



Venta de Helados

Aplicación de microservicios.

Integrantes:

- José Maximiliano Díaz Méndez
- Barajas Zavala Ulises

Descripción

Este proyecto consta de un punto de venta de una heladería en el que se divide en 2 servicios la funcionalidad del servidor para ser adaptada a una arquitectura de microservicios, a través del uso de un cluster de kubernetes.





Servicios

- **CRUD:** Esta parte se ocupa de hacer las llamadas a la Base de Datos, todas la llamadas relacionadas al CRUD, ya sea la de crear registros, leer, modificar y eliminar.
- **Auth:** Esta parte se ocupa de autenticar las llamadas que se hacen desde la web app, para agregar seguridad y privacidad, de modo que nadie más que la web-app puedan hacer solicitudes al CRUD.

Esto a través de un token que se entrega a la web-app, este token tiene una duración de 24 hrs y el CRUD confirma este token con auth cada que CRUD interactúa con la base de datos.

*Estos servicios utilizan una **réplica** para garantizar la consistencia de los servicios en la app*

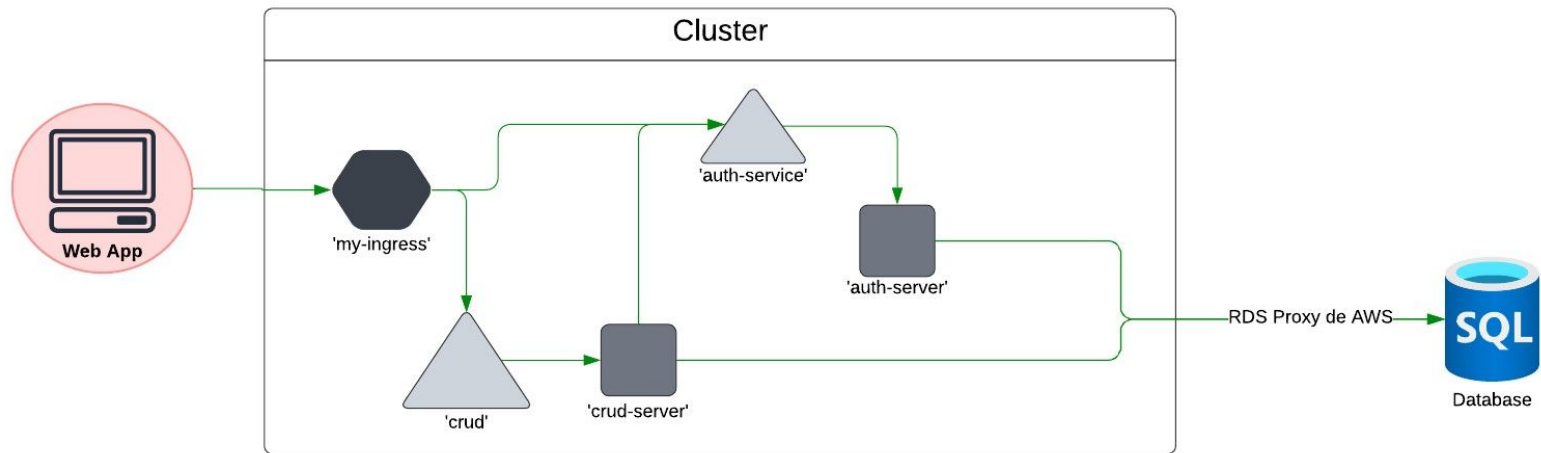


WEB - APP

La web - app creada usando Vue.js, un framework de JavaScript utilizado para construir **interfaces de usuario y aplicaciones de una sola página (SPA)**.

También fue envuelto en electron.js, un framework de código abierto que permite desarrollar aplicaciones de escritorio multiplataforma utilizando tecnologías web como HTML, CSS y JavaScript.

Con esto en mente la parte de la web app no cuenta como un servicio ya que es una app en sí mismo que se comunica con los servicios antes descritos.



Arquitectura

● Un secreto:

'db-secrets'

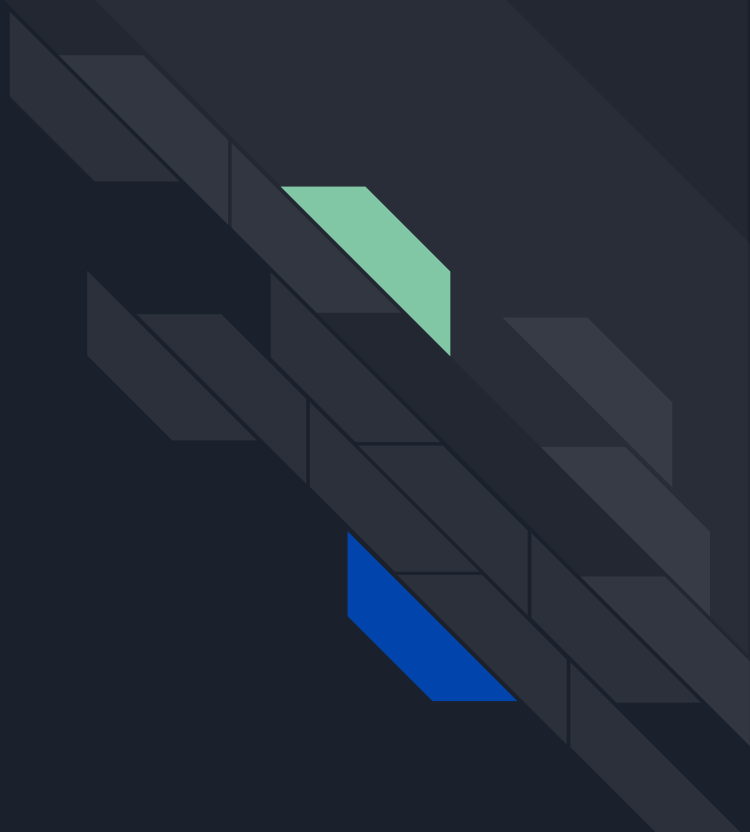
Almacena de forma segura los datos de inicio de sesión de la base de datos, incluyendo el nombre de host, usuario, contraseña y nombre de la base de datos.

● Pod:

'crud-server'

Ejecuta el servidor CRUD, se replica 2 veces para proporcionar redundancia y escalabilidad.

El servidor CRUD se comunica con la base de datos utilizando los datos de inicio de sesión almacenados en el secreto db-secrets.



Arquitectura

● Pod:

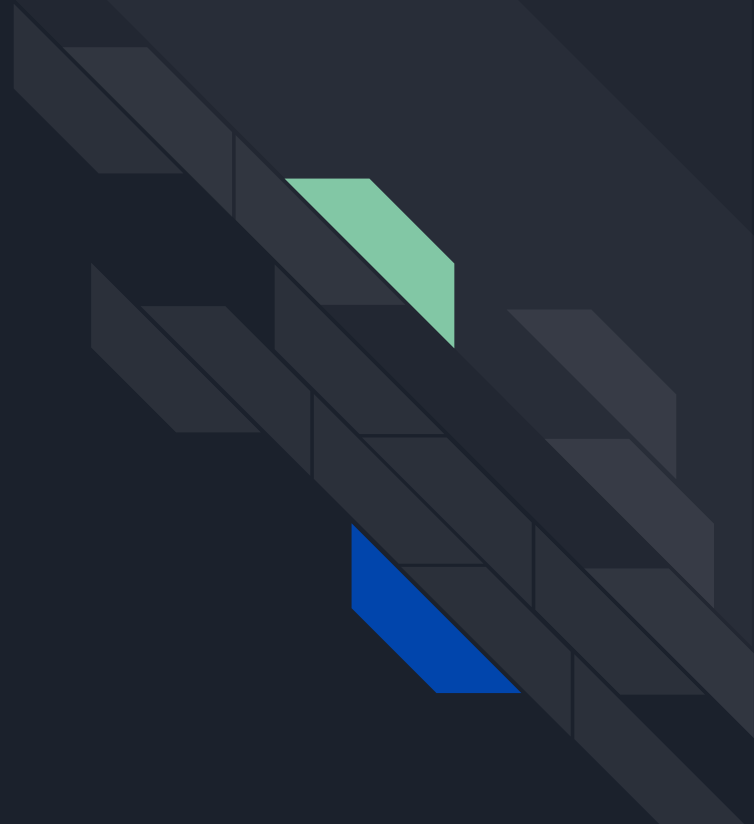
'auth-server'

Ejecuta el servidor de autenticación, que al igual que el servidor CRUD, se replica 2 veces para garantizar la disponibilidad. Este servidor utiliza los mismos datos de inicio de sesión de la base de datos del secreto db-secrets.

● 2 servicios internos:

'crud-service' 'auth-service'

Exponen los pods *crud-server* y *auth-server*, respectivamente y facilitan la comunicación entre los diferentes componentes de la aplicación dentro del clúster de Kubernetes.



Arquitectura

● Ingress:

'my-ingress'

Actúa como punto de entrada a la aplicación desde el exterior del clúster, enruta el tráfico HTTP entrante a los servicios internos *crud-service* y *auth-service* según las rutas especificadas.

De esta manera, permite que las solicitudes externas lleguen a los componentes correspondientes de la aplicación.

- La arquitectura también incluye **Istio** como una capa de servicio de malla, que proporciona funcionalidades como el enrutamiento, el control de tráfico, la seguridad y la observabilidad para los servicios desplegados en el clúster de Kubernetes.

A través de la configuración de reglas de tráfico y políticas de seguridad, Istio ayuda a gestionar y asegurar las comunicaciones entre los diferentes componentes de la aplicación.

