LAB 4

1. Install Terraform in your local system.

```
akshatsrivastava@Akshats-Air terraform % brew tap hashicorp/tap
==> Auto-updating Homebrew...
Adjust how often this is run with HOMEBREW_AUTO_UPDATE_SECS or disable with
HOMEBREW_NO_AUTO_UPDATE. Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
==> Auto-updated Homebrew!
Updated 3 taps (homebrew/services, homebrew/core and homebrew/cask).
==> New Formulae
codecov-cli             geni                    gensio                  kubelogin               poutine
==> New Casks
elgato-capture-device-utility   font-palemonasmufi-bold         font-palemonasmufi-regular      macsymbolicator
emclient@beta                   font-palemonasmufi-bolditalic   impel                           vcam
font-beiruti                    font-palemonasmufi-italic       itermai                         xnapper

You have 4 outdated formulae installed.

==> Tapping hashicorp/tap
Cloning into '/opt/homebrew/Library/Taps/hashicorp/homebrew-tap'...
remote: Enumerating objects: 4509, done.
remote: Counting objects: 100% (848/848), done.
remote: Compressing objects: 100% (291/291), done.
remote: Total 4509 (delta 660), reused 700 (delta 557), pack-reused 3661
Receiving objects: 100% (4509/4509), 826.28 KiB | 5.54 MiB/s, done.
Resolving deltas: 100% (2975/2975), done.
Tapped 2 casks and 31 formulae (92 files, 1.1MB).
akshatsrivastava@Akshats-Air terraform % brew install hashicorp/tap/terraform
==> Fetching hashicorp/tap/terraform
==> Downloading https://releases.hashicorp.com/terraform/1.8.5/terraform_1.8.5_darwin_arm64.zip
######################################################################################################################################## 100.0%
==> Installing terraform from hashicorp/tap
🍺  /opt/homebrew/Cellar/terraform/1.8.5: 5 files, 83MB, built in 4 seconds
==> Running `brew cleanup terraform`...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
akshatsrivastava@Akshats-Air terraform % brew update
==> Updating Homebrew...
Already up-to-date.
akshatsrivastava@Akshats-Air terraform % brew upgrade hashicorp/tap/terraform
Warning: hashicorp/tap/terraform 1.8.5 already installed
akshatsrivastava@Akshats-Air terraform % terraform -help
Usage: terraform [global options] <subcommand> [args]

The available commands for execution are listed below.
The primary workflow commands are given first, followed by
less common or more advanced commands.
```

```
Initializing provider plugins...
- Reusing previous version of kreuzwerker/docker from the dependency lock file
- 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 made some changes to the provider dependency selections recorded
in the .terraform.lock.hcl file. Review those changes and commit them to your
version control system if they represent changes you intended to make.

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.
```

2. Create a tf configuration for running docker resource. Apply the initial resources. Change infrastructure (your choice of what change it is). **Docker use case**

```
akshatsrivastava@Akshats-Air terraform % terraform plan -out docker
docker_image.nginx: Refreshing state... [id=sha256:e4720093a3c1381245b53a5a51b417963b3c4472d3f47fc301930a4f3b17666anginx]
docker_container.nginx: Refreshing state... [id=3d58383f891e361e3ec897fb2249967f710c6c753d374066353296d1f232972c]

Note: Objects have changed outside of Terraform

Terraform detected the following changes made outside of Terraform since the last "terraform apply" which may have affected
this plan:

  # docker_image.nginx has been deleted
  - resource "docker_image" "nginx" {
      id          = "sha256:e4720093a3c1381245b53a5a51b417963b3c4472d3f47fc301930a4f3b17666anginx"
    - image_id    = "sha256:e4720093a3c1381245b53a5a51b417963b3c4472d3f47fc301930a4f3b17666a" -> null
      name        = "nginx"
      # (2 unchanged attributes hidden)
    }


Unless you have made equivalent changes to your configuration, or ignored the relevant attributes using ignore_changes, the
following plan may include actions to undo or respond to these changes.

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

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)
      + start                                       = true
      + stdin_open                                  = false
      + stop_signal                                 = (known after apply)
      + stop_timeout                                = (known after apply)
      + tty                                         = false
      + wait                                        = false
      + wait_timeout                                = 60

      + ports {
          + external = 8123
          + 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.

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

Saved the plan to: docker

To perform exactly these actions, run the following command to apply:
    terraform apply "docker"
```

```
akshatsrivastava@Akshats-Air terraform % terraform apply
docker_image.nginx: Refreshing state... [id=sha256:e4720093a3c1381245b53a5a51b417963b3c4472d3f47fc301930a4f3b17666anginx]
docker_container.nginx: Refreshing state... [id=3d58383f891e361e3ec897fb2249967f710c6c753d374066353296d1f232972c]

Note: Objects have changed outside of Terraform

Terraform detected the following changes made outside of Terraform since the last "terraform apply" which may have affected
this plan:

  # docker_image.nginx has been deleted
  - resource "docker_image" "nginx" {
      id          = "sha256:e4720093a3c1381245b53a5a51b417963b3c4472d3f47fc301930a4f3b17666anginx"
    - image_id    = "sha256:e4720093a3c1381245b53a5a51b417963b3c4472d3f47fc301930a4f3b17666a" -> null
      name        = "nginx"
      # (2 unchanged attributes hidden)
    }


Unless you have made equivalent changes to your configuration, or ignored the relevant attributes using ignore_changes, the
following plan may include actions to undo or respond to these changes.

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


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)
      + 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 = 8123
          + 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 4s [id=sha256:11ceee7cdc57225711b8382e1965974bbb259de14a9f5f7d6b9f161ced50a10anginx
]
docker_container.nginx: Creating...
docker_container.nginx: Creation complete after 1s [id=df3c33e85adbb0c5509c314d3bcd413c8b93961ea9bf1c6bfb189df9cbb0c853]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
```

```
akshatsrivastava@Akshats-Air terraform % docker ps
CONTAINER ID   IMAGE                     COMMAND                  CREATED          STATUS          PORTS
                    NAMES
df3c33e85adb   11ceee7cdc57              "/docker-entrypoint.…"   7 seconds ago    Up 6 seconds    0.0.0.0:8123->80/tcp
```

3. Create a plan and use the plan to make changes to the resource.

```
akshatsrivastava@Akshats-Air terraform % terraform plan
docker_image.nginx: Refreshing state... [id=sha256:11ceee7cdc57225711b8382e1965974bbb259de14a9f5f7d6b9f161ced50a10anginx]
docker_container.nginx: Refreshing state... [id=1b38557ecc9a81dad43453138f74c914bce6aeb7de5c8aec4314ea0e657f6c8c]

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                                    = "1b38557ecc9a" -> (known after apply)
      ~ id                                          = "1b38557ecc9a81dad43453138f74c914bce6aeb7de5c8aec4314ea0e657f6c8c" -> (kn
own 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_prefix_length = 0
              - ip_address                = "172.17.0.2"
              - ip_prefix_length          = 16
```

```
        name                                        = "tutorial"
      ~ network_data                                = [
          - {
              - gateway                   = "172.17.0.1"
              - global_ipv6_prefix_length = 0
              - ip_address                = "172.17.0.2"
              - ip_prefix_length          = 16
              - mac_address               = "02:42:ac:11:00:02"
              - network_name              = "bridge"
                # (2 unchanged attributes hidden)
            },
        ] -> (known after apply)
      - network_mode                                = "bridge" -> null # forces replacement
      - 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
        # (20 unchanged attributes hidden)

      ~ ports {
          ~ external = 8123 -> 8000 # 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.
```

```
akshatsrivastava@Akshats-Air terraform % terraform apply
docker_image.nginx: Refreshing state... [id=sha256:11ceee7cdc57225711b8382e1965974bbb259de14a9f5f7d6b9f161ced50a10anginx]
docker_container.nginx: Refreshing state... [id=1b38557ecc9a81dad43453138f74c914bce6aeb7de5c8aec4314ea0e657f6c8c]

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                                    = "1b38557ecc9a" -> (known after apply)
      ~ id                                          = "1b38557ecc9a81dad43453138f74c914bce6aeb7de5c8aec4314ea0e657f6c8c" -> (kn
own 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_prefix_length = 0
              - ip_address                = "172.17.0.2"
```

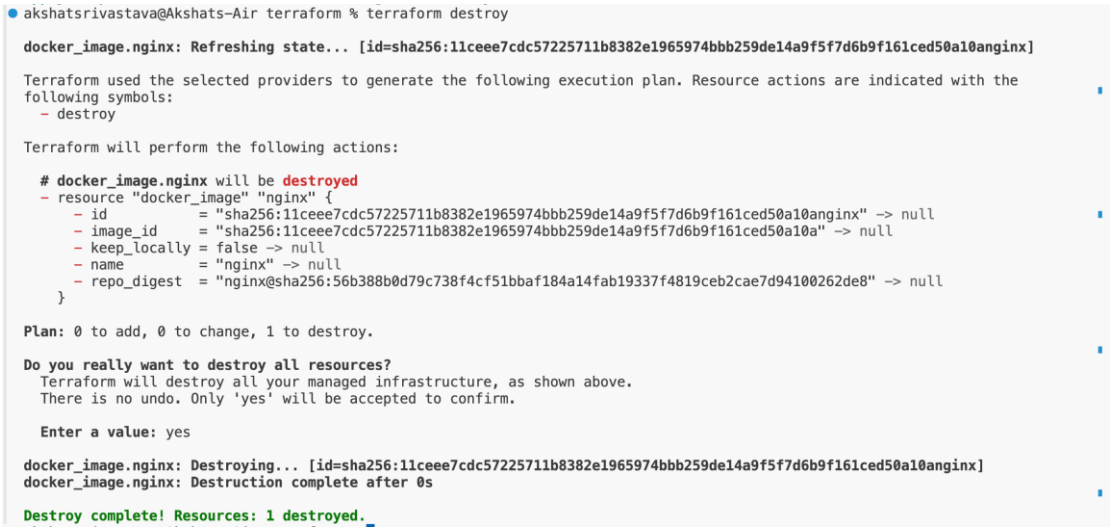
```
        name                                        = "tutorial"
      ~ network_data                                = [
          - {
              - gateway                   = "172.17.0.1"
              - global_ipv6_prefix_length = 0
              - ip_address                = "172.17.0.2"
              - ip_prefix_length          = 16
              - mac_address               = "02:42:ac:11:00:02"
              - network_name              = "bridge"
                # (2 unchanged attributes hidden)
            },
        ] -> (known after apply)
      - network_mode                                = "bridge" -> null # forces replacement
      - 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
        # (20 unchanged attributes hidden)

      ~ ports {
          ~ external = 8123 -> 8000 # 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=1b38557ecc9a81dad43453138f74c914bce6aeb7de5c8aec4314ea0e657f6c8c]
docker_container.nginx: Destruction complete after 0s
docker_container.nginx: Creating...
docker_container.nginx: Creation complete after 1s [id=0934e7114cffead03c3c54927b33496d8558a3b183ae1e3882a75409bb213c61]

Apply complete! Resources: 1 added, 0 changed, 1 destroyed.
```

4. Make destructive changes (like removing one of the docker images).

```
akshatsrivastava@Akshats-Air terraform % terraform plan

docker_image.nginx: Refreshing state... [id=sha256:11ceee7cdc57225711b8382e1965974bbb259de14a9f5f7d6b9f161ced50a10anginx]
docker_container.nginx: Refreshing state... [id=0934e7114cffead03c3c54927b33496d8558a3b183ae1e3882a75409bb213c61]

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
  # (because docker_container.nginx is not in configuration)
  - 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                                  = [
          - "/docker-entrypoint.sh",
        ] -> null
      - env                                         = [] -> null
      - group_add                                   = [] -> null
      - hostname                                    = "0934e7114cff" -> null
      - id                                          = "0934e7114cffead03c3c54927b33496d8558a3b183ae1e3882a75409bb213c61" -> nul
l
      - image                                       = "sha256:11ceee7cdc57225711b8382e1965974bbb259de14a9f5f7d6b9f161ced50a10a"
 -> 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                                = [
          - mac_address                             = "02:42:ac:11:00:02"
          - network_name                            = "bridge"
            # (2 unchanged attributes hidden)
        },
      ] -> null
      - network_mode                                = "bridge" -> 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
        # (7 unchanged attributes hidden)

      - ports {
          - external = 8000 -> null
          - internal = 80 -> null
          - ip       = "0.0.0.0" -> null
          - protocol = "tcp" -> null
        }
    }

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

```
akshatsrivastava@Akshats-Air terraform % terraform apply
docker_image.nginx: Refreshing state... [id=sha256:11ceee7cdc57225711b8382e1965974bbb259de14a9f5f7d6b9f161ced50a10anginx]
docker_container.nginx: Refreshing state... [id=0934e7114cffead03c3c54927b33496d8558a3b183ae1e3882a75409bb213c61]

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
  # (because docker_container.nginx is not in configuration)
  - 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                                  = [
          - "/docker-entrypoint.sh",
        ] -> null
      - env                                         = [] -> null
      - group_add                                   = [] -> null
      - hostname                                    = "0934e7114cff" -> null
      - id                                          = "0934e7114cffead03c3c54927b33496d8558a3b183ae1e3882a75409bb213c61" -> nul
l
      - image                                       = "sha256:11ceee7cdc57225711b8382e1965974bbb259de14a9f5f7d6b9f161ced50a10a"
 -> 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                                = [
          - {
      - network_mode                                = "bridge" -> 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
        # (7 unchanged attributes hidden)

      - ports {
          - external = 8000 -> 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 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=0934e7114cffead03c3c54927b33496d8558a3b183ae1e3882a75409bb213c61]
docker_container.nginx: Destruction complete after 0s

Apply complete! Resources: 0 added, 0 changed, 1 destroyed.
```

5. Destory the complete resource.



```
akshatsrivastava@Akshats-Air terraform % terraform destroy

docker_image.nginx: Refreshing state... [id=sha256:11ceee7cdc57225711b8382e1965974bbb259de14a9f5f7d6b9f161ced50a10anginx]

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_image.nginx will be destroyed
  - resource "docker_image" "nginx" {
      - id           = "sha256:11ceee7cdc57225711b8382e1965974bbb259de14a9f5f7d6b9f161ced50a10anginx" -> null
      - image_id     = "sha256:11ceee7cdc57225711b8382e1965974bbb259de14a9f5f7d6b9f161ced50a10a" -> null
      - keep_locally = false -> null
      - name         = "nginx" -> null
      - repo_digest  = "nginx@sha256:56b388b0d79c738f4cf51bbaf184a14fab19337f4819ceb2cae7d94100262de8" -> 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_image.nginx: Destroying... [id=sha256:11ceee7cdc57225711b8382e1965974bbb259de14a9f5f7d6b9f161ced50a10anginx]
docker_image.nginx: Destruction complete after 0s

Destroy complete! Resources: 1 destroyed.
```

Google Cloud Use Case

1. Create a plan and use the plan to make changes to the resource.Create resource with dependencies (implicit and explicit).

```
akshatsrivastava@Akshats-Air terraform % 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:

  # google_compute_instance.vm_instance will be created
  + resource "google_compute_instance" "vm_instance" {
      + can_ip_forward       = false
      + cpu_platform         = (known after apply)
      + current_status       = (known after apply)
      + deletion_protection  = false
      + guest_accelerator    = (known after apply)
      + id                   = (known after apply)
      + instance_id          = (known after apply)
      + label_fingerprint    = (known after apply)
      + machine_type         = "e2-micro"
      + metadata_fingerprint = (known after apply)
      + min_cpu_platform     = (known after apply)
      + name                 = "terraform-instance"
      + project              = (known after apply)
      + self_link            = (known after apply)
      + tags_fingerprint     = (known after apply)
      + zone                 = (known after apply)

      + boot_disk {
          + auto_delete                = true
          + device_name                = (known after apply)
          + disk_encryption_key_sha256 = (known after apply)
          + kms_key_self_link          = (known after apply)
          + mode                       = "READ_WRITE"
          + source                     = (known after apply)

          + initialize_params {
              + image  = "debian-cloud/debian-11"
              + labels = (known after apply)
              + size   = (known after apply)
              + type   = (known after apply)
            }
        }

      + network_interface {
          + name               = (known after apply)
          + network            = (known after apply)
```

```
          + initialize_params {
              + image  = "debian-cloud/debian-11"
              + labels = (known after apply)
              + size   = (known after apply)
              + type   = (known after apply)
            }
        }

      + network_interface {
          + name               = (known after apply)
          + network            = (known after apply)
          + network_ip         = (known after apply)
          + subnetwork         = (known after apply)
          + subnetwork_project = (known after apply)

          + access_config {
              + nat_ip       = (known after apply)
              + network_tier = (known after apply)
            }
        }
    }

  # google_compute_network.vpc_network will be created
  + resource "google_compute_network" "vpc_network" {
      + auto_create_subnetworks         = true
      + delete_default_routes_on_create = false
      + gateway_ipv4                    = (known after apply)
      + id                              = (known after apply)
      + mtu                             = (known after apply)
      + name                            = "terraform-network"
      + project                         = (known after apply)
      + routing_mode                    = (known after apply)
      + self_link                       = (known after apply)
    }

Plan: 2 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.
```

```
akshatsrivastava@Akshats-Air terraform % terraform apply
google_compute_network.vpc_network: Refreshing state... [id=projects/silent-window-426701-m1/global/networks/terraform-network]

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:

  # google_compute_instance.vm_instance will be created
  + resource "google_compute_instance" "vm_instance" {
      + can_ip_forward       = false
      + cpu_platform         = (known after apply)
      + current_status       = (known after apply)
      + deletion_protection  = false
      + guest_accelerator    = (known after apply)
      + id                   = (known after apply)
      + instance_id          = (known after apply)
      + label_fingerprint    = (known after apply)
      + machine_type         = "e2-micro"
      + metadata_fingerprint = (known after apply)
      + min_cpu_platform     = (known after apply)
      + name                 = "terraform-instance"
      + project              = (known after apply)
      + self_link            = (known after apply)
      + tags_fingerprint     = (known after apply)
      + zone                 = "northamerica-northeast1-a"

      + boot_disk {
          + auto_delete                = true
          + device_name                = (known after apply)
          + disk_encryption_key_sha256 = (known after apply)
          + kms_key_self_link          = (known after apply)
          + mode                       = "READ_WRITE"
          + source                     = (known after apply)

          + initialize_params {
              + image  = "debian-cloud/debian-11"
              + labels = (known after apply)
              + size   = (known after apply)
              + type   = (known after apply)
            }
        }
```

```
      + network_interface {
          + name               = (known after apply)
          + network            = "https://www.googleapis.com/compute/v1/projects/silent-window-426701-m1/global/networks/terraf
orm-network"
          + network_ip         = (known after apply)
          + subnetwork         = (known after apply)
          + subnetwork_project = (known after apply)

          + access_config {
              + nat_ip       = (known after apply)
              + network_tier = (known after apply)
            }
        }
    }

Plan: 1 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

google_compute_instance.vm_instance: Creating...
google_compute_instance.vm_instance: Still creating... [10s elapsed]
google_compute_instance.vm_instance: Creation complete after 16s [id=projects/silent-window-426701-m1/zones/northamerica-northe
ast1-a/instances/terraform-instance]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

2. Make destructive changes

```
akshatsrivastava@Akshats-Air terraform % terraform apply
google_compute_network.vpc_network: Refreshing state... [id=projects/silent-window-426701-m1/global/networks/terraform-network]
google_compute_instance.vm_instance: Refreshing state... [id=projects/silent-window-426701-m1/zones/northamerica-northeast1-a/instances/terraform-instance]

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

Terraform will perform the following actions:

  # google_compute_instance.vm_instance will be updated in-place
  ~ resource "google_compute_instance" "vm_instance" {
        id                        = "projects/silent-window-426701-m1/zones/northamerica-northeast1-a/instances/terraform-instanc
e"
        name                      = "terraform-instance"
        tags                      = []
        # (21 unchanged attributes hidden)

      ~ network_interface {
            name               = "nic0"
          ~ network            = "https://www.googleapis.com/compute/v1/projects/silent-window-426701-m1/global/networks/terraf
orm-network" -> (known after apply)
            # (4 unchanged attributes hidden)

            # (1 unchanged block hidden)
        }

        # (3 unchanged blocks hidden)
    }

  # google_compute_network.vpc_network must be replaced
-/+ resource "google_compute_network" "vpc_network" {
      + gateway_ipv4                    = (known after apply)
      ~ id                              = "projects/silent-window-426701-m1/global/networks/terraform-network" -> (known after
apply)
      ~ mtu                             = 0 -> (known after apply)
      ~ name                            = "terraform-network" -> "terraform-network1" # forces replacement
      ~ project                         = "silent-window-426701-m1" -> (known after apply)
      ~ routing_mode                    = "REGIONAL" -> (known after apply)
      ~ self_link                       = "https://www.googleapis.com/compute/v1/projects/silent-window-426701-m1/global/networ
ks/terraform-network" -> (known after apply)
        # (3 unchanged attributes hidden)
    }
```

3. Destory the complete resource.

```
akshatsrivastava@Akshats-Air terraform % terraform destroy
google_compute_network.vpc_network: Refreshing state... [id=projects/silent-window-426701-m1/global/networks/terraform-network]
google_compute_instance.vm_instance: Refreshing state... [id=projects/silent-window-426701-m1/zones/northamerica-northeast1-a/instances/terraform-instance]

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:

  # google_compute_instance.vm_instance will be destroyed
  - resource "google_compute_instance" "vm_instance" {
      - can_ip_forward          = false -> null
      - cpu_platform            = "Intel Broadwell" -> null
      - current_status          = "RUNNING" -> null
      - deletion_protection     = false -> null
      - enable_display          = false -> null
      - guest_accelerator       = [] -> null
      - id                      = "projects/silent-window-426701-m1/zones/northamerica-northeast1-a/instances/terraform-instanc
e" -> null
      - instance_id             = "3041315647326919360" -> null
      - label_fingerprint       = "42WmSpB8rSM=" -> null
      - labels                  = {} -> null
      - machine_type            = "e2-micro" -> null
      - metadata                = {} -> null
      - metadata_fingerprint    = "Zq45301vpXA=" -> null
      - name                    = "terraform-instance" -> null
      - project                 = "silent-window-426701-m1" -> null
      - resource_policies       = [] -> null
      - self_link               = "https://www.googleapis.com/compute/v1/projects/silent-window-426701-m1/zones/northamerica-no
rtheast1-a/instances/terraform-instance" -> null
      - tags                    = [] -> null
      - tags_fingerprint        = "42WmSpB8rSM=" -> null
      - zone                    = "northamerica-northeast1-a" -> null
        # (4 unchanged attributes hidden)

      - boot_disk {
          - auto_delete                = true -> null
          - device_name                = "persistent-disk-0" -> null
          - mode                       = "READ_WRITE" -> null
          - source                     = "https://www.googleapis.com/compute/v1/projects/silent-window-426701-m1/zones/northame
rica-northeast1-a/disks/terraform-instance" -> null
            # (3 unchanged attributes hidden)
```

```
  # google_compute_network.vpc_network will be destroyed
  - resource "google_compute_network" "vpc_network" {
      - auto_create_subnetworks         = true -> null
      - delete_default_routes_on_create = false -> null
      - id                              = "projects/silent-window-426701-m1/global/networks/terraform-network" -> null
      - mtu                             = 0 -> null
      - name                            = "terraform-network" -> null
      - project                         = "silent-window-426701-m1" -> null
      - routing_mode                    = "REGIONAL" -> null
      - self_link                       = "https://www.googleapis.com/compute/v1/projects/silent-window-426701-m1/global/networ
ks/terraform-network" -> null
        # (2 unchanged attributes hidden)
    }

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

google_compute_instance.vm_instance: Destroying... [id=projects/silent-window-426701-m1/zones/northamerica-northeast1-a/instanc
es/terraform-instance]
google_compute_instance.vm_instance: Still destroying... [id=projects/silent-window-426701-m1/zones/...theast1-a/instances/terr
aform-instance, 10s elapsed]
google_compute_instance.vm_instance: Still destroying... [id=projects/silent-window-426701-m1/zones/...theast1-a/instances/terr
aform-instance, 20s elapsed]
google_compute_instance.vm_instance: Still destroying... [id=projects/silent-window-426701-m1/zones/...theast1-a/instances/terr
aform-instance, 30s elapsed]
google_compute_instance.vm_instance: Still destroying... [id=projects/silent-window-426701-m1/zones/...theast1-a/instances/terr
aform-instance, 40s elapsed]
google_compute_instance.vm_instance: Still destroying... [id=projects/silent-window-426701-m1/zones/...theast1-a/instances/terr
aform-instance, 50s elapsed]
google_compute_instance.vm_instance: Destruction complete after 52s
google_compute_network.vpc_network: Destroying... [id=projects/silent-window-426701-m1/global/networks/terraform-network]
google_compute_network.vpc_network: Still destroying... [id=projects/silent-window-426701-m1/global/networks/terraform-network,
 10s elapsed]
google_compute_network.vpc_network: Still destroying... [id=projects/silent-window-426701-m1/global/networks/terraform-network,
 20s elapsed]
google_compute_network.vpc_network: Still destroying... [id=projects/silent-window-426701-m1/global/networks/terraform-network,
 30s elapsed]
google_compute_network.vpc_network: Still destroying... [id=projects/silent-window-426701-m1/global/networks/terraform-network,
```

```
google_compute_network.vpc_network: Still destroying... [id=projects/silent-window-426701-m1/global/networks/terraform-network,
 40s elapsed]
google_compute_network.vpc_network: Still destroying... [id=projects/silent-window-426701-m1/global/networks/terraform-network,
 50s elapsed]
google_compute_network.vpc_network: Still destroying... [id=projects/silent-window-426701-m1/global/networks/terraform-network,
 1m0s elapsed]
google_compute_network.vpc_network: Still destroying... [id=projects/silent-window-426701-m1/global/networks/terraform-network,
 1m10s elapsed]
google_compute_network.vpc_network: Destruction complete after 1m12s

Destroy complete! Resources: 2 destroyed.
```

Main.tf (googel)

```
terraform {
required_providers {
google = {
source = "hashicorp/google"
version = "3.75.0"
}
}
}

provider "google" {
credentials = file("./silent-window-426701-m1-
5d41fde74341.json")
project = "silent-window-426701-m1"
region = "canada"
}

resource "google_compute_network" "vpc_network" {
name = "terraform-network"
}

resource "google_compute_instance" "vm_instance"
{
name = "terraform-instance"
machine_type = "e2-micro"
zone = "northamerica-northeast1-a"

boot_disk {
initialize_params {
image = "debian-cloud/debian-11"
}
}

network_interface {
network =
google_compute_network.vpc_network.self_link
access_config {
```

```
}
}
}
```

Main.tf (Docker)

```
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 = 8123
external = 8000
  }
}
```