

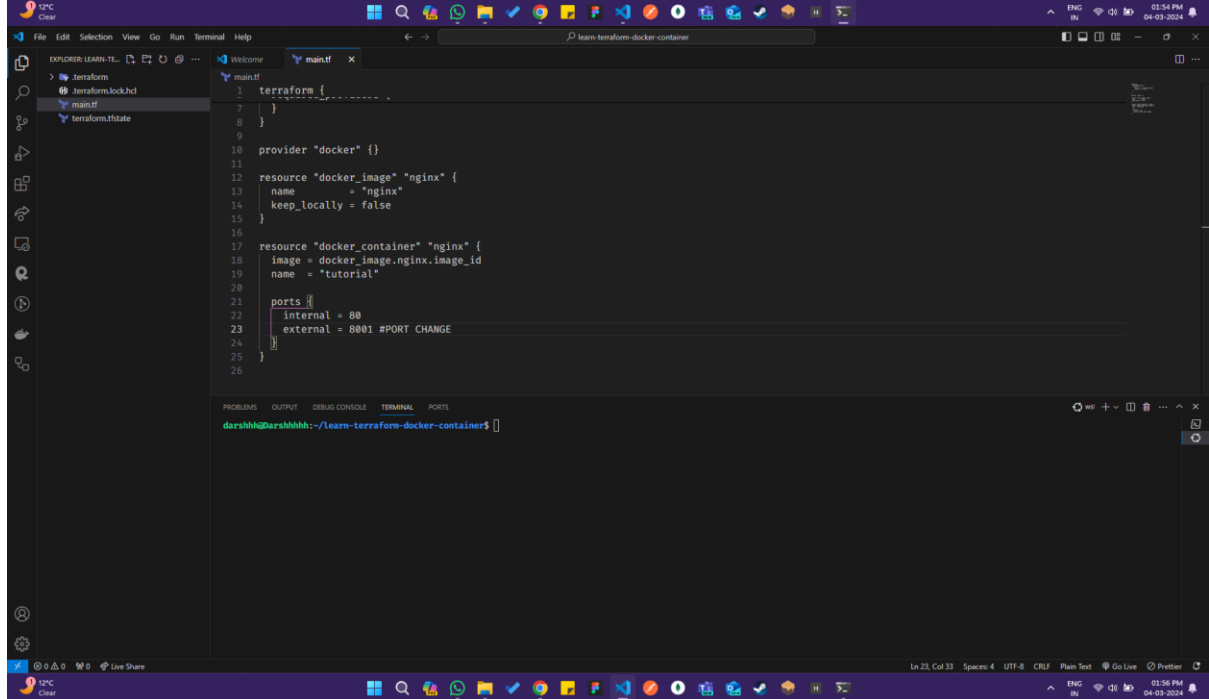
```
darshhh@Darshhhh: ~$ sudo apt-get install terraform
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package terraform
darshhh@Darshhhh:~$ terraform -help
Command 'terraform' not found, but can be installed with:
sudo snap install terraform
darshhh@Darshhhh:~$ wget -O- https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg
--2024-03-04 13:50:21-- https://apt.releases.hashicorp.com/gpg
Resolving apt.releases.hashicorp.com (apt.releases.hashicorp.com)... File '/usr/share/keyrings/hashicorp-archive-keyring.gpg' exists. Overwrite? (y/N) 18.245.96.72, 18.245.96.102, 18.245.96.14, ...
Connecting to apt.releases.hashicorp.com (apt.releases.hashicorp.com)[18.245.96.72]:443... connected.
HTTP request sent, awaiting response... 200 OK

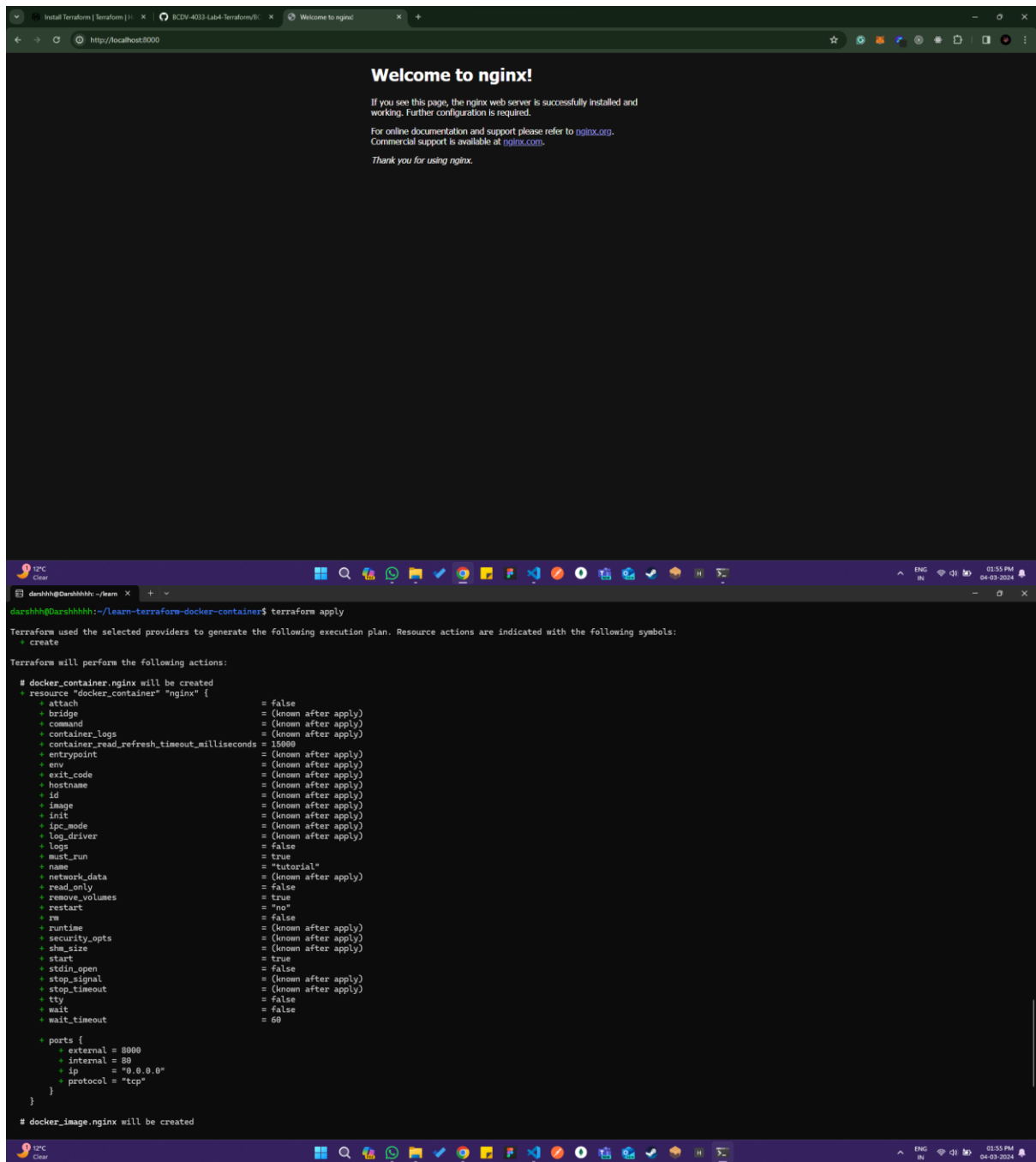
Length: 3980 (3.9K) [binary/octet-stream]
Saving to: 'STDOUT'

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

2024-03-04 13:50:21 (135 MB/s) - written to stdout [3980/3980]

Overwrite? (y/N) y
darshhh@Darshhhh:~$ echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d/hashicorp.list
deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com jammy main
darshhh@Darshhhh:~$ sudo apt update && sudo apt install terraform
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 https://download.docker.com/Linux/ubuntu jammy InRelease
Get:3 https://apt.releases.hashicorp.com jammy InRelease [12.9 kB]
Hit:4 http://archive.ubuntu.com/ubuntu jammy InRelease
Hit:5 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:6 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:7 https://apt.releases.hashicorp.com jammy/main amd64 Packages [121 kB]
Fetched 134 kB in 6s (300 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
33 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 apt-key(8) for details.
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 33 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 3s (7809 kB/s)
Selecting previously unselected package terraform.
(Reading database ... 82825 files and directories currently installed.)
Preparing to unpack .../terraform_1.7.4-1_amd64.deb ...
Unpacking terraform (1.7.4-1) ...
```





```
darshhh@Darshhhh: ~/learn X + v
darshhh@Darshhhh:~/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 BD088C4571C6104C)

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.
darshhh@Darshhhh:~/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)
  + id                   = (known after apply)
  + image                 = (known after apply)
  + init                  = (known after apply)
  + ipc_mode              = (known after apply)
  + log_driver            = (known after apply)
  + logs                  = false
}
```