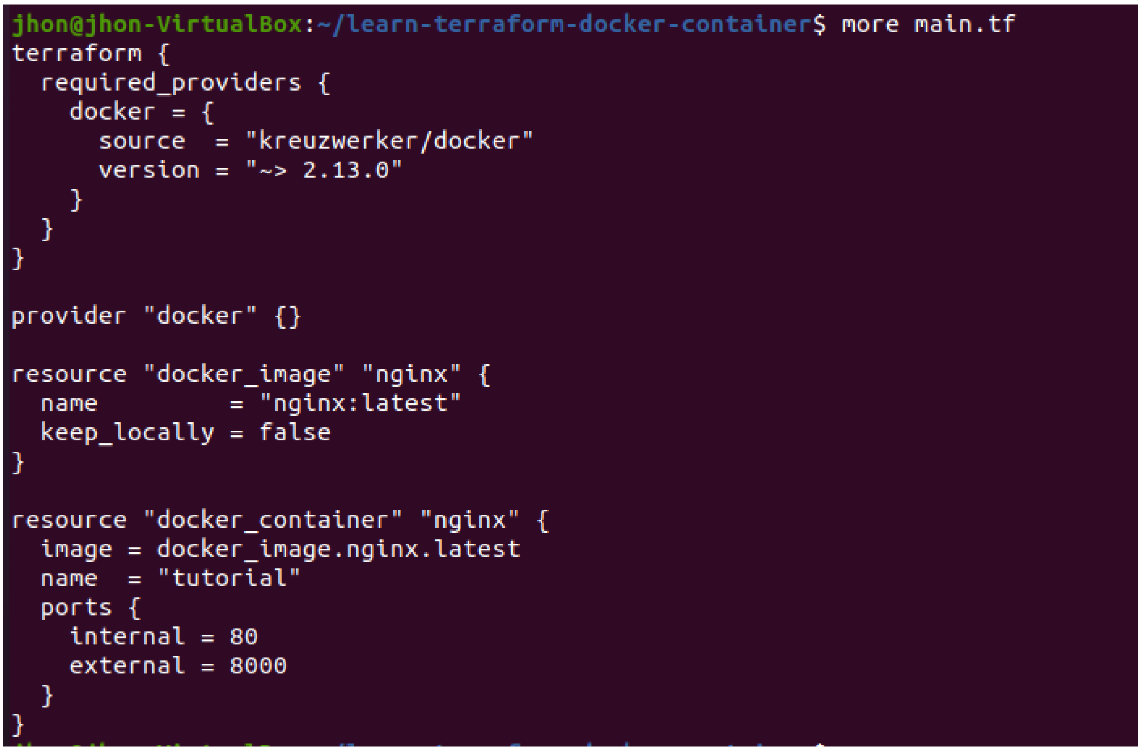
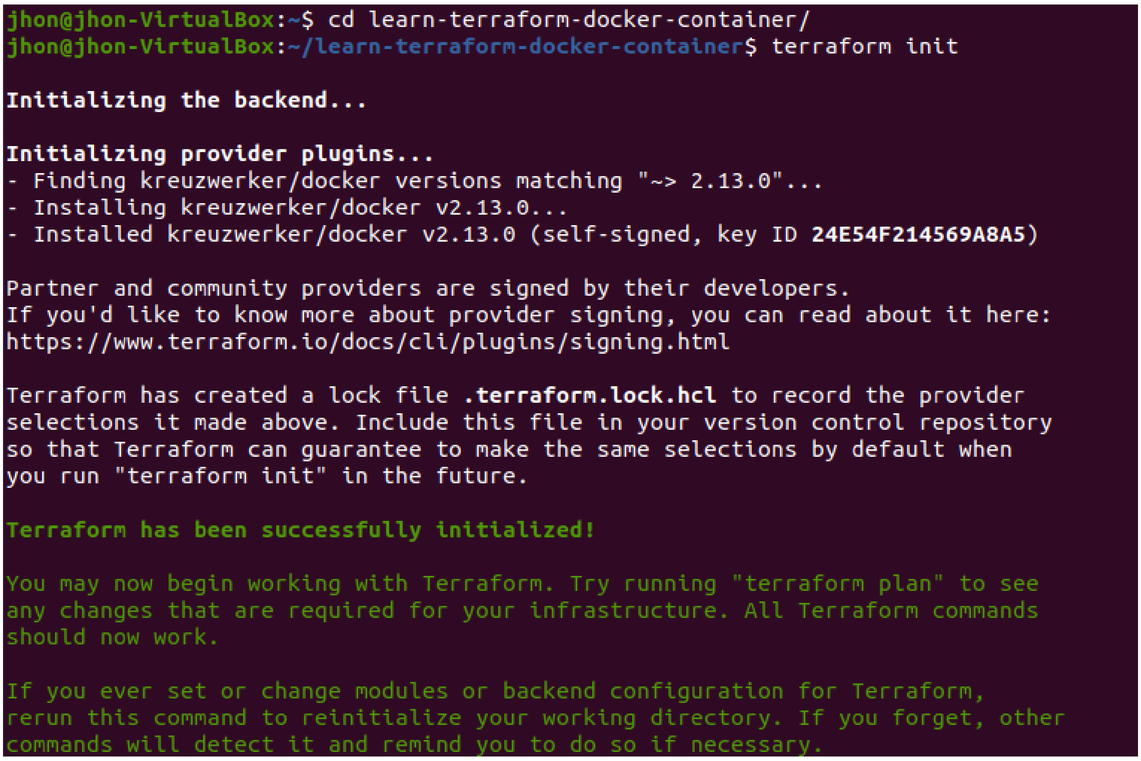
# INIT FILE



```
jhon@jhon-VirtualBox:~/learn-terraform-docker-container$ more main.tf
terraform {
  required_providers {
    docker = {
      source  = "kreuzwerker/docker"
      version = "~> 2.13.0"
    }
  }
}

provider "docker" {}

resource "docker_image" "nginx" {
  name         = "nginx:latest"
  keep_locally = false
}

resource "docker_container" "nginx" {
  image = docker_image.nginx.latest
  name  = "tutorial"
  ports {
    internal = 80
    external = 8000
  }
}
```

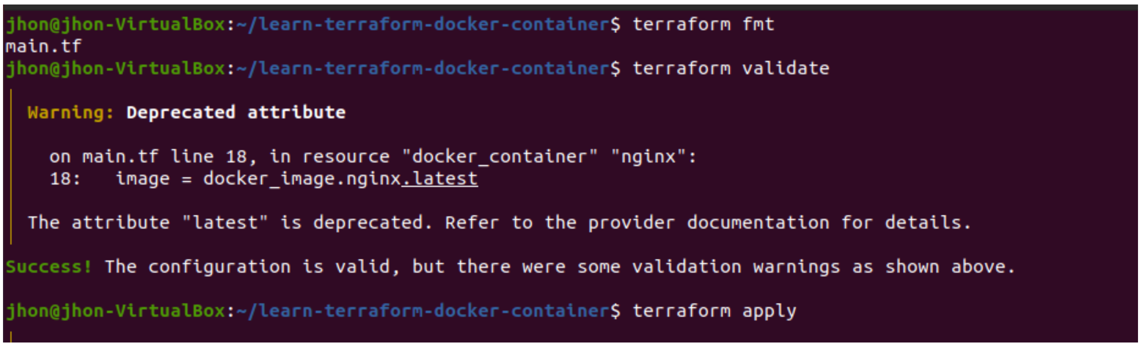
# Terraform init



```
jhon@jhon-VirtualBox:~$ cd learn-terraform-docker-container/
jhon@jhon-VirtualBox:~/learn-terraform-docker-container$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding kreuzwerker/docker versions matching "~> 2.13.0"...
- Installing kreuzwerker/docker v2.13.0...
- Installed kreuzwerker/docker v2.13.0 (self-signed, key ID 24E54F214569A8A5)

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.
```

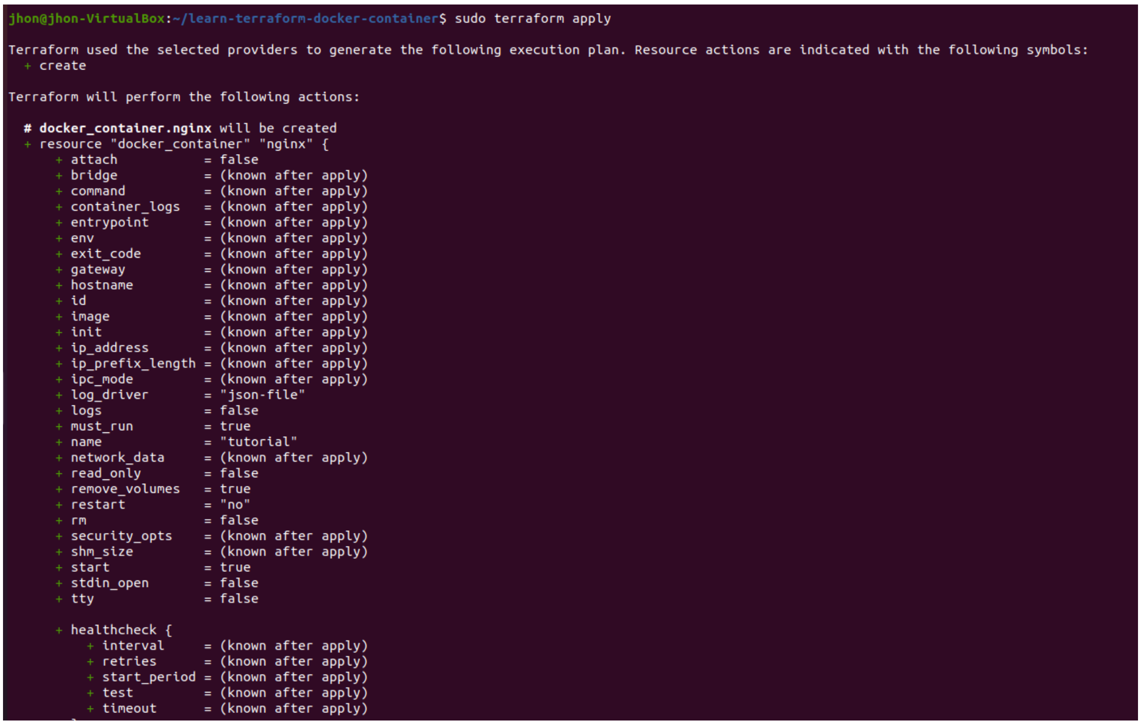
# Terraform fmt

# Terraform validate

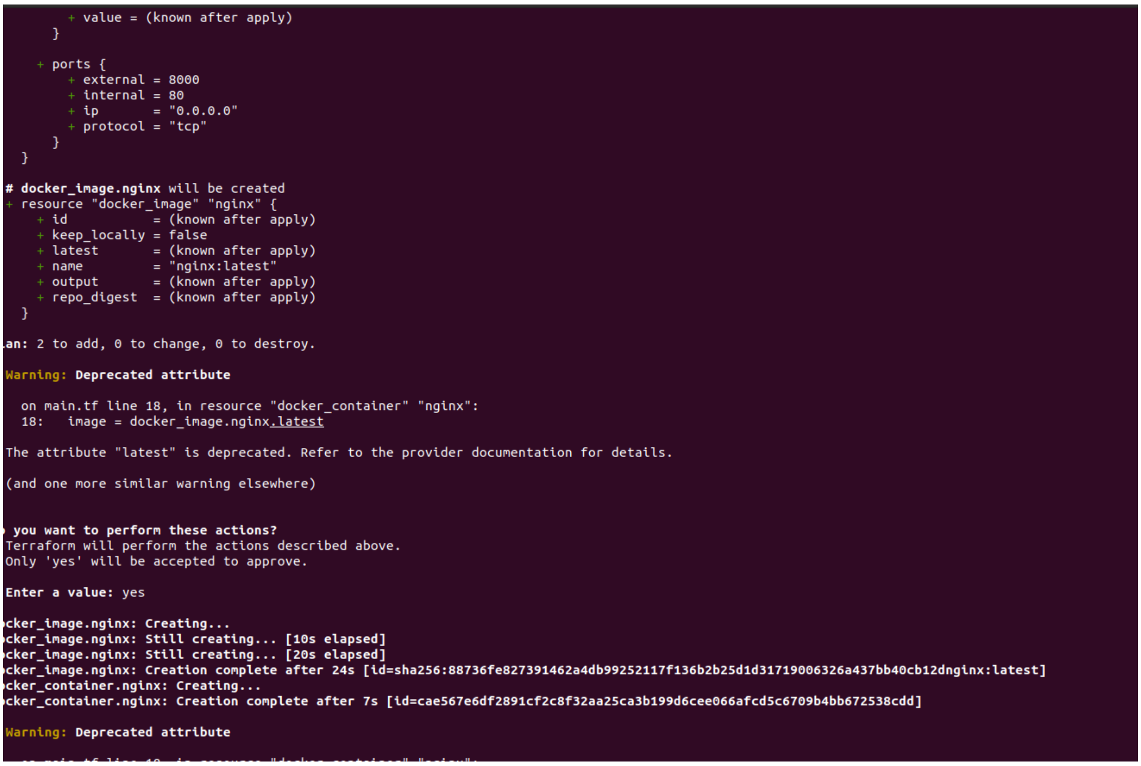


```
jhon@jhon-VirtualBox:~/learn-terraform-docker-container$ terraform fmt
main.tf
jhon@jhon-VirtualBox:~/learn-terraform-docker-container$ terraform validate
│
│ Warning: Deprecated attribute
│
│   on main.tf line 18, in resource "docker_container" "nginx":
│   18:   image = docker_image.nginx.latest
│
│ The attribute "latest" is deprecated. Refer to the provider documentation for details.
╵
Success! The configuration is valid, but there were some validation warnings as shown above.

jhon@jhon-VirtualBox:~/learn-terraform-docker-container$ terraform apply
```

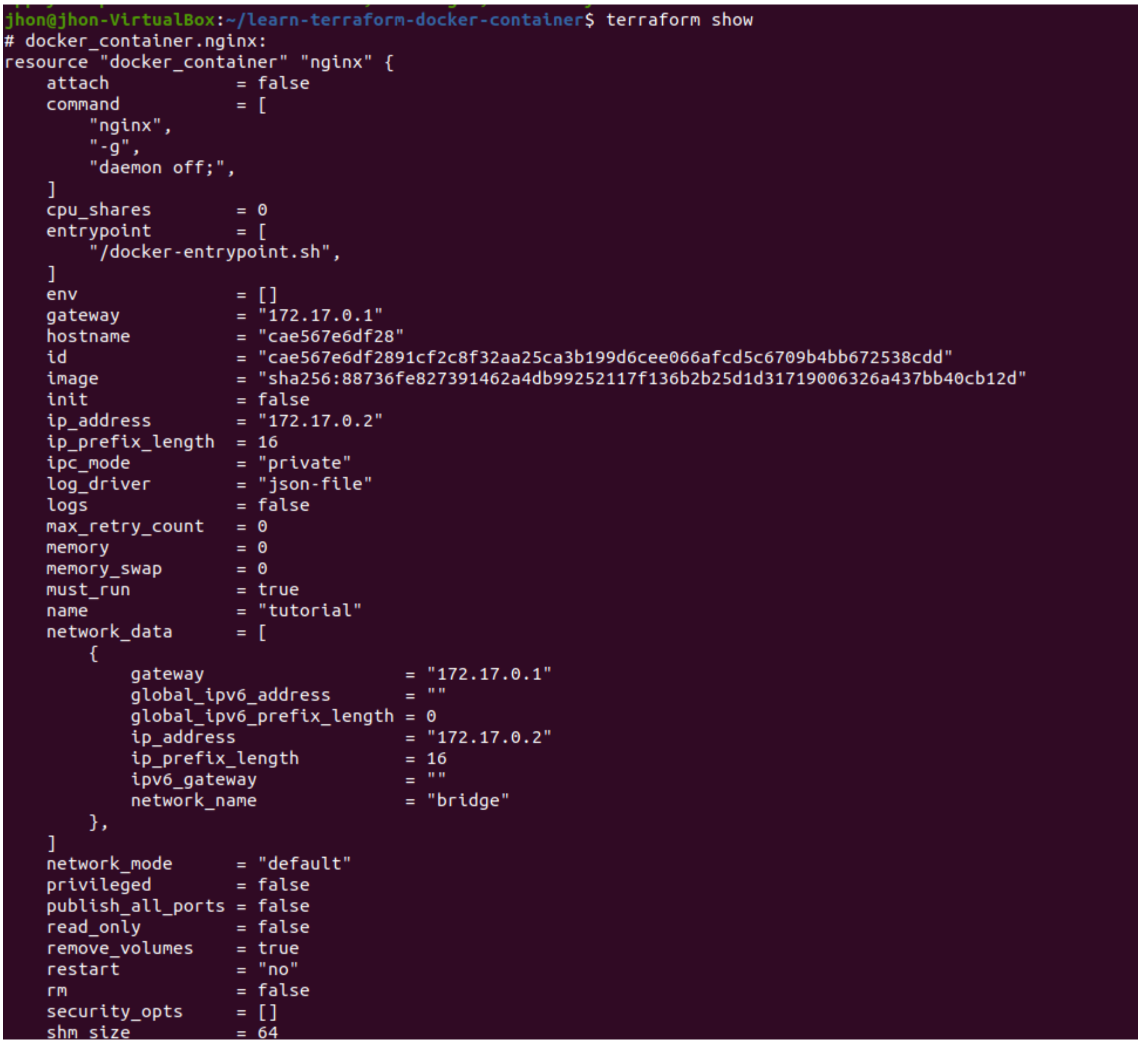
# Terraform aplly



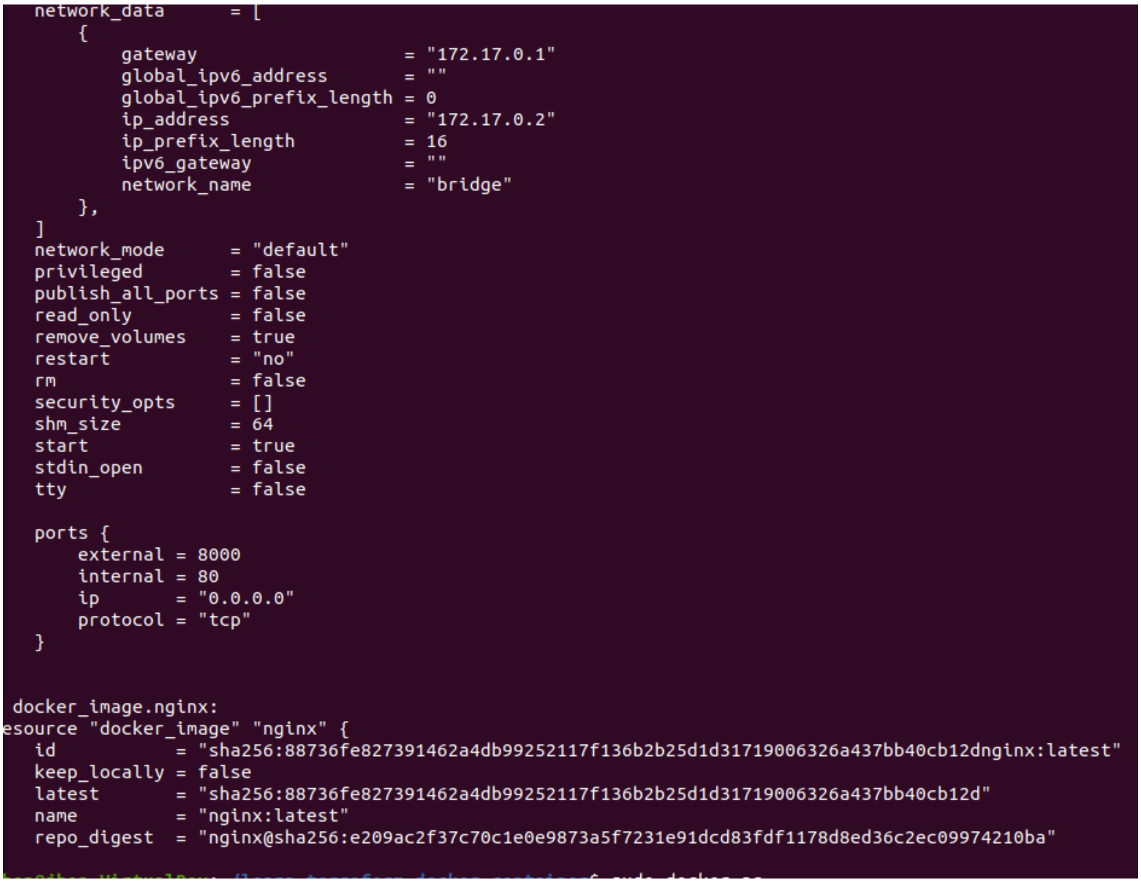
```
jhon@jhon-VirtualBox:~/learn-terraform-docker-container$ sudo terraform apply

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)
      + 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        = "json-file"
      + logs              = false
      + must_run          = true
      + name              = "tutorial"
      + network_data      = (known after apply)
      + read_only         = false
      + remove_volumes    = true
      + restart           = "no"
      + rm                = false
      + security_opts     = (known after apply)
      + shm_size          = (known after apply)
      + start             = true
      + stdin_open        = false
      + tty               = false

      + healthcheck {
          + interval     = (known after apply)
          + retries      = (known after apply)
          + start_period = (known after apply)
          + test         = (known after apply)
          + timeout      = (known after apply)
```



```
          + value = (known after apply)
        }

      + 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)
    + keep_locally = false
    + latest       = (known after apply)
    + name         = "nginx:latest"
    + output       = (known after apply)
    + repo_digest  = (known after apply)
  }

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

Warning: Deprecated attribute

  on main.tf line 18, in resource "docker_container" "nginx":
  18:   image = docker_image.nginx.latest

The attribute "latest" is deprecated. Refer to the provider documentation for details.

(and one more similar warning elsewhere)


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

Enter a value: yes

cker_image.nginx: Creating...
cker_image.nginx: Still creating... [10s elapsed]
cker_image.nginx: Still creating... [20s elapsed]
cker_image.nginx: Creation complete after 24s [id=sha256:88736fe827391462a4db99252117f136b2b25d1d31719006326a437bb40cb12dnginx:latest]
cker_container.nginx: Creating...
cker_container.nginx: Creation complete after 7s [id=cae567e6df2891cf2c8f32aa25ca3b199d6cee066afcd5c6709b4bb672538cdd]

Warning: Deprecated attribute
```

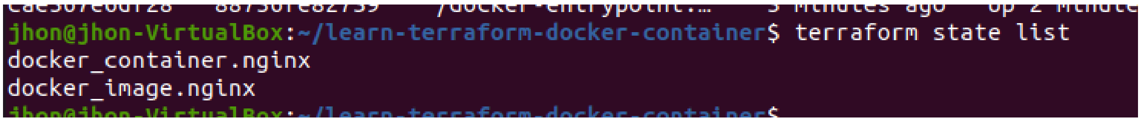


```
jhon@jhon-VirtualBox:~/learn-terraform-docker-container$ terraform show
# docker_container.nginx:
resource "docker_container" "nginx" {
    attach                = false
    command               = [
        "nginx",
        "-g",
        "daemon off;",
    ]
    cpu_shares            = 0
    entrypoint            = [
        "/docker-entrypoint.sh",
    ]
    env                   = []
    gateway               = "172.17.0.1"
    hostname              = "cae567e6df28"
    id                    = "cae567e6df2891cf2c8f32aa25ca3b199d6cee066afcd5c6709b4bb672538cdd"
    image                 = "sha256:88736fe827391462a4db99252117f136b2b25d1d31719006326a437bb40cb12d"
    init                  = false
    ip_address            = "172.17.0.2"
    ip_prefix_length      = 16
    ipc_mode              = "private"
    log_driver            = "json-file"
    logs                  = false
    max_retry_count       = 0
    memory                = 0
    memory_swap           = 0
    must_run              = true
    name                  = "tutorial"
    network_data          = [
        {
            gateway                   = "172.17.0.1"
            global_ipv6_address       = ""
            global_ipv6_prefix_length = 0
            ip_address                = "172.17.0.2"
            ip_prefix_length          = 16
            ipv6_gateway              = ""
            network_name              = "bridge"
        },
    ]
    network_mode          = "default"
    privileged            = false
    publish_all_ports     = false
    read_only             = false
    remove_volumes        = true
    restart               = "no"
    rm                    = false
    security_opts         = []
    shm_size              = 64
```



```
   network_data      = [
       {
           gateway                   = "172.17.0.1"
           global_ipv6_address       = ""
           global_ipv6_prefix_length = 0
           ip_address                = "172.17.0.2"
           ip_prefix_length          = 16
           ipv6_gateway              = ""
           network_name              = "bridge"
       },
   ]
   network_mode      = "default"
   privileged        = false
   publish_all_ports = false
   read_only         = false
   remove_volumes    = true
   restart           = "no"
   rm                = false
   security_opts     = []
   shm_size          = 64
   start             = true
   stdin_open        = false
   tty               = false

   ports {
       external = 8000
       internal = 80
       ip       = "0.0.0.0"
       protocol = "tcp"
   }


 docker_image.nginx:
esource "docker_image" "nginx" {
   id            = "sha256:88736fe827391462a4db99252117f136b2b25d1d31719006326a437bb40cb12dnginx:latest"
   keep_locally = false
   latest        = "sha256:88736fe827391462a4db99252117f136b2b25d1d31719006326a437bb40cb12d"
   name          = "nginx:latest"
   repo_digest   = "nginx@sha256:e209ac2f37c70c1e0e9873a5f7231e91dcd83fdf1178d8ed36c2ec09974210ba"

hon@jhon-VirtualBox:~/learn-terraform-docker-container$ sudo docker ps
```

## terraform state list



```
cae307e0df28   88736fe82739   "/docker-entrypoint.…"   3 minutes ago   Up 2 minute
jhon@jhon-VirtualBox:~/learn-terraform-docker-container$ terraform state list
docker_container.nginx
docker_image.nginx
jhon@jhon-VirtualBox:~/learn-terraform-docker-container$
```