



Centro de Enseñanza Técnica Industrial

Desarrollo de Software

Microservicios - Storix

Jesús Alberto Aréchiga Carrillo

22310439 6N

Profesor

José Francisco Pérez Reyes

Febrero 2025

Guadalajara, Jalisco

Introducción

Storix es una plataforma modular de almacenamiento y gestión de datos basada en una arquitectura de microservicios. El objetivo principal es ofrecer una solución escalable, resiliente y de alta disponibilidad, que integre diversas funcionalidades de almacenamiento, administración, seguridad y monitorización. Este diseño contempla los siguientes aspectos clave:

- **Capacidad y Demanda:** Cada microservicio se escala de manera independiente según la carga y las necesidades del sistema.
- **Continuidad del Servicio:** Se implementan mecanismos de tolerancia a fallos y redundancia para garantizar la disponibilidad continua.
- **Calidad en los Servicios:** Se establecen métricas de rendimiento (SLA/SLO) y se integran procesos de testeo automatizado.
- **Desarrollo e Integración Continuos:** Uso de pipelines CI/CD para la construcción, prueba y despliegue automatizado de cada microservicio.

La solución se compone de múltiples microservicios interconectados, cada uno responsable de una función específica. La orquestación se realiza mediante contenedores gestionados a través de Kubernetes, lo que permite la escalabilidad y la resiliencia del sistema.



Descripción de los microservicios:

Administración y autorización

Se encarga de gestionar la configuración global del sistema, permitiendo la creación y administración de volúmenes, snapshots y políticas de backup, y está diseñado para escalar horizontalmente según la cantidad de solicitudes, asegurando la consistencia y calidad del sistema mediante pruebas unitarias e integraciones continuas.

Compartición de archivos

Proporciona servicios de acceso y compartición de datos a través de protocolos como SMB, NFS o mediante una API REST, adaptándose a picos de tráfico mediante balanceadores de carga y cachés, y garantizando la disponibilidad y rendimiento del servicio mediante pruebas de estrés y monitorización constante.

Almacenamiento en bloque

Administra el acceso a recursos de almacenamiento en bloque, asignando dinámicamente recursos de disco y ofreciendo baja latencia para los clientes, con capacidades de escalado automático y validaciones constantes de integridad a través de simulaciones de carga y tests de resiliencia.

Autenticación y autorización

Gestiona la seguridad del sistema verificando credenciales, controlando accesos y generando tokens para la comunicación entre microservicios, implementando mecanismos de caché y escalado automático para soportar un alto volumen de solicitudes, junto a pruebas de penetración y revisiones de código regulares para mantener altos estándares de seguridad.

Monitorización y alertas

Recopila y analiza métricas de rendimiento y estado de todos los microservicios, centralizando logs y enviando notificaciones en tiempo real mediante dashboards integrados, lo que permite detectar y reaccionar

rápidamente ante incidencias, garantizando la continuidad y calidad del servicio en un entorno de alta disponibilidad.

Backup y replicación

Coordina y ejecuta tareas de respaldo y replicación de datos, asegurando la recuperación ante fallos y la integridad de la información, adaptándose tanto a pequeños volúmenes como a grandes cantidades de datos a través de escalado automático, pruebas periódicas de restauración y validaciones de integridad.

Integración y webhooks

Facilita la conexión e integración con sistemas externos mediante APIs y webhooks, permitiendo la automatización de flujos de trabajo y la incorporación de nuevas funcionalidades sin afectar la operatividad del núcleo del sistema, garantizando comunicaciones seguras y eficientes entre servicios y plataformas.

Consideraciones transversales

Capacidad y escalabilidad

Cada microservicio se ejecuta en contenedores que permiten una fácil replicación y orquestación a través de plataformas como Kubernetes, facilitando el escalado horizontal para responder dinámicamente a las variaciones en la carga de trabajo y garantizando que el sistema se adapte a las necesidades de capacidad sin afectar el rendimiento general.

Continuidad del servicio

Se implementan mecanismos de redundancia, balanceo de carga y tolerancia a fallos, junto con estrategias de replicación de datos y circuit breakers, para asegurar que el sistema mantenga una alta disponibilidad y continuidad operativa incluso en escenarios de fallos o picos de demanda.

Calidad del servicio

Se definen y monitorean métricas de rendimiento (SLA/SLO) y se utilizan pipelines de integración y pruebas automatizadas que incluyen pruebas unitarias, de integración y de rendimiento, permitiendo detectar y corregir rápidamente cualquier incidencia que pueda afectar la experiencia del usuario final.

Desarrollo e integración continuos

La adopción de pipelines CI/CD permite que cada microservicio se desarrolle, pruebe y despliegue de manera independiente y automatizada, asegurando una entrega constante de mejoras y correcciones, además de mantener la consistencia entre los entornos de desarrollo, pruebas y producción.

para

Conclusiones

Al segmentar las funcionalidades en microservicios independientes, se garantiza una mayor flexibilidad para adaptarse a las variaciones en la demanda y mantener la continuidad del servicio. La implementación de consideraciones transversales, tales como escalabilidad, continuidad, calidad y desarrollo continuo, refuerza la robustez del sistema, permitiendo una rápida integración de mejoras y una respuesta ágil ante incidencias. En conjunto, **NimbusStorage** se posiciona como una propuesta innovadora que no solo satisface las necesidades actuales de almacenamiento, sino que también está preparada para evolucionar en entornos cada vez más dinámicos y exigentes.