**Experiment No 6**

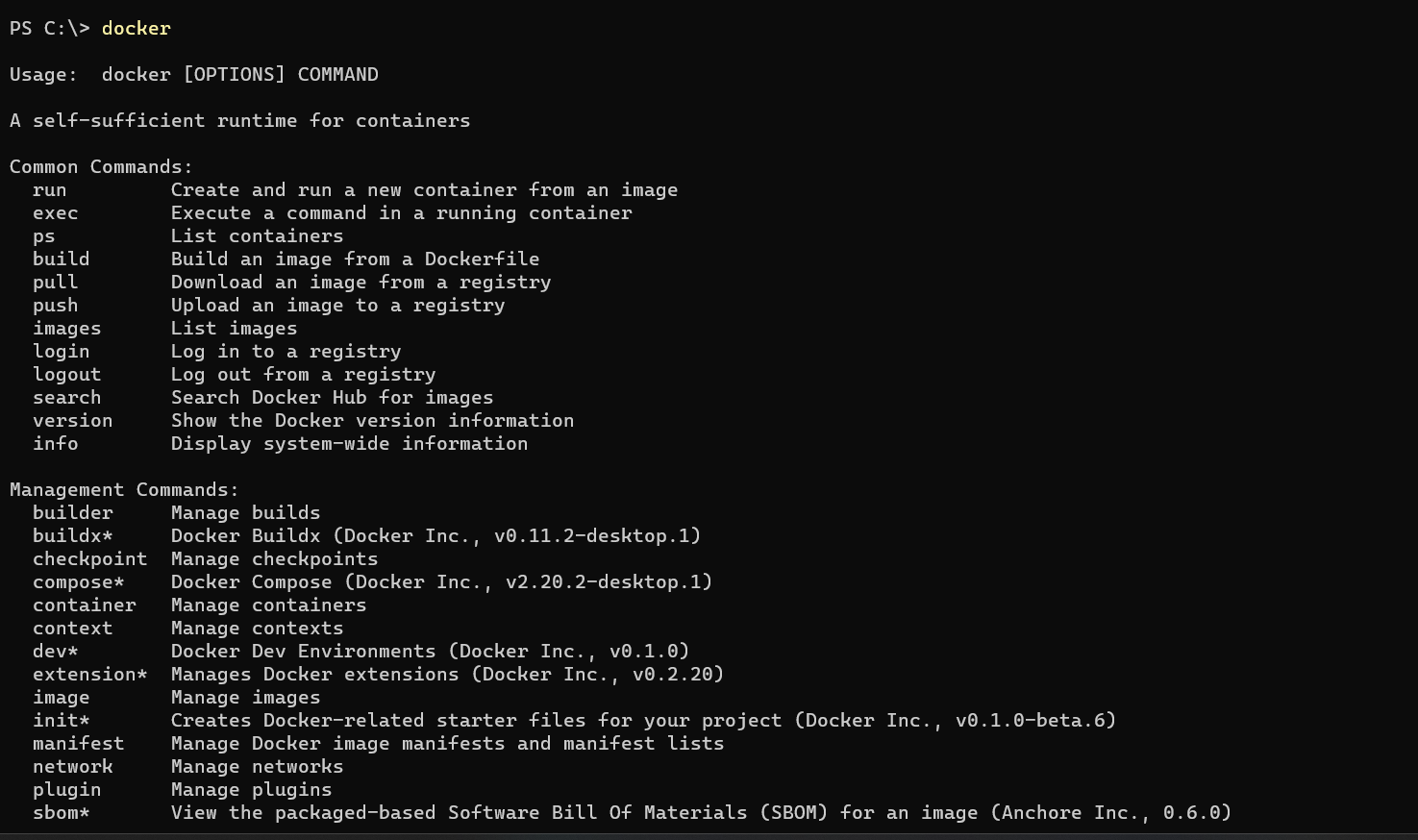
**Shashwat Tripathi  
D15A 64  
Batch C**

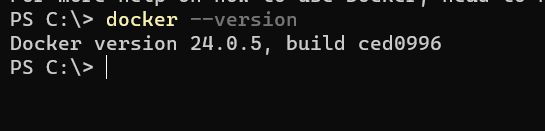
**AIM: Creating docker image using terraform**

Prerequisite:

1) Download and Install Docker Desktop from https://www.docker.com/

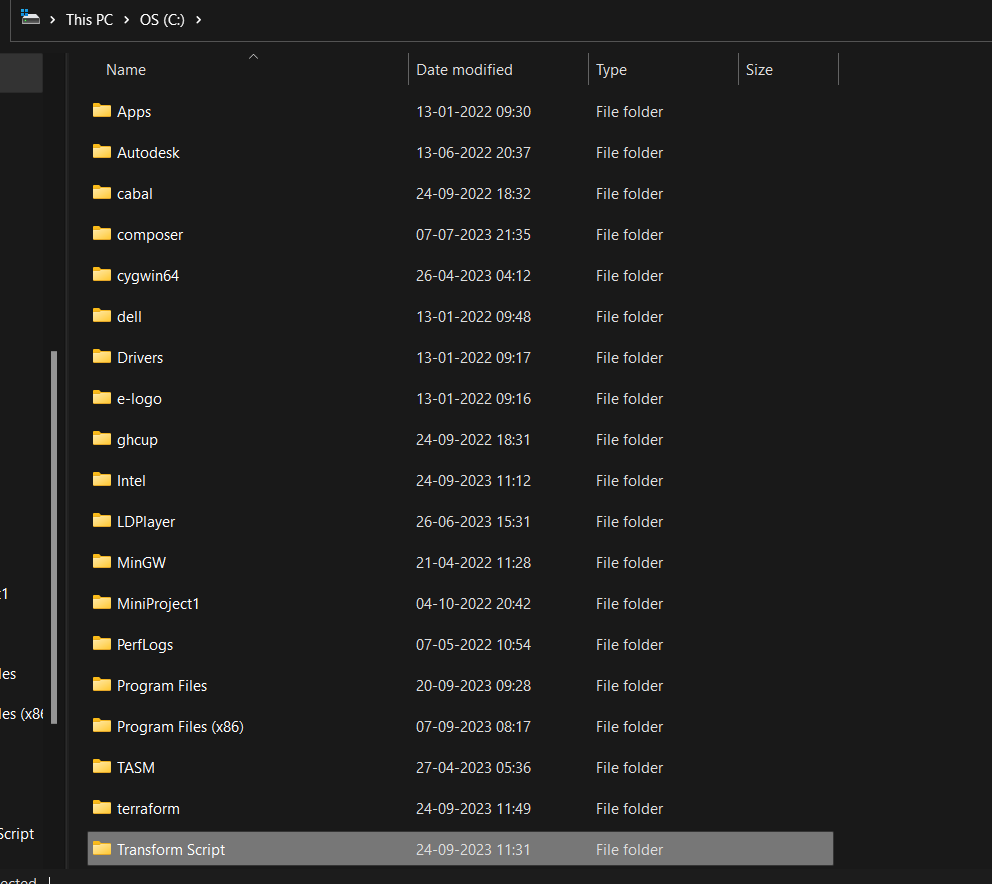
**Step 1:** Check the docker functionality

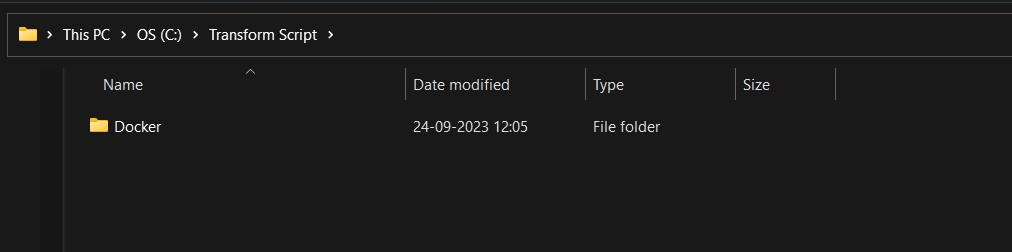




**Now, create a folder named ‘Terraform Scripts’ in which we save our different types of scripts which will be further used in this experiment.**

**Step 2:** Firstly create a new folder named ‘Docker’ in the ‘TerraformScripts’ folder.





**Then create a new docker.tf file using vs code editor and write the following contents into it to create a Ubuntu Linux container.**

Script:

terraform{

required\_providers{

docker = {

source = "kreuzwerker/docker"

version = "2.21.0"

}

}

}

provider "docker" {

host = "npipe:////.//pipe//docker\_engine"

}

# Pulls the image

resource "docker\_image" "nginx" {

name = "nginx:latest"

keep\_locally =false

}

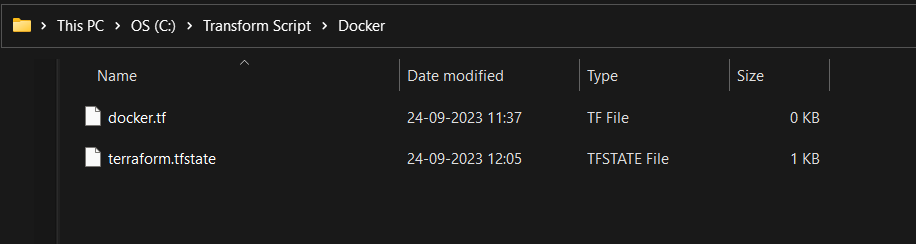
# Create a container

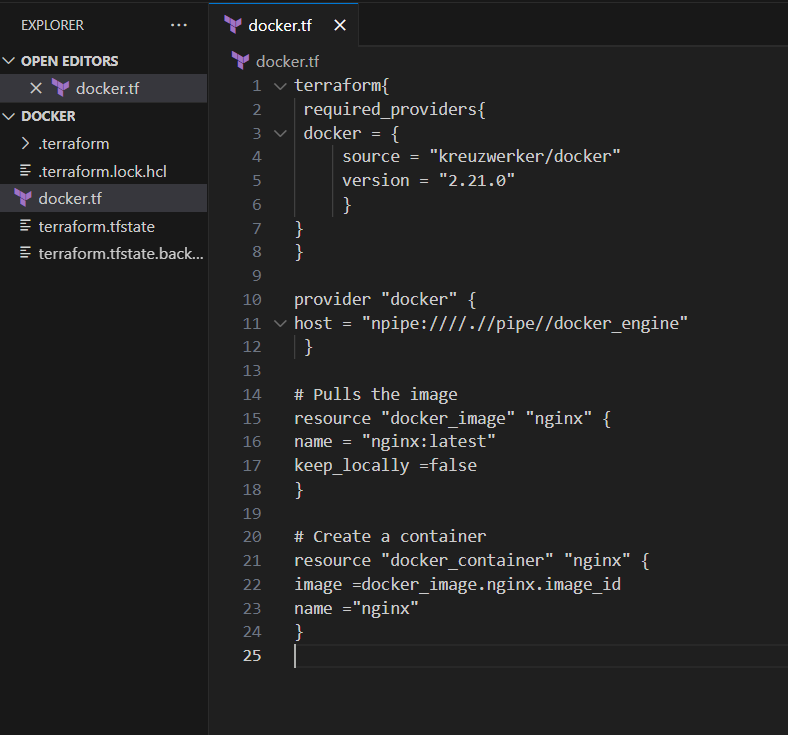
resource "docker\_container" "nginx" {

image =docker\_image.nginx.image\_id

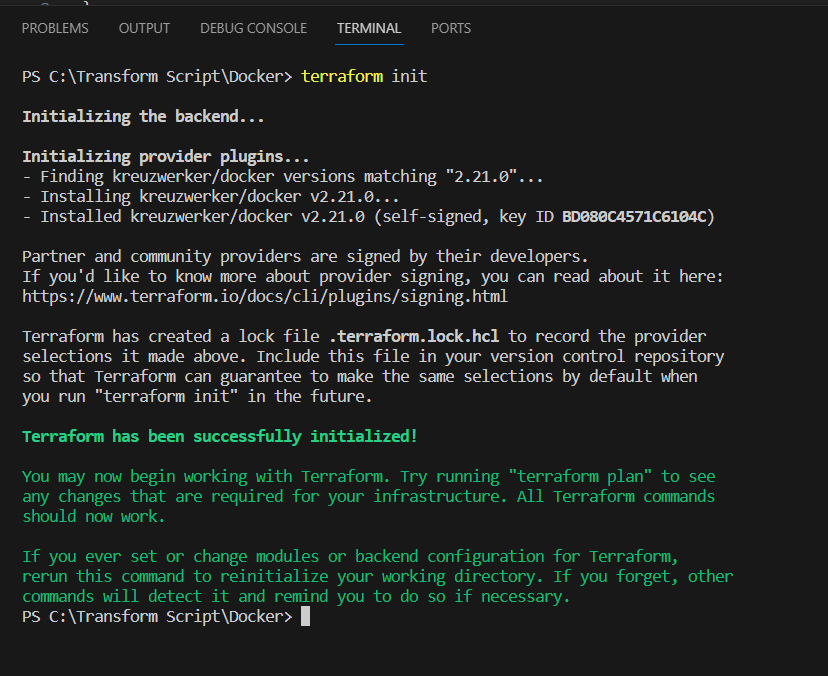
name ="nginx"

}

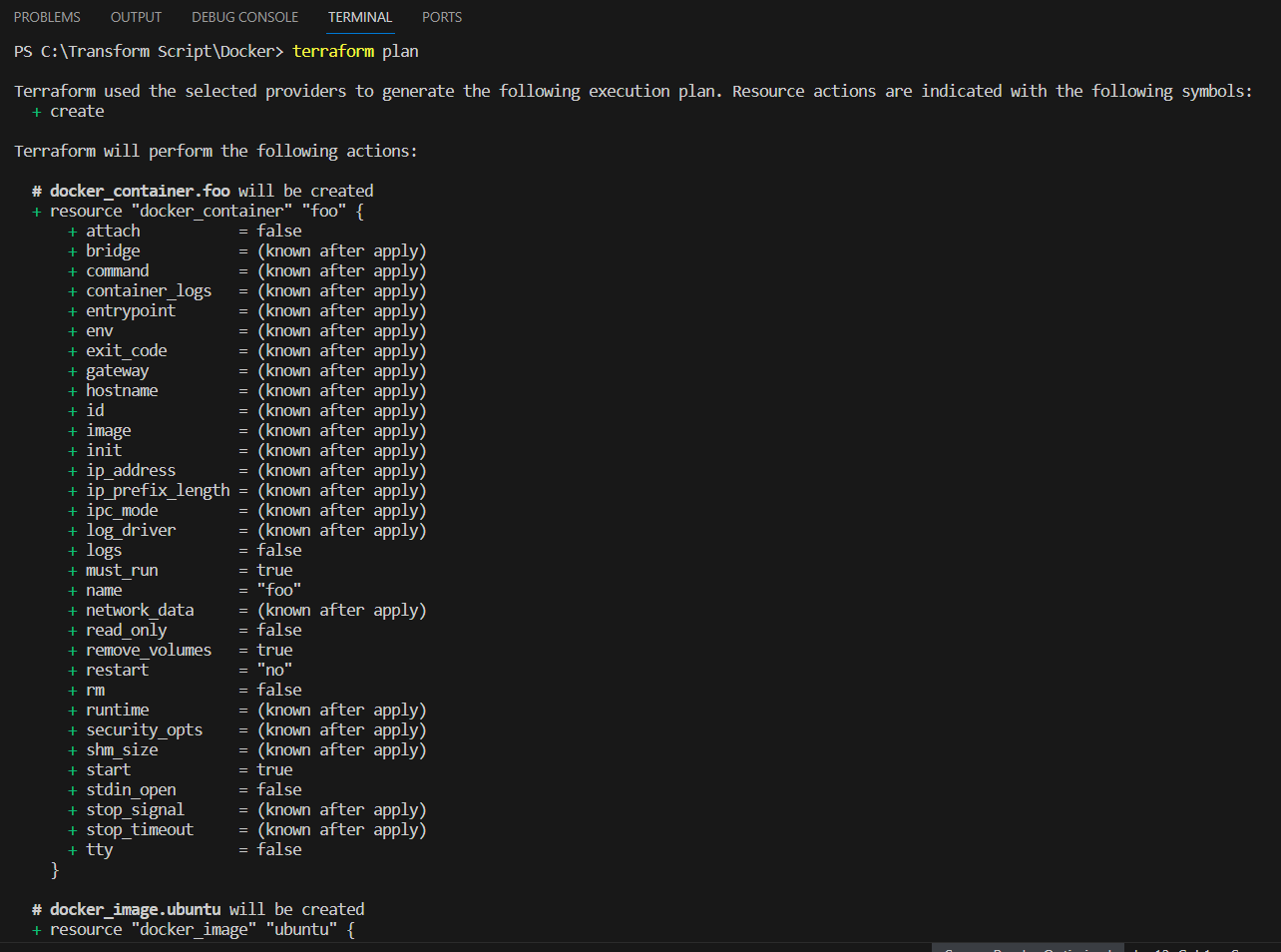


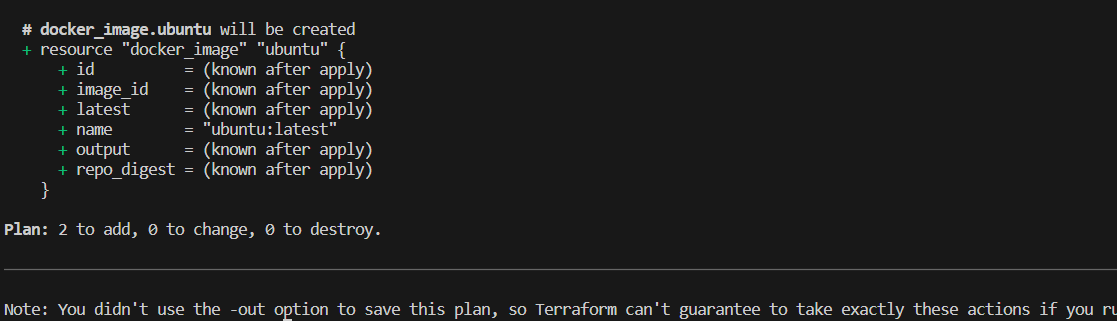


**Step 3:** Execute Terraform Init command to initialize the resources

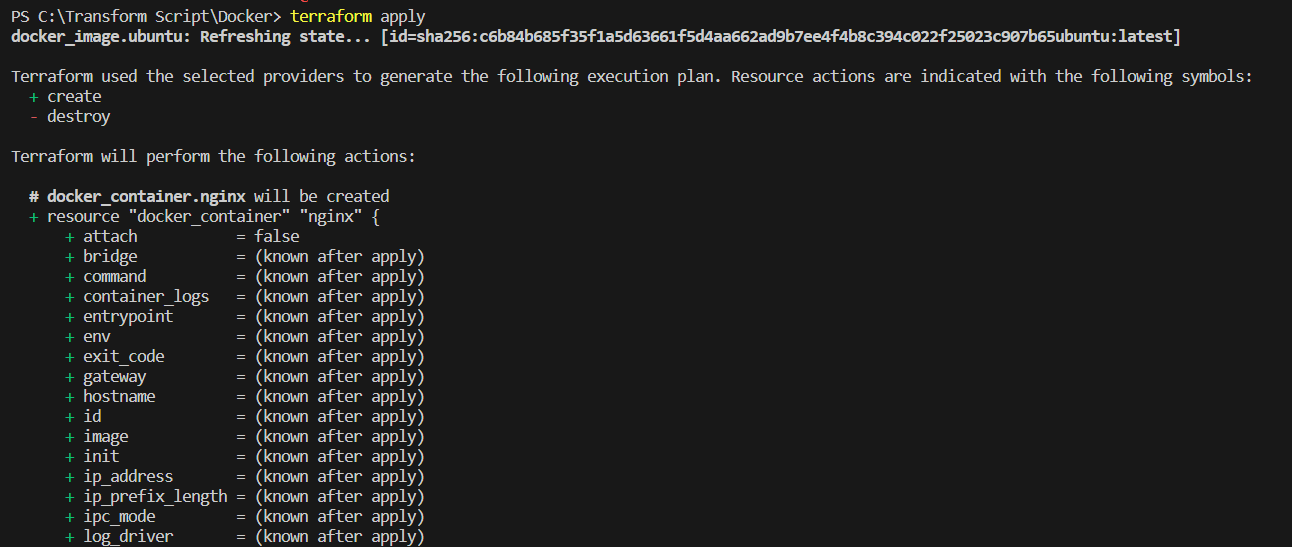


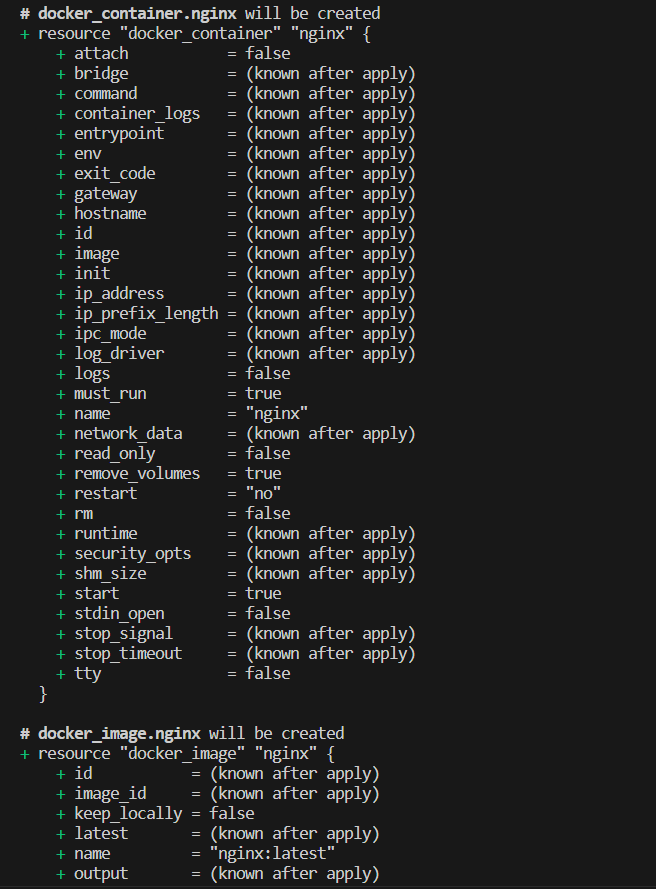
**Step 4:** Execute Terraform plan to see the available resources

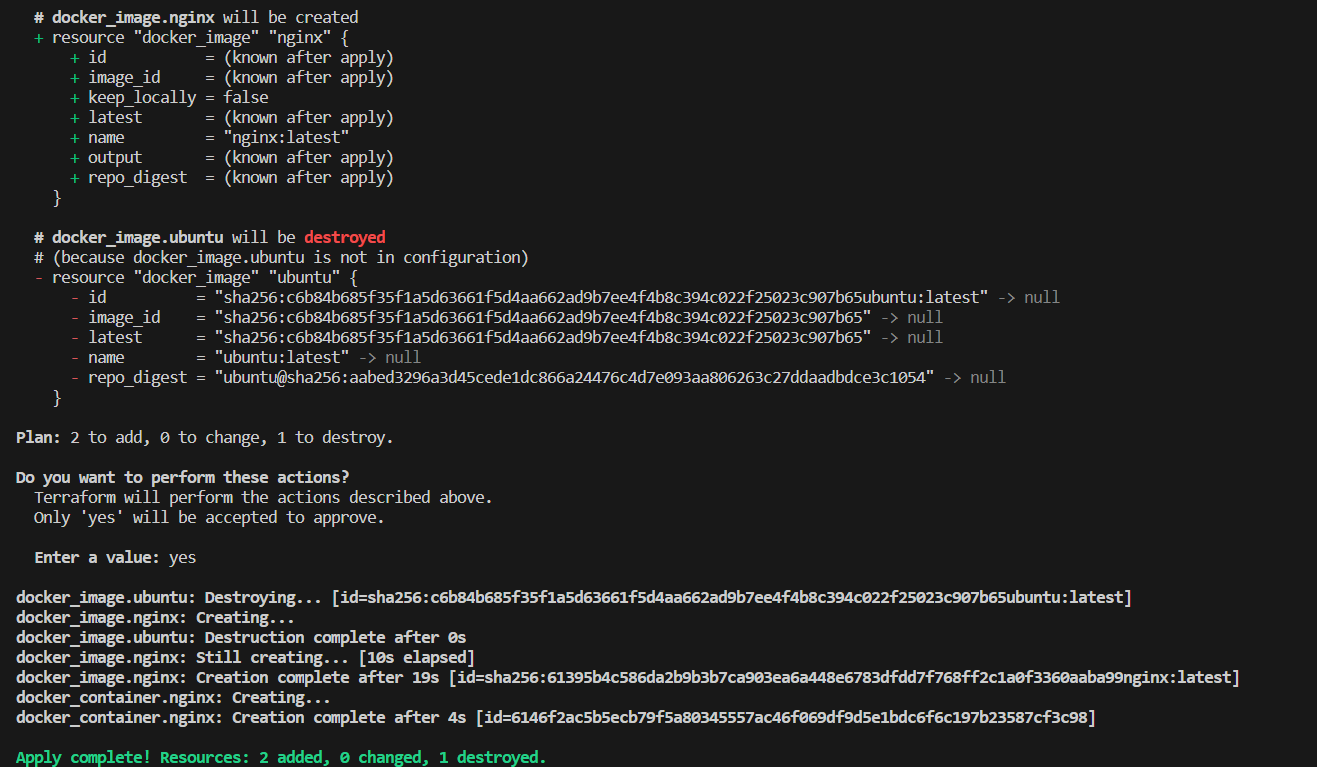




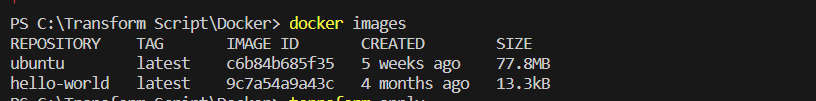
**Step 5:** Execute Terraform apply to apply the configuration, which will automatically create and run the Ubuntu Linux container based on our configuration. Using command : “**terraform apply”**



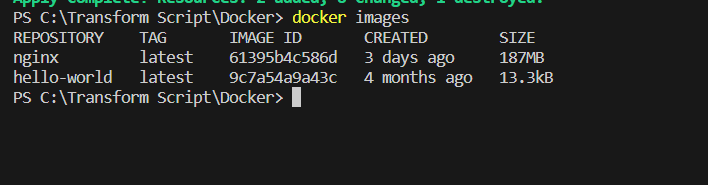




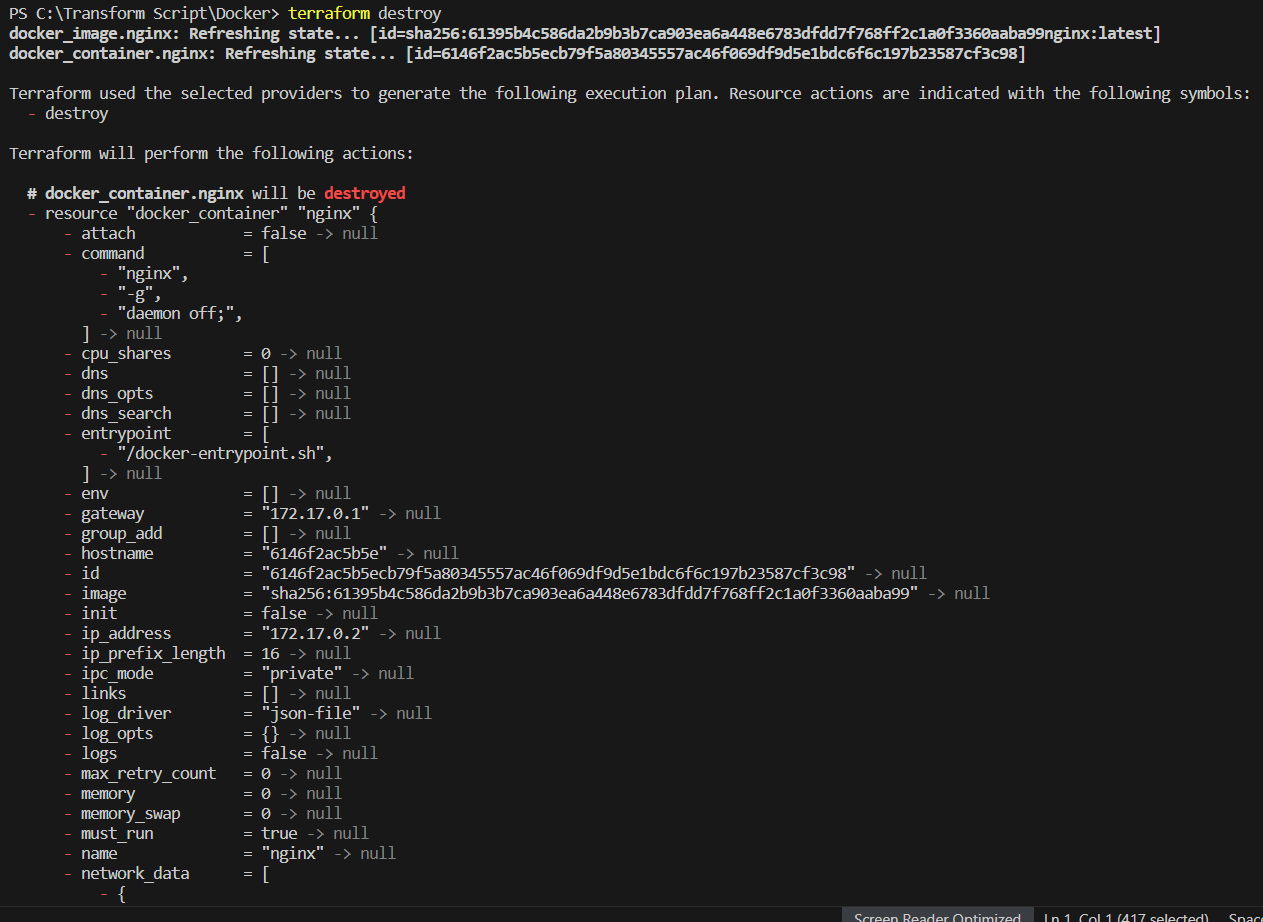
Docker images, Before Executing Apply step:

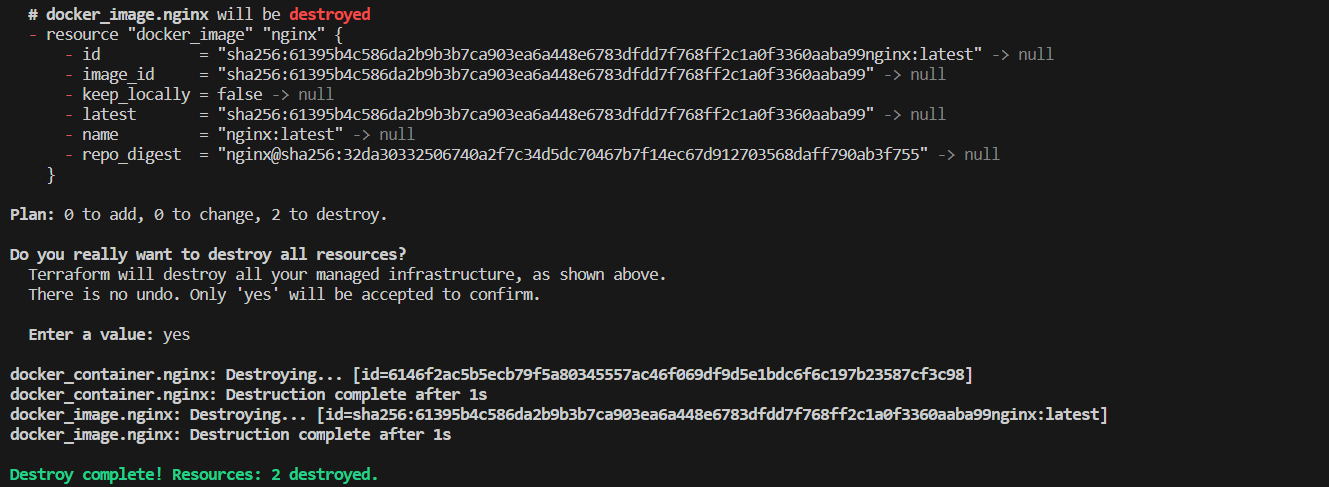


Docker images, After Executing Apply step

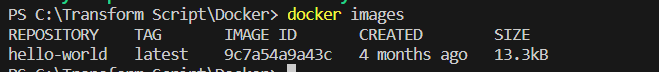


**Step 6:** Execute Terraform destroy to delete the configuration, which will automatically delete the Ubuntu Container.





**Docker images After Executing Destroy step**



**Conclusion:** Thus, we have created a docker image using Terraform.