



New Relicを活用した SREの最初のステップ

2024.11.19

New Relic User Group(NRUG) OKINAWA VOL.3

@isaoshimizu



清水 勲 (X: @isaoshimizu)

家族アルバム みてね Engineering Manager (SRE/CRE/セキュリティ領域)

新卒入社

ミクシィ (現MIXI) 入社

2022年1月～EM

Sler時代 (受託・自社開発)

SNS 「mixi」

モンスター
ストライクなど

みてね

2003年

2011年

2014年

2018年

2024年

おもに関わった技術

C/C++/C#/PHP/Python/iOS/AWS

Fedora/MySQL/LXC/
OpenStack

Linux/MySQL/Ruby

AWS/MySQL/Ruby

- New Relic User Group 運営メンバー
- AWS Summit Tokyo 2014/2019、SRE NEXT 2020/2022、その他SRE関連勉強会に多数登壇
- 技術評論社「Software Design」寄稿、オライリー・ジャパン「SREをはじめよう」翻訳レビュー
- 週末は社会人吹奏楽団での活動（楽団長、トロンボーン約30年、たまに指揮者）、キャンプとクラフトビールが好き

本セッションの主な対象者

- SREってなに？という方
- SRE初学者の方
- SREをこれから始める方、始めたばかりの方

もちろん

- SREをすでに実践されている方

也大歓迎です！！

1. SREとは
2. システムの信頼性を知る
3. New RelicはSREにどう役に立つ？
4. まとめ

SREとは

2024年10月にオライリー・ジャパンより出版された「SREをはじめよう」※では、SREとは
“組織がシステム、サービス、製品において適切なレベルの信頼性を持続的に達成できるように支援することを目的とした工学分野”
と表現している。

これを3つに分解してみる

1. システム、サービス、製品（プロダクト）のためのものである
2. 適切なレベルの信頼性
3. 持続的に達成できるように支援する

みなさんが関わっているシステム、サービス、プロダクトの
適切なレベルの信頼性とはどのくらいなのか？
それを持続的に達成できるように支援するためになにをすればよいのか？

※SREをはじめよう 一個人と組織による信頼性獲得への第一歩 David N. Blank-Edelman 著、山口 能迪 訳
<https://www.oreilly.co.jp/books/9784814400904/>

適切な信頼性レベルの設定

- ビジネスとして成り立つ信頼性レベルはどのくらいだろうか？
 - 言い換えれば、ユーザーの満足度を維持できるレベルとはどのくらいか？
- 高ければ高いほど良いってものではない
 - 100%の信頼性レベルは非現実的、バグや障害は起こり得るものという前提
 - 過度な目標設定だと、様々な機能開発や構成変更などができなくなってしまう
- ビジネスサイドも含めて一緒に目指すべき信頼性レベルはどの程度かを決める（会話する）
 - 最初はざっくりでもよい
 - 現状大きな問題が起きてないのであれば、今と同レベルという考え方でもよい
 - あとから目標を修正していければ良い

まずは現状の信頼性レベルを知る（計測する）ところから

システムの信頼性を知る

信頼性をどうやって知るのか？

ユーザー体験に影響する箇所において

- どのくらいエラーが出ているのか、その原因となりうる箇所はどこか
- どのくらいスループットが出ているのか
- どのくらいのレスポンスタイムで返せているのか

を把握することがまず第一歩。その方法として

- メトリクスを収集、集計する
- ログを収集、集計する

というのが挙げられる

信頼性を計測するために必要な情報を収集・記録し集計する

メトリクス

- CloudWatch
- Cloud Monitoring
- Prometheus
- Grafana

ログ

- CloudWatch Logs
- Cloud Logging
- Fluentd/Fluent Bit → Data Firehose → S3/OpenSearch Service/Athena
- Fluentd/Fluent Bit/Promtail → Grafana Loki → S3
- BigQuery

ほかにもいろいろ

- New Relic
- Datadog
- Dynatrace
- AppDynamics
- Sentry
- Splunk
- Mackerel
- Grafana Cloud
- Elastic Observability

ほかにもいろいろ

SaaSはメトリクス、ログ含め様々な情報を一元的に収集・集計できるというメリット

New RelicはSREにどう役に立つ？

New Relicには多くの機能がありますが、最低限これだけ覚えておくと良いもの

- APM (Application Performance Monitoring)
 - サーバーアプリのパフォーマンスモニタリング
- Infrastructure
 - サーバー、コンテナ、データベース、Kubernetesのモニタリング
- Mobile
 - モバイルアプリのパフォーマンス、クラッシュのモニタリング
- Browser
 - Webサイトのパフォーマンスモニタリング
- Log Management
 - ログの一元化、検索、可視化、アラート

あらゆるアプリケーション・インフラの状態を一元的に収集し、検索、可視化できる

New Relic APMでオブザーバビリティの第一歩

特にサーバーアプリケーションの状態を知ることがオブザーバビリティの第一歩。

サーバー1台ずつにSSHしてログをgrep・・・なんて面倒なことはしたくありませんよね？

(面倒だけでなく欲しい情報を探す難しさ、見逃してしまうといった課題がある)

APMを導入することですべてのサーバーで動作しているアプリケーションの状態を知ることができる！

(いちいちSSHはしなくていい！)

New Relic APMはGo、Java、.NET、Node.js、PHP、Python、Rubyに対応（ExperimentalとしてElixirも対応している様子）

APM導入手順は公式ドキュメントを参照

[APMによるアプリのパフォーマンス改善 | New Relic Documentation](#)

The screenshot shows the New Relic website's documentation page for APM. The main heading is 'APMによるアプリのパフォーマンス改善' (Improving application performance with APM). The text explains that APM provides visibility into application performance, helping to identify and resolve issues before they impact users. It lists several benefits: early detection of issues, faster resolution, and improved user experience. The page also includes a sidebar with navigation links and a footer with logos for various supported languages and frameworks.

オブザーバビリティを確保することによって

アプリケーションの状態が

「なんとなく良さそう...」 「たまになんかおかしい...」

から

「このレスポンスタイム値なら大丈夫だ」 「このエラー今日xx回出てるから直そう」

に変わる

オブザーバビリティによって現在の信頼性を持続的に把握できる。
必要に応じて改善のアクションにつなげられる（これぞSRE）。

まとめ

- いま扱っているシステムの信頼性がどの程度なのかを知ることがSREの第一歩
- 信頼性を測るためにはオブザーバビリティは欠かせない
- オブザーバビリティツールは世の中に数多くある
 - なにを選ぶかは要件（機能やコストなど）に合わせて
- APMの導入は第一歩目としてオススメ
 - 中規模～大規模なサービスだとデータ量（コスト増）に注意
- SREのプラクティスとしては、他にもインシデントレスポンス、ポストモータム、トイルなどがあるが「**目指すべき信頼性レベルを決める、今の状況を知る**」ためのオブザーバビリティをまず整えておきたい
- New Relicのノウハウは世の中に増加中！
 - 特にQiitaのアドベントカレンダーには多くのTipsが溢れているのでぜひご覧ください

心もつなごう。

MIXI

