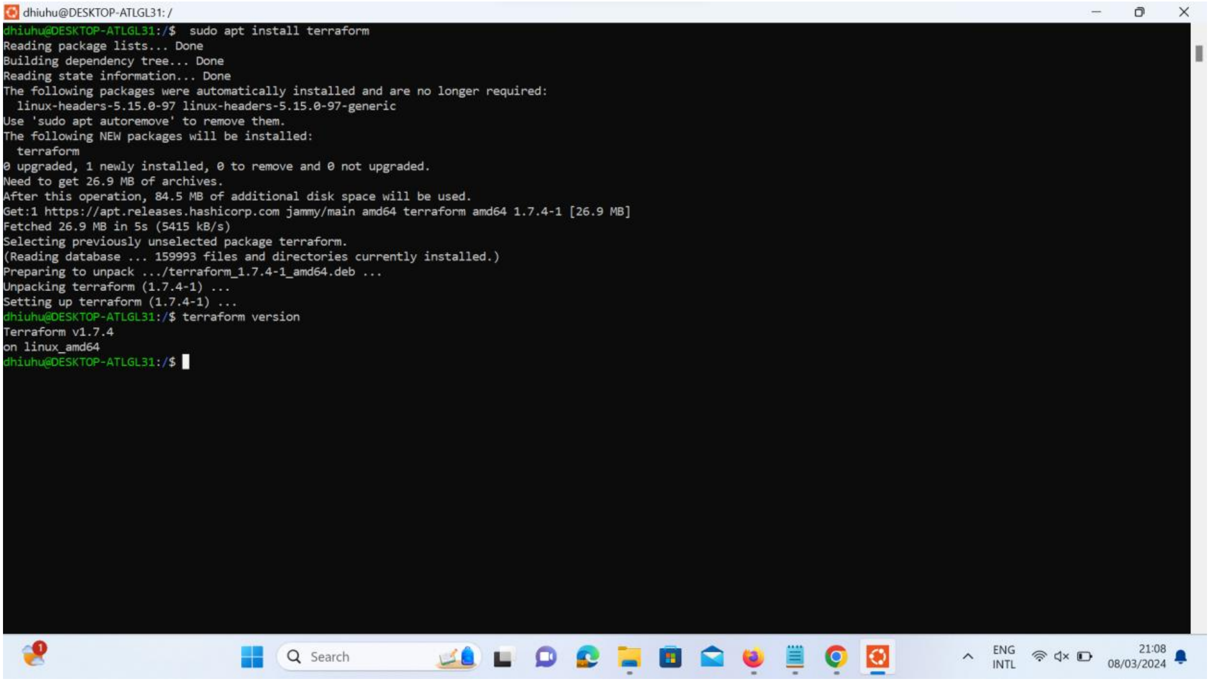
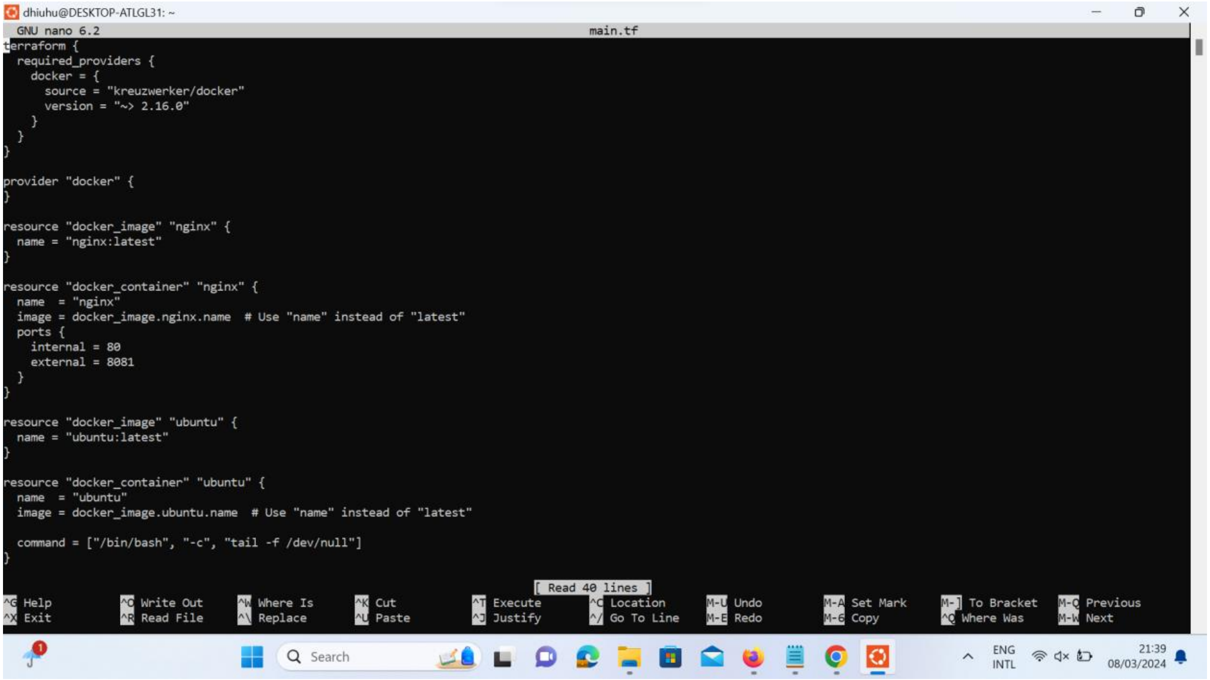
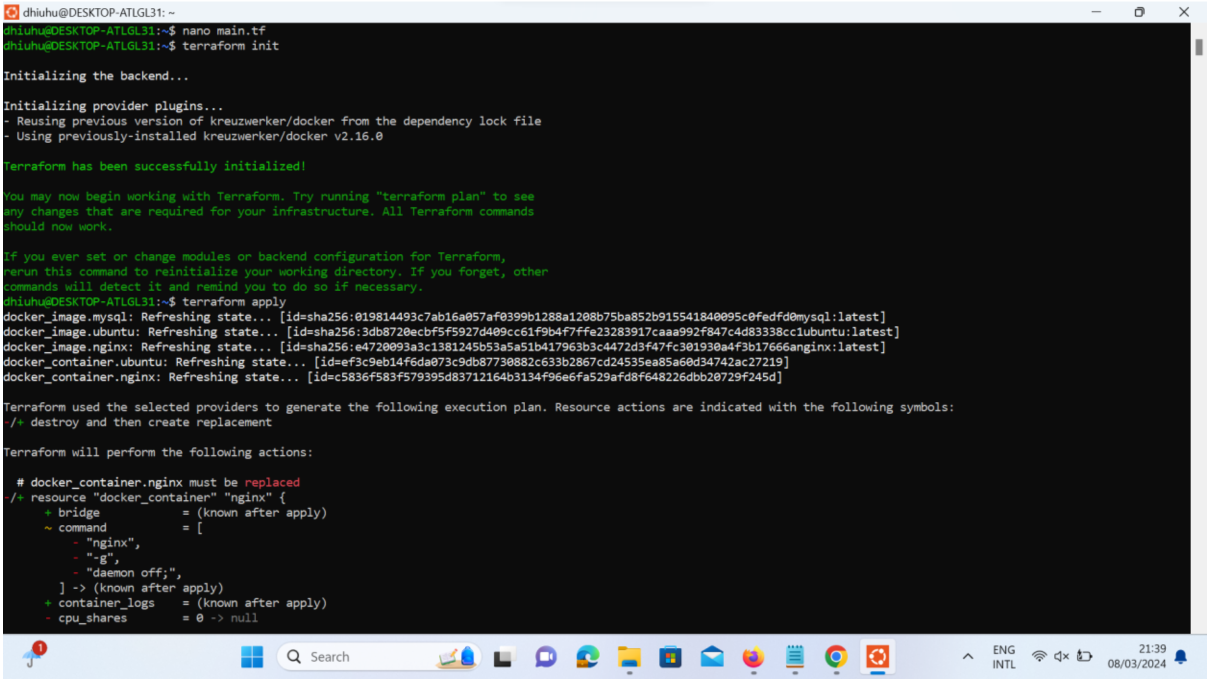


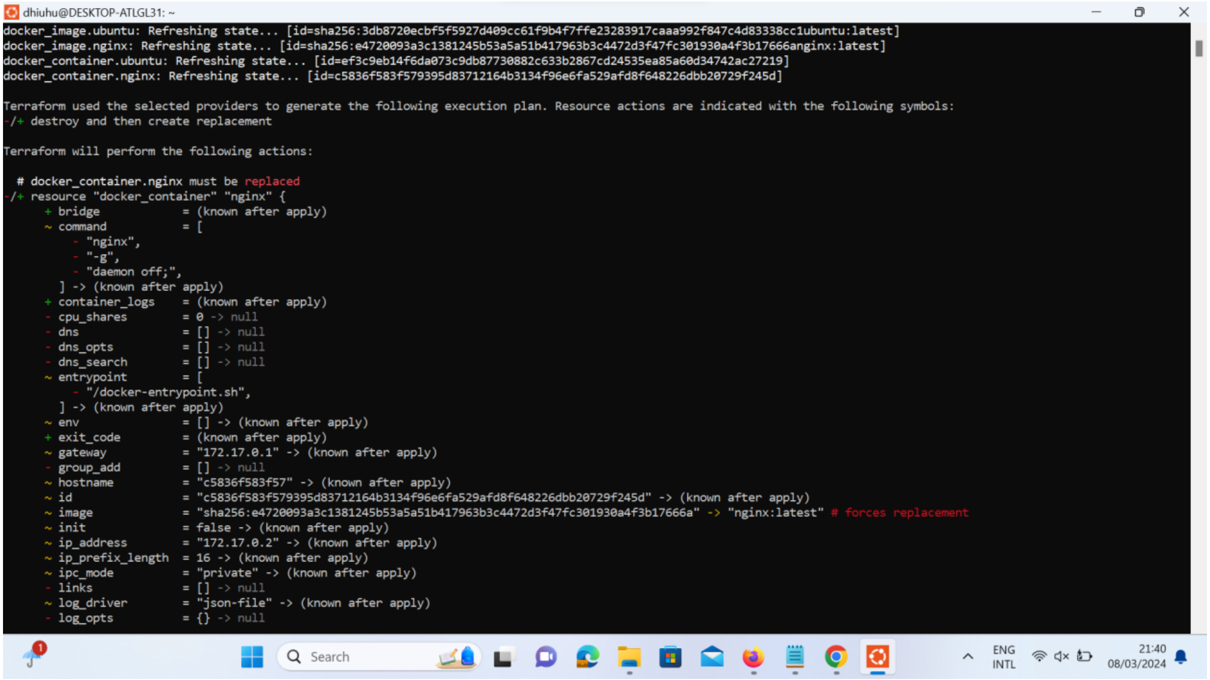
```
dhiuhu@DESKTOP-ATLGL31:/$  sudo apt install terraform
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-headers-5.15.0-97 linux-headers-5.15.0-97-generic
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  terraform
0 upgraded, 1 newly installed, 0 to remove and 0 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 5s (5415 kB/s)
Selecting previously unselected package terraform.
(Reading database ... 159993 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) ...
dhiuhu@DESKTOP-ATLGL31:/$ terraform version
Terraform v1.7.4
on linux_amd64
dhiuhu@DESKTOP-ATLGL31:/$
```



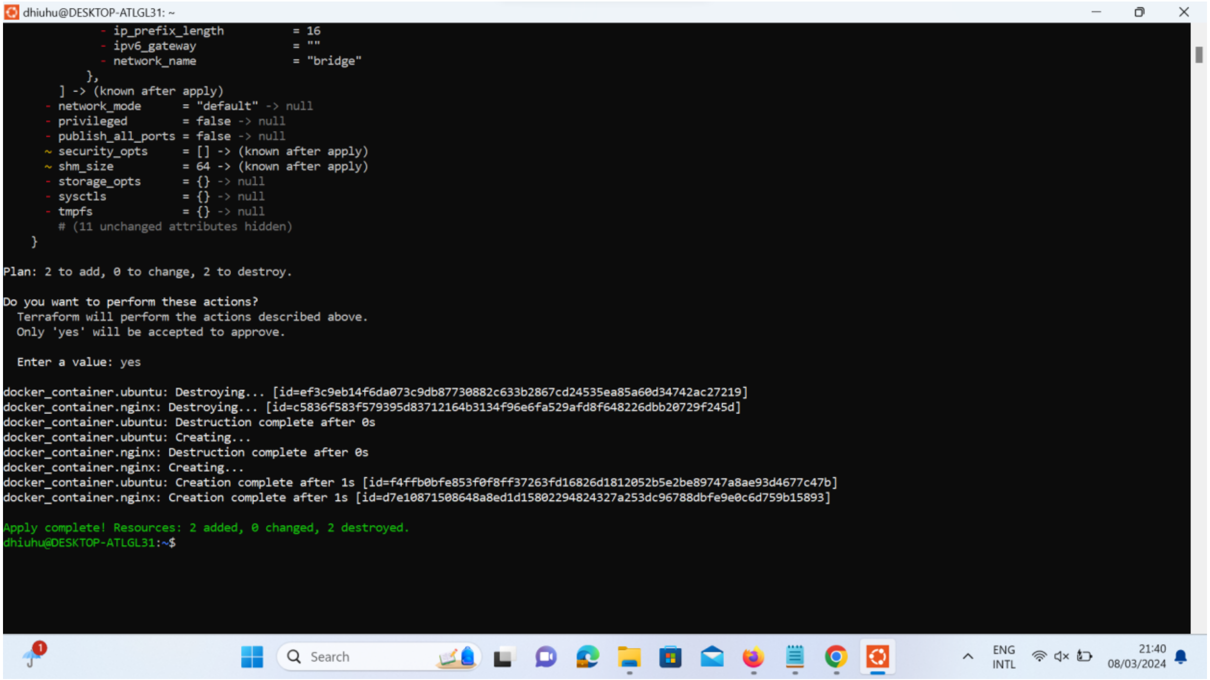
```
terraform {
  required_providers {
    docker = {
      source = "kreuzwerker/docker"
      version = "~> 2.16.0"
    }
  }
}

provider "docker" {
}

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

resource "docker_container" "nginx" {
  name  = "nginx"
  image = docker_image.nginx.name  # Use "name" instead of "latest"
  ports {
    internal = 80
    external = 8081
  }
}

resource "docker_image" "ubuntu" {
  name = "ubuntu:latest"
}

resource "docker_container" "ubuntu" {
  name  = "ubuntu"
  image = docker_image.ubuntu.name  # Use "name" instead of "latest"

  command = ["/bin/bash", "-c", "tail -f /dev/null"]
}
```



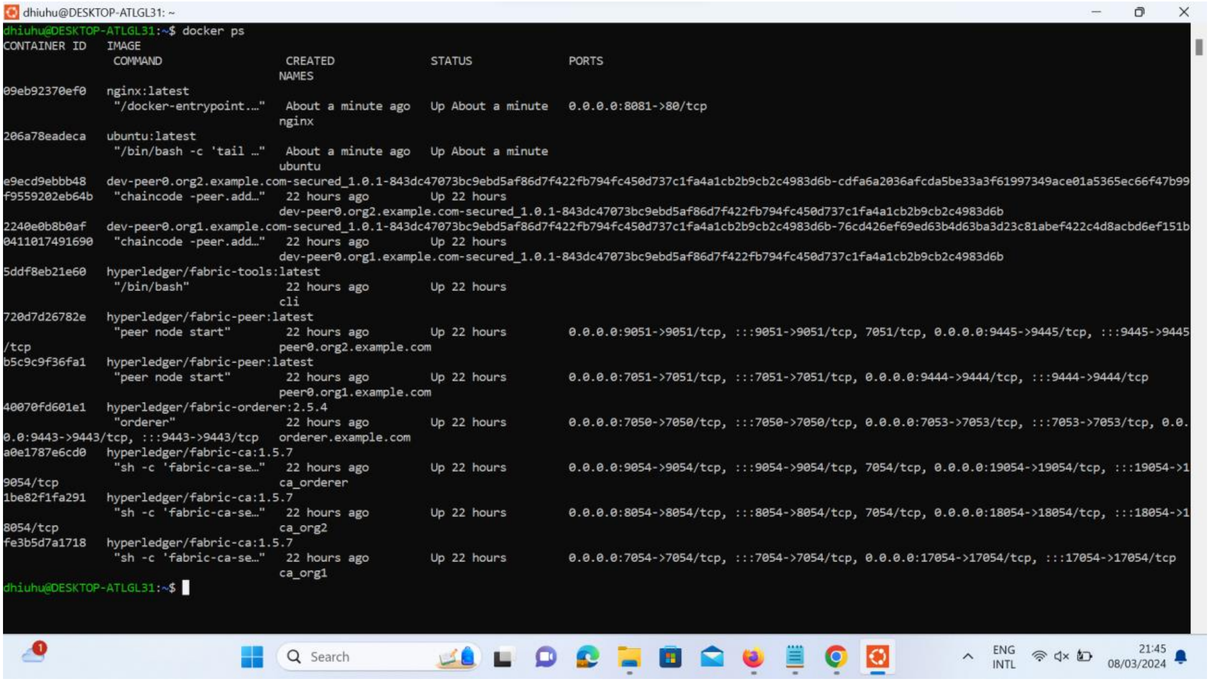
```
dhiuhu@DESKTOP-ATLGL31:~$ nano main.tf
dhiuhu@DESKTOP-ATLGL31:~$ terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of kreuzwerker/docker from the dependency lock file
- Using previously-installed kreuzwerker/docker v2.16.0

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.
dhiuhu@DESKTOP-ATLGL31:~$ terraform apply
docker_image.mysql: Refreshing state... [id=sha256:019814493c7ab16a057af0399b1288a1208b75ba852b915541840095c0fedfd0mysql:latest]
docker_image.ubuntu: Refreshing state... [id=sha256:3db8720ecbf5f5927d409cc61f9b4f7ffe23283917caaa992f847c4d83338cc1ubuntu:latest]
docker_image.nginx: Refreshing state... [id=sha256:e4720093a3c1381245b53a5a51b417963b3c4472d3f47fc301930a4f3b17666anginx:latest]
docker_container.ubuntu: Refreshing state... [id=ef3c9eb14f6da073c9db87730882c633b2867cd24535ea85a60d34742ac27219]
docker_container.nginx: Refreshing state... [id=c5836f583f579395d83712164b3134f96e6fa529afd8f648226dbb20729f245d]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
-/+ destroy and then create replacement

Terraform will perform the following actions:

  # docker_container.nginx must be replaced
-/+ resource "docker_container" "nginx" {
      + bridge            = (known after apply)
      ~ command           = [
          - "nginx",
          - "-g",
          - "daemon off;",
        ] -> (known after apply)
      + container_logs    = (known after apply)
      - cpu_shares        = 0 -> null
```



```
docker_image.ubuntu: Refreshing state... [id=sha256:3db8720ecbf5f5927d409cc61f9b4f7ffe23283917caaa992f847c4d83338cc1ubuntu:latest]
docker_image.nginx: Refreshing state... [id=sha256:e4720093a3c1381245b53a5a51b417963b3c4472d3f47fc301930a4f3b17666anginx:latest]
docker_container.ubuntu: Refreshing state... [id=ef3c9eb14f6da073c9db87730882c633b2867cd24535ea85a60d34742ac27219]
docker_container.nginx: Refreshing state... [id=c5836f583f579395d83712164b3134f96e6fa529afd8f648226dbb20729f245d]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
-/+ destroy and then create replacement

Terraform will perform the following actions:

  # docker_container.nginx must be replaced
-/+ resource "docker_container" "nginx" {
      + bridge            = (known after apply)
      ~ command           = [
          - "nginx",
          - "-g",
          - "daemon off;",
        ] -> (known after apply)
      + container_logs    = (known after apply)
      - cpu_shares        = 0 -> null
      - dns               = [] -> null
      - dns_opts          = [] -> null
      - dns_search        = [] -> null
      ~ entrypoint        = [
          - "/docker-entrypoint.sh",
        ] -> (known after apply)
      ~ env               = [] -> (known after apply)
      + exit_code         = (known after apply)
      ~ gateway           = "172.17.0.1" -> (known after apply)
      - group_add         = [] -> null
      ~ hostname          = "c5836f583f57" -> (known after apply)
      ~ id                = "c5836f583f579395d83712164b3134f96e6fa529afd8f648226dbb20729f245d" -> (known after apply)
      ~ image             = "sha256:e4720093a3c1381245b53a5a51b417963b3c4472d3f47fc301930a4f3b17666a" -> "nginx:latest" # forces replacement
      ~ init              = false -> (known after apply)
      ~ ip_address        = "172.17.0.2" -> (known after apply)
      ~ ip_prefix_length  = 16 -> (known after apply)
      ~ ipc_mode          = "private" -> (known after apply)
      - links             = [] -> null
      ~ log_driver        = "json-file" -> (known after apply)
      - log_opts          = {} -> null
```

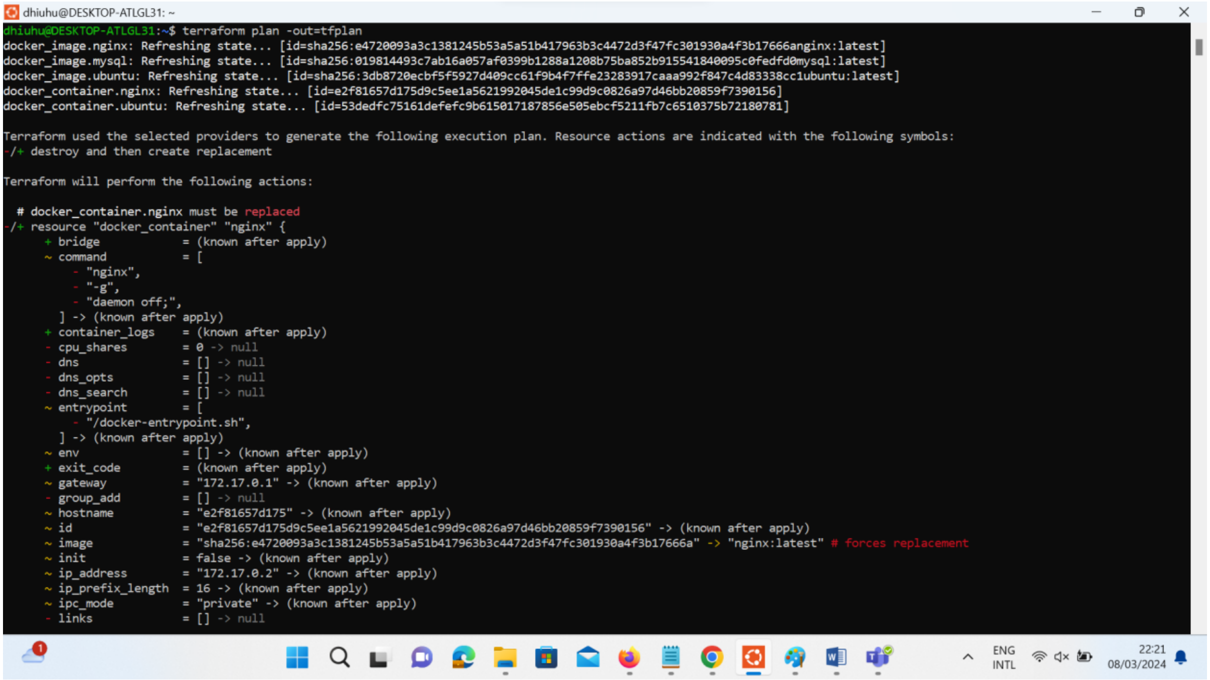


```
              - ip_prefix_length          = 16
              - ipv6_gateway              = ""
              - network_name              = "bridge"
            },
        ] -> (known after apply)
      - network_mode      = "default" -> null
      - privileged        = false -> null
      - publish_all_ports = false -> null
      ~ security_opts     = [] -> (known after apply)
      ~ shm_size          = 64 -> (known after apply)
      - storage_opts      = {} -> null
      - sysctls           = {} -> null
      - tmpfs             = {} -> null
        # (11 unchanged attributes hidden)
    }

Plan: 2 to add, 0 to change, 2 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_container.ubuntu: Destroying... [id=ef3c9eb14f6da073c9db87730882c633b2867cd24535ea85a60d34742ac27219]
docker_container.nginx: Destroying... [id=c5836f583f579395d83712164b3134f96e6fa529afd8f648226dbb20729f245d]
docker_container.ubuntu: Destruction complete after 0s
docker_container.ubuntu: Creating...
docker_container.nginx: Destruction complete after 0s
docker_container.nginx: Creating...
docker_container.ubuntu: Creation complete after 1s [id=f4ffb0bfe853f0f8ff37263fd16826d1812052b5e2be89747a8ae93d4677c47b]
docker_container.nginx: Creation complete after 1s [id=d7e10871508648a8ed1d15802294824327a253dc96788dbfe9e0c6d759b15893]

Apply complete! Resources: 2 added, 0 changed, 2 destroyed.
dhiuhu@DESKTOP-ATLGL31:~$
```

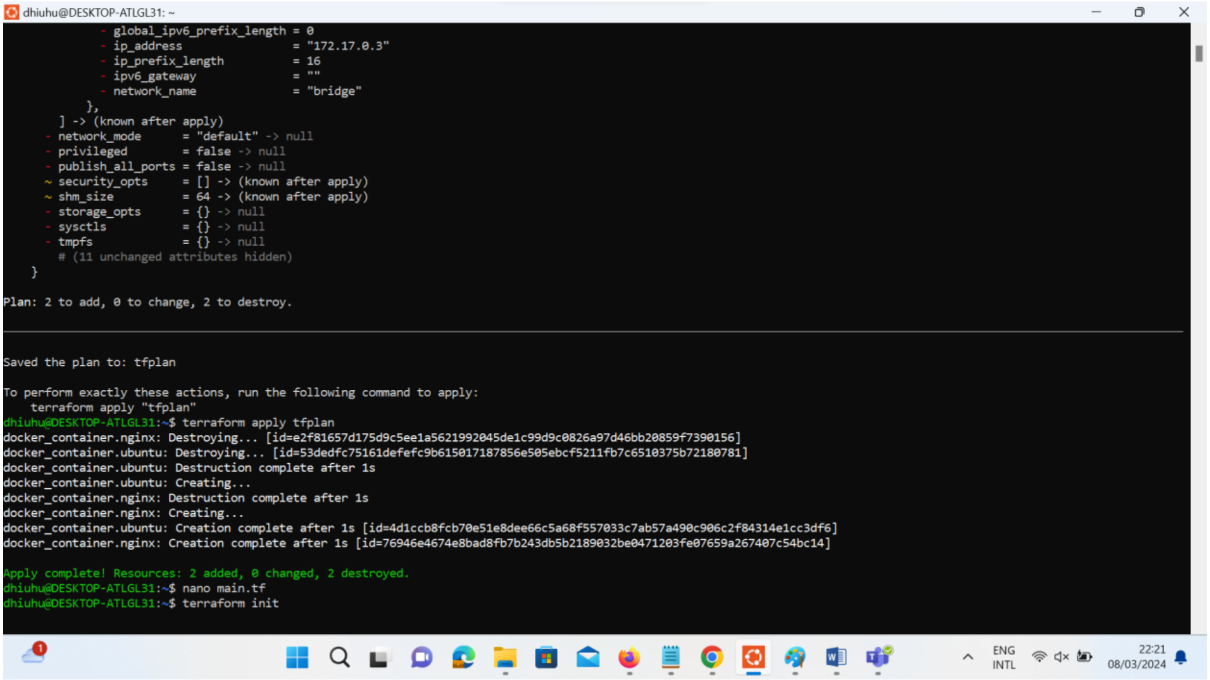


```
dhiuhu@DESKTOP-ATLGL31:~$ docker ps
CONTAINER ID   IMAGE
                COMMAND                  CREATED              STATUS               PORTS
                                        NAMES
09eb92370ef0   nginx:latest
                "/docker-entrypoint.…"   About a minute ago   Up About a minute   0.0.0.0:8081->80/tcp
                                        nginx
206a78eadeca   ubuntu:latest
                "/bin/bash -c 'tail …"   About a minute ago   Up About a minute
                                        ubuntu
e9ecd9ebbb48   dev-peer0.org2.example.com-secured_1.0.1-843dc47073bc9ebd5af86d7f422fb794fc450d737c1fa4a1cb2b9cb2c4983d6b-cdfa6a2036afcda5be33a3f61997349ace01a5365ec66f47b99
f9559202eb64b   "chaincode -peer.add…"   22 hours ago         Up 22 hours
                                        dev-peer0.org2.example.com-secured_1.0.1-843dc47073bc9ebd5af86d7f422fb794fc450d737c1fa4a1cb2b9cb2c4983d6b
2240e0b8b0af   dev-peer0.org1.example.com-secured_1.0.1-843dc47073bc9ebd5af86d7f422fb794fc450d737c1fa4a1cb2b9cb2c4983d6b-76cd426ef69ed63b4d63ba3d23c81abef422c4d8acbd6ef151b
0411017491690   "chaincode -peer.add…"   22 hours ago         Up 22 hours
                                        dev-peer0.org1.example.com-secured_1.0.1-843dc47073bc9ebd5af86d7f422fb794fc450d737c1fa4a1cb2b9cb2c4983d6b
5ddf8eb21e60   hyperledger/fabric-tools:latest
                "/bin/bash"              22 hours ago         Up 22 hours
                                        cli
720d7d26782e   hyperledger/fabric-peer:latest
                "peer node start"        22 hours ago         Up 22 hours         0.0.0.0:9051->9051/tcp, :::9051->9051/tcp, 7051/tcp, 0.0.0.0:9445->9445/tcp, :::9445->9445
/tcp                                    peer0.org2.example.com
b5c9c9f36fa1   hyperledger/fabric-peer:latest
                "peer node start"        22 hours ago         Up 22 hours         0.0.0.0:7051->7051/tcp, :::7051->7051/tcp, 0.0.0.0:9444->9444/tcp, :::9444->9444/tcp
                                        peer0.org1.example.com
40070fd601e1   hyperledger/fabric-orderer:2.5.4
                "orderer"                22 hours ago         Up 22 hours         0.0.0.0:7050->7050/tcp, :::7050->7050/tcp, 0.0.0.0:7053->7053/tcp, :::7053->7053/tcp, 0.0.
0.0:9443->9443/tcp, :::9443->9443/tcp   orderer.example.com
a0e1787e6cd0   hyperledger/fabric-ca:1.5.7
                "sh -c 'fabric-ca-se…"   22 hours ago         Up 22 hours         0.0.0.0:9054->9054/tcp, :::9054->9054/tcp, 7054/tcp, 0.0.0.0:19054->19054/tcp, :::19054->1
9054/tcp                                ca_orderer
1be82f1fa291   hyperledger/fabric-ca:1.5.7
                "sh -c 'fabric-ca-se…"   22 hours ago         Up 22 hours         0.0.0.0:8054->8054/tcp, :::8054->8054/tcp, 7054/tcp, 0.0.0.0:18054->18054/tcp, :::18054->1
8054/tcp                                ca_org2
fe3b5d7a1718   hyperledger/fabric-ca:1.5.7
                "sh -c 'fabric-ca-se…"   22 hours ago         Up 22 hours         0.0.0.0:7054->7054/tcp, :::7054->7054/tcp, 0.0.0.0:17054->17054/tcp, :::17054->17054/tcp
                                        ca_org1
dhiuhu@DESKTOP-ATLGL31:~$ 
```

Change infrastructure (your choice of what change it is).

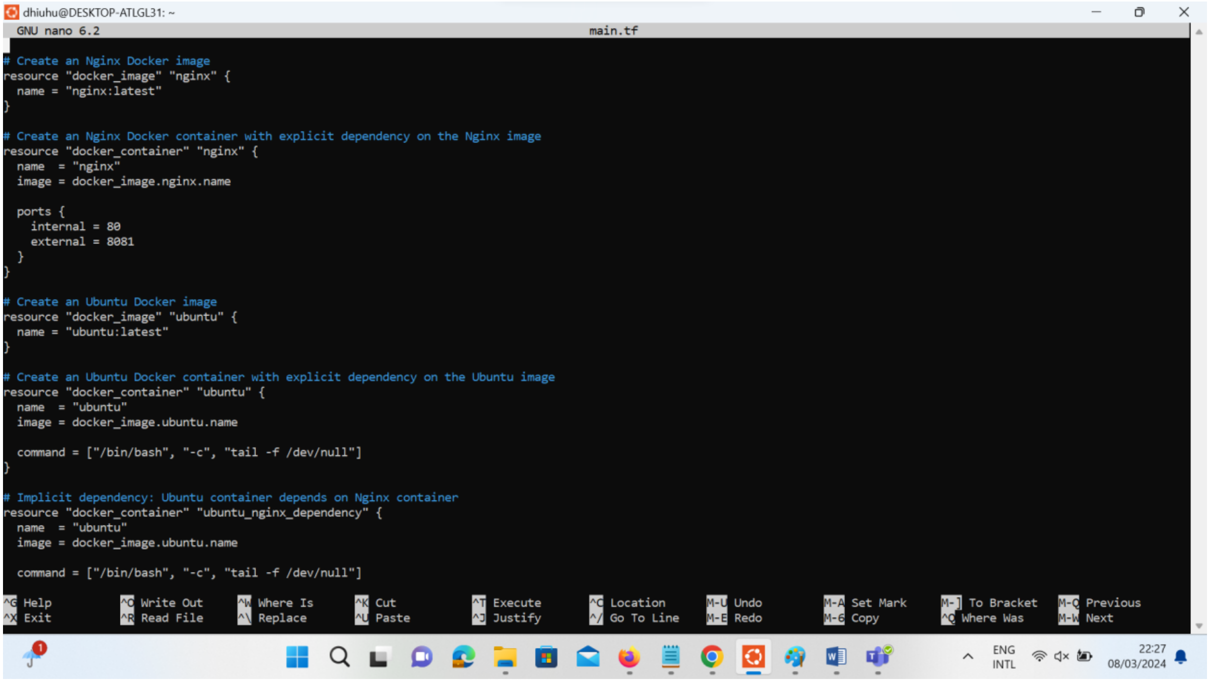


```
dhiuhu@DESKTOP-ATLGL31:~$ terraform plan -out=tfplan
docker_image.nginx: Refreshing state... [id=sha256:e4720093a3c1381245b53a5a51b417963b3c4472d3f47fc301930a4f3b17666anginx:latest]
docker_image.mysql: Refreshing state... [id=sha256:019814493c7ab16a057af0399b1288a1208b75ba852b915541840095c0fedfd0mysql:latest]
docker_image.ubuntu: Refreshing state... [id=sha256:3db8720ecbf5f5927d409cc61f9b4f7ffe23283917caaa992f847c4d83338cc1ubuntu:latest]
docker_container.nginx: Refreshing state... [id=e2f81657d175d9c5ee1a5621992045de1c99d9c0826a97d46bb20859f7390156]
docker_container.ubuntu: Refreshing state... [id=53dedfc75161defefc9b615017187856e505ebcf5211fb7c6510375b72180781]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
-/+ destroy and then create replacement

Terraform will perform the following actions:

  # docker_container.nginx must be replaced
-/+ resource "docker_container" "nginx" {
      + bridge            = (known after apply)
      ~ command           = [
          - "nginx",
          - "-g",
          - "daemon off;",
        ] -> (known after apply)
      + container_logs    = (known after apply)
      - cpu_shares        = 0 -> null
      - dns               = [] -> null
      - dns_opts          = [] -> null
      - dns_search        = [] -> null
      ~ entrypoint        = [
          - "/docker-entrypoint.sh",
        ] -> (known after apply)
      ~ env               = [] -> (known after apply)
      + exit_code         = (known after apply)
      ~ gateway           = "172.17.0.1" -> (known after apply)
      - group_add         = [] -> null
      ~ hostname          = "e2f81657d175" -> (known after apply)
      ~ id                = "e2f81657d175d9c5ee1a5621992045de1c99d9c0826a97d46bb20859f7390156" -> (known after apply)
      ~ image             = "sha256:e4720093a3c1381245b53a5a51b417963b3c4472d3f47fc301930a4f3b17666a" -> "nginx:latest" # forces replacement
      ~ init              = false -> (known after apply)
      ~ ip_address        = "172.17.0.2" -> (known after apply)
      ~ ip_prefix_length  = 16 -> (known after apply)
      ~ ipc_mode          = "private" -> (known after apply)
      - links             = [] -> null
```

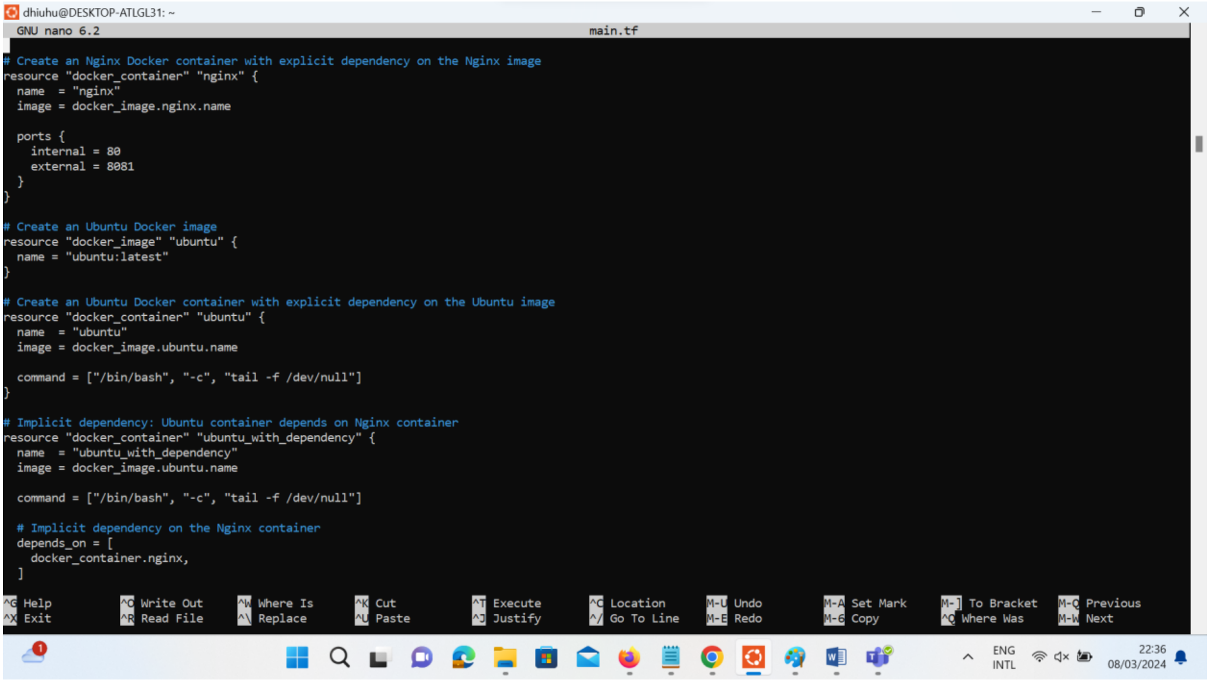


```
              - global_ipv6_prefix_length = 0
              - ip_address                = "172.17.0.3"
              - ip_prefix_length          = 16
              - ipv6_gateway              = ""
              - network_name              = "bridge"
            },
        ] -> (known after apply)
      - network_mode      = "default" -> null
      - privileged        = false -> null
      - publish_all_ports = false -> null
      ~ security_opts     = [] -> (known after apply)
      ~ shm_size          = 64 -> (known after apply)
      - storage_opts      = {} -> null
      - sysctls           = {} -> null
      - tmpfs             = {} -> null
        # (11 unchanged attributes hidden)
    }

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

─────────────────────────────────────────────────────────────────────────────

Saved the plan to: tfplan

To perform exactly these actions, run the following command to apply:
    terraform apply "tfplan"
dhiuhu@DESKTOP-ATLGL31:~$ terraform apply tfplan
docker_container.nginx: Destroying... [id=e2f81657d175d9c5ee1a5621992045de1c99d9c0826a97d46bb20859f7390156]
docker_container.ubuntu: Destroying... [id=53dedfc75161defefc9b615017187856e505ebcf5211fb7c6510375b72180781]
docker_container.ubuntu: Destruction complete after 1s
docker_container.ubuntu: Creating...
docker_container.nginx: Destruction complete after 1s
docker_container.nginx: Creating...
docker_container.ubuntu: Creation complete after 1s [id=4d1ccb8fcb70e51e8dee66c5a68f557033c7ab57a490c906c2f84314e1cc3df6]
docker_container.nginx: Creation complete after 1s [id=76946e4674e8bad8fb7b243db5b2189032be0471203fe07659a267407c54bc14]

Apply complete! Resources: 2 added, 0 changed, 2 destroyed.
dhiuhu@DESKTOP-ATLGL31:~$ nano main.tf
dhiuhu@DESKTOP-ATLGL31:~$ terraform init
```

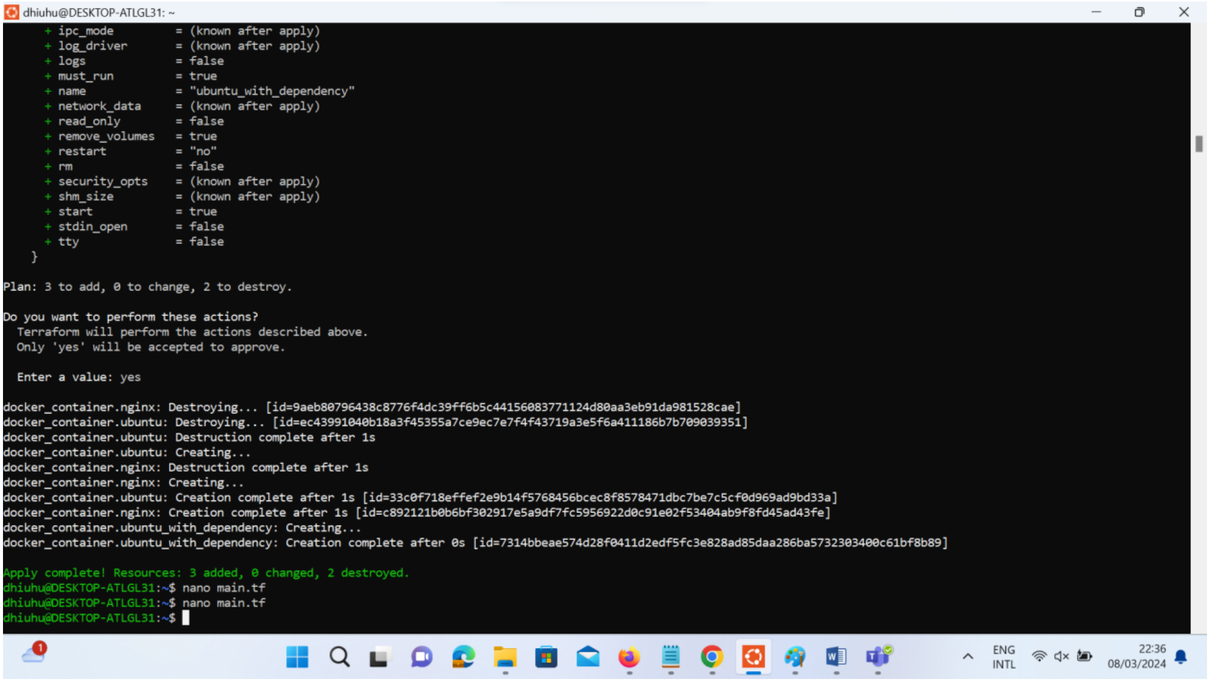
Create resource with dependencies (implicit and explicit).



```
# Create an Nginx Docker image
resource "docker_image" "nginx" {
  name = "nginx:latest"
}

# Create an Nginx Docker container with explicit dependency on the Nginx image
resource "docker_container" "nginx" {
  name  = "nginx"
  image = docker_image.nginx.name

  ports {
    internal = 80
    external = 8081
  }
}

# Create an Ubuntu Docker image
resource "docker_image" "ubuntu" {
  name = "ubuntu:latest"
}

# Create an Ubuntu Docker container with explicit dependency on the Ubuntu image
resource "docker_container" "ubuntu" {
  name  = "ubuntu"
  image = docker_image.ubuntu.name

  command = ["/bin/bash", "-c", "tail -f /dev/null"]
}

# Implicit dependency: Ubuntu container depends on Nginx container
resource "docker_container" "ubuntu_nginx_dependency" {
  name  = "ubuntu"
  image = docker_image.ubuntu.name

  command = ["/bin/bash", "-c", "tail -f /dev/null"]
```



```
# Create an Nginx Docker container with explicit dependency on the Nginx image
resource "docker_container" "nginx" {
  name  = "nginx"
  image = docker_image.nginx.name

  ports {
    internal = 80
    external = 8081
  }
}

# Create an Ubuntu Docker image
resource "docker_image" "ubuntu" {
  name = "ubuntu:latest"
}

# Create an Ubuntu Docker container with explicit dependency on the Ubuntu image
resource "docker_container" "ubuntu" {
  name  = "ubuntu"
  image = docker_image.ubuntu.name

  command = ["/bin/bash", "-c", "tail -f /dev/null"]
}

# Implicit dependency: Ubuntu container depends on Nginx container
resource "docker_container" "ubuntu_with_dependency" {
  name  = "ubuntu_with_dependency"
  image = docker_image.ubuntu.name

  command = ["/bin/bash", "-c", "tail -f /dev/null"]

  # Implicit dependency on the Nginx container
  depends_on = [
    docker_container.nginx,
  ]
```



```
      + ipc_mode          = (known after apply)
      + log_driver        = (known after apply)
      + logs              = false
      + must_run          = true
      + name              = "ubuntu_with_dependency"
      + 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
    }

Plan: 3 to add, 0 to change, 2 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_container.nginx: Destroying... [id=9aeb80796438c8776f4dc39ff6b5c44156083771124d80aa3eb91da981528cae]
docker_container.ubuntu: Destroying... [id=ec43991040b18a3f45355a7ce9ec7e7f4f43719a3e5f6a411186b7b709039351]
docker_container.ubuntu: Destruction complete after 1s
docker_container.ubuntu: Creating...
docker_container.nginx: Destruction complete after 1s
docker_container.nginx: Creating...
docker_container.ubuntu: Creation complete after 1s [id=33c0f718effef2e9b14f5768456bcec8f8578471dbc7be7c5cf0d969ad9bd33a]
docker_container.nginx: Creation complete after 1s [id=c892121b0b6bf302917e5a9df7fc5956922d0c91e02f53404ab9f8fd45ad43fe]
docker_container.ubuntu_with_dependency: Creating...
docker_container.ubuntu_with_dependency: Creation complete after 0s [id=7314bbeae574d28f0411d2edf5fc3e828ad85daa286ba5732303400c61bf8b89]

Apply complete! Resources: 3 added, 0 changed, 2 destroyed.
dhiuhu@DESKTOP-ATLGL31:~$ nano main.tf
dhiuhu@DESKTOP-ATLGL31:~$ nano main.tf
dhiuhu@DESKTOP-ATLGL31:~$ 
```