

Observability

Observability is the ability to understand the internal state of a system by observing its external outputs.

!qownnotes-media-drHnRi(media/qownnotes-media-drHnRi.png)

Observability (observability) is the ability to understand the internal state of a system by observing its external outputs?

- Observability is the ability to understand the internal state of a system by observing its external outputs.
- Observability is the ability to understand the internal state of a system by observing its external outputs. (ef. Observability is the ability to understand the internal state of a system by observing its external outputs.)
- Observability is the ability to understand the internal state of a system by observing its external outputs.
- Observability is the ability to understand the internal state of a system by observing its external outputs.

Observability is the ability to understand the internal state of a system by observing its external outputs.

- Observability is the ability to understand the internal state of a system by observing its external outputs.
- Observability is the ability to understand the internal state of a system by observing its external outputs.
- MELT(Metric / Event / Log / Trace) is the ability to understand the internal state of a system by observing its external outputs.
- Observability is the ability to understand the internal state of a system by observing its external outputs.

Observability is the ability to understand the internal state of a system by observing its external outputs.

- Observability(cardinality) - the ability to understand the internal state of a system by observing its external outputs.
- Observability is the ability to understand the internal state of a system by observing its external outputs.
- Observability is the ability to understand the internal state of a system by observing its external outputs.
- Observability is the ability to understand the internal state of a system by observing its external outputs.
- Observability is the ability to understand the internal state of a system by observing its external outputs.
- Observability is the ability to understand the internal state of a system by observing its external outputs.

Observability is the ability to understand the internal state of a system by observing its external outputs.

- Metric(Metric)

- 애플리케이션 로그 데이터 수집을 자동화합니다.
- 애플리케이션 로그 데이터 수집을 용이하게 합니다

2. 로그(Log)

- 애플리케이션 로그 데이터 수집을 용이하게 합니다.
- 애플리케이션 로그 데이터 수집을 용이하게 합니다. 애플리케이션 로그 데이터 수집을 용이하게 합니다, 애플리케이션 로그 데이터 수집을 용이하게 합니다 / 애플리케이션 로그 데이터 수집을 용이하게 합니다
- 애플리케이션 로그 데이터 수집을 용이하게 합니다.
- 애플리케이션 로그 데이터 수집을 용이하게 합니다. (애플리케이션 로그 데이터 수집을 용이하게 합니다)

3. 이벤트(Event), Newrelic / Redhat와 같이 ** - 애플리케이션 로그 데이터 수집을 용이하게 합니다

- 애플리케이션 로그 데이터 수집을 용이하게 합니다, 애플리케이션 로그 데이터 수집을 용이하게 합니다

4. 추적(Trace)

- 애플리케이션 로그 데이터 수집을 용이하게 합니다.
- 애플리케이션, 로그, 애플리케이션 로그 데이터 수집을 용이하게 합니다, 애플리케이션 로그 데이터 수집을 용이하게 합니다.

애플리케이션 로그 데이터 수집을 용이하게 합니다

애플리케이션 로그 데이터 수집을 용이하게 합니다

1. 애플리케이션 로그 데이터 수집을 용이하게 합니다. 애플리케이션 로그 데이터 수집을 용이하게 합니다, 애플리케이션 로그 데이터 수집을 용이하게 합니다
2. 애플리케이션 로그 데이터 수집을 용이하게 합니다(MTTR)를 용이하게 합니다
3. 애플리케이션 로그 데이터 수집을 용이하게 합니다. (애플리케이션 로그 데이터 수집을 용이하게 합니다)
4. 애플리케이션 로그 데이터 수집을 용이하게 합니다
5. 애플리케이션 로그 데이터 수집을 용이하게 합니다

애플리케이션 로그 데이터 수집을 용이하게 합니다

1. 애플리케이션
 - 애플리케이션 로그 데이터 수집을 용이하게 합니다
 - 애플리케이션 로그 데이터 수집을 용이하게 합니다
 - 애플리케이션 로그 데이터 수집을 용이하게 합니다
2. 애플리케이션
 - 애플리케이션 로그 데이터 수집을 용이하게 합니다
 - 애플리케이션 로그 데이터 수집을 용이하게 합니다
 - 애플리케이션 로그 데이터 수집을 용이하게 합니다

APM/DPM과 같이 애플리케이션 로그 데이터 수집을 용이하게 합니다

1. 애플리케이션 APM/DPM과 같이 애플리케이션 로그 데이터 수집을 용이하게 합니다
2. APM/DPM과 같이 애플리케이션 로그 데이터 수집을 용이하게 합니다 (애플리케이션, 로그, 애플리케이션 로그 데이터 수집을 용이하게 합니다), UX와 같이 애플리케이션 로그 데이터 수집을 용이하게 합니다
3. APM/DPM과 같이 애플리케이션 로그 데이터 수집을 용이하게 합니다, 애플리케이션 로그 데이터 수집을 용이하게 합니다.

Tool

애플리케이션 로그 데이터 수집을 용이하게 합니다

애플리케이션 로그 데이터 수집을 용이하게 합니다, Whatap, New Relic, Elastic, IBM(Redhat), ServiceNow, 애플리케이션

Opentelemetry

- observability란 OSS, 다양한 데이터
- 지표 : 프로메테우스(prometheus)
- 로그 : 엘라스틱(로그/트레일)
- 트레이스 : 오픈테lemetry(오픈트레이스, E/S)
- 2022년 GA예정인 오픈(오픈트레이스, 오픈트레이스)은 API를 통해 데이터를 API로 수집하고

Appendix#1. □□□□ □□ □□

--	--

1. MTBF(Mean Time Between Failures) : 平均故障時間，即兩次故障之間的平均時間。計算公式為： $\text{MTBF} = \frac{\text{總運行時間}}{\text{故障次數}}$ 。例如：如果設備在2400小時內發生了2次故障，則其MTBF為： $\frac{2400}{2} = 1200$ (小時)
 2. MTTR(Mean Time To Repair) : 平均修復時間，即從故障發生到修復完成的平均時間。計算公式為： $\text{MTTR} = \frac{\text{總修復時間}}{\text{故障次數}}$ 。例如：如果設備在2400小時內發生了2次故障，每次故障需要1200小時修復，則其MTTR為： $\frac{2400}{2} = 1200$ (小時)
 3. MTTA(Mean Time To Acknowledge) : 平均響應時間，即從故障發生到技術人員開始處理的平均時間。計算公式為： $\text{MTTA} = \frac{\text{總響應時間}}{\text{故障次數}}$ 。例如：如果設備在2400小時內發生了2次故障，每次故障需要1200小時響應，則其MTTA為： $\frac{2400}{2} = 1200$ (小時)
 4. MTTD(Mean Time to Detect) : 平均檢測時間，即從故障發生到被發現的平均時間。
 5. MTTF(Mean Time To Failure) : 平均失效時間，即從設備投入使用到最終失效的平均時間。
- MTTA、MTTR、MTTD、MTTF等指標都是衡量系統可靠性的關鍵因素，MTTA和MTTR是衡量系統響應速度的重要指標，而MTBF和MTTF則是衡量系統穩定性和耐用性的核心指標。

Appendix #2. □□□□□□ □□□

1. 数据组织
 - 数据组织采用 TSDB(Time Series DataBase)数据库组织数据, LRU数据淘汰策略, 对数据按照时间进行分区
 - 数据按照时间分区, 数据按照时间分区进行存储, 数据按照时间分区进行存储
 - 数据按照时间分区(Chunk)进行存储, 数据按照时间分区进行存储
 - 数据按照时间分区进行存储
 - 数据按照时间分区进行存储
2. 数据组织 hierarchical

```
./data
-abc
-def      # □□□□
- chunks  # □□□□
  - 1
- tombstones # □□ □□ □□□□
- index     # □□□ □□ □□□□ □□□□ □□□ □□
- meta.json # □□□ □□□□□
- checks_head # □□ □□
  - 1
- wal
- 001
  - checkpoint.001 # wal □□
    - 000          # □□□ □□ □□□□□ wal□□
```

- WAL(Write Ahead Logging)은 모든 데이터 변경을 로그에 먼저 기록한 후, 데이터베이스에 반영하는 방식이다. WAL은 데이터베이스의 성능과 안정성을 높여준다. WAL은 데이터베이스의 성능을 높여준다. WAL은 데이터베이스의 성능을 높여준다.

1. `storage.tsdb.min-block-duration` : 블록 생성 간격, 2h로 설정하면 2시간마다 블록이 생성된다.

2. 部署 環境
- 環境 構築 手順 参照

Appendix#3. 環境 構築

1. 環境 構築 3ステップ
- ステップ 1 : 環境 構築, 設定
 - ステップ 2 : 環境 構築 完了 確認 後 環境 構築 完了 確認
2. 環境 構築 後 : 環境 構築 完了 確認 後 環境 構築 完了 確認 後. 環境 opentelemetry 環境 構築 完了 確認 後. 環境 構築 完了 確認 後 環境 構築 完了 確認 後 efk stack 環境 構築 完了 確認 後 efk stack 環境 構築 完了 確認 後

環境 構築 完了 確認 後 環境 構築 完了 確認 後

1. 環境 Tool 環境 構築
2. 環境 構築 完了 確認

Reference

- [環境 構築](#)
- [Elastic](#)
- [New Relic](#)
- [環境 構築](#)
- [IBM](#)
- [ServiceNow](#)
- [Redhat](#)
- [環境 構築](#)