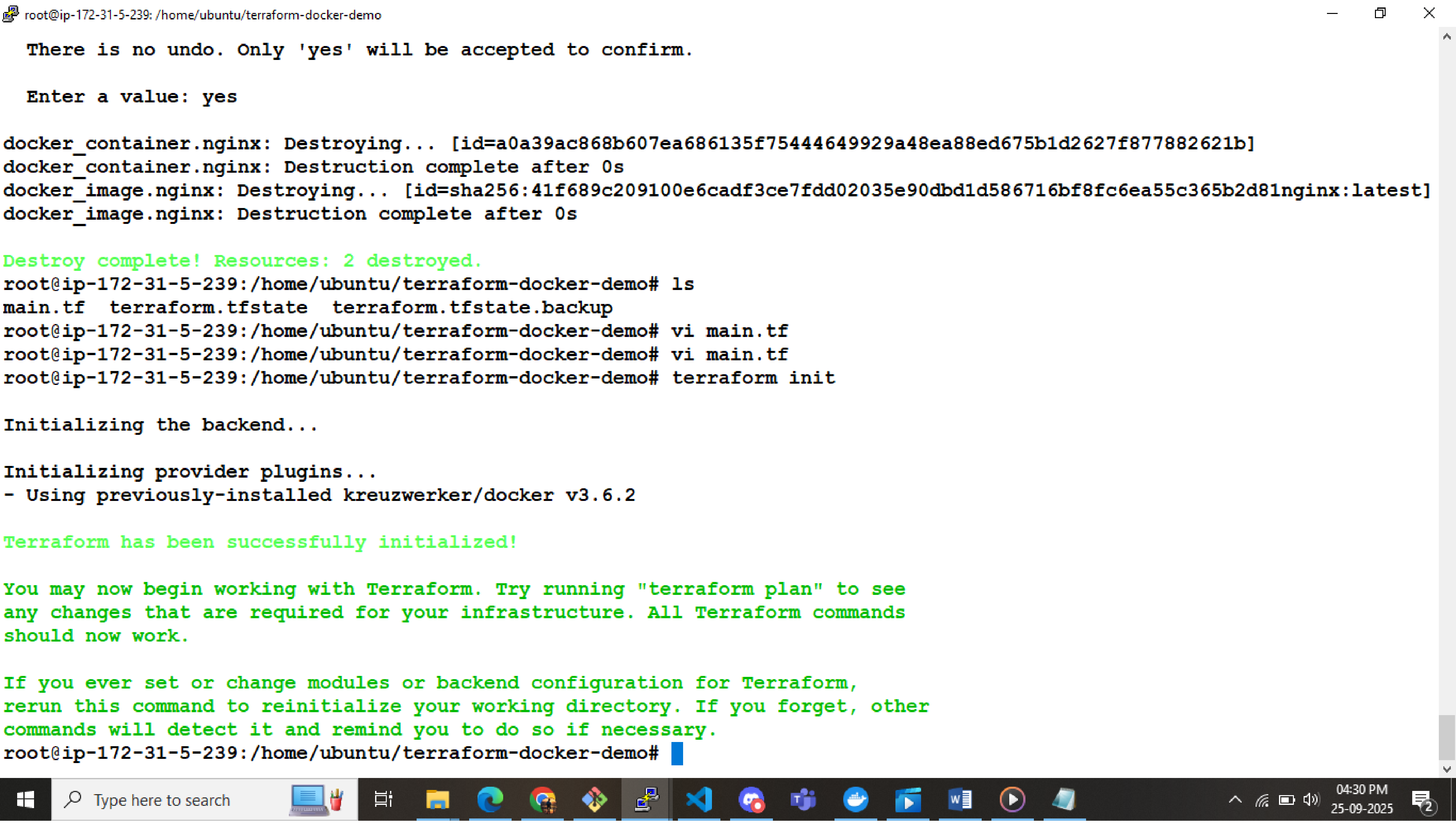
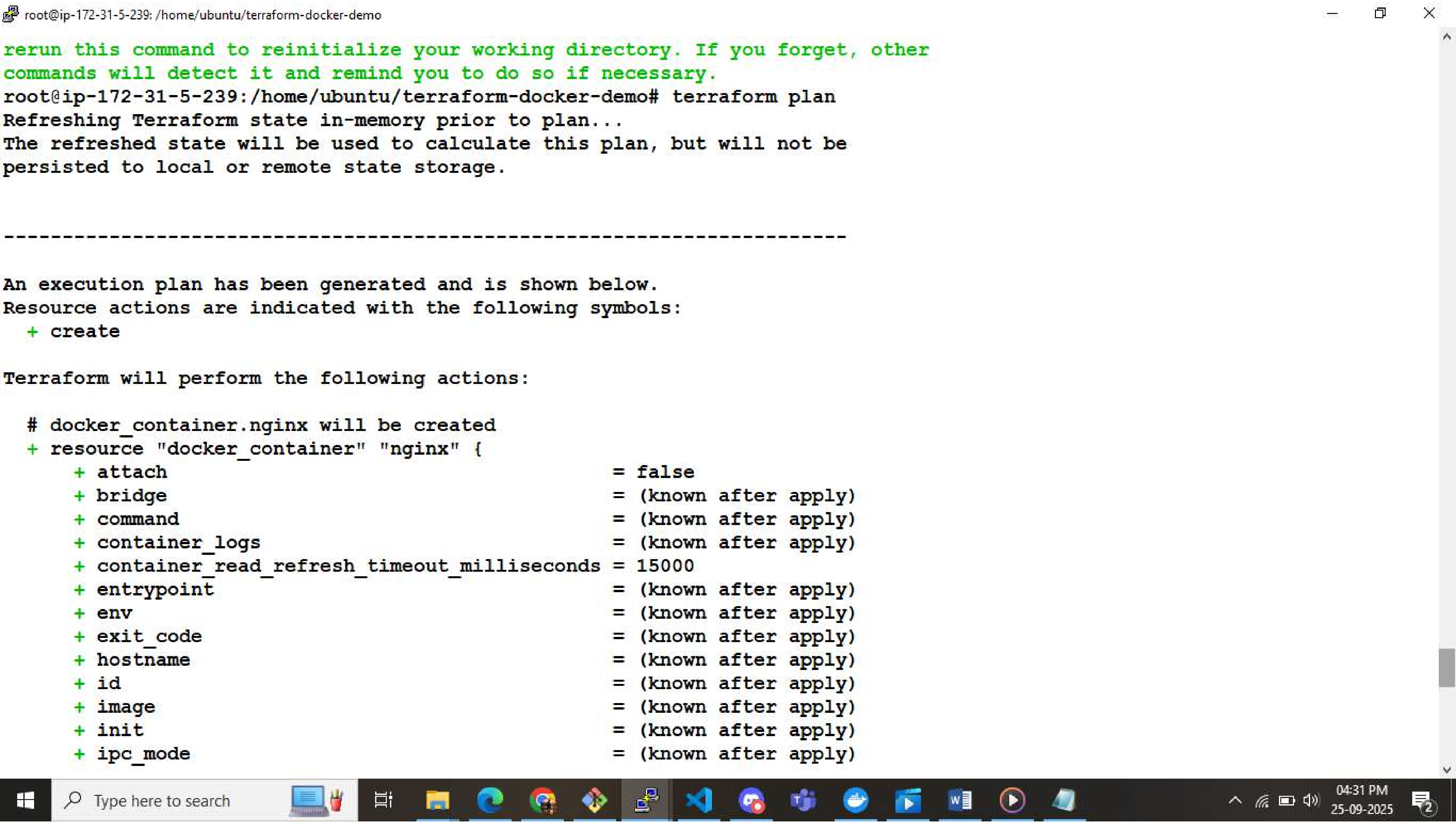


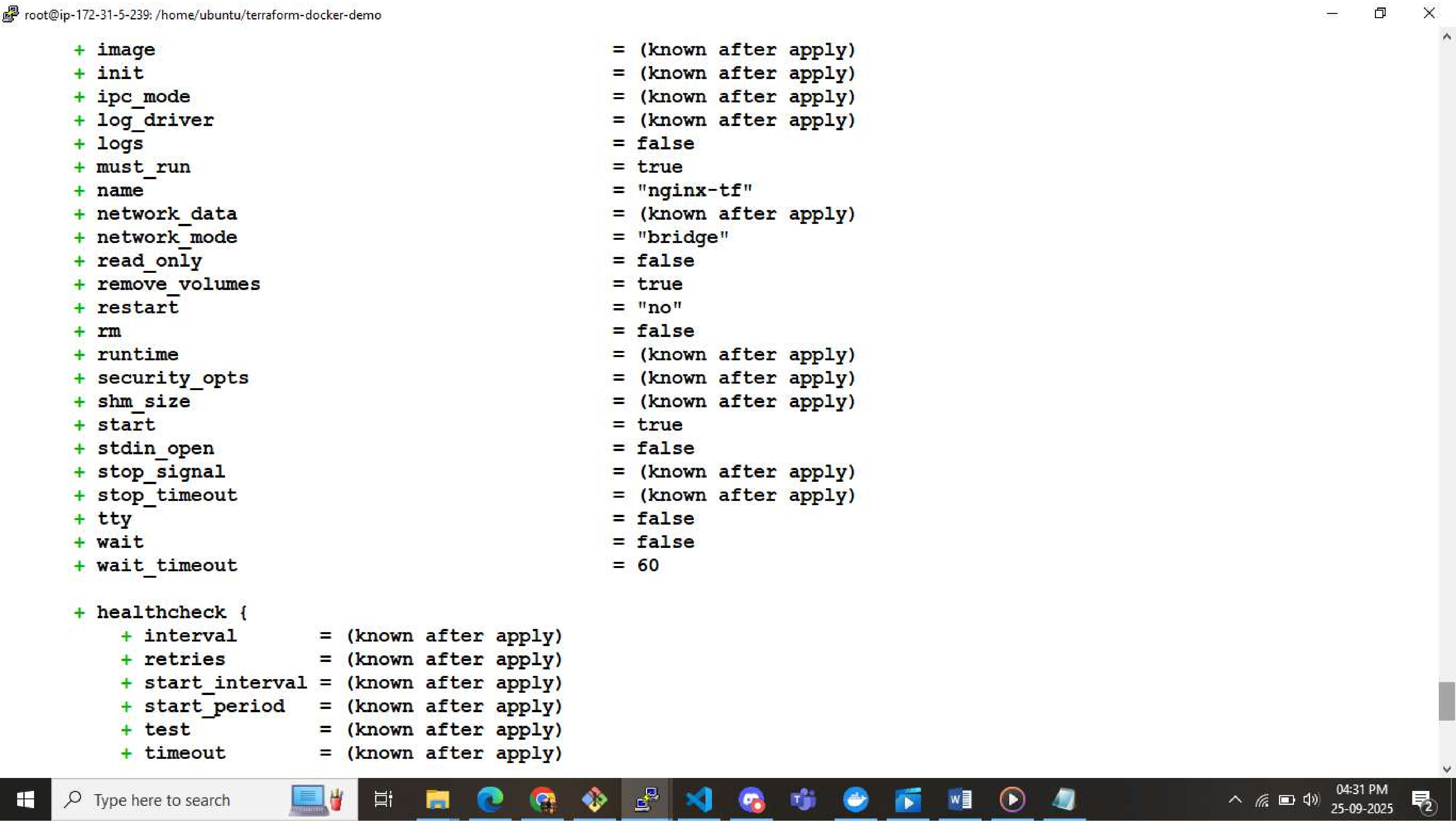
root@ip-172-31-5-239: /home/ubuntu/terraform-docker-demo

```
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

docker_container.nginx: Destroying... [id=a0a39ac868b607ea686135f75444649929a48ea88ed675b1d2627f877882621b]
docker_container.nginx: Destruction complete after 0s
docker_image.nginx: Destroying... [id=sha256:41f689c209100e6cadf3ce7fdd02035e90dbd1d586716bf8fc6ea55c365b2d81nginx:latest]
docker_image.nginx: Destruction complete after 0s

Destroy complete! Resources: 2 destroyed.
root@ip-172-31-5-239:/home/ubuntu/terraform-docker-demo# ls
main.tf  terraform.tfstate  terraform.tfstate.backup
root@ip-172-31-5-239:/home/ubuntu/terraform-docker-demo# vi main.tf
root@ip-172-31-5-239:/home/ubuntu/terraform-docker-demo# vi main.tf
root@ip-172-31-5-239:/home/ubuntu/terraform-docker-demo# terraform init

Initializing the backend...

Initializing provider plugins...
- Using previously-installed kreuzwerker/docker v3.6.2

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.
root@ip-172-31-5-239:/home/ubuntu/terraform-docker-demo#
```



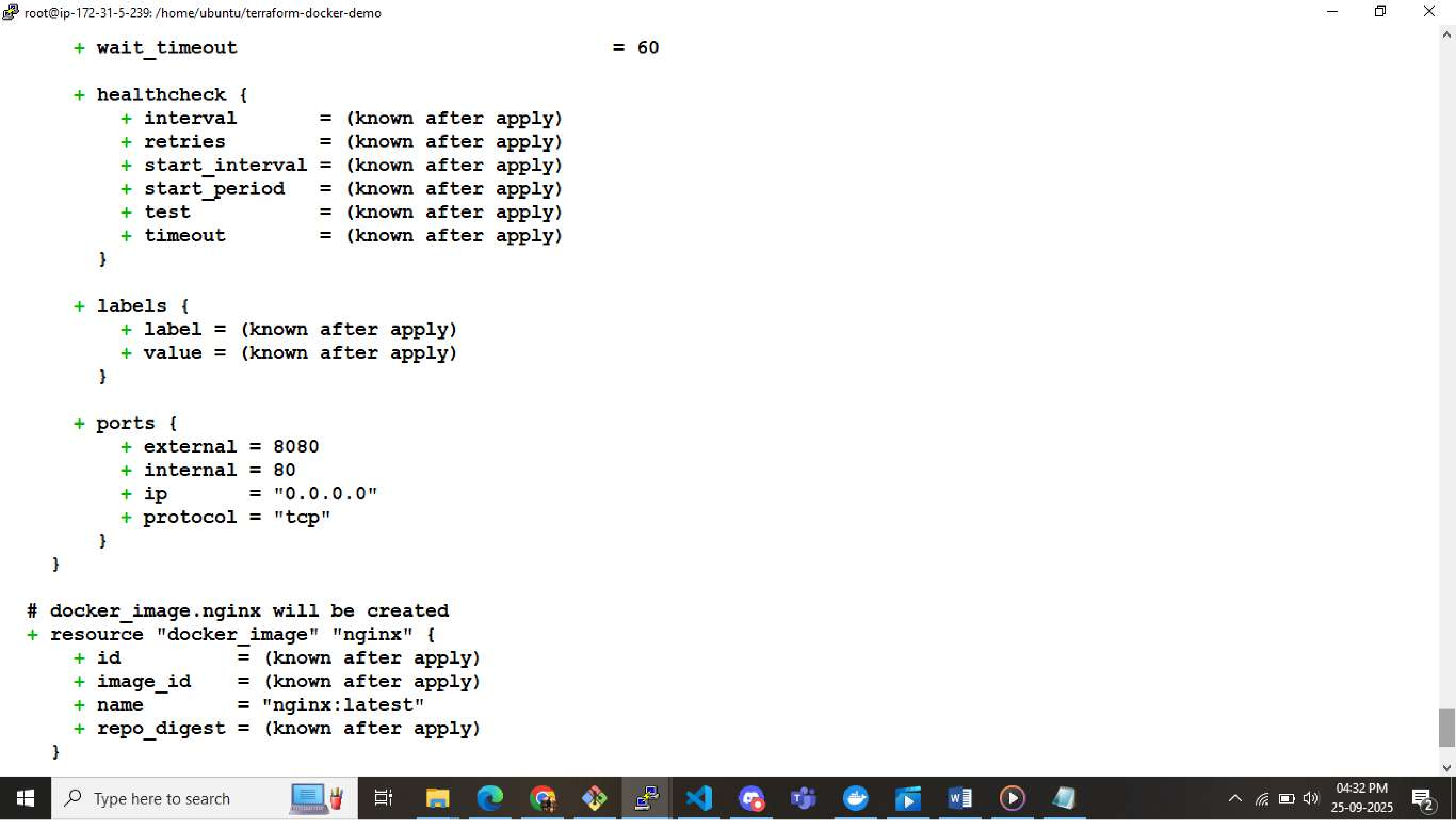
root@ip-172-31-5-239: /home/ubuntu/terraform-docker-demo

```
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
root@ip-172-31-5-239:/home/ubuntu/terraform-docker-demo# terraform plan
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.


------------------------------------------------------------------------


An execution plan has been generated and is shown below.
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)
```

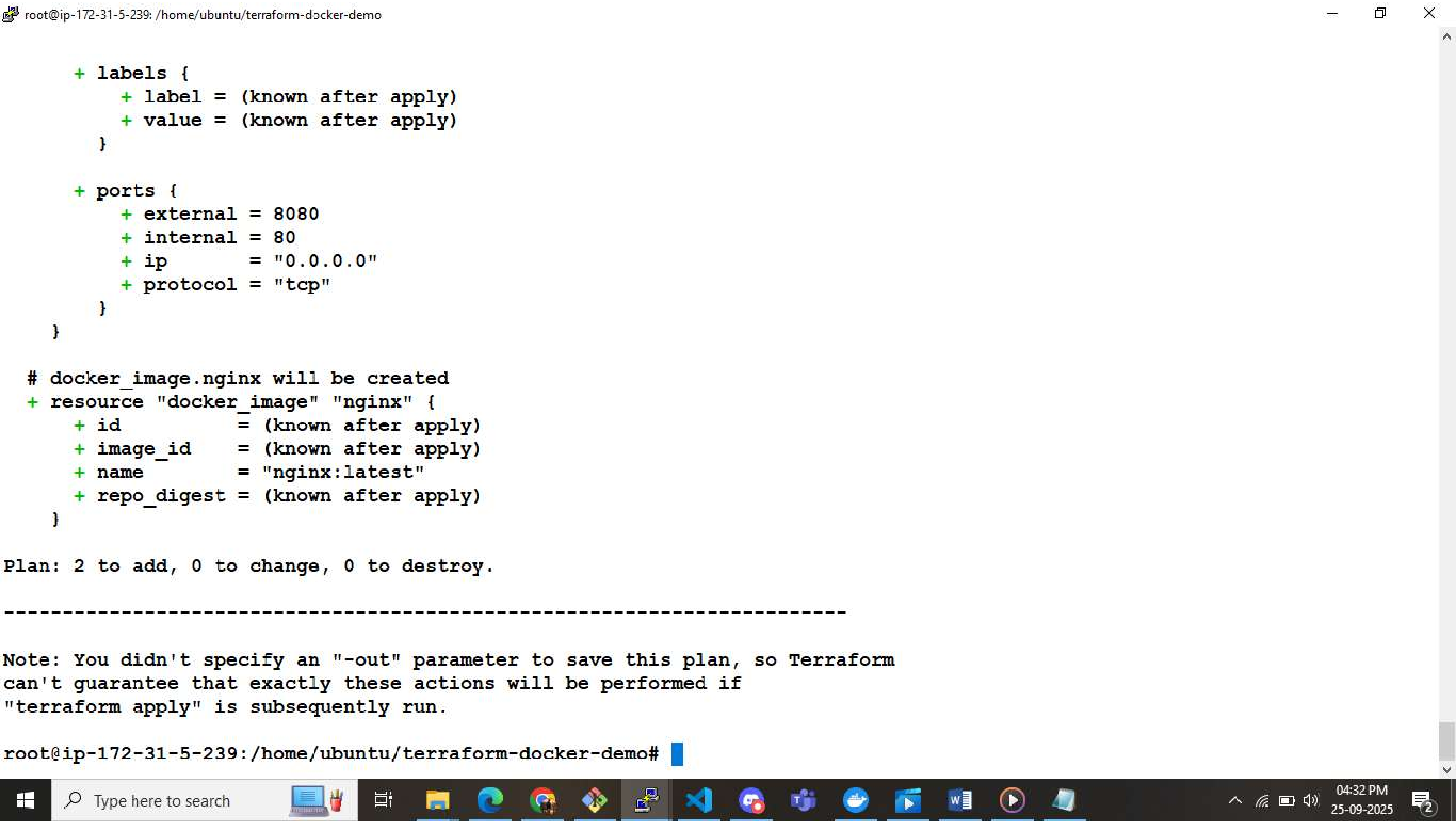


root@ip-172-31-5-239: /home/ubuntu/terraform-docker-demo

```
      + image                                      = (known after apply)
      + init                                       = (known after apply)
      + ipc_mode                                   = (known after apply)
      + log_driver                                 = (known after apply)
      + logs                                       = false
      + must_run                                   = true
      + name                                       = "nginx-tf"
      + network_data                               = (known after apply)
      + network_mode                               = "bridge"
      + 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

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

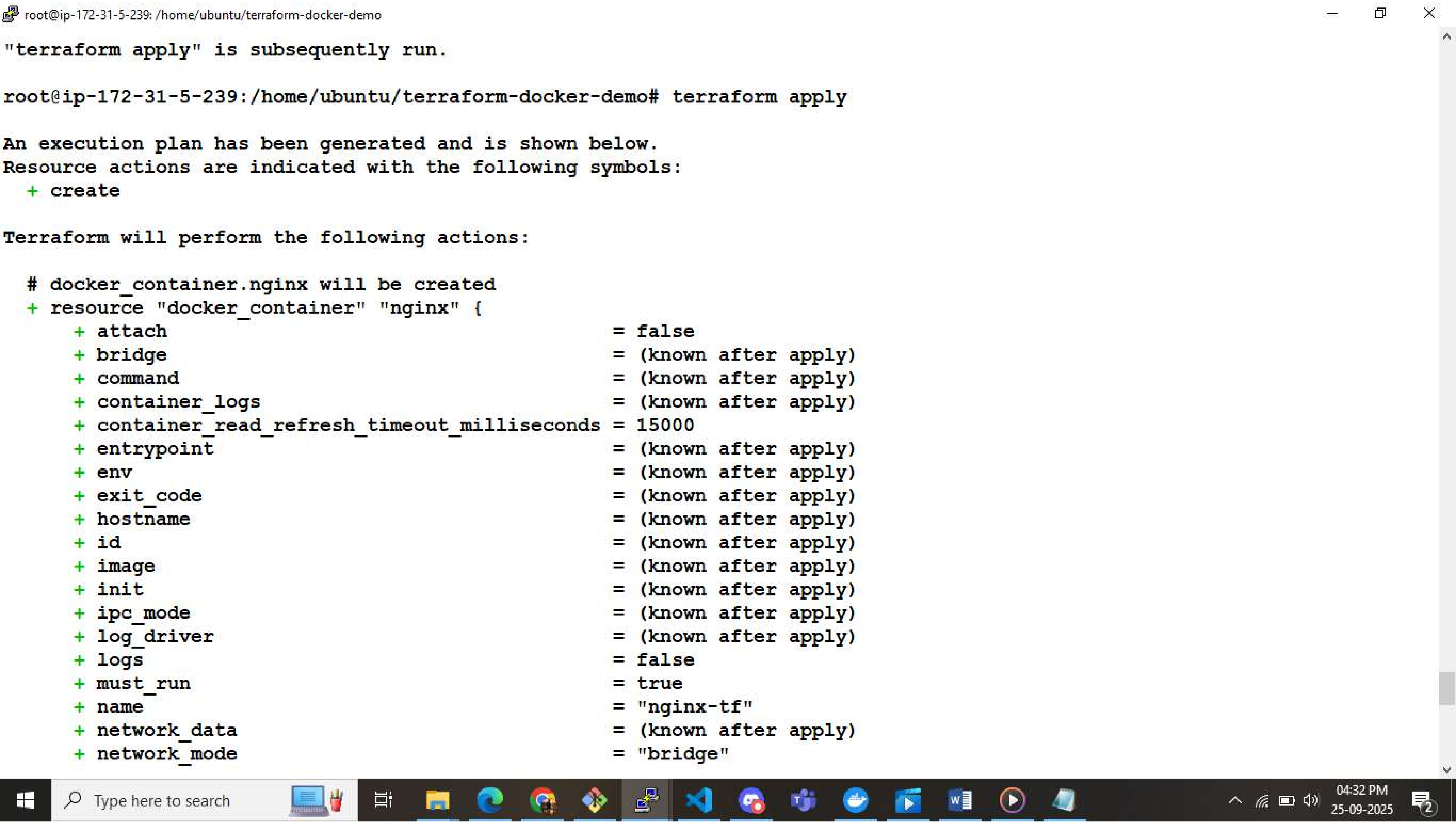

root@ip-172-31-5-239: /home/ubuntu/terraform-docker-demo

```
      + wait_timeout                                = 60

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

      + labels {
          + label = (known after apply)
          + value = (known after apply)
        }

      + ports {
          + external = 8080
          + 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)
      + name         = "nginx:latest"
      + repo_digest  = (known after apply)
    }
```



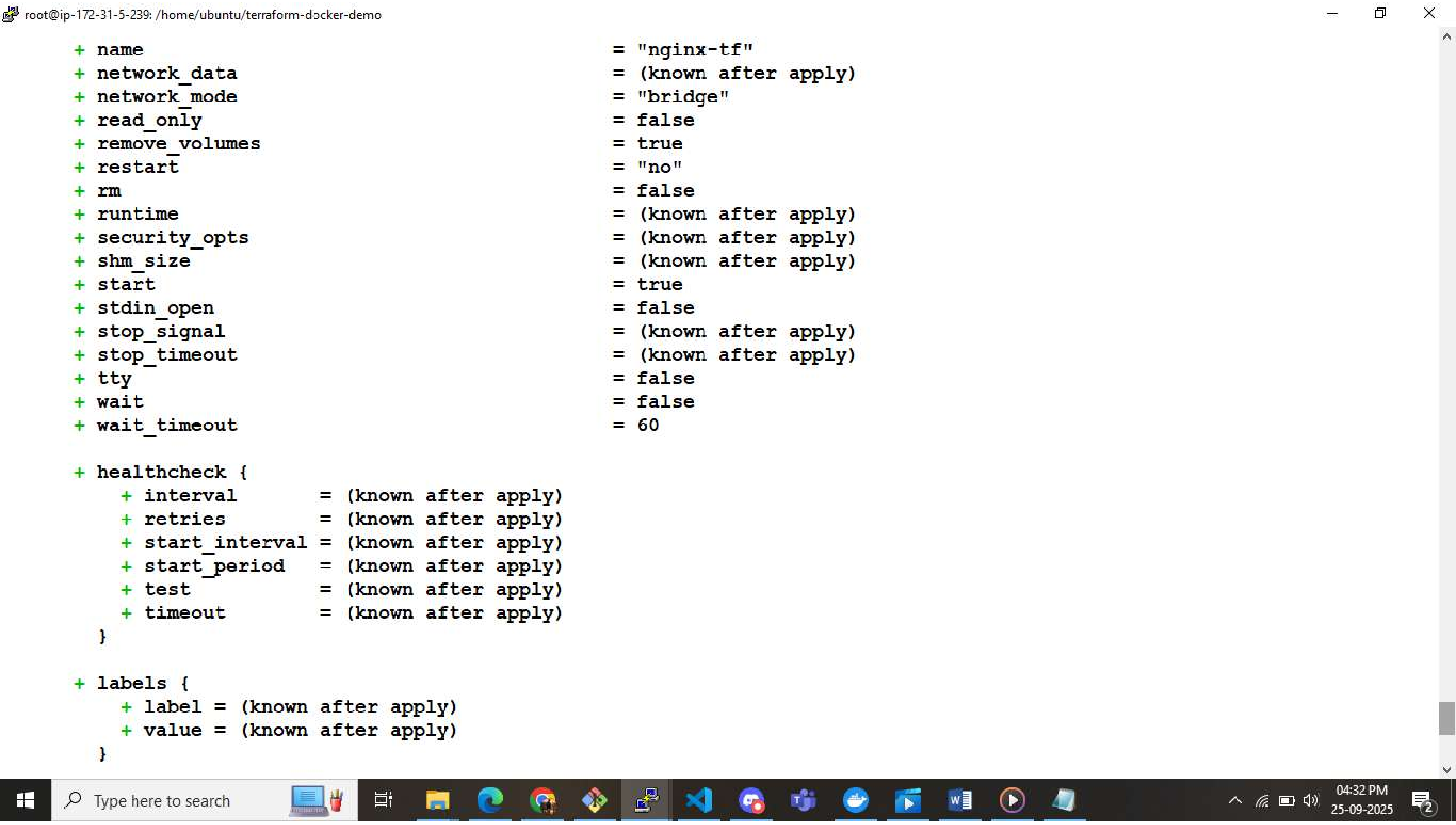
root@ip-172-31-5-239: /home/ubuntu/terraform-docker-demo

```
      + labels {
          + label = (known after apply)
          + value = (known after apply)
        }

      + ports {
          + external = 8080
          + 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)
      + name        = "nginx:latest"
      + repo_digest = (known after apply)
    }

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

------------------------------------------------------------------------

Note: You didn't specify an "-out" parameter to save this plan, so Terraform
can't guarantee that exactly these actions will be performed if
"terraform apply" is subsequently run.

root@ip-172-31-5-239:/home/ubuntu/terraform-docker-demo#
```



root@ip-172-31-5-239: /home/ubuntu/terraform-docker-demo

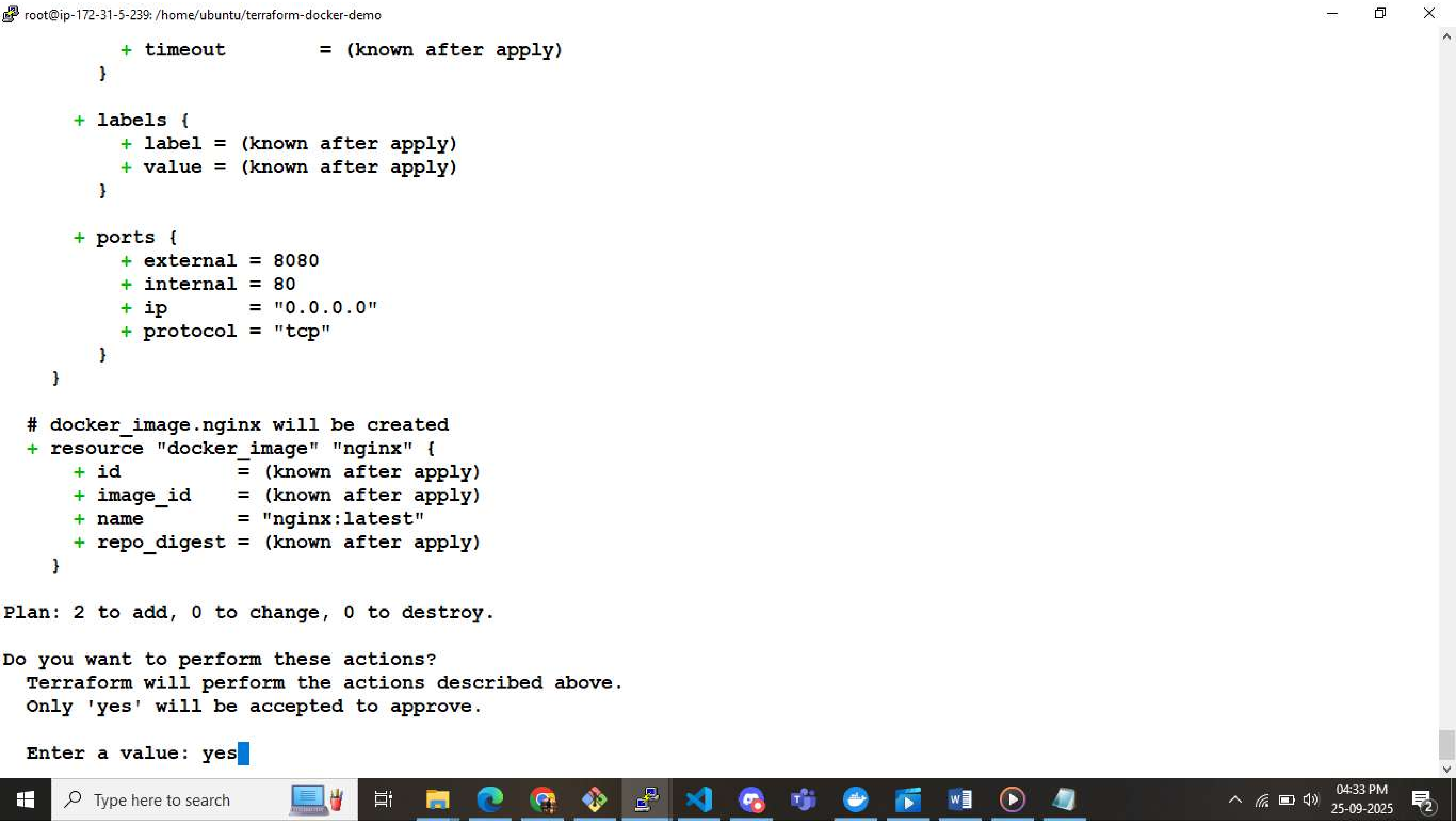
```
"terraform apply" is subsequently run.

root@ip-172-31-5-239:/home/ubuntu/terraform-docker-demo# terraform apply

An execution plan has been generated and is shown below.
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                                        = "nginx-tf"
      + network_data                                = (known after apply)
      + network_mode                                = "bridge"
```


root@ip-172-31-5-239: /home/ubuntu/terraform-docker-demo

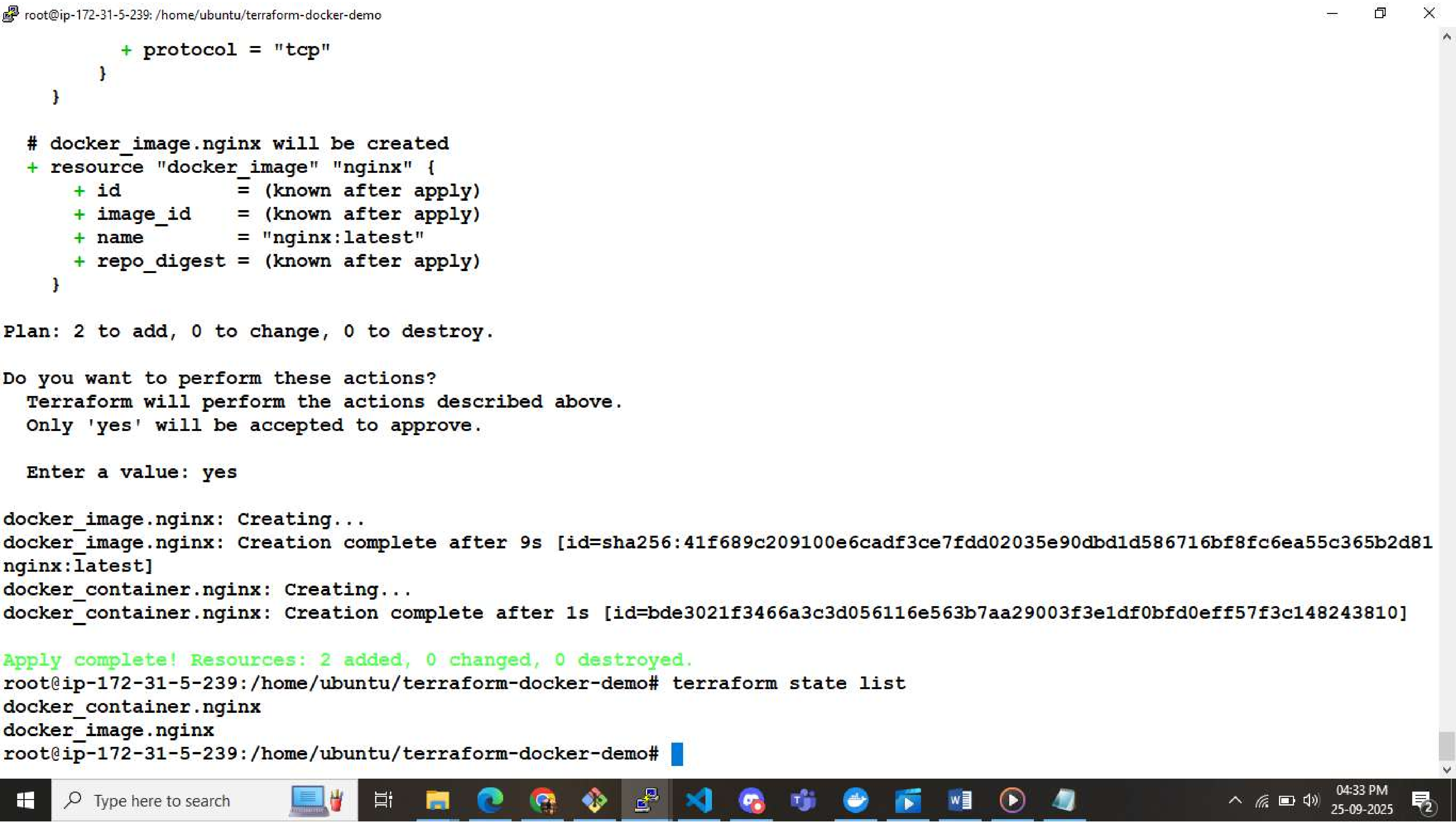
```
    + name                                              = "nginx-tf"
    + network_data                                      = (known after apply)
    + network_mode                                      = "bridge"
    + read_only                                         = false
    + remove_volumes                                    = 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

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

    + labels {
        + label = (known after apply)
        + value = (known after apply)
      }
```

Type here to search

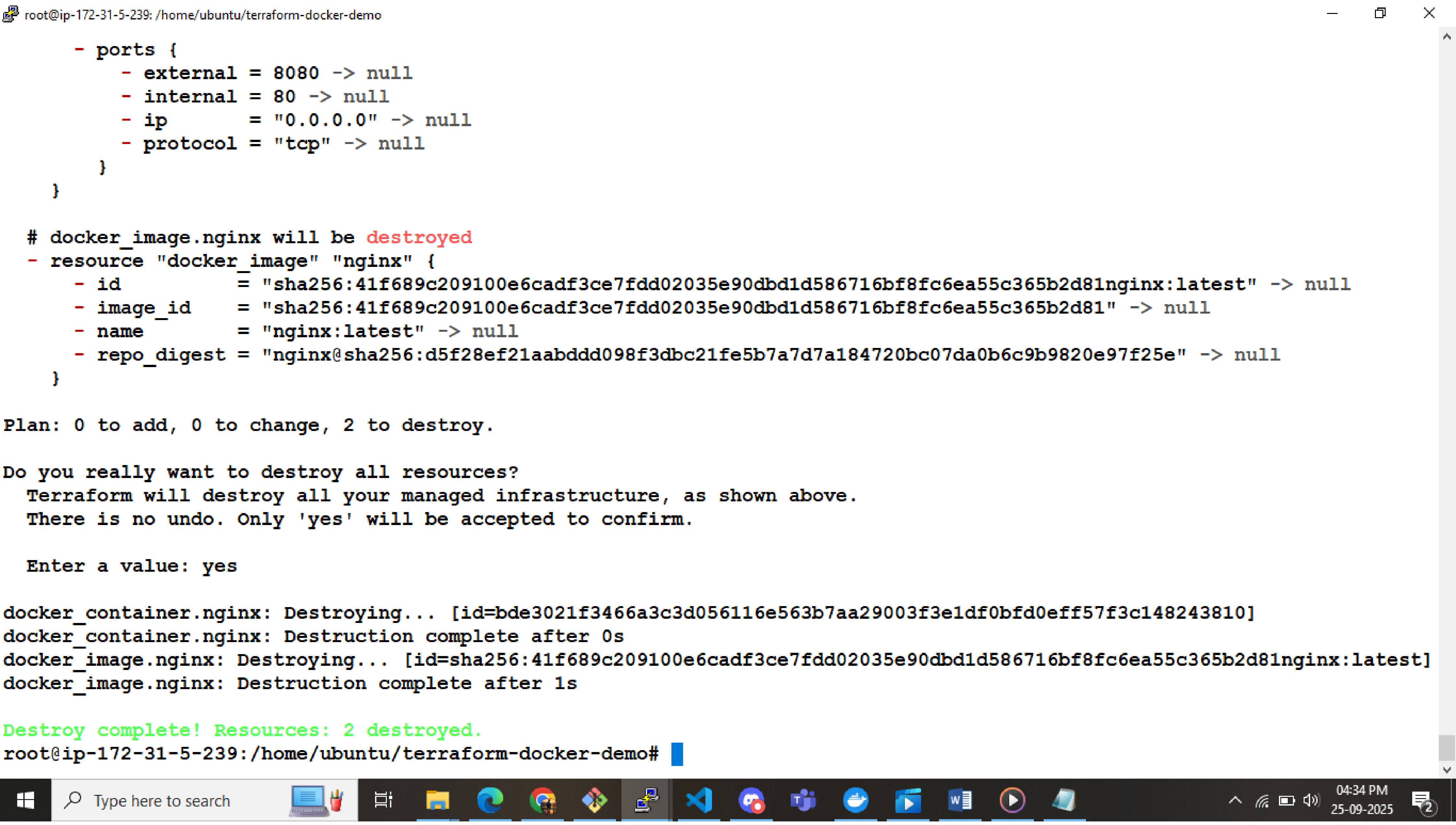
04:32 PM
25-09-2025


root@ip-172-31-5-239: /home/ubuntu/terraform-docker-demo

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

      + labels {
          + label = (known after apply)
          + value = (known after apply)
        }

      + ports {
          + external = 8080
          + 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)
      + name        = "nginx:latest"
      + 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
```



```
          + protocol = "tcp"
        }
    }

  # docker_image.nginx will be created
  + resource "docker_image" "nginx" {
      + id           = (known after apply)
      + image_id     = (known after apply)
      + name         = "nginx:latest"
      + 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: Creation complete after 9s [id=sha256:41f689c209100e6cadf3ce7fdd02035e90dbd1d586716bf8fc6ea55c365b2d81
nginx:latest]
docker_container.nginx: Creating...
docker_container.nginx: Creation complete after 1s [id=bde3021f3466a3c3d056116e563b7aa29003f3e1df0bfd0eff57f3c148243810]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
root@ip-172-31-5-239:/home/ubuntu/terraform-docker-demo# terraform state list
docker_container.nginx
docker_image.nginx
root@ip-172-31-5-239:/home/ubuntu/terraform-docker-demo#
```


root@ip-172-31-5-239: /home/ubuntu/terraform-docker-demo

```
      - ports {
          - external = 8080 -> null
          - internal = 80 -> null
          - ip       = "0.0.0.0" -> null
          - protocol = "tcp" -> null
        }
    }

  # docker_image.nginx will be destroyed
  - resource "docker_image" "nginx" {
      - id          = "sha256:41f689c209100e6cadf3ce7fdd02035e90dbd1d586716bf8fc6ea55c365b2d81nginx:latest" -> null
      - image_id    = "sha256:41f689c209100e6cadf3ce7fdd02035e90dbd1d586716bf8fc6ea55c365b2d81" -> null
      - name        = "nginx:latest" -> null
      - repo_digest = "nginx@sha256:d5f28ef21aabddd098f3dbc21fe5b7a7d7a184720bc07da0b6c9b9820e97f25e" -> 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=bde3021f3466a3c3d056116e563b7aa29003f3e1df0bfd0eff57f3c148243810]
docker_container.nginx: Destruction complete after 0s
docker_image.nginx: Destroying... [id=sha256:41f689c209100e6cadf3ce7fdd02035e90dbd1d586716bf8fc6ea55c365b2d81nginx:latest]
docker_image.nginx: Destruction complete after 1s

Destroy complete! Resources: 2 destroyed.
root@ip-172-31-5-239:/home/ubuntu/terraform-docker-demo#
```