「家族アルバム みてね」におけるスポットインスタンスを安定運用するための工夫

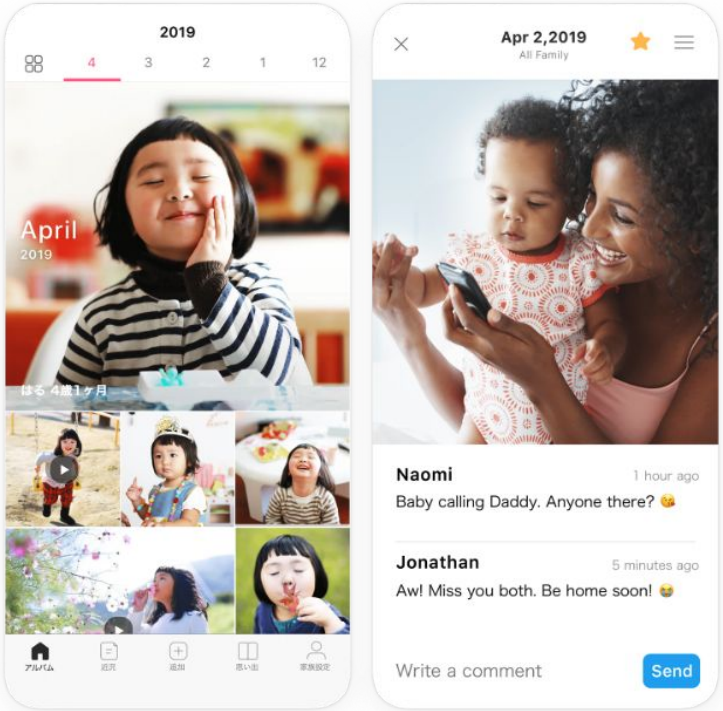
©mixi, Inc. All rights reserved

自己紹介

- 生島光
- 株式会社ミクシィ みてね事業部 SREチーム
- 経歴
  - 2013年 株式会社ミクシィ 入社
  - SNS mixi、モンスターストライクのデータ分析基盤の運用を担当
  - 2019年～「家族アルバム みてね」のSRE

©mixi, Inc. All rights reserved

家族アルバム みてね



全世界1,000万人以上の利用者

※ 2021年3月時点
※ (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

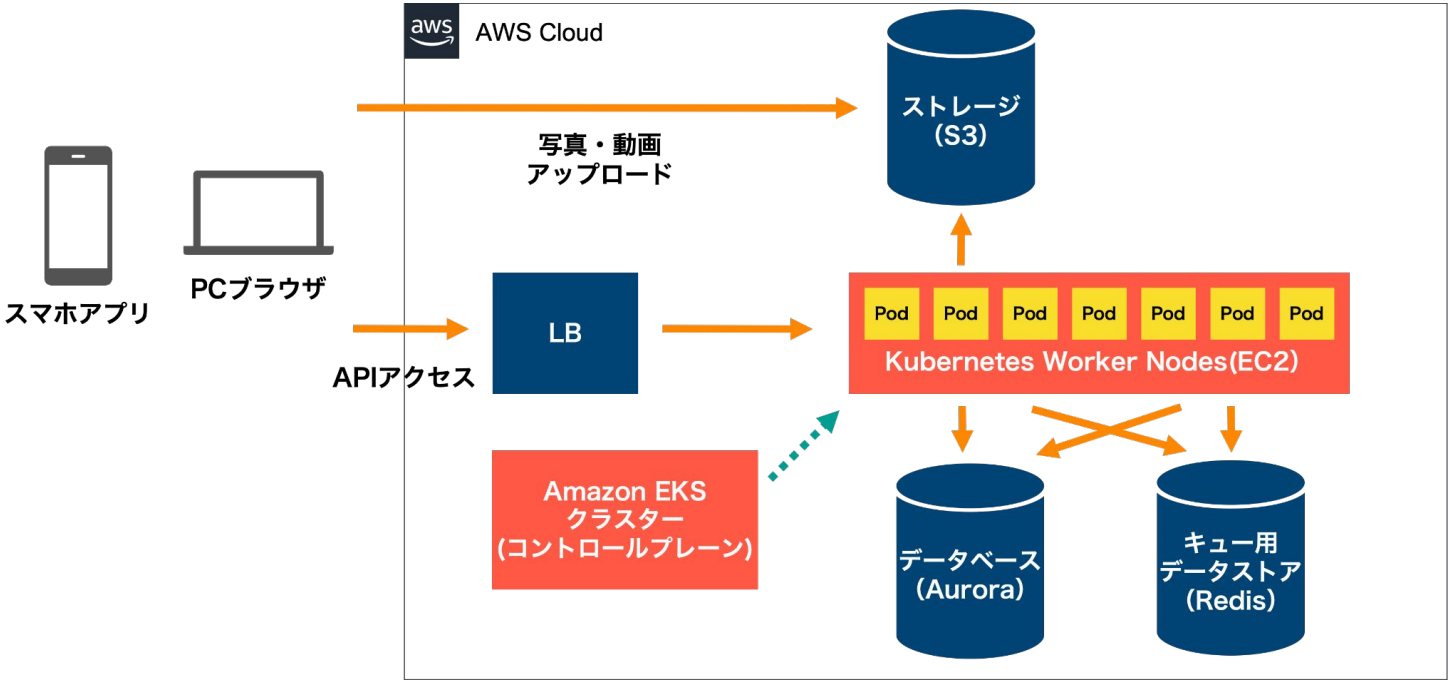
©mixi, Inc. All rights reserved

提供する機能の例

- スマホで撮った写真の保存・共有
  - Amazon S3へのアップロードをトリガーに、サムネイル画像の生成や動画のエンコードを実行
- フォトブックやDVDの作成
  - ユーザーが選んだ写真・動画をもとに、組版データや 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写真や動画を組み合わせて、期間ごとに成長を振り返れるようなダイジェスト動画を自動生成
  - 様々な機械学習技術を応用

©mixi, Inc. All rights reserved

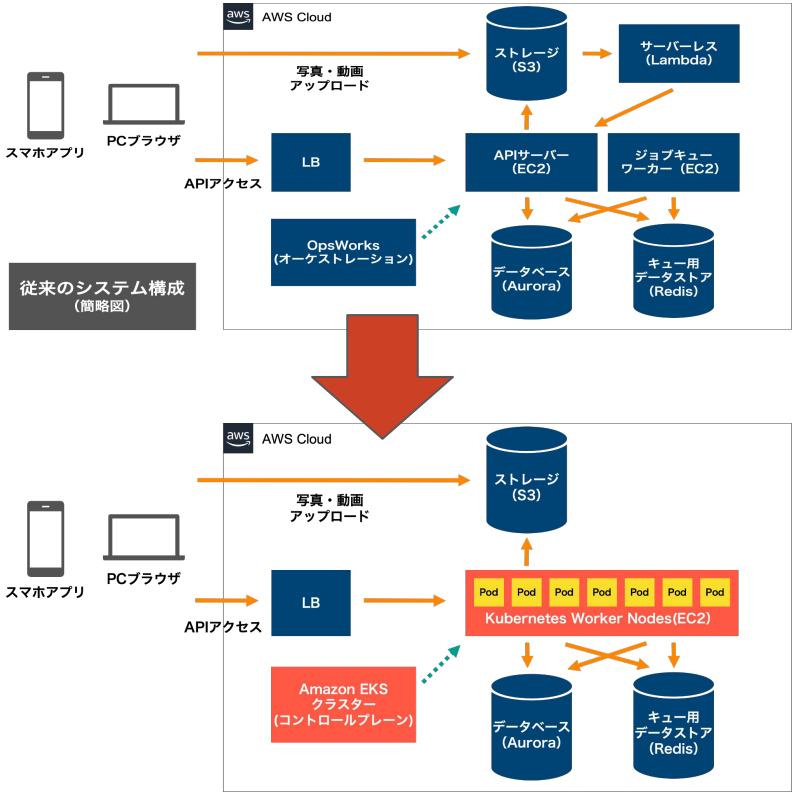
システム構成



©mixi, Inc. All rights reserved

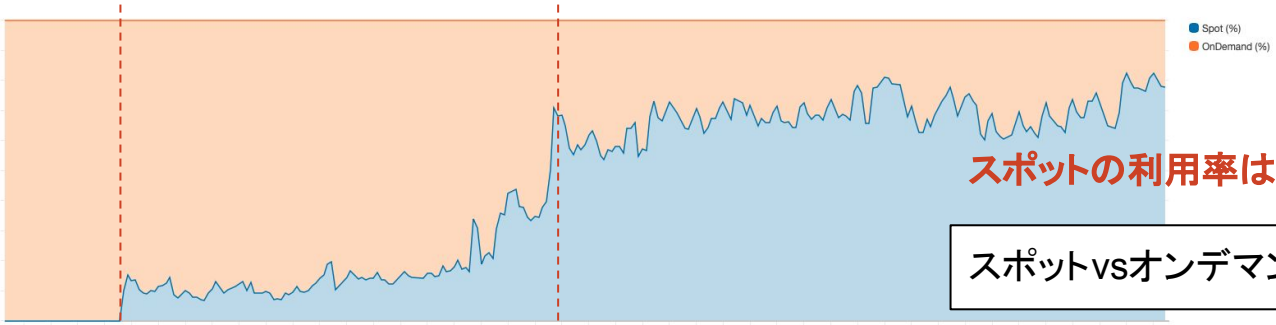
EKS移行とスポットインスタンスの導入

- 今年の春頃に、OpsWorksからEKSに移行しました。
- EKS移行によりさまざまな問題が解決：
  - オートスケーリングの高速化
  - Chef → Kubernetes
  - 障害復旧の自動化
  - SSHを前提としたオペレーションからの脱却
  - **スポットインスタンスの活用によるコスト削減**
- ref. AWS Summit 2019 (清水)



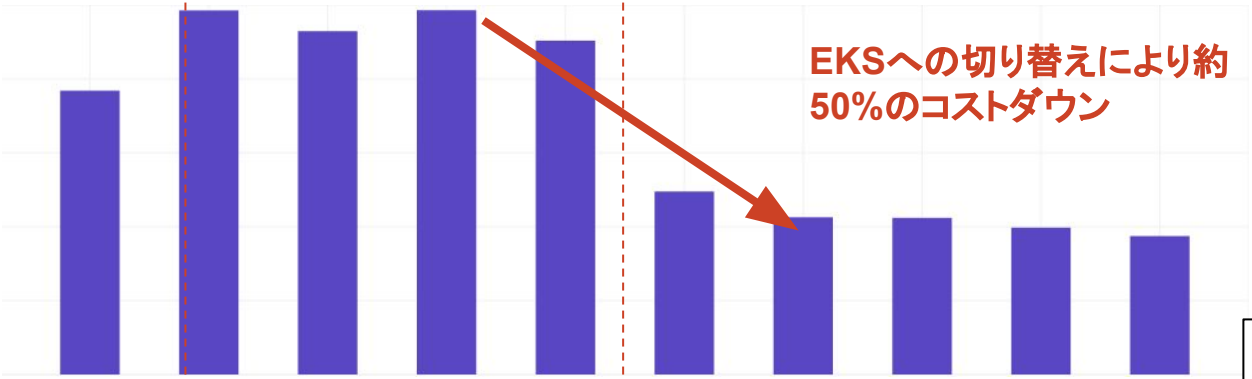
©mixi, Inc. All rights reserved

スポットインスタンスの効果



スポットの利用率は70%以上

スポットvsオンデマンドの比率の推移



EKSへの切り替えにより約
50%のコストダウン

月ごとのEC2費用の推移

EKS移行に着手
スポット部分的に導入

EKS移行がほぼ完了
スポット全面的に導入

©mixi, Inc. All rights reserved

スポットインスタンスを使う上での工夫

- スポットインスタンスの中断に備える
- スポットインスタンスの枯渇に備える
- オンデマンドインスタンスを効果的に使う

©mixi, Inc. All rights reserved

スポットインスタンスの中断に備える

©mixi, Inc. All rights reserved

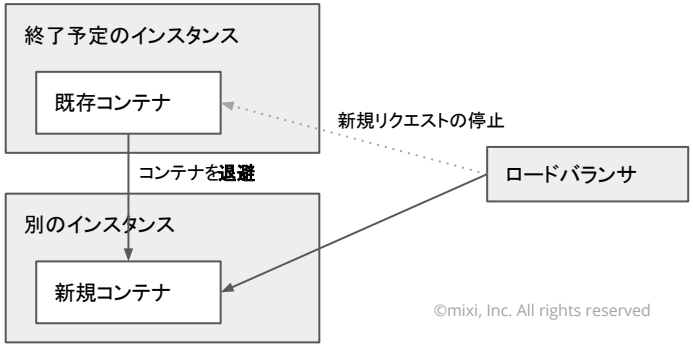
スポットインスタンスの中断に備える

- スポットインスタンスは、EC2の在庫状況などにより強制的にシャットダウン（中断）されることがある
- スポットインスタンスが中断された時でも、アプリケーション側に影響が出ないような工夫が必要

©mixi, Inc. All rights reserved

スポットインスタンスの中断に備える

- スポットインスタンスが終了する前に通知を受け取ることができる
  - スポットインスタンスが終了する可能性が高まった時（Amazon EC2 Spot Capacity Rebalancing)
  - スポットインスタンスが終了する2分前
- これらの通知を受け取った際に、コンテナを安全に終了させる
  - 中断される予定のインスタンスの上で動くコンテナは、新規のリクエストの受付を停止する。
  - 既存のリクエストの処理が終わったら、コンテナを終了する。
  - 別のインスタンスに新しいコンテナを作成する。



©mixi, Inc. All rights reserved

スポットインスタンスの中断に備える

みてねでは Managed Node Group を利用

- Managed Node Group を利用することで、スポットインスタンス中断時の処理を自分で管理しなくてよいため、スポットインスタンスを簡単に安定して使える。
  - Managed Node Group を利用しない場合は aws-node-termination-handler などをインストールする
  - その他、AMIの更新やノード数の変更などもいい感じにしてくれる。

©mixi, Inc. All rights reserved

スポットインスタンスの中断に備える

- アプリケーション側での対応も必要
- 処理の途中であっても、安全に中断できるように、実装を工夫する
  - ステートレス
  - 冪等性
  - 自動リトライ

©mixi, Inc. All rights reserved

スポットインスタンスの中断に備える

みてねにおける主要なワークロード3パターン

- Webサーバー・APIサーバー
  - 終了通知を受け取ると、新規リクエストの受付を停止し、処理中のリクエストが全て完了してから終了する
  - 中断耐性◎
- ジョブワーカー（Sidekiq や SQSワーカーなど）
  - 終了通知を受け取ると、処理中のジョブが全て完了してから終了する。
  - 中断耐性◎
- バッチ処理（CronJobなど）
  - 中断耐性があるものとないものが混在しているので、1つ1つ確認する必要があった。
  - 長時間実行されるタスクも要注意
  - 中断耐性△〜◎

©mixi, Inc. All rights reserved

スポットインスタンスの中断に備える

中断耐性◎ の例

- Webサーバー
  - 1つのリクエストは短時間で完了する
  - リトライも容易
- ジョブワーカー
  - ほとんど全てのジョブはリトライ可能な作りになっている

中断耐性△ の例

- 実行時間が長いバッチ処理
  - 1回の実行に12時間以上かかるレポート処理
- リトライできない処理
  - ある条件を満たすユーザー全員にプッシュを送信する処理
- 中断によりデータの不整合が生じる処理
  - DBとSFTPの状態を同期させる処理。工場との連携など。

©mixi, Inc. All rights reserved

スポットインスタンスの枯渇に備える

©mixi, Inc. All rights reserved

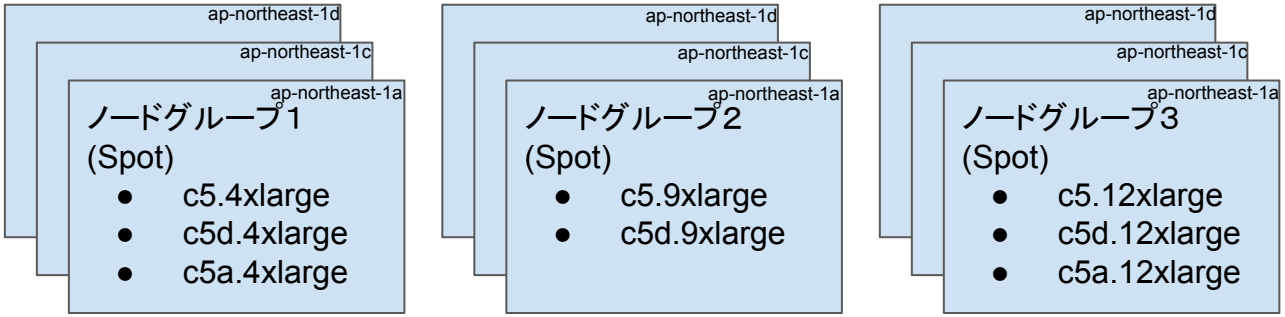
スポットインスタンスの枯渇に備える

- 複数のインスタンスタイプを混ぜることで、特定のインスタンスタイプの在庫が枯渇しても、クラスタ全体で必要なノードを確保できる。
- ただし、Cluster Autoscaler の制約で、1つの AutoScalingGroup に登録できるインスタンスタイプはvCPUとメモリのサイズが全て一致する必要がある。
- 同一のvCPU・メモリのインスタンスタイプからなる AutoScalingGroup を複数用意することで、インスタンスタイプの種類を増やす。
- さらに複数のAZに配置することで、安定性を高めている

©mixi, Inc. All rights reserved

スポットインスタンスの枯渇に備える

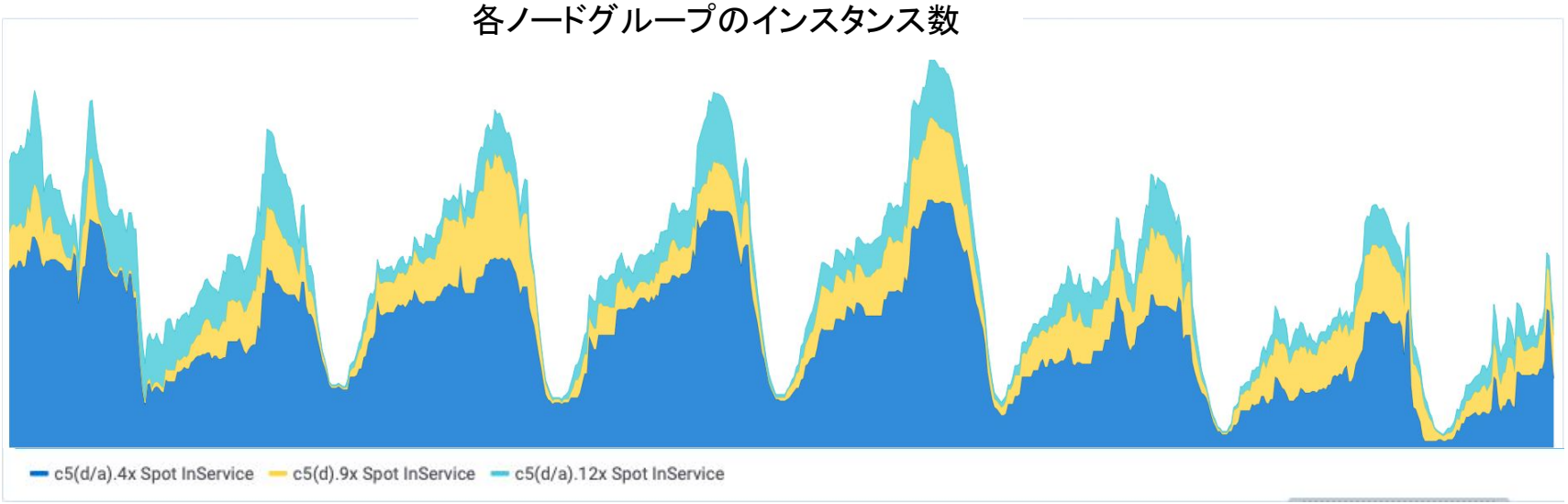
- みてねでは、8種類のインスタンスタイプを利用
- CPU・メモリのサイズごとに3つのAutoScalingGroupに分ける
- 各ノードグループを3AZに展開



©mixi, Inc. All rights reserved

スポットインスタンスの枯渇に備える

運用を開始して6ヶ月ほど経つが、スポットインスタンスの枯渇によるサービス障害はまったく発生していない。



©mixi, Inc. All rights reserved

オンデマンドインスタンスを効果的に使う

©mixi, Inc. All rights reserved

オンデマンドインスタンスを効果的に使う

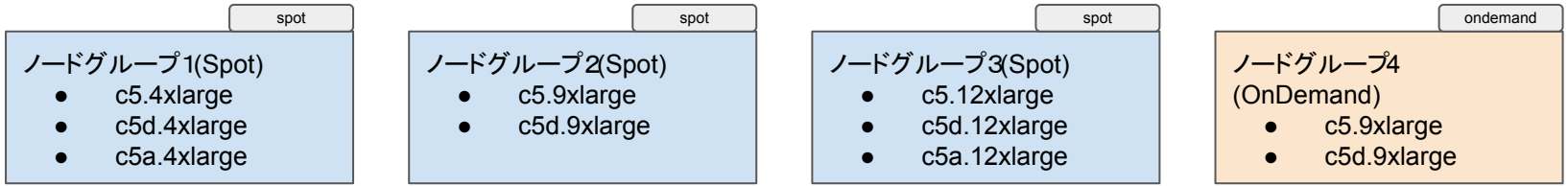
- 中断耐性のないコンテナを動かす
- スポットインスタンス枯渇時の影響を軽減する

©mixi, Inc. All rights reserved

オンデマンドインスタンスを効果的に使う

中断耐性のないコンテナを動かす

- スポットインスタンスには適さないワークロードはどうしても存在する
  - 冪等性を担保できないタスク
  - 長時間実行されるタスク
  - アドホックな作業で利用するコンテナ
- オンデマンドインスタンスのみの AutoScalingGroup を作成する。
  - Node Label と Node Selector を利用して、特定のコンテナだけをオンデマンドインスタンスで実行する。

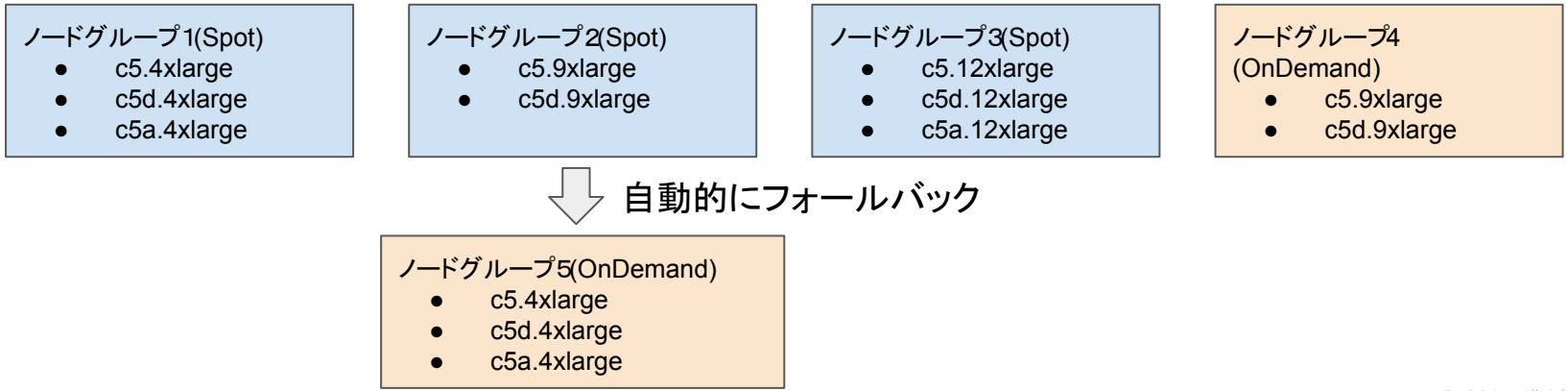


©mixi, Inc. All rights reserved

オンデマンドインスタンスを効果的に使う

スポットインスタンス枯渇時の影響を軽減する

- Cluster Autoscaler の Priority Expander の機能を利用
  - 通常のノードグループがいずれも利用できなくなった際に、自動的にオンデマンドインスタンスにフォールバックするような設定をしている。
  - （ただ、機能テストには成功しているものの、一度もこの機能の出番が発生したことがないので、本番環境での効果は未知数・・）



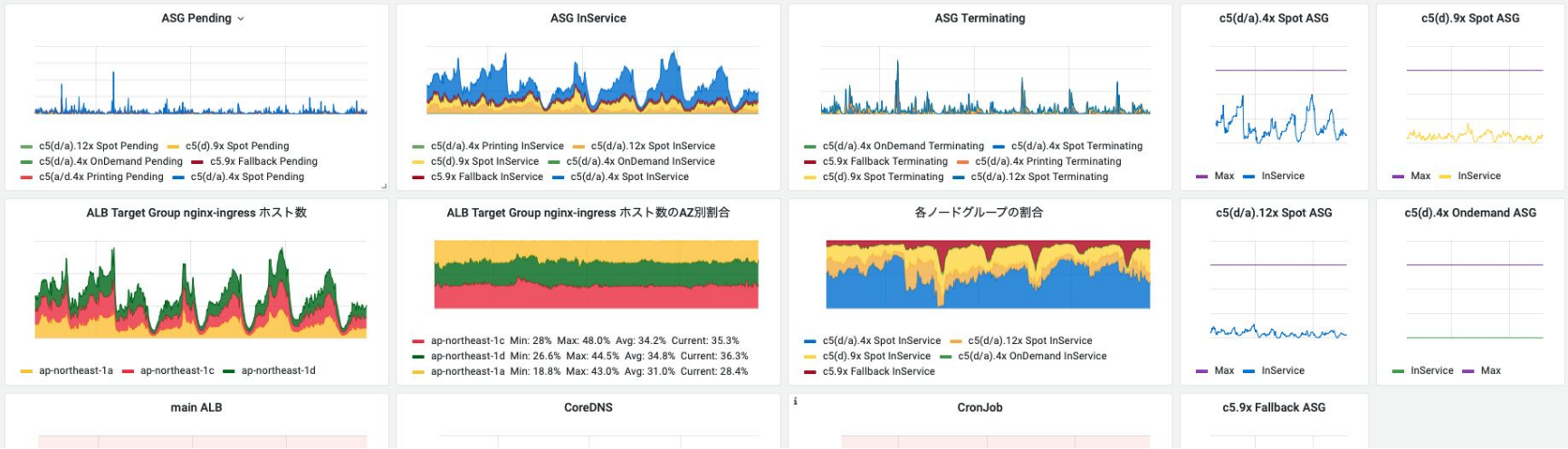
©mixi, Inc. All rights reserved

（おまけ）モニタリングについて

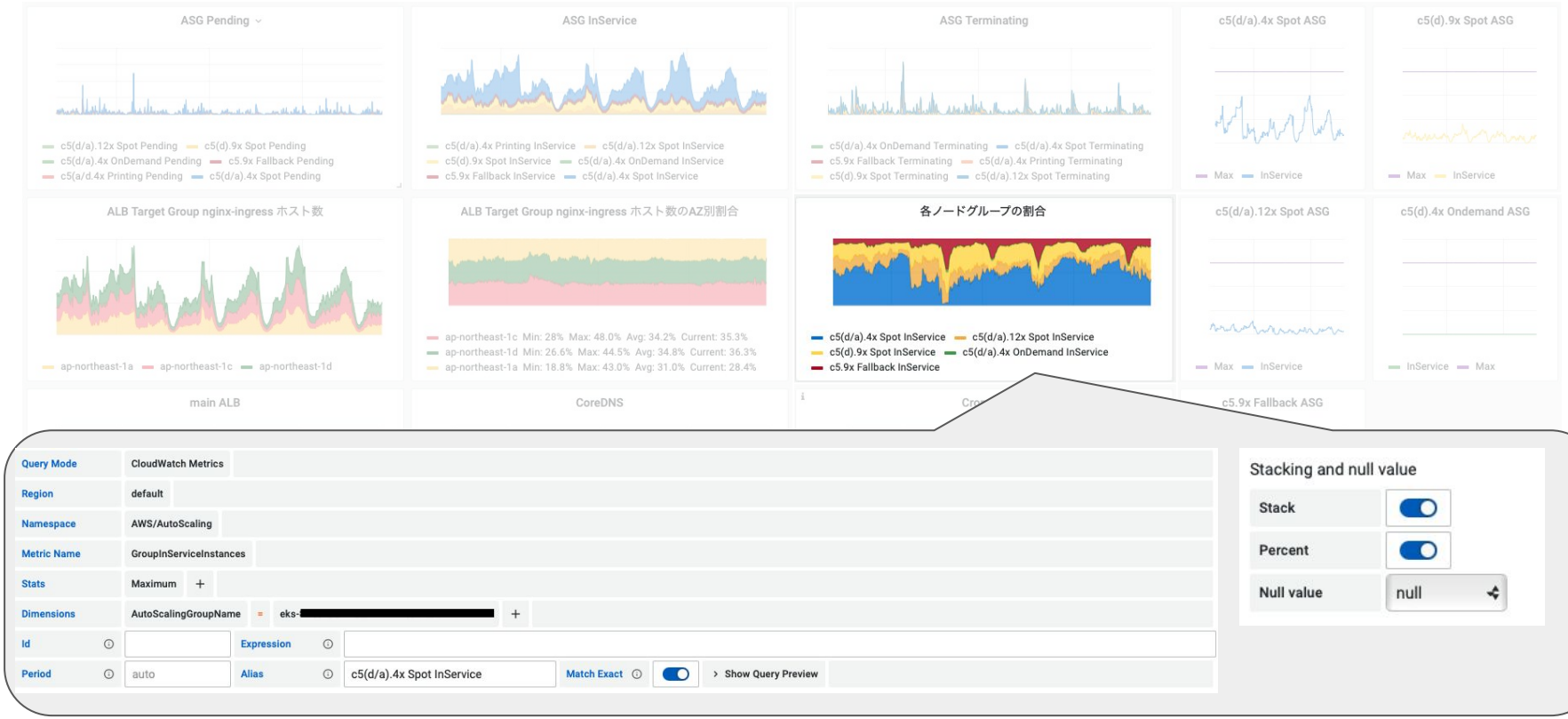
- 朝会でGrafanaのダッシュボードを確認
  - 意図せぬ変化があったときに気がつけるように
  - スポットインスタンスの状況も可視化
- アプリケーションのパフォーマンスやエラー率などを NewRelic APM を利用して監視している
  - スポットインスタンスの不足などによりパフォーマンスが劣化していれば　APM からアラートが飛ぶ

その他、みてねにおける異常検知やチューニングに関する取り組みを以下のリンクで紹介しています。
https://newrelic.com/jp/resources/webinars/mitene-20210617-ty

©mixi, Inc. All rights reserved



©mixi, Inc. All rights reserved


ASG Pending
c5(d/a).12x Spot Pending
c5(d).9x Spot Pending
c5(d/a).4x OnDemand Pending
c5.9x Fallback Pending
c5(a/d.4x Printing Pending
c5(d/a).4x Spot Pending
ASG InService
c5(d/a).4x Printing InService
c5(d/a).12x Spot InService
c5(d).9x Spot InService
c5(d/a).4x OnDemand InService
c5.9x Fallback InService
c5(d/a).4x Spot InService
ASG Terminating
c5(d/a).4x OnDemand Terminating
c5(d/a).4x Spot Terminating
c5.9x Fallback Terminating
c5(d/a).4x Printing Terminating
c5(d).9x Spot Terminating
c5(d/a).12x Spot Terminating
c5(d/a).4x Spot ASG
Max
InService
c5(d).9x Spot ASG
Max
InService
ALB Target Group nginx-ingress ホスト数
ap-northeast-1a
ap-northeast-1c
ap-northeast-1d
ALB Target Group nginx-ingress ホスト数のAZ別割合
ap-northeast-1c Min: 28% Max: 48.0% Avg: 34.2% Current: 35.3%
ap-northeast-1d Min: 26.6% Max: 44.5% Avg: 34.8% Current: 36.3%
ap-northeast-1a Min: 18.8% Max: 43.0% Avg: 31.0% Current: 28.4%
各ノードグループの割合
c5(d/a).4x Spot InService
c5(d/a).12x Spot InService
c5(d).9x Spot InService
c5(d/a).4x OnDemand InService
c5.9x Fallback InService
c5(d/a).12x Spot ASG
Max
InService
c5(d).4x Ondemand ASG
InService
Max
main ALB
CoreDNS
c5.9x Fallback ASG
Query Mode
CloudWatch Metrics
Region
default
Namespace
AWS/AutoScaling
Metric Name
GroupInServiceInstances
Stats
Maximum
Dimensions
AutoScalingGroupName
eks-
Id
Expression
Period
auto
Alias
c5(d/a).4x Spot InService
Match Exact
Show Query Preview
Stacking and null value
Stack
Percent
Null value
null

おわりに

- スポットインスタンスは、いくつかのポイントを押さえれることで、安定して運用できました。コスト効果は抜群でした。
- 今後も更なるコスト削減を目指していきます。
  - Graviton2 インスタンス
  - アプリケーションの効率化・軽量化
  - ストレージコストの削減

©mixi, Inc. All rights reserved

WE ARE HIRING!!

各ポジション積極採用中

team.mitene.us