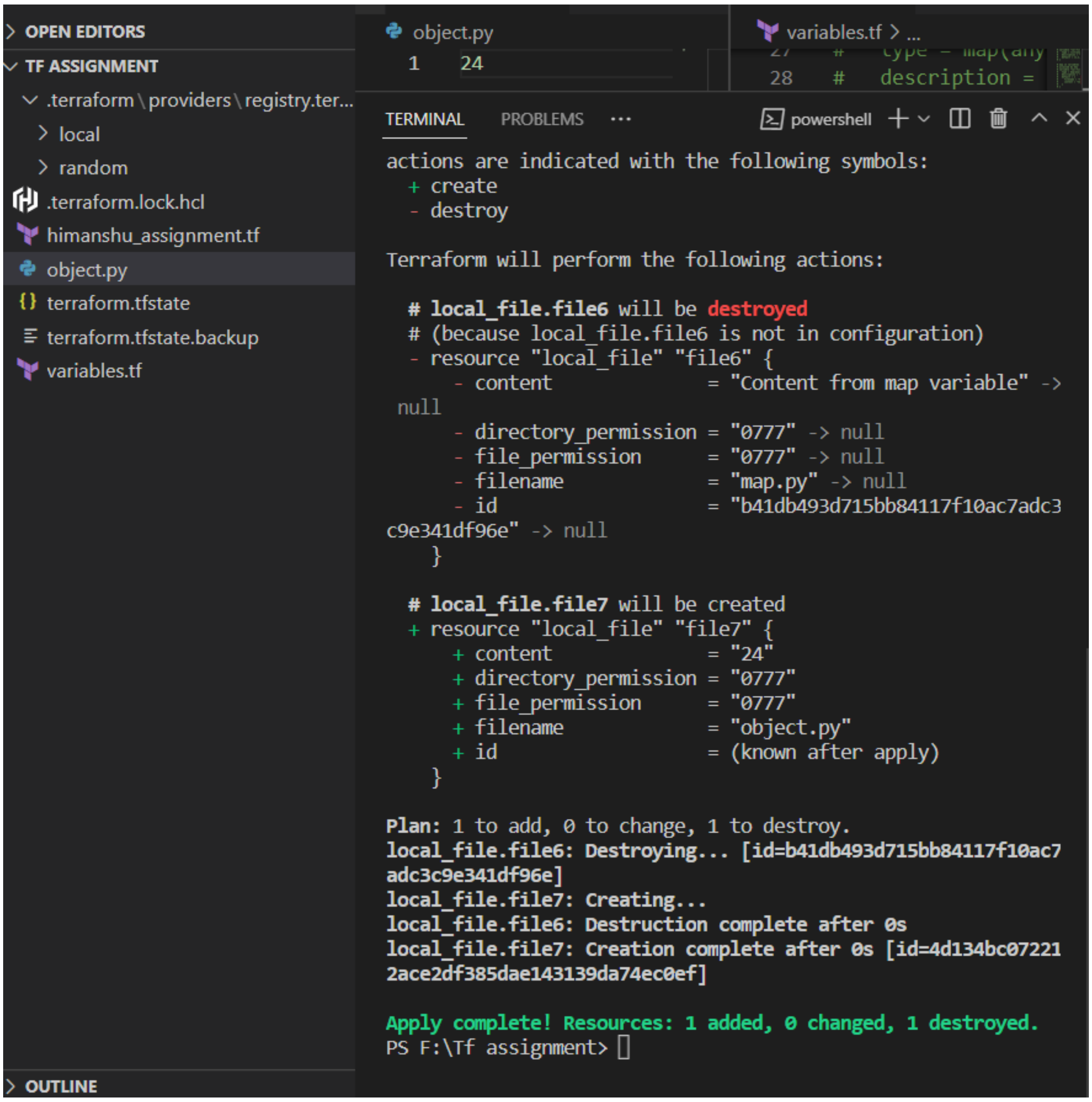
```
locals {
  e = "object.py"
  content = "python file"
}

# # Accessing object variable
resource "local_file" "file7" {
  content  = var.var-fileobj
  filename = local.e
}
```

After that executing terraform plan will show:



```
│   on <value for var.var-fileobj> line 1:
PS F:\Tf assignment> terraform plan
local_file.file6: Refreshing state... [id=b41db493d715bb84117f10ac7adc3c9e341df96e]

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

Terraform will perform the following actions:

  # local_file.file6 will be destroyed
  # (because local_file.file6 is not in configuration)
  - resource "local_file" "file6" {
      - content              = "Content from map variable" -> null
      - directory_permission = "0777" -> null
      - file_permission      = "0777" -> null
      - filename             = "map.py" -> null
      - id                   = "b41db493d715bb84117f10ac7adc3c9e341df96e" -> null
    }

  # local_file.file7 will be created
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "object.py"
      + id                   = (known after apply)
    }

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

Then terraform apply –auto-approve:



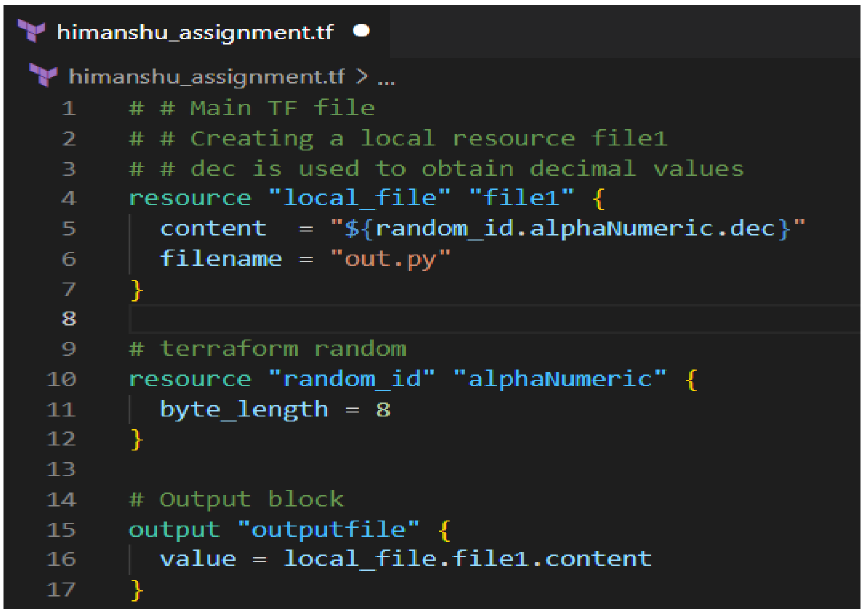
```
1    24
```

```
28   #   description =
```

```
actions are indicated with the following symbols:
  + create
  - destroy

Terraform will perform the following actions:

  # local_file.file6 will be destroyed
  # (because local_file.file6 is not in configuration)
  - resource "local_file" "file6" {
      - content              = "Content from map variable" ->
 null
      - directory_permission = "0777" -> null
      - file_permission      = "0777" -> null
      - filename             = "map.py" -> null
      - id                   = "b41db493d715bb84117f10ac7adc3
c9e341df96e" -> null
    }

  # local_file.file7 will be created
  + resource "local_file" "file7" {
      + content              = "24"
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "object.py"
      + id                   = (known after apply)
    }

Plan: 1 to add, 0 to change, 1 to destroy.
local_file.file6: Destroying... [id=b41db493d715bb84117f10ac7
adc3c9e341df96e]
local_file.file7: Creating...
local_file.file6: Destruction complete after 0s
local_file.file7: Creation complete after 0s [id=4d134bc07221
2ace2df385dae143139da74ec0ef]

Apply complete! Resources: 1 added, 0 changed, 1 destroyed.
PS F:\Tf assignment> 
```

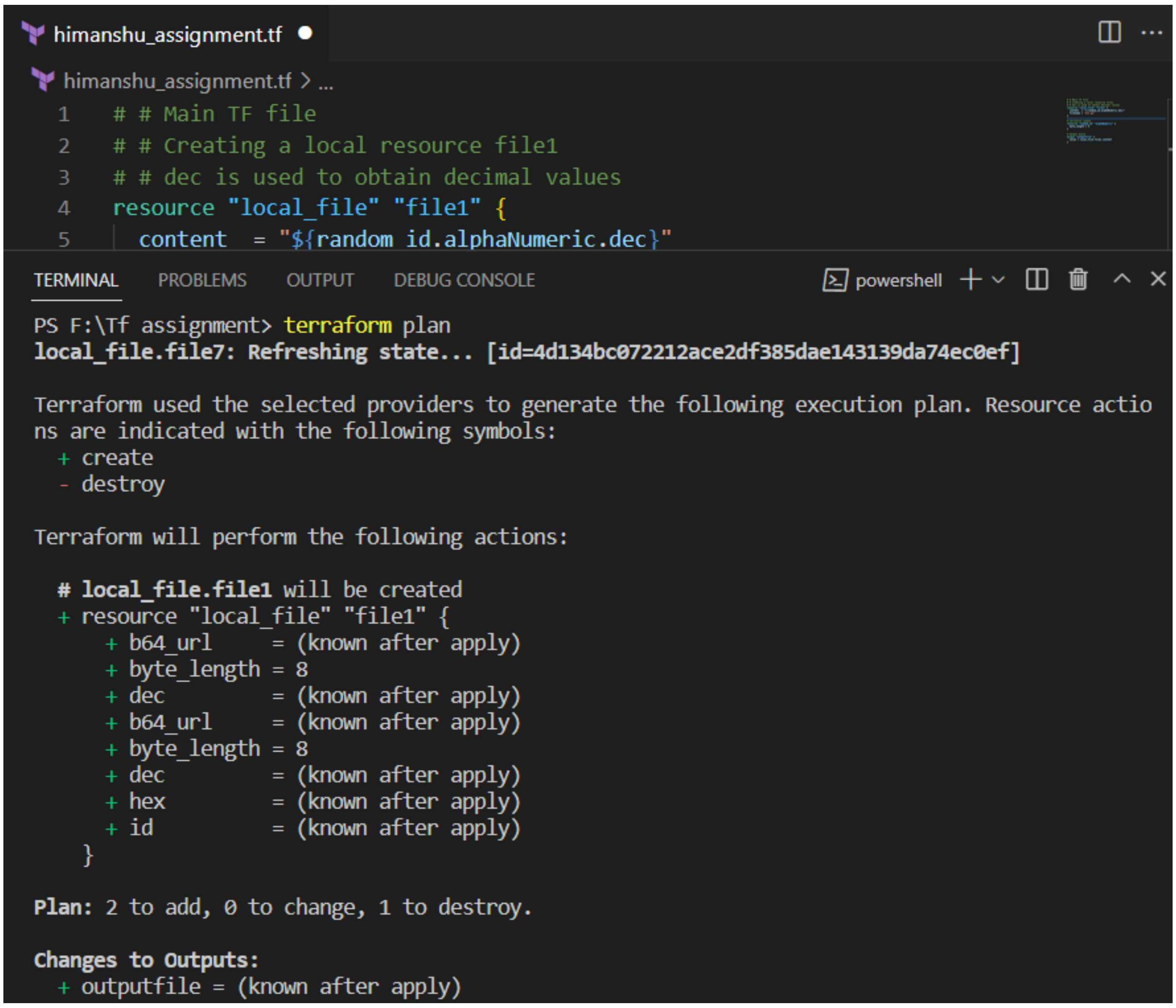
Output block

Initializing a output block



```
# # Main TF file
# # Creating a local resource file1
# # dec is used to obtain decimal values
resource "local_file" "file1" {
  content  = "${random_id.alphaNumeric.dec}"
  filename = "out.py"
}

# terraform random
resource "random_id" "alphaNumeric" {
  byte_length = 8
}

# Output block
output "outputfile" {
  value = local_file.file1.content
}
```

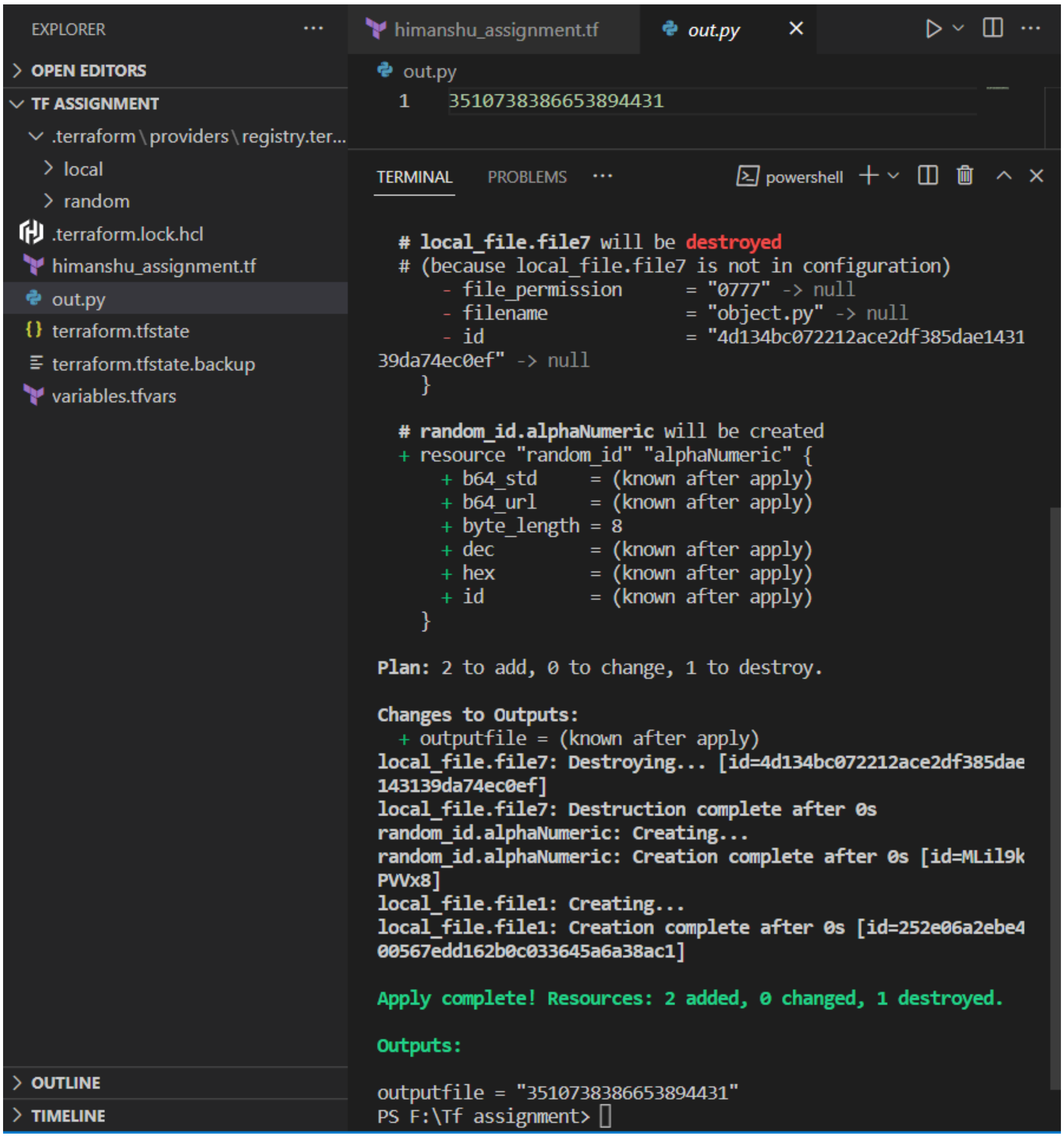
Terraform plan



```
# # Main TF file
# # Creating a local resource file1
# # dec is used to obtain decimal values
resource "local_file" "file1" {
  content  = "${random_id.alphaNumeric.dec}"
```

```
PS F:\Tf assignment> terraform plan
local_file.file7: Refreshing state... [id=4d134bc072212ace2df385dae143139da74ec0ef]

Terraform used the selected providers to generate the following execution plan. Resource actio
ns are indicated with the following symbols:
  + create
  - destroy

Terraform will perform the following actions:

  # local_file.file1 will be created
  + resource "local_file" "file1" {
      + b64_url     = (known after apply)
      + byte_length = 8
      + dec         = (known after apply)
      + b64_url     = (known after apply)
      + byte_length = 8
      + dec         = (known after apply)
      + hex         = (known after apply)
      + id          = (known after apply)
    }

Plan: 2 to add, 0 to change, 1 to destroy.

Changes to Outputs:
  + outputfile = (known after apply)
```

Terraform apply –auto-approve



```
35107383866538944431
```

```
  # local_file.file7 will be destroyed
  # (because local_file.file7 is not in configuration)
      - file_permission      = "0777" -> null
      - filename             = "object.py" -> null
      - id                   = "4d134bc072212ace2df385dae1431
39da74ec0ef" -> null
    }

  # random_id.alphaNumeric will be created
  + resource "random_id" "alphaNumeric" {
      + b64_std     = (known after apply)
      + b64_url     = (known after apply)
      + byte_length = 8
      + dec         = (known after apply)
      + hex         = (known after apply)
      + id          = (known after apply)
    }

Plan: 2 to add, 0 to change, 1 to destroy.

Changes to Outputs:
  + outputfile = (known after apply)
local_file.file7: Destroying... [id=4d134bc072212ace2df385dae
143139da74ec0ef]
local_file.file7: Destruction complete after 0s
random_id.alphaNumeric: Creating...
random_id.alphaNumeric: Creation complete after 0s [id=MLil9k
PVVx8]
local_file.file1: Creating...
local_file.file1: Creation complete after 0s [id=252e06a2ebe4
00567edd162b0c033645a6a38ac1]

Apply complete! Resources: 2 added, 0 changed, 1 destroyed.

Outputs:

outputfile = "35107383866538944431"
PS F:\Tf assignment>
```