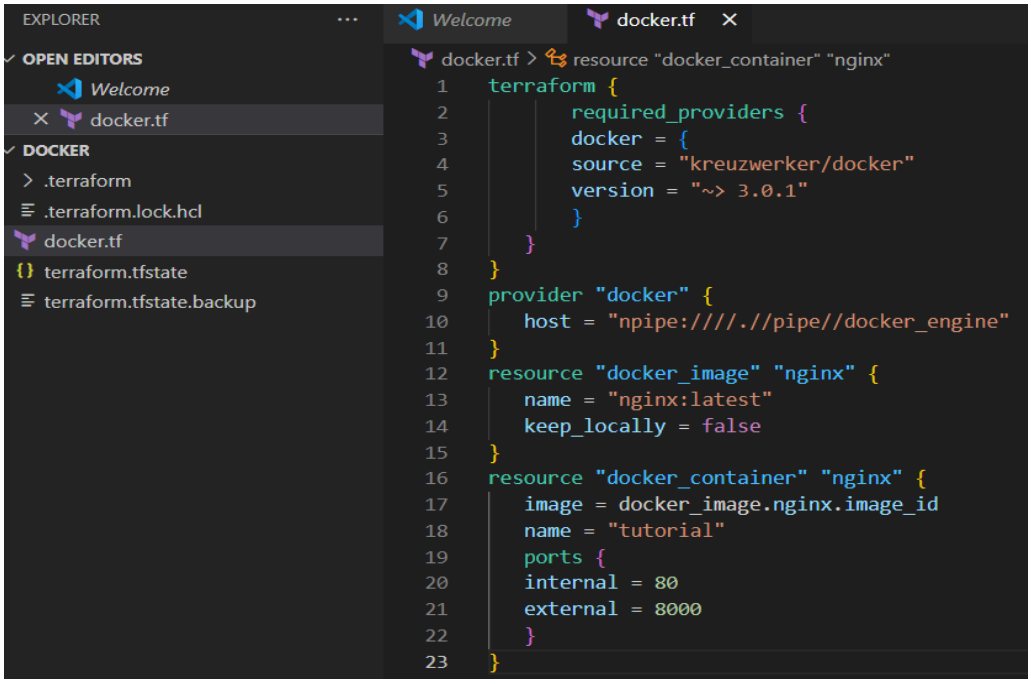
Siddhant Sathe **D15A/51**

Aim: Exp 6 To Build, change, and destroy AWS / GCP /Microsoft Azure/ DigitalOcean infrastructure Using Terraform. (S3 bucket or Docker) fdp

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
  }
}
```



Output:

```
C:\Users\sathe>docker --version
Docker version 26.1.4, build 5650f9b
```

Docker Images before using terraform commands:

```
D:\Siddhant\Terraform Scripts\Docker>docker images
REPOSITORY                            TAG         IMAGE ID       CREATED         SIZE
mindsdb/mindsdb-docker-extension      1.0.8       17f9318c547a   2 months ago    8.01MB
mindsdb/mindsdb                       latest      d33051c6962a   2 months ago    1.64GB
mindsdb/mindsdb                       v24.6.4.1   d33051c6962a   2 months ago    1.64GB
docker/welcome-to-docker              latest      c1f619b6477e   9 months ago    18.6MB
hello-world                           latest      d2c94e258dcb   16 months ago   13.3kB
ngrok/ngrok                           latest      2932f7e14783   54 years ago    144MB
```

Terraform init:

```
PS D:\Siddhant\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!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
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, other
commands will detect it and remind you to do so if necessary.
```

Terraform plan:

```
PS D:\Siddhant\Terraform Scripts\Docker> terraform plan

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.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)
      + image                                       = (known after apply)
      + init                                        = (known after apply)
      + ipc_mode                                    = (known after apply)
      + log_driver                                  = (known after apply)
      + logs                                        = false
      + must_run                                    = true
      + name                                        = "tutorial"
      + 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)
      + start                                       = true
      + stdin_open                                  = false
      + stop_signal                                 = (known after apply)
      + stop_timeout                                = (known after apply)
      + tty                                         = false
      + wait                                        = false
      + wait_timeout                                = 60
      + ports {
          + external = 8000
          + internal = 80
          + ip       = "0.0.0.0"
          + protocol = "tcp"
        }
    }

  # docker_image.nginx will be created
  + resource "docker_image" "nginx" {
      + id           = (known after apply)
      + image_id     = (known after apply)
      + keep_locally = false
      + name         = "nginx:latest"
      + repo_digest  = (known after apply)
    }

Plan: 2 to add, 0 to change, 0 to destroy.
```

Terraform apply:

```
PS D:\Siddhant\Terraform Scripts\Docker> 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.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)
```

```
Plan: 2 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  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: Creation complete after 29s [id=sha256:5ef79149e0ec84a7a9f9284c3f91aa3c20608f8391f5445eabe92ef07dbda03cnginx:latest]
docker_container.nginx: Creating...
docker_container.nginx: Creation complete after 5s [id=21a0a0573bc9ec62f30a6ee5924e7706bbdddc14dcc4cd934dd146b084dbe38c]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
```

Docker Images after using terraform apply:

```
D:\Siddhant\Terraform Scripts\Docker>docker images
REPOSITORY                          TAG         IMAGE ID       CREATED         SIZE
nginx                               latest      5ef79149e0ec   10 days ago     188MB
mindsdb/mindsdb-docker-extension    1.0.8       17f9318c547a   2 months ago    8.01MB
mindsdb/mindsdb                     latest      d33051c6962a   2 months ago    1.64GB
mindsdb/mindsdb                     v24.6.4.1   d33051c6962a   2 months ago    1.64GB
docker/welcome-to-docker            latest      c1f619b6477e   9 months ago    18.6MB
hello-world                         latest      d2c94e258dcb   16 months ago   13.3kB
ngrok/ngrok                         latest      2932f7e14783   54 years ago    144MB
```

Terraform destroy:

```
PS D:\Siddhant\Terraform Scripts\Docker> terraform destroy
docker_image.nginx: Refreshing state... [id=sha256:5ef79149e0ec84a7a9f9284c3f91aa3c20608f8391f5445eabe92ef07dbda03cnginx:latest]
docker_container.nginx: Refreshing state... [id=21a0a0573bc9ec62f30a6ee5924e7706bbdddc14dcc4cd934dd146b084dbe38c]

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.nginx will be destroyed
  - resource "docker_container" "nginx" {
      - attach                                      = false -> null
      - command                                     = [
          - "nginx",
          - "-g",
          - "daemon off;",
        ] -> null
      - container_read_refresh_timeout_milliseconds = 15000 -> null
      - cpu_shares                                  = 0 -> null
      - dns                                         = [] -> null
      - dns_opts                                    = [] -> null
      - dns_search                                  = [] -> null
      - entrypoint                                  = [
          - "/docker-entrypoint.sh",
        ] -> null
      - env                                         = [] -> null
      - group_add                                   = [] -> null
      - hostname                                    = "21a0a0573bc9" -> null
      - id                                          = "21a0a0573bc9ec62f30a6ee5924e7706bbdddc14dcc4cd934dd146b084dbe38c" -> null
```

```
  # docker_image.nginx will be destroyed
  - resource "docker_image" "nginx" {
      - id           = "sha256:5ef79149e0ec84a7a9f9284c3f91aa3c20608f8391f5445eabe92ef07dbda03cnginx:latest" -> null
      - image_id     = "sha256:5ef79149e0ec84a7a9f9284c3f91aa3c20608f8391f5445eabe92ef07dbda03c" -> null
      - keep_locally = false -> null
      - name         = "nginx:latest" -> null
      - repo_digest  = "nginx@sha256:447a8665cc1dab95b1ca778e162215839ccbb9189104c79d7ec3a81e14577add" -> null
    }

Plan: 0 to add, 0 to change, 2 to destroy.

Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

docker_container.nginx: Destroying... [id=21a0a0573bc9ec62f30a6ee5924e7706bbdddc14dcc4cd934dd146b084dbe38c]
docker_container.nginx: Destruction complete after 1s
docker_image.nginx: Destroying... [id=sha256:5ef79149e0ec84a7a9f9284c3f91aa3c20608f8391f5445eabe92ef07dbda03cnginx:latest]
docker_image.nginx: Destruction complete after 0s

Destroy complete! Resources: 2 destroyed.
```

Docker Images after using terraform destroy:

```
D:\Siddhant\Terraform Scripts\Docker>docker images
REPOSITORY                          TAG         IMAGE ID       CREATED         SIZE
mindsdb/mindsdb-docker-extension    1.0.8       17f9318c547a   2 months ago    8.01MB
mindsdb/mindsdb                     latest      d33051c6962a   2 months ago    1.64GB
mindsdb/mindsdb                     v24.6.4.1   d33051c6962a   2 months ago    1.64GB
docker/welcome-to-docker            latest      c1f619b6477e   9 months ago    18.6MB
hello-world                         latest      d2c94e258dcb   16 months ago   13.3kB
ngrok/ngrok                         latest      2932f7e14783   54 years ago    144MB
```