**Name: Raksha**

**Date : 19/7/25**

**Terraform class 3 tasks- subsequent apply destroy**

**Batch11**

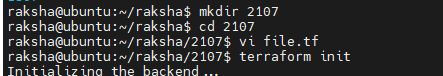
1.subsequent apply, destroy

=> .tf files = configuration files that describe what infrastructure you want.

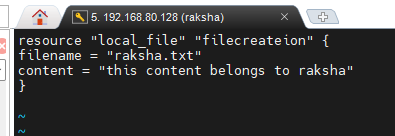
=> terraform.tfstate = a state file that stores what infrastructure currently exists, so Terraform knows what to do next.

=> Terraform works based on "desired state" (from code) vs "current state" (from the cloud).

Firstly created one directory and inside that created one configuration file



Tf. File-> content wrote inside .tf file



**terraform apply Flow**

apply

CLI -> Core : User runs `terraform apply`

Core <- Read config : Core reads .tf configuration files

Core -> Provider : Core calls provider to create the resource

Provider -> Resource : Provider uses plugin logic to create actual resource (e.g., create file)

Provider -> Core ->Sends back resource creation status and output

Core -> State File -> Updates terraform.tfstate to persist new resource state

Result:

->Resource created ->raksha.txt

->terraform.tfstate updated

**terraform destroy Flow**

**destroy**

CLI -> Core -> User runs `terraform destroy`

Core -> State File -> Reads current state to know existing resources

Core -> Provider -> Core instructs provider to destroy those resources

Provider -> Resource -> Deletes/destroys the actual resource (e.g., deletes file)

Provider -> Core -> Sends back destruction status

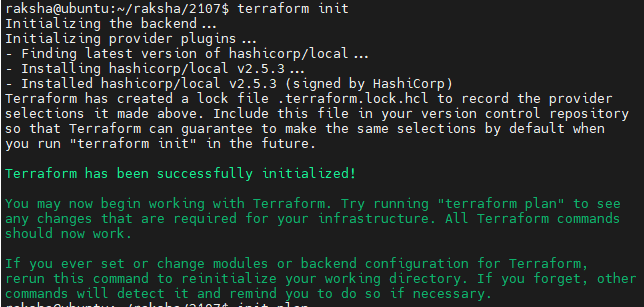
Core -> State File -> Removes resource from terraform.tfstate (updates it)

Result:

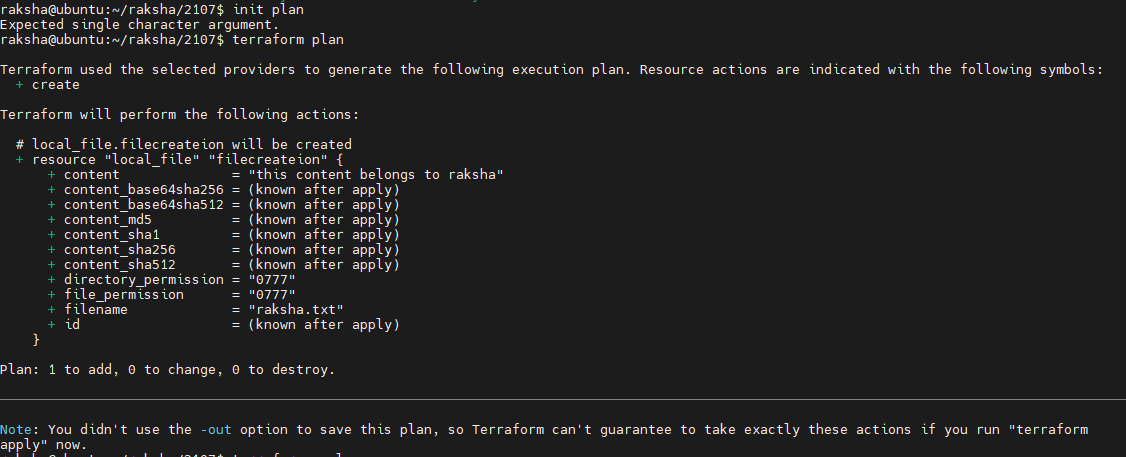
-> Resource deleted (e.g., raksha.txt)

-> terraform.tfstate updated (resource entry removed)

Terraform init



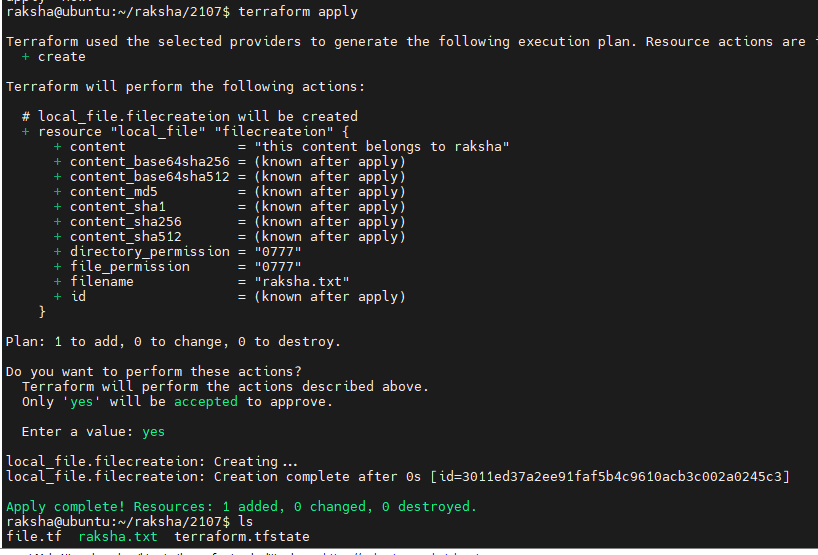
Terraform plan



Terraform apply

**Terraform Actions:**

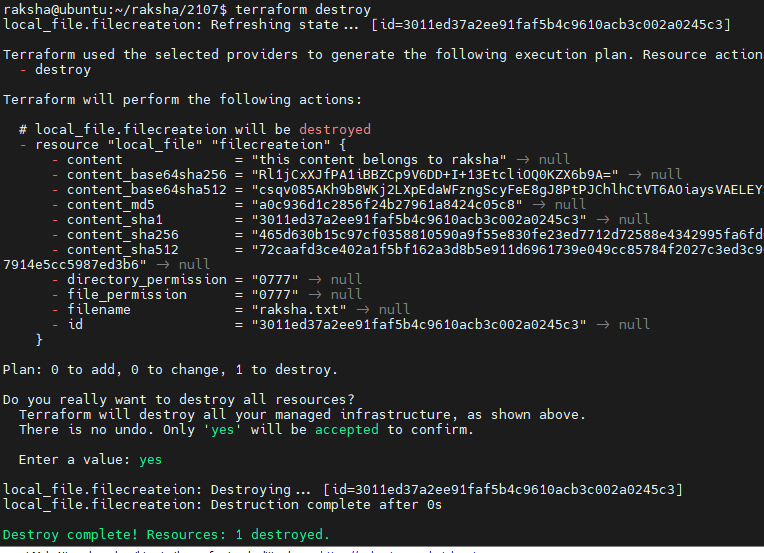
* Checks that raksha.txt doesn't exist.
* Creates the file raksha.txt with the content.
* Saves the file's metadata (like path, content hash) in terraform.tfstate.
* File raksha.txt is created in the current directory.
* terraform.tfstate contains the resource local\_file.filecreateion.



Terraform destroy

**Terraform Actions:**

* Looks at the state file and finds local\_file.filecreateion.
* Deletes raksha.txt from the filesystem.
* Removes the resource from the state file.
* raksha.txt is deleted from disk.
* terraform.tfstate is updated to remove the file resource.
* Your .tf file remains unchanged.



Terraform apply

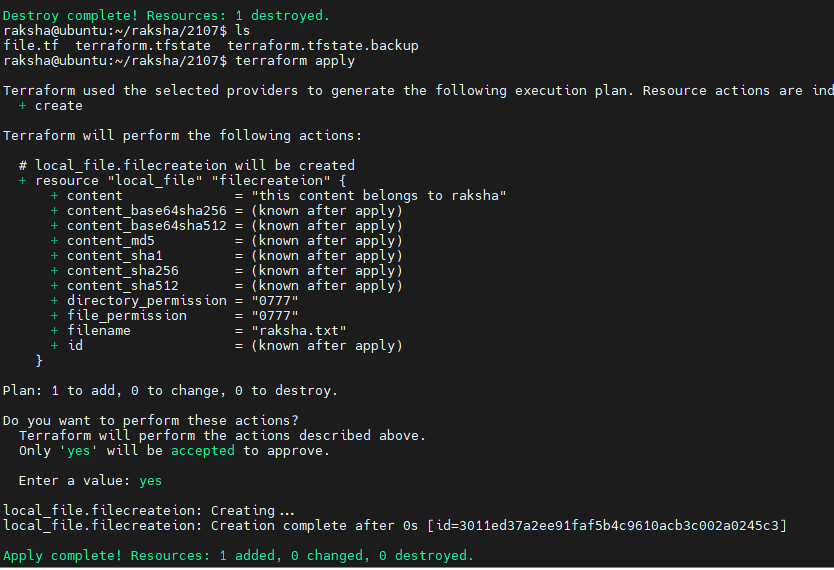
**Subsequent terraform apply**

“terraform apply”

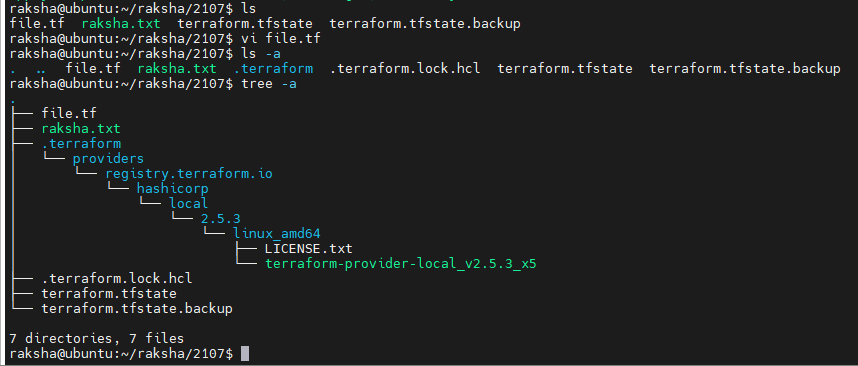
**Terraform Actions what it does**

* Reads main.tf and sees that raksha.txt should exist.
* Checks terraform.tfstate (now it is empty).
* Re-creates raksha.txt again with the same content.
* Updates the state file with the new file metadata.
* raksha.txt is re-created with original content.
* Infrastructure is back to original state.

This loop can be repeated any number of times



Ls-a and tree -a command applied to see detailed structure



.