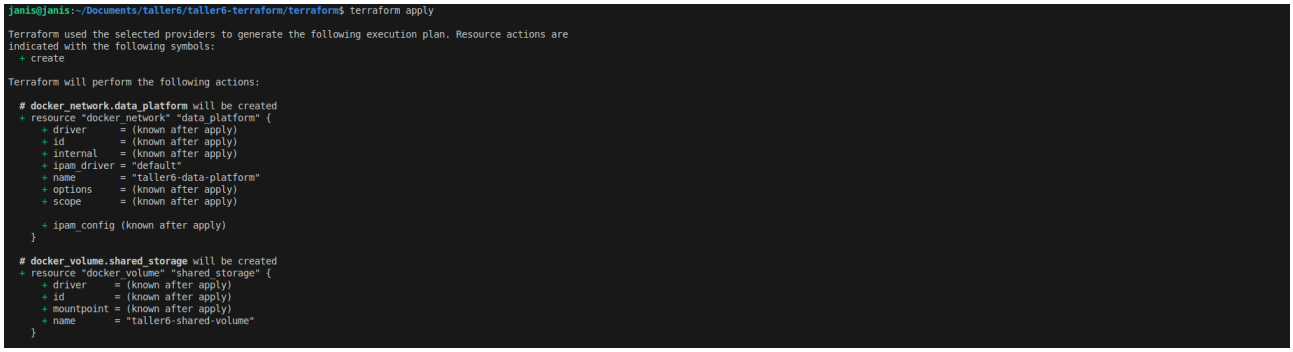
21016, Javier Chavez
21085, Andres Quezada
21600, Javier Ramirez
21631, Mario Cristales

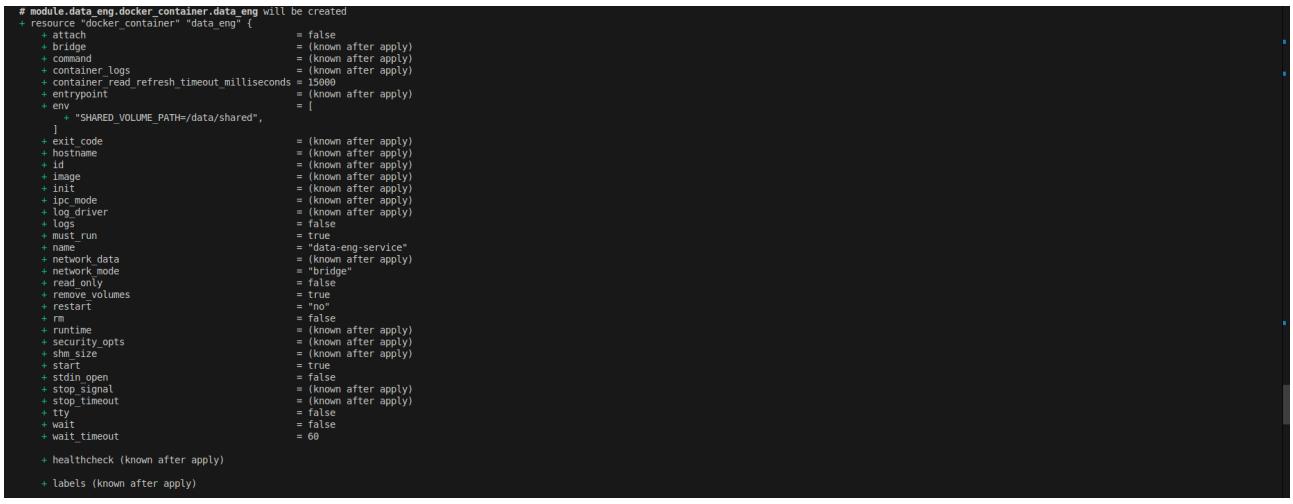
Taller 6

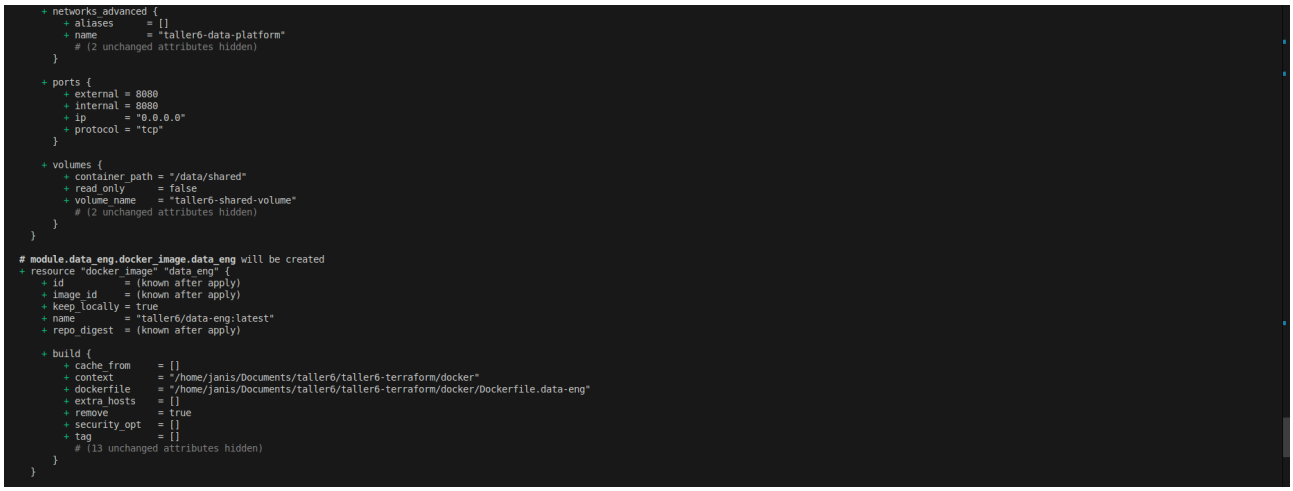
1. Link a repositorio de GitHub con código del tutorial de Terraform y la arquitectura planteada.

https://github.com/JaniMariQuesiRami/mlops-taller-6

2. Capturas de pantalla con terminal mostrando la creación exitosa de los diferentes recursos que maneja Docker (imágenes, contenedores, redes, etc).







```
  # module.data_science.docker_container.data_science will be created
  + resource "docker_container" "data_science" {
      + 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                                         = [
          + "SHARED_VOLUME_PATH=/data/shared",
        ]
      + 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                                        = "data-science-service"
      + network_data                                = (known after apply)
      + network_mode                                = "bridge"
      + read_only                                   = false
      + remove_volumes                              = true
      + restart                                     = "no"
      + rm                                          = false
      + runtime                                     = (known after apply)
      + security_opts                               = (known after apply)
      + shm_size                                    = (known after apply)
      + start                                       = true
      + stdin_open                                  = false
      + stop_signal                                 = (known after apply)
      + stop_timeout                                = (known after apply)
      + tty                                         = false
      + wait                                        = false
      + wait_timeout                                = 60

      + healthcheck (known after apply)

      + labels (known after apply)
```

```
      + networks_advanced {
          + aliases     = []
          + name        = "taller6-data-platform"
            # (2 unchanged attributes hidden)
        }

      + ports {
          + external = 8888
          + internal = 8888
          + ip       = "0.0.0.0"
          + protocol = "tcp"
        }

      + volumes {
          + container_path = "/data/shared"
          + read_only      = false
          + volume_name    = "taller6-shared-volume"
            # (2 unchanged attributes hidden)
        }
    }

  # module.data_science.docker_image.data_science will be created
  + resource "docker_image" "data_science" {
      + id          = (known after apply)
      + image_id    = (known after apply)
      + keep_locally = true
      + name        = "taller6/data-science:latest"
      + repo_digest  = (known after apply)

      + build {
          + cache_from     = []
          + context        = "/home/janis/Documents/taller6/taller6-terraform/docker"
          + dockerfile     = "/home/janis/Documents/taller6/taller6-terraform/docker/Dockerfile.data-science"
          + extra_hosts    = []
          + remove         = true
          + security_opt   = []
          + tag            = []
            # (13 unchanged attributes hidden)
        }
    }
```

```
Plan: 6 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + data_eng_container_id     = (known after apply)
  + data_science_container_id = (known after apply)
  + network_id                = (known after apply)
  + network_name              = "taller6-data-platform"
  + shared_volume_name        = "taller6-shared-volume"

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_volume.shared_storage: Creating...
docker_network.data_platform: Creating...
module.data_eng.docker_image.data_eng: Creating...
module.data_science.docker_image.data_science: Creating...
docker_volume.shared_storage: Creation complete after 0s [id=taller6-shared-volume]
module.data_science.docker_image.data_science: Creation complete after 1s [id=sha256:a487d15def7a076ab0a74600454ea94946345c7642e08957170e5d494fbc41c9taller6/data-science:latest]
module.data_eng.docker_image.data_eng: Creation complete after 1s [id=sha256:5a72ebb5ff00f438fef5e706058ce9ec16eeffd390ba22608567ba1a39fb08a0taller6/data-eng:latest]
docker_network.data_platform: Creation complete after 2s [id=35496f26340139a651c7902c24c24af4df2520b83d3c3c89f70f69e1c48a4aca]
module.data_science.docker_container.data_science: Creating...
module.data_eng.docker_container.data_eng: Creating...
module.data_science.docker_container.data_science: Creation complete after 1s [id=2a7cb419282852066b48da5eb52bebe0419260685cf5a7940d8ca34964f41d4e]
module.data_eng.docker_container.data_eng: Creation complete after 1s [id=fd0a1fff92592dd0a12712b6561bb5d1a598d57f5cdb34eb15418bb938327169]

Apply complete! Resources: 6 added, 0 changed, 0 destroyed.

Outputs:

data_eng_container_id = "fd0a1fff92592dd0a12712b6561bb5d1a598d57f5cdb34eb15418bb938327169"
data_science_container_id = "2a7cb419282852066b48da5eb52bebe0419260685cf5a7940d8ca34964f41d4e"
network_id = "35496f26340139a651c7902c24c24af4df2520b83d3c3c89f70f69e1c48a4aca"
network_name = "taller6-data-platform"
shared_volume_name = "taller6-shared-volume"
```

```
janis@janis:~/Documents/taller6/taller6-terraform/terraform$ sudo docker image ls
[sudo] password for janis:
REPOSITORY              TAG       IMAGE ID       CREATED          SIZE
taller6/data-science    latest    a487d15def7a   4 minutes ago    290MB
taller6/data-eng        latest    5a72ebb5ff00   5 minutes ago    275MB
hello-world             latest    74cc54e27dc4   8 months ago     10.1kB
```

3. Link a Docker Hub de las imágenes generadas.

https://hub.docker.com/repository/docker/jannisce2508/taller6-data-science/general

https://hub.docker.com/repository/docker/jannisce2508/taller6-data-eng/general

4. Link a Documentación de Terraform con comando(s) para desplegar recursos de Docker.

https://developer.hashicorp.com/terraform/tutorials/docker-get-started

5. Conclusiones del ejercicio pensando en las aplicaciones que tendría la Infraestructura como Código (IaC) en proyectos de Machine Learning.

IaC facilita reproducibilidad total de entornos de ML porque versiona red, volúmenes e imágenes junto al código. Automatizar despliegues reduce errores manuales y acelera el time-to-data para nuevos equipos. Terraform aporta trazabilidad y revisiones por código para la infraestructura que soporta pipelines. Separar ingeniería y ciencia de datos en contenedores favorece portabilidad y escalabilidad independiente. Declarar todo en código habilita integración con CI/CD para validar y promover entornos consistentes desde desarrollo hasta producción.