

## PROYECTO VIRTUALIZACIÓN

Mar Rosado, Paola Mérida



Virtualización Mayo 2024

Creación de un monolito



Framework



FLASK

Lenguaje



PYTHON

Crear una imagen y un contenedor en Docker para la aplicación



```
◆ Dockerfile ×
FROM python:3.10-alpine
     WORKDIR /app
     COPY requirements.txt .
     RUN pip3 install -r requirements.txt
     COPY . .
  8
  9
 10
     EXPOSE 8000
 11
     CMD gunicorn --bind 0.0.0.0:8000 app:app
 12
 13
```

Tecnologia de contenedores



Escalar horizontalmente el monolito por medio de un balanceador de carga



```
nginx.conf ×
dockerize > flask > 🌼 nginx.conf
       events {
           worker_connections 1000;
       http [
           server{
               listen 9999;
               location / {
   9
 10
                    proxy_pass http://flask_app:8000;
 11
 12
 13
```

Balanceador



```
docker-compose.yml X
dockerize > flask > 🐡 docker-compose.yml
       services:
 28
 29
         flask_app:
 30
           image: awesome-flask
 31
           ports:
 32
             - "8000"
 33
         nginx:
           image: nginx:latest
 34
           volumes:
 35
 36
             - ./logs:/var/log/nginx
             - "./nginx.conf:/etc/nginx/nginx.conf"
 37
           depends_on:
 38
             - flask_app
 39
 40
           ports:
               0000-0000
```



Creación de logs



```
docker-compose.yml X
dockerize > flask > � docker-compose.yml
       version: '3.3'
       services:
         grafana:
           image: grafana/grafana-oss
           container_name: grafana
           restart: unless-stopped
           ports:
            - '3000:3000'
           volumes:
 10
 11
             - grafana_data:/var/lib/grafana
         loki:
 12
          image: grafana/loki:2.9.0
 13
           container_name: grafana-loki
 14
           command: -config.file=/etc/loki/local-config.yaml
 15
 16
           volumes:
            - ./loki-config.yaml:/etc/loki/local-config.yaml
 17
           ports:
 18
             - "3100:3100"
 19
        promtail:
 20
           image: grafana/promtail:2.9.0
 21
           container_name: promtail
 22
 23
           volumes:
 24
             - ./logs:/var/log/nginx
             - ./promtail-config.yaml:/etc/promtail/config.yml
 25
```

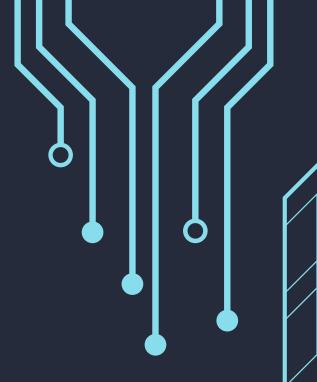
Herramienta de obtención de datos



Almacenamiento de logs

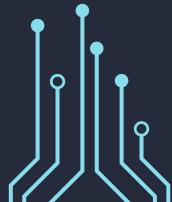


LOKI



```
≡ access.log ×
```

```
dockerize > flask > logs > ≡ access.log
  1 172.23.0.1 - - [04/Apr/2024:18:27:15 +0000] "GET / HTTP/1.1" 200 38 "-" "Mozilla/5.0 (Windows NT 10.0;
      172.23.0.1 - - [04/Apr/2024:18:27:16 +0000] "GET / HTTP/1.1" 200 38 "-" "Mozilla/5.0 (Windows NT 10.0;
      172.23.0.1 - - [04/Apr/2024:18:27:17 +0000] "GET / HTTP/1.1" 200 38 "-" "Mozilla/5.0 (Windows NT 10.0;
      172.23.0.1 - - [04/Apr/2024:18:27:17 +0000] "GET / HTTP/1.1" 200 38 "-" "Mozilla/5.0 (Windows NT 10.0;
      172.23.0.1 - - [04/Apr/2024:18:27:19 +0000] "GET / HTTP/1.1" 200 38 "-" "Mozilla/5.0 (Windows NT 10.0;
      172.23.0.1 - - [04/Apr/2024:18:28:45 +0000] "GET / HTTP/1.1" 200 38 "-" "Mozilla/5.0 (Windows NT 10.0;
      172.23.0.1 - - [04/Apr/2024:18:31:42 +0000] "GET / HTTP/1.1" 200 38 "-" "Mozilla/5.0 (Windows NT 10.0;
      172.23.0.1 - - [04/Apr/2024:18:31:44 +0000] "GET / HTTP/1.1" 200 38 "-"
                                                                              "Mozilla/5.0 (Windows NT 10.0;
      172.23.0.1 - - [04/Apr/2024:18:31:55 +0000] "GET / HTTP/1.1" 200 38 "-"
                                                                              "Mozilla/5.0 (Windows NT 10.0;
     172.23.0.1 - - [04/Apr/2024:18:31:58 +0000] "GET / HTTP/1.1" 200 38 "-" "Mozilla/5.0 (Windows NT 10.0;
 10
      172.23.0.1 - - [04/Apr/2024:18:32:16 +0000] "GET / HTTP/1.1" 200 38 "-" "Mozilla/5.0 (Windows NT 10.0;
 11
 12
      172.23.0.1 - - [04/Apr/2024:19:24:58 +0000] "GET / HTTP/1.1" 200 38 "-" "Mozilla/5.0 (Windows NT 10.0;
      172.23.0.1 - - [04/Apr/2024:19:24:59 +0000] "GET / HTTP/1.1" 200 38 "-" "Mozilla/5.0 (Windows NT 10.0;
      172.23.0.1 - - [04/Apr/2024:19:24:59 +0000] "GET / HTTP/1.1" 200 38 "-" "Mozilla/5.0 (Windows NT 10.0;
      172.23.0.1 - - [04/Apr/2024:19:24:59 +0000] "GET / HTTP/1.1" 200 38 "-" "Mozilla/5.0 (Windows NT 10.0;
 15
      172.23.0.1 - - [04/Apr/2024:19:24:59 +0000] "GET / HTTP/1.1" 200 38 "-" "Mozilla/5.0 (Windows NT 10.0;
 16
      172.23.0.1 - - [04/Apr/2024:19:25:00 +0000] "GET / HTTP/1.1" 200 38 "-" "Mozilla/5.0 (Windows NT 10.0;
 17
      172.23.0.1 - - [04/Apr/2024:19:25:00 +0000] "GET / HTTP/1.1" 200 38 "-" "Mozilla/5.0 (Windows NT 10.0;
 18
      172.19.0.1 - - [05/Apr/2024:02:30:13 +0000] "GET / HTTP/1.1" 200 38 "-" "Mozilla/5.0 (Windows NT 10.0;
      172.19.0.1 - - [05/Apr/2024:02:30:13 +0000] "GET /favicon.ico HTTP/1.1" 404 207 "http://localhost:9999/
     172.19.0.1 - - [05/Apr/2024:02:31:06 +0000] "GET / HTTP/1.1" 200 38 "-" "Mozilla/5.0 (Windows NT 10.0;
 21
 22 172.19.0.1 - - [05/Apr/2024:02:31:07 +0000] "GET / HTTP/1.1" 200 38 "-" "Mozilla/5.0 (Windows NT 10.0;
      172.19.0.1 - - [05/Apr/2024:02:31:08 +0000] "GET / HTTP/1.1" 200 38 "-" "Mozilla/5.0 (Windows NT 10.0;
 23
      172.19.0.1 - - [05/Apr/2024:02:31:09 +0000] "GET / HTTP/1.1" 200 38 "-" "Mozilla/5.0 (Windows NT 10.0;
```



Almacenamiento de logs en la nube

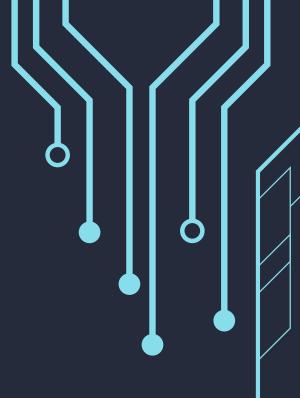


```
app2.py X
dockerize > flask > logs > ♦ app2.py > ...
      while True:
          #inicializa pyrebase
 17
 18
          counter=0
          firebase = pyrebase.initialize_app(firebaseConfig)
 19
          db=firebase.database()
 21
          #verificacion
 22
          if not db.child('Logs').shallow().get().val():
              #no hay nada aun en la base de datos
 23
              #1. Leer el archivo
              f = open("access.log", "r")
 25
              lines = f.readlines()
 27
              #escribir archivo
              count = 0
 29
               for line in lines:
                  count += 1
                  linea = line.strip()
 32
                  data = {
                       'LOG' : linea
 34
                  result = Connection.post('/Logs/', data)
              print(count)
          else:
              #Existen logs
              #Consultar cantidad
              counter = 0
              result= db.child("Logs").get()
 41
              for res in result.each():
```

Base de datos



FIREBASE



#### **Realtime Database**

Datos Reglas Copias de seguridad Uso 🕏 Extensions

Protege tus recursos de Realtime Database contra los abusos, como fraudes de facturación o phishing.

Configurar la Verificación de aplicaciones

https://virtualizacion-991a2-default-rtdb.firebaseio.com







https://virtualizacion-991a2-default-rtdb.firebaseio.com/



-NyM8MdZluou2t9oa7xN

L0G: "172.23.0.1 - - [04/Apr/2024:18:27:15 +0000] "GET / HTTP/1.1" 20

▼ — -NyM8MjWxSrzXzsqK-pV



Automatización de procesos



```
main.tf
          ×
dockerize > flask > app > 🚏 main.tf
       # Specifying Docker provider
       terraform {
         required_providers {
           docker = {
             source = "kreuzwerker/docker"
            version = "2.23.1"
 10
 11
       resource "null_resource" "run_command" {
 12
         triggers = {
 13
           always_run = "${timestamp()}"
 14
 15
 16
         provisioner "local-exec" {
 17
           command = "docker-compose up -d --scale flask_app=3"
 18
 19
 20
```

Programa para gestionar IaC



**TERRAFORM** 

# DEMOSTRACIÓN DE FUNCIONAMIENTO



