みてねSREチームの取り組み

みてねのMeetup #2 for サーバーサイド/SRE

September 5, 2018

Isao Shimizu

About me

清水 勲 @isaoshimizu

株式会社ミクシィ
ヴァンテージスタジオ みてね事業部 開発グループ

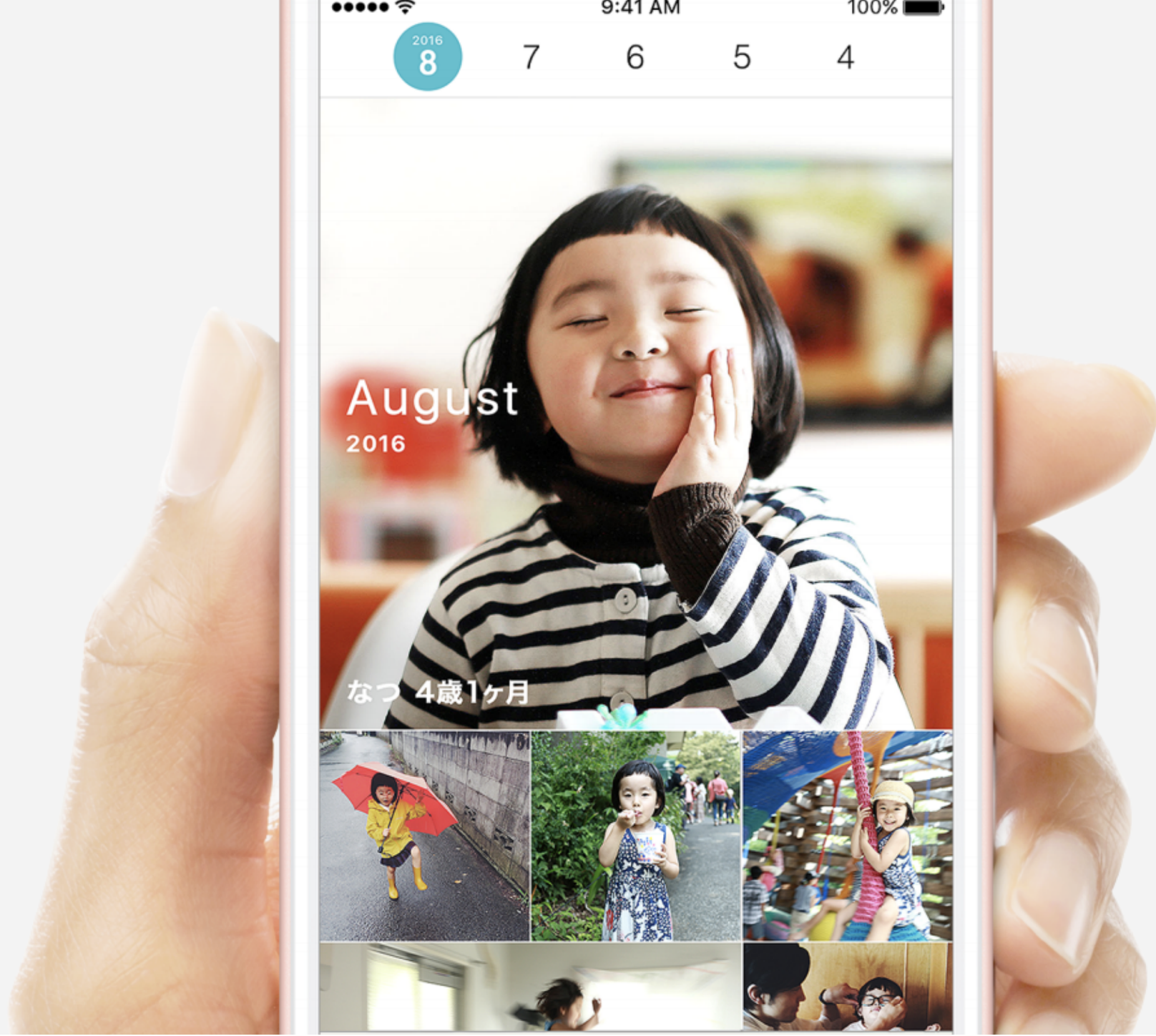
- 2011年8月 中途入社、SNS mixiのアプリ運用チーム
- 2014年4月 モンストのサーバーエンジニア、SRE
- 2018年2月 みてねSRE

AWS Summit, TechLION, hbstudy, Internet Week, Think Japan IBM Code Dayなどの登壇

Qiita, Medium, Software Design 記事執筆

記事: 大規模インフラのSREから社内スタートアップへの挑戦。ベテランエンジニアのキャリアの作り方

「みてね」とは


9:41 AM
100%
2016
8
7
6
5
4
August
2016
なつ 4歳1ヶ月

300万人が使う No.1 家族アルバムアプリ「みてね」

- 公式ページ https://mitene.us/
- 特徴
  - 子供の写真や動画を共有、整理
  - 家族だけで安心して使えるアルバムアプリ
  - フォトブック、DVDの販売
- 2015年4月 サービス開始
- 2017年7月 英語版提供開始
- 2018年7月 ユーザー数 300万人突破

この約半年、SREチームがやってきたこと

OS、ライブラリのアップデート

- 古いAmazon Linuxを最新に
- ImageMagick
- OpenCV

ミドルウェアのアップデート

- Amazon ElastiCache
- Redis
- Connection増問題
  - `tcp-keepalive` が起因していた可能性
  - Redis 3.2以降の場合 `tcp-keepalive` がデフォルトで設定される(300秒)

ログ転送の最適化

- Fluentd Aggregatorの撤廃
  - よく詰まる問題が発生していた
  - 設定のチューニングに追われる日々
- Amazon Kinesis Data Firehoseの利用
  - S3へのログ保存がより簡単に

CDN最適化

- APIアクセスをCDN経由に
- 署名付きURLの効率化
  - ワイルドカードを使った一括取得
  - URL自体をクライアントサイドでキャッシュ
    - 海外展開を目指すアプリで実装したセキュアで高速な画像配信の話
- TTL調整
  - 長くて良いものはより長く

IPv6対応（結果: 未対応）

- 転送、レイテンシーの高速化を狙う
- 対象はAPI、画像、動画アクセス（CloudFront、S3）
- 端末、OS、キャリアとの相性により接続不能な環境があった
- 本格的に対応した事例があったらぜひ教えてください

Amazon Aurora移行

- RDS(MySQL 5.6)からの移行
- クエリのキャパシティが大幅に増えた
- ディスクサイズ制限からの開放
- 年間におけるピークも無事に乗り切った

開発環境の再整備

- ステージング環境の整備
  - エンジニアの増員により、さらに必要になった
- CircleCI最適化
  - Ubuntu -> Amazon LinuxのDockerイメージ利用
  - 契約プラン見直し、並列度の調整
  - Schedule Jobの活用
    - 1日1回 `bundle outdated` で古いgemの検出
    - `simplecov` を使ったテストカバレッジ計測

Ruby、Gemのアップデート

- 2.2.7 -> 2.3.7 -> 2.4.4までアップデート
- さらにアップデートし続けていく
  - 環境によっては最新版
- 数多くのgemもアップデート
- Railsアップデートは進行中（4系 -> 5系へ）

Amazon S3最適化

- Amazon S3 Transfer Accelerationの利用
  - 主に海外からのアップロードの高速化
- バージョニングのライフサイクル設定
- 不要なバケットの削除

分析基盤のリニューアル（進行中）

- EC2上でのスクリプト実行やCron実行から、Luigiを使ったワークフロー管理へ
  - https://github.com/spotify/luigi
- マネージドサービスを中心に設計
- インフラのコード化、デプロイの自動化などもあわせて

Terraformへ移行

- Terraform https://www.terraform.io/
- 手軽なCloudWatch Alarmから移行し、IAM、Security Group
- miam、piculetから移行（さらに管理対象を増やしていく）
- CIとの連携実装済み
  - GitHub Pull Request作成で `terraform plan`、マージで `terraform apply`
- tfnotifyの活用 https://github.com/mercari/tfnotify
- workspaceは使わずディレクトリで分ける構成

コンテナ環境への移行（進行中）

- AWS OpsWorksから移行
- Amazon ECS, EKSの検証
- コンテナイメージのビルドフロー構築
  - AWS CodeBuild積極活用
- Docker HubからAmazon ECRへ移行
- 開発環境のコンテナ化
- 本番環境のコンテナ化（Kubernetes利用予定）

その他

- Nginx gzip対応
  - JSONレスポンスの圧縮
- AWS SDKのアップデート
- New Relic Mobile導入
- AWSメンテ対応（Redis）
- 不要なEBS、RDS、EC2インスタンス、スナップショットの削除

これからもさらなる改善を続けていきます!

Join our team!!

https://mitene.us/recruit