



Ruby on Rails x Kubernetes におけるObservability

Isao Shimizu

NRUG : New Relic User Group

Nerd Life Talk (LT)

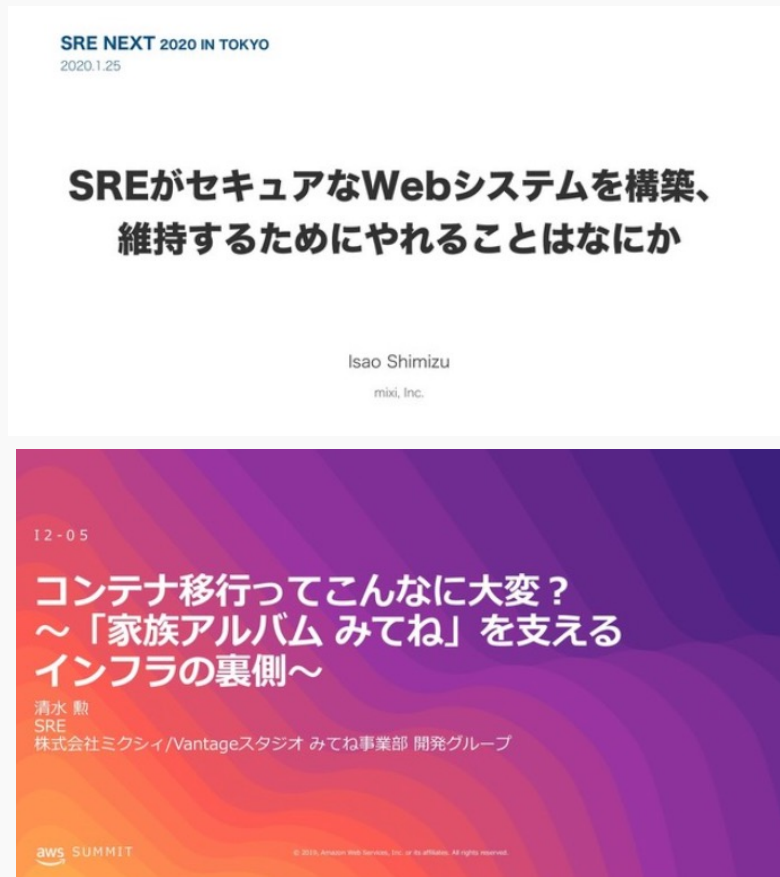
2021.9.15

自己紹介

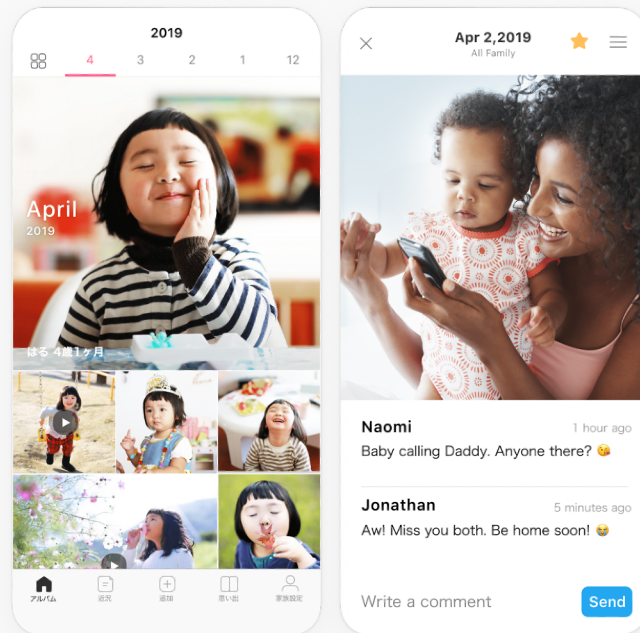
清水 勲 @isaoshimizu

- 2011年～ 株式会社ミクシィ
 - 2011年～2014年 SNS mixi 運用エンジニア
 - 2014年～2018年 モンスターストライク SRE
 - 2018年～現在 家族アルバム みてね SRE
- SRE NEXT 2020 IN Tokyo 登壇
- AWS Summit Tokyo 2014, 2019 登壇
- AWS Dev Day Tokyo 2018 LT大会 ベストスピーカー
- Software Design 特集記事寄稿 など

mixi, Inc.



家族アルバム みてね



全世界1,000万人以上の利用者

※ (株) ミクシィ調べ。iOS・Androidアプリ登録者数、ブラウザ版登録者数の合計

mixi, Inc.

多国語対応、世界150か国以上でサービス提供
日本語、英語、中国語（繁体字）、韓国語、フランス語
ドイツ語、スペイン語に対応



「家族アルバム みてね」のシステムが提供する機能の一例

● スマホで撮った写真・動画の保存と共有

- ユーザーの端末からAmazon S3へのアップロードをトリガーに、画像のサムネイル生成や動画のエンコードを実行

● フォトブック、フォトプリント、DVDの生成

- ユーザーが選んだ写真・動画をもとに組版データ、プリントデータ、DVD向け動画のエンコード、ISOイメージの生成、工場との連携

● 1秒動画の生成

- アップロードされた動画や写真を組み合わせて、期間ごとに成長を振り返ることができるようにしたダイジェスト動画を自動生成
- さまざまな機械学習技術を応用



本日お伝えしたいこと

KubernetesでRailsアプリケーションを 運用する際の課題と解決法

KubernetesでRailsアプリケーションを運用する際の課題

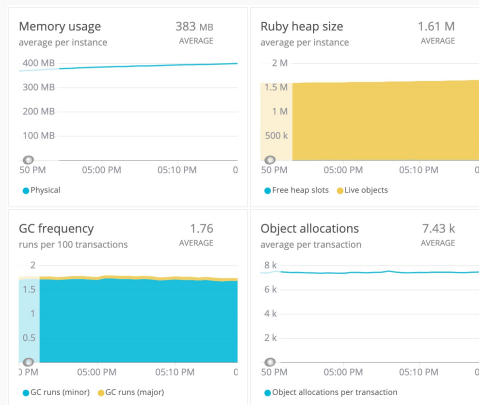
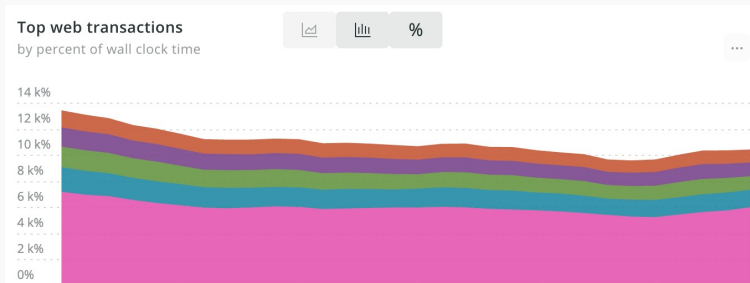
- アプリケーションエラーやレスポンス悪化にいち早く気づけるか
- オートスケールが想定通りに機能しているか
- Pumaが安定して動作しているか
- CPUやメモリを効率よく利用できているか
- OOM(Out of Memory)やCrashLoopBackoffが発生していないか
- 冪等性を持たない（リトライしにくい）バッチ処理が正常かどうか

これらをObservabilityプラットフォームであるNew Relicを使ってどう解決するか



アプリケーションエラーやレスポンス悪化にいち早く気づけるか

- New Relic APM (Application Performance Monitoring)で解決
- gemでの導入ということもあってコンテナとの相性が良い（余分なプロセス不要）
- 環境変数での設定が可能。ConfigMap/SecretsからAPMの設定ができる
- コンテナが大量でも一元的にメトリクスやエラーログなどを扱えるのが非常に便利



オートスケールが想定通りに機能しているか

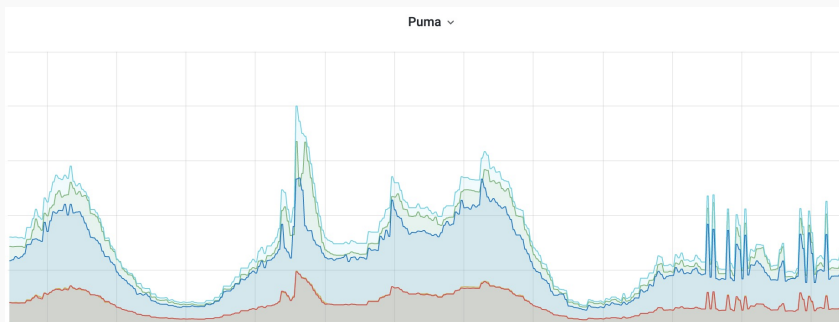
- New Relic Kubernetes integrationで解決
- Deployment単位でPod数がどのように増減しているか把握できる
- オートスケールのトリガーとなるメトリクスの把握
 - ただし、対象となるメトリクスがCPUやメモリだけではないケースがある
- Prometheusを利用してジョブキューのサイズなどを収集している場合
 - HPAとKEDAを連携させてPrometheusのメトリクスをベースにオートスケール
 - Prometheus側で保持するメトリクスはあくまでオートスケールのため（短期の保持で良い）
 - メトリクスを長期で保持させるために、PrometheusからNew Relicへremote writeする
 - これによって一通りのメトリクスがもれなくNew Relicのダッシュボードで確認できる



Pumaが安定して動作しているか

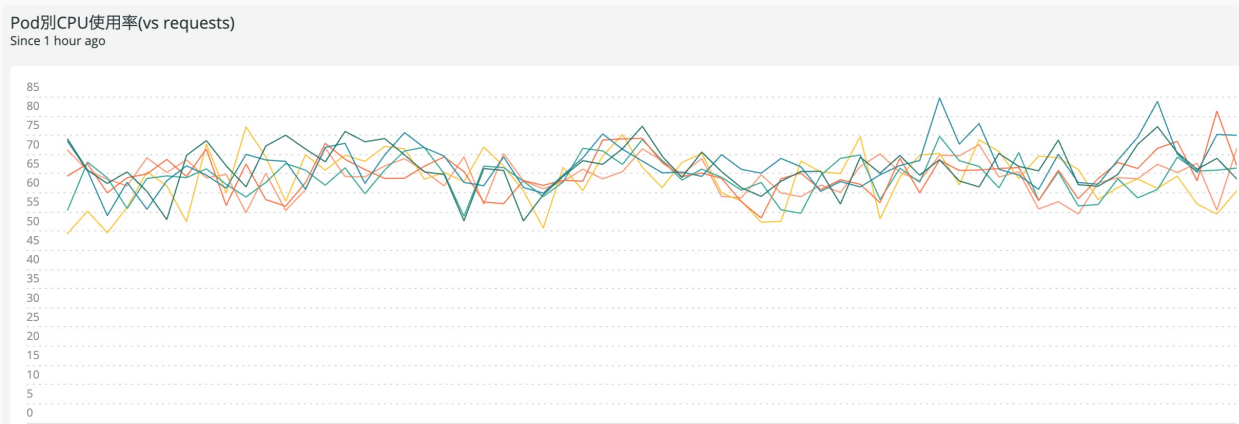
● Pumaの状態を把握する

- yabedaというgemを利用 <https://github.com/yabeda-rb/yabeda>
- さらにPumaのステータス取得するプラグインと、Prometheus Exporterのプラグインを利用
- Pumaのワーカー数、スレッド数、バックログなどのメトリクスを取得してダッシュボードに反映



CPUやメモリを効率よく利用できているか

- New Relic Kubernetes integrationで解決
- コンテナごとのCPU、メモリ利用率が要求(requests)に対してどの程度利用されているかを把握できる



OOM(Out of Memory)やCrashLoopBackoffが発生していないか

- New Relic Kubernetes integrationで解決
- Kubernetes内のさまざまなリソースに関するデータを収集
- OOMの発生 → リソース調整
- CrashLoopBackoffの発生 → エラーログなどを確認して対処

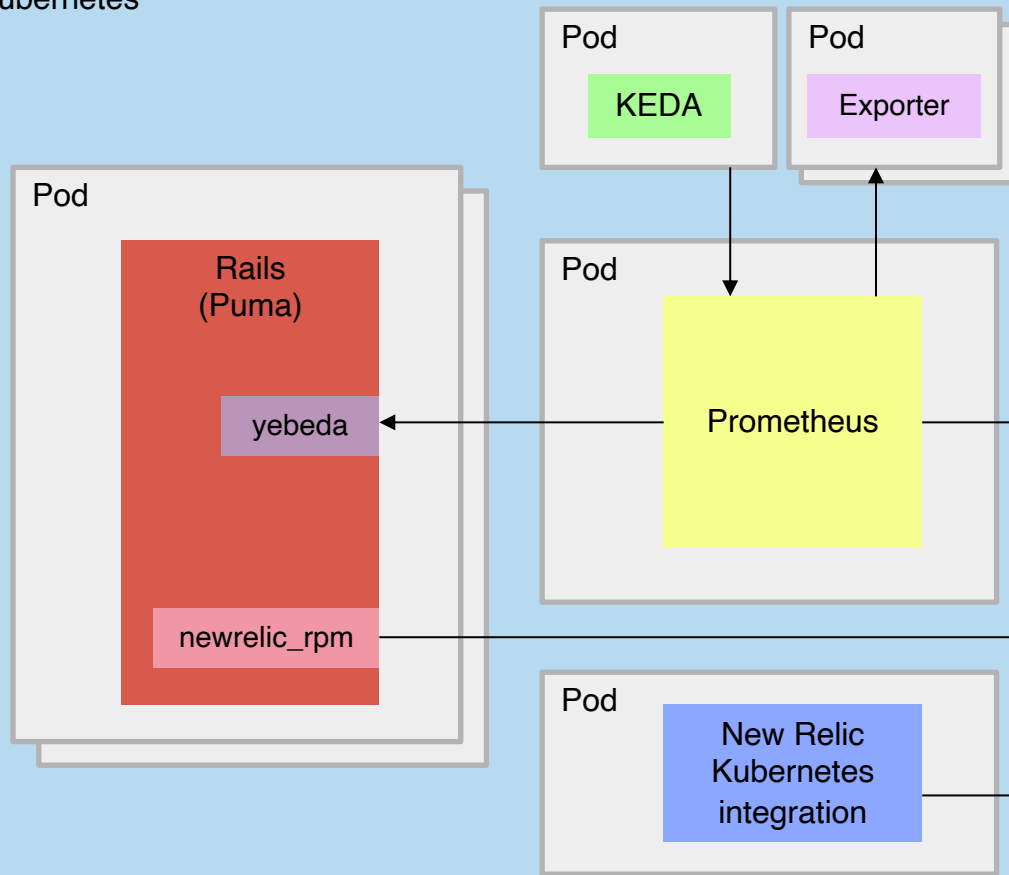
冪等性を持たない（リトライしにくい）バッチ処理が正常かどうか

- New Relic APM (Application Performance Monitoring)で解決
- バッチ処理において自動リトライできるもの・できないものがある
 - 冪等性がないもの、長時間実行されるものなどはリトライしにくい
 - whenever(crontab)からCronJobに移行したものの中にいくつかあった
- アプリケーション側でエラーハンドリングをしっかりと行うことでAPMで異常を検知
- 適切にアラートして人間が対処するものも



ざっくりと構成図

Kubernetes



Grafana

Internet

Internet

New Relic



まとめ

まとめ

- RailsをKubernetesで運用し始める際の課題を紹介しました
- Observabilityという点では、Railsだから苦労という点はほとんどありませんでした（APMがあったから安心できた）
- New Relic Kubernetes integrationは非常に便利
- PrometheusとNew Relicの連携（remote write）の機能に助けられた
 - GrafanaからPrometheusのプロトコル（OpenMetrics）を通じてグラフ描画可能
- 多くのメトリクスを取得しておくトラブルシューティングしやすい
- リソース効率を可視化することでコスト削減にもつなげられた



WE ARE HIRING!!
各ポジション積極採用中

team.mitene.us



ありがとうございました！



mixi, Inc.