TASKS ON TERRAFORM

- ❖ **NAME: S.MUZZAMMIL HUSSAIN**
- ❖ **DATE: 21/07/2025**
- ❖ **BATCH: 11**
- ❖ **NO.OF TASKS: 1**

---

1. Create local resources (multiple resources) each resource value should be injected from different ways of precedence?

To create multiple local resources in Terraform, each using **a different input variable injection method**, illustrating Terraform's **variable precedence system**.

Terraform Variable Injection Modes Demonstrated

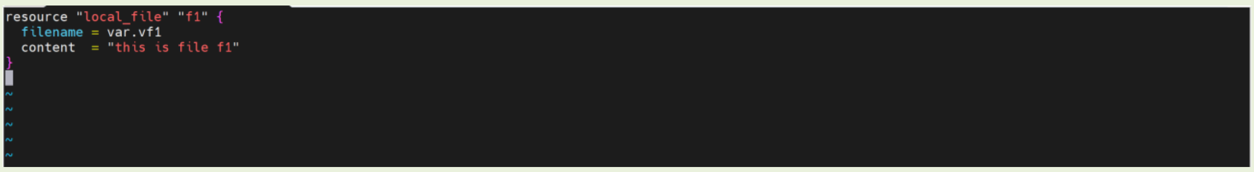
The document walks through **six variable assignment methods**, consistent with Terraform's official precedence hierarchy. Each method is demonstrated using a local_file resource, where the filename field is dynamically set via a variable.

1. Interactive Input (CLI Prompt)

If a variable is defined **without a default** and isn't set via CLI, env, or .tfvars file, Terraform will prompt the user at the time of apply.

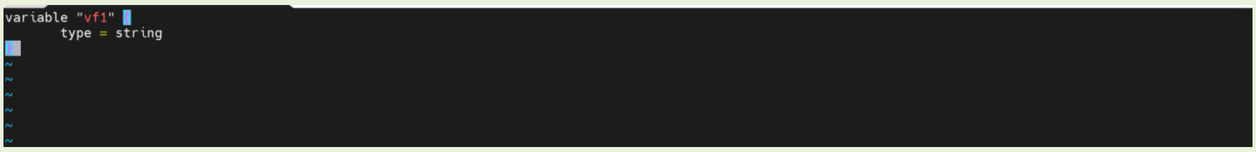
Create a file named file.tf using **vi file.tf** and enter the following and save using **esc:wq**

```
resource "local_file" "f1" {
  filename = var.vf1
  content  = "this is file f1"
}
```



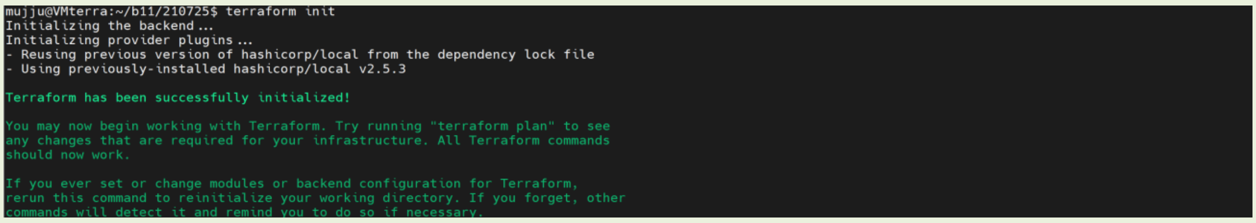
```
resource "local_file" "f1" {
  filename = var.vf1
  content  = "this is file f1"
}
```

Add variable using **vi variables.tf** as follows



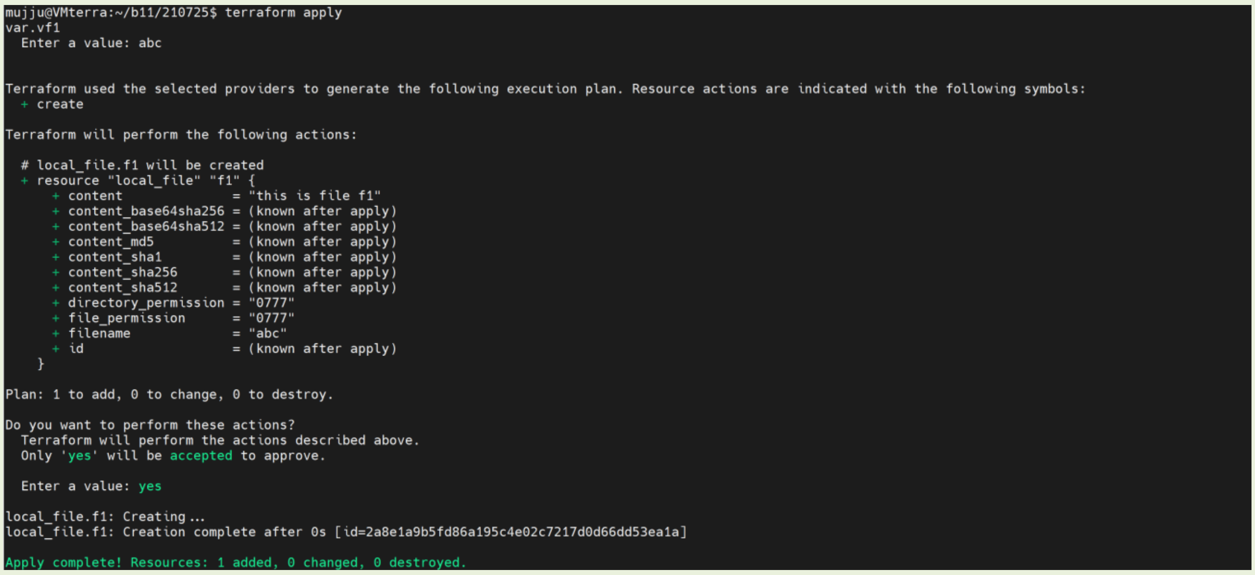
```
variable "vf1" {
        type = string
}
```

Run the command **terraform init**



```
mujju@VMterra:~/b11/210725$ terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/local from the dependency lock file
- Using previously-installed hashicorp/local v2.5.3

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.
```

Run the command **terraform apply**



```
mujju@VMterra:~/b11/210725$ terraform apply
var.vf1
  Enter a value: abc


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.f1 will be created
  + resource "local_file" "f1" {
      + content              = "this is file f1"
      + 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             = "abc"
      + 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.f1: Creating...
local_file.f1: Creation complete after 0s [id=2a8e1a9b5fd86a195c4e02c7217d0d66dd53ea1a]

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

Prompts: var.vf1
Enter a value: abc

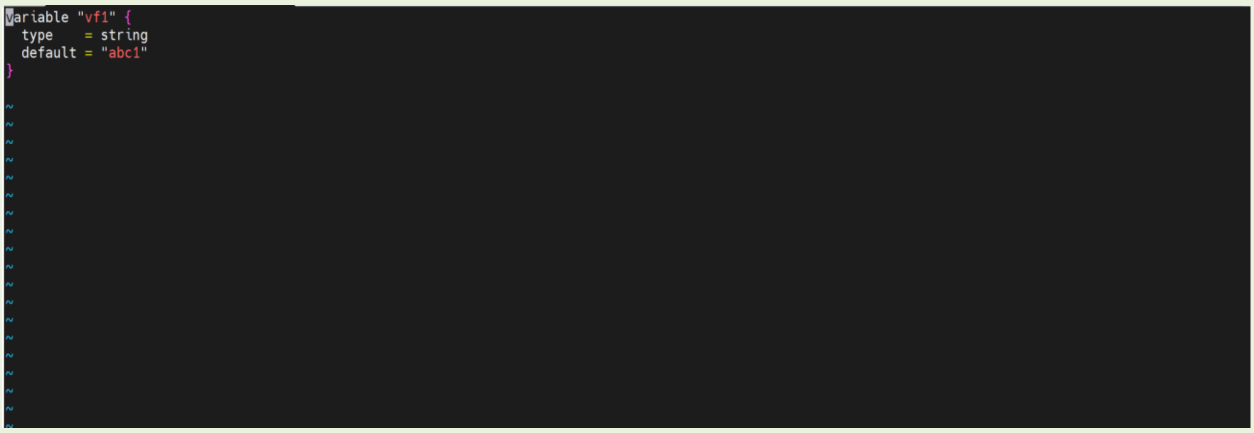
Terraform replaces the resource with filename = **"abc"**

2. Default Value in Variable Block

Description:
Setting a default value means Terraform uses it unless overridden.

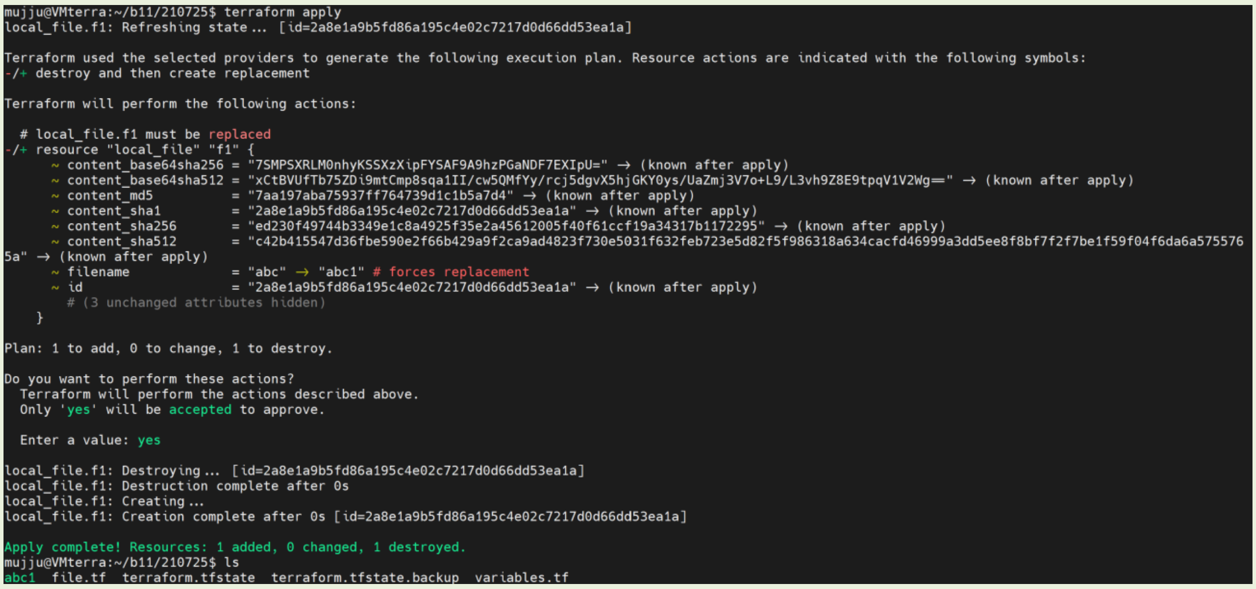
In **variables.tf** file enter the following

```
variable "vf1" {
  type    = string
  default = "abc1"
}
```



```
variable "vf1" {
  type    = string
  default = "abc1"
}
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

Run the command **terraform apply**



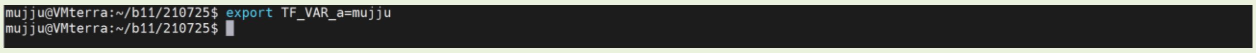
```
mujju@VMterra:~/b11/210725$ terraform apply
local_file.f1: Refreshing state... [id=2a8e1a9b5fd86a195c4e02c7217d0d66dd53ea1a]

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.f1 must be replaced
-/+ resource "local_file" "f1" {
      ~ content_base64sha256 = "7SMPSXRLM0nhyKSSXzXipFYSAF9A9hzPGaNDF7EXIpU=" → (known after apply)
      ~ content_base64sha512 = "xCtBVUfTb75ZDi9mtCmp8sqa1II/cw5QMfYy/rcj5dgvX5hjGKY0ys/UaZmj3V7o+L9/L3vh9Z8E9tpqV1V2Wg==" → (known after apply)
      ~ content_md5          = "7aa197aba75937ff764739d1c1b5a7d4" → (known after apply)
      ~ content_sha1         = "2a8e1a9b5fd86a195c4e02c7217d0d66dd53ea1a" → (known after apply)
      ~ content_sha256       = "ed230f49744b3349e1c8a4925f35e2a45612005f40f61ccf19a34317b1172295" → (known after apply)
      ~ content_sha512       = "c42b415547d36fbe590e2f66b429a9f2ca9ad4823f730e5031f632feb723e5d82f5f986318a634cacfd46999a3dd5ee8f8bf7f2f7be1f59f04f6da6a575576
5a" → (known after apply)
      ~ filename             = "abc" → "abc1" # forces replacement
      ~ id                   = "2a8e1a9b5fd86a195c4e02c7217d0d66dd53ea1a" → (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.f1: Destroying... [id=2a8e1a9b5fd86a195c4e02c7217d0d66dd53ea1a]
local_file.f1: Destruction complete after 0s
local_file.f1: Creating...
local_file.f1: Creation complete after 0s [id=2a8e1a9b5fd86a195c4e02c7217d0d66dd53ea1a]

Apply complete! Resources: 1 added, 0 changed, 1 destroyed.
mujju@VMterra:~/b11/210725$ ls
abc1  file.tf  terraform.tfstate  terraform.tfstate.backup  variables.tf
```

No user input required; the value **abc1** is used in the resource definition.

3. Environment Variable (TF_VAR_name)

Terraform automatically uses environment variables prefixed with **TF_VAR_.**

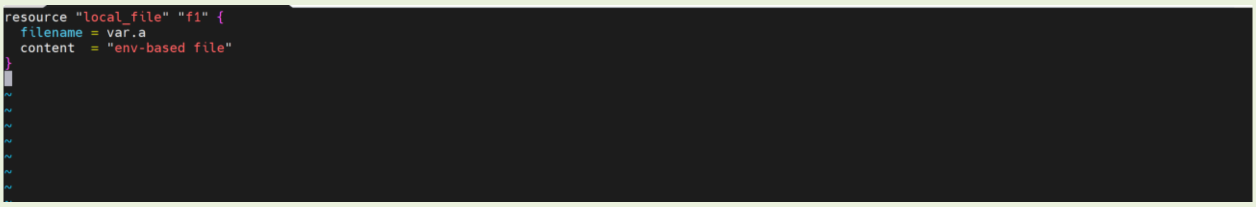
export TF_VAR_a=mujju



```
mujju@VMterra:~/b11/210725$ export TF_VAR_a=mujju
mujju@VMterra:~/b11/210725$ 
```

In **file.tf** file add the contents as follows

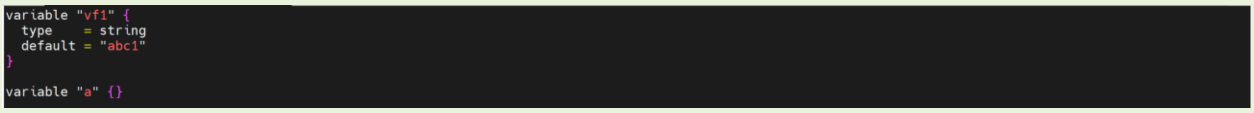
```
resource "local_file" "f1" {
  filename = var.a
  content  = "env-based file"
}
```



```
resource "local_file" "f1" {
  filename = var.a
  content  = "env-based file"
}

~
~
~
~
~
~
~
```

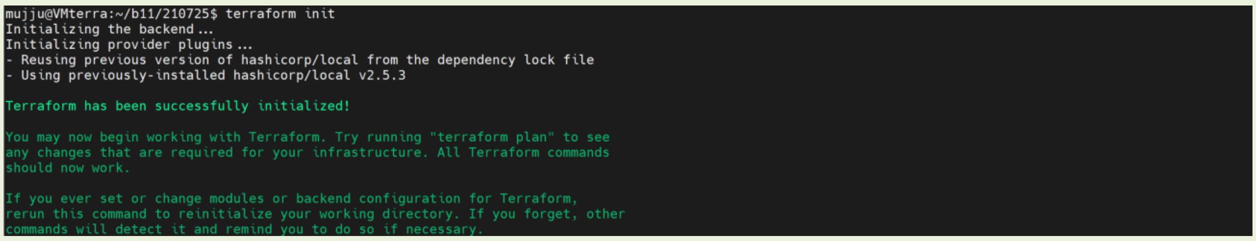
In **variables.tf** file add the contents as follows

```
variable "vf1" {
  type    = string
  default = "abc1"
}
variable "a" {}
```



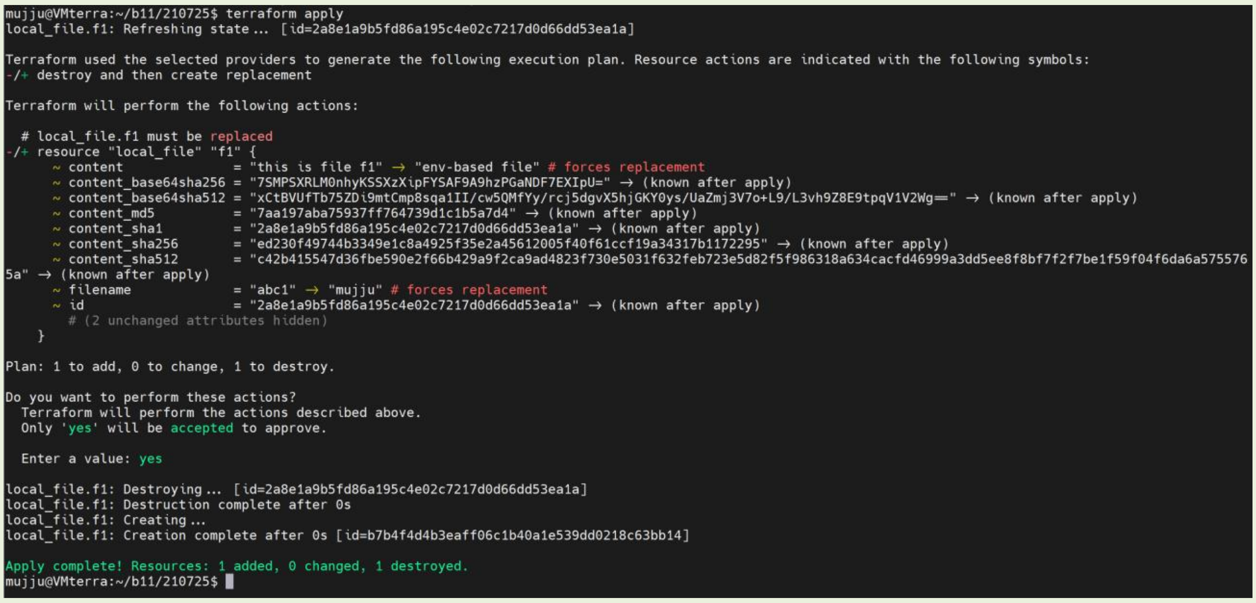
```
variable "vf1" {
  type    = string
  default = "abc1"
}

variable "a" {}
```

Run the command **terraform init**



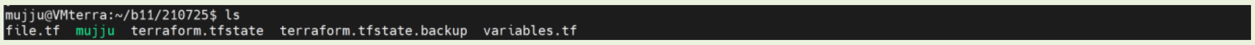
```
mujju@VMterra:~/b11/210725$ terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/local from the dependency lock file
- Using previously-installed hashicorp/local v2.5.3

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.
```

Run the command **terraform apply**



```
mujju@VMterra:~/b11/210725$ terraform apply
local_file.f1: Refreshing state... [id=2a8e1a9b5fd86a195c4e02c7217d0d66dd53ea1a]

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.f1 must be replaced
-/+ resource "local_file" "f1" {
      ~ content              = "this is file f1" → "env-based file" # forces replacement
      ~ content_base64sha256 = "7SMPSXRLM0nhyKSSXzXipFYSAF9A9hzPGaNDF7EXIpU=" → (known after apply)
      ~ content_base64sha512 = "xCtBVUfTb75ZDi9mtCmp8sqa1II/cw5QMfYy/rcj5dgvX5hjGKY0ys/UaZmj3V7o+L9/L3vh9Z8E9tpqV1V2Wg==" → (known after apply)
      ~ content_md5          = "7aa197aba75937ff764739d1c1b5a7d4" → (known after apply)
      ~ content_sha1         = "2a8e1a9b5fd86a195c4e02c7217d0d66dd53ea1a" → (known after apply)
      ~ content_sha256       = "ed230f49744b3349e1c8a4925f35e2a45612005f40f61ccf19a34317b1172295" → (known after apply)
      ~ content_sha512       = "c42b415547d36fbe590e2f66b429a9f2ca9ad4823f730e5031f632feb723e5d82f5f986318a634cacfd46999a3dd5ee8f8bf7f2f7be1f59f04f6da6a575576
5a" → (known after apply)
      ~ filename             = "abc1" → "mujju" # forces replacement
      ~ id                   = "2a8e1a9b5fd86a195c4e02c7217d0d66dd53ea1a" → (known after apply)
        # (2 unchanged attributes hidden)
    }

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

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

  Enter a value: yes

local_file.f1: Destroying... [id=2a8e1a9b5fd86a195c4e02c7217d0d66dd53ea1a]
local_file.f1: Destruction complete after 0s
local_file.f1: Creating...
local_file.f1: Creation complete after 0s [id=b7b4f4d4b3eaff06c1b40a1e539dd0218c63bb14]

Apply complete! Resources: 1 added, 0 changed, 1 destroyed.
mujju@VMterra:~/b11/210725$
```

Validates that the filename is overridden by the shell environment value **mujju**.



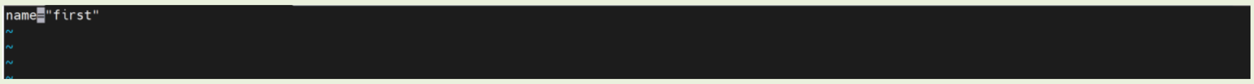
```
mujju@VMterra:~/b11/210725$ ls
file.tf  mujju  terraform.tfstate  terraform.tfstate.backup  variables.tf
```

4. terraform.tfvars File

A file named **terraform.tfvars** is created and contents are added as follows

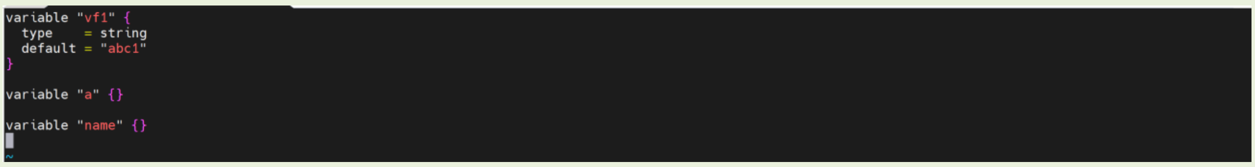
Contents (terraform.tfvars):

name = "first"



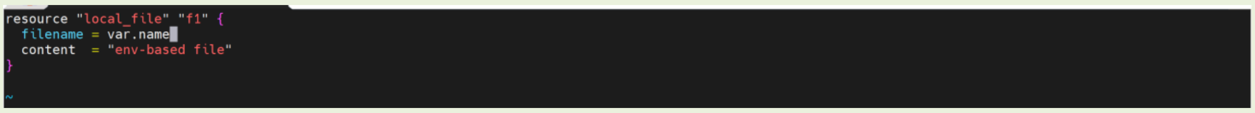
```
name="first"
~
~
~
~
```

In **variables.tf** file add the contents as follows



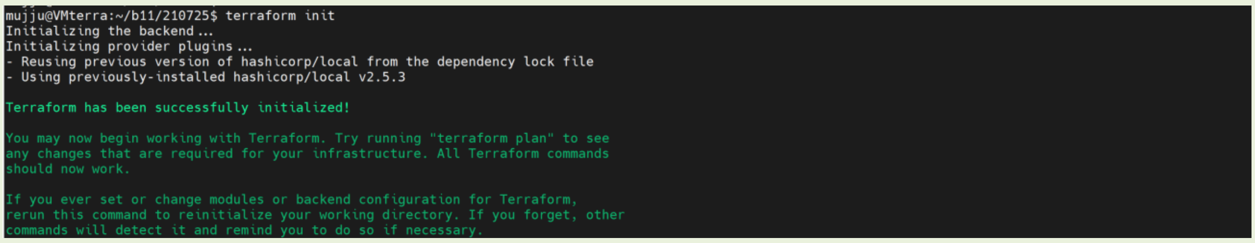
```
variable "vf1" {
  type    = string
  default = "abc1"
}

variable "a" {}

variable "name" {}

~
```

In **file.tf** add the contents as follows



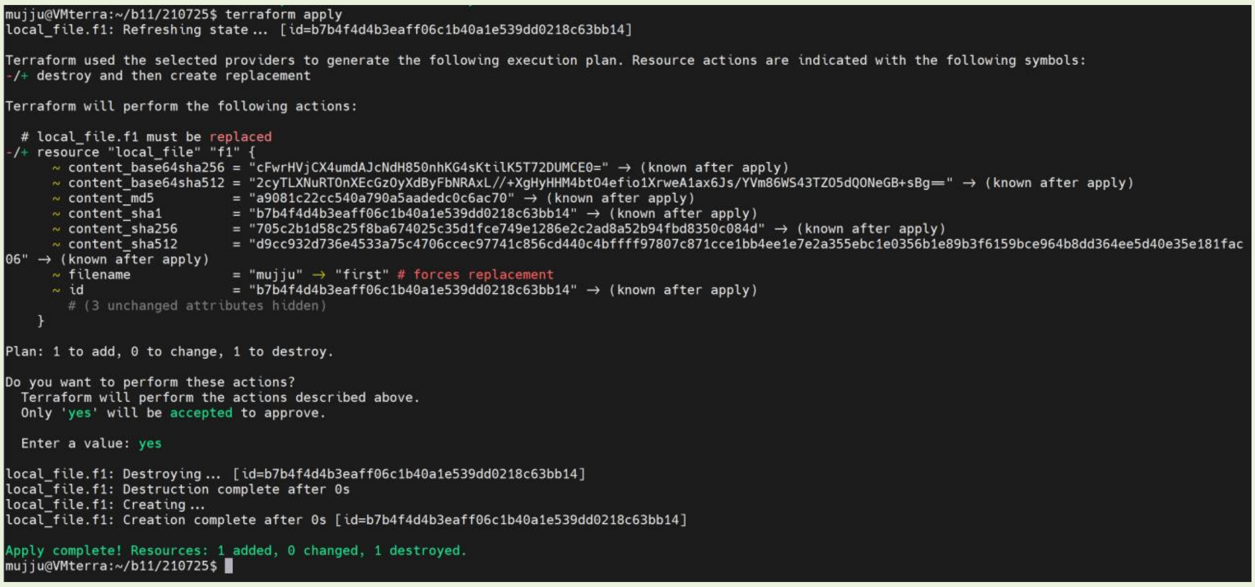
```
resource "local_file" "f1" {
  filename = var.name
  content  = "env-based file"
}
~
```

Run the command **terraform init**



```
mujju@VMterra:~/b11/210725$ terraform init
Initializing the backend ...
Initializing provider plugins ...
- Reusing previous version of hashicorp/local from the dependency lock file
- Using previously-installed hashicorp/local v2.5.3

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.
```

Run the command **terraform init**



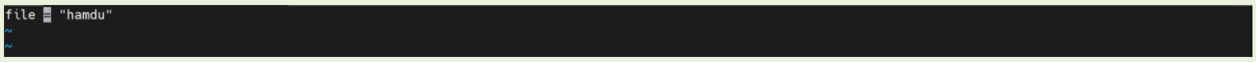
```
mujju@VMterra:~/b11/210725$ terraform apply
local_file.f1: Refreshing state ... [id=b7b4f4d4b3eaff06c1b40a1e539dd0218c63bb14]

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.f1 must be replaced
-/+ resource "local_file" "f1" {
      ~ content_base64sha256 = "cFwrHVjCX4umdAJcNdH850nhKG4sKtilK5T72DUMCE0=" → (known after apply)
      ~ content_base64sha512 = "2cyTLXNuRTOnXEcGzOyXdByFbNRAxL//+XgHyHHM4btO4efio1XrweA1ax6Js/YVm86WS43TZO5dQONeGB+sBg==" → (known after apply)
      ~ content_md5          = "a9081c22cc540a790a5aadedc0c6ac70" → (known after apply)
      ~ content_sha1         = "b7b4f4d4b3eaff06c1b40a1e539dd0218c63bb14" → (known after apply)
      ~ content_sha256       = "705c2b1d58c25f8ba674025c35d1fce749e1286e2c2ad8a52b94fbd8350c084d" → (known after apply)
      ~ content_sha512       = "d9cc932d736e4533a75c4706ccec97741c856cd440c4bffff97807c871cce1bb4ee1e7e2a355ebc1e0356b1e89b3f6159bce964b8dd364ee5d40e35e181fac
06" → (known after apply)
      ~ filename             = "mujju" → "first" # forces replacement
      ~ id                   = "b7b4f4d4b3eaff06c1b40a1e539dd0218c63bb14" → (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.f1: Destroying ... [id=b7b4f4d4b3eaff06c1b40a1e539dd0218c63bb14]
local_file.f1: Destruction complete after 0s
local_file.f1: Creating ...
local_file.f1: Creation complete after 0s [id=b7b4f4d4b3eaff06c1b40a1e539dd0218c63bb14]

Apply complete! Resources: 1 added, 0 changed, 1 destroyed.
mujju@VMterra:~/b11/210725$ 
```

Usage: Applies **var.name = "first"** automatically.

5 .auto.tfvars File

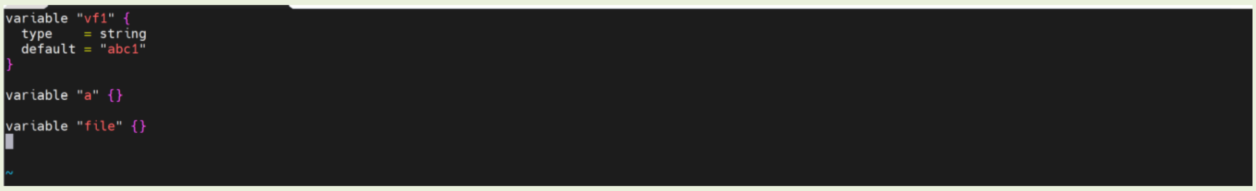
Create a file **mujju.auto.tfvars** using **vi mujju.auto.tfvars** and add the content as follows

file=hamdu



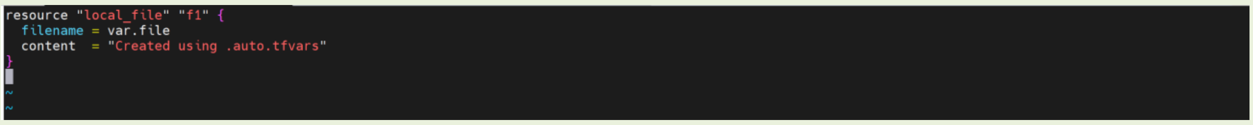
```
file = "hamdu"
~
~
```

In **variables.tf** file add the contents as follows



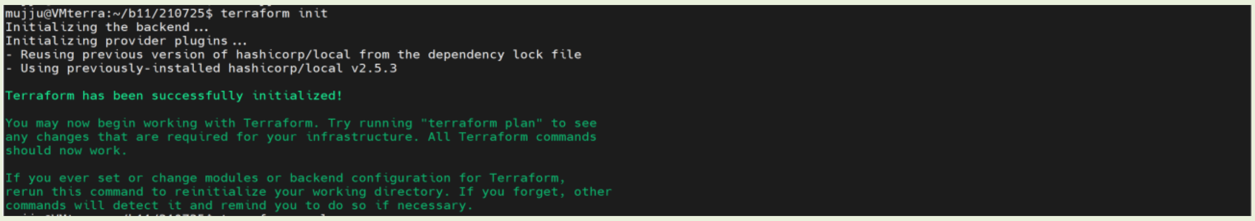
```
variable "vf1" {
  type    = string
  default = "abc1"
}

variable "a" {}

variable "file" {}

~
```

In **file.tf** file add the contents as follows



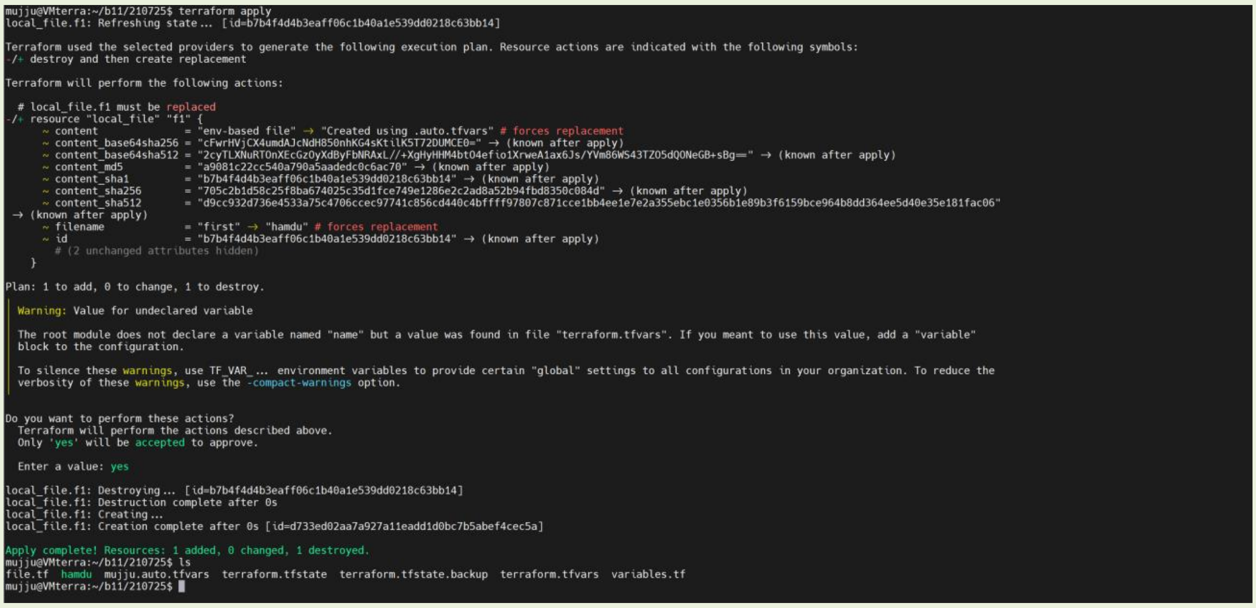
```
resource "local_file" "f1" {
  filename = var.file
  content  = "Created using .auto.tfvars"
}

~
~
```

Run the command **terraform init**



```
mujju@VMterra:~/b11/210725$ terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/local from the dependency lock file
- Using previously-installed hashicorp/local v2.5.3

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.
```

Run the command **terraform init**



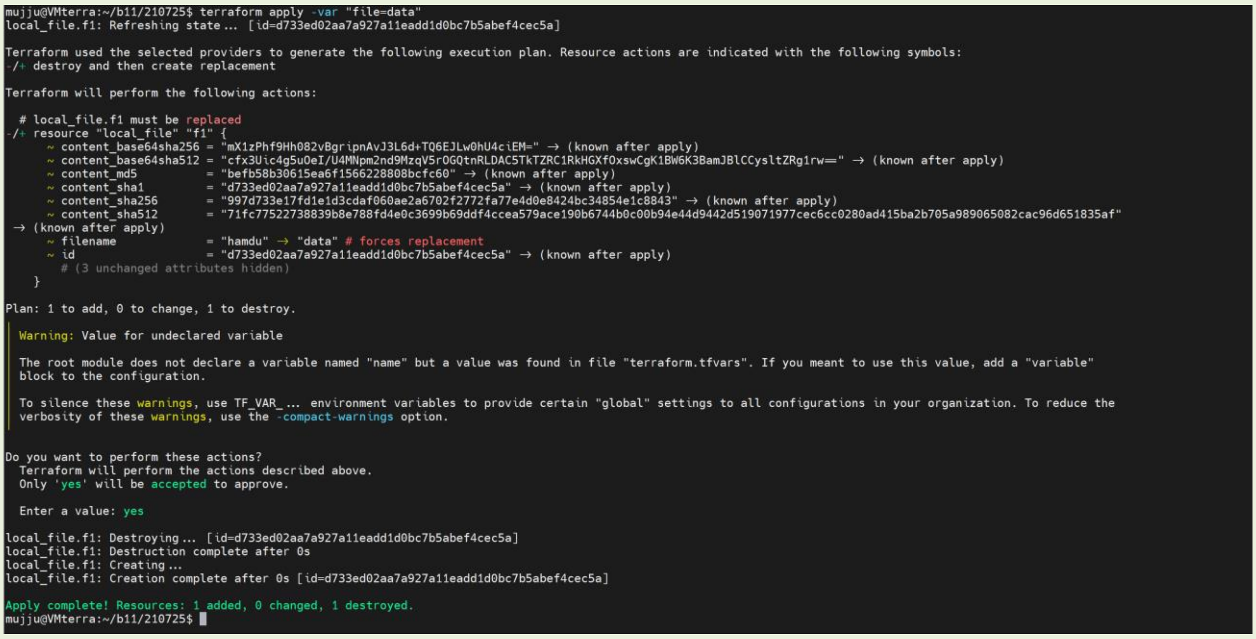
```
mujju@VMterra:~/b11/210725$ terraform apply
local_file.f1: Refreshing state... [id=b7b4f4d4b3eaff06c1b40a1e539dd0218c63bb14]

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.f1 must be replaced
-/+ resource "local_file" "f1" {
      ~ content              = "env-based file" → "Created using .auto.tfvars" # forces replacement
      ~ content_base64sha256 = "cFwrHVjCX4umdAJcNdH850nhKG4sKtilK5T72DUMCE0=" → (known after apply)
      ~ content_base64sha512 = "2cyTLXNuRTOnXEcGzOyXdByFbNRAxL//+XgHyHHM4btO4efio1XrweA1ax6Js/YVm86WS43TZO5dQONeGB+sBg==" → (known after apply)
      ~ content_md5          = "a9081c22cc540a790a5aadedc0c6ac70" → (known after apply)
      ~ content_sha1         = "b7b4f4d4b3eaff06c1b40a1e539dd0218c63bb14" → (known after apply)
      ~ content_sha256       = "705c2b1d58c25f8ba674025c35d1fce749e1286e2c2ad8a52b94fbd8350c084d" → (known after apply)
      ~ content_sha512       = "d9cc932d736e4533a75c4706ccec97741c856cd440c4bffff97807c871cce1bb4ee1e7e2a355ebc1e0356b1e89b3f6159bce964b8dd364ee5d40e35e181fac06"
 → (known after apply)
      ~ filename             = "first" → "hamdu" # forces replacement
      ~ id                   = "b7b4f4d4b3eaff06c1b40a1e539dd0218c63bb14" → (known after apply)
        # (2 unchanged attributes hidden)
    }

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

  Warning: Value for undeclared variable

  The root module does not declare a variable named "name" but a value was found in file "terraform.tfvars". If you meant to use this value, add a "variable"
  block to the configuration.

  To silence these warnings, use TF_VAR_... environment variables to provide certain "global" settings to all configurations in your organization. To reduce the
  verbosity of these warnings, use the -compact-warnings option.


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.f1: Destroying... [id=b7b4f4d4b3eaff06c1b40a1e539dd0218c63bb14]
local_file.f1: Destruction complete after 0s
local_file.f1: Creating...
local_file.f1: Creation complete after 0s [id=d733ed02aa7a927a11eadd1d0bc7b5abef4cec5a]

Apply complete! Resources: 1 added, 0 changed, 1 destroyed.
mujju@VMterra:~/b11/210725$ ls
file.tf  hamdu  mujju.auto.tfvars  terraform.tfstate  terraform.tfstate.backup  terraform.tfvars  variables.tf
mujju@VMterra:~/b11/210725$
```

Changes filename = **"first"** → **"hamdu"** showing **auto.tfvars** value gets precedence over earlier input.

6. Command-Line -var Input

Run the Command:

terraform apply -var "file=data"



```
mujju@VMterra:~/b11/210725$ terraform apply -var "file=data"
local_file.f1: Refreshing state... [id=d733ed02aa7a927a11eadd1d0bc7b5abef4cec5a]

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.f1 must be replaced
-/+ resource "local_file" "f1" {
      ~ content_base64sha256 = "mX1zPhf9Hh082vBgripnAvJ3L6d+TQ6EJLw0hU4ciEM=" → (known after apply)
      ~ content_base64sha512 = "cfx3Uic4g5u0eI/U4MNpm2nd9MzqV5rOGQtnRLDAC5TkTZRC1RkHGXfOxswCgK1BW6K3BamJBlCCysltZRg1rw==" → (known after apply)
      ~ content_md5          = "befb58b30615ea6f1566228808bcfc60" → (known after apply)
      ~ content_sha1         = "d733ed02aa7a927a11eadd1d0bc7b5abef4cec5a" → (known after apply)
      ~ content_sha256       = "997d733e17fd1e1d3cdaf060ae2a6702f2772fa77e4d0e8424bc34854e1c8843" → (known after apply)
      ~ content_sha512       = "71fc77522738839b8e788fd4e0c3699b69ddf4ccea579ace190b6744b0c00b94e44d9442d519071977cec6cc0280ad415ba2b705a989065082cac96d651835af"
 → (known after apply)
      ~ filename             = "hamdu" → "data" # forces replacement
      ~ id                   = "d733ed02aa7a927a11eadd1d0bc7b5abef4cec5a" → (known after apply)
        # (3 unchanged attributes hidden)
    }

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

  Warning: Value for undeclared variable

  The root module does not declare a variable named "name" but a value was found in file "terraform.tfvars". If you meant to use this value, add a "variable"
  block to the configuration.

  To silence these warnings, use TF_VAR_... environment variables to provide certain "global" settings to all configurations in your organization. To reduce the
  verbosity of these warnings, use the -compact-warnings option.


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.f1: Destroying... [id=d733ed02aa7a927a11eadd1d0bc7b5abef4cec5a]
local_file.f1: Destruction complete after 0s
local_file.f1: Creating...
local_file.f1: Creation complete after 0s [id=d733ed02aa7a927a11eadd1d0bc7b5abef4cec5a]

Apply complete! Resources: 1 added, 0 changed, 1 destroyed.
mujju@VMterra:~/b11/210725$
```

Effect Observed here is "**hamdu**" replaced by "**data**" showing **CLI** value takes the highest precedence.

Terraform Variable Precedence Summary (as demonstrated):

Precedence	Method	Example
1 (highest)	CLI -var	terraform apply -var="file=..."
2	.auto.tfvars file	filename.auto.tfvars
3	terraform.tfvars file	terraform.tfvars
4	Environment variables	export TF_VAR_varname=value
5	Default in variable block	default = "abc"
6 (lowest)	Interactive prompt	CLI prompt when no other input