Ingeniería de Software V

Proyecto final: 24-2

Observabilidad, experimentación y retroalimentación

Prerequisito: Entender la funcionalidad de Grafana, Prometheus (incluyendo Alert Manager y Node Explorer) y Grafana Loki (incluyendo Grafana Agent o Alloy).

Considere que se tiene una estrategia de despliegue básica que tiene un contenedor con un servidor web para el front en javascript, un conetendor para saamfi y otro contenedor para la aplicación de currículo.

Ambiente configurado para gestionar logs, métricas y alertas de un sistema en ejecución.

10% Realizar la instalación y configuración de las herramientas, de tal forma que las métricas de Prometheus y los logs de Grafana Loki se puedan visualizar en Grafana. Puede iniciar desplegando el Loki Quickstart que encuentra en la documentación de Loki.

Modificaciones de aplicación objetivo

La aplicación de gestión curricular cuenta con algunos servicios implementados. Para estos servicios debe implementar las métricas y logs relevantes para tener una mejor observabilidad del sistema. Entre los elementos a observar están la cantidad de solicitudes, elementos creados, duración de las transacciones, cantidad de errores, tiempo de acceso a la base de datos, entre otros.

10% Modificar la aplicación de Currículo para obtener métricas por medio de micrometer exponiendo los end-points info, health y prometheus (revisar ejercicio previo sobre proyecto dummy), e implementando las métricas apropiadas.

10% Escoja algunos servicios objetivo y modifique los servicios a probar para que tengan los logs adecuados para las condiciones exitosas y de error (realice los ajustes necesarios para que estos logs sean estructurados, se recomienda utilizar logstash-logback-encoder). Adicionalmente a la estructura por defecto de los logs estructurados, proponga capos relevantes para el contexto del proyecto.

Despliegue y enlace de servicios

10% Despliegue la aplicación de gestión curricular para utilizar las métricas en prometheus y logs con Grafana Alloy.

10% Definición de dashboards en grafana con las métricas obtenidas.

10% Genere alertas apropiadas teniendo en cuentas las métricas o logs obtenidos. Por ejemplo, para un nivel de error mayor a un límite.

10% Para los servicios escogidos, definir o ajustar las pruebas incluyendo “schedulers” para que las funcionalidades se llamen periódicamente. Revise alternativas para controlar la ejecución de varias instancias simultáneas desde el pipeline.

10% Ejecute diferentes configuraciones de las pruebas y pipelines para evaluar el desempeño de la aplicación en operaciones de consulta y transaccionales. Registre las ejecuciones realizadas y los hallazgos identificados.

10% Teniendo en cuenta la estrategia de despliegue mencionada y los resultados vistos en las métricas, proponga las modificaciones necesarias para el despliegue, la arquitectura y el código que se podrían implementar para mejorar requerimientos no funcionales del sistema de software.

10% Elaboración de informe sobre las configuraciones, modificaciones y ejecuciones realizadas.