Teoría

- ¿Que és?
  - Observabilidad
  - ¿Observability o Monitoring?

¿Que és?

¿Que és?

Observabilidad

La **observabilidad** es la capacidad de comprender la condición o el estado interno de un sistema **mediante el examen de** sus salidas externas especialmente sus **datos.**

¿Por qué es importante la observabilidad? ¿Por qué la necesitamos?

La **observabilidad** es importante porque permite a los equipos evaluar, monitorizar y mejorar el rendimiento de sistemas de IT distribuidos. Cuanto más **observable** es un **sistema**, más **rápido** y con mayor **precisión** pueden los **equipos** de TI pasar de un **problema** de rendimiento **identificado** a su causa raíz sin necesidad de realizar pruebas o codificaciones adicionales.

¿Que necesitamos?

La **observabilidad** requiere de **herramientas** de **software** que **agreguen**, **correlacionen** y **analicen** un **flujo constante de datos** de **rendimiento** de las aplicaciones, el hardware y las redes en las que se ejecutan.

Los **equipos** pueden **utilizar** estos **datos** para **supervisar**, **solucionar problemas** y depurar aplicaciones, redes e infraestructuras, y en última instancia, optimizar la experiencia del cliente y cumplir los acuerdos de nivel de servicio (**SLA**) y otros requisitos empresariales.

¿Cómo funcionan las herramientas de observabilidad?

La **observabilidad** funciona mediante la **recopilación** continua de **datos** de **rendimiento** para crear un **registro** completo y **correlacionado** de cada transacción y solicitud de usuario. Mientras **más** se pueda **observar** el sistema, más **rápida** y **precisamente** puedes **identificar** y **rastrear** un **problema** de **rendimiento** hasta sus **orígenes**.

Los conceptos clave de observabilidad incluyen lo siguiente:

Log's:

Los logs ofrecen registros con marcas de tiempo de los eventos de aplicaciones.

Métricas:

Las métricas son datos temporales que miden el estado del sistema y las aplicaciones, y el rendimiento en el tiempo.

Rastreo distribuido:

El rastreo distribuido brinda un registro del nivel del código integral de cada solicitud de usuario a través de la arquitectura distribuida completa.

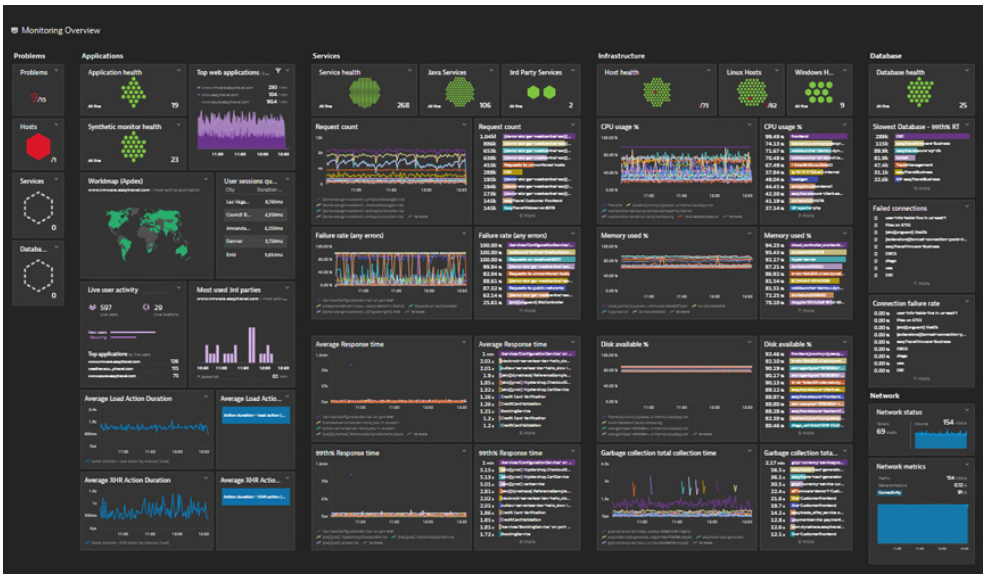
Mapeo de dependencias:

Los mapas de dependencias muestran cómo se conectan las aplicaciones, los servicios y los componentes de la infraestructura.

Detección de anomalías:

La detección de anomalías usa puntos de referencia, análisis estadístico o machine learning para proporcionar una advertencia temprana de comportamiento anormal en un sistema.

En resumen una plataforma de observabilidad correlaciona grandes cantidades de datos de telemetría en tiempo real, lo que brinda a los equipos de DevOps, SRE e IT un panorama contextual completo de cualquier evento o problema.



¿Que és?

¿Observability o Monitoring?

De forma resumida podemos decir que son conceptos relacionados pero con enfoques distintos:

Monitorización

Se centra en **rastrear métricas conocidas y predefinidas,** básicamente respondes a la pregunta: "*¿Está funcionando esto como esperaba?*"

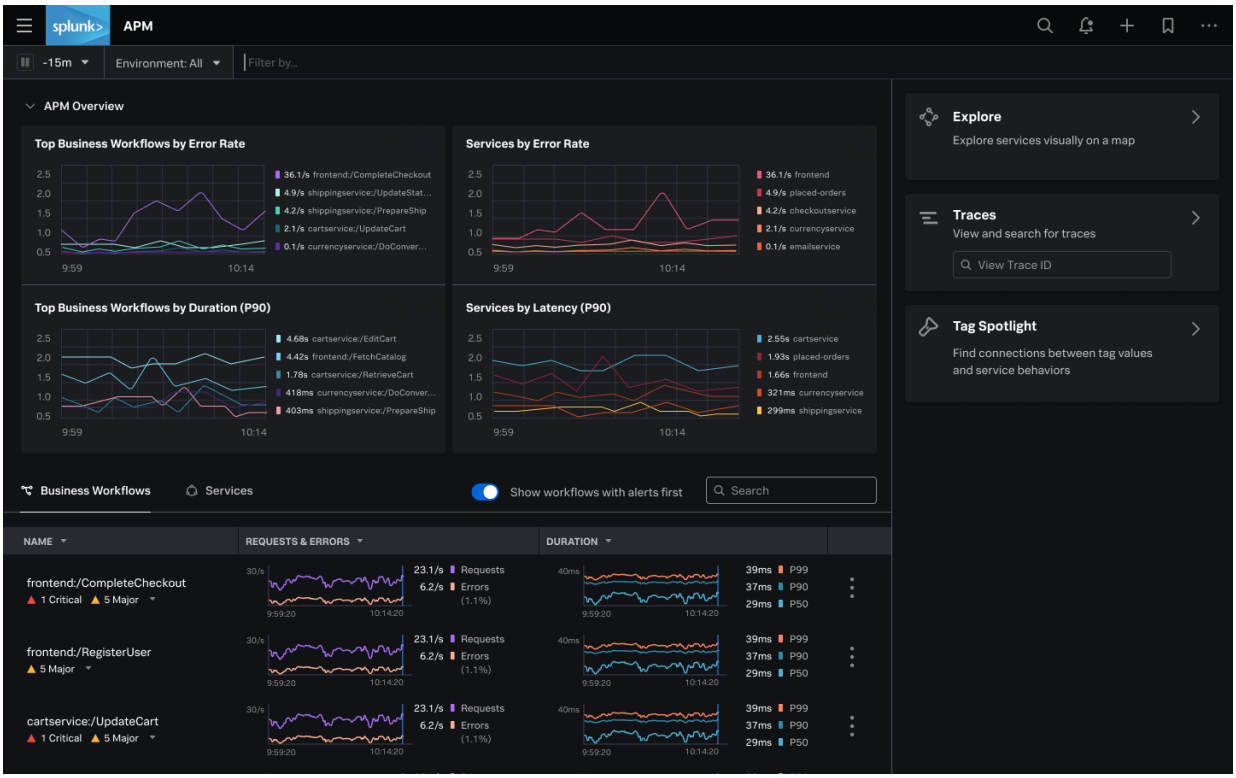
- Defines umbrales y alertas de antemano (CPU > 90%, tiempo de respuesta > 500ms...)
- Sabes qué vas a medir antes de que ocurra el problema
- Ideal para sistemas bien conocidos y fallos predecibles
- Ejemplo: Nagios, Zabbix, Prometheus + Alertmanager

Observabilidad

Se centra en **entender el estado interno de un sistema a partir de sus salidas externas**. La pregunta es: "*¿Qué está pasando exactamente y por qué?*"

- Permite investigar problemas que **no anticipaste**

- Se basa en los tres pilares: **logs, métricas y trazas distribuidas**
- Especialmente útil en sistemas distribuidos y microservicios donde los fallos son más complejos
- Ejemplo: Grafana + Loki + Tempo, Datadog, Jaeger



Conclusión:

En resumen el monitoreo actúa como una alerta temprana y la observabilidad como un diagnóstico profundo para la resolución de problemas (debugging).