**Rujuta Medhi D15A-28**

**EXPERIMENT NO. 6**

**Aim :**To Build, change, and destroy AWS infrastructure Using Terraform (S3 bucket or Docker) .

**Theory:**

### What is Terraform?

Terraform is an open-source Infrastructure as Code (IaC) tool developed by HashiCorp. It allows you to define, provision, and manage infrastructure resources in a consistent and repeatable way. Terraform uses a high-level configuration language called HashiCorp Configuration Language (HCL) or JSON to describe the desired state of your infrastructure, and it can manage resources across various cloud providers, including AWS, Azure, Google Cloud, and others.

### Benefits of Using Terraform with AWS

1.Multi-Cloud Support: Terraform can manage infrastructure across multiple cloud providers, including AWS. This allows you to create a consistent infrastructure environment across different clouds or migrate between them.

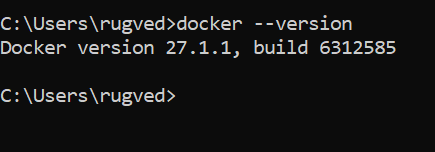
2.Infrastructure as Code: By using Terraform, you can define your AWS infrastructure as code. This makes it easier to version control your infrastructure, automate deployments, and collaborate with others.

3. Scalability and Flexibility: Terraform can manage infrastructure of any size, from small projects to large-scale, complex environments. It allows you to define reusable modules, which can be shared across different projects.

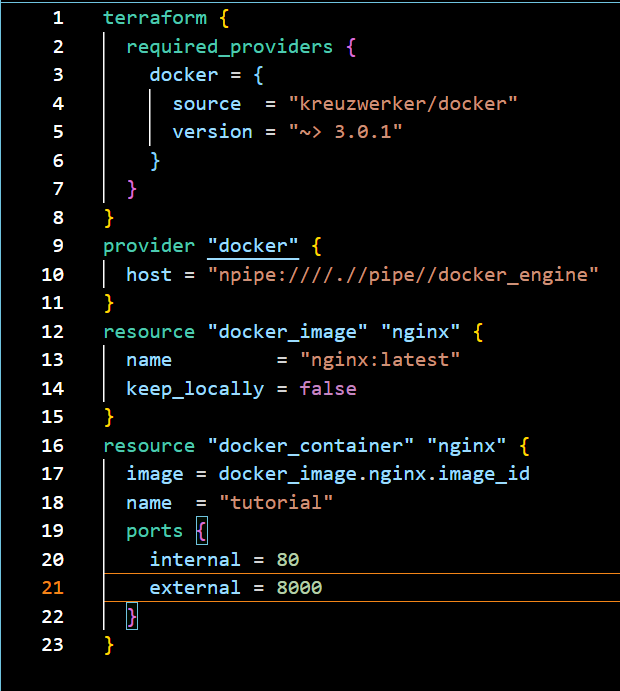
4. State Management: Terraform keeps track of the current state of your infrastructure in a state file. This allows Terraform to determine the actions required to achieve the desired state, ensuring that your infrastructure remains consistent.

**Implementation** :

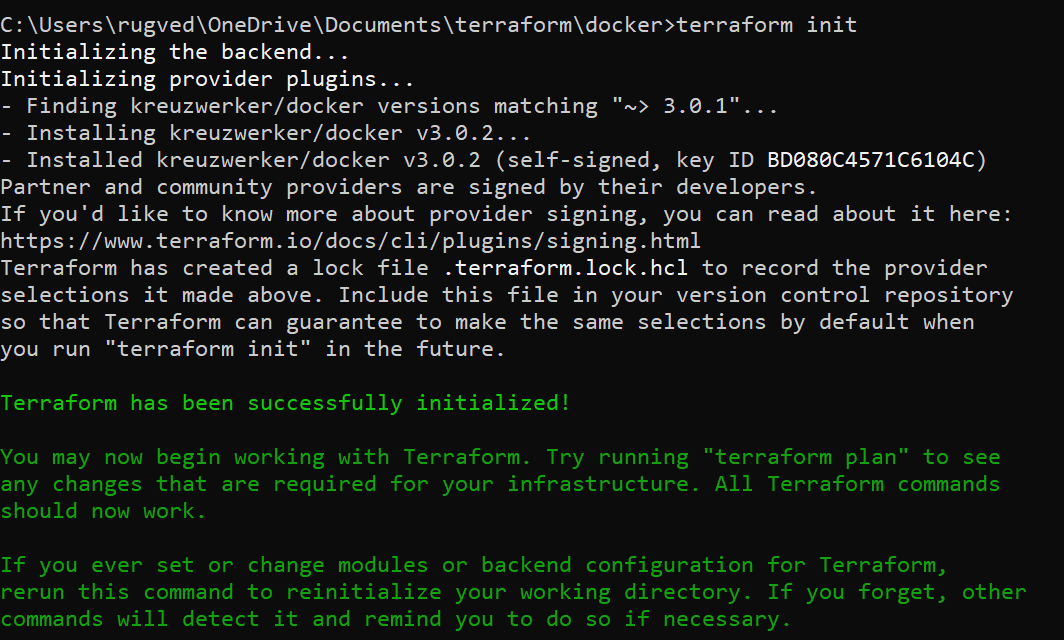
Step 1 : check docker installation and version



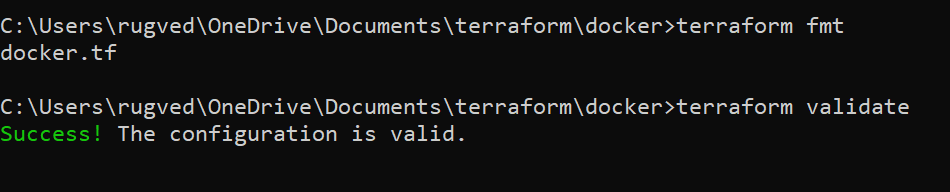
Step 2 : create docker.tf file and write following code for terraform and docker



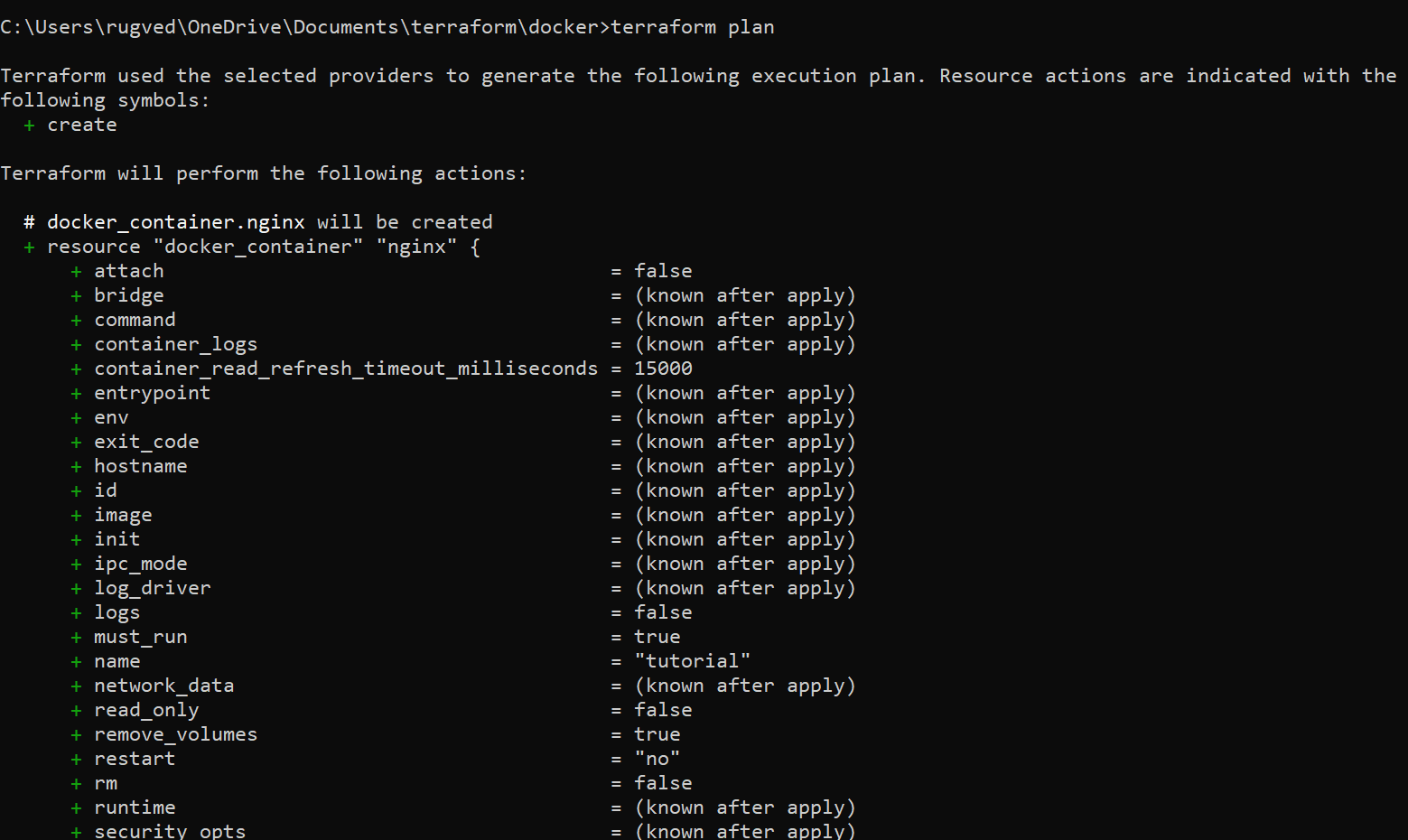
Step 3 : Type terraform init command to initialize terraform backend

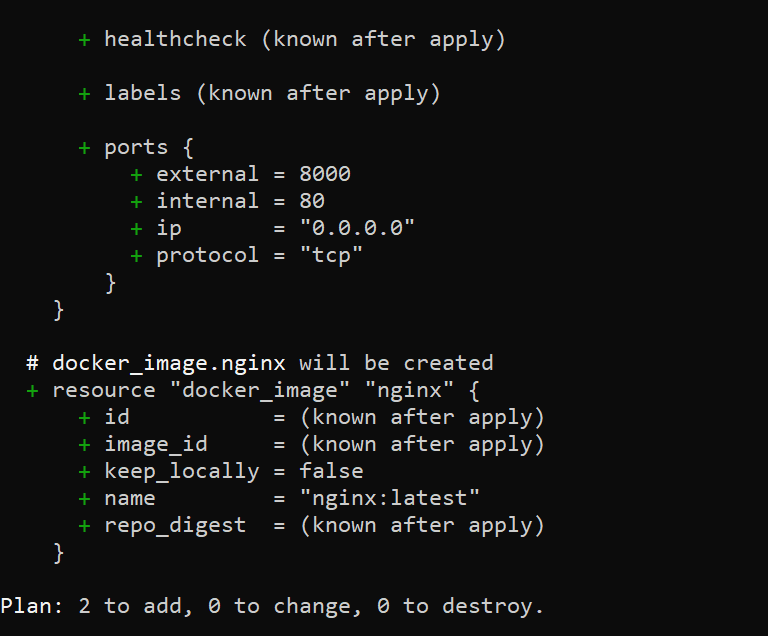
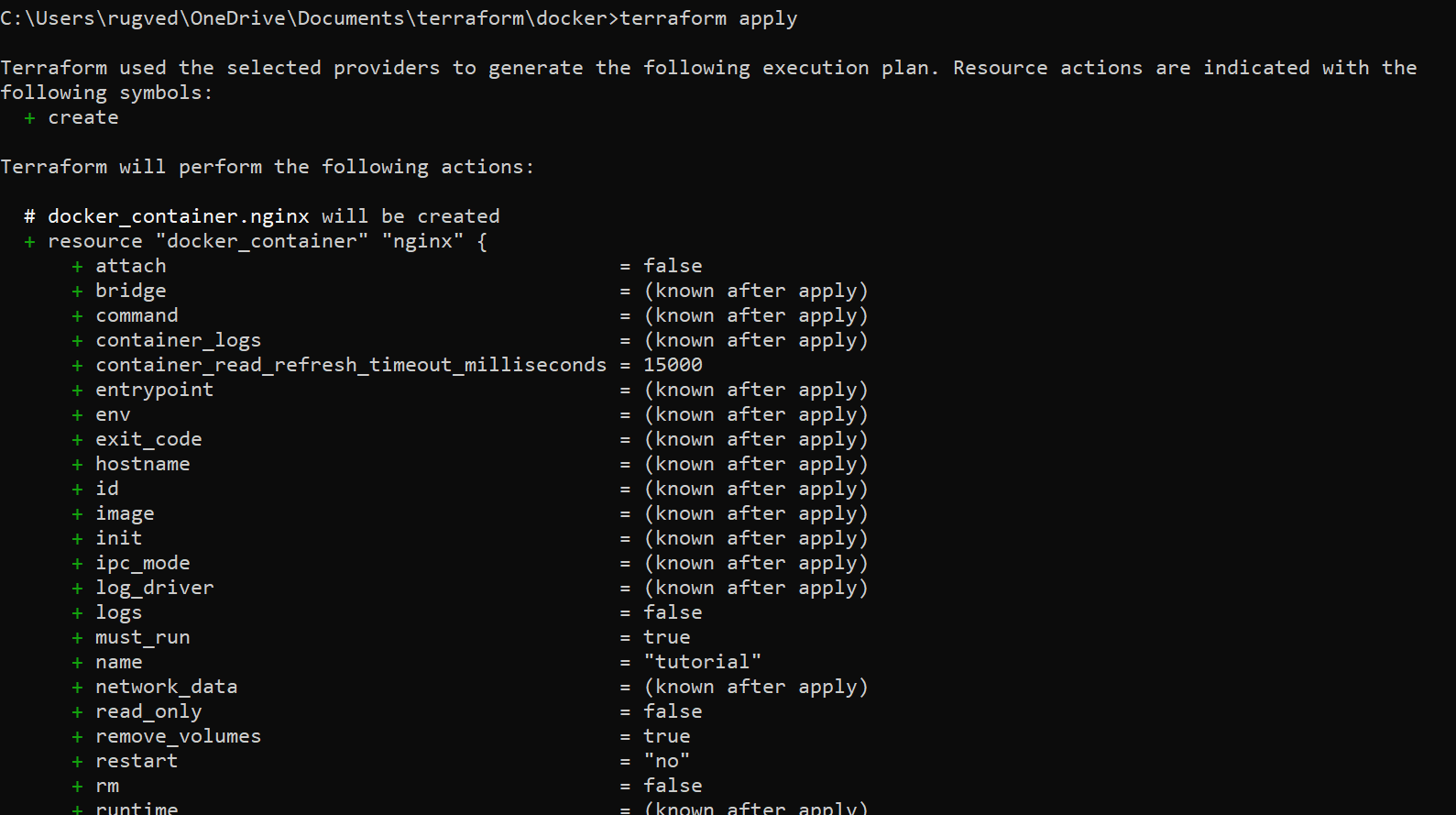


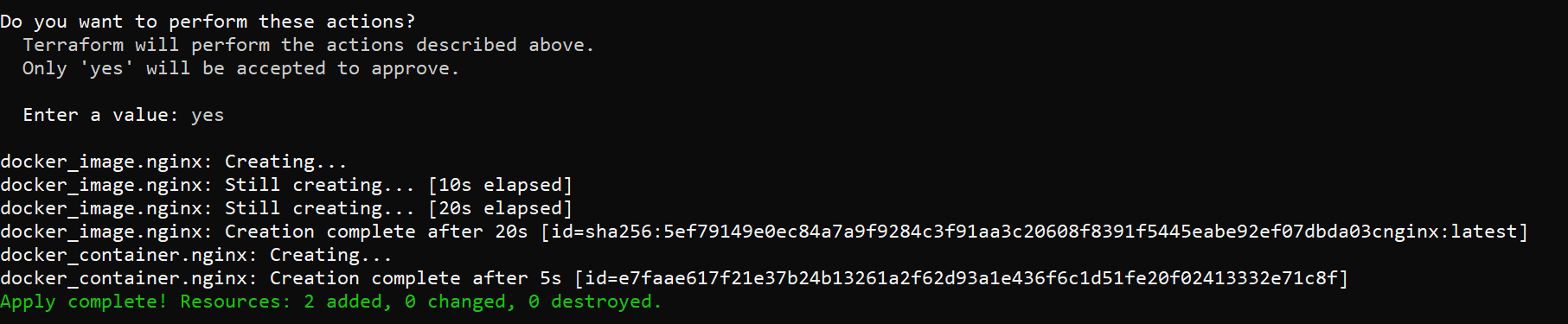
Step 4(EXTRA) : type terraform fmt and validate commands . The two Terraform commands – terraform validate and terraform fmt – are used to maintain a clean, error-free, and well-structured Terraform codebase.



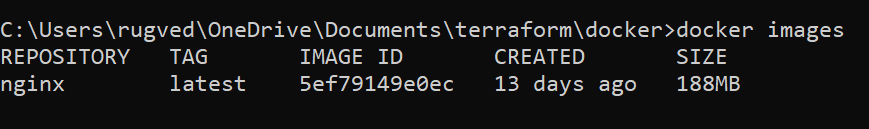
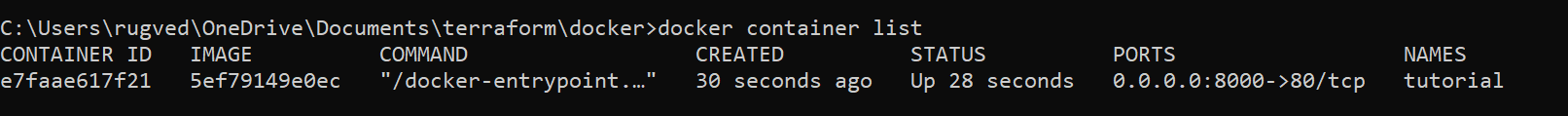
Step 5 : Type Terraform plan command to create execution plan .



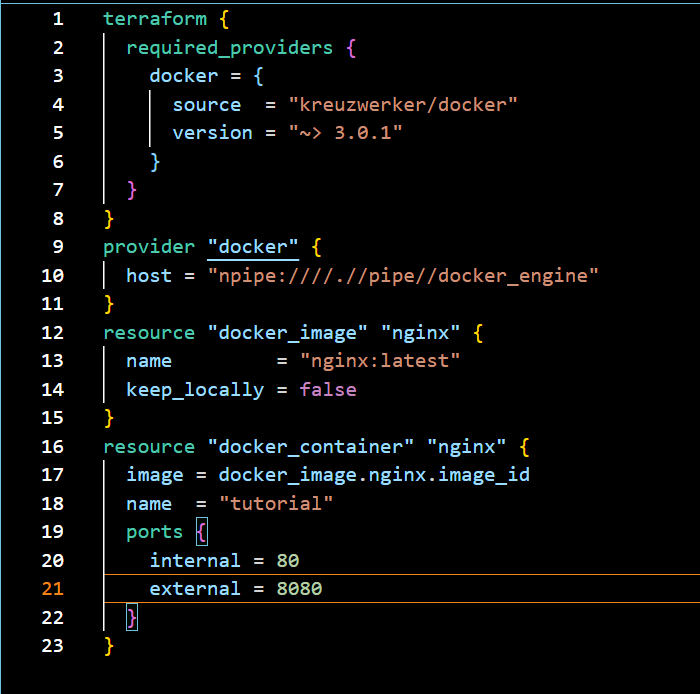
Step 6 : Type terraform apply to apply changes .

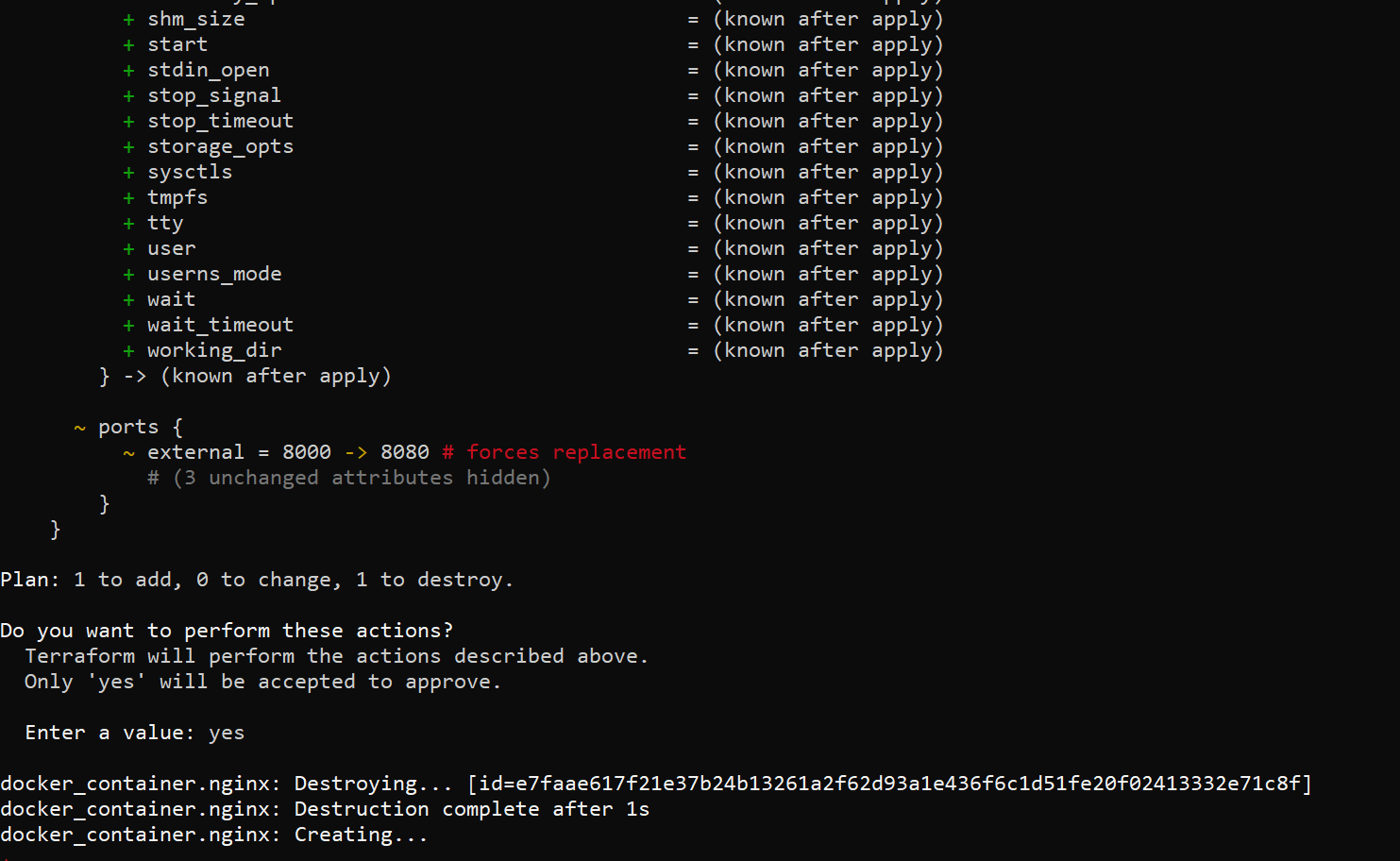


Step 7 : Docker container before and after step 6 execution

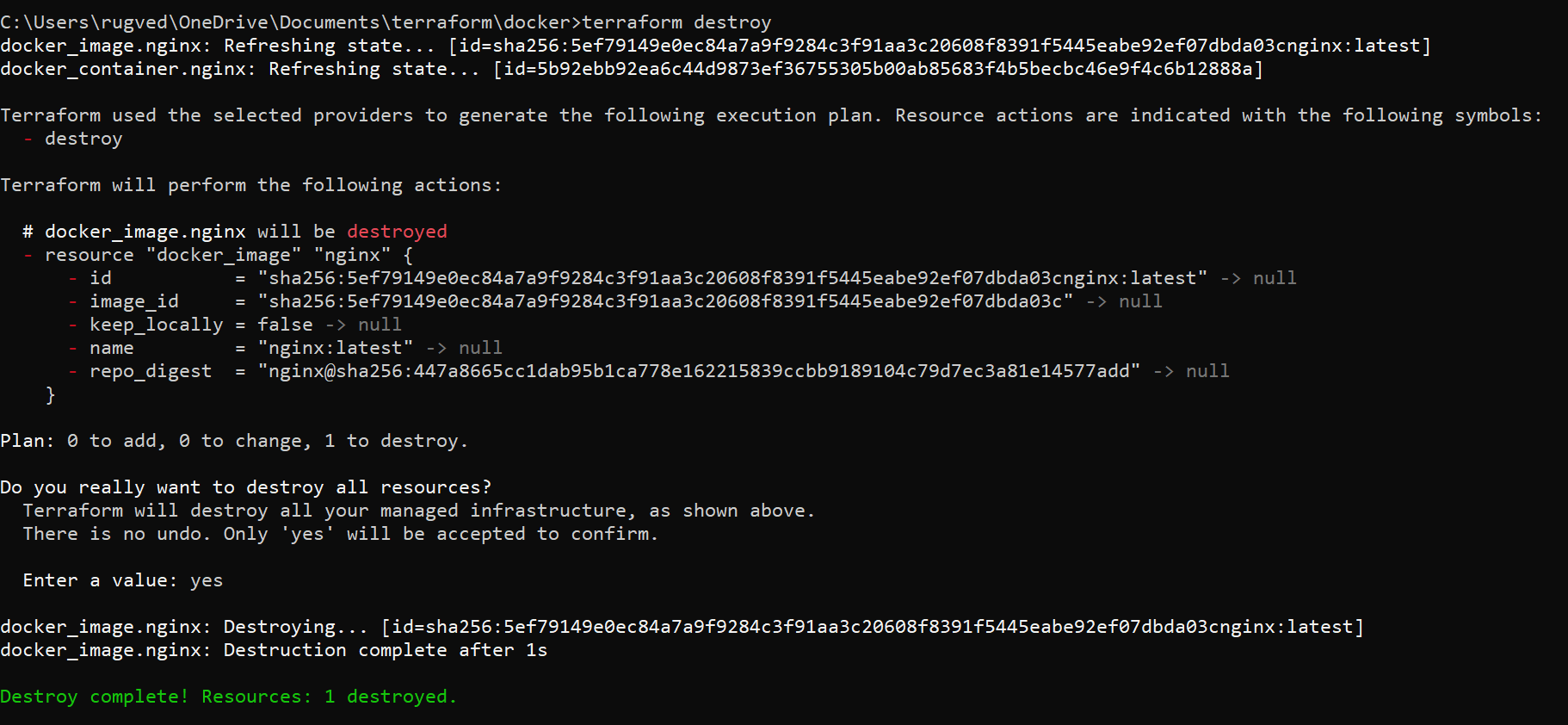


Step 8 (EXTRA ) : Execution of change .





Step 9 : terraform destroy to destroy infrastructure.



Step 10 : Docker after destroy command.

