

Trabajo Práctico 4 - Arquitectura de Microservicios

Desarrollo:

1- Instanciación del sistema

- Clonar el repositorio


```
Last login: Thu Aug 24 14:22:16 on ttys000
pedrofernandez@MacBook-Air-de-Pedro 04-arquitectura-microservicios % mkdir -p socks-demo
pedrofernandez@MacBook-Air-de-Pedro 04-arquitectura-microservicios % cd socks-demo
pedrofernandez@MacBook-Air-de-Pedro socks-demo % git clone https://github.com/microservices-demo/microservices-demo.git
Cloning into 'microservices-demo'...
remote: Enumerating objects: 10197, done.
remote: Total 10197 (delta 0), reused 0 (delta 0), pack-reused 10197
Receiving objects: 100% (10197/10197), 52.95 MiB | 1.10 MiB/s, done.
Resolving deltas: 100% (6208/6208), done.
pedrofernandez@MacBook-Air-de-Pedro socks-demo % cd microservices-demo
```

- Ejecutar lo siguiente

```
pedrofernandez@MacBook-Air-de-Pedro microservices-demo % cd microservices-demo
pedrofernandez@MacBook-Air-de-Pedro microservices-demo % docker-compose -f deploy/docker-compose/docker-compose.yml up -d
WARN[0000] The "MYSQL_ROOT_PASSWORD" variable is not set. Defaulting to a blank string.
[+] Running 19/15
  ✓ orders-db Pulling
  ✓ payment 4 layers [#####] 0B/0B Pulled
  ✓ edge-router 5 layers [#####] 0B/0B Pulling
[+] Running 11/15 [#####] 0B/0B Pulling
  ✓ orders-db Pulled
  ✓ payment 4 layers [#####] 0B/0B Pulled
  ✓ edge-router 5 layers [#####] 0B/0B Pulled
  ✓ orders 8 layers [#####] 0B/0B Pulled
  ✓ rabbitmq 14 layers [#####] 0B/0B Pulled
  ✓ shipping 3 layers [#####] 0B/0B Pulled
  ✓ user-sim 11 layers [#####] 0B/0B Pulled
  ✓ catalogue 4 layers [#####] 0B/0B Pulled
  ✓ user 3 layers [#####] 0B/0B Pulled
  ✓ queue-master 2 layers [#####] 0B/0B Pulled
  ✓ catalogue-db 11 layers [#####] 0B/0B Pulled
  ✓ carts 3 layers [#####] 0B/0B Pulled
  ✓ front-end 8 layers [#####] 0B/0B Pulled
  ✓ user-db 12 layers [#####] 0B/0B Pulled
  ✓ carts-db 14 layers [#####] 0B/0B Pulled
[+] Running 29/29ers [#####] 0B/0B Pulling 272.3s 7 carts-db 14 layers [#####] 0B/0B Pulling 272.3s
  ✓ Network docker-compose_default
  ✓ Container docker-compose-orders-db-1 Started
  ✓ Container docker-compose-queue-master-1 Started
  ✓ Container docker-compose-user-db-1 Started
  ✓ Container docker-compose-rabbitmq-1 Started
  ✓ Container docker-compose-payment-1 Started
  ✓ Container docker-compose-front-end-1 Started
  ✓ Container docker-compose-catalogue-db-1 Started
  ✓ Container docker-compose-user-sim-1 Started
  ✓ Container docker-compose-catalogue-1 Started
  ✓ Container docker-compose-carts-1 Started
  ✓ Container docker-compose-edge-router-1 Started
  ✓ Container docker-compose-shipping-1 Started
  ✓ Container docker-compose-orders-1 Started
  ✓ Container docker-compose-user-1 Started
  ✓ Container docker-compose-carts-db-1 Started
! user-sim The requested image's platform (linux/amd64) does not match the detected host platform (linux/arm64/v8) and no specific platform was requested
```

OFFER OF THE DAYBuy 1000 socks, get a shoe for free!

Logged in as Manuel Bobadilla | Logout

weaveworks
SOCKS

HOME CATALOGUE ACCOUNT

0 items in cart

Home > My orders

Customer section

My orders

My orders

Your orders in one place.

If you have any questions, please feel free to [contact us](#), our customer service center is working for you 24/7.

Order	Date	Total	Status	Action
# 6310f4239bd1160007ba8449	2022-09-01 18:04:19	\$ 22.99	Shipped	View

Pages

Home Catalogue



Top categories

Magic Sport Action

Where to find us

Weaveworks Ltd.
32 – 38 Scrutton Street
London
EC2A 4RQ
UK

Stay in touch

2- Investigación de los componentes

1. Describa los contenedores creados, indicando cuales son los puntos de ingreso del sistema
2. Clonar algunos de los repositorios con el código de las aplicaciones

```
pedrofernandez@MacBook-Air-de-Pedro microservices-demo % cd ..
pedrofernandez@MacBook-Air-de-Pedro socks-demo % git clone https://github.com/microservices-demo/front-end.git
git clone https://github.com/microservices-demo/user.git
git clone https://github.com/microservices-demo/edge-router.git

Cloning into 'front-end'...
remote: Enumerating objects: 1236, done.
remote: Total 1236 (delta 0), reused 0 (delta 0), pack-reused 1236
Receiving objects: 100% (1236/1236), 47.90 MiB | 1.12 MiB/s, done.
Resolving deltas: 100% (687/687), done.
Cloning into 'user'...
remote: Enumerating objects: 1063, done.
remote: Total 1063 (delta 0), reused 0 (delta 0), pack-reused 1063
Receiving objects: 100% (1063/1063), 172.83 KiB | 988.00 KiB/s, done.
Resolving deltas: 100% (601/601), done.
Cloning into 'edge-router'...
remote: Enumerating objects: 50, done.
remote: Total 50 (delta 0), reused 0 (delta 0), pack-reused 50
Receiving objects: 100% (50/50), 14.51 KiB | 362.00 KiB/s, done.
Resolving deltas: 100% (12/12), done.
pedrofernandez@MacBook-Air-de-Pedro socks-demo %
```

3. ¿Por qué cree usted que se está utilizando repositorios separados para el código y/o la configuración del sistema? Explique puntos a favor y en contra.

Para poder desarrollar cada servicio por separado, de forma autónoma, mediante equipos de desarrollo distintos que utilizan herramientas distintas, que luego serán consumidos por un cliente que hará uso de todas estas funcionalidades.

4. ¿Cuál contenedor hace las veces de API Gateway?

El contenedor que hace de API Gateway es el edge_router, ER ApiGateway (Traefik)

5. Cuando ejecuto este comando:

```
curl http://localhost/customers
```

6. ¿Cuál de todos los servicios está procesando la operación?

El servicio de users.

7. ¿Y para los siguientes casos?

```
curl http://localhost/catalogue
curl http://localhost/tags
```

para /catalog Y /tags se usa el servicio catalog

8. ¿Como persisten los datos los servicios?

Para la persistencia, los servicios que lo necesiten tienen bases de datos, por ejemplo el servicio user tiene mongodb, catalog usa mysql. Cada servicio es el encargado de su propia persistencia para su funcionamiento.

9. ¿Cuál es el componente encargado del procesamiento de la cola de mensajes?

10. ¿Qué tipo de interfaz utilizan estos microservicios para comunicarse?

Los servicios se comunican mediante REST.