



Developers Summit B-2

少数組織で開発する 大規模サービス

New Relic K.K.
Solutions Consultant

Furugaki Tomohiro



自己紹介



古垣 智裕 (ふるがき ともひろ)

New Relic 株式会社 技術統括コンサルティング部
ソリューションコンサルタント

前職でクラウドアーキテクトとセールスエンジニアを験経。商談から設計開発/運用まで一貫した経験を活かして、現在は主にメディア業界向けにオブザーバビリティのソリューションを提供している。

2019 年AWS APN Top Engineersを受賞。

好きなNew RelicサービスはNRQL。

少数組織で大規模サービスを開発するために

何が重要でしょうか？

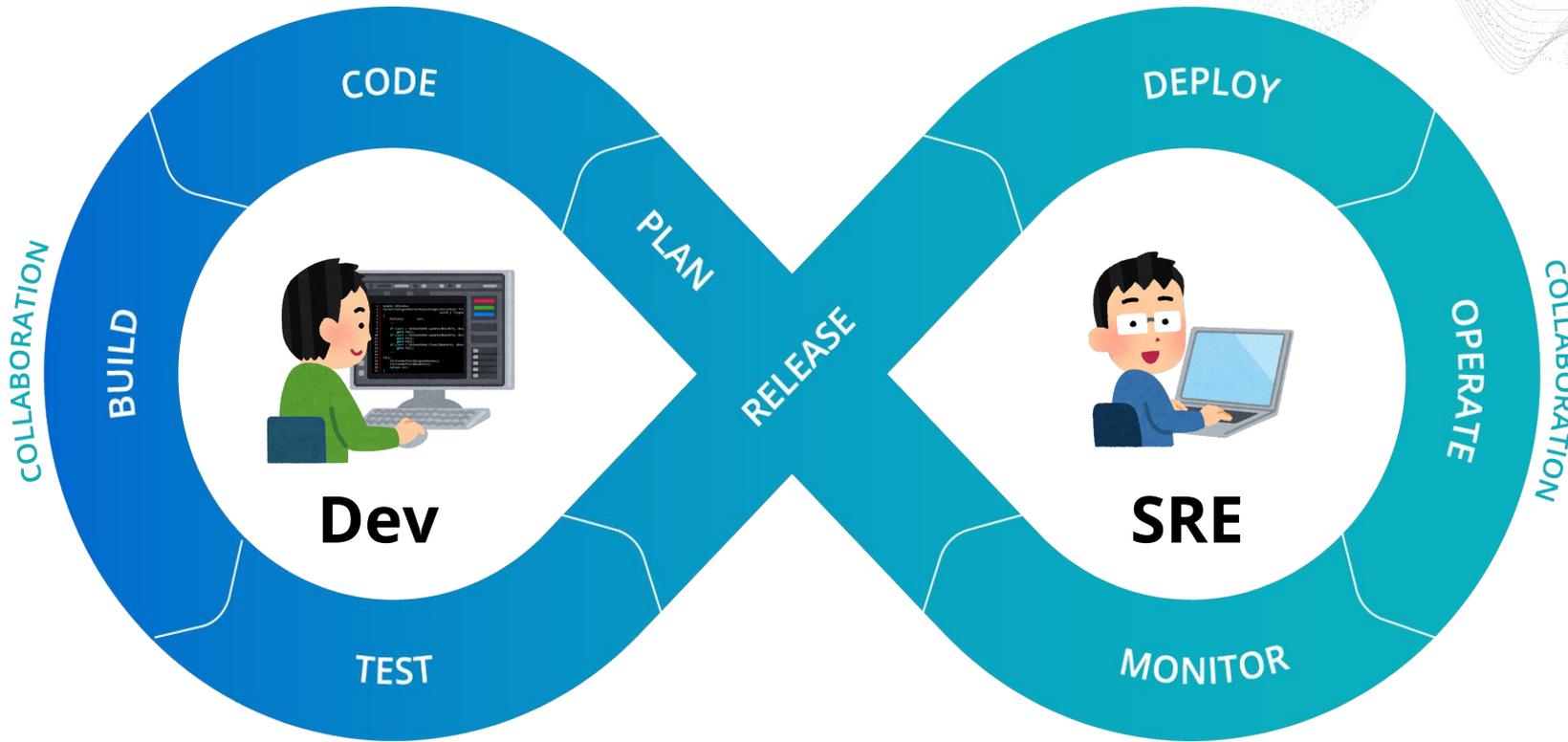


少数組織で大規模サービスを開発するために

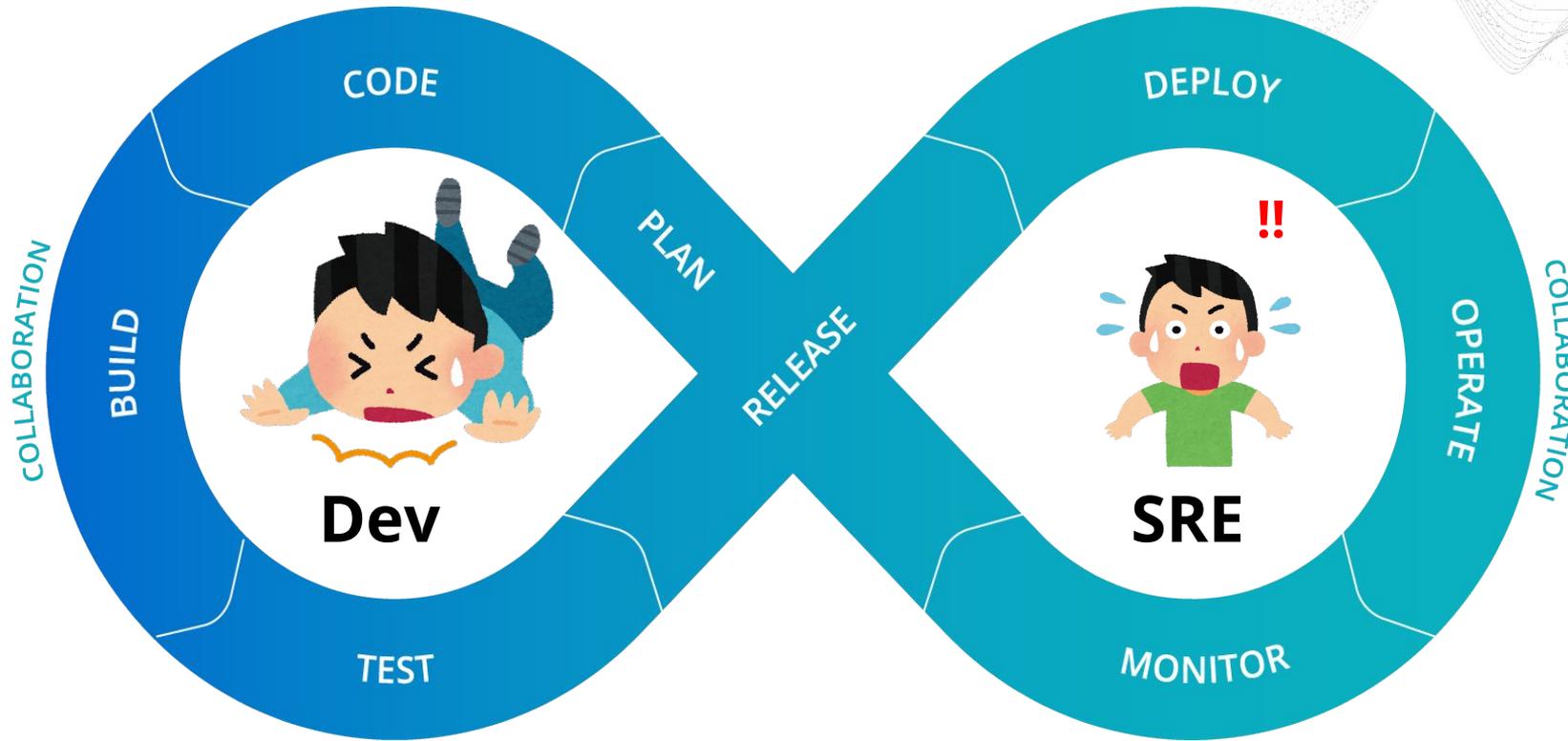
高速に回して開発！



DevとSREが自律的に協業して高速にサービスを開発



回せなかった時のインパクトは少数の方が大きい。



少数組織で大規模サービスを開発するために

“コケず”に高速に回して開発！



障害物を発見/対処し、全力で走れる(開発できる)



障害物を発見/対処し、全力で走れる(開発できる)

少数組織だからこそ
素早くリアルタイムに
障害物に対応できるためには？

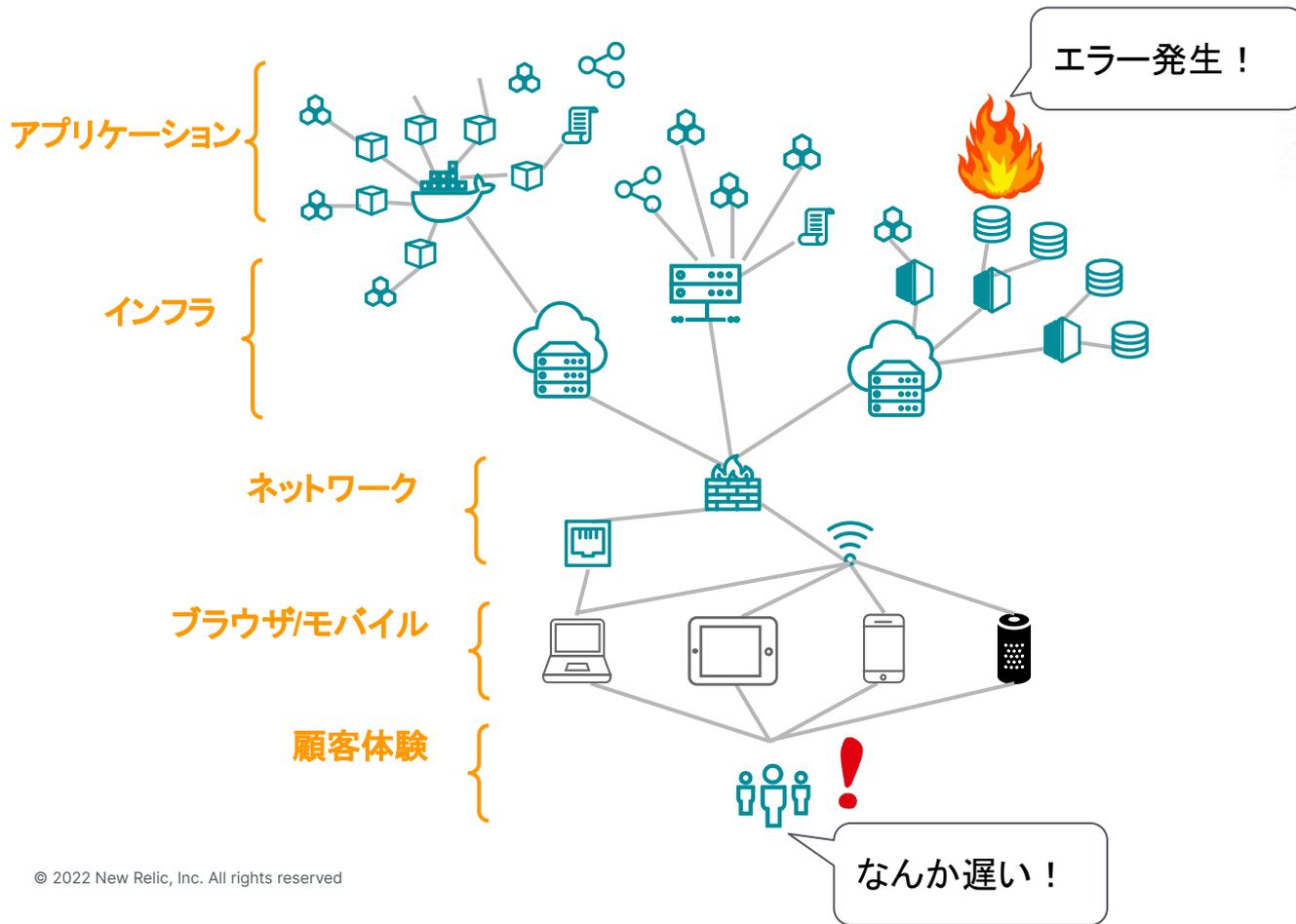


障害物を発見/対処し、全力で走れる(開発できる)

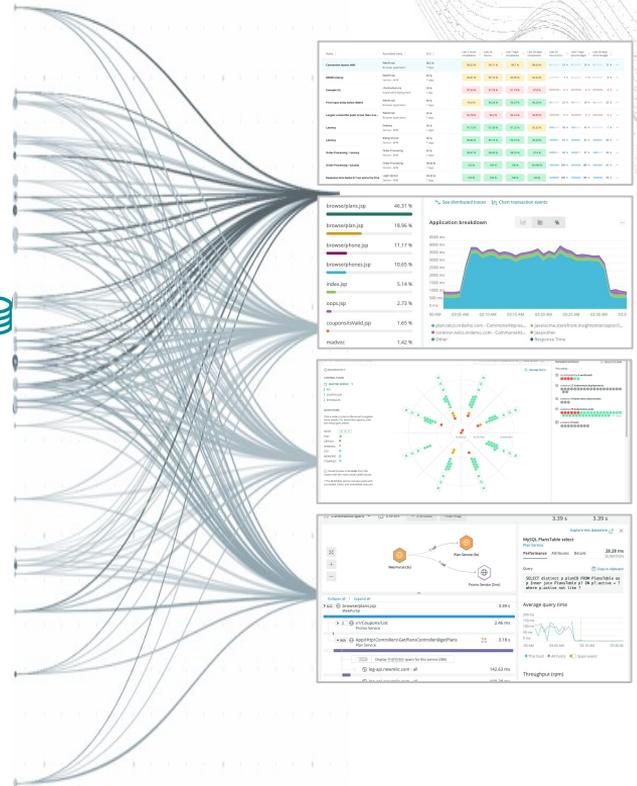
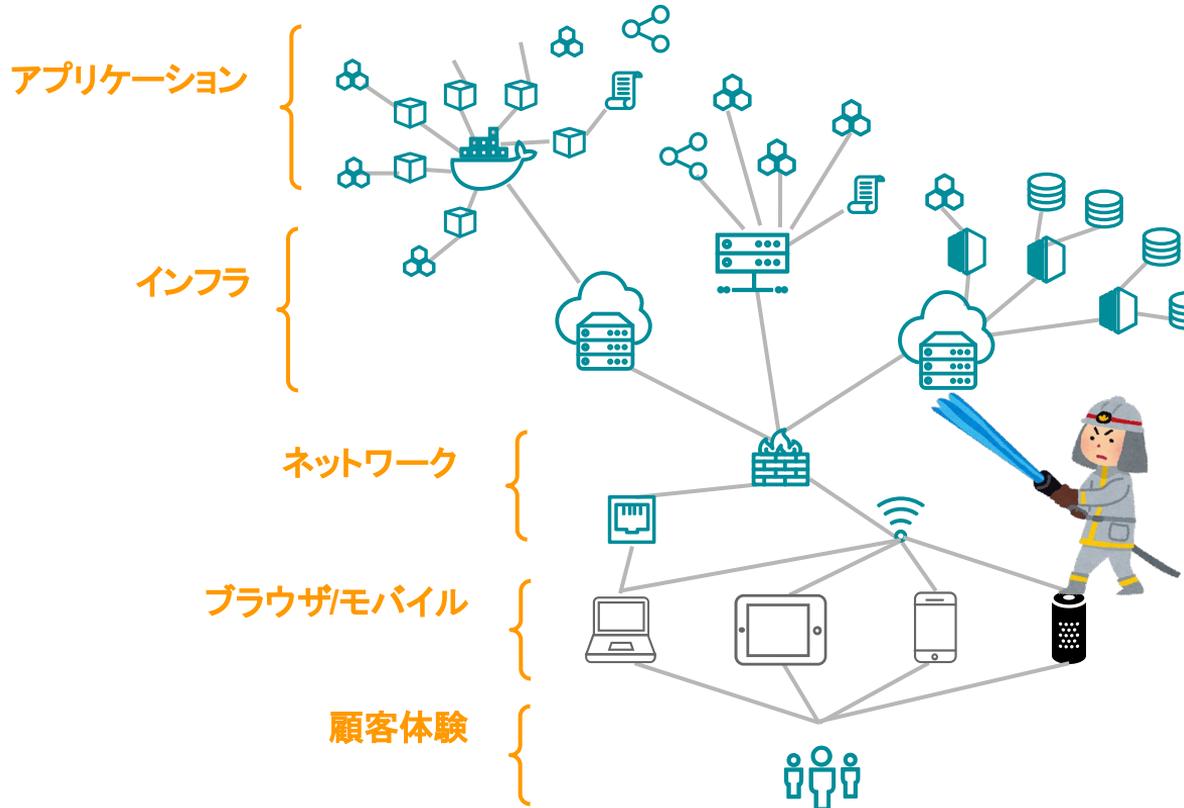
データを活用することで 障害物を特定する！



障害物はどこでも発生し見えにくいもの



プロダクトを計測しデータから障害物を特定

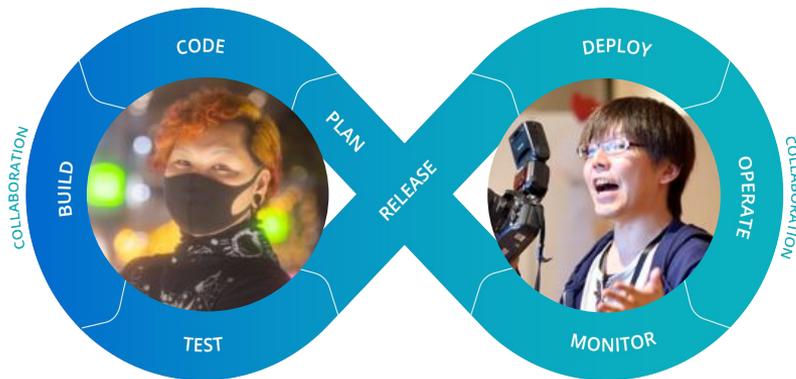


ここから聞ける話

TVerのSRE加我さん、Devの内海さんから以下のような実例をお話しいたします。

DevとSREの役割(ロール)別に大規模開発における、
プロダクトの計測とそのデータの活用に関する取り組みをお話しいたします。

- Role & Responsibility (DevとSREの職能横断)
- モニタリングの刷新とサービス改善の取り組み
- 大規模開発におけるエラートリアージ、ボトルネックの解消
- リリース後の改善活動



少数組織で開発する大規模サービス

株式会社 TVer / 加我 貴志 (SRE)

株式会社 TVer / 内海 恵介 (Backend)

2022/07/21

Developers Summit 2022 Summer



本日本話すること

- TVerリニューアルにおけるモニタリング改善
- 少人数で開発と改善を行うための
New Relic活用方法



- ・加我 貴志 (@TAKA_0411)
- ・株式会社TVer
サービス事業本部 SRE
- ・趣味はサウナとゲームとVTuber
- ・ホームサウナは溝口温泉 喜楽里

- TVerの紹介
- サービスリニューアルについて
- リリース準備編
- リリース後の改善
- まとめ

- **TVerの紹介**
- サービスリニューアルについて
- リリース準備編
- リリース後の改善
- まとめ

「TVer」は民放テレビ局のコンテンツを配信する民放公式テレビ配信サービス
毎週約**500番組**を**無料**でパソコン・スマートフォン・タブレット・テレビに配信

アプリダウンロード数

5,000万

(2022年7月予定)

民放公式テレビ配信サービス

TVer

月間動画再生数

2億5,300万

(2022年3月時点・株式会社ビデオリサーチ調べ)

MUB(月間ユニークブラウザ数)

1,850万

(2022年1月時点・株式会社ビデオリサーチ調べ)

番組放送終了から
約7日間見放題!

※番組終了から配信までは、
時間がかかることがあります

コネクテッドTVでの視聴割合

25%

(2022年3月時点・株式会社ビデオリサーチ調べ)

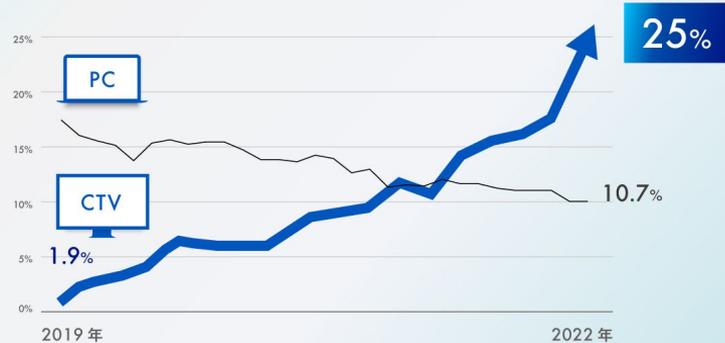
TVer 500番組以上が無料で見放題 民放公式テレビ配信サービス

TVer 月間の動画再生数

2億5千万回
を突破!

⇒コネクテッドTV視聴が初の25%超え

コネクテッドTVでの視聴増加



[TVer] 2022年3月の動画再生数が2億5千万回/月を突破!

<https://tver.co.jp/news/20220426-1.html>

TVerの 技術スタック



AWS

信頼性と拡張性に優れた
クラウドコンピューティング
サービス



Go

シンプルで信頼性が高く、効率的な
ソフトウェアの構築を容易にする
オープンソースプログラミング言語



GitLab

Gitの分散バージョン管理と
ソースコード管理(SCM)機能の
すべてを提供するだけでなく、
様々な独自の機能を有する



Google Cloud Platform

GCP / BigQuery

サーバーレスでスケーラビリティと
費用対効果が高い
データウェアハウス

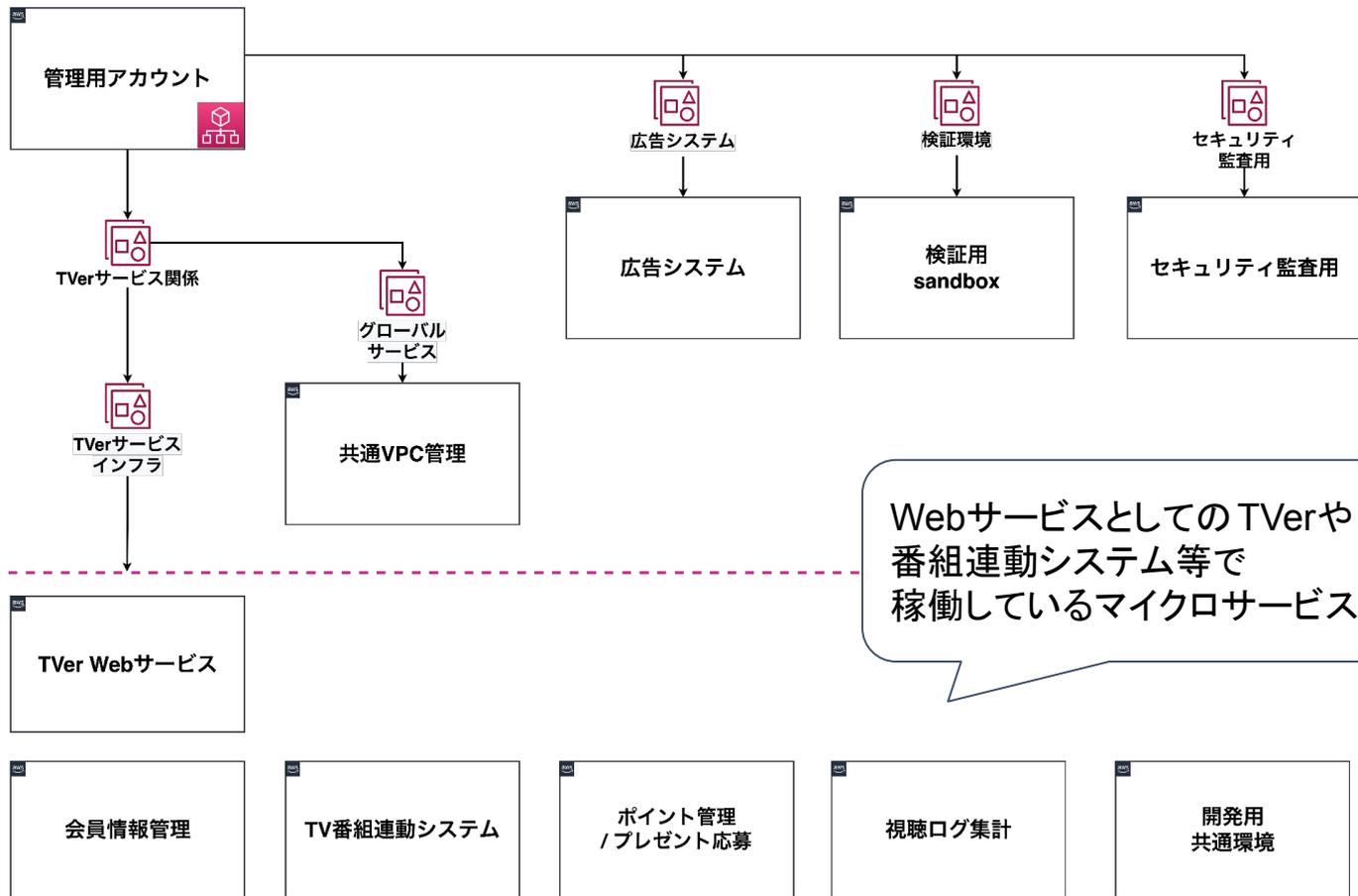
主に大規模な負荷を低コストで対応するために、パフォーマンスが高いGo言語で実装しています。

またCIでテストからデプロイまでのサイクル自動化を実施しています。

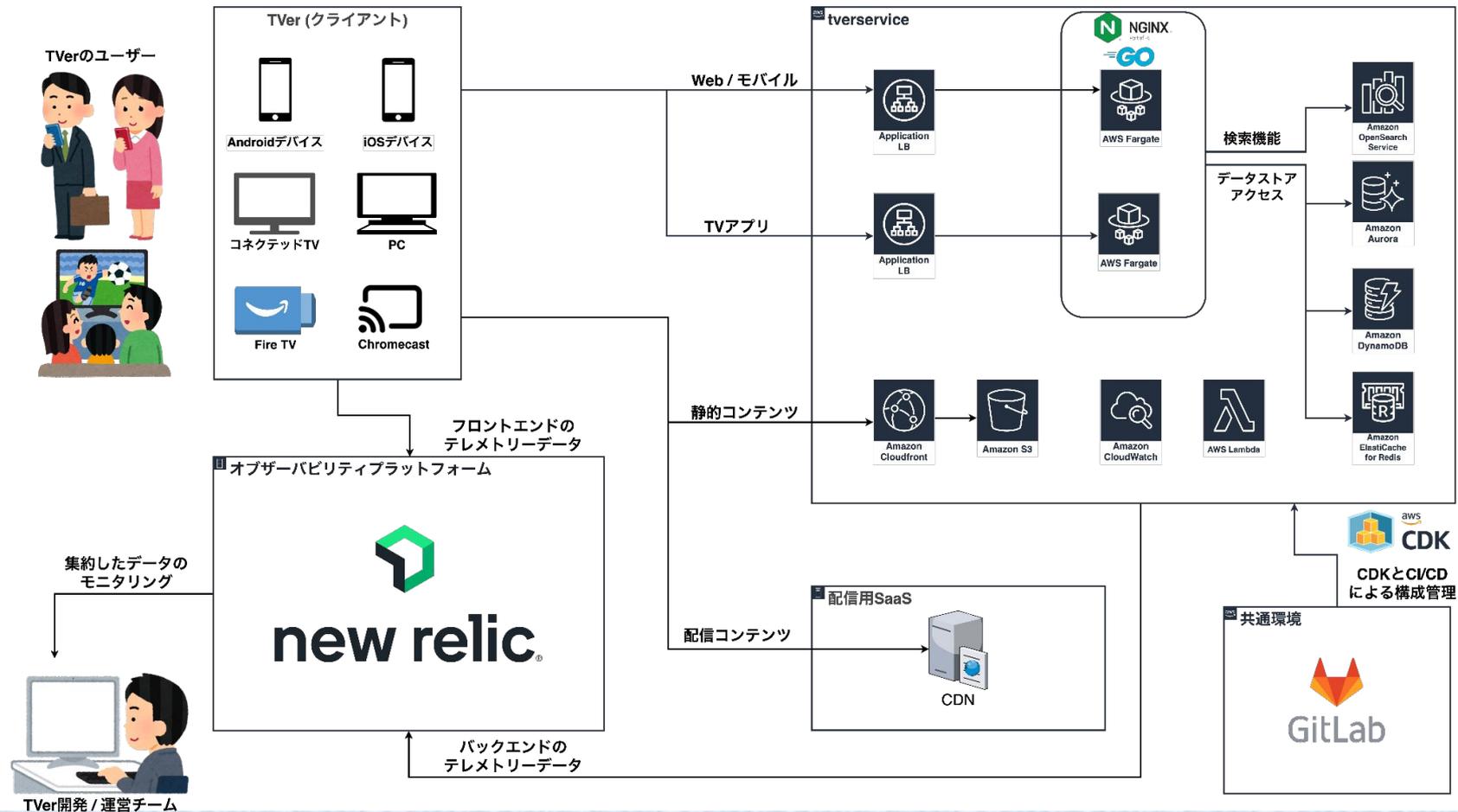
大規模データを放送局ごとに効率のかつ安全にマルチテナントで処理・提供するために GCP / BigQuery を利用しています。

※フロントエンド、TVアプリ、モバイルアプリケーションは
外部の協力会社のお世話になっております

TVerの インフラストラクチャ



TVer (Webサービス) のインフラストラクチャ



- TVerの紹介
- **サービスリニューアルについて**
- リリース準備編
- リリース後の改善
- まとめ

TV_{er} ID

ログイン機能がスタート！



マイページ

アカウントを作って
自分のページをカスタマイズ！

リストの共有

ログインすれば、PCやTVなど他の端末で
“お気に入り”や“あとでみる”のリストを
同期できる！

機能拡充

“追っかけ再生”など、一部ログインする
ことで利用可能な機能も！

※従来通りログインしなくても動画の視聴をお楽しみいただけます。

©TVer INC.

[TVer] テレビコンテンツの視聴体験や広告体験の向上を目指す「TVer ID」の提供を開始

<https://tver.co.jp/news/20220404-1.html>

システムの変更点

- インフラの内製化
- バックエンドの刷新
- モニタリングの刷新

システムの変更点

- インフラの内製化
- バックエンドの刷新
- **モニタリングの刷新** ←今回はこの話

**リニューアルを機に
New Relic Oneをフル活用**

モニタリングの刷新

- 責務の整理
- 導入する機能の取捨選択
- ダッシュボードとデータの活用

モニタリングの刷新

- **責務の整理**
- 導入する機能の取捨選択
- ダッシュボードとデータの活用

Before (～2021年12月)

<p>Backend</p> 	<ul style="list-style-type: none">New Relic 周りの対応APIの開発チューニング要件整理アプリやフロントエンドとの調整	<p>インフラ</p> 	<ul style="list-style-type: none">New Relic 周りの対応Fargate検証負荷テストCI/CD整備インフラ構築とCDK周りの整備	<p>情シス</p> 	<ul style="list-style-type: none">New Relic 周りの対応アカウント設計と整備コスト試算および契約各部署からの問い合わせ対応
--	--	--	---	---	---

「New Relicを **片手間** で見るのがつらい！」

After (2022年1月～)

みんな



New Relic周りのタスク
まるごと引き受けますね

SRE (私)



じゃあリードよろしく！
他のタスク進めるね

最近

Backend



ダッシュボードで
こんなメトリクスが見たい
んだけど～

この場合にはアラート
飛ばして欲しいよね

〇〇という値をキーにして
エラーを調査したい
と思ってるわけよ

やります

SRE

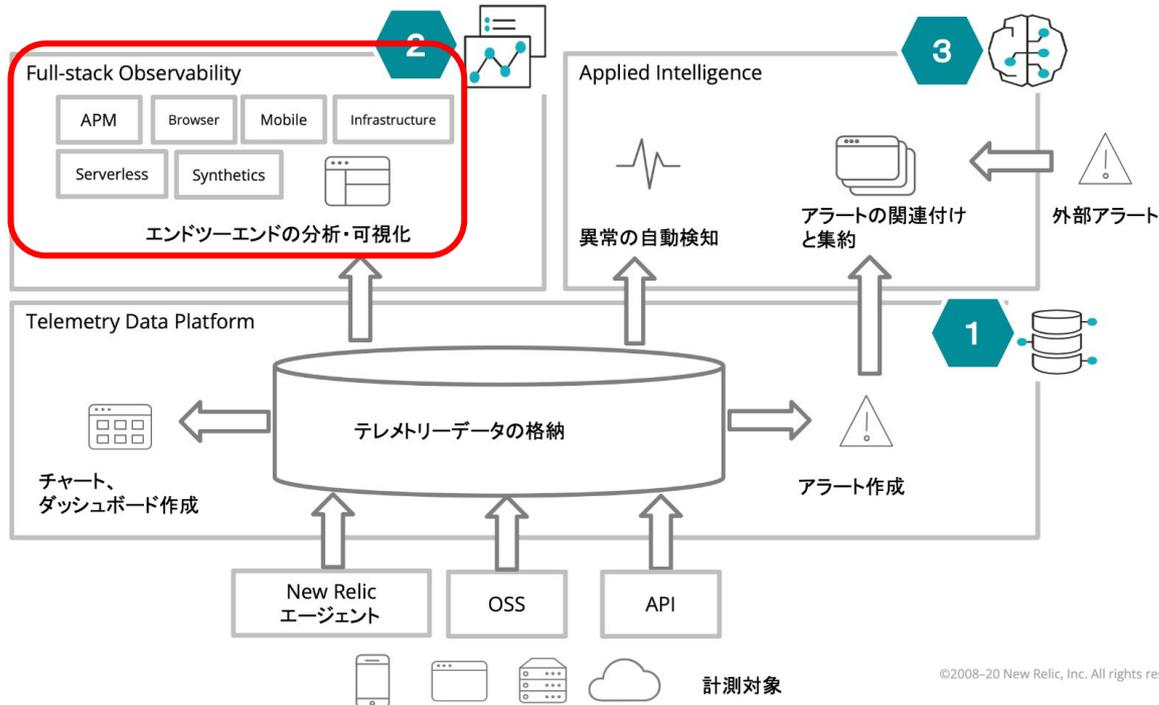


やりたい事を遠慮せずに相談・実現できる体制へ

モニタリングの刷新

- 責務の整理
- **導入する機能の取捨選択**
- ダッシュボードとデータの活用

New Relic One の全体像と関係性



自分の理想

- MELTを全て満たしたい
 - Metrics, Events, Logs, Traces
- **エラー調査のためにTraces(APM)を導入したい**



現実

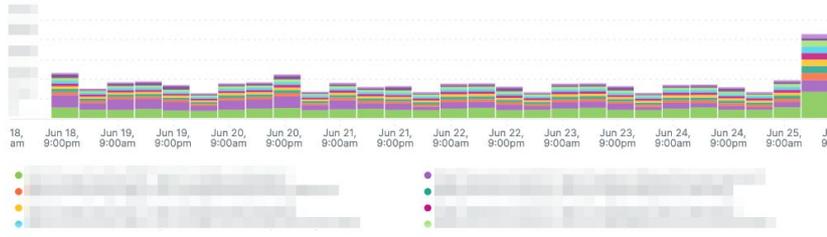
- New RelicのGo Agent(APM)の導入が若干面倒
- APMのオーバーヘッドが不安
- pprofで良くない？
- っていうか今必要なんだっけ？

リリースまであと2ヶ月・・・

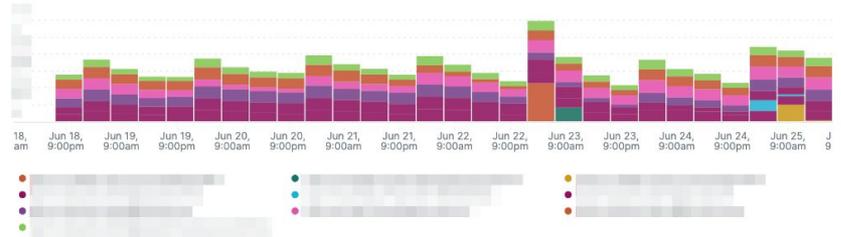


**とりあえず New Relic Go Agent を
必要最小限の箇所にだけ導入した**

[service-api] Transaction by API
Since 1 week ago



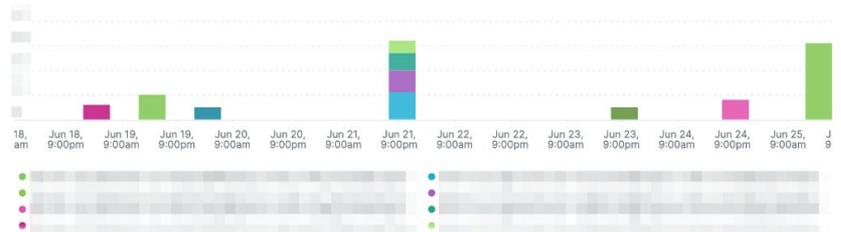
[service-api] Duration by API
Since 1 week ago



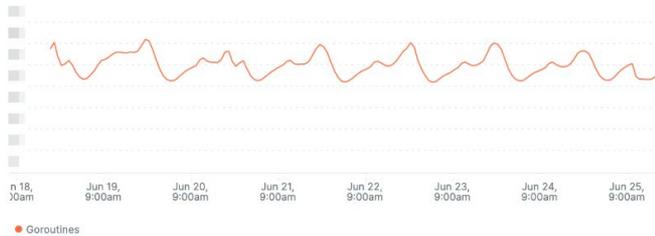
[service-api] Transaction Error by API
Since 1 week ago



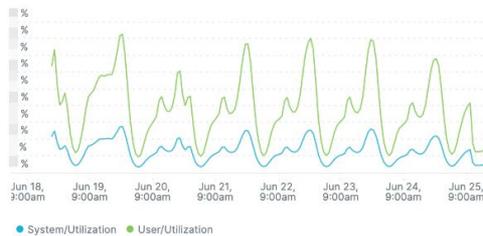
[service-api] Transaction Error by PostBody
Since 1 week ago



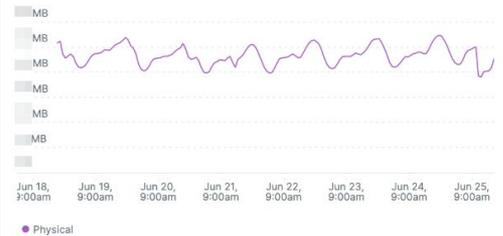
[service-api] Goroutines
Since 1 week ago



[service-api] CPU utilization
Since 1 week ago



[service-api] Memory (runtime.MemStats.Alloc)
Since 1 week ago



導入結果

- EchoのIntegrationを設定するだけで大幅に可視化できた
- エラー発生時のトリアージが可能になった
- **最小限の労力で最大の効果を得ることができた**



Dev (Backend)

Ops (SRE)

少数組織 × New Relic One

- 機能が豊富でオールインワン
- 導入のハードルが低い
- ドキュメント・サポート・コミュニティが充実
- とりあえず導入するだけで大幅に可視化できる

導入・運用のコストが最小限で済んだ

モニタリングの刷新

- 責務の分割
- 導入する機能の取捨選択
- **ダッシュボードとデータの活用**

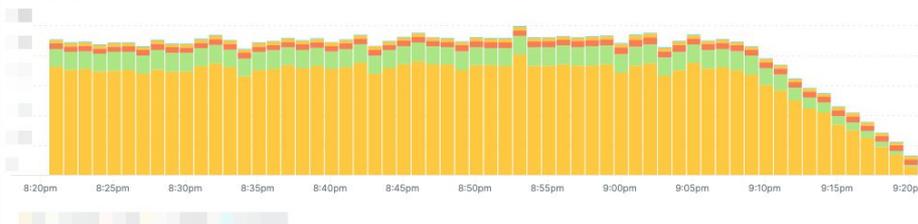
クライアントアプリ状況

- ・同時接続数
- ・アプリのクラッシュ
- ・ネットワークエラー

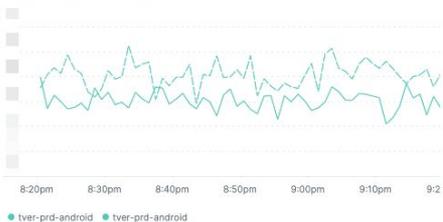
【前週比】同時接続数(ユニークブラウザ数)
Since 1 hour ago vs. 1 week ago



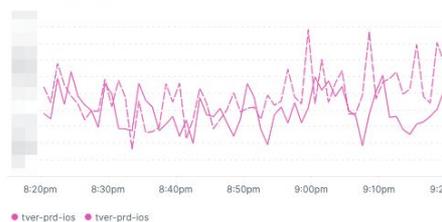
同時接続数(ユニークブラウザ数)
Since 1 hour ago



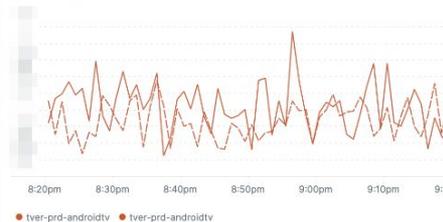
【前週比】モバイルアプリの平均クラッシュ率 (Android)
Since 1 hour ago vs. 1 week ago



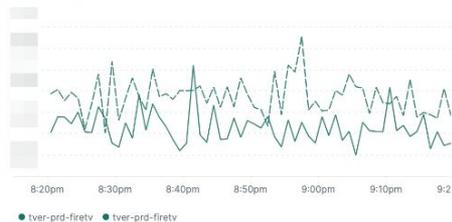
【前週比】モバイルアプリの平均クラッシュ率 (iOS)
Since 1 hour ago vs. 1 week ago



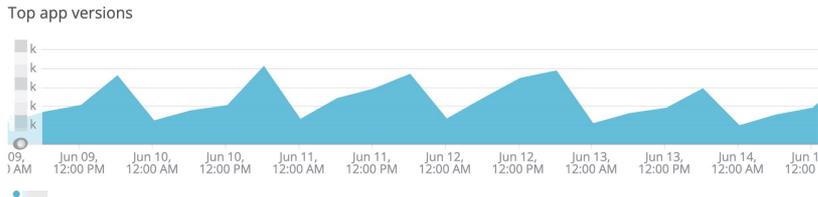
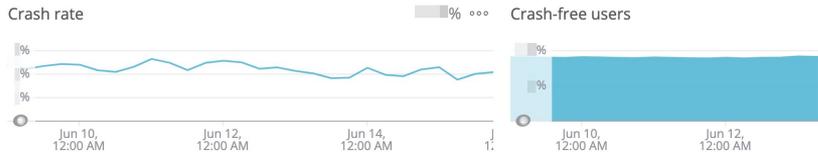
【前週比】TVアプリの平均クラッシュ率 (AndroidTV)
Since 1 hour ago vs. 1 week ago



【前週比】TVアプリの平均クラッシュ率 (FireTV)
Since 1 hour ago vs. 1 week ago



クライアントの同時接続数やクラッシュ率のメトリクス



Crash types (361) Crash profiles New

LOCATION	EXCEPTION	FIRST	LAST	APP VERSION
[Redacted]	java.lang.IllegalStateException	[Redacted]	[Redacted]	[Redacted]
[Redacted]	java.lang.NullPointerException	[Redacted]	[Redacted]	[Redacted]
[Redacted]	java.lang.NullPointerException	[Redacted]	[Redacted]	[Redacted]
[Redacted]	java.lang.IllegalStateException	[Redacted]	[Redacted]	[Redacted]
[Redacted]	java.lang.ArrayIndexOutOfBoundsException	[Redacted]	[Redacted]	[Redacted]



加我貴志 12:10 PM

- @channel お疲れ様です！NewRelicによる [Redacted] のTVerサービスエラー状況です。
- [Redacted] で頻発していたクラッシュがアプデにより改善されました🎉
 - ネットワークエラーの傾向が固まってきた印象があります。頻発してるやつから潰していきたいですね。
 - 既知ですが、 [Redacted] 状況です。

TVerサービス エラー状況 (2022/ [Redacted] - 2022/ [Redacted])

エラー状況 (ブラウザ)
ダッシュボード -> [Redacted]

tver-prd-browser
・先週に引き続き [Redacted] が大半を占めている状況。
・ [Redacted] を入れると発生。 [Redacted] 確認中。

tver-prd-browsertv
・先週に引き続き [Redacted] と [Redacted] が大半を占めている状況。
・前者は [Redacted] に関するエラー。後者は [Redacted] に関するエラー。

tver-prd-tversdk
・ [既知] [Redacted] 周りのエラーが若干増えている状況。 [Redacted]

tver-prd-chromecast
・引き続き下記エラーが大半を占めている状況。
[Redacted]

エラー状況 (モバイルアプリ)
ダッシュボード -> [Redacted]

tver-prd-android
・ [既知] [Redacted] が送られています
・ [Redacted] のクラッシュレートは %前後。
・直近1週間だと下記のクラッシュが多い。
[Redacted]
・直近1週間だと下記のネットワークエラーが多い。
[Redacted] (Secure connection Failed, DNS lookup failed)
[Redacted] (500 Internal Server Error)

tver-prd-androidtv
・ [既知] [Redacted] が送られています
・ [Redacted] のクラッシュレートは %前後。
・直近1週間だと下記のクラッシュが多い。
[Redacted]
・ [解決]

週1でモバイルアプリのクラッシュやネットワークエラーを調査し、広く社内に共有

ダッシュボードとデータの活用

- 必要なメトリクスでダッシュボードを整備
- フロントエンドからバックエンドまでカバー
- エラー傾向を調査して社内に共有
- **APIの改善にAPMを活用**



- ・内海 恵介 (@voidofglans)
- ・株式会社TVer サービス事業本部
リードエンジニア
- ・好きなもの:炭火で焼いたホルモン
- ・奈良在住の遠隔勤務中
- ・NFC埋め込んだりするのが好き

- TVerの紹介
- サービスリニューアルについて
- **リリース準備編**
- リリース後の改善
- まとめ

- フェーズによってNew Relicの使い分け
- 少人数でスピード感を落とさないために
- パフォーマンス・チューニング/負荷対策

- **フェーズによってNew Relicの使い分け**
- 少人数でスピード感を落とさないために
- パフォーマンス・チューニング/負荷対策

APM導入後のリリースまでのフェーズ

- 2ヶ月前
- 1ヶ月前
- 2週間前

- やるべきことはバグ潰し
- 主に利用した機能
 - Errors Inbox
 - Errors
 - NRQL
- 4xx/5xx系エラーの内容を分析
 - 4xx: クライアントが想定していないパラメータがないか
 - 5xx: 想定していない処理やタイムアウトがないか

- やるべきことは負荷試験によるボトルネック調査
- 主に利用した機能
 - Transactions
 - Go Runtime
 - CloudWatch Metrics
- 高負荷時のアプリケーション全体の挙動を分析
 - TransactionsでのSlowestや95percentileなどを指標に
 - メモリリークや、放置されたgoroutineの存在
 - GC Allocationによる断続的な性能劣化

- やるべきことは要求性能の達成
- 主に利用した機能
 - Transactions
 - CloudWatch Metrics
- 全体を通した負荷試験での安定パフォーマンス
 - 10ms以上かかるAPI
 - 各ミドルウェアへのクエリや接続の数
 - 書き込み系APIでのLock,Waitの分布

- フェーズによってNew Relicの使い分け
- **少人数でスピード感を落とさないために**
- パフォーマンス・チューニング/負荷対策

早く行きたければ、ひとりで行け。

遠くまで行きたければ、みんなで行け。

If you want to go fast, go alone. If you want to go far, go together.



おっけ！

じゃあちょっと1人で行ってくる！



リリース日までに
みんな現地集合で！

- それぞれがそれぞれの最速で目的地に
- 責務の明確な分解
- 明確な目的の共有

- コケないことが重要
- Backend/Infra/SREそれぞれが専任性を持って独立したメンバー
- コミュニケーションコストが最低限に抑えられた



- 地盤は整った
- 次はみんなで遠くへ行こう

- フェーズによってNew Relicの使い分け
- 少人数でスピード感を落とさないために
- **パフォーマンス・チューニング/負荷対策**

- よくある突発的なスパイク
 - 地上波連動や放送枠切れによる流入
 - 1min後はさっきまでの10倍以上のリクエスト数
 - 増える同時接続
 - 某アイドル事務所タレント様出演番組
 - リアルタイム配信番組終了時
 - ユーザーが一斉にザッピング
 - 同期のとれた数万～数十万のユーザーアクション

ALB Request Count by LoadBalancer

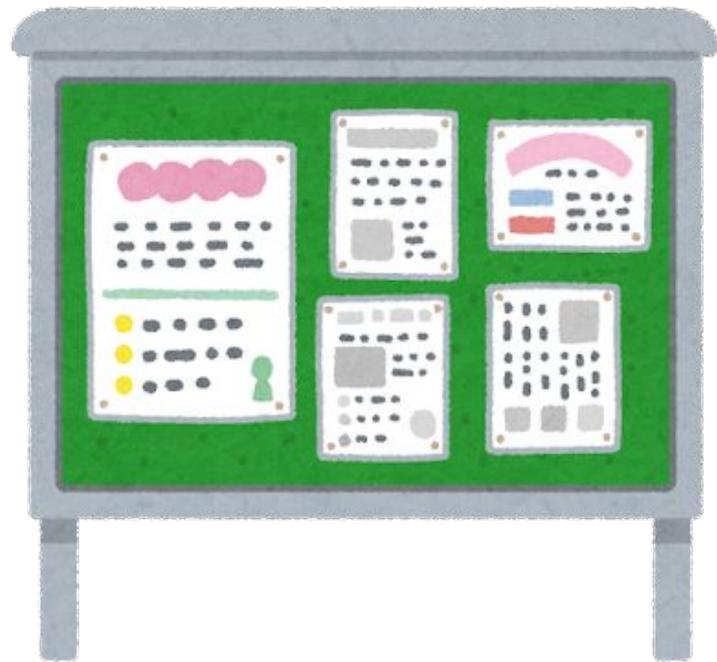


- リリース前にSREがしっかり専任してくれた
 - New Relicを見ておけば一般的な指標は全部見れる
 - 想定されるアクセスパターンにおけるMetricsの変化を事前に認識する
 - 負荷試験/対策に集中することができた

- TVerの紹介
- サービスリニューアルについて
- リリース準備編
- **リリース後の改善**
- まとめ

- 横断的に可視化されたMetricsの恩恵
- 定期的に実施するダッシュボード眺める会
- 責務の分離による対応速度の担保

- 安定稼働していることが遠目でわかる安心感



- デバイスごとの実装による差異の検知
 - デバイス種別をCustomAttributeに格納する
 - リクエスト傾向やエラー傾向の把握
 - iOSアプリだけ特定のページでのAPIリクエストが異常に多い
 - リスト表示内で要素ごとにAPIを呼び出していた
 - 30min足らずで原因特定→開発ベンダーへの共有
 - 3日後のリリースに包含し、無事解消 🎉

- 開発当初想定されていないユーザーシナリオ
 - 思ったものの3倍くらい検索機能が使われている！！
 - 思ったよりもxxの機能が使われてない！！

想定と運用は乖離するもの

- 過剰なリソースがすぐわかる
 - リリース初期は安心を金で買っていたので潤沢なリソースを用意していた
 - コストインパクトを軸に順次リソースの最適化を行えた
 - もちろんこれも可視化されているのでダメな場合の検知も迅速



- 昨日今日気になった指標をダッシュボードに追加
- Metricsの違和感に気づく嗅覚の訓練
- 障害後やトラブル発生時の知見の集積

- 正しく専任に任せる

- 中途半端に兼務していると、憶測や主観が入ってしまいがち
- 明確な責務の分離
 - SREは事実とログを渡すところに専念

- TVerの紹介
- サービスリニューアルについて
- リリース準備編
- リリース後の改善
- **まとめ**

- 1人で抱え込むのはよくない
- 早くやるためには責務を明確に分解する必要がある
- **でもやっぱりみんなですぐ早く遠くへ行きたい！！**

- 無事リリース & 安定運用が始まった 🎉
- NewRelicを開発環境でも活用
 - 開発時のバグトラッキング
 - 課題管理との連携
 - QAチームによるNewRelicでの事実共有
- 開発組織のリファクタリングを始めていきます
 - 今までの担務を数人チームで担当
 - 専任チームによるスループットを担保した開発組織



TVer

ホーム

• ミッション

価値観

メンバー

シゴト

会社情報

テレビの未来を放送局と一緒に作っていく。

2020年7月、民放キー局5社が資本を増強し、社名を株式会社TVerとし新たなスタートを切りました。テレビ番組を時間・場所・デバイスにとらわれず良質な体験としてお届けできるよう、テレビの未来と本気で向き合っていきます。

テレビを開放して、
もっとワクワクする未来を

TVer