Proyecto: Calculadora - Microservicios.

Integrantes:

Sebastian Navia

Nicolas Gomez Botero.

Fecha de entrega: miércoles 22 /2024

**1. Resumen Ejecutivo**

**Proyecto:**

* Desarrollo de una Calculadora con Microservicios y Automatización CI/CD.

**Descripción:**

* El proyecto consiste en crear una calculadora web con microservicios para cada operación aritmética. Se incluirá una interfaz de usuario para acceder a la calculadora desde una página web.

**Objetivo:**

* Automatizar completamente el desarrollo y despliegue de la aplicación mediante la integración continua (CI) y el despliegue continuo (CD).

**Tecnologías Destacadas:**

* Microservicios
* Kubernetes y Helm
* Github Action (CI)
* Argos CD (CD)
* Prometheus y Grafana, ELK Stack para observabilidad.

**Beneficios Esperados:**

1. Automatización eficiente del ciclo de vida de la aplicación.
2. Escalabilidad y flexibilidad gracias a la arquitectura de microservicios.
3. Monitoreo detallado del rendimiento y estado de la aplicación.

**2. Introducción**

**Antecedentes:**

El contexto actual del mercado demanda soluciones tecnológicas eficientes y escalables que puedan adaptarse rápidamente a las necesidades cambiantes de los usuarios. En este sentido, el desarrollo de aplicaciones modernas con arquitecturas basadas en microservicios ha ganado relevancia debido a su capacidad para ofrecer flexibilidad, modularidad y escalabilidad. En este contexto, surge la necesidad de desarrollar una calculadora con microservicios, una aplicación que pueda aprovechar las ventajas de esta arquitectura para proporcionar una experiencia de usuario óptima y una gestión eficiente de los recursos.

**Objetivos:**

El principal objetivo de este proyecto es desarrollar una calculadora web utilizando una arquitectura de microservicios, donde cada operación matemática se implementará como un servicio independiente. Esto permitirá una mayor modularidad, escalabilidad y mantenibilidad de la aplicación. Además, se busca automatizar completamente el proceso de desarrollo y despliegue de la aplicación mediante la implementación de integración continua (CI) y despliegue continuo (CD). Esto garantizará una entrega rápida y confiable de nuevas versiones de la aplicación al entorno de producción, mejorando la eficiencia operativa y la capacidad de respuesta a las demandas del mercado.

**Alcance:**

El alcance del proyecto incluye el desarrollo completo de la calculadora web con microservicios, así como la implementación de la automatización CI/CD para facilitar el proceso de desarrollo y despliegue. Se abordarán las cuatro operaciones aritméticas básicas (suma, resta, multiplicación, división), cada una como un microservicio separado.

1. **Descripción del Proyecto**

**Descripción General:**

El proyecto consiste en el desarrollo de una calculadora web utilizando una arquitectura de microservicios, donde cada operación matemática (suma, resta, multiplicación, división) será implementada como un servicio independiente. La interfaz de usuario estará diseñada para permitir a los usuarios ingresar los números y seleccionar la operación deseada, facilitando así la interacción con la calculadora. Además, se implementará un microservicio adicional que contendrá la lógica central de la aplicación, así como la interfaz gráfica de usuario, desarrollada con tecnologías web como HTML, JavaScript y CSS.

**Componentes Principales:**

* Microservicios:

- Se implementarán cuatro microservicios en total, uno para cada operación matemática (suma, resta, multiplicación, división), desarrollados en Python. Cada microservicio será responsable de ejecutar su operación específica y proporcionar el resultado al cliente.

- Un quinto microservicio contendrá la lógica central de la aplicación y la interfaz gráfica de usuario, desarrollada con HTML, JavaScript y CSS. Este microservicio actuará como un orquestador, gestionando las solicitudes del cliente y comunicándose con los microservicios de operaciones para realizar los cálculos.

* Ingress

- Se utilizará un controlador de Ingress en el clúster de Kubernetes para administrar el acceso externo a los servicios de la aplicación. El Ingress actuará como una puerta de entrada al clúster, dirigiendo el tráfico de red a los servicios correspondientes en función de las reglas de enrutamiento configuradas.

**Tecnologías Utilizadas:**

- Python:

- Se utilizará Python para implementar los microservicios de suma, resta, multiplicación y división debido a su versatilidad y eficiencia en el procesamiento de operaciones matemáticas.

- \*HTML, JavaScript y CSS:

- Estas tecnologías se utilizarán para desarrollar la interfaz gráfica de usuario de la calculadora, permitiendo una experiencia de usuario interactiva y atractiva.

- Kubernetes y Helm:

- Kubernetes se empleará como plataforma de orquestación de contenedores para gestionar y escalar los microservicios de la aplicación. Helm se utilizará para facilitar la gestión de los despliegues de la aplicación en Kubernetes mediante la definición de paquetes de recursos (charts).

- Github Actions:

Se utilizará github actions para la integración continua del servicio

- Argo CD

Se utilizará Argo CD para el despliegue continuo

- Slack:

- La herramienta Slack se utilizará para enviar notificaciones y alertas durante el proceso de CI/CD, manteniendo informados a los miembros del equipo sobre el estado y los resultados de las compilaciones y despliegues.

- SonarQube:

- SonarQube se empleará como herramienta de análisis estático de código para garantizar la calidad y la seguridad del código fuente de la aplicación, identificando posibles problemas y vulnerabilidades.

1. **Metodología**

**Metodología de Trabajo:**

El proyecto se organizó y gestionó de manera eficiente utilizando una combinación de metodologías ágiles y prácticas de desarrollo colaborativo. Se decidió dividir el servicio en 7 repositorios de Github y se adoptó una estrategia de ramificación de tipo trunk-based, que permite una integración continua fluida y una colaboración efectiva entre los miembros del equipo. Además, se implementó la metodología Kanban para la gestión de tareas, lo que facilitó la visualización y priorización de las actividades en curso, así como la identificación de posibles cuellos de botella y la asignación de recursos de manera óptima.

**5. Desarrollo**

**Fases del Proyecto:**

El desarrollo del proyecto se dividió en cinco fases principales, cada una con objetivos específicos y actividades definidas para lograrlos:

1. Planificación:

En esta fase inicial, el equipo de trabajo se reunió para establecer los parámetros del proyecto. Se definieron los requisitos del cliente, se establecieron los objetivos del proyecto y se asignaron roles y responsabilidades a los miembros del equipo. Además, se realizó una revisión de las tecnologías y herramientas disponibles para la implementación del proyecto.

1. Codificación:

La segunda fase se centró en la codificación de la aplicación. El equipo de desarrollo trabajó en la implementación de las funcionalidades principales de la calculadora, siguiendo las especificaciones y los requisitos establecidos en la fase de planificación. Se llevaron a cabo reuniones periódicas para revisar el progreso y abordar cualquier problema o desafío que surgiera durante el proceso de codificación.

1. Separación en Microservicios:

En esta fase, la aplicación se dividió en microservicios independientes para cada operación aritmética. Se implementaron microservicios dedicados para la suma, resta, multiplicación y división, así como un microservicio adicional para la lógica central de la aplicación y la interfaz gráfica de usuario. Esta separación permitió una mayor modularidad y flexibilidad en el desarrollo y la gestión de la aplicación.

1. Proceso de Automatización CI/CD:

La cuarta fase se enfocó en la implementación del proceso de automatización de integración continua (CI) y despliegue continuo (CD). Se configuraron pipelines de CI/CD utilizando Jenkins para automatizar las pruebas, la compilación y el despliegue de la aplicación en un entorno de prueba. Se realizaron pruebas exhaustivas para garantizar la estabilidad y la calidad del proceso de automatización.

1. Pruebas:

En la fase final, se llevaron a cabo pruebas exhaustivas para validar el funcionamiento y la calidad de la aplicación. Se realizaron pruebas unitarias, de integración y de aceptación para asegurar que la aplicación cumpliera con los requisitos del cliente y funcionara correctamente en diferentes escenarios. Se documentaron los resultados de las pruebas y se corrigieron los errores identificados antes de proceder con el cierre del proyecto.

|  |  |
| --- | --- |
| Tecnología -Implementación. | Check |
| Git | x |
| Branch Strategies | x |
| Microservices | x |
| KUBERNETES /HELEM | x |
| Github Actions | x |
| Argo CD | x |
| CI | x |
| Checkout | x |
| Unit test |  |
| Docker image | x |
| Docker push | x |
| CD | x |
| Deply | x |
| Observability | x |
| Prometheus y grafana | x |
| Elk |  |