Api Metrics

Solution architecture v0.1

Outline – draft 2024-03-05

Contenido

1 Introduccion

1.1 Antecedentes

API Metrics es un proyecto GoLang diseñado para gestionar métricas de dispositivos a través de una API RESTful. Es una solución on premise para gestionar métricas de dispositivos utilizando una API RESTful. Proporciona una interfaz simple y eficiente para registrar dispositivos, almacenar y recuperar métricas, y autenticar usuarios.

A continuación, se detallan los objetivos clave que se pretenden con las mejoras al sistema:

1. Escalabilidad:
    - Implementar una arquitectura que pueda escalar de manera automática y elástica para manejar el crecimiento en el volumen de datos y la cantidad de dispositivos monitoreados.
    - Utilizar servicios en la nube altamente escalables como AWS Lambda o Azure Functions para procesar las métricas de manera eficiente y sin problemas de capacidad.
2. Eficiencia:
    - Optimizar el procesamiento de datos en tiempo real para reducir la latencia y mejorar la velocidad de respuesta del sistema.
    - Implementar técnicas de almacenamiento en caché y distribución de carga para maximizar la eficiencia en el acceso a los datos.
    - Utilizar herramientas de monitoreo y análisis en la nube para identificar cuellos de botella y áreas de mejora en el rendimiento del sistema.
3. Mantenibilidad:
    - Diseñar una arquitectura modular y bien estructurada que facilite la incorporación de nuevas funcionalidades y la realización de cambios sin afectar la estabilidad del sistema.
    - Utilizar contenedores Docker y orquestación de contenedores con Kubernetes para simplificar el despliegue y la gestión de la infraestructura.
    - Implementar prácticas de desarrollo ágil y DevOps, incluyendo integración continua (CI) y entrega continua (CD), para acelerar el ciclo de desarrollo y garantizar la calidad del software.
    - Documentar adecuadamente el código, la infraestructura y los procesos para facilitar la colaboración entre equipos y garantizar la transferencia de conocimientos.

Al cumplir con estos objetivos, el nuevo sistema mejorado de métricas en la nube estará preparado para enfrentar los desafíos del crecimiento y proporcionar un servicio confiable, eficiente y fácil de mantener para los usuarios finales.

Impacto del negocio

La implementación de esta solución implicará una mejora en los tiempos de respuesta para la obtención y procesamiento de métricas, reduciendo errores y proporcionando integridad y calidad en la información.

1.2 Vision

-

1.3 Documentacion suplementaria

Reference	Description	Location
Diagrama C1	Modelo de la arquitectura	
Diagrama AWS	Arquitectura del Sistema de metricas	

1.4 Revisiones

Revision	Description	Published (2024-03-05)	Author
1.0			Angel Quintanilla

2 Descripcion general de la solucion

2.1 Impacto en el negocio

La implementación de esta solución implicará una mejora en los tiempos de respuesta para la obtención y procesamiento de métricas, reduciendo errores y proporcionando integridad y calidad en la información.

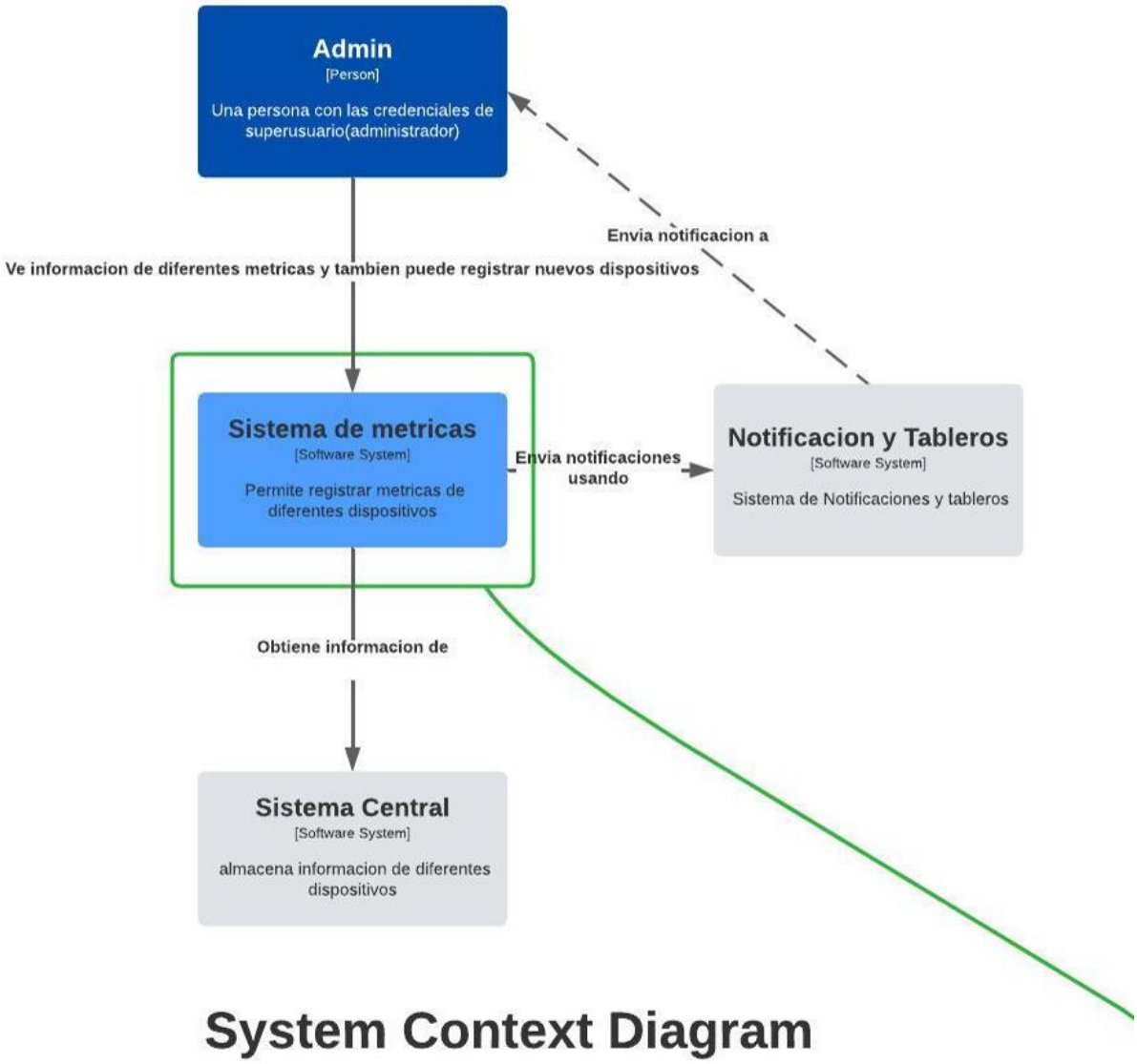
2.2 Resumen ejecutivo

Este documento proporciona una visión general de la arquitectura de una API para la recopilación de métricas. Los usuarios pueden enviar métricas a través de solicitudes POST a los endpoints designados.

2.3 Sistema de metricas

El sistema permite a los usuarios enviar métricas mediante solicitudes POST a los endpoints específicos de la API. Estas métricas se almacenan en una base de datos para su posterior análisis y procesamiento.

2.4 Level C1


Admin
[Person]
Una persona con las credenciales de superusuario(administrador)
Envia notificacion a
Ve informacion de diferentes metricas y tambien puede registrar nuevos dispositivos
Sistema de metricas
[Software System]
Permite registrar metricas de diferentes dispositivos
Envia notificaciones usando
Notificacion y Tableros
[Software System]
Sistema de Notificaciones y tableros
Obtiene informacion de
Sistema Central
[Software System]
almacena informacion de diferentes dispositivos
System Context Diagram

-

3 Requerimientos funcionales

3.1 Evaluar si un usuario es valido

El sistema valida si un usuario que envía métricas es válido y autorizado para hacerlo, para ellos se definen roles

3.2 Procesar una metrica valida

Una vez validada la autenticidad del usuario, el sistema debe procesar la métrica recibida y almacenarla en la base de datos correspondiente.

3.3 Funcionalidad de Notificacion

El sistema debe envia notificaciones al usuario para confirmar que la métrica ha sido recibida y procesada correctamente

3.4 Informes

Se debe proporcionar una funcionalidad para generar informes sobre las métricas recopiladas, incluyendo estadísticas y análisis de tendencias.

4 No funcionales

4.1 Encriptacion

La información de los usuarios es encriptada antes de ser almacenada en la base de datos.

4.2 Eligibilidad

Solo los usuarios autorizados deben poder enviar métricas a la API

4.3 Seguridad

El administrador accede a un sitio web seguro (https) y cuenta con un sistema de autenticación 2f para validar el acceso al sistema, además cuenta con un firewall en la nube para evitar ataques de terceros. Evalua también si las credenciales de los usuarios son válidas.

4.4 Integridad and Precision

El sistema es responsable de almacenamiento seguro, procesamiento de las metricas. Debido a la gran importancia de estos proceso, el software garantiza que no haya posibilidad de realizar ninguna operación inesperada o no autorizada.

4.5 Autenticacion

Se asegurará que las credenciales del usuario sean válidas conectándose a través del sistema de central ,proporcionando un mensaje si las credenciales no son legítimas o válidas.

4.6 Autorizacion

Solo los usuarios con credencial válida pueden tener acceso a los endpoints, además solo el administrador designado puede tener acceso a la funcionalidad de notificaciones y paneles.

4.7 Verificabilidad

El sistema proporciona a los usuarios la capacidad de verificar que sus métricas han sido recibidas y procesadas correctamente.

4.8 Robustez

El sistema es robusto y resistente a fallos, garantizando su disponibilidad y fiabilidad en todo momento.

4.9 Disponibilidad

El sistema estara disponible en todo momento, con una infraestructura de alta disponibilidad y redundancia para evitar interrupciones del servicio

4.10 Recoverabilidad

El sistema cuenta con una réplica de la base de datos, además tiene dos zonas dedicadas por si alguna se cae.

5 Descripcion Arquitectura actual

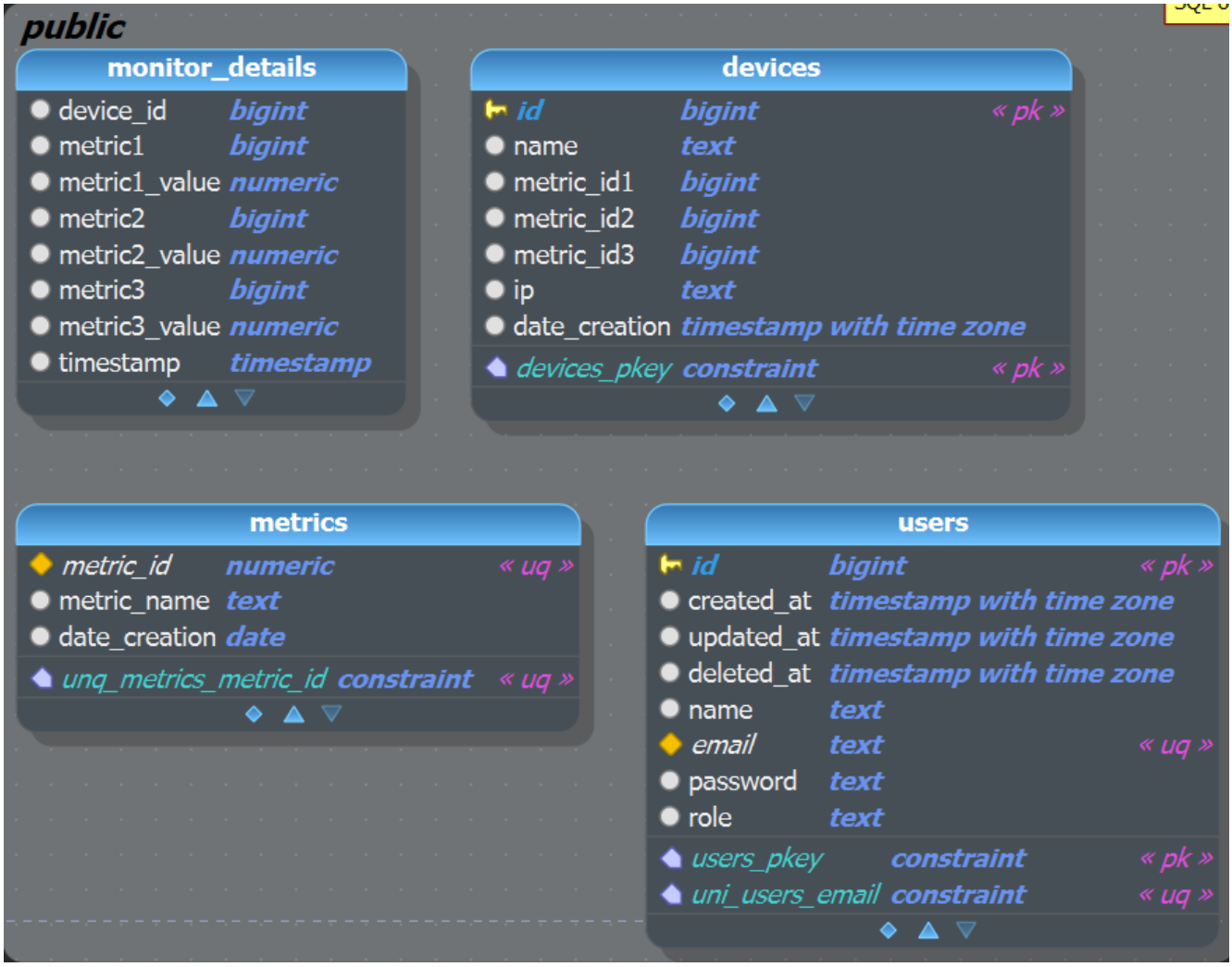
5.1 General description

El estili arquitectonico utilizado fue api rest son los debido a su facilidad para construir rápidamente y proporcionar acceso seguro.

5.2 Supuestos

Solo se necesita asegurar el registro de los dispositivos, el agente lee de un archive yaml, que se ha generado previamente, posteriormente se va a desarrollar un front end para accesar por diferentes medios, se necesita un datawarehouse para analizar la informacion de manera sumarizada , hay metricas que no las considera dynotrace, se necesitan reports customizados.

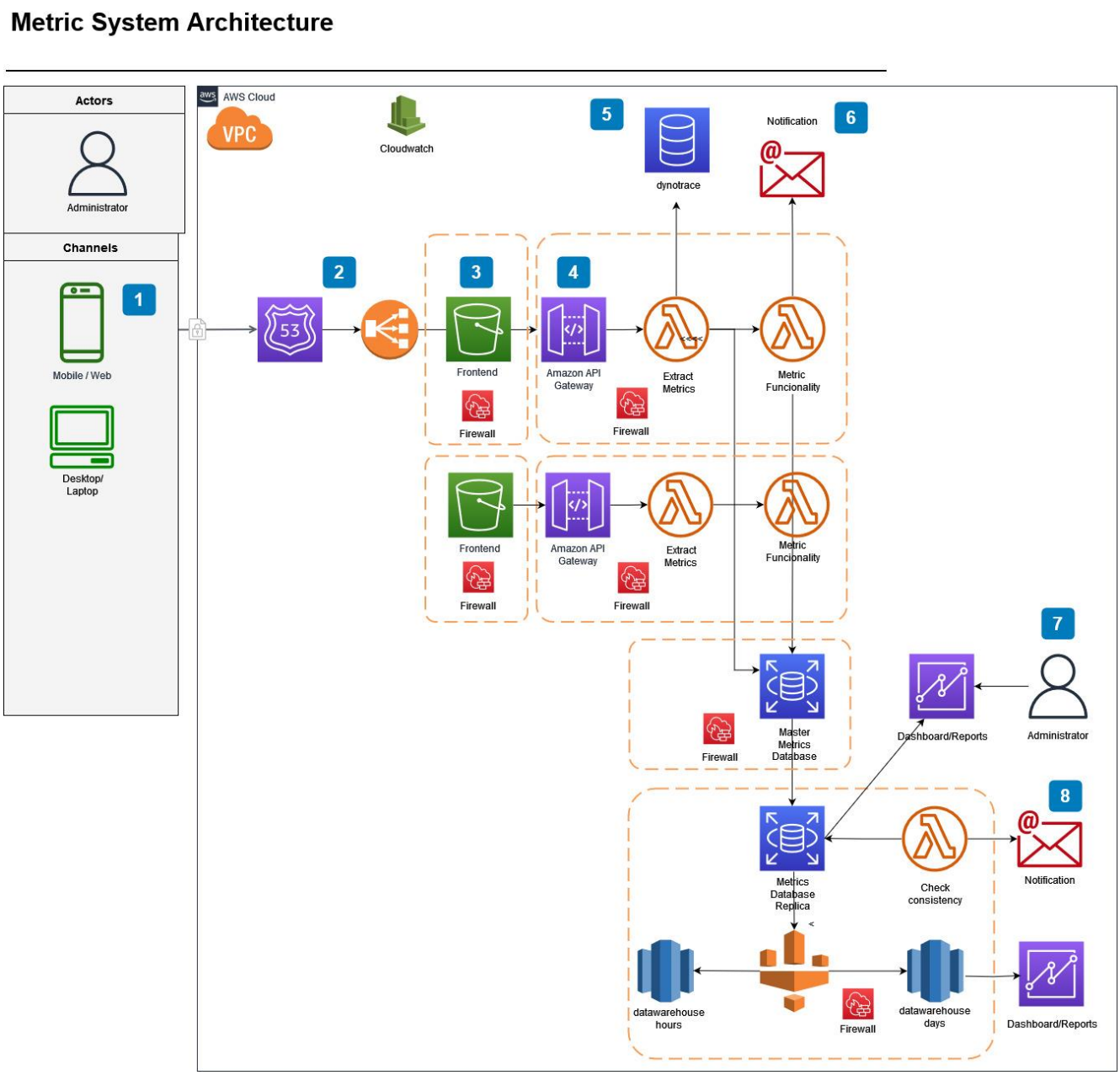
5.3 Base de datos actual



6 Desglose de componentes

La nube elegida fue AWS porque es el proveedor líder y ofrece una gran cantidad de servicios con gran madurez.

6.1 Diagrama de Aws



1. El sistema se accede a través de dispositivos móviles o una aplicación web.
2. El administrador accede mediante una conexión segura utilizando un certificado seguro y también ofrece el servicio de balanceador de carga para gestionar las solicitudes elevadas de los usuarios. En caso de que falle una zona de disponibilidad, hay otra, o en caso de que la carga de trabajo sea grande, se distribuye entre las 2 zonas.
3. El front end está alojado en un bucket S3 proporcionando un bajo costo y excelente rendimiento, también con un firewall para prevenir los ataques de los hackers.
4. El backend está en otra zona con su propio firewall utilizando dos APIs, una para acceder a la base de datos en la nube de Dynatrace con información de varias métricas y la otra con toda la lógica para procesar las métricas.

5. La base de datos de Dynatrace es accedida para extraer las métricas.
6. El sistema envía una notificación al administrador para asegurar que la métrica fue enviada.
7. La información de las métricas se guarda en una base de datos primaria de Mongo con un firewall, y su información se consume en QuickSight en informes en tiempo real.
8. Hay una réplica de la base de datos de métricas a la que se accede para evitar la reducción de la base de datos primaria, a través de una API para verificar cualquier fallo votos de acceso solo para el administrador, todo el sistema está en VPC con una política que hace disponible el front end y el backend, y monitorea los recursos con CloudWatch. También hay un AWS Glue para extraer información a intervalos de 2 horas y días, y esta información puede ser consumida a través de informes.

6.2 Tech Stack

Aplicación backend: para el backend, el lenguaje es go, utilizando el framework gin

Aplicación front-end: React js debido a su versatilidad y gran comunidad y soporte

Aplicación móvil: React Native, porque ya se usa JavaScript para el front-end de la aplicación web, para cubrir todos los dispositivos y reutilizar al equipo de trabajo.

7 Reportes

El sistema ofrece una variedad de informes en tiempo real (A definir)