Name : Pratik Patil **class :D15C** **Roll No:40**

Experiment No.: 6

Implementation:
A. Creating docker image using terraform
Prerequisite:
1) Download and Install Docker Desktop from https://www.docker.com/
Step 1: Check the docker functionality
Step 3: Execute Terraform Init command to initialize the resources
Step 4: Execute Terraform plan to see the available resources

```
C:\Users\91799\Desktop\TerraformScripts\Docker>terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of kreuzwerker/docker from the dependency lock fi
le
- Using previously-installed kreuzwerker/docker v2.21.0

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to se
e
any changes that are required for your infrastructure. All Terraform command
s
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, ot
her
commands will detect it and remind you to do so if necessary.
```

```
C:\Users\91799\Desktop\TerraformScripts\Docker>terraform plan
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e1
37cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]

Terraform used the selected providers to generate the following execution
plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # docker_container.foo will be created
  + resource "docker_container" "foo" {
      + attach           = false
      + bridge           = (known after apply)
      + command          = [
          + "/bin/bash",
          + "-c",
```

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”

```
C:\Users\91799\Desktop\TerraformScripts\Docker>terraform apply
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e1
37cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]

Terraform used the selected providers to generate the following execution
plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # docker_container.foo will be created
  + resource "docker_container" "foo" {
      + attach           = false
      + bridge           = (known after apply)
      + command          = [
          + "/bin/bash",
          + "-c",
          + "while true; do sleep 1000; done",
        ]
      + container_logs   = (known after apply)
      + entrypoint       = (known after apply)
```

Docker images, Before Executing Apply step:

```
C:\Users\91799\Desktop\TerraformScripts\Docker>docker images
REPOSITORY                  TAG       IMAGE ID       CREATED        SIZE
hello-img                   latest    d94a4f16d8e9   9 days ago     133MB
ubuntu                      latest    edbfe74c41f8   2 weeks ago    78.1MB
codelabz-app                latest    88655427baae   7 months ago   2.91GB
spine3/firebase-emulator    latest    95b479b53b4b   2 years ago    671MB
```

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

```
C:\Users\91799\Desktop\TerraformScripts\Docker>terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e1
37cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Refreshing state... [id=fd5508a62f7ac30c0f23250b4887a8
a9891de512c4be26026f937054dbf71c13]

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

Terraform will perform the following actions:

  # docker_container.foo will be destroyed
  - resource "docker_container" "foo" {
      - attach            = false -> null
      - command           = [
          - "/bin/bash",
          - "-c",
          - "while true; do sleep 1000; done",
        ] -> null
```

Docker images After Executing Destroy step

```
Destroy complete! Resources: 2 destroyed.

C:\Users\91799\Desktop\TerraformScripts\Docker>docker images
REPOSITORY                  TAG       IMAGE ID       CREATED        SIZE
hello-img                   latest    d94a4f16d8e9   9 days ago     133MB
codelabz-app                latest    88655427baae   7 months ago   2.91GB
spine3/firebase-emulator    latest    95b479b53b4b   2 years ago    671MB

C:\Users\91799\Desktop\TerraformScripts\Docker>
```