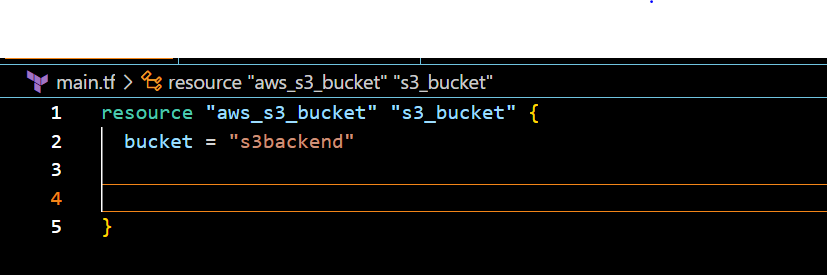
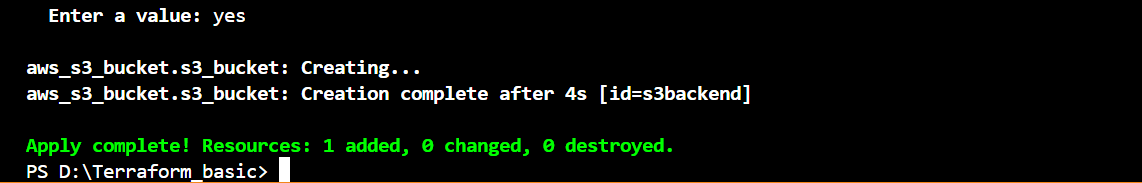
TERRAFORM [TASK-05]

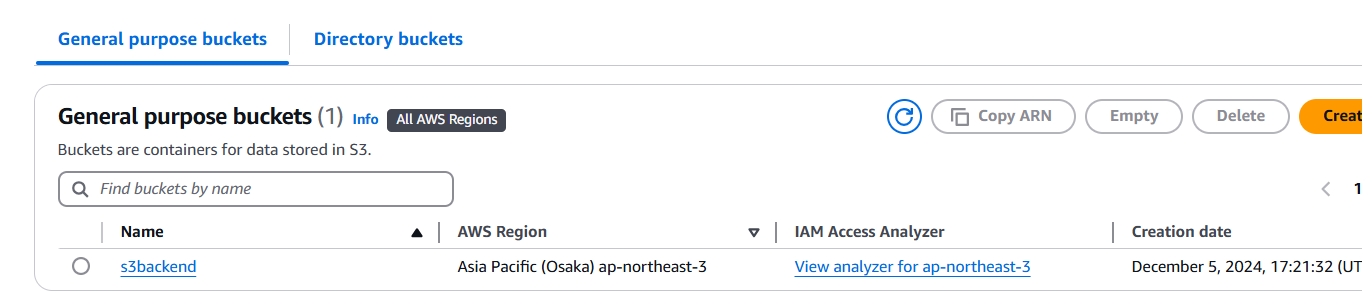
**1) Watch terraform-05 video.  
2) Execute the script shown in video.**

Creating the private s3 bucket

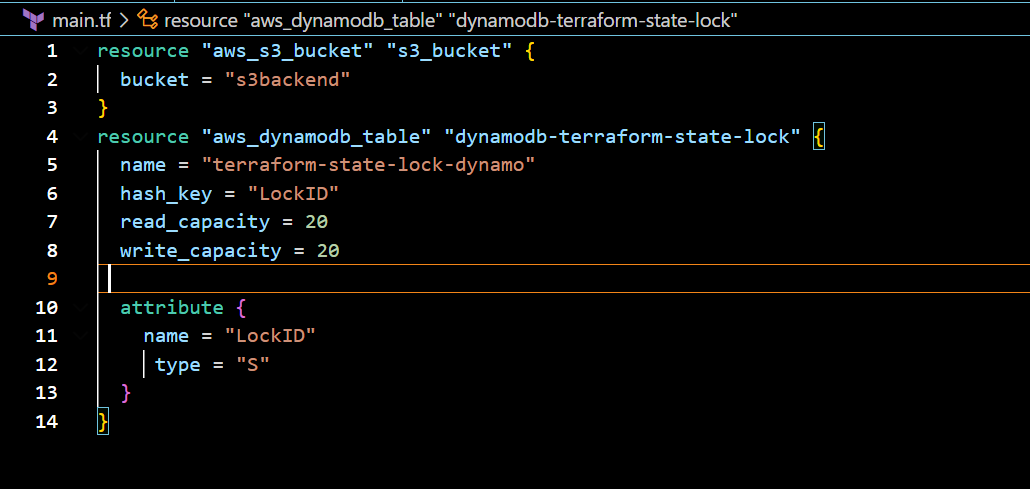


* terraform init
* terraform apply



We can check this 

Create dynamo db using terraform:



**TERRAFORM PROVISIONERS**

Terraform provisioners allow us to execute command, scripts on remote machines or

local place were terraform is installed.

Provisioners will be written inside the resource blocks.

We have two types of provisioners

**1) Remote provisioner**

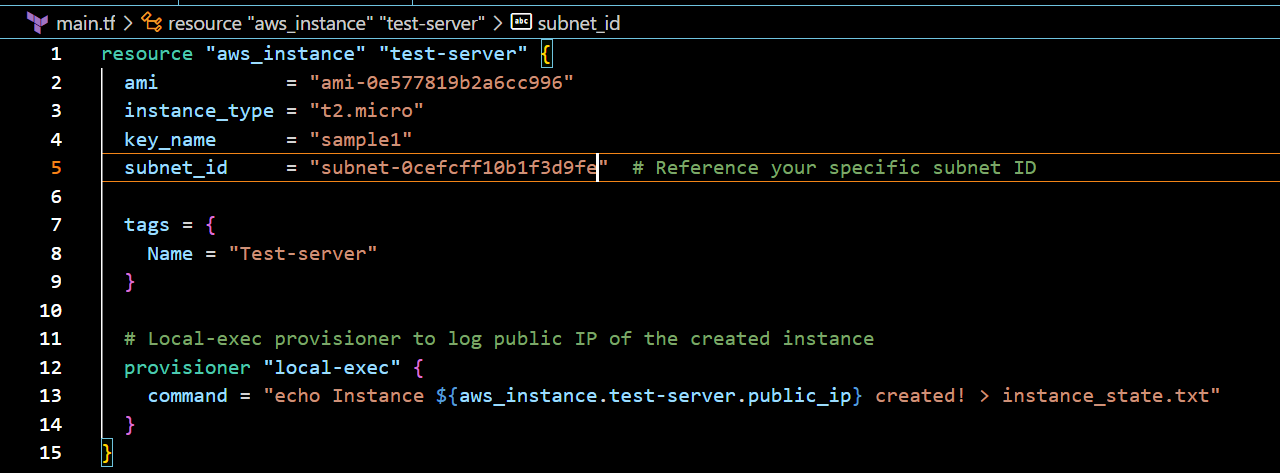
This is used to execute commands at the run time on remote machines.

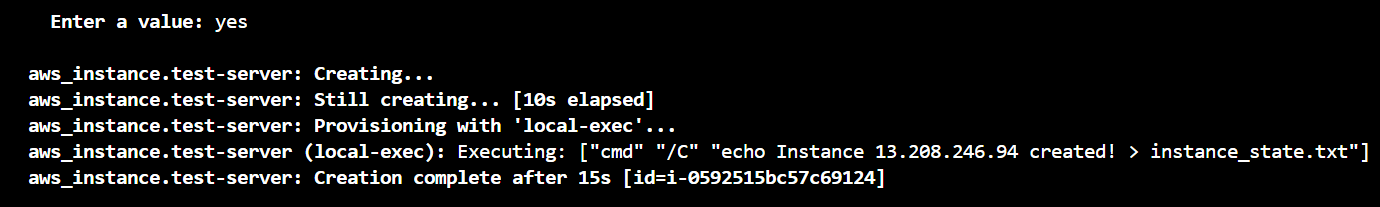
**2) Local provisioner**

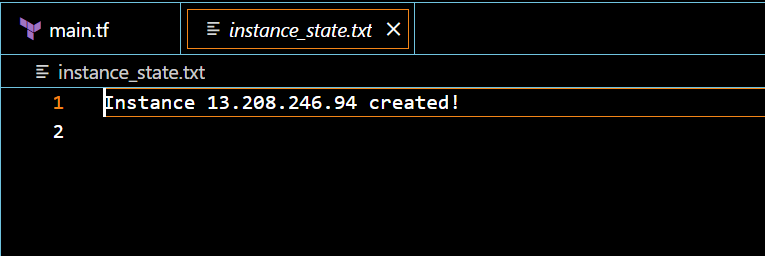
This is used to execute commands at the run time on local

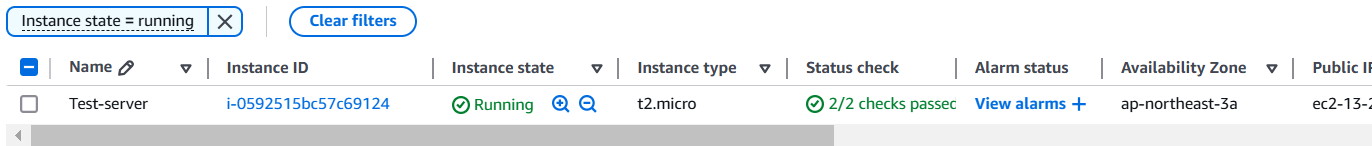
machine. (means where terraform is installed)

**Example**







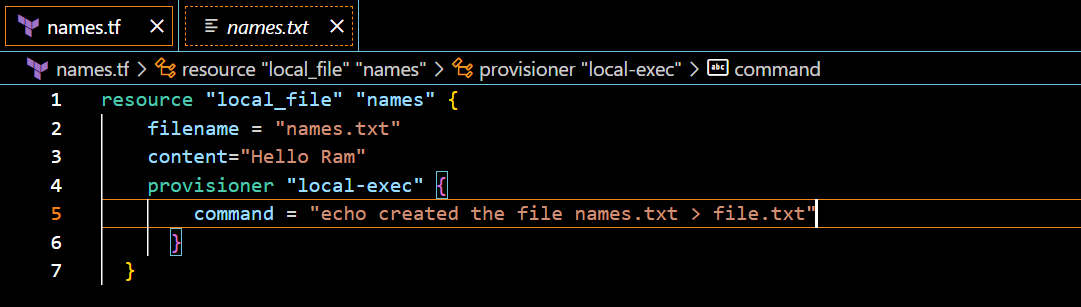


**Terraform taint and untaint**

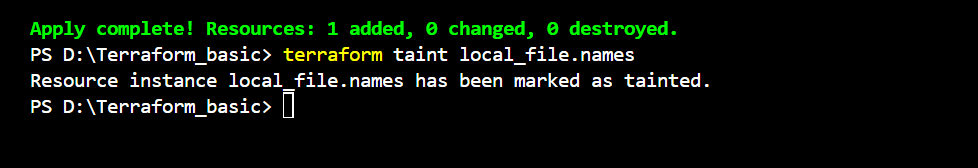
The would be cases when resource creation will get failed,if this happens

then terraform will be marked as "Tainted".

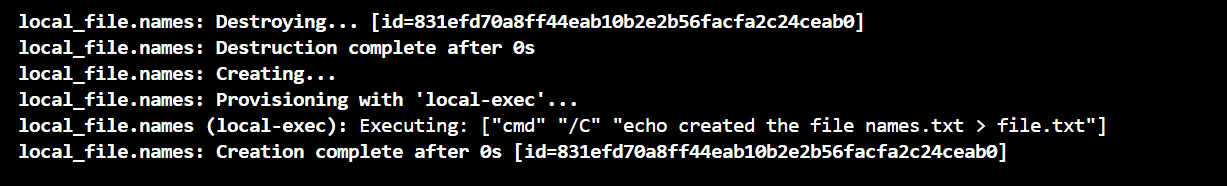
we can see this when we execute terraform plan command and this will

be replaced when we use terraform apply command.

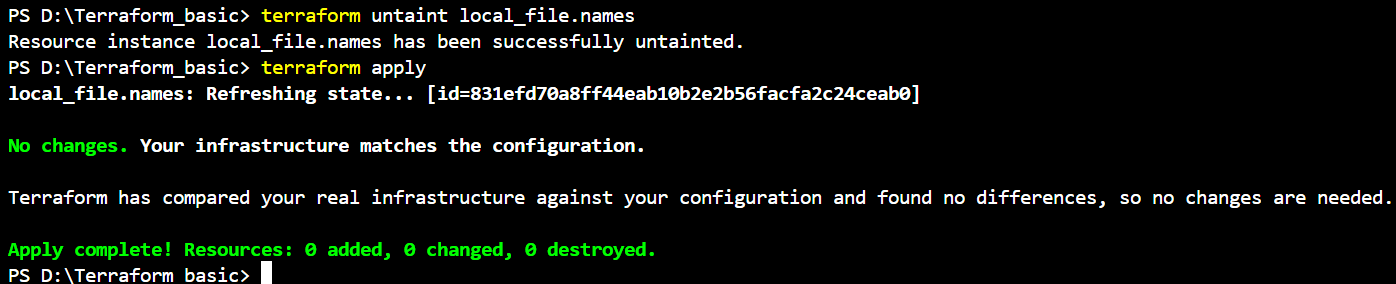
By using using cli,making the status as taint





If we do terraform apply [The existing file is replaced as we marked it as tainted]

**Untaint[Not to replace the resources]**

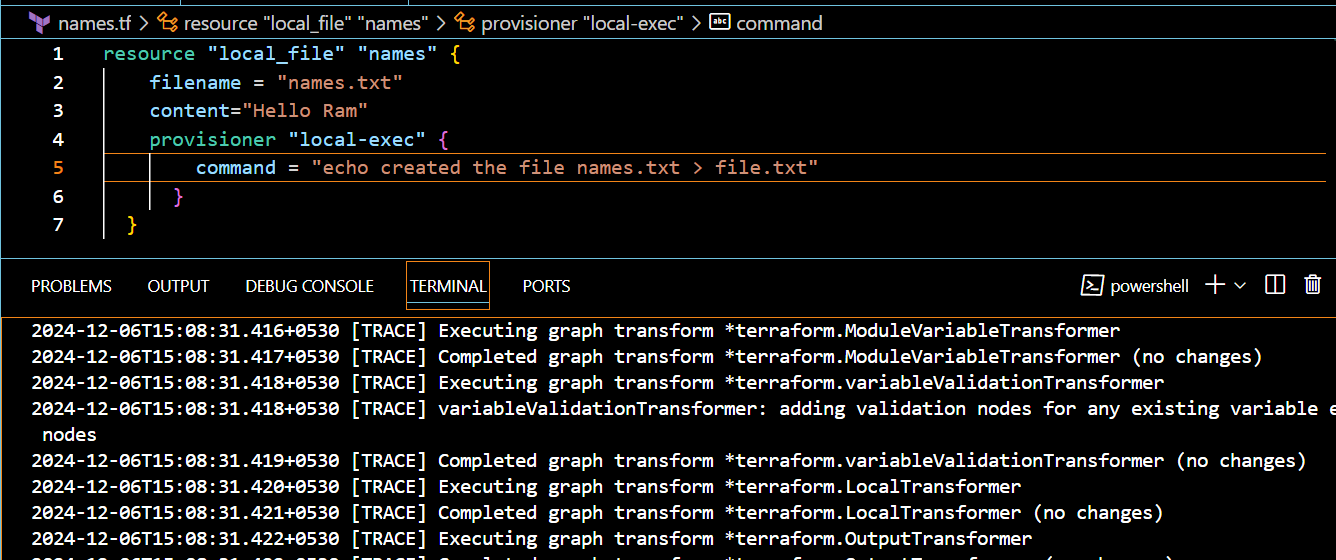


**DEBUGGING**

terraform apply will provide us the logs/cause of the issue,

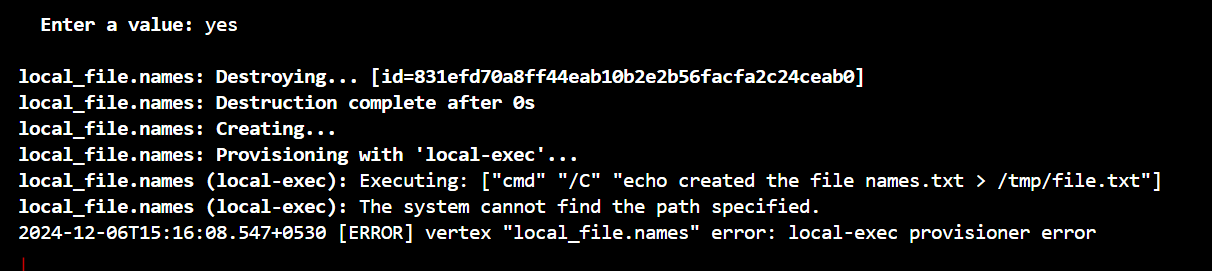
but still if we want to dig deeper then we need to export a variable

**cmd:** **Set-Item -Path env:TF\_LOG -value "TRACE"**



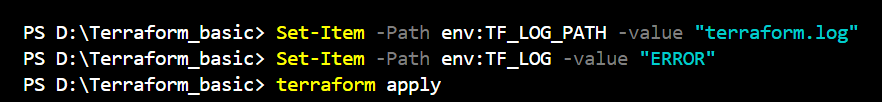
**If we want only errors**

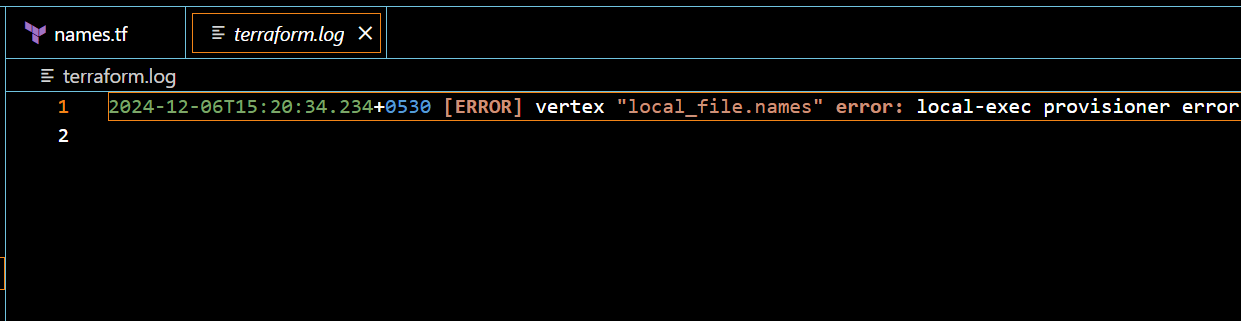
**Set-Item -Path env:TF\_LOG -value "ERROR"**



If we want to store that errors permanently in terraform logs

**Set-Item -Path env:TF\_LOG\_PATH -value "terraform.log"**



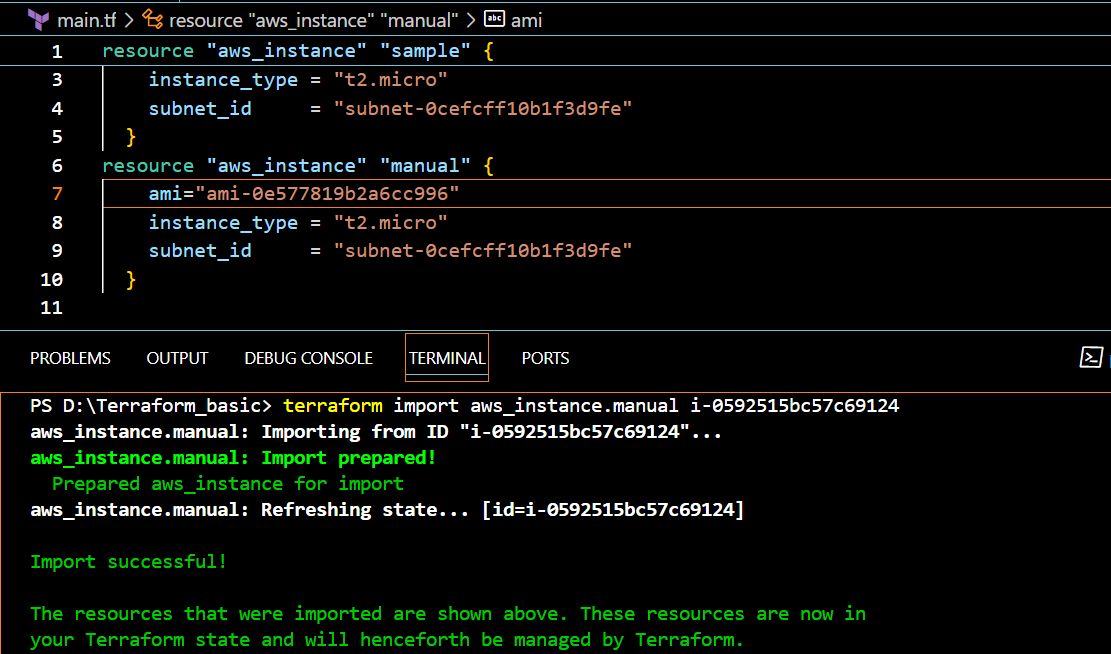


**Terraform import:**

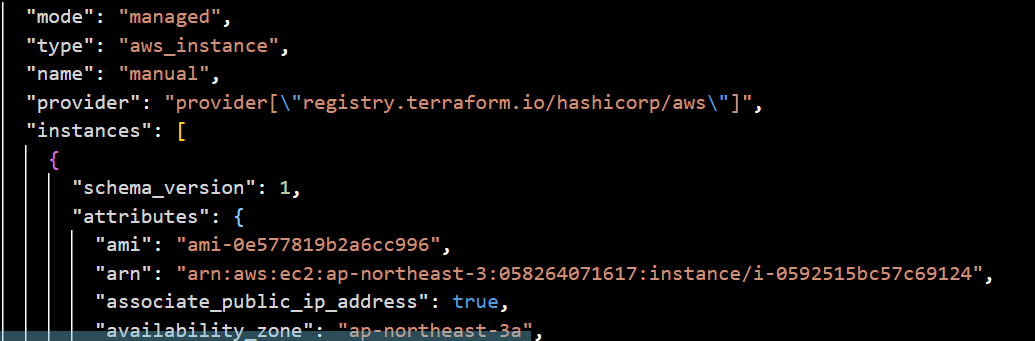
Terraform import is used to import the existing infrastructure in terraform state file.

Once import is done then we cna be able to create/delete and manage the infrastructure.

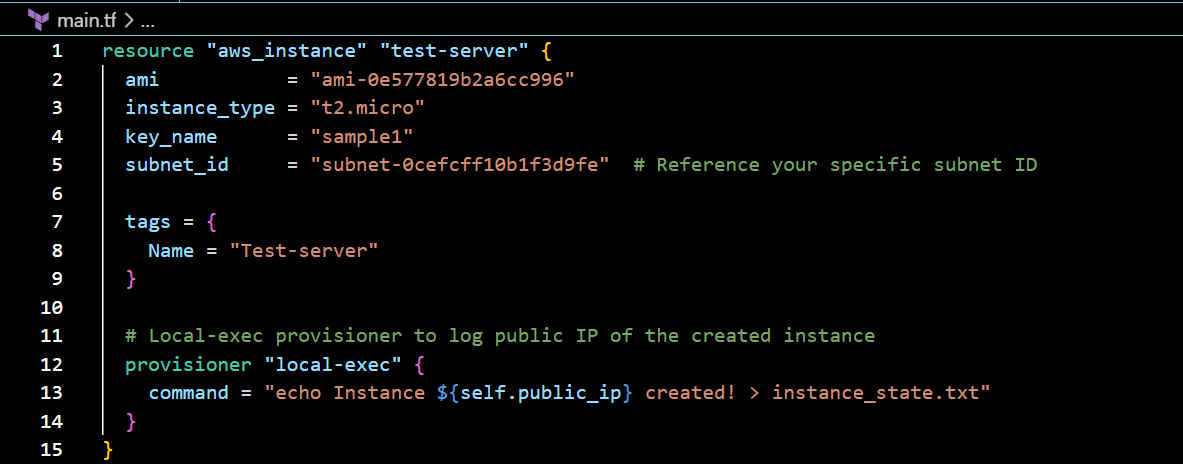
In order to import any resource we need to write the resource details in configuration file.



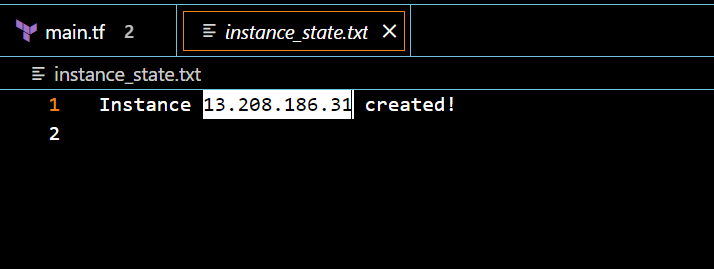
**In terraform.tfstate**



**3) Create one ec2 instance with httpd installed using terraform script.**



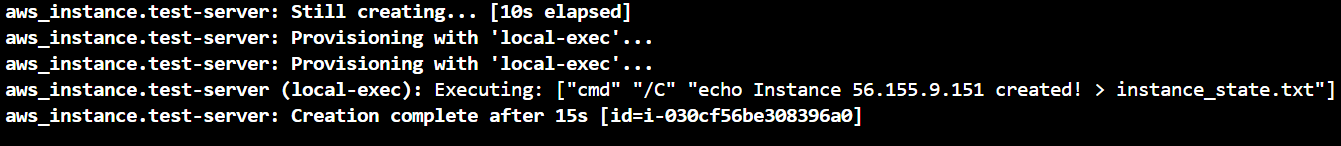
Run the command terraform init and terraform apply



Ec2-instances created

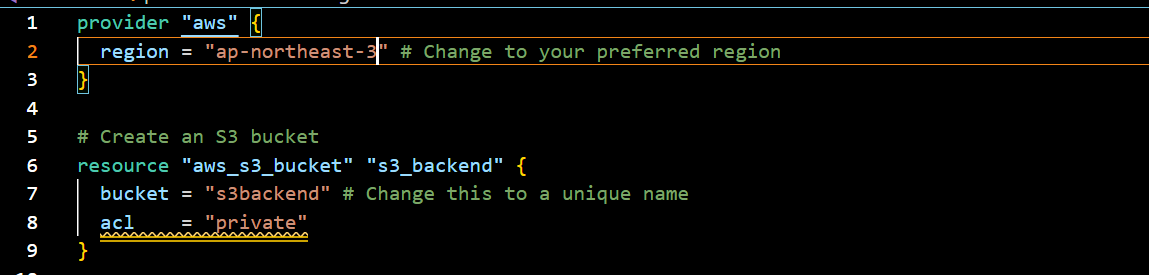
Now adding the user data to the terraform

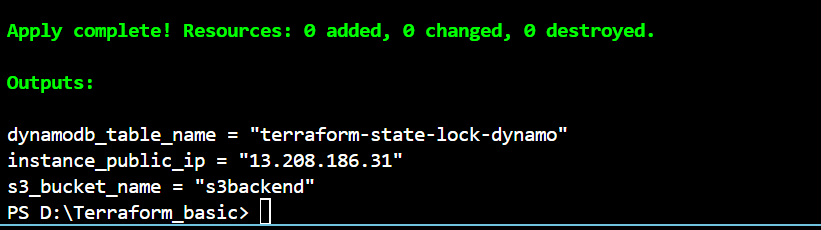


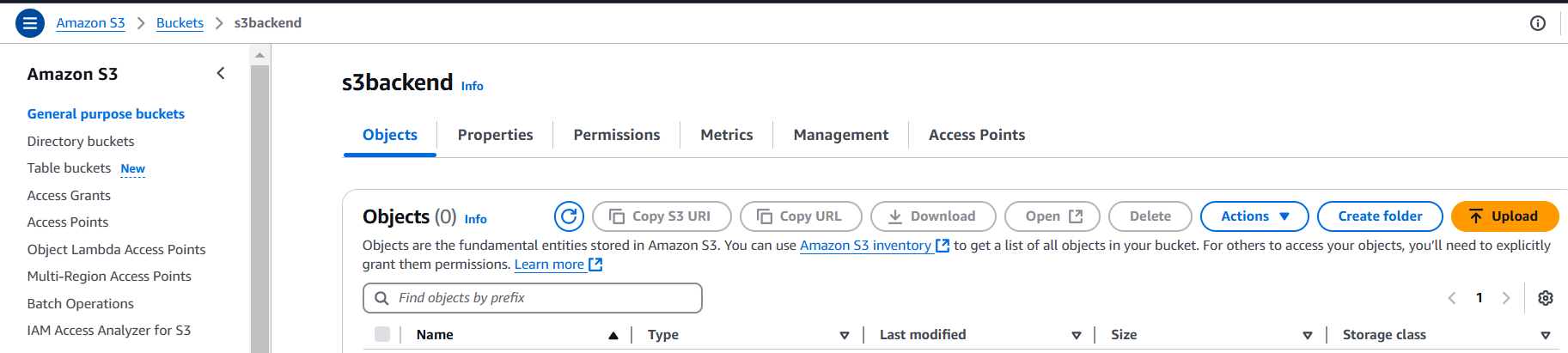


 **4) Setup s3 as backend to the task 3.**

Adding the s3 bucket to the existing the main.tf

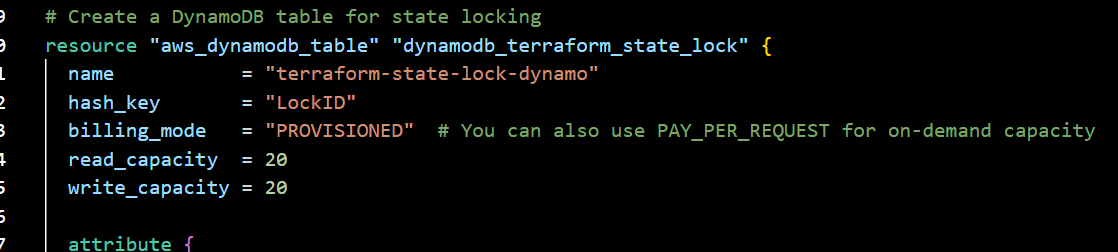


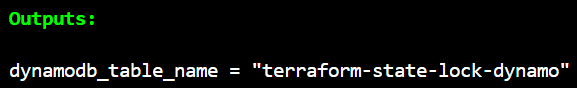


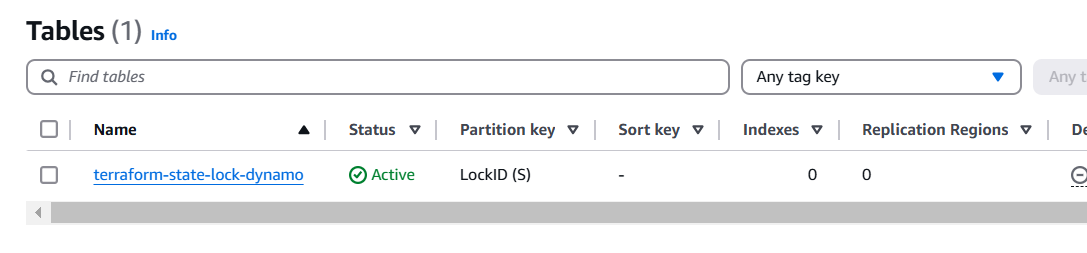


**5) Setup dynamo db locking for task3.**

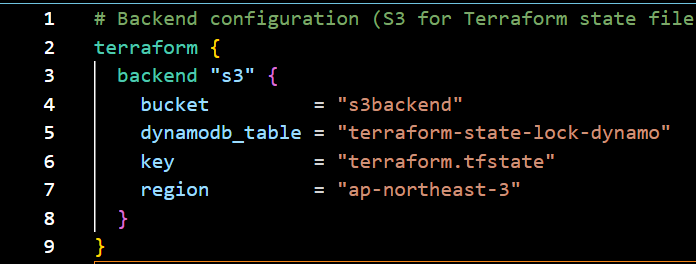
Created a dynamo db table

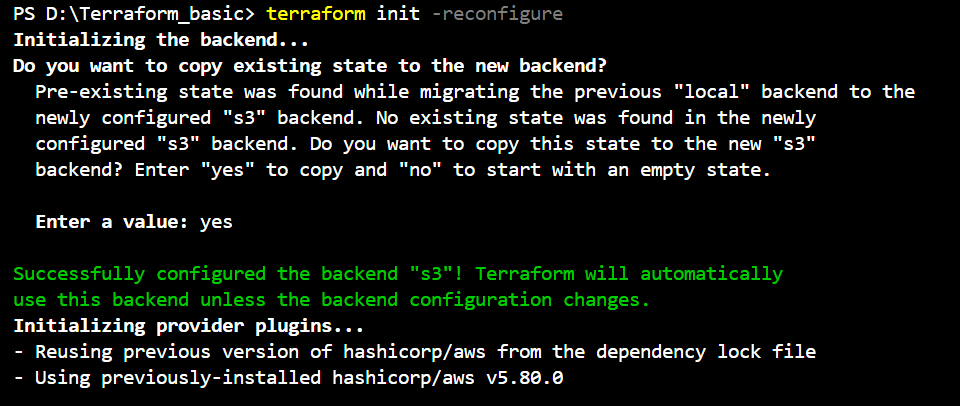


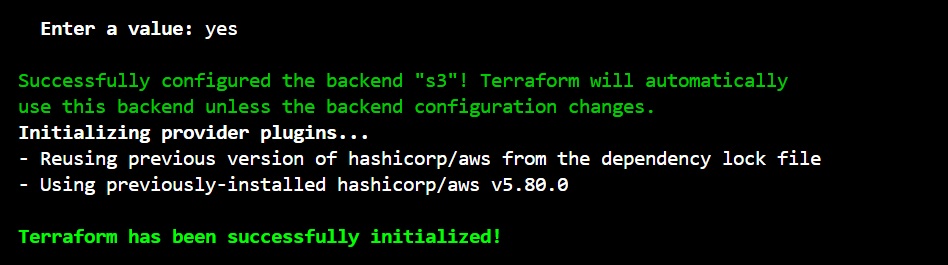


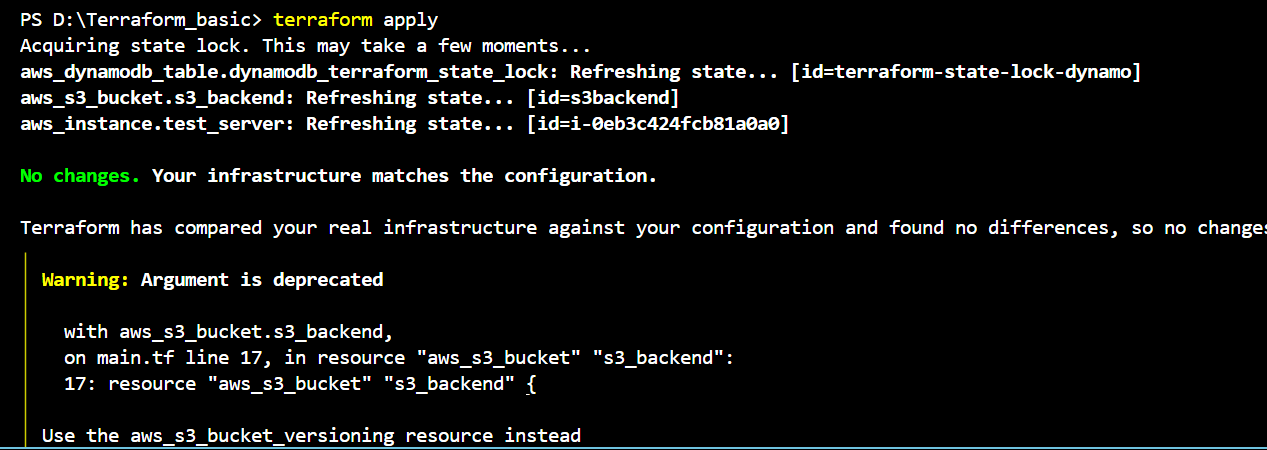


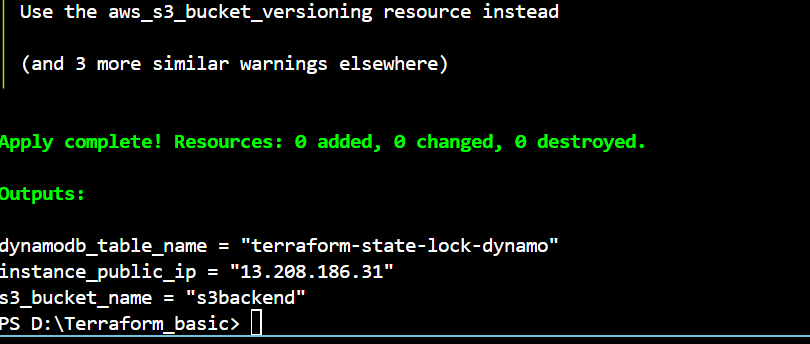
Now s3 as backend for terraform.tfstate





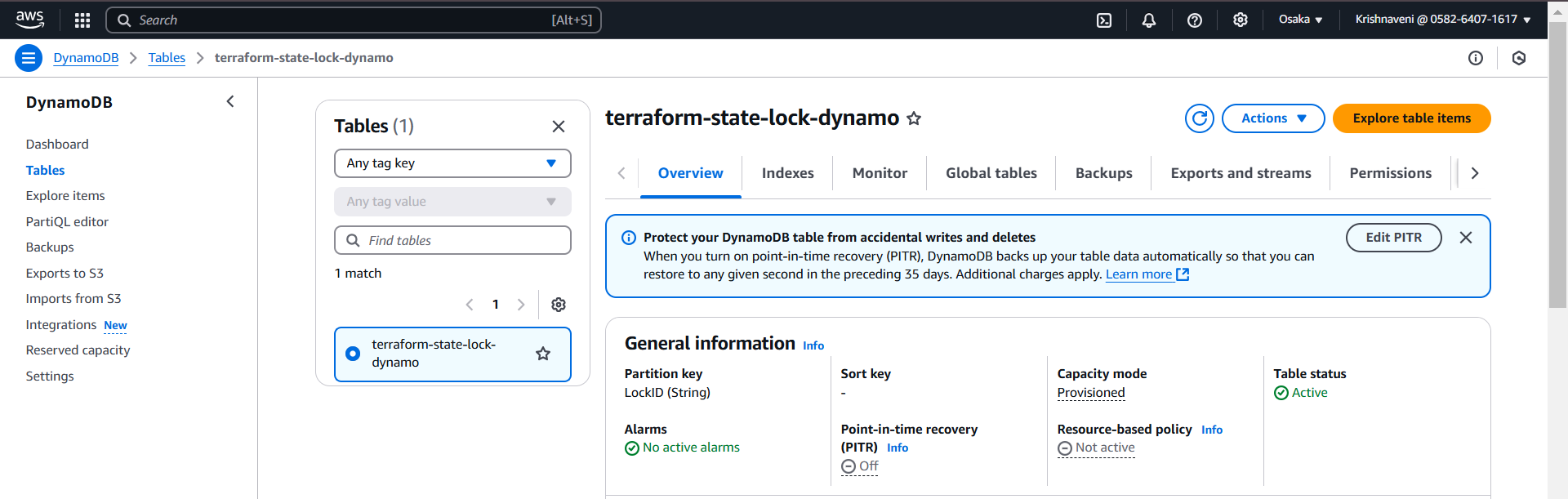






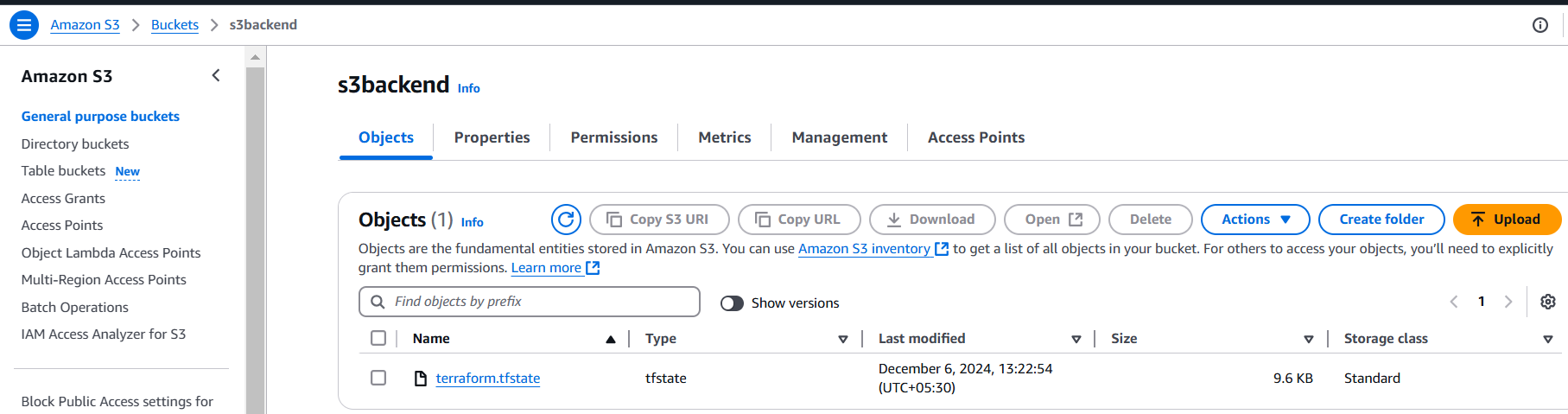
Output:

In dynamodb tables

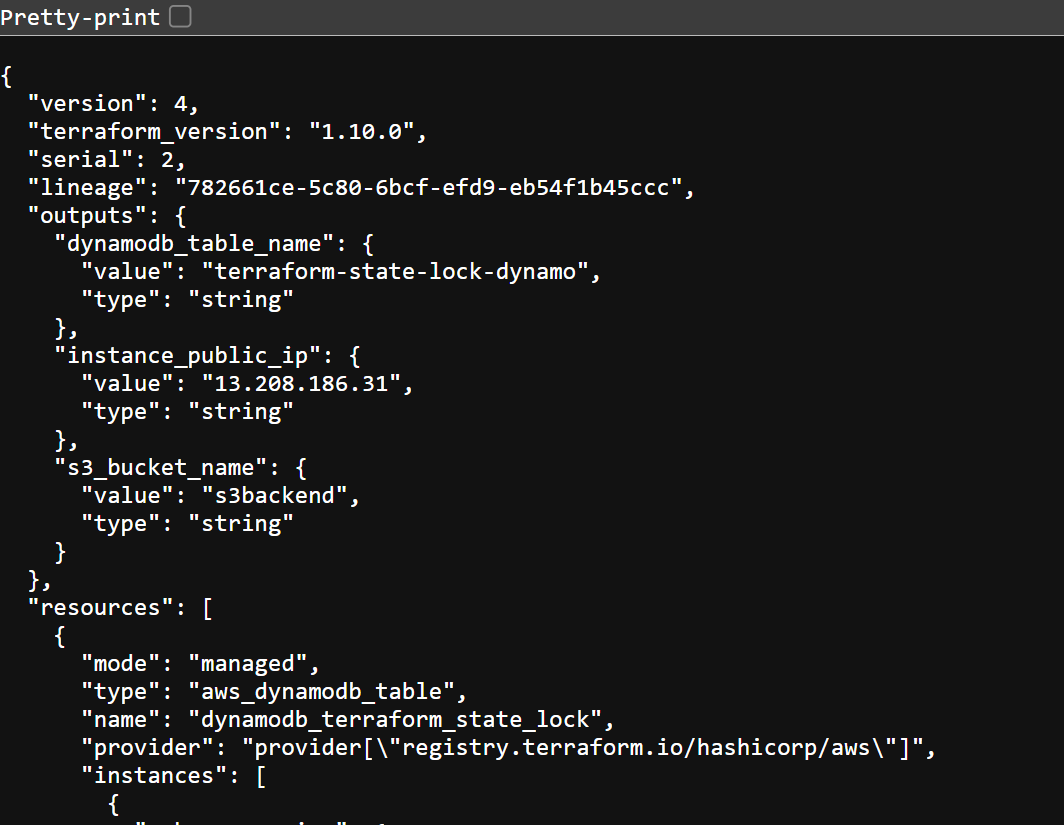


And our terraform.tfstate is stored in s3 bucket





If we open the terraform.tffile



We can see complete blueprint of our Infrastructure