



今年1年のEKS運用振り返り

3-shake SRE Tech Talk #8

Lightning Talk

株式会社MIXI Vantageスタジオ みてねプロダクト開発部 基盤開発G
清水 勲 @isaoshimizu



清水 勲 @isaoshimizu

家族アルバム みてね Engineering Manager (SRE/CXE/Security)

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」

モンスターなど

みてね

2003年

2011年

2014年

2018年

2024年

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

Fedora/MySQL/LXC
/OpenStack

Linux/MySQL/Ruby

AWS/MySQL/Ruby

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。

New Relic User Group 運営

みてねでは2021年からAmazon EKSを 運用しています

(参考) 4年間のEKS移行の取り組みを振り返って
<https://gihyo.jp/article/2022/11/mitene-02eks>

**EKSに移行して2年ほど。
2023年はどんな運用をしてきたのか。**

これからKubernetesやEKSの運用に関わる方
すでに運用している方

の参考になれば幸いです

みてねのKubernetes構成

- Amazon EKS v1.27 (リリース日: May 24, 2023、標準サポート: July 2024、延長サポート: July 2025)
- Helm Chart
 - EKSクラスタ内のリソースはTerraformでは管理しないルール
 - EKSクラスタ自体はTerraformで管理
- Argo CD
- Argo Workflows
- KEDA (SQSのキュー長ベースのオートスケール用)
- Prometheus (In-Cluster, Short Term)
- Amazon Managed Service for Prometheus (Long Term)
- Grafana Loki
- Grafana
- みてねでの規模感 (2023/12時点)
 - 多い時で18,000Pods、少ない時で2,000Pods

2023年にやってきたこと 7つの事例

ちなみに...
1年間でHelm ChartのPull Request数
600強くらいでした

1.リソース調整

1.リソース調整

- CPU、メモリの調整
- それなりの頻度でさまざまなDeploymentのリソースを調整をしている
- 調整のきっかけ
 - **OOMKilled**
 - 例) Cluster AutoscalerでOOMKilledの発生（つらい）
 - limitsの調整、CPU: 500m -> 2000m、Memory: 1000Mi -> 4000Mi
 - オートスケールされにくいとき
 - CPU利用率が低すぎてHPAが機能しないなど
 - CPUの割り当てが足りなくて性能が出ていないとき
 - Prometheusのメトリクスを見ながら最適値を探す

2.deschedulerの導入

- 導入のきっかけ
 - Railsアプリのメモリリーク。デプロイされない期間（例えば週末や連休）など、ある程度の日数再起動がかからないと、メモリ使用率が高まっていき、502エラーが少し目立つようになってきた。
- deschedulerを利用してPodの最大起動時間(PodLifeTime)をコントロールして、定期的に再起動がかかるようにし、メモリを解放できるように。

```
PodLifeTime:
  enabled: true
  params:
    podLifeTime:
      maxPodLifeTimeSeconds: 28800
  namespaces:
    include:
      - "NAMESPACE"
  labelSelector:
    matchLabels:
      app: APPNAME
```

3.CronJobの追加

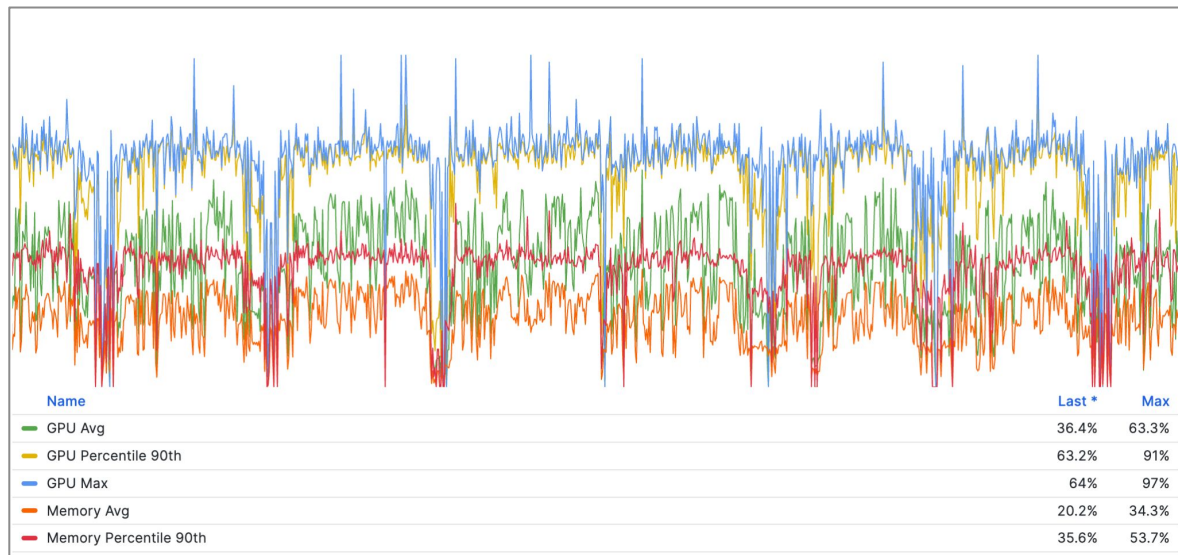
3.CronJobの追加

- 機能開発の中で定時バッチ処理が必要になるケースがある
- 開発者にHelm ChartのPull Requestを作ってもらう
- Templateが用意されているのでValuesとして3行くらい追加するだけ

4.DCGM Exporterの導入

4.DCGM Exporterの導入

- <https://github.com/NVIDIA/dcgm-exporter>
- 一部の機能においてGPUを必要としている
- GPUの各種メトリクスをPrometheusで取得できるように
- 細かく数値を見るというよりは傾向の変化をモニタリングしている



5.Helm Chartのアップデート

- CommunityのHelm Chartをアップデート
 - たとえば、Prometheus、Cluster Autoscaler、kube-state-metricsなど
- アップデートによるメリット
 - 新しいKubernetesバージョンへの対応
 - 新機能の利用
 - 各種脆弱性対応
- 現在は手動運用ということもあって、アップデートが放置されがち
 - Renovateなどによる自動アップデート（自動Pull Request作成）を実現したい
 - まもなくできそう

6.Opsクラスタ導入 & Kubernetesアップデート

6.Opsクラスタ導入 & Kubernetesアップデート(1)

- SRE NEXT 2023で弊社チームメンバーの杉本が登壇した際の内容



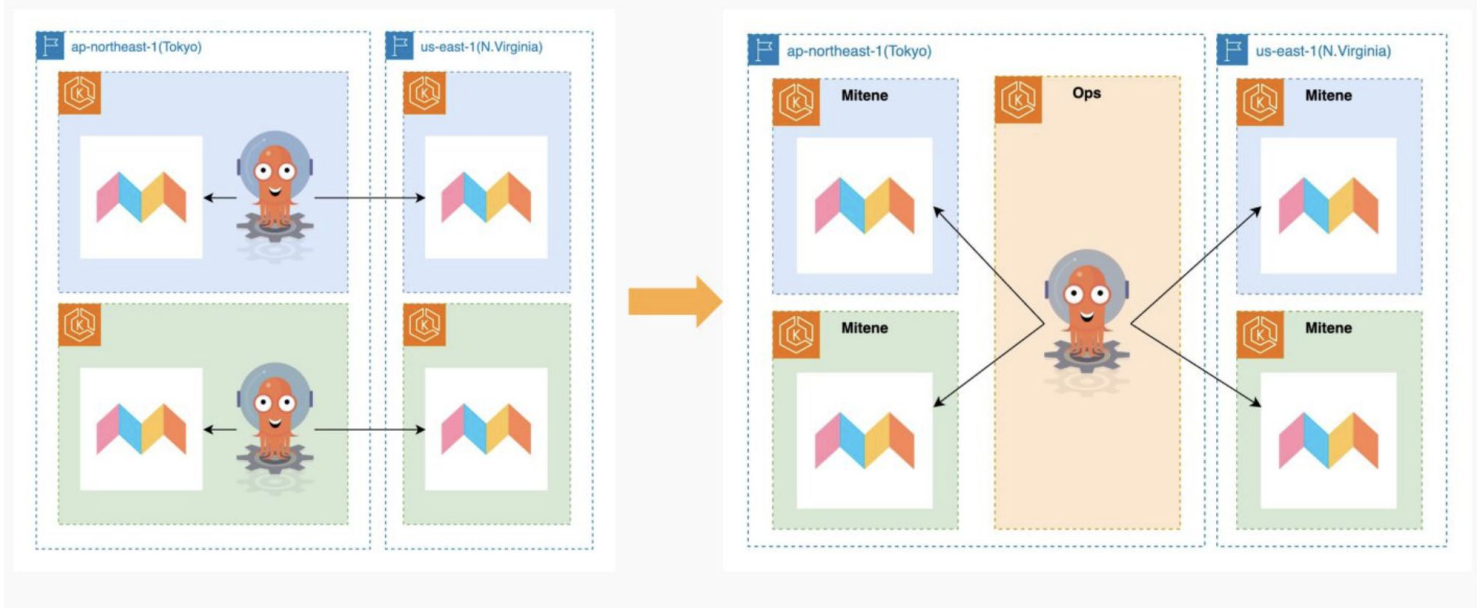
**1,800万人が利用する
『家族アルバム みてね』における
K8s基盤のアップグレード戦略と
継続的改善**

@kohbis
SRE NEXT 2023

<https://speakerdeck.com/kohbis/familyalbums-upgrade-strategy-and-continuous-improvement-for-k8s-infrastructure>

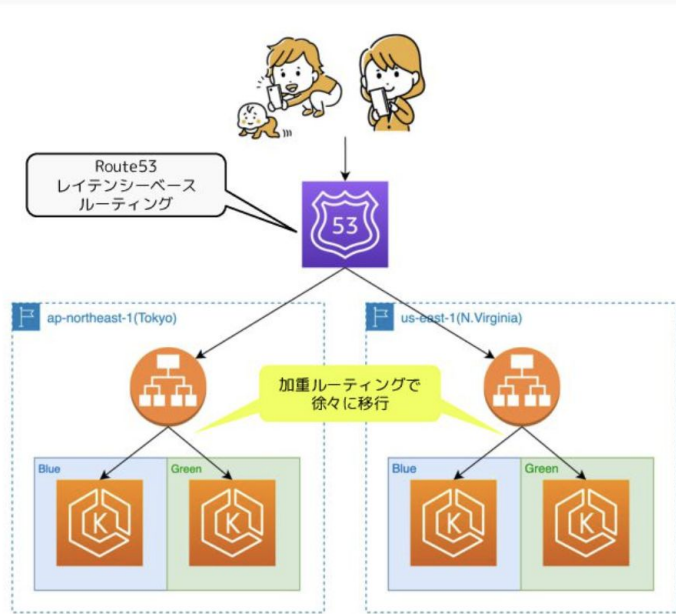
6.Opsクラスタ導入 & Kubernetesアップデート(2)

Argo CDと『みてね』サービスのクラスタ分割



<https://speakerdeck.com/kohbis/familyalbums-upgrade-strategy-and-continuous-improvement-for-k8s-infrastructure?slide=27>

EKSクラスタのBlue/Green戦略による無停止アップグレード



- みてねはAWSマルチリージョン構成
- 複数リージョンにEKSクラスタが存在

各リージョンに新バージョンのクラスタ作成



ALBの加重ルーティングで徐々にトラフィック移行



メトリクス監視に異常がなければ完全移行

<https://speakerdeck.com/kohbis/familyalbums-upgrade-strategy-and-continuous-improvement-for-k8s-infrastructure?slide=12>

7.kubectlでPod作業するのを禁止

7.kubectlでPod作業するのを禁止

- kubectl execでPodのシェルを起動して作業するのを禁止しました
 - 作業のログが残らないため
 - 開発者の pods/exec などの権限を剥奪
- 代替手段として、ローカルPCからSession Managerを使って踏み台サーバーを経由してkubectl execを実行する仕組みを構築
- これによってSession ManagerのログがS3に保存され、すべてのPod内での作業が記録されるように

まとめ

- Kubernetes(EKS)運用3年目の実態についてお話ししました
- コストパフォーマンスや安定性のためにリソース量の調整はいつまでも続いていく
- 開発者が直接運用できる箇所は今後も増やしていきたい
- Kubernetesのアップグレード運用はだいぶこなれてきていますが、まだまだ改善の余地はあります
- クラスタのアップグレードよりもHelm Chartのアップデート運用のほうが課題
 - アプリケーションのライブラリアップデート運用と同じような課題だが、自動化のツール整備がまだ追いついていなかった（Renovateで解決しよう）
- Kubernetesにまつわる様々なツール群（OSS）に支えられています

心もつなごう。

MIXI

