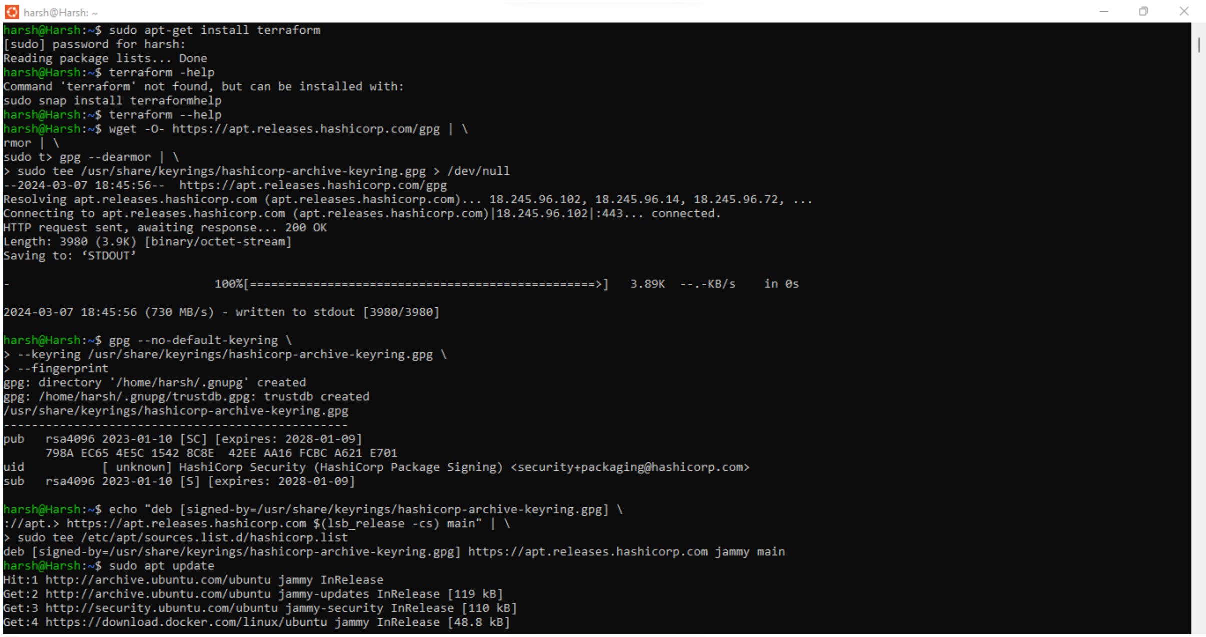
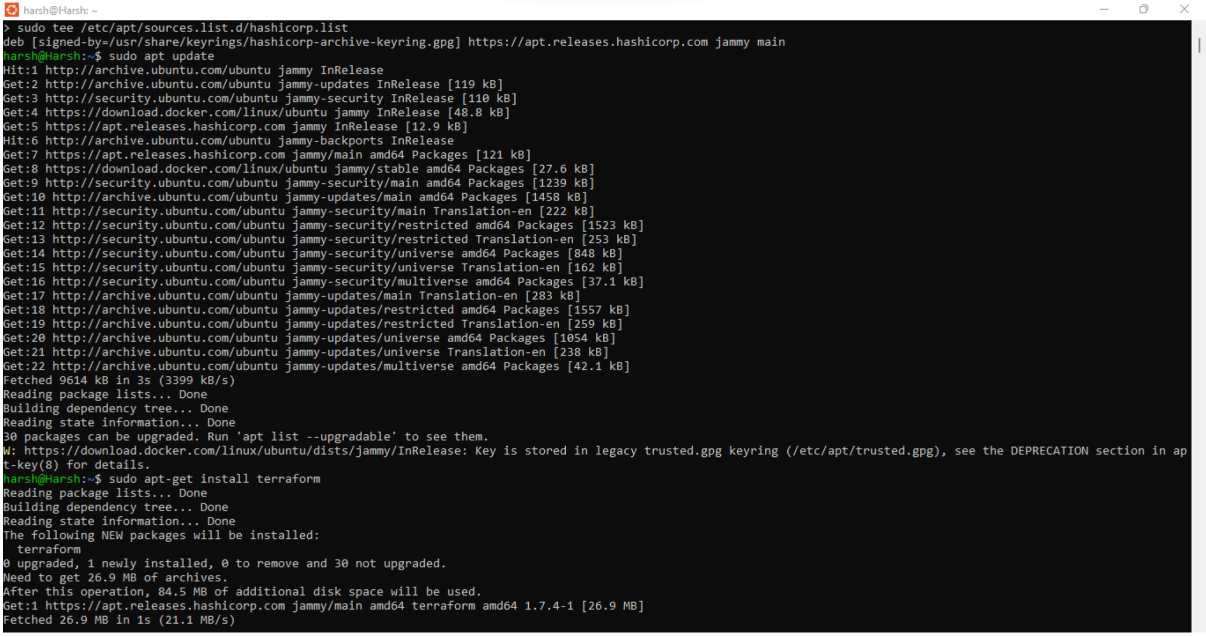
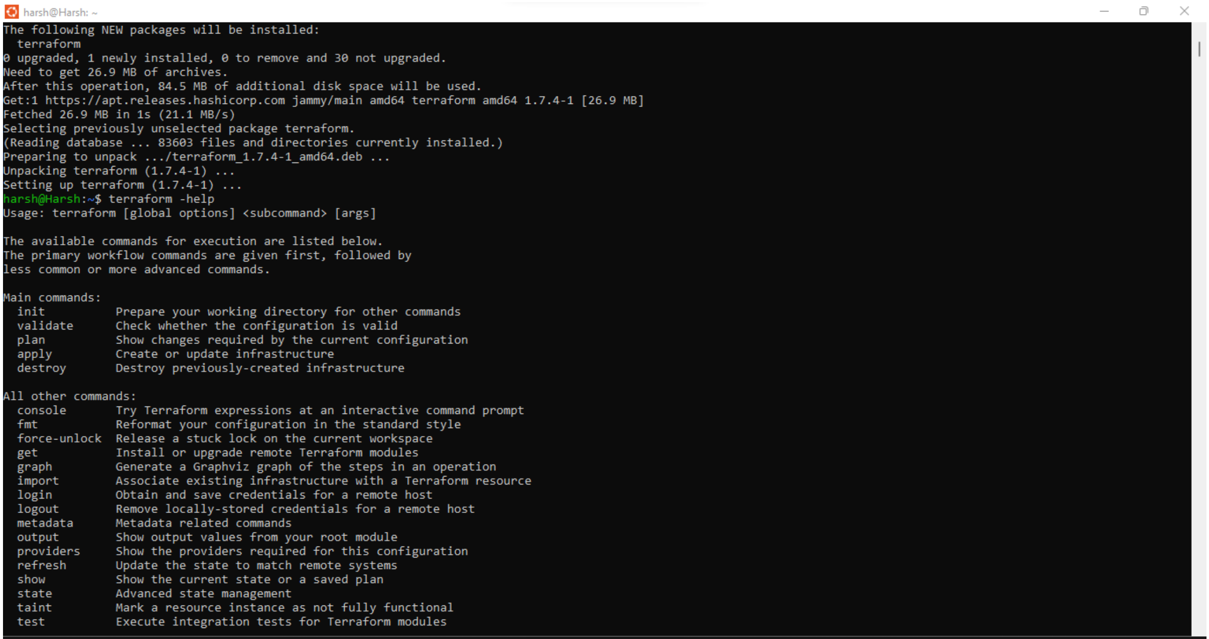


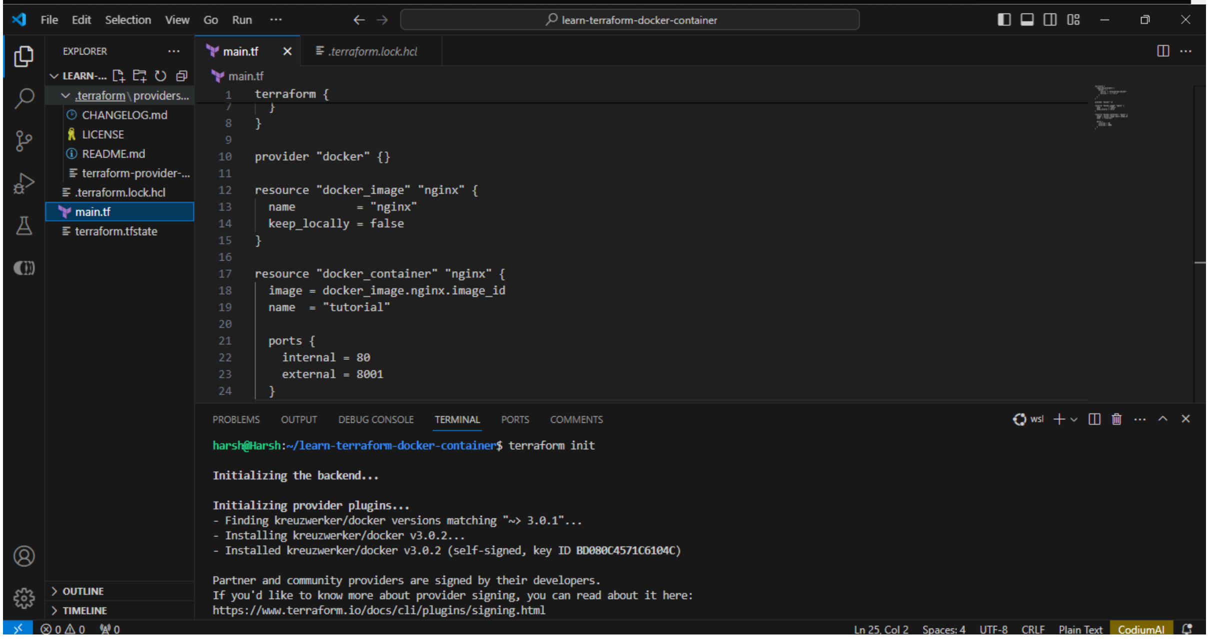
```
harsh@Harsh:~$ sudo apt-get install terraform
[sudo] password for harsh:
Reading package lists... Done
harsh@Harsh:~$ terraform -help
Command 'terraform' not found, but can be installed with:
sudo snap install terraformhelp
harsh@Harsh:~$ terraform --help
harsh@Harsh:~$ wget -O- https://apt.releases.hashicorp.com/gpg | \
rmor | \
sudo t> gpg --dearmor | \
> sudo tee /usr/share/keyrings/hashicorp-archive-keyring.gpg > /dev/null
--2024-03-07 18:45:56--  https://apt.releases.hashicorp.com/gpg
Resolving apt.releases.hashicorp.com (apt.releases.hashicorp.com)... 18.245.96.102, 18.245.96.14, 18.245.96.72, ...
Connecting to apt.releases.hashicorp.com (apt.releases.hashicorp.com)|18.245.96.102|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3980 (3.9K) [binary/octet-stream]
Saving to: ‘STDOUT’

-                              100%[=================================================>]   3.89K  --.-KB/s    in 0s

2024-03-07 18:45:56 (730 MB/s) - written to stdout [3980/3980]

harsh@Harsh:~$ gpg --no-default-keyring \
> --keyring /usr/share/keyrings/hashicorp-archive-keyring.gpg \
> --fingerprint
gpg: directory '/home/harsh/.gnupg' created
gpg: /home/harsh/.gnupg/trustdb.gpg: trustdb created
/usr/share/keyrings/hashicorp-archive-keyring.gpg
-------------------------------------------------
pub   rsa4096 2023-01-10 [SC] [expires: 2028-01-09]
      798A EC65 4E5C 1542 8C8E  42EE AA16 FCBC A621 E701
uid           [ unknown] HashiCorp Security (HashiCorp Package Signing) <security+packaging@hashicorp.com>
sub   rsa4096 2023-01-10 [S] [expires: 2028-01-09]

harsh@Harsh:~$ echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] \
://apt.> https://apt.releases.hashicorp.com $(lsb_release -cs) main" | \
> sudo tee /etc/apt/sources.list.d/hashicorp.list
deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com jammy main
harsh@Harsh:~$ sudo apt update
Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:4 https://download.docker.com/linux/ubuntu jammy InRelease [48.8 kB]
```



```
> sudo tee /etc/apt/sources.list.d/hashicorp.list
deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com jammy main
harsh@Harsh:~$ sudo apt update
Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:4 https://download.docker.com/linux/ubuntu jammy InRelease [48.8 kB]
Get:5 https://apt.releases.hashicorp.com jammy InRelease [12.9 kB]
Hit:6 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:7 https://apt.releases.hashicorp.com jammy/main amd64 Packages [121 kB]
Get:8 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages [27.6 kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1239 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1458 kB]
Get:11 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [222 kB]
Get:12 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1523 kB]
Get:13 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [253 kB]
Get:14 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [848 kB]
Get:15 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [162 kB]
Get:16 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [37.1 kB]
Get:17 http://archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [283 kB]
Get:18 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [1557 kB]
Get:19 http://archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [259 kB]
Get:20 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1054 kB]
Get:21 http://archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [238 kB]
Get:22 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [42.1 kB]
Fetched 9614 kB in 3s (3399 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
30 packages can be upgraded. Run 'apt list --upgradable' to see them.
W: https://download.docker.com/linux/ubuntu/dists/jammy/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in ap
t-key(8) for details.
harsh@Harsh:~$ sudo apt-get install terraform
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  terraform
0 upgraded, 1 newly installed, 0 to remove and 30 not upgraded.
Need to get 26.9 MB of archives.
After this operation, 84.5 MB of additional disk space will be used.
Get:1 https://apt.releases.hashicorp.com jammy/main amd64 terraform amd64 1.7.4-1 [26.9 MB]
Fetched 26.9 MB in 1s (21.1 MB/s)
```

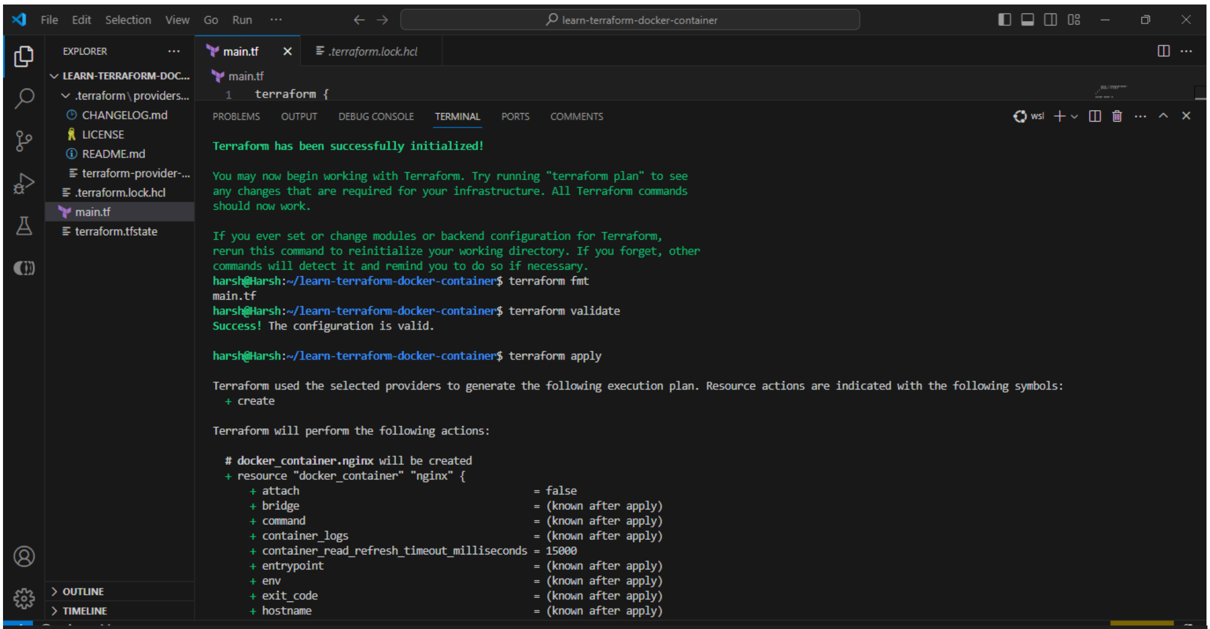


```
The following NEW packages will be installed:
  terraform
0 upgraded, 1 newly installed, 0 to remove and 30 not upgraded.
Need to get 26.9 MB of archives.
After this operation, 84.5 MB of additional disk space will be used.
Get:1 https://apt.releases.hashicorp.com jammy/main amd64 terraform amd64 1.7.4-1 [26.9 MB]
Fetched 26.9 MB in 1s (21.1 MB/s)
Selecting previously unselected package terraform.
(Reading database ... 83603 files and directories currently installed.)
Preparing to unpack .../terraform_1.7.4-1_amd64.deb ...
Unpacking terraform (1.7.4-1) ...
Setting up terraform (1.7.4-1) ...
harsh@Harsh:~$ terraform -help
Usage: terraform [global options] <subcommand> [args]

The available commands for execution are listed below.
The primary workflow commands are given first, followed by
less common or more advanced commands.

Main commands:
  init          Prepare your working directory for other commands
  validate      Check whether the configuration is valid
  plan          Show changes required by the current configuration
  apply         Create or update infrastructure
  destroy       Destroy previously-created infrastructure

All other commands:
  console       Try Terraform expressions at an interactive command prompt
  fmt           Reformat your configuration in the standard style
  force-unlock  Release a stuck lock on the current workspace
  get           Install or upgrade remote Terraform modules
  graph         Generate a Graphviz graph of the steps in an operation
  import        Associate existing infrastructure with a Terraform resource
  login         Obtain and save credentials for a remote host
  logout        Remove locally-stored credentials for a remote host
  metadata      Metadata related commands
  output        Show output values from your root module
  providers     Show the providers required for this configuration
  refresh       Update the state to match remote systems
  show          Show the current state or a saved plan
  state         Advanced state management
  taint         Mark a resource instance as not fully functional
  test          Execute integration tests for Terraform modules
```



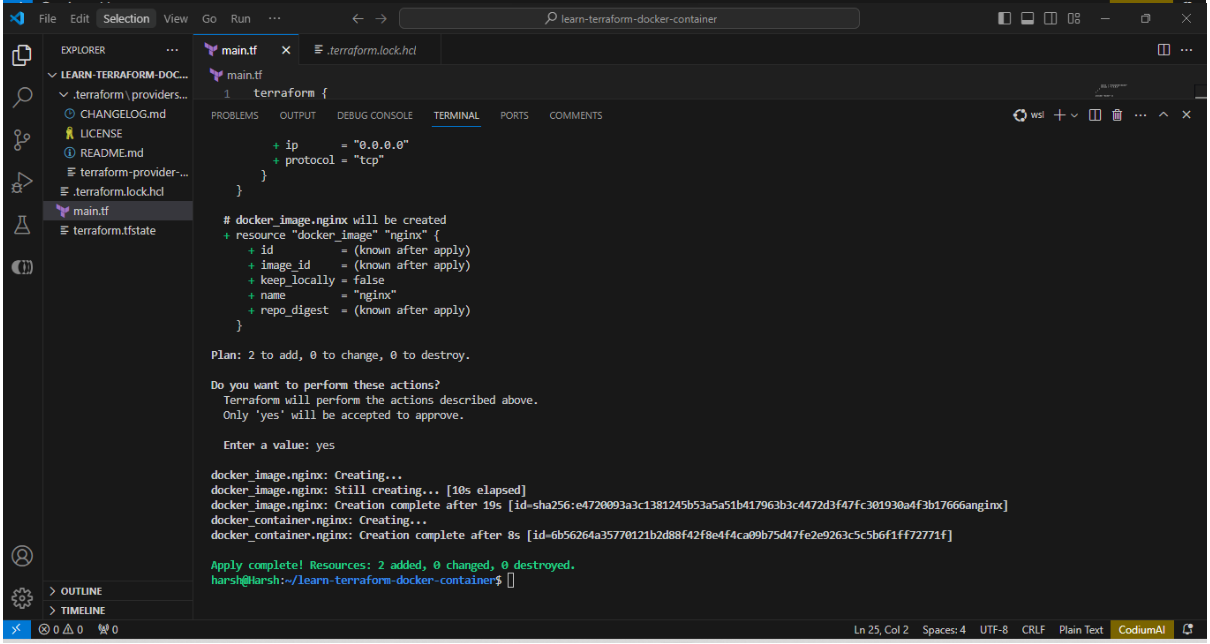
```
terraform {
  }
}

provider "docker" {}

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

resource "docker_container" "nginx" {
  image = docker_image.nginx.image_id
  name  = "tutorial"

  ports {
    internal = 80
    external = 8001
  }
```

```
harsh@Harsh:~/learn-terraform-docker-container$ 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 {
```

```
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.
harsh@Harsh:~/learn-terraform-docker-container$ terraform fmt
main.tf
harsh@Harsh:~/learn-terraform-docker-container$ terraform validate
Success! The configuration is valid.

harsh@Harsh:~/learn-terraform-docker-container$ 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)
      + exit_code                                   = (known after apply)
      + hostname                                    = (known after apply)
```


File Edit Selection View Go Run ···
learn-terraform-docker-container
EXPLORER
LEARN-TERRAFORM-DOC...
.terraform \ providers...
CHANGELOG.md
LICENSE
README.md
terraform-provider-...
.terraform.lock.hcl
main.tf
terraform.tfstate
OUTLINE
TIMELINE
main.tf
.terraform.lock.hcl
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
wsl
0 0 0
Ln 25, Col 2 Spaces: 4 UTF-8 CRLF Plain Text CodiumAI

```
terraform {
```

```
          + 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"
      + repo_digest  = (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: Creation complete after 19s [id=sha256:e4720093a3c1381245b53a5a51b417963b3c4472d3f47fc301930a4f3b17666anginx]
docker_container.nginx: Creating...
docker_container.nginx: Creation complete after 8s [id=6b56264a35770121b2d88f42f8e4f4ca09b75d47fe2e9263c5c5b6f1ff72771f]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
harsh@Harsh:~/learn-terraform-docker-container$ 
```