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

```
Microsoft Windows [Version 10.0.22621.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\athar>docker --version
Docker version 27.1.1, build 6312585
```

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 Atom editor and write the followingcontents into it to create a Ubuntu Linux container.
Script:

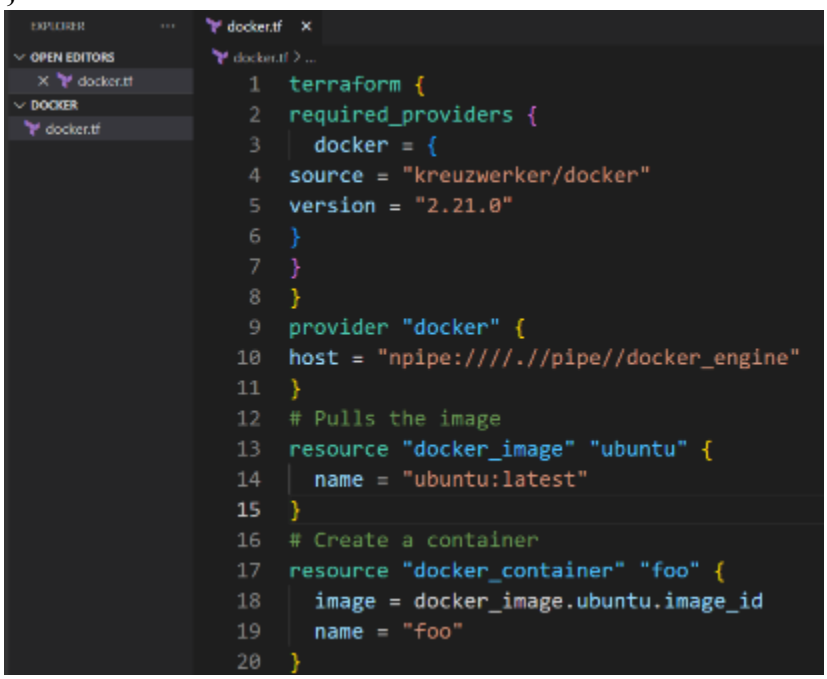
```
terraform
  { required_providers
  {docker = {
     source = "kreuzwerker/docker"
     version = "2.21.0"
    }
  }
}

provider "docker" {
  host = "npipe:////.//pipe//docker_engine"
}

# Pulls the image
resource "docker_image" "ubuntu"
```

```
  {name = "ubuntu:latest"
}

# Create a container
resource "docker_container" "foo"
  { image =
 docker_image.ubuntu.image_idname =
 "foo"
}
```



Step 3: Execute Terraform Init command to initialize the resources

```
Initializing the backend...
Initializing provider plugins...
- Finding kreuzwerker/docker versions matching "2.21.0"...
- Installing kreuzwerker/docker v2.21.0...
- Installed kreuzwerker/docker v2.21.0 (self-signed, key ID BD080C4571C
Partner and community providers are signed by their developers.
If you'd like to know more about provider signing, you can read about i
https://www.terraform.io/docs/cli/plugins/signing.html
Terraform has created a lock file .terraform.lock.hcl to record the pro
selections it made above. Include this file in your version control rep
so that Terraform can guarantee to make the same selections by default
you run "terraform init" in the future.

Terraform has been successfully initialized!

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

If you ever set or change modules or backend configuration for Terrafor
rerun this command to reinitialize your working directory. If you forge
commands will detect it and remind you to do so if necessary.
PS D:\all code\Terraform Scripts\Docker>
```

Step 4: Execute Terraform plan to see the available resources

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

Terraform will perform the following actions:

  # docker_container.foo will be created
  + resource "docker_container" "foo" {
      + attach           = false
      + bridge           = (known after apply)
      + command          = (known after apply)
      + container_logs   = (known after apply)
      + entrypoint       = (known after apply)
      + env              = (known after apply)
      + exit_code        = (known after apply)
      + gateway          = (known after apply)
      + hostname         = (known after apply)
      + id               = (known after apply)
      + image            = (known after apply)
      + init             = (known after apply)
      + ip_address       = (known after apply)
      + ip_prefix_length = (known after apply)
      + ipc_mode         = (known after apply)
      + log_driver       = (known after apply)
      + logs             = false
      + must_run         = true
      + name             = "foo"
      + network_data     = (known after apply)
      + read_only        = false
      + remove_volumes   = true
      + restart          = "no"
      + rm               = false
      + runtime          = (known after apply)
      + security_opts    = (known after apply)
      + shm_size         = (known after apply)
```

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**"

```
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          = (known after apply)
      + container_logs   = (known after apply)
      + entrypoint       = (known after apply)
      + env              = (known after apply)
      + exit_code        = (known after apply)
      + gateway          = (known after apply)
      + hostname         = (known after apply)
      + id               = (known after apply)
      + image            = (known after apply)
      + init             = (known after apply)
      + ip_address       = (known after apply)
      + ip_prefix_length = (known after apply)
      + ipc_mode         = (known after apply)
      + log_driver       = (known after apply)
      + logs             = false
      + must_run         = true
      + name             = "foo"
      + network_data     = (known after apply)
      + read_only        = false
      + remove_volumes   = true
      + restart          = "no"
      + rm               = false
      + runtime          = (known after apply)
```

Docker images,

Before Executing Apply step:

```
C:\Users\Ayush Maurya>docker images
REPOSITORY    TAG       IMAGE ID       CREATED       SIZE
react-img     latest    d8b8903ee063   8 days ago    320MB
```

Docker images, After Executing Apply step:

```
Windows PowerShell
PS E:\Terraform Script\Docker> docker images
REPOSITORY                    TAG       IMAGE ID       CREATED          SIZE
docker101tutorial             latest    e5bc7d28c9a0   30 minutes ago   28.9MB
manavjawrani/docker101tutorial latest   e5bc7d28c9a0   30 minutes ago   28.9MB
ubuntu                        latest    2dc39ba059dc   2 weeks ago      77.8MB
alpine/git                    latest    692618a0d74d   2 weeks ago      43.4MB
PS E:\Terraform Script\Docker>
```

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

```
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce962a137cf28ea04dd03e6df8c9d66b19b6ad761c2598eubuntu:latest]
docker_container.foo: Refreshing state... [id=6017524175e17094f75020bb6485b3e4cbbb985037e3a3d302301c4a075a39509b]

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           = [
          - "sh",
          - "-c",
          - "while true; do sleep 1; cons",
        ] -> null
      - cpu_shares        = 0 -> null
      - dns               = [] -> null
      - dns_opts          = [] -> null
      - dns_search        = [] -> null
      - entrypoint        = [] -> null
      - env               = [] -> null
      - gateway           = "172.17.0.1" -> null
      - group_add         = [] -> null
      - hostname          = "6017524175e17" -> null
      - id                = "6017524175e17094f75020bb6485b3e4cbbb985037e3a3d302301c4a075a39509b" -> null
      - image             = "sha256:edbfe74c41f8a3501ce962a137cf28ea04dd03e6df8c9d66b19b6ad761c2598a" -> null
      - init              = false -> null
      - ip_address        = "172.17.0.2" -> null
      - ip_prefix_length  = 16 -> null
      - ipc_mode          = "private" -> null
      - links             = [] -> null
      - log_driver        = "json-file" -> null
      - log_opts          = {} -> null
      - logs              = false -> null
      - max_retry_count   = 0 -> null
      - memory            = 0 -> null
      - memory_swap       = 0 -> null
      - must_run          = true -> null
```

Docker images After Executing Destroy step

```
PS C:\Users\Ayush Maurya\Desktop\TerraformScript\Docker> docker images
REPOSITORY    TAG       IMAGE ID       CREATED       SIZE
react-img     latest    d8b8903ee063   8 days ago    320MB
PS C:\Users\Ayush Maurya\Desktop\TerraformScript\Docker>
```