



インフラエンジニアからSREへの変革

Tsuyoshi SHIMIZU

Senior Solutions Consultant

New Relic KK

Safe Harbor

This presentation and the information herein (including any information that may be incorporated by reference) is provided for informational purposes only and should not be construed as an offer, commitment, promise or obligation on behalf of New Relic, Inc. (“New Relic”) to sell securities or deliver any product, material, code, functionality, or other feature. Any information provided hereby is proprietary to New Relic and may not be replicated or disclosed without New Relic’s express written permission.

Such information may contain forward-looking statements within the meaning of federal securities laws. Any statement that is not a historical fact or refers to expectations, projections, future plans, objectives, estimates, goals, or other characterizations of future events is a forward-looking statement. These forward-looking statements can often be identified as such because the context of the statement will include words such as “believes,” “anticipates,” “expects” or words of similar import.

Actual results may differ materially from those expressed in these forward-looking statements, which speak only as of the date hereof, and are subject to change at any time without notice. Existing and prospective investors, customers and other third parties transacting business with New Relic are cautioned not to place undue reliance on this forward-looking information. The achievement or success of the matters covered by such forward-looking statements are based on New Relic’s current assumptions, expectations, and beliefs and are subject to substantial risks, uncertainties, assumptions, and changes in circumstances that may cause the actual results, performance, or achievements to differ materially from those expressed or implied in any forward-looking statement. Further information on factors that could affect such forward-looking statements is included in the filings New Relic makes with the SEC from time to time. Copies of these documents may be obtained by visiting New Relic’s Investor Relations website at ir.newrelic.com or the SEC’s website at www.sec.gov.

New Relic assumes no obligation and does not intend to update these forward-looking statements, except as required by law. New Relic makes no warranties, expressed or implied, in this presentation or otherwise, with respect to the information provided.

- Package Vender - E-Commerce **Infra**
- Package Vender - E-Commerce **SaaS SRE**
- Public Cloud - **SaaS** Solutions Architect
- New Relic - Solutions Consultant



New Relic®

清水 毅
Senior Solutions Consultant
New Relic



**More Perfect
Software.**



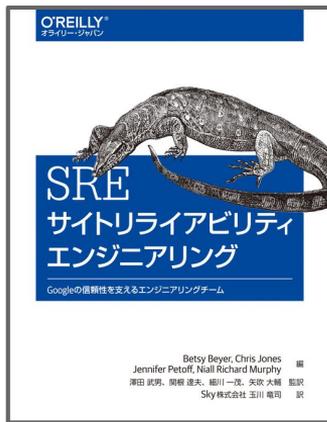
本セッションの対象者とゴール

対象者:

インフラエンジニアとSREについての違い
を学びたいエンジニア
SREへ変貌したいエンジニア

ゴール:

- SREの基本を理解
- インフラエンジニアがSREへの一歩を踏み出せる



Google : Site Reliability Engineering
<https://landing.google.com/sre/books/>

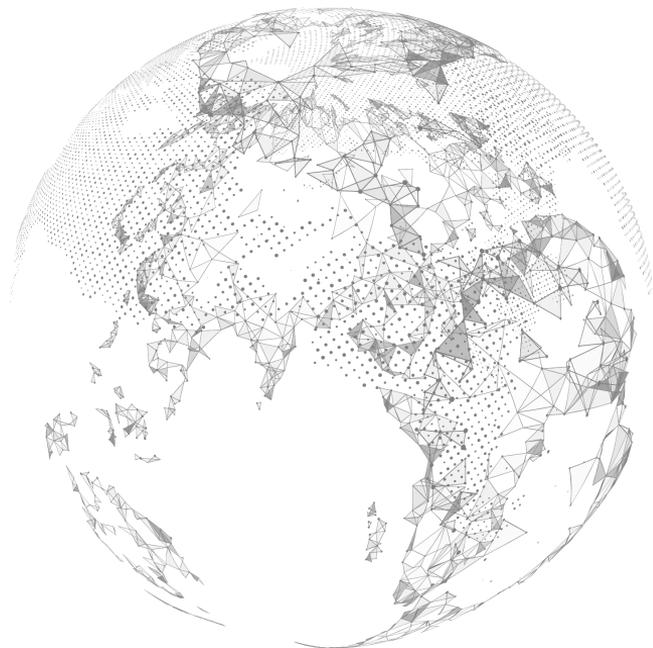
セッションの流れ

1. SRE の基本
2. インフラエンジニアと SREの違い
3. SRE へと変革していく3ステップ
4. New Relic One のデモ



SREの基本

デジタル変革は全企業の最優先課題



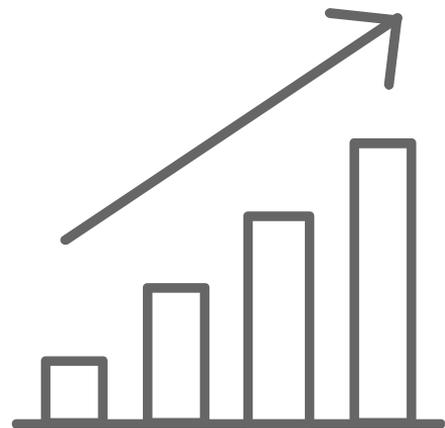
2022年までに
デジタルサービスが
GDPに占める割合

60%以上

IDC Future Scope: Multiplied Innovation Takes Off, Powered by AI, Distributed Public Cloud, Microservices, Developer Population Explosion, Greater Specialization and Verticalization, and Scaling Trust

ビジネスアジリティの必要性

- デジタルサービスを用いてビジネスを推進する必要性の高まり
- 新しいデジタルビジネスへの移行の要求
- ビジネスの迅速な市場への投入



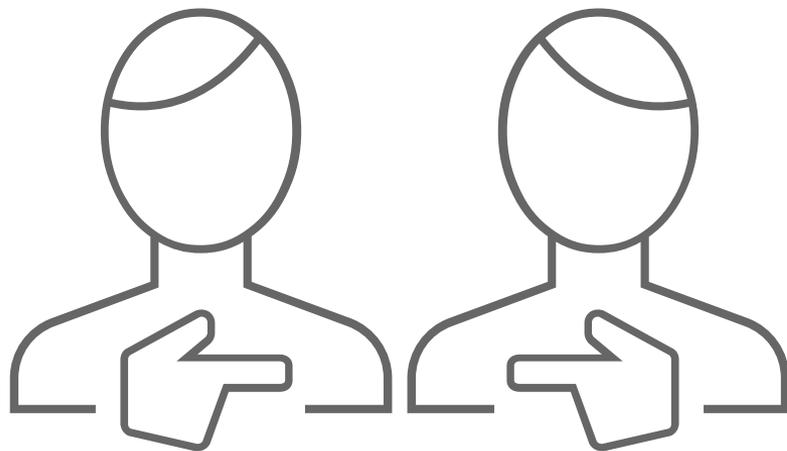
開発と運用の対立

新しい機能をできるだけ多くリリースしたい開発 (Dev)

と

システムを安定化させたい運用
(開発運用やインフラ運用)

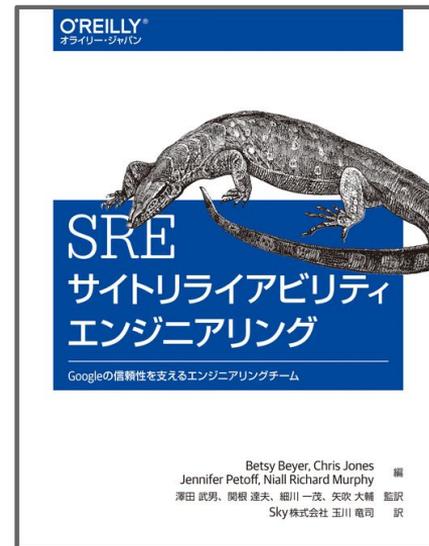
ミッションの違いにより
たびたび対立構造に



DevOpsとは

「DevOps」という言葉は、2008年の終わり頃から業界で使われ始め、本書の執筆時点（2016年初頭）の時点でも、その意味するところは未だに固まっていません。

その中核となるのは、システムの設計と開発の各フェーズにおいて IT に役割を持たせること、人手ではなく自動化に頼ること、運用タスクにエンジニアリングのプラクティスとツールを適用することなど ”



本セッションでの定義:

DevOpsとは文化・ツール・プラクティスの3つで構成され、
開発(Dev)と運用(Ops)のフィードバックサイクルの高速化を行うこと

ACCELERATE

State of DevOps

2019

Sponsored by

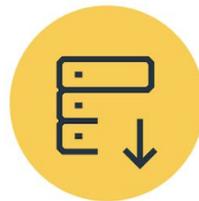


DevOps の Key Metrics



ソフトウェア開発

リードタイム



ソフトウェアデプロイメント

変更失敗



サービス運用

可用性

デプロイ頻度

復旧時間

FOUR KEY METRICS

成長企業とその他の企業の差



208

TIMES MORE

frequent code deployments

デプロイ頻度

106

TIMES FASTER

lead time from
commit to deploy

リードタイム



2,604

TIMES FASTER

time to recover from incidents

復旧時間

7

TIMES LOWER

change failure rate
(changes are $\frac{1}{7}$ as likely to fail)

変更失敗



Throughput Stability

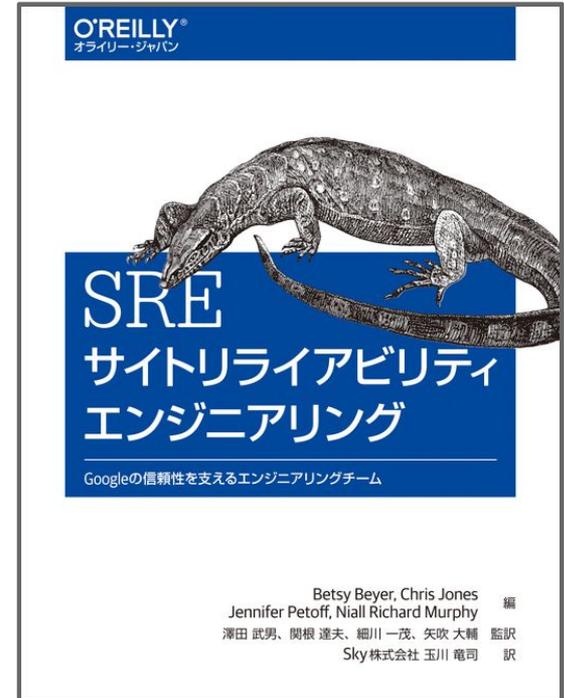
SRE (Site Reliability Engineering) とは

SRE とは “ソフトウェアエンジニアが 運用を設計したらどうなるのか？” ということ

- Benjamin Treynor Sloss, Vice President, Engineering, Google

- “SREはシステムの信頼性に重きをおく”
- “SREとはDevOpsに独自拡張を加えた1プラクティスと考えることもできる”
- “一般的に、SREチームは可用性、レイテンシー、パフォーマンス、効率、変更管理、監視、緊急対応、サービスのキャパシティプランニングに責任をもつ”

※“運用にかけていいリソースは活動の50%まで、50%以上は信頼性を向上するためのコード生成を行うこと”

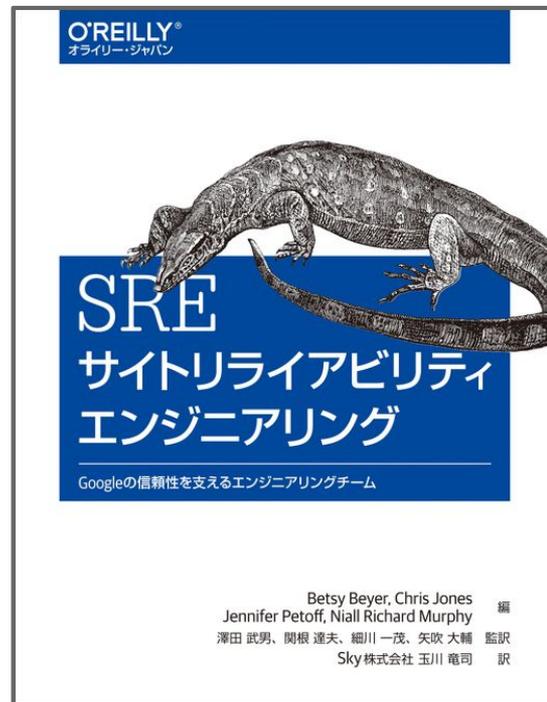


サービスの信頼性とは

“ここでは、信頼性とは「[システムが]求められる機能を、定められた条件の下で、定められた期間にわたり、障害を起こすことなく実行する確率」です。これは[Oco12]の定義に従っています。”

Oco12:P. O'Connor and A. Kleyner, 「Practical Reliability Engineering」, 5th edition: Wiley, 2012.

サービスの信頼性とは
“障害を起こすことなく
実行する確率”



インフラエンジニアとSREの違い

DevOps の Key MetricsとSRE

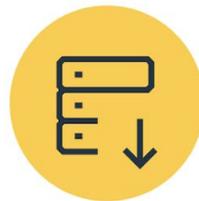


ソフトウェア開発

リードタイム

Dev

デプロイ頻度



ソフトウェアデプロイメント

変更失敗

Ops



サービス運用
SRE

可用性

downtime
Infra

FOUR KEY METRICS

インフラエンジニアとSREの違い

インフラエンジニア

- インフラの運用・監視
- インフラの安定

- スキル(例)
 - インフラ
 - サーバーサイド
 - モニタリング
 - トラブル対応
 - etc

SRE

- ソフトウェアで運用を設計
- サービス信頼性・変更速度最大化

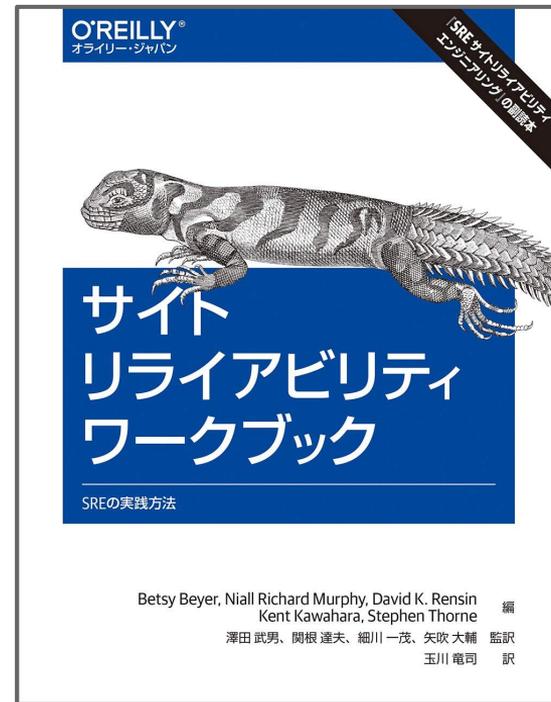
- スキル(例)
 - インフラ
 - サーバーサイド
 - (モダン)モニタリング
 - トラブル対応
 - プログラミング
 - etc

SREへ変革していく3ステップ

『サイトリライアビリティワークブック』

SREの行動の基礎となる原則と理論をのべた『SREサイトリライアビリティエンジニアリング』本の実践編としての副読本。

SREを組織やプロジェクトで導入するにあたり、必要となる具体的な方法や手順を解説



<https://www.amazon.co.jp/dp/4873119138>

インフラエンジニアがSREへ変革していく 3ステップ

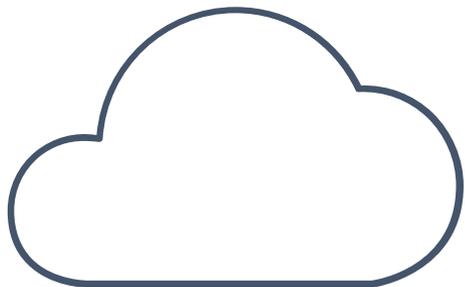
ステップ1. SLIの計測

ステップ2. 顧客視点のSLO設定

ステップ3. SLO範囲内で変更速度の最大化

ステップ1 SLIの計測

ステップ1. SLIの計測



SaaS

SLA
Service
Level
Agreement

Service Level のタイプ

SLI

Service Level Indicator

計測値

SLO

Service Level Objective

目標値

SLA

Service Level Agreement

契約値

SREがフォーカスするのはSLIとSLO

SLIのタイプ

SLIには種類があり、
代表的なものは可用性や
レイテンシー。
システムによって使い分
ける。

サービスの種類	SLIのタイプ
リクエストドリブン	可用性
リクエストドリブン	レイテンシー
リクエストドリブン	品質
パイプライン	新鮮さ
パイプライン	正確性
パイプライン	カバレッジ
ストレージ	耐久性



稼働時間の可用性

Uptime	Daily	Weekly	Monthly	Yearly
99%	14 分 24 秒	1 時間 40 分 48 秒	7 時間 12 分	3 日 15 時間 36 分
99.9%	1 分 26 秒	10 分 5 秒	43 分 12 秒	8 時間 45 分 36 秒
99.99%	9 秒	1 分	4 分 19 秒	52 分 34 秒
99.999%	1 秒以下	6 秒	26 秒	5 分 15 秒

99.9% - 人が調査、修正、解決するのに十分な時間がある

99.99% - 自動化を実装して、停電を検出し、リダイレクトし、セルフヒーリングを実行する必要がある

99.999% - 分散システムのうち、ごく一部の機能だけが使えなくなる程度

ステップ2 顧客視点の SLO設定

ステップ2. 顧客視点のSLO設定

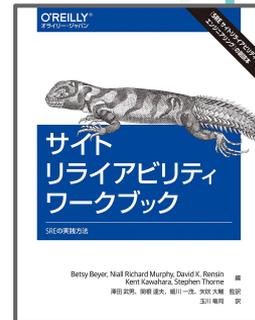
- SLOは、サービス顧客の目標レベルで信頼性を設定します。
- このしきい値を下回ると、ユーザーは不平を言い始めるか、サービスの利用を停止する可能性があります。
- 最終的に、ユーザーの幸せが重要です。(中略)

お客様の満足を維持するためにサービスの信頼性を維持しています。



SLOとエラーバジェット

それぞれの目的には個別のエラーバジェットがあり、
100%からそのSLOを引いたものとして定義されます。



$$100\% - \text{SLO} = \text{エラーバジェット}$$

$$\text{SLI} - \text{SLO} = \text{“残”エラーバジェット}$$

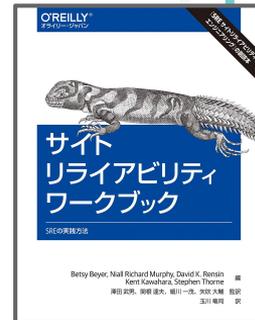
DevとSREは同じSLOにて合意

1. SLO設定とエラーバジェット導入をして、

DevチームとSREチームは、生産的な協同関係を享受

2. エラーバジェットポリシーの確立

例)エラーバジェットが1/3になったら変更の凍結を検討など



可用性のSLIとエラーバジェット(NRQL)

SLI 可用性

```
FROM SyntheticCheck
SELECT percentage(count(*), where result = 'SUCCESS')
facet monitorName LIMIT 10 since this month
```

可用性・エラーバジェット(成功 SLO 99%)

```
FROM SyntheticCheck
SELECT percentage(count(*), where result = 'SUCCESS') - 99
facet monitorName LIMIT 10 since this month
```

レイテンシーのSLIとエラーバジェット(NRQL)

SLI レイテンシー

```
FROM SyntheticCheck
SELECT percentage(count(*), WHERE duration <= 3000)
facet monitorName LIMIT 10 since this month
```

レイテンシー・エラーバジェット(レイテンシー3秒以内 SLO 99%)

```
FROM SyntheticCheck
SELECT percentage(count(*), where duration <= 3000) - 99
facet monitorName LIMIT 10 since this month
```

Dashboards

Service Level Dashboard - tshimizu ★ ⓘ

Demotron V2



🕒 Since 7 days ago ▾

☰ ▾ Search for any attribute or value.

SLI 可用性 (SLO 99%)

Since 1 week ago

100 %

Verify all APIs are working

100 %

Create alert violations data

100 %

Load homepage and assets

100 %

Verify homepage is working

69.66 %

Verify checkout flow is working

SLI レイテンシー (SLO 3秒以内99%)

Since 1 week ago

100 %

Verify homepage is working

99.97 %

Verify all APIs are working

99.17 %

Create alert violations data

97.02 %

Load homepage and assets

0 %

Verify checkout flow is working

可用性・エラーバジェット

Since 1 week ago

Verify all APIs are working 1 %

Create alert violations data 1 %

Load homepage and assets 1 %

Verify homepage is working 1 %

Verify checkout flow is working -29.34 %

レイテンシー・エラーバジェット

Since 1 week ago

Verify homepage is working 1 %

Verify all APIs are working 0.97 %

Create alert violations data 0.17 %

Load homepage and assets -1.98 %

Verify checkout flow is working -99 %

Service Level Indicator 推移

Since 1 week ago

ステップ3 SLO範囲で 変更速度の 最大化

ステップ4. SLO範囲内で変更速度の最大化

- モダン・モニタリング(オブザーバビリティ)
- インフラ以外の障害対応

監視 Monitoring

“監視とは、あるシステムやそのシステムの
コンポーネントの振る舞いや出力を
観察しチェックし続ける行為である。
Greg Poirier - Monitorama 2016”

“入門監視” - Mike Julian 著 松浦隼人 訳
オライリー・ジャパン

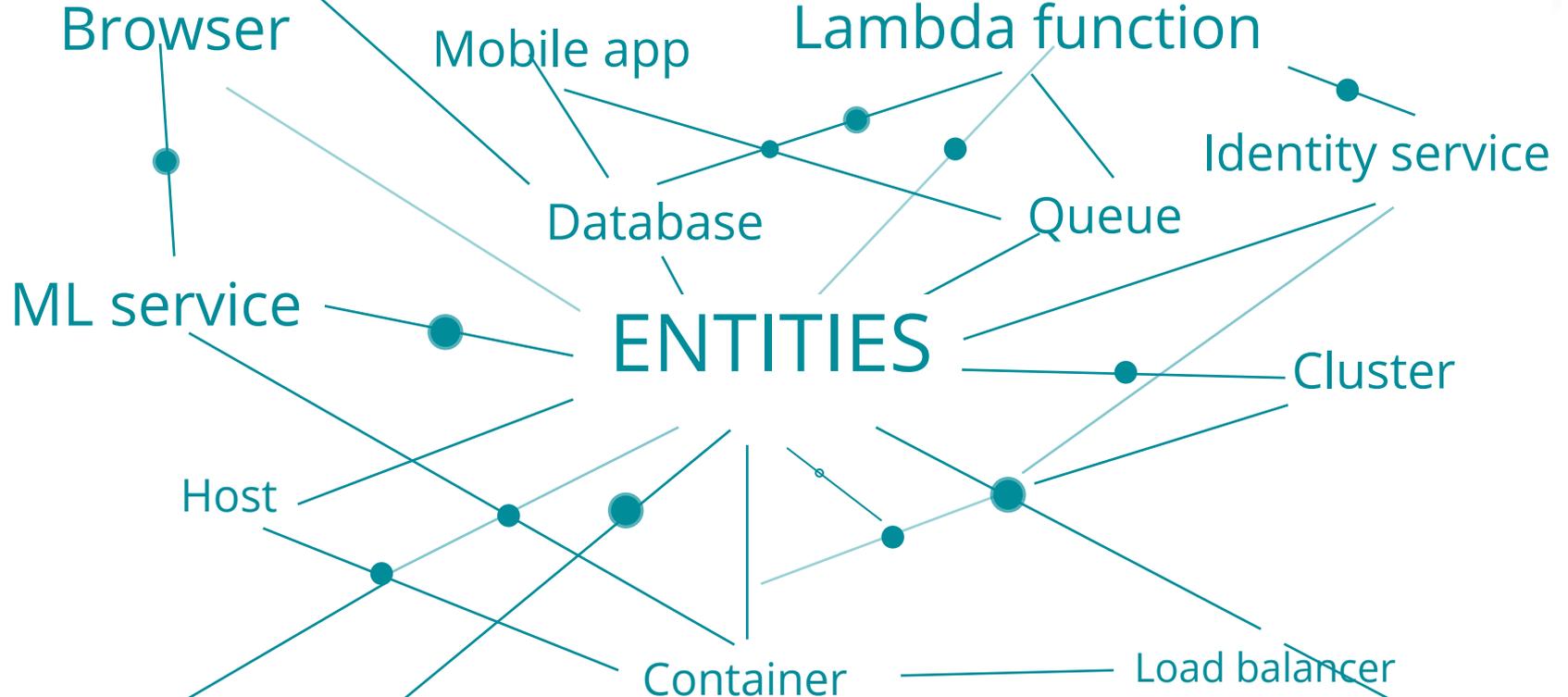
O'REILLY®
オライリー・ジャパン

入門 監視

モダンなモニタリングのための
デザインパターン



Mike Julian 著
松浦隼人 訳



(モダン)モニタリング

その監視ツールは、

ビジネス > サービス > アプリケーション

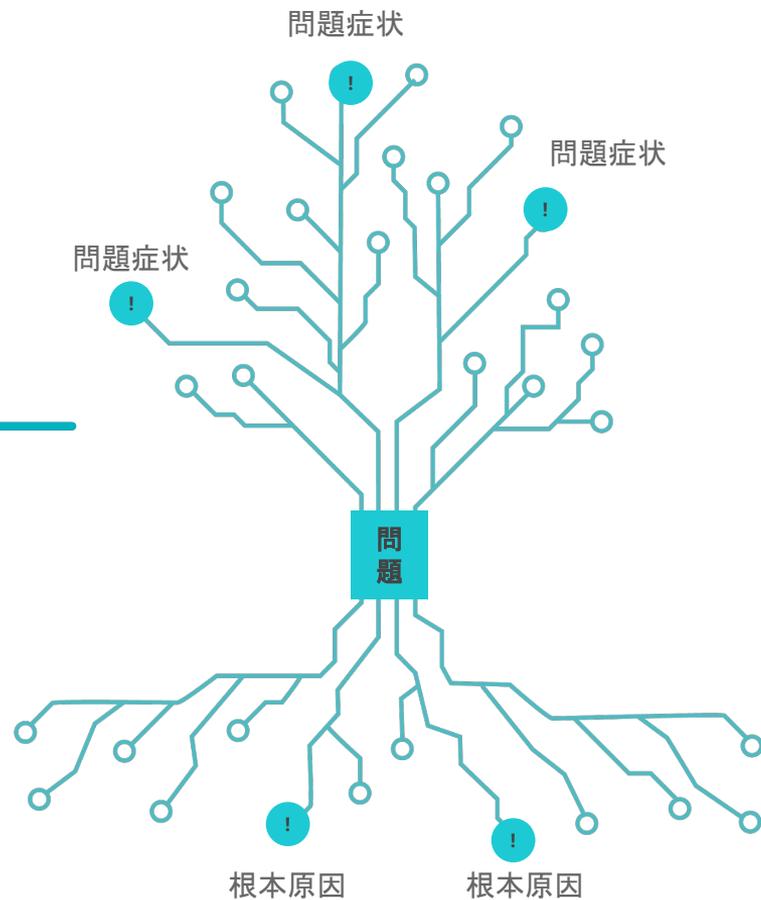
> ミドルウェア > インフラ > クラウド

すべて串刺しで見られますか？

オブザーバビリティ

Observability

システムのメトリクス・イベント・
ログ・トレースのデータを
リアルタイムに取得し続け、
常にシステム全容の
状態把握と改善ができる状態



オブザーバビリティ・プラットフォーム New Relic One

より良いソフトウェアの開発と実行



Front-End

New Relic BROWSER

ブラウザ体験モニタリング
ユーザー目線でページロードやエラーを把握

New Relic MOBILE

モバイル環境をモニタリング
iOSとAndroidアプリに対応

New Relic SYNTHETICS

外形モニタリング
世界複数拠点からの外形監視



Back-End

New Relic APM

アプリケーション性能モニタリング
8言語と70を超えるフレームワークに対応

New Relic INFRASTRUCTURE

あらゆるインフラ環境をモニタリング
パブリッククラウドとオンプレミス

New Relic LOGS

ログ収集と高速検索
MELTを高速収集し検索可能に



Analysis

New Relic ONE

ダッシュボード開発チャートビルダー
NRQLで分析を超高速度し、あらゆるテレメトリデータの可視化を実現



Perfect
Software

顧客体験の改善

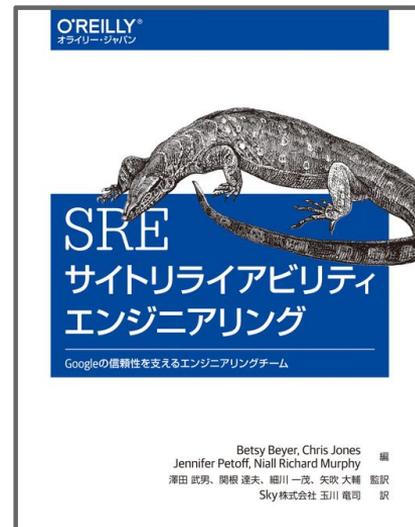
複雑かつ大規模システムの管理

NRDB

世界最速のデータ収集と検索

4 Golden Signalsの追跡

Indicator	Measure
可用性	Service Uptime %
遅延	応答時間
トラフィック/スループット	1分あたりの実行数
正確性/エラー	エラー率、クラッシュ率
リソース	CPU, Memory, Disk, I/O

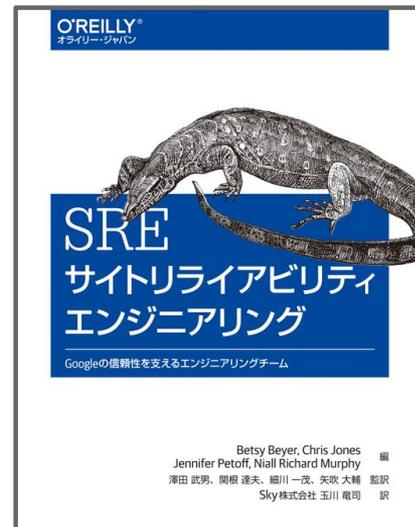


Source: [Google SRE Book - Monitoring Distributed Systems](#)

4 Golden Signalsの追跡

Indicator	Measure	New Relic Capability
可用性	Service Uptime %	S Synthetics
遅延	応答時間	B Browser M Mobile A APM
トラフィック/スループット	1分あたりの実行数	B Browser M Mobile A APM
正確性/エラー	エラー率、クラッシュ率	B Browser M Mobile A APM
リソース	CPU, Memory, Disk, I/O	I Infrastructure

A New Relic Full Stack Observability



インフラ以外のトラブル対応も

インフラエンジニアのトラブル対応スキルと
フルスタックオブザーバビリティを活用して、
アプリケーションの障害対応にも進出し
真のSREへ一歩踏み出す！

E2Eでのトラブル対応 - Service map

Summary

📄 Add to dashboard ↻ Reset map

MONITOR

Distributed tracing

Service map

Dependencies

Transactions

Databases

External services

JVMs

Threads

EVENTS

Errors

Violations

Deployments

Thread profiler

REPORTS

SLA

Scalability

Web transactions

Database

📄 WebPortal
Browser application

📄 WebPortal
Service

Add alerting connections:

- No alerting incoming connections
- All alerting outgoing are showing

Add or remove more connections

- 📄 Fulfillment Service Service
- 📄 Login Service Service
- 📄 Inventory Service Service
- 📄 Plan Service Service
- 📄 Promo Service Service
- 🌐 External 5 external services

📄 WebPortal Service

Preview | Connections | Violations

Details

Account name Demotron V2
Account ID 1606862

Web transaction breakdown
Since 30 minutes ago

● Java ● Response time ● Web external

Throughput
Since 30 minutes ago

● Today 45

Error rate

E2Eでのトラブル対応 - APM Transaction

Tsuyoshi Shimizu

Services

WebPortal ☆ ⓘ

CRITICAL Developer Toolkit Test Account

Show new view Since 30 minutes ago

Summary

Transaction type Web

Sort by Most time consuming

Instances All Instances

MONITOR

Distributed tracing

Service map

Dependencies

Transactions

Databases

External services

JVMs

Threads

EVENTS

Errors

Violations

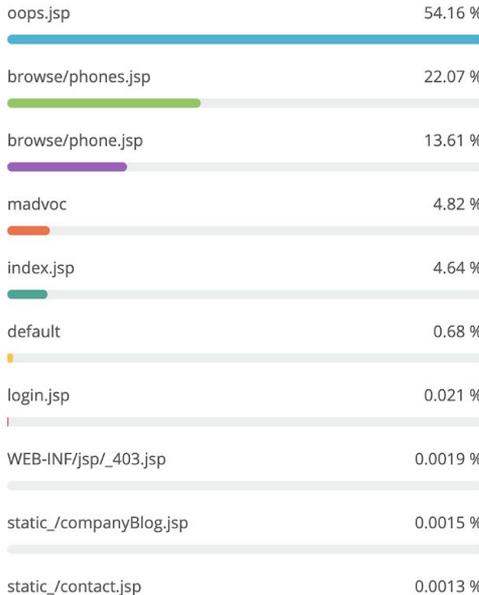
Deployments

Thread profiler

REPORTS

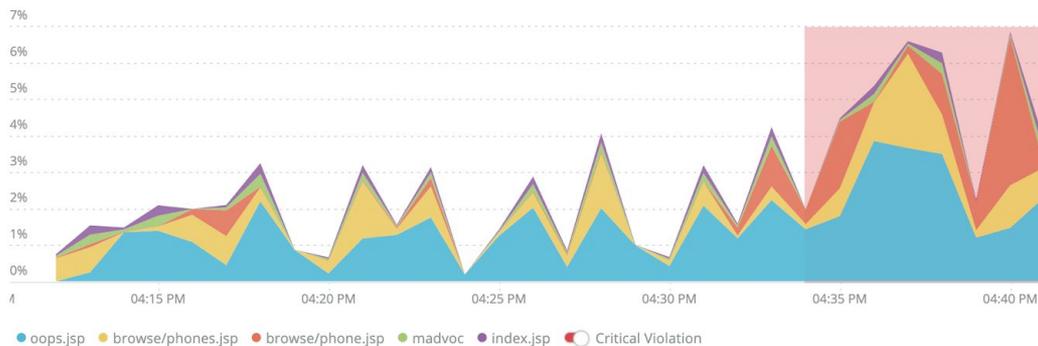
Top 20 transactions

by most time consuming



Top web transactions

by percent of wall clock time



Throughput

by requests per minute



変更障害の削減 - New Relic Deployment Marker

JVMs
All JVMs

Deployments

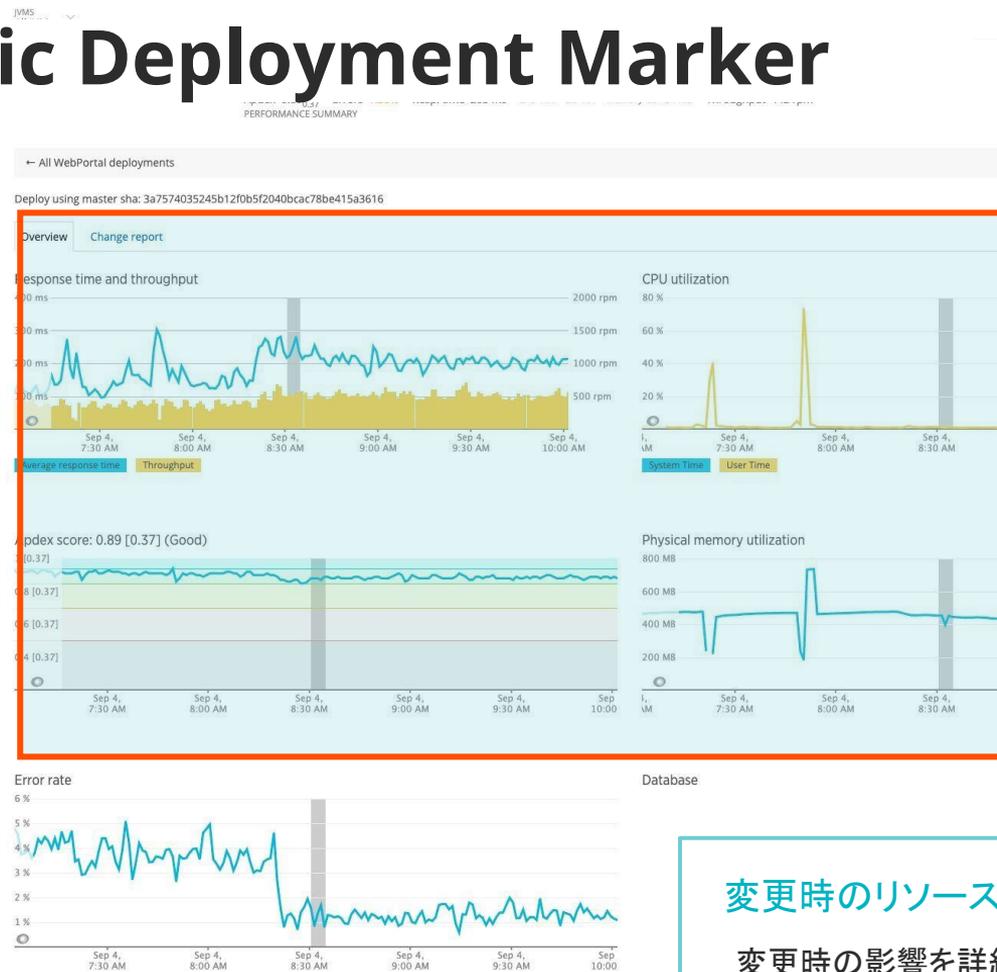
Show instructions

Our deployments feature reveals the impact of deployments on your application's performance. You'll quickly see whether the deployment errors.

< Previous 1 2 3 Next >

End user

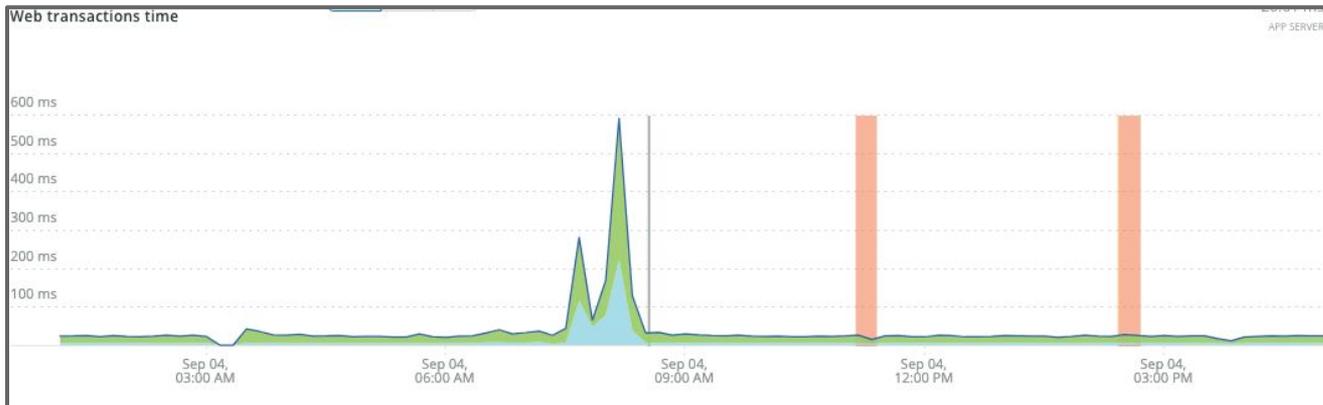
Time	Deployer	Apdex score	Resp. time	Throughput	Apdex
09/04, 8:32	jenkins	0.81 _{1,7}	1.17 sec	133 ppm	0.0
07/11, 7:32	jenkins	0.81 _{1,7}	1.17 sec	128 ppm	0.0
05/02, 7:34	jenkins	0.82 _{1,7}	1.17 sec	125 ppm	0.0
04/04, 7:33	jenkins	0.92 _{1,7}	1.02 sec	106 ppm	0.0
03/28, 5:13	jenkins	0.91 _{1,7}	1.04 sec	112 ppm	0.0
02/18, 14:24	jenkins	0.92 _{1,7}	1.02 sec	112 ppm	0.0
02/04, 14:35	jenkins	0.84 _{1,7}	1.23 sec	110 ppm	0.0
01/27, 16:09	jenkins	0.84 _{1,7}	1.24 sec	104 ppm	0.0
01/17, 13:48	jenkins	0.84 _{1,7}	1.23 sec	107 ppm	0.0
01/09, 16:06	jenkins	0.78 _{1,7}	1.32 sec	104 ppm	0.0
09/12/19, 11:51	jenkins	0.82 _{1,7}	1.28 sec	84 ppm	0.0
09/07/19, 15:48	jenkins	0.83 _{1,7}	1.26 sec	117 ppm	0.0



変更時のリソース

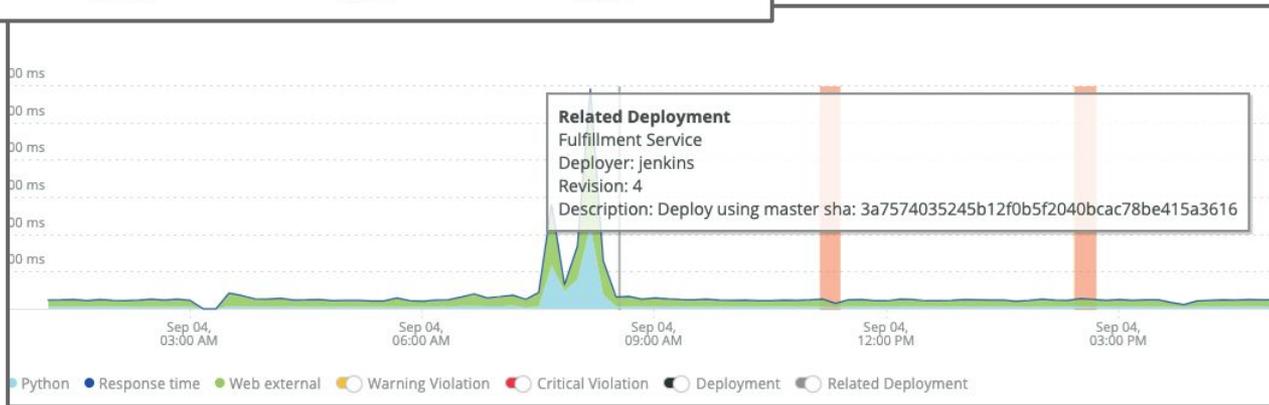
変更時の影響を詳細

変更障害の削減 - Deployment Marker



変更前後の応答時間確認

変更時の影響を詳細に可視化



New Relic One Demo

DEMO

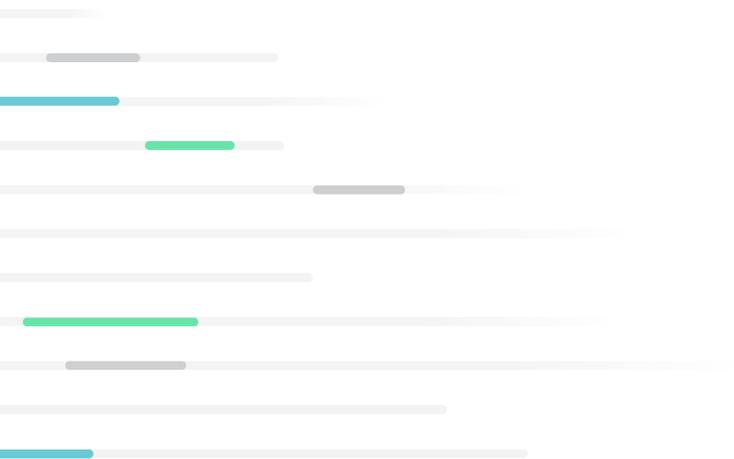
Service Level Dashboards

DEMO

E2Eトラブルシューティング

まとめ

1. インフラエンジニアが持つインフラ・サーバーサイド・モニタリング・トラブル対応のスペシャルスキルを活かして、
“インフラの安定からの脱却”
“SLOの達成をしながら変更速度最大化”に視点をシフトして、
Devチームと共にビジネスを加速させましょう！
2. そして、フルスタック・オブザーバビリティを活用し、
アプリケーションのスキル自体にも手を広げ真のSREへ！



Thank You

tshimizu@newrelic.com

@photographed

