```
Already up-to-date.
[simarchawla@Simars-MacBook-Air terraform % brew upgrade hashicorp/tap/terraform
Warning: hashicorp/tap/terraform 1.7.4 already installed
[simarchawla@Simars-MacBook-Air terraform % terraform -v
Terraform v1.7.4
on darwin_arm64
simarchawla@Simars-MacBook-Air terraform %
```

```
  name  = "tutorial"

  ports {
    internal = 80
    external = 8000
  }
}
terraform {
  required_providers {
    docker = {
      source  = "kreuzwerker/docker"
      version = "~> 3.0.1"
    }
  }
}

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 = 8000
  }
}

[simarchawla@Simars-MacBook-Air terraform % vi main.tf
[simarchawla@Simars-MacBook-Air terraform % vi main.tf
[simarchawla@Simars-MacBook-Air terraform % 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.
simarchawla@Simars-MacBook-Air terraform %
```

```
      + 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 9s [id=sha256:760b7cbba31e196288effd2af692
docker_container.nginx: Creating...
docker_container.nginx: Creation complete after 4s [id=d8bf1cc7f8438363190233825685730

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
simarchawla@Simars-MacBook-Air terraform %
```

```
simarchawla@Simars-MacBook-Air terraform % terraform plan
docker_image.nginx: Refreshing state... [id=sha256:760b7cbba31e196288effd2af6924c42637ac5e0d67db4de6309f24518844676nginx]
docker_container.nginx: Refreshing state... [id=d8bf1cc7f8438363190233825685 73c0316757f388f26860c67124266531f0a0]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following
  + create
  - destroy
-/+ 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                                    = "d8bf1cc7f843" -> (known after apply)
      ~ id                                          = "d8bf1cc7f8438363190233825685 73c0316757f388f26860c67124266531f0a0" -> (known
      ~ image                                       = "sha256:760b7cbba31e196288effd2af6924c42637ac5e0d67db4de6309f24518844676" #
      ~ 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
```

```
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 "te
[simarchawla@Simars-MacBook-Air terraform % terraform apply
docker_image.nginx: Refreshing state... [id=sha256:760b7cbba31e196288effd2af6924c42637ac5e0d67db4de6309f24518844676nginx]
docker_container.nginx: Refreshing state... [id=d8bf1cc7f8438363190233825685 73c0316757f388f26860c67124266531f0a0]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following
  + create
  - destroy
-/+ 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                                  = "d8bf1cc7f843" -> (known after apply)
      ~ id                                        = "d8bf1cc7f8438363190233825685 68573c0316757f388f26860c67124266531f0a0" -> (known
      ~ image                                     = "sha256:760b7cbba31e196288effd2af6924c42637ac5e0d67db4de6309f24518844676" #
      ~ 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)
```

```
      ~ 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                                    = "5e908e64c2b0" -> (known after apply)
      ~ id                                          = "5e908e64c2b064d52ff08a69f3f0043660e0dae792ae54ce147c43a8631f4e8c" -> (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                                        = "nginx_web_server"
      ~ 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 -> 8081 # 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.
simarchawla@Simars-MacBook-Air terraform %
```

```
      - dns_search                                = [] -> null
      ~ entrypoint                                = [
          - "/docker-entrypoint.sh",
        ] -> (known after apply)
      ~ env                                       = [] -> (known after apply)
      + exit_code                                 = (known after apply)
      - group_add                                 = [] -> null
      ~ hostname                                  = "5e908e64c2b0" -> (known after apply)
      ~ id                                        = "5e908e64c2b064d52ff08a69f3f0043660e0dae792ae54ce147c43a8631f4e8c" -> (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                                      = "nginx_web_server"
      ~ 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 -> 8081 # 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.web_server: Destroying... [id=5e908e64c2b064d52ff08a69f3f0043660e0dae792ae54ce147c43a8631f4e8c]
docker_container.web_server: Destruction complete after 0s
docker_container.web_server: Creating...
docker_container.web_server: Creation complete after 3s [id=071c100d2df49deee0d4d4d60fda33b10fa2f3a21bf8fe8f42461a6ce01c5dea]

Apply complete! Resources: 1 added, 0 changed, 1 destroyed.
simarchawla@Simars-MacBook-Air terraform %
```

```
Apply complete! Resources: 1 added, 0 changed, 1 destroyed.
simarchawla@Simars-MacBook-Air terraform % docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS         PORTS                  NAMES
071c100d2df4   760b7cbba31e   "/docker-entrypoint.…"   2 minutes ago   Up 2 minutes   0.0.0.0:8081->80/tcp   nginx_web_server
simarchawla@Simars-MacBook-Air terraform %
```

```
  # docker_image.nginx will be destroyed
  - resource "docker_image" "nginx" {
      - id           = "sha256:760b7cbba31e196288effd2af6924c42637ac5e0d67db4de6309f24518844676nginx" -> null
      - image_id     = "sha256:760b7cbba31e196288effd2af6924c42637ac5e0d67db4de6309f24518844676" -> null
      - keep_locally = false -> null
      - name         = "nginx" -> null
      - repo_digest  = "nginx@sha256:c26ae7472d624ba1fafd296e73cecc4f93f853088e6a9c13c0d52f6ca5865107" -> 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.web_server: Destroying... [id=071c100d2df49deee0d4d4d60fda33b10fa2f3a21bf8fe8f42461a6ce01c5dea]
docker_container.web_server: Destruction complete after 0s
docker_image.nginx: Destroying... [id=sha256:760b7cbba31e196288effd2af6924c42637ac5e0d67db4de6309f24518844676nginx]
docker_image.nginx: Destruction complete after 0s

Destroy complete! Resources: 2 destroyed.
simarchawla@Simars-MacBook-Air terraform %
```

```
          + "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 6s [id=sha256:760b7cbba31e196288effd2af6924c42637ac5e0d67db4de6309f24518844676nginx]
docker_container.nginx: Creating...
docker_container.nginx: Creation complete after 3s [id=28cf6925edea2a380df33039d5271ce14f28380242d4842b648f0c08dfb82218]
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=6271256328195987832]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
simarchawla@Simars-MacBook-Air terraform %
```