

## **LAB 4 - Terraform by Athika Fatima (101502209)**

```
Last login: Sat Mar  9 00:09:37 on ttys002
You have new mail.
/Users/athikafatima/.zshrc:5: command not found: ng
|athikafatima@Athikas-MBP ~ % cd Desktop/lab4
|athikafatima@Athikas-MBP lab4 % terraform init
Terraform initialized in an empty directory!

The directory has no Terraform configuration files. You may begin working
with Terraform immediately by creating Terraform configuration files.
|athikafatima@Athikas-MBP lab4 % ls
main.tf
|athikafatima@Athikas-MBP lab4 % terraform apply

| Error: Inconsistent dependency lock file

| The following dependency selections recorded in the lock file are inconsistent with the current configuration:
| - provider registry.terraform.io/kreuzwerker/docker: required by this configuration but no version is selected

| To make the initial dependency selections that will initialize the dependency lock file, run:
|   terraform init
|
|athikafatima@Athikas-MBP lab4 % 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 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.
|athikafatima@Athikas-MBP lab4 % terraform apply

| Error: Error pinging Docker server: Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?

|   with provider["registry.terraform.io/kreuzwerker/docker"],
|   on main.tf line 10, in provider "docker":
|   10: provider "docker" {}

|
|athikafatima@Athikas-MBP lab4 % 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:
```

```

|
|
|athikafatima@Athikas-MBP lab4 % 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
    + must_run                            = true
    + name                                 = "tutorial"
    + network_data                        = (known after apply)
    + 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

    + 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)
    + 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.

```

```

+ 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 = "tutorial"
+ network_data = (known after apply)
+ 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

+ 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)
  + 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: Creation complete after 10s [id=sha256:e4720093a3c1381245b53a5a51b417963b3c4472d3f47fc301930a4f3b17666nginx]
docker_container.nginx: Creating...
docker_container.nginx: Creation complete after 2s [id=c4623109a263aac1eb9197d5a2774111f90db894a732fe82c0ed379d2be70a3d]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
athikafatima@Athikas-MBP lab4 %

```

```

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
athikafatima@Athikas-MBP lab4 % terraform plan
docker_image.nginx: Refreshing state... [id=sha256:e4720093a3c1381245b53a5a51b417963b3c4472d3f47fc301930a4f3b17666anginx]
docker_container.nginx: Refreshing state... [id=c4623109a263aac1eb9197d5a2774111f90db894a732fe82c0ed379d2be70a3d]

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)
    - group_add                           = □ -> null
    ~ hostname                           = "c4623109a263" -> (known after apply)
    ~ id                                 = "c4623109a263aac1eb9197d5a2774111f90db894a732fe82c0ed379d2be70a3d" -> (known after apply)
    ~ init                               = false -> (known after apply)
    ~ ipc_mode                            = "private" -> (known after apply)
    ~ log_driver                          = "json-file" -> (known after apply)
    - log_opts                            = {} -> null
    - max_retry_count                    = 0 -> null
    - memory                             = 0 -> null
    - memory_swap                        = 0 -> null
    ~ 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                   = ""
            - mac_address                     = "02:42:ac:11:00:02"
            - network_name                   = "bridge"
        },
    ] -> (known after apply)
    - network_mode                       = "default" -> null
    - privileged                          = false -> null
    - publish_all_ports                  = false -> null
    ~ runtime                            = "runc" -> (known after apply)
    ~ security_opts                      = □ -> (known after apply)
    ~ shm_size                           = 64 -> (known after apply)
    ~ stop_signal                         = "SIGQUIT" -> (known after apply)
    ~ stop_timeout                        = 0 -> (known after apply)
    - storage_opts                       = {} -> null
    - sysctls                            = {} -> null
    - tmpfs                             = {} -> null
}

```

```

~ command
  - "nginx",
  - "-g",
  - "daemon off;";
] -> (known after apply)
+ container_logs
- cpu_shares
- dns
- dns_opts
- dns_search
~ entrypoint
  - "/docker-entrypoint.sh",
] -> (known after apply)
~ env
+ exit_code
- group_add
~ hostname
~ id
~ init
~ ipc_mode
~ log_driver
- log_opts
- max_retry_count
- memory
- memory_swap
  name
~ network_data
  - {
    - gateway
    - global_ipv6_address
    - global_ipv6_prefix_length
    - ip_address
    - ip_prefix_length
    - ipv6_gateway
    - mac_address
    - network_name
  },
] -> (known after apply)
- network_mode
- privileged
- publish_all_ports
~ runtime
~ security_opts
~ shm_size
~ stop_signal
~ stop_timeout
- storage_opts
- sysctls
- tmpfs
# (14 unchanged attributes hidden)

~ ports {
  ~ external = 8000 -> 8001 # forces replacement
  # (3 unchanged attributes hidden)
}
}

Plan: 1 to add, 0 to change, 1 to destroy.

```

---

Note: You didn't use the `-out` option to save this plan, so Terraform can't guarantee to take exactly these actions if you run `"terraform apply"` now.  
athikafatima@Athikas-MBP lab4 %

```

athikafatima@Athikas-MBP lab4 % terraform apply
docker_image.nginx: Refreshing state... [id=sha256:e4720093a3c1381245b53a5d51b417963b3c4472d3f47fc301930a4f3b17666anginx]
docker_container.nginx: Refreshing state... [id=c4623109a263aac1eb9197d5a2774111f90db894a732fe82c0ed379d2be70a3d]

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)
    - group_add                            = [] -> null
    ~ hostname                             = "c4623109a263" -> (known after apply)
    ~ id                                    = "c4623109a263aac1eb9197d5a2774111f90db894a732fe82c0ed379d2be70a3d" -> (known after apply)
    ~ init                                  = false -> (known after apply)
    ~ ipc_mode                             = "private" -> (known after apply)
    ~ log_driver                           = "json-file" -> (known after apply)
    - log_opts                             = {} -> null
    - max_retry_count                     = 0 -> null
    - memory                               = 0 -> null
    - memory_swap                          = 0 -> null
    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                      = ""
            - mac_address                       = "02:42:ac:11:00:02"
            - network_name                      = "bridge"
        },
    ],
}

```

```

- memory = 0 -> null
- memory_swap = 0 -> null
  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 = ""
      - mac_address = "02:42:ac:11:00:02"
      - network_name = "bridge"
    },
  ] -> (known after apply)
- network_mode = "default" -> null
- privileged = false -> null
- publish_all_ports = false -> null
~ runtime = "runc" -> (known after apply)
~ security_opts = [] -> (known after apply)
~ shm_size = 64 -> (known after apply)
~ stop_signal = "SIGQUIT" -> (known after apply)
~ stop_timeout = 0 -> (known after apply)
- storage_opts = {} -> null
- sysctls = {} -> null
- tmpfs = {} -> null
# (14 unchanged attributes hidden)

~ ports {
  ~ external = 8000 -> 8001 # forces replacement
  # (3 unchanged attributes hidden)
}
}

```

**Plan:** 1 to add, 0 to change, 1 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=c4623109a263aac1eb9197d5a2774111f90db894a732fe82c0ed379d2be70a3d]
docker_container.nginx: Destruction complete after 0s
docker_container.nginx: Creating...
docker_container.nginx: Creation complete after 1s [id=40afb450e89228b2a5083bfffcc48c7fa64914e991537585e3c2e4c97cc65458f8]

```

**Apply complete! Resources: 1 added, 0 changed, 1 destroyed.**  
athikafatima@Athikas-MBP lab4 %

```

athikafatima@Athikas-MBP lab4 % terraform plan
docker_image.nginx: Refreshing state... [id=sha256:e4720093a3c1381245b53a5a51b417963b3c4472d3f47fc301930a4f3b17666anginx]
docker_container.nginx: Refreshing state... [id=40afb450e89228b2a5083bfc48c7fa64914e991537585e3c2e4c97cc65458f8]

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

Terraform will perform the following actions:

# docker_container.nginx must be replaced
-/+ resource "docker_container" "nginx" {
    + bridge
    ~ command
        - "nginx",
        - "-g",
        - "daemon off;",
    ] -> (known after apply)
    + container_logs
    - cpu_shares
    - dns
    - dns_opts
    - dns_search
    ~ entrypoint
        - "/docker-entrypoint.sh",
    ] -> (known after apply)
    ~ env
    + exit_code
    - group_add
    ~ hostname
    ~ id
    ~ image
    ~ init
    ~ ipc_mode
    ~ log_driver
    - log_opts
    - max_retry_count
    - memory
    - memory_swap
    name
    ~ network_data
        - {
            - gateway
            - global_ipv6_address
            - global_ipv6_prefix_length = 0
            - ip_address
            - ip_prefix_length
            - ipv6_gateway
            - mac_address
            - network_name
        },
    ] -> (known after apply)
    - network_mode
    - privileged
    - publish_all_ports
    ~ runtime
}

```

```

    - "/docker-entrypoint.sh",
    ] -> (known after apply)
~ env
+ exit_code
- group_add
~ hostname
~ id
~ image
~ init
~ ipc_mode
~ log_driver
- log_opts
- max_retry_count
- memory
- memory_swap
~ name
~ 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   = ""
    - mac_address    = "02:42:ac:11:00:02"
    - network_name   = "bridge"
},
] -> (known after apply)
~ network_mode
- privileged
- publish_all_ports
~ runtime
~ security_opts
~ shm_size
~ stop_signal
~ stop_timeout
~ storage_opts
~ sysctls
- tmpfs
# (13 unchanged attributes hidden)

~ ports {
    ~ external = 8001 -> 8080 # forces replacement
    # (3 unchanged attributes hidden)
}
}

# docker_image.nginx will be destroyed
# (because docker_image.nginx is not in configuration)
resource "docker_image" "nginx" {
    - id          = "sha256:e4720093a3c1381245b53a51b417963b3c4472d3f47fc301930a4f3b17666a"
    - image_id    = "sha256:e4720093a3c1381245b53a51b417963b3c4472d3f47fc301930a4f3b17666a"
    - keep_locally = false -> null
    - name        = "nginx" -> null
    - repo_digest = "nginx@sha256:c26ae7472d624ba1fafd296e73cecc4f93f853088e6a9c13c0d52f6cd5865107" -> null
}

```

**Plan:** 1 to add, 0 to change, 2 to destroy.

---

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.  
athikafatima@Athikas-MBP lab4 %

```

@thikafatima@Athikas-MBP lab4 % terraform apply
docker_image.nginx: Refreshing state... [id=sha256:e4720093a3c1381245b53a5a51b417963b3c4472d3f47fc301930a4f3b17666nginx]
docker_container.nginx: Refreshing state... [id=40afb450e89228b2a5083bfc48c7fa64914e991537585e3c2e4c97cc65458f8]

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

Terraform will perform the following actions:

# docker_container.nginx must be replaced
-/+ resource "docker_container" "nginx" {
    + bridge
        ~ command = (known after apply)
        ~ [
            - "nginx",
            - "-g",
            - "daemon off;";
        ] -> (known after apply)
    + container_logs
    - cpu_shares
    - dns
    - dns_opts
    - dns_search
    ~ entrypoint
        - "/docker-entrypoint.sh",
        ] -> (known after apply)
    ~ env
    + exit_code
    ~ group_add
    ~ hostname
    ~ id
    ~ image
    ~ init
    ~ ipc_mode
    ~ log_driver
    - log_opts
    - max_retry_count
    - memory
    - memory_swap
    name
    ~ network_data
        - {
            - gateway = "172.17.0.1"
            - global_ipv6_address =
            - global_ipv6_prefix_length =
            - ip_address = "172.17.0.2"
            - ip_prefix_length = 16
            - ipv6_gateway =
            - mac_address = "02:42:ac:11:00:02"
        }
}

```

```

- max_retry_count = 0 -> null
- memory = 0 -> null
- memory_swap = 0 -> null
  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 = ""
    - mac_address = "02:42:ac:11:00:02"
    - network_name = "bridge"
  },
  ] -> (known after apply)
- network_mode = "default" -> null
- privileged = false -> null
- publish_all_ports = false -> null
~ runtime = "runc" -> (known after apply)
~ security_opts = [] -> (known after apply)
~ shm_size = 64 -> (known after apply)
~ stop_signal = "SIGQUIT" -> (known after apply)
~ stop_timeout = 0 -> (known after apply)
- storage_opts = {} -> null
- sysctls = {} -> null
- tmpfs = {} -> null
# (13 unchanged attributes hidden)

~ ports {
  ~ external = 8001 -> 8080 # forces replacement
  # (3 unchanged attributes hidden)
}
}

# docker_image.nginx will be destroyed
# (because docker_image.nginx is not in configuration)
resource "docker_image" "nginx" {
  - id = "sha256:e4720093a3c1381245b53a5a51b417963b3c4472d3f47fc301930a4f3b17666anginx" -> null
  - image_id = "sha256:e4720093a3c1381245b53a5a51b417963b3c4472d3f47fc301930a4f3b17666a" -> null
  - keep_locally = false -> null
  - name = "nginx" -> null
  - repo_digest = "nginx@sha256:c26ae7472d624ba1fafd296e73cecc4f93f853088e6a9c13c0d52f6ca5865107" -> null
}

```

**Plan:** 1 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=40afb450e89228b2a5083bfc48c7fa64914e991537585e3c2e4c97cc65458f8]
docker_container.nginx: Destruction complete after 0s
docker_image.nginx: Destroying... [id=sha256:4720093a3c1381245b53a5a51b417963b3c4472d3f47fc301930a4f3b17666anginx]
docker_image.nginx: Destruction complete after 1s
docker_container.nginx: Creating...
docker_container.nginx: Still creating... [10s elapsed]
docker_container.nginx: Creation complete after 11s [id=10f48e90268b15eb933bech9bc2149f2d2a5f4914eaed9653b427e63912b8235]

```

**Apply complete!** Resources: 1 added, 0 changed, 2 destroyed.  
athikafatima@Athikas-MBP lab4 %

```

athikafatima@Athikas-MBP lab4 % terraform destroy
docker_container.nginx: Refreshing state... [id=10f48e90268b15eb933becb9bc2149f2d2a5f4914eaed9653b427e63912b8235]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
- destroy

Terraform will perform the following actions:

# docker_container.nginx will be destroyed
- resource "docker_container" "nginx" {
    - attach                                = false -> null
    - command                               = [
        - "nginx",
        - "-g",
        - "daemon off;";
    ] -> null
    - container_read_refresh_timeout_milliseconds = 15000 -> null
    - cpu_shares                            = 0 -> null
    - dns                                    = [] -> null
    - dns_opts                             = [] -> null
    - dns_search                           = [] -> null
    - entrypoint                           = [
        - "/dockerc-entrypoint.sh",
    ] -> null
    - env                                    = [] -> null
    - group_add                            = [] -> null
    - hostname                             = "10f48e90268b" -> null
    - id                                     = "10f48e90268b15eb933becb9bc2149f2d2a5f4914eaed9653b427e63912b8235" -> null
    - image                                  = "sha256:e4720093a3c1381245b53a5a51b417963b3c4472d3f47fc301930a4f3b17666a" -> null
    - init                                    = false -> null
    - ipc_mode                             = "private" -> null
    - log_driver                           = "json-file" -> null
    - log_opts                             = {} -> null
    - logs                                    = false -> null
    - max_retry_count                      = 0 -> null
    - memory                                 = 0 -> null
    - memory_swap                           = 0 -> null
    - must_run                             = true -> null
    - name                                    = "tutorial" -> null
    - 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                      = ""
            - mac_address                        = "02:42:ac:11:00:02"
            - network_name                      = "bridge"
        },
    ] -> null
    - network_mode                          = "default" -> null
}

```

```

- ipc_mode = "private" -> null
- log_driver = "json-file" -> null
- log_opts = {} -> null
- logs = false -> null
- max_retry_count = 0 -> null
- memory = 0 -> null
- memory_swap = 0 -> null
- must_run = true -> null
- name = "tutorial" -> null
- 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 = ""
        - mac_address = "02:42:ac:11:00:02"
        - network_name = "bridge"
    },
],
] -> null
- network_mode = "default" -> null
- privileged = false -> null
- publish_all_ports = false -> null
- read_only = false -> null
- remove_volumes = true -> null
- restart = "no" -> null
- rm = false -> null
- runtime = "runc" -> null
- security_opts = [] -> null
- shm_size = 64 -> null
- start = true -> null
- stdin_open = false -> null
- stop_signal = "SIGQUIT" -> null
- stop_timeout = 0 -> null
- storage_opts = {} -> null
- sysctls = {} -> null
- tmpfs = {} -> null
- tty = false -> null
- wait = false -> null
- wait_timeout = 60 -> null

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

```

**Plan:** 0 to add, 0 to change, 1 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=10f48e90268b15eb933becb9bc2149f2d2a5f4914eaed9653b427e63912b8235]
docker_container.nginx: Destruction complete after 1s
```

**Destroy complete! Resources: 1 destroyed.**

athikafatima@Athikas-MBP lab4 %

```
[athikafatima@Athikas-MBP lab4 % terraform plan

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
    + must_run                            = true
    + name                                = "tutorial"
    + network_data                        = (known after apply)
    + 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

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

# null_resource.example will be created
+ resource "null_resource" "example" {
    + id          = (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                          = "tutorial"
+ network_data                  = (known after apply)
+ 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                   = false
+ tty                           = false
+ wait                          = false
+ wait_timeout                  = 60

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

# null_resource.example will be created
+ resource "null_resource" "example" {
  + id      = (known after apply)
  + triggers = {
    + "container_id" = (known after apply)
  }
}

Plan: 3 to add, 0 to change, 0 to destroy.

```

---

Note: You didn't use the `-out` option to save this plan, so Terraform can't guarantee to take exactly these actions if you run `"terraform apply"` now.

athikafatima@Athikas-MBP lab4 % █

```

}

Plan: 3 to add, 0 to change, 0 to destroy.

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
athikafatima@Athikas-MBP:~/lab4 % 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_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 = "tutorial"
    + network_data = (known after apply)
    + read_only = false
    + remove_volumes = true
    + restart = "no"
    + rm = false
    + runtime = (known after apply)
    + security_opts = (known after apply)
    + shm_size = true
    + start = false
    + stdin_open = (known after apply)
    + stop_signal = (known after apply)
    + stop_timeout = false
    + tty = false
    + wait = false
    + wait_timeout = 60

    + 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)
    + keep_locally = false
}

```

```

+ name = "tutorial"
+ network_data
+ read_only
+ remove_volumes
+ restart
+ rm
+ runtime
+ security_opts
+ shm_size
+ start
+ stdin_open
+ stop_signal
+ stop_timeout
+ tty
+ wait
+ wait_timeout = 60

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

# null_resource.example will be created
+ resource "null_resource" "example" {
  + id      = (known after apply)
  + triggers = {
    + "container_id" = (known after apply)
  }
}

Plan: 3 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 0s [id=sha256:e4720093a3c1381245b53a5a51b417963b3c4472d3f47fc301930a4f3b17666anginx]
docker_container.nginx: Creating...
docker_container.nginx: Creation complete after 1s [id=89d1f778a4482f6e1c126121db88d89921205073d955275289813896b0a55e28]
null_resource.example: Creating...
null_resource.example: Provisioning with 'local-exec'...
null_resource.example (local-exec): Executing: ["#!/bin/sh" "-c" "echo This resource depends on the nginx Docker container and image"]
null_resource.example (local-exec): This resource depends on the nginx Docker container and image
null_resource.example: Creation complete after 0s [id=5580821479632812673]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
athikafatima@Athikas-MBP lab4 %

```