**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

cd 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

.

.

1. ¿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.

**Ventajas de utilizar repositorios separados:**

1. **Separación de preocupaciones**: Al separar el código y la configuración en repositorios diferentes, se puede lograr una mejor separación de preocupaciones entre la lógica de la aplicación y la infraestructura/subsistema en sí. Esto puede facilitar la gestión y la colaboración en equipos multidisciplinarios.
2. **Gestión de versiones independientes**: Los repositorios separados permiten la posibilidad de gestionar las versiones de la aplicación y la configuración de forma independiente. Esto puede ser útil si necesitas realizar cambios en la infraestructura sin afectar directamente el código de la aplicación, y viceversa.
3. **Escalabilidad y despliegue**: La separación puede facilitar la implementación escalable, ya que puedes actualizar o cambiar componentes de la infraestructura sin afectar todo el código de la aplicación. Esto también es útil en entornos de microservicios, donde diferentes componentes pueden evolucionar a diferentes velocidades.

**Desventajas de utilizar repositorios separados:**

1. **Complejidad de gestión**: Manejar múltiples repositorios puede ser más complejo que tener todo en un solo lugar. Requiere una planificación cuidadosa para asegurarse de que los cambios en diferentes partes de la aplicación se coordinen correctamente.
2. **Problemas de sincronización**: Puede surgir el problema de mantener la coherencia entre el código de la aplicación y la configuración. Asegurarse de que las versiones sean compatibles puede requerir esfuerzo adicional.
3. **Mayor barrera de entrada**: Puede ser más difícil para nuevos desarrolladores entender cómo están interconectados los componentes si están en repositorios separados.
4. **Complicaciones en la integración continua (CI) y la entrega continua (CD)**: El proceso de integración continua y entrega continua puede volverse más complicado al manejar múltiples repositorios, ya que se deben coordinar las implementaciones de diferentes componentes.
5. ¿Cuál contenedor hace las veces de API Gateway?

El contenedor que hace las veces de API Gateway es el que pertenece al repositorio edge-router.git. Este servicio actúa como una pasarela que enruta las solicitudes de cliente a los microservicios correspondientes.

1. Cuando ejecuto este comando:

curl http://localhost/customers

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

La operación está siendo procesada por el servicio de **user.git**. En este contexto, "customers" parece ser un endpoint que maneja el servicio de usuarios.

1. ¿Y para los siguientes casos?

curl http://localhost/catalogue

curl http://localhost/tags

1. ¿Como perisisten los datos los servicios?
2. ¿Cuál es el componente encargado del procesamiento de la cola de mensajes?
3. ¿Qué tipo de interfaz utilizan estos microservicios para comunicarse?