Improving Business Insight: Observability Patterns for Software Systems

MobotStone 2023-10-14

Introduction

The complex nature of software systems makes failures inevitable. Rather than attempting to perfect a system without failures, which is incredibly difficult and expensive, it's more practical to take the necessary steps to respond and resolve failures when they occur. The term "observability" wasn't coined by software developers.

> “Observability” is a term from control theory that describes the property of a system where one can infer its internal state by inspecting its external outputs.

The same definition can be applied to enterprise software systems, where we use the following external outputs to infer the internal state of the system:

- log
- index
- track

Logs are the most common way to record the output of enterprise applications. These logs are usually stored in files and rotated after a period of time to maintain a certain storage period.

Metrics are statistics related to the behavior of your application.

Traces are very useful for gaining insight into what is happening in a system at a fine-grained level. Typically, these traces record the delivery of every message in the system.

Implementing Observability

Developers often don't prioritize implementing observability because it requires extra work during the development phase. Constrained by time pressures and other challenges, they often view it as a post-release task or a to-do item that never seems to get completed. That is, until a serious production issue arises, forcing support engineers to struggle to find the root cause. By then, the problem has already occurred, and significant time has been wasted due to a lack of observability in the system.

The four main steps to implementing observability include:

- Instrumentation
- Correlation
- automation
- Insights and predictions

Instrumentation is the first step to implementing observability, where applications need to generate necessary telemetry data from the source code so that data collectors can ingest and aggregate this data for further analysis.

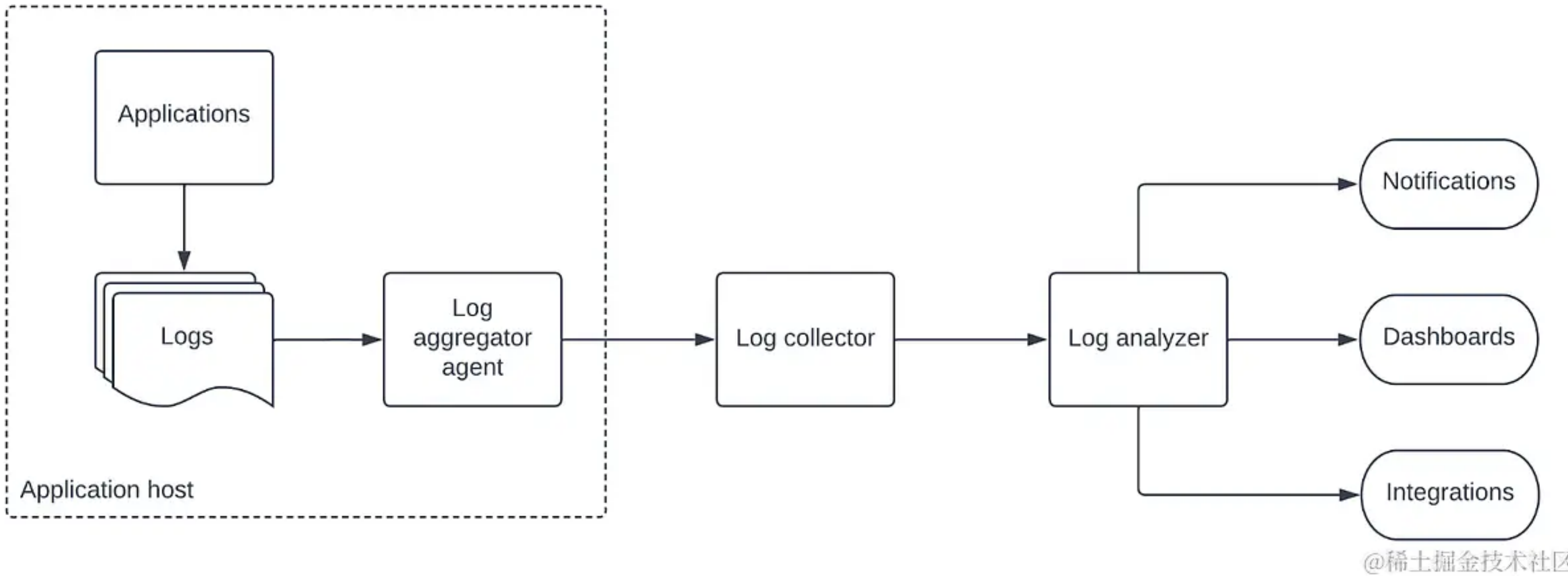
Once the data is collected and aggregated, there must be a mechanism to correlate telemetry data from different events across different applications to troubleshoot issues and identify the root cause. It is important to have a common approach across different applications to correlate these events.

Allocating resources to review every telemetry event and make decisions based on it is an impractical task. Instead, we need to automate the process of analyzing these events as much as possible, so that only those events that require attention trigger human intervention.

Furthermore, in addition to responding to critical incidents, we can also use observability data to analyze user behavior, generate insights and predictions to support business decisions to improve system performance and business.

Using logs for observability

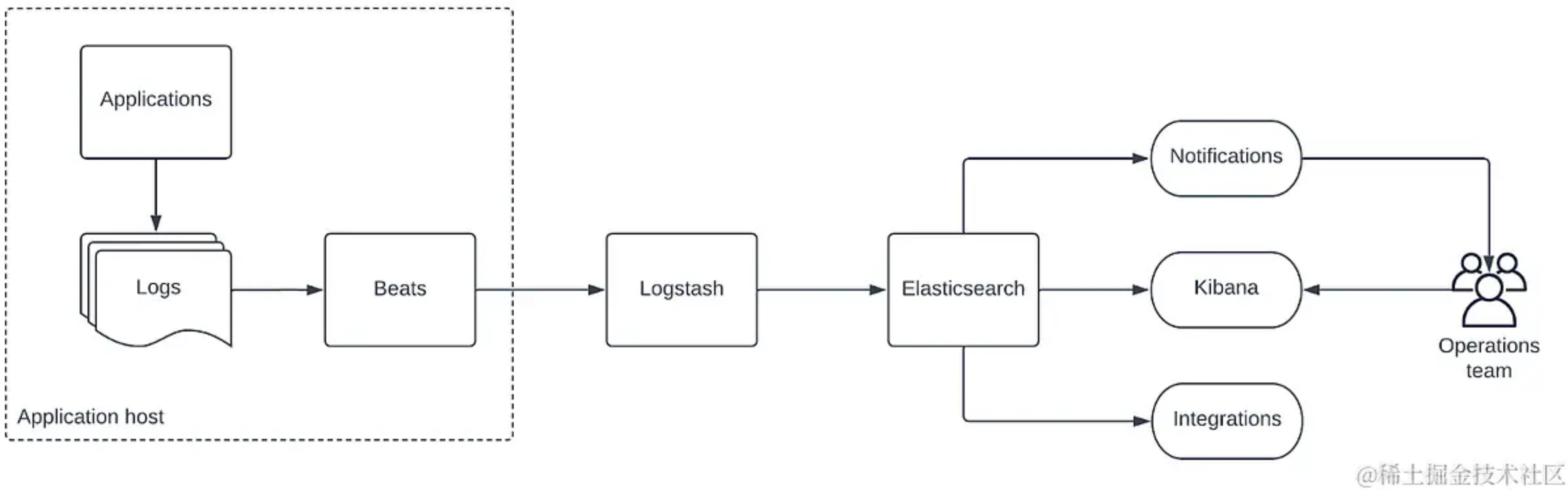
Logging is the most common and popular method for troubleshooting enterprise applications. Applications can be instrumented to output different types of log entries, such as errors, warnings, debug messages, and informational details, to these log files. The following diagram shows a typical pattern for using logs for observability.



As shown in the previous figure, different types of applications expose their internal state as external output through log entries. These logs can be aggregated or read by agents running next to the applications. The main task of these agents is to read these log entries and publish them to log collectors. Log collectors are responsible for aggregating these log entries and preprocessing them before further analysis. With log analyzers, users can choose to analyze these logs directly using a query language or use external dashboard components for analysis. Some popular log analysis tools include:

- ELK（Elasticsearch、Logstash、Kibana）
- Grafana Labs（Promtail、Loki 和 Grafana）
- Splunk
- New Relic
- Sumo Logic

We can use the ELK stack to implement the observability pattern as shown below.

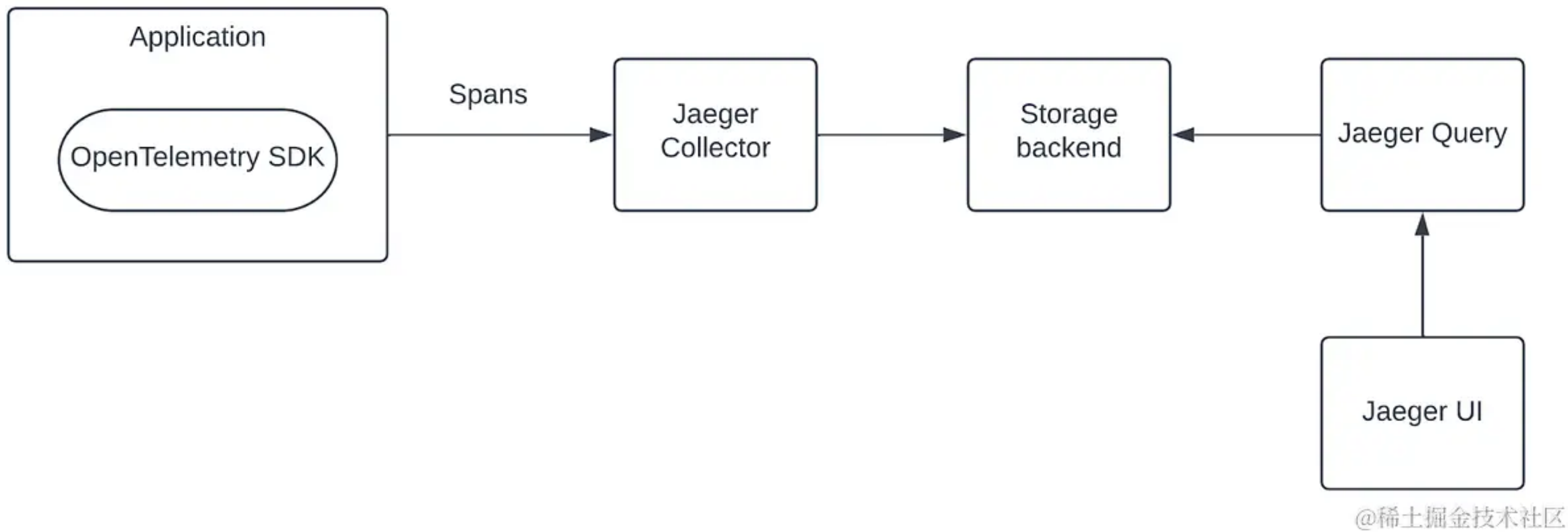


As shown in the previous figure, the Beats agent, located next to the application, reads log files and transmits these log entries to Logstash, which acts as a log aggregator. Logstash then stores these aggregated logs in its unique storage and indexes, searches, and analyzes the data in the log files. Elasticsearch is a powerful tool capable of analyzing various types of data, including structured and unstructured data, as well as numerical, textual, and geospatial data, allowing users to extract valuable insights and context from the data. Kibana allows users to interactively explore, visualize, and share insights, and monitor systems through visual dashboards.

Using tracing for observability

Tracing is another way to implement observability for enterprise applications. In this case, we use common standards such as OpenTelemetry or OpenTracing to publish detailed information about the data transmitted through the application.

Jaeger is an open-source distributed tracing platform used to implement observability solutions for modern cloud-native applications. It helps operations teams capture trace data across multiple applications and use this information to troubleshoot issues and improve system performance. The following diagram illustrates how to implement tracing observability using Jaeger.



The diagram above depicts a use case where an application is intelligently leveraging an OpenTelemetry-based SDK to instrument the application and publish telemetry data as a series of "spans" to a Jaeger collector. During this process, the Jaeger collector validates and transforms the data before storing it. Possible storage backends include in-memory storage, Elasticsearch, Kafka, or a database. Once the data is stored in the backend storage, the Jaeger query component is used to perform search and query operations to retrieve the trace data, which is then visualized in the Jaeger user interface.

Label: Backend architecture topic: Jinshi Project Essay Competi...

Comments 0

Login/ Register Post your comment!

No comment data yet

Table of contents

Close

Introduction

Implementing Observability

Using logs for observability

Using tracing for observability

Search suggestions

Search keywords

to Observability Introduction

using Observation Cloud Building business observability

Beyond Monitoring: How Observability 2.0 Revolutionizes the Developer Experience

Modern Observability Platform (3)

How AI Log Analysis Can Shape of Observability the Future

based on ASM Simplify observability management and enhance business insights

Monitoring and Observability : Uncovering the Truth

Data Observability The of Components

Modernizing Observability : The Shift from DIY ELK to SaaS

Observability: Observability Monitoring is Different from 3 Reasons Why

Featured Content

Continue Vibe Coding Tools: Markdown Writing + One-Click Publishing

Programmer DD · 55 reads · 0 likes

Hadoop Data Governance Practice: Metadata Management and Data Quality Assurance

Homi · 16 Reads · 1 like

Exceptions that need to be uniformly intercepted by the backend (applicable to nestjs)

atwednesday · 31 reads · 1 like

How to efficiently handle complex data structures passed by ARGV in Lua scripts?

IT Orange Peel · 18 Reads · 1 like

What are the best practices for ARGV and KEYS in Redis Lua scripts?

IT Orange Peel · 17 Reads · 0 likes

为你推荐

使用 OpenTelemetry 构建 .NET 应用可观测性（1）：什么是可观测性

黑洞视界 1年前 196 点赞 评论 后端

可观测性与传统监控的区别和联系

Vicla 1年前 300 3 评论 运维

Observability：什么是可观察性?

Elasticsearch 4月前 159 点赞 评论 Elasticsearch

基于 ASM 简化可观测管理、提升业务洞察力

阿里云云原生 2年前 577 点赞 评论 云原生

云原生时代如何用 Prometheus 实现性能压测可观测-Metrics 篇

阿里云云栖号 3年前 663 1 评论 云原生

什么是可观测性（翻译）

白狼杰洛特 1年前 266 1 评论 后端

《SRE运维闭环：从故障诊断到事后总结的Google实践全景》

柠檬树结柚子 1月前 46 点赞 评论 运维

了解可观察性指标：类型、黄金信号和最佳实践

Elasticsearch 5月前 74 点赞 评论 Elasticsearch

《数据可观测性的基础原理》第二章：数据可观测性的组成部分

数据智能老司机 2年前 779 1 评论 大数据 数据分析 架构

基于 eBPF 的 Kubernetes 可观测实践

阿里云云原生 3年前 1.9k 2 评论 云原生 监控 容器

云原生时代如何用 Prometheus 实现性能压测可观测-Metrics 篇

阿里云云原生 3年前 2.3k 3 1 Kubernetes

可观测性驱动软件研发效能提升

Harlon 2年前 3.6k 4 评论 云原生 DevOps

别再删库跑路了，快来读《SRE生存指南》

已注销 4年前 532 2 评论 运维

从运维告警到业务决策：可观测性正在重新定义企业数据基础设施

可观测性用观测云 4月前 69 点赞 评论 人工智能 监控

编译时插桩，Go 应用监控最佳选择