**Name: Raksha**

**Date: 19/7/25**

**Terraform class 3-tasks- local, random ,null explore**

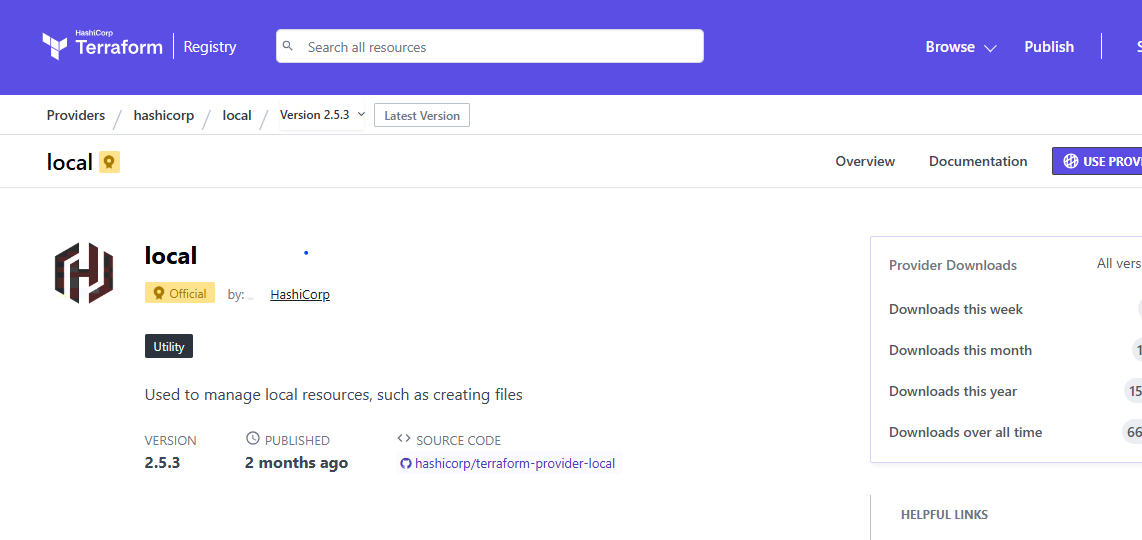
**Batch 11**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

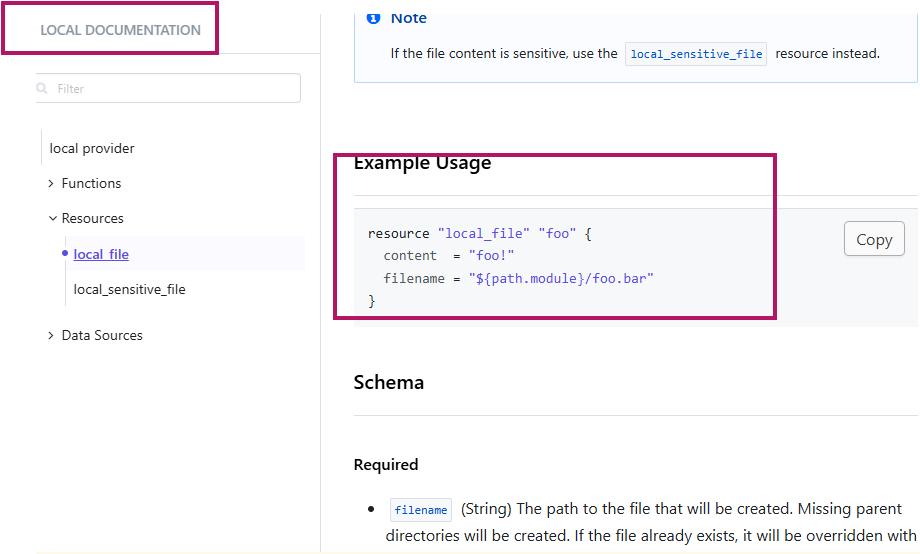
**2.explore these providers -> local, random, null**

**Goto registry.terraform.io**

**Local\_file -provider**



Search local-> go to documentation-> in that we can explore below page



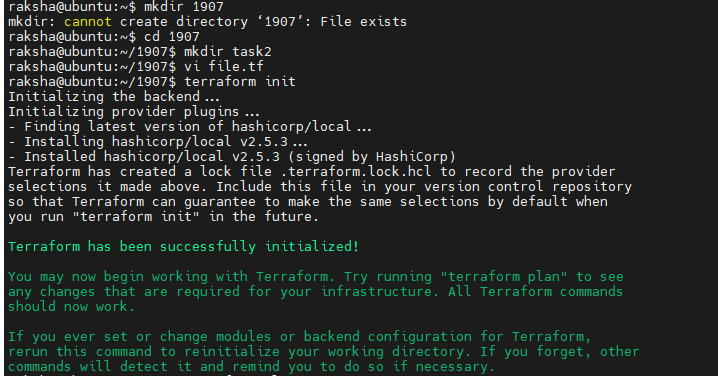
Local\_file->provider

resource "local\_file" "registry\_local" {

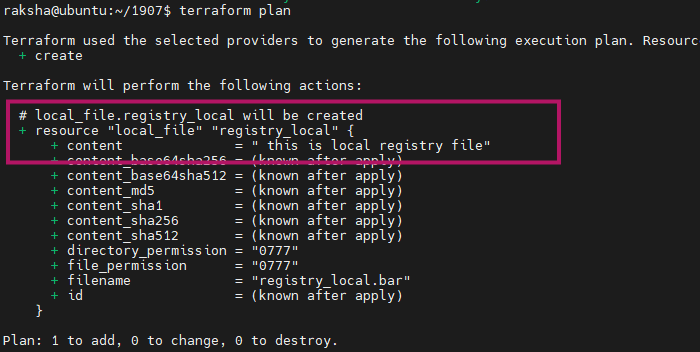
content = " this is local registry file"

filename = "registry\_local.bar"

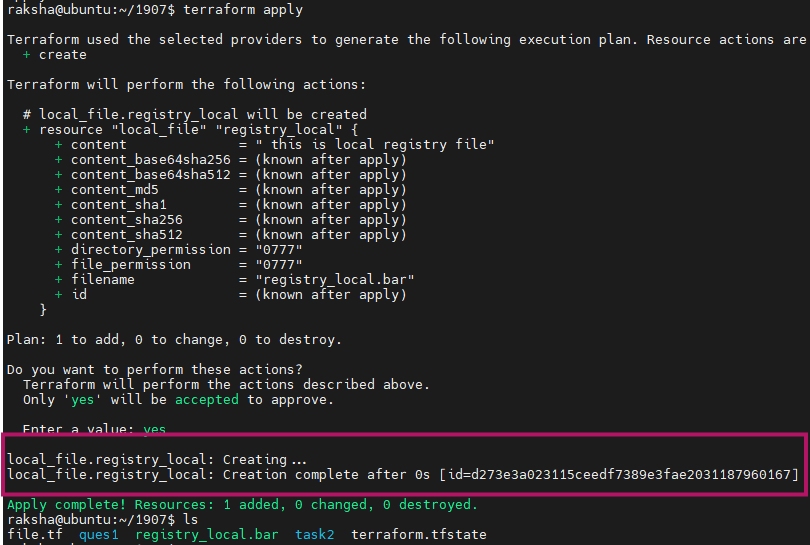
}



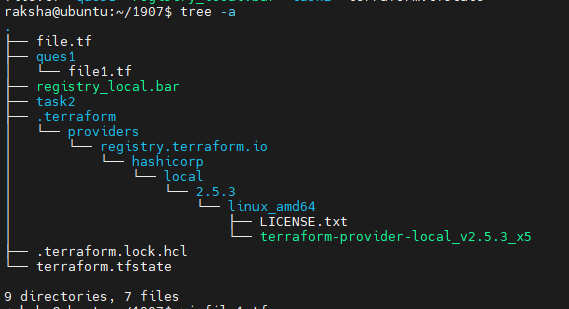
Terraform.plan



Terraform apply



Tree -a



If working with the local\_sensitive\_file resource -it's part of the **local provider**, similar to local\_file, but it treats the content as **sensitive**.

->The file content won’t be printed in the plan/apply output

->It's useful when handling secrets locally (eg :passwords, keys)

Local\_sensitive\_file-provider

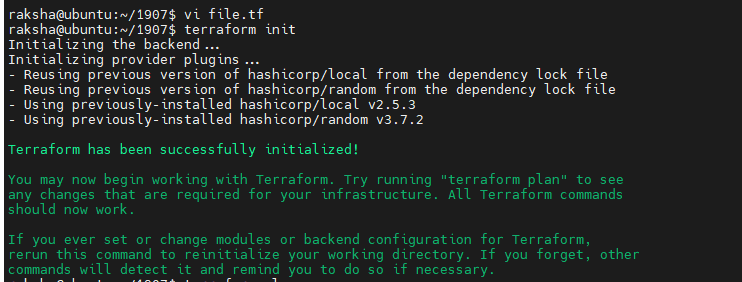
resource "local\_sensitive\_file" "raksha" {

content = "this is same as local file provider but contains store sensitive data"

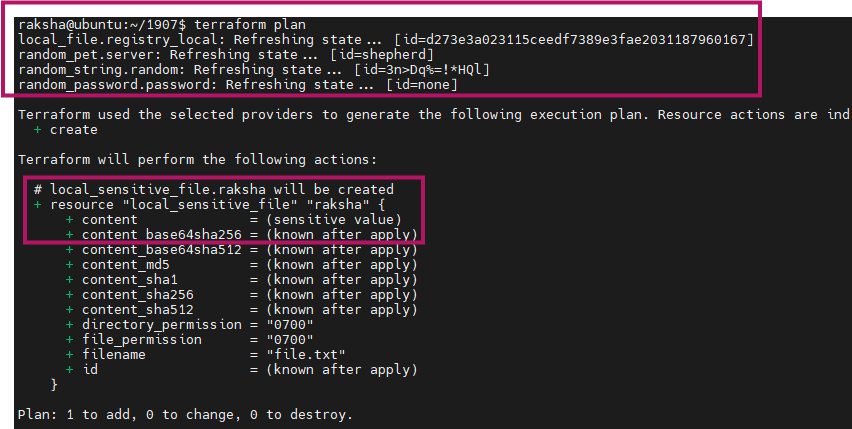
filename = "file.txt"

}

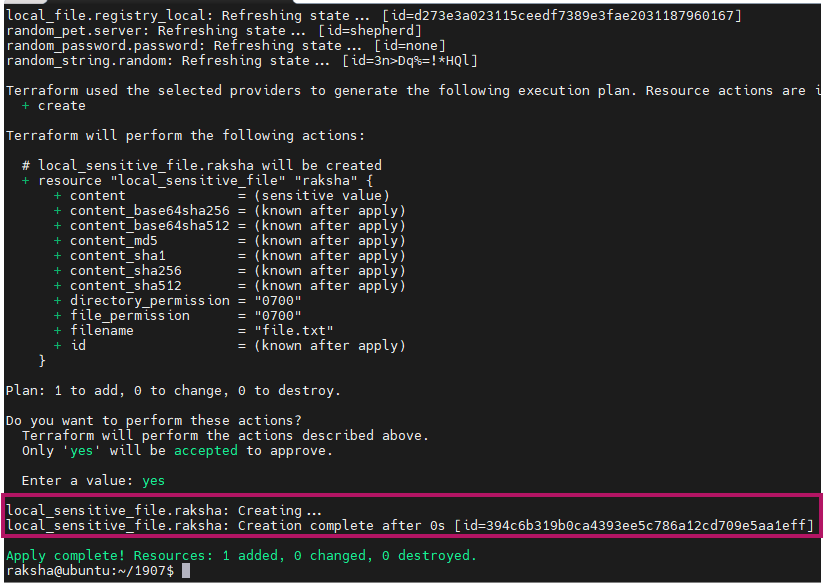
Terraform init stage



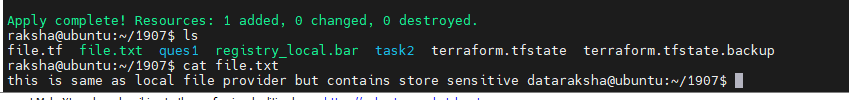
Terraform plan stage



Terraform apply



Cat file.txt



**Random string-provider**

Random string:

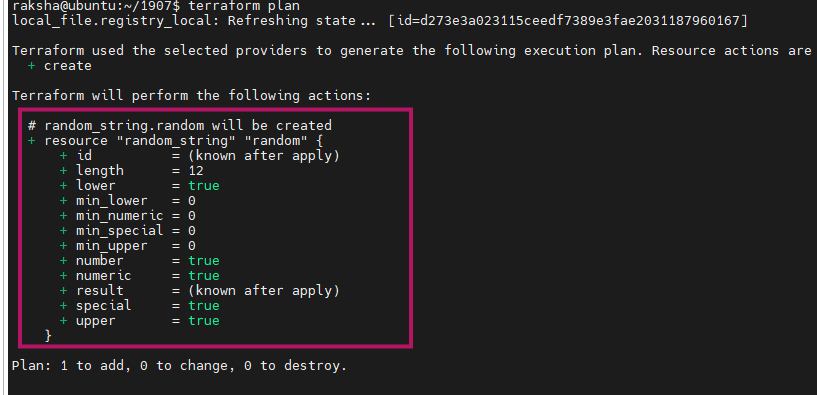
resource "random\_string" "random" {

length = 12

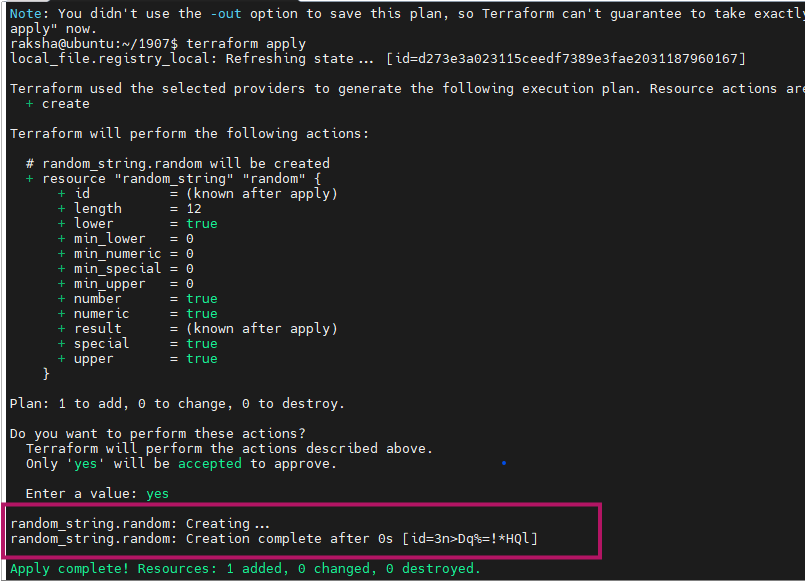
}

Terraform init -upgrade

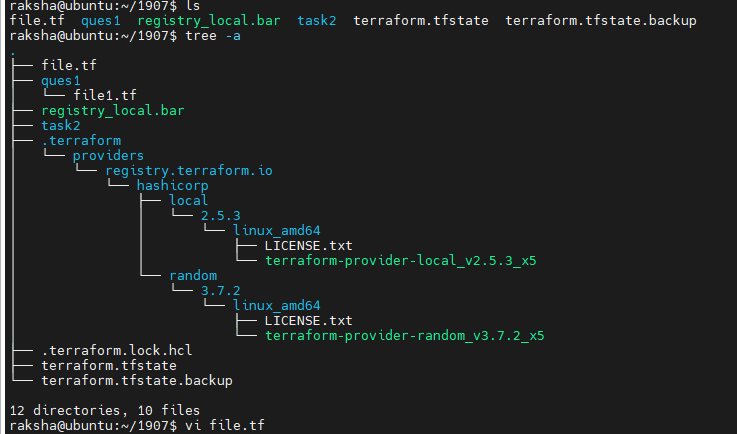
Terraform plan



Terraform apply



Tree-a



**Random-pet => providers**

variable "ami\_id" {

type = string

default = "pet name"

}

resource "random\_pet" "server" {

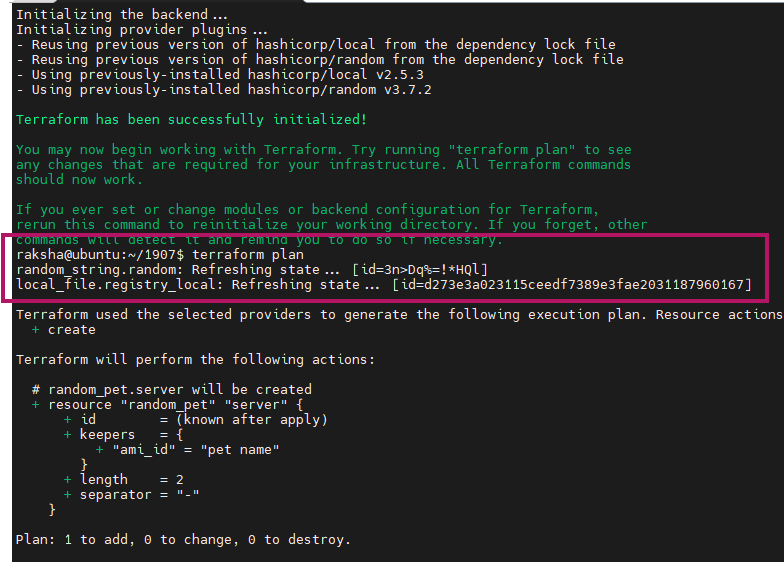
keepers = {

ami\_id = var.ami\_id

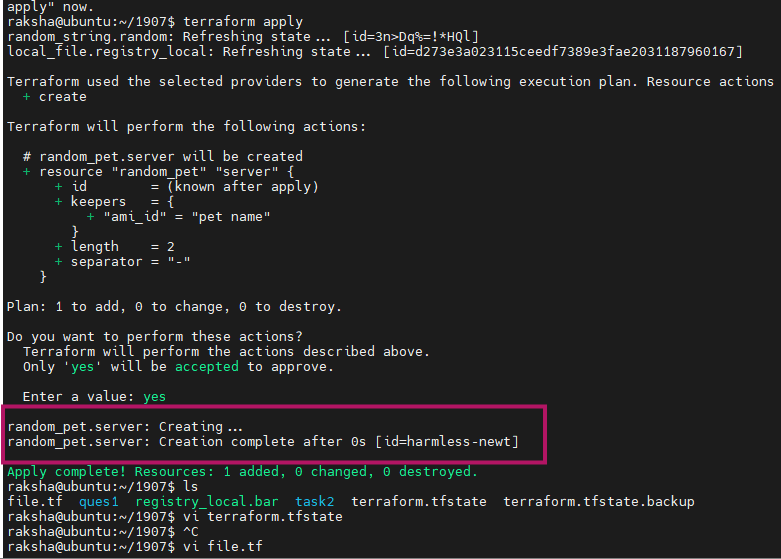
}

length = 2

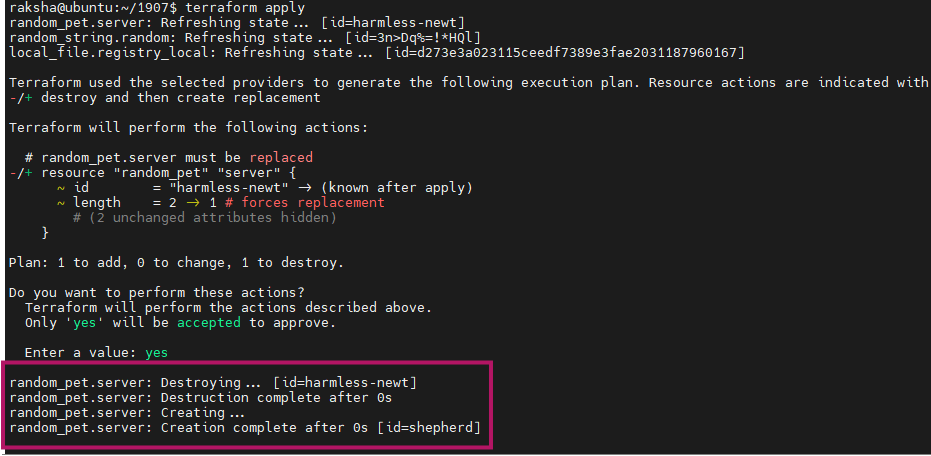
}



Terraform apply



Here changed a value means length to 1



Random password

resource "random\_password" "password" {

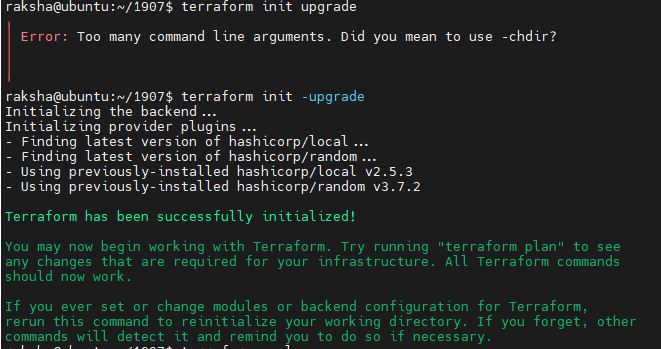
length = 16

special = true

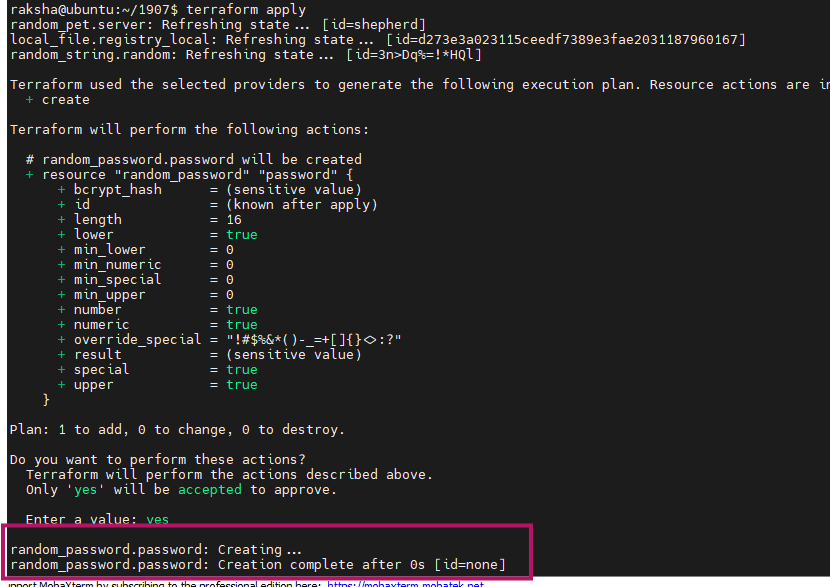
override\_special = "!#$%&\*()-\_=+[]{}<>:?"

}

Terraform init



Terraform apply



Null-resources-provider

resource "null\_resource" "create\_folder" {

provisioner "local-exec" {

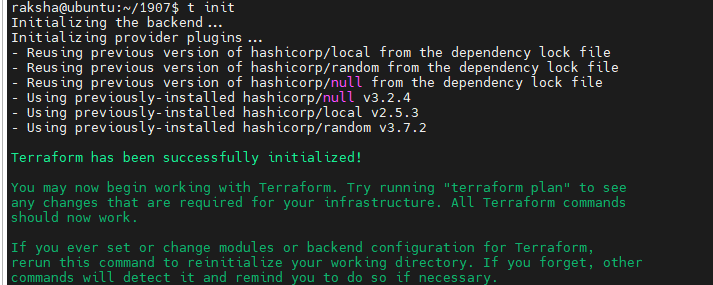
command = "mkdir -p ./resource2\_raksha"

}

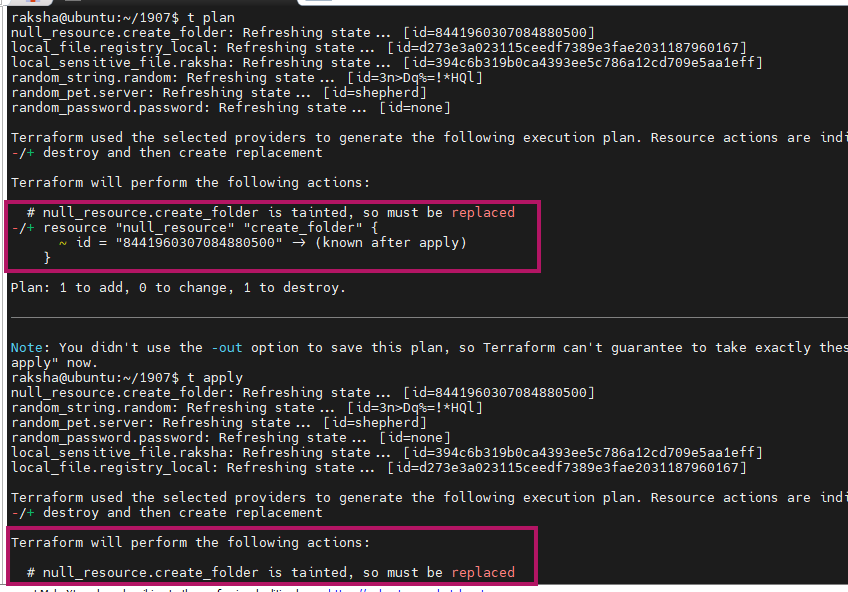
}

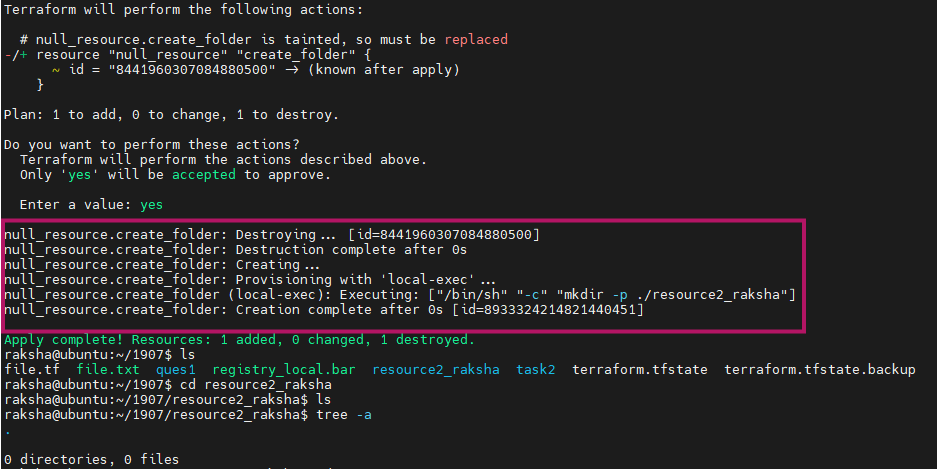
Alias t = ‘terraform’

t init



T plan and t apply





Tree -a

