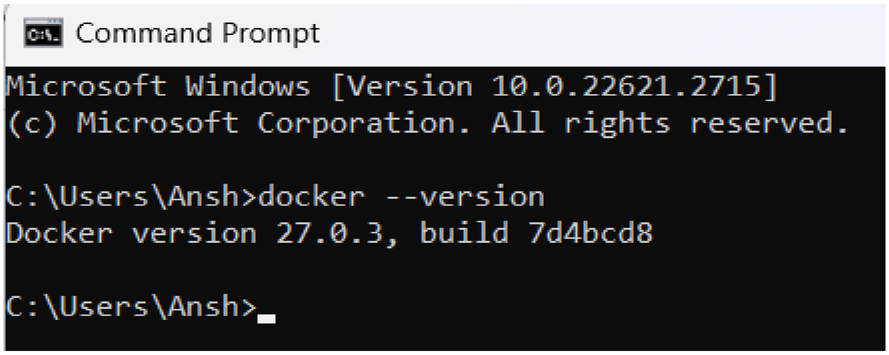
ADVANCE DEVOPS EXP 6

ANSH SARFARE **D15A/50**

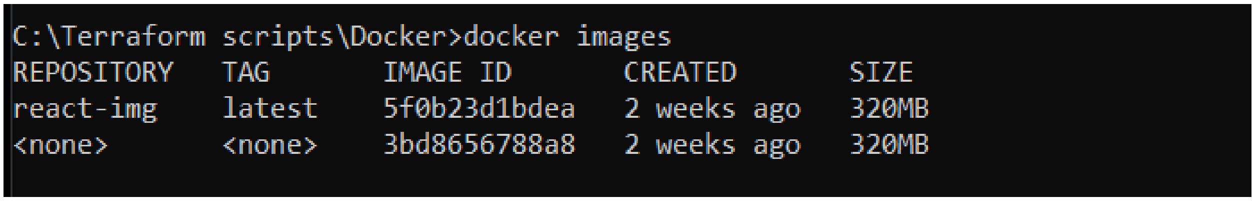
Aim:- To Build, change, and destroy AWS / GCP /Microsoft Azure/ DigitalOcean infrastructure Using Terraform. (S3 bucket or Docker) fdp

Docker Installation:

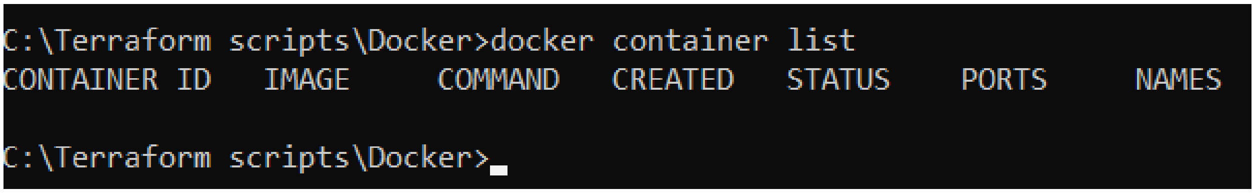


```
Microsoft Windows [Version 10.0.22621.2715]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Ansh>docker --version
Docker version 27.0.3, build 7d4bcd8

C:\Users\Ansh>
```

Before using terraform commands:

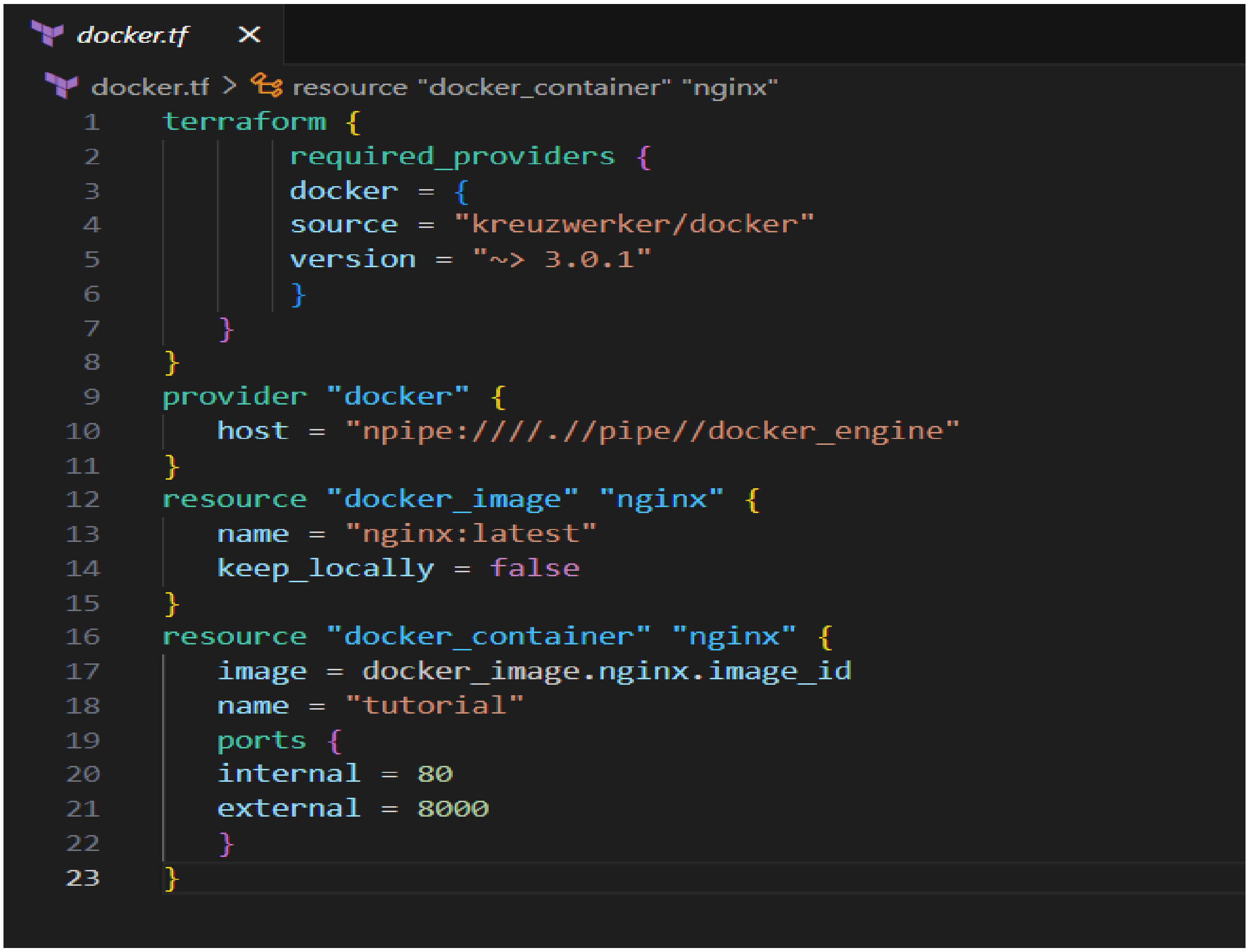


```
C:\Terraform scripts\Docker>docker images
REPOSITORY   TAG       IMAGE ID       CREATED        SIZE
react-img    latest    5f0b23d1bdea   2 weeks ago    320MB
<none>       <none>    3bd8656788a8   2 weeks ago    320MB
```



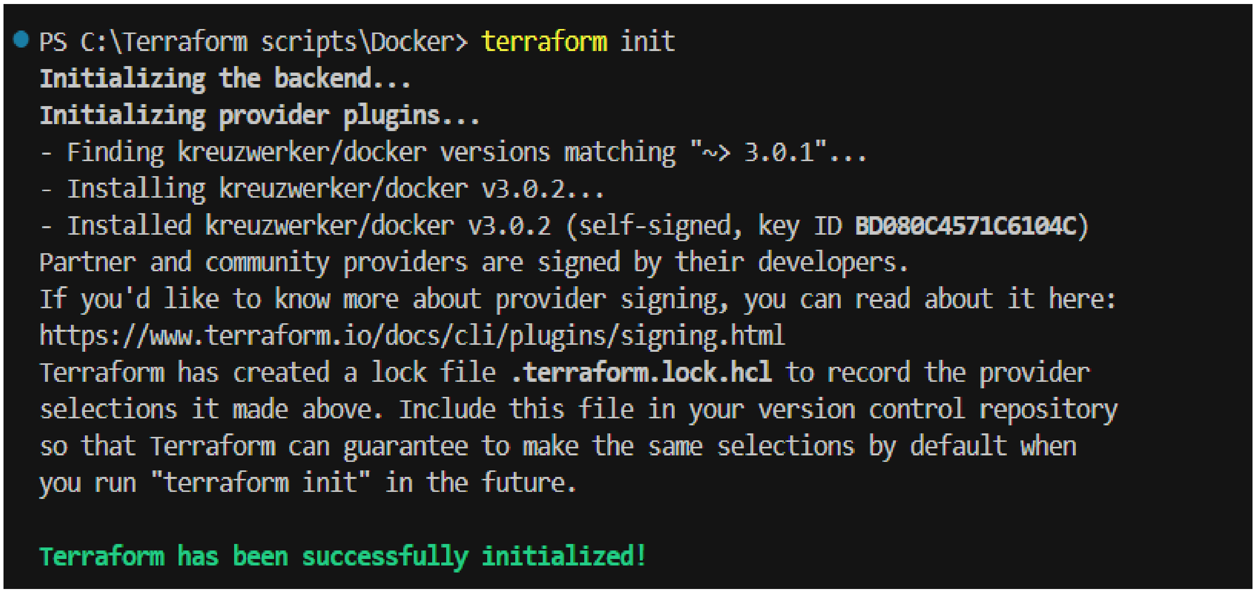
```
C:\Terraform scripts\Docker>docker container list
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES

C:\Terraform scripts\Docker>
```

Code:

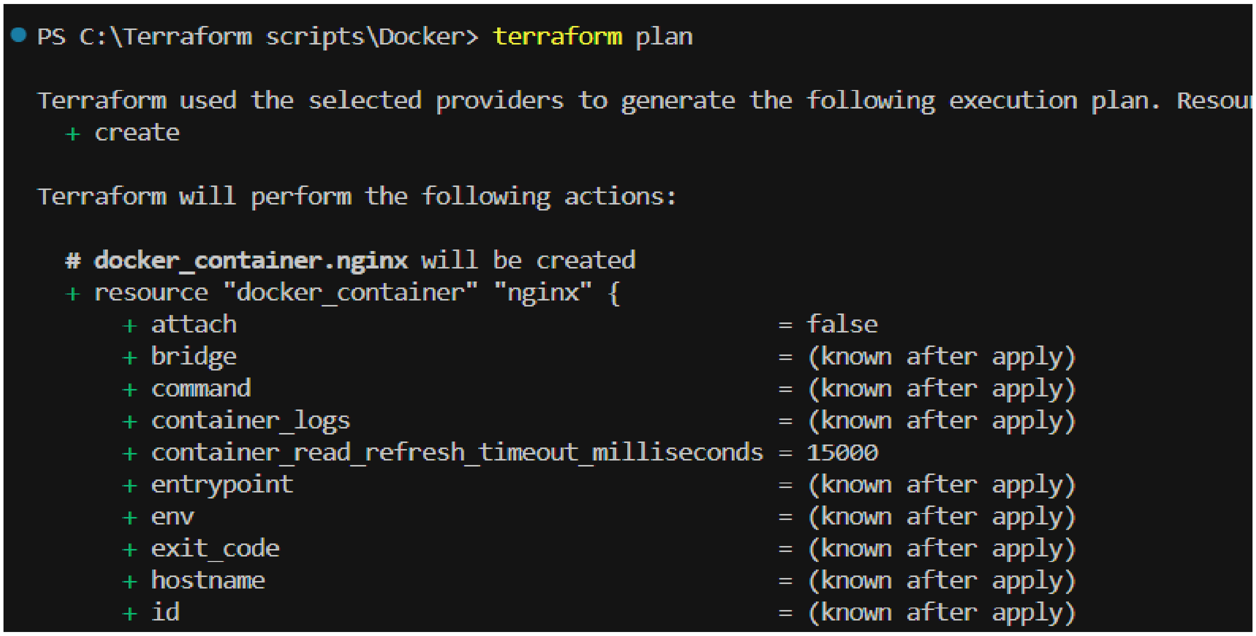


```
terraform {
       required_providers {
       docker = {
       source = "kreuzwerker/docker"
       version = "~> 3.0.1"
       }
   }
}
provider "docker" {
   host = "npipe:////.//pipe//docker_engine"
}
resource "docker_image" "nginx" {
   name = "nginx:latest"
   keep_locally = false
}
resource "docker_container" "nginx" {
   image = docker_image.nginx.image_id
   name = "tutorial"
   ports {
   internal = 80
   external = 8000
   }
}
```

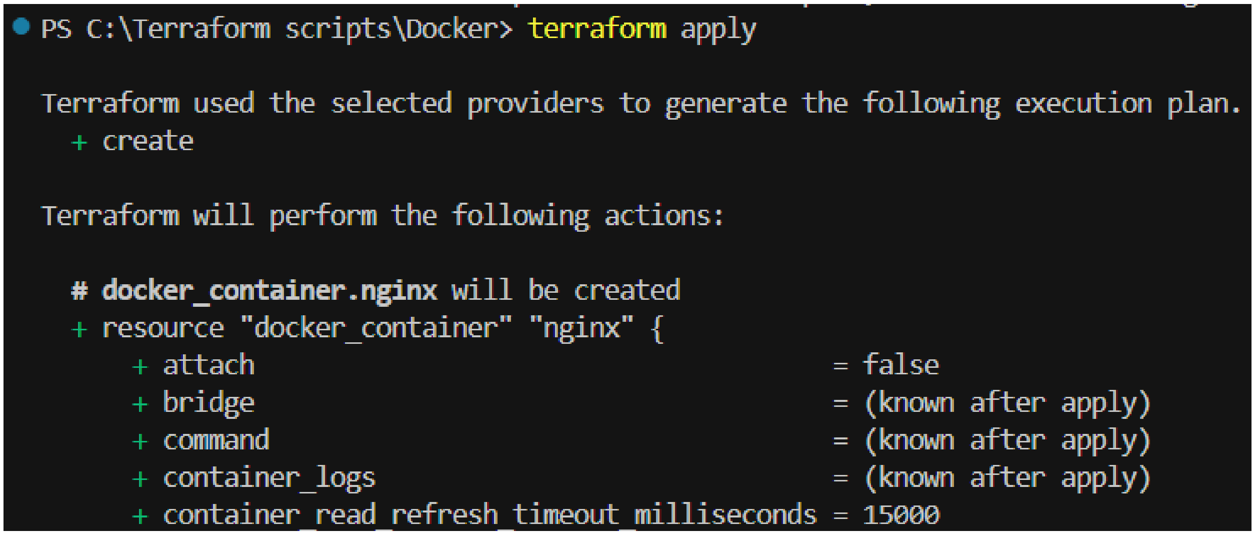
Terraform Commands:



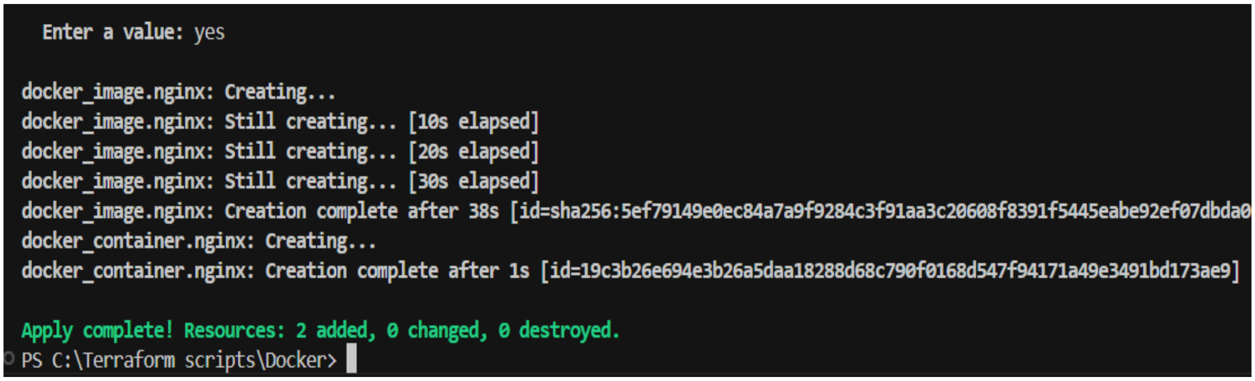
```
PS C:\Terraform scripts\Docker> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding kreuzwerker/docker versions matching "~> 3.0.1"...
- Installing kreuzwerker/docker v3.0.2...
- Installed kreuzwerker/docker v3.0.2 (self-signed, key ID BD080C4571C6104C)
Partner and community providers are signed by their developers.
If you'd like to know more about provider signing, you can read about it here:
https://www.terraform.io/docs/cli/plugins/signing.html
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!
```



```
PS C:\Terraform scripts\Docker> terraform plan

  Terraform used the selected providers to generate the following execution plan. Resou
    + create

  Terraform will perform the following actions:

    # docker_container.nginx will be created
    + resource "docker_container" "nginx" {
        + attach                                      = false
        + bridge                                      = (known after apply)
        + command                                     = (known after apply)
        + container_logs                              = (known after apply)
        + container_read_refresh_timeout_milliseconds = 15000
        + entrypoint                                  = (known after apply)
        + env                                         = (known after apply)
        + exit_code                                   = (known after apply)
        + hostname                                    = (known after apply)
        + id                                          = (known after apply)
```

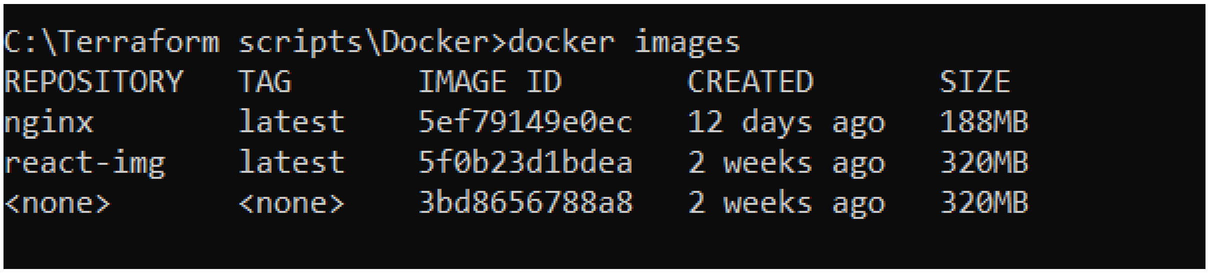


```
PS C:\Terraform scripts\Docker> terraform apply

  Terraform used the selected providers to generate the following execution plan.
    + create

  Terraform will perform the following actions:

    # docker_container.nginx will be created
    + resource "docker_container" "nginx" {
        + attach                                      = false
        + bridge                                      = (known after apply)
        + command                                     = (known after apply)
        + container_logs                              = (known after apply)
        + container_read_refresh_timeout_milliseconds = 15000
```

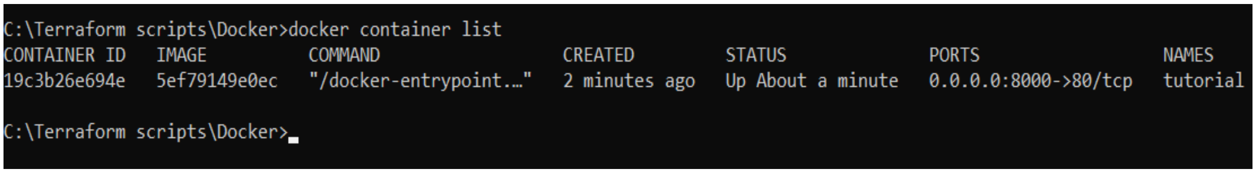


```
  Enter a value: yes

docker_image.nginx: Creating...
docker_image.nginx: Still creating... [10s elapsed]
docker_image.nginx: Still creating... [20s elapsed]
docker_image.nginx: Still creating... [30s elapsed]
docker_image.nginx: Creation complete after 38s [id=sha256:5ef79149e0ec84a7a9f9284c3f91aa3c20608f8391f5445eabe92ef07dbda0
docker_container.nginx: Creating...
docker_container.nginx: Creation complete after 1s [id=19c3b26e694e3b26a5daa18288d68c790f0168d547f94171a49e3491bd173ae9]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
PS C:\Terraform scripts\Docker>
```

After using Terraform commands:

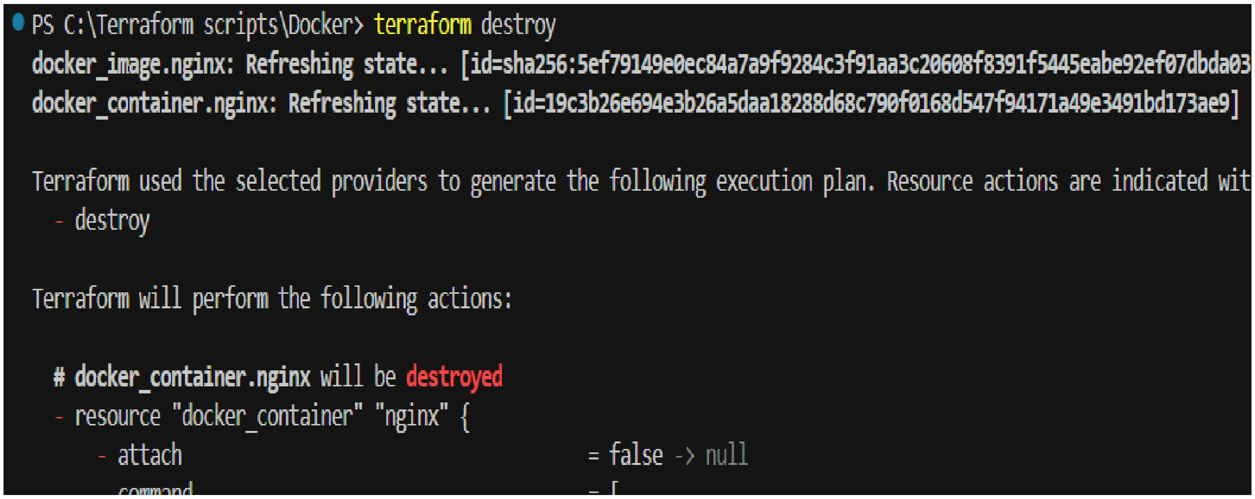


```
C:\Terraform scripts\Docker>docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
nginx         latest    5ef79149e0ec   12 days ago    188MB
react-img     latest    5f0b23d1bdea   2 weeks ago    320MB
<none>        <none>    3bd8656788a8   2 weeks ago    320MB
```



```
C:\Terraform scripts\Docker>docker container list
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS             PORTS                  NAMES
19c3b26e694e   5ef79149e0ec   "/docker-entrypoint.…"   2 minutes ago   Up About a minute   0.0.0.0:8000->80/tcp   tutorial

C:\Terraform scripts\Docker>
```

To Delete the Containers created:



```
PS C:\Terraform scripts\Docker> terraform destroy
docker_image.nginx: Refreshing state... [id=sha256:5ef79149e0ec84a7a9f9284c3f91aa3c20608f8391f5445eabe92ef07dbda03
docker_container.nginx: Refreshing state... [id=19c3b26e694e3b26a5daa18288d68c790f0168d547f94171a49e3491bd173ae9]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated wit
  - destroy

Terraform will perform the following actions:

  # docker_container.nginx will be destroyed
  - resource "docker_container" "nginx" {
      - attach                       = false -> null
        command                      = [
```