SRE NEXT 2020 IN TOKYO

2020.1.25

# SREがセキュアなWebシステムを構築、維持するためにやれることはなにか

Isao Shimizu

mixi, Inc.

# 自己紹介

## 清水 勲 （しみず いさお） @isaoshimizu

所属

- 株式会社ミクシィ Vantageスタジオ みてね事業部 開発グループ SREチーム

経歴

- SIerで受託開発、プロダクト開発を経験後、2011年に株式会社ミクシィへ入社
- SNS運用、モンスターストライクのSREを経て、**現在「家族アルバム みてね」のSRE**
- 2019年1月 Software Design 2019年2月号 寄稿
  **SREって，なにをやればいいの？ 先駆者に訊く，誰がための信頼性**
- 2019年5月 AWS Summit Tokyo
  **コンテナ移行ってこんなに大変？ ～「家族アルバム みてね」を支えるインフラの裏側～**
- その他「SRE Tech Talks #2」「hbstudy #76」「Internet Week 2017」等

# アジェンダ

1. 「家族アルバム みてね」について
2. はじめに
3. SREについて
4. SREとセキュリティ
5. セキュリティへの取り組み事例
6. クラウドを扱う際に取り組むべきセキュリティ対策（AWS編）
7. まとめ

# 1 家族アルバム みてね

# 1.「家族アルバム みてね」について

**子どもの写真・動画を、無料・無制限に共有できるアプリ**

**2015年サービスリリース**

現在、**国内外**の利用者数 **600**万人以上

**多言語対応**（日本語、英語、韓国語、繁体字）

App Store レビュー **★4.7**

Google Play ストア レビュー **★4.7**

みてね　FamilyAlbum　家家相簿

# 1.「家族アルバム みてね」について

## 海外での受賞歴

## 2019 THE WEBBY AWARDS

国際デジタル芸術科学アカデミー (IADAS) によって毎年主催され、優れたインターネットに贈られる賞

## 2019 NATIONAL Parenting Product(NAPPA) AWARDS

家族にフォーカスした賞（おもちゃや家族向けの商品に特化）

# 1.「家族アルバム みてね」について

**「みてねプレミアム」を2019年4月リリース**

月額課金型の有料プラン

月額 480円

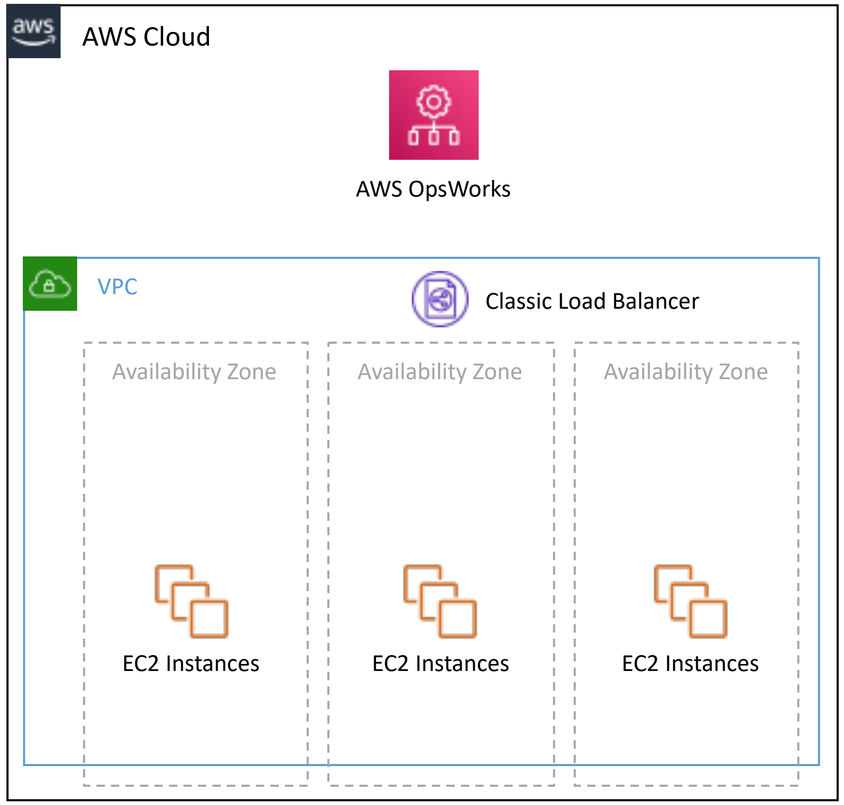
- ブラウザから写真や動画のアップロードが可能に
- 1秒動画の**毎月&年間版**配信
- 全商品が**送料無料**（家族全員）
- 公開範囲を細かく設定可能
- 動画アップロードの時間制限を延長 3分から**10分**へ

**みてねをお使いの方、ぜひお試しください！**

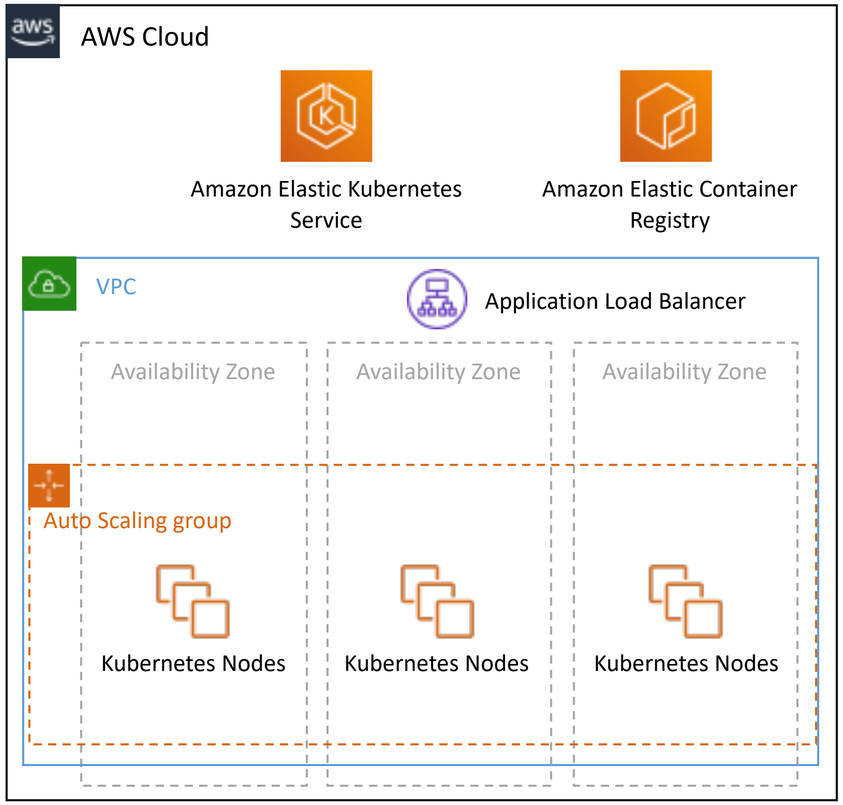
# 1. 「家族アルバム みてね」について

## みてねのインフラはKubernetes（Amazon EKS）への移行中

**Before**




**After**



# 2 はじめに

# 2. はじめに

## この発表について

- SREがセキュリティにどう取り組むと良いかにフォーカスしてお話します。
- セキュリティがなぜ重要なのかについてはあまり話しません。
- みてねでの取り組みの事例について一部紹介します。
- みてねのインフラやアプリケーション開発の詳細については取り上げません。
- AWSを使っている関係で、AWSに特化した内容がありますがご了承ください。
- ハッシュタグ **#srenext #srenextA**

# 2. はじめに

## 本日お持ち帰りいただきたいこと

- SREについての基本的なこと
- SREとセキュリティに関する各種プラクティス
- セキュリティへの取り組み事例からの学び

3

# SREについて

# DevOpsとの違い

# 3. SREについて

## *class SRE implements DevOps*

- “DevOpsをプログラミング言語におけるインターフェースと捉えると、クラス SRE はDevOpsの実装である。”
- “SREには、DevOpsのインターフェースの部分に限らない追加のプラクティスと推奨事項が含まれる。"

SRE vs. DevOps: competing standards or close friends?
https://cloud.google.com/blog/products/gcp/sre-vs-devops-competing-standards-or-close-friends

# 3. SREについて

*DevOps and SRE are not two competing methods for software development and operations, but rather close friends designed to break down organizational barriers to deliver better software faster.*

“DevOpsとSREは、ソフトウェア開発と運用が競合するような手法ではなく、**より良いソフトウェアをより速く提供するために組織の障壁を打破する**ように設計された親密なものである。”

SRE vs. DevOps: competing standards or close friends?
https://cloud.google.com/blog/products/gcp/sre-vs-devops-competing-standards-or-close-friends

# 3. SREについて

Home » Blogs » Leadership Suite » Defining the Role of a SRE

## Defining the Role of a SRE

BY DAWN PARZYCH ON APRIL 25, 2018 — 0 COMMENTS

Defining the Role of a SRE
https://devops.com/defining-the-role-of-a-site-reliability-engineer-sre/

# 3. SREについて

*Site reliability engineering (SRE) is a relatively new discipline, having only been in existence for about 15 years. It originated at Google and has gained popularity recently with more companies advertising SRE positions or trying to implement SRE practices. In the technology field, 15 years may seem like an eternity, but the SRE role is very much still in its infancy. There are still challenges defining the role and understanding exactly what it is. Just look through one of the 1,000-plus job listings on LinkedIn for an SRE in the United States—you will see many different job expectations and requirements.*

“SREは比較的新しい分野で、約15年しか存在していません。Googleで始まり、最近ではSREの職種を宣伝したり、SREのプラクティスを実装しようとする企業が増えています。テクノロジー分野では、15年は長い時間のように思えるかもしれませんが、**SREの役割はまだ初期段階にあります。役割を定義し、それが何であるかを正確に理解するという課題がまだあります。**LinkedInの1,000を超える求人リストの1つで北米のSREを調べるだけで、さまざまな求人と要件が表示されます。”

Defining the Role of a SRE
https://devops.com/defining-the-role-of-a-site-reliability-engineer-sre/

# 3. SREについて

*An SRE must possess a blend of technical as well as soft skills and, perhaps most of all, must be adept at communicating effectively and inspiring a teamwide approach to excellence.*

“**SREは技術的スキルとソフトスキルの両方を備えている必要**があり、おそらく何よりも、**効果的なコミュニケーションに熟達していることや、チーム全体の卓越性へのアプローチを促す**ことができなければなりません。”

Defining the Role of a SRE
https://devops.com/defining-the-role-of-a-site-reliability-engineer-sre/

# 3. SREについて

*Ashi Sareen*

*Director of Engineering, Site Reliability LinkedIn*

*“A lot is people are just renaming their ops teams to call SRE. I don’t think that’s the right approach. SRE discipline requires investment it has a very clear set of practices set of team dynamics that makes a successful SRE team.”*

“多くの人は、運用チームの名前を変更してSREと呼んでいるだけです。私はそれが正しいアプローチだとは思いません。 SREの統制には、SREチームを成功させる明確なチームダイナミクスのプラクティスセットが必要です。”

NGINX Conf 2019: Site Reliability Engineering Panel - LinkedIn, Dell, and Gremlin
https://www.youtube.com/watch?v=OODH72ZDgM8

# 4 SREとセキュリティ

# SREcon

**年に3〜4回開催されるグローバルなSREカンファレンス**

2020: Americas West, Asia/Pacific, Europe/Middle East/Africa, Americas East
2019: Americas, Asia/Pacific, Europe/Middle East/Africa

# SREconにおけるセキュリティ関連セッション（抜粋）

# 4. SREとセキュリティ

## SREcon18 America / March 27–29

LinkedIn

### Security and SRE: Natural Force Multipliers

https://www.usenix.org/conference/srecon18americas/presentation/scott

*“Remove single points of security failure like you do for availability”*

Facebook

### Security as a Service

https://www.usenix.org/conference/srecon18americas/presentation/wojtyniak

*“Understand your threats. Defense in depth. Make it simple And stable.”*

# 4. SREとセキュリティ

## SREcon19 America / March 25–27

Pivotal

### Extending the Error Budget Model to Security and Feature Freshness

https://www.usenix.org/conference/srecon19americas/presentation/thomson

*“Vulnerability budget: How long can we afford to be vulnerable to CVEs in our dependencies?*
*Legacy budget: New enough that it gets support; and not so old that no-one wants to use it.”*

# 4. SREとセキュリティ

## SREcon19 Asia / June 12-14

PayPal

## Ensuring Site Reliability through Security Controls

https://www.usenix.org/conference/srecon19asia/presentation/janakiraman

*“1/5th of internet traffic is bad!*
*Layer 7 Application Defense is the most effective mechanism to detect & mitigate bad traffic”*

# 4. SREとセキュリティ

## SREcon19 Europe Middle East Africa / 2–4 October

MIT

## A Systems Approach to Safety and Cybersecurity

https://www.usenix.org/conference/srecon19emea/presentation/leveson

*“Looks at system as whole, not just components.*
*Takes s larger view of causes than just failures.”*

# 4. SREとセキュリティ

**SREcon 2020 Americas West**
SANTA CLARA, CA, USA March 24–26, 2020

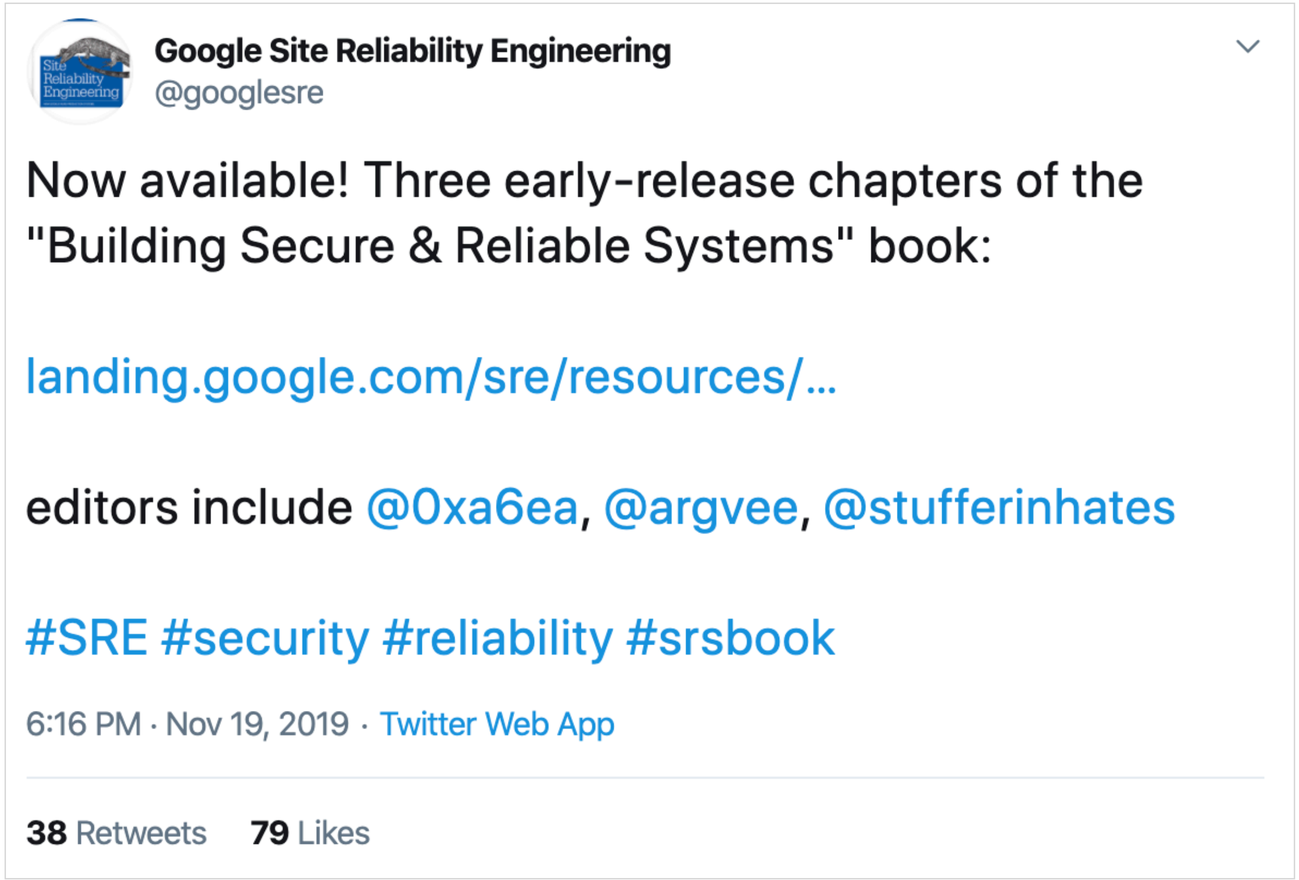
**SREcon 2020 Asia/Pacific**
SYDNEY, AUSTRALIA June 15–17, 2020

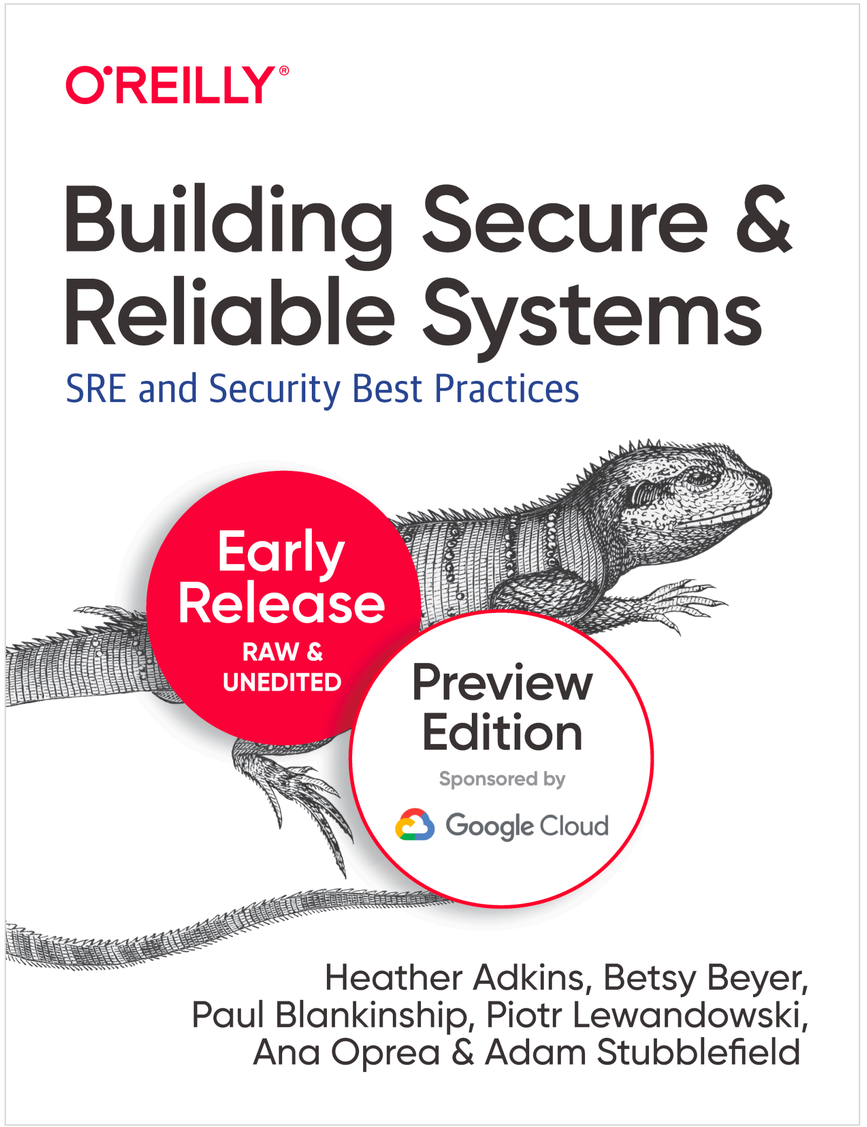
**SREcon 2020 Europe/Middle East/Africa**
AMSTERDAM, NETHERLANDS

**SREcon 2020 Americas East**
BOSTON, MA, USA

# "Building Secure & Reliable Systems SRE and Security Best Practices”

# 4. SREとセキュリティ




Tweet https://twitter.com/googlesre/status/1196718932247355392
公式 https://landing.google.com/sre/resources/foundationsandprinciples/srs-book/
Amazon.com https://www.amazon.com/Building-Secure-Reliable-Systems-Practices/dp/1492083127

# 4. SREとセキュリティ

## "Building Secure & Reliable Systems SRE and Security Best Practice” Early Release

- セキュリティーと信頼性をテーマにした書籍。
- Googleのエンジニア6名による執筆。
- Early Releaseのチャプターは以下の3つのみ（その他の章は現在不明）
  - 1. The Intersection of Security and Reliability
  - 2. Understanding Adversaries
  - 3. Testing Code

# Chapter 1:
# The Intersection of Security and Reliability
# の要点まとめ

# 4. SREとセキュリティ

- 信頼性とセキュリティは、ソフトウェアとシステムのライフサイクルに不可欠である。
- 信頼性とセキュリティを両立したシステムを構築するのは簡単ではない。
- どちらもあとで実装しようとするのは難しい。
- **初期から考慮できていること**が望ましい。
- 問題が起きないとコストをかけない領域。
- **問題が起きてからだと深刻になりがち**。
- **シンプルな設計**、**最小化された権限管理**が重要である。

# 4. SREとセキュリティ

- よくできた**ロギング**は脅威の検出と障害への備えとなる。
- 有事の際の**指揮系統**、**チェックリスト**、**プレイブック**、**プロトコル**は重要。
- 脆弱性パッチは**素早い適用が重要**。
- パッチの内容に次第では**問題（バグ、性能劣化など）を起こす可能性**もある。
- リスクを考慮して、適用タイミングについてビジネスサイドとも調整する。

**他にも様々なプラクティスが載っているので興味のある方はお読みください。**

- 第1章初めの「On passwords and power drills」。
- Early Release版なので出版時には内容に変更がある可能性があります。

5

# セキュリティへの取り組み事例

# 5. セキュリティへの取り組み事例

**SREチームは組織内の様々なチームと連携して**
**様々な課題に取り組む**

**チーム間におけるコミュニケーションはとても大事**

# SREチームにおけるコミュニケーションの原則

# 5. セキュリティへの取り組み事例

**（私が考える）SREチームにおけるコミュニケーションの原則**

- 開発チームなど、他チームとの間に壁を作らない
- 押し付け合わない
- 冷静にしっかりと議論する
- お互いの状況の理解し合う
- ビジネスや事業にとって優先すべきかどうか話し合う

# セキュリティの向上に貢献できる施策事例

# 5. セキュリティへの取り組み事例

## セキュリティの向上に貢献できる施策事例

- 開発チームへのヒアリング
- Infrastructure as Code
- ログ収集と検索
- アップデート
- モニタリング、定期レポート
- ポストモーテム

**SREチームが開発チームに対して**
**随時ヒアリングを実施**

（例: 新機能や新サービスを開発する時など）

# ヒアリング内容の例（1）

- どんなライブラリを使用しているか
- ユーザーのトークンの取り扱い
  - 有効期限
  - 保存場所はどこか
- 一般的なXSS、CSRFの対策はできているか
- APIのパラメータを直接変更して、他人のデータを見ることはできないか

# ヒアリング内容の例（2）

- アプリケーション内部に秘匿情報を持っていないかどうか
- ユーザー管理のパスワード
  - 平文で保存していないか
  - 平文で送信（メールなど）していないか
  - パスワードの強度（文字数、文字種など）は適切か
  - リマインダの手法はどんなものか

# ヒアリング内容の例（3）

- クライアント・サーバー間の通信にTLSv1.2以降のプロトコルを使っているか
- クライアントに保持されるデータ
  - ライフタイムはどのくらいか
  - データの保存場所はどこか

# ヒアリング内容の例（4）

- アクセスログはユーザーを特定できる形で残しているか
- ログ上の個人情報（氏名、メールアドレス、住所など）はマスク処理しているか
- パスワード総当り攻撃を検知出来るか
- アプリ側に脆弱性が見つかったとき等、アップデートを強制できるようになっているか
- どこの決済サービスを使っているか
- どんな実装になっているか

# Infrastructure as Code

# 5. セキュリティへの取り組み事例

## Infrastructure as Code (IaC)

- インフラの構築を手作業でやらない
  - 属人的、変更管理、設定やドキュメントの抜け漏れなどの問題が起きやすい
- IaCのツールを使う（Terraform、CloudFormationなど）
- Pull Requestを作り、コードレビューを受ける
  - **例: IAMやセキュリティグループが最小限となっているか**
- CIツールと連携してdry-runを実行する
- Pull Requestのマージによってインフラに適用する

# ログ収集と検索

# 5. セキュリティへの取り組み事例

## ログ収集と検索

- 異常なアクセス、トラフィックの検出
- アクセス元の脅威検出
- ログには認証情報や個人情報などの機密情報は載せない
- ログの量はコストに影響しやすい（一時保存先、長期保存先）
- 不要なログがないかどうか定期的にチェックする
- クラウド側のログ（例: S3、CloudFront、ELBなど）もできるだけ有効化
- ログを収集、検索できるSaaS利用も検討する

# アップデート

# 5. セキュリティへの取り組み事例

**アップデート対象の例**

- アプリケーションに依存するライブラリ
- OS、MySQL、Redis、Nginx、Fluentdなどのミドルウェア、各種パッケージ
- モニタリングなど常時起動しているエージェントソフトウェア
- Kubernetesクラスタ、Helmチャート
- Terraform、モジュール

脆弱性は定期的に発見され修正される。いずれも自動化しないと放置されがち。

自動化が困難な場合、定期的なアップデートの仕組み作りが大事。

# モニタリング、定期レポート

# 5. セキュリティへの取り組み事例

**モニタリング、定期レポートの例**

- ネットワークトラフィックやデータベースの負荷など、意図しない負荷が発生していないか定期的に確認。
- 認証の失敗回数、しきい値を超えた回数を定期レポートする。

ログには保存しているが、手動で毎回集計するのは忘れがちになる。

自動で定期的に集計してグラフ化するなど工夫するのがおすすめ。

# ポストモーテム

# 5. セキュリティへの取り組み事例

**ポストモーテム**

- 障害や問題が発生した際に事象や対応を記録する。
- セキュリティに限定したものではない。
- サービスへの影響の有無に関わらず、積極的に作成する。
- ポストモーテムの**作成者を称賛する**文化を作る。
- **絶対に責めたり、批判しない。**

**SRE以外（できるだけ上の役職の人がよい）も巻き込んで、定期的にポストモーテムを振り返り、改善を続けることが理想。**

6

# クラウドを扱う際に取り組むべきセキュリティ対策（AWS編）

# 6. クラウドを扱う際に取り組むべきセキュリティ対策（AWS編）

## （例）AWS のセキュリティ基本コンセプト

“AWS は 2006 年からクラウドコンピューティングの先駆者としてセキュアなシステムを構築し、お客様のイノベーションに迅速に対応可能なクラウドインフラストラクチャを創造してきました。**クラウドセキュリティは AWS の最優先事項です**。AWS ではセキュリティやコンプライアンス上の統制を実装、オートメーション・システムを構築し、第三者監査によるセキュリティやコンプライアンスについての検証が実施されています。”

https://aws.amazon.com/jp/aws-ten-reasons/

# AWS Well-Architected Framework

“クラウドアーキテクトがアプリケーション向けに実装可能な、
**安全で高いパフォーマンス**、障害耐性を備え、効率的なインフラストラクチャを構築するのをサポートする目的で開発”

https://aws.amazon.com/jp/architecture/well-architected/

## AWS Well-Architected Framework

5つの柱

- 運用上の優秀性
- **セキュリティ**
- 信頼性
- パフォーマンス効率
- コスト最適化

# 6. クラウドを扱う際に取り組むべきセキュリティ対策（AWS編）

## AWS Well-Architected Framework 「セキュリティ」

1. アイデンティティ管理とアクセス管理
2. 発見的統制
3. インフラストラクチャ保護
4. データ保護
5. インシデント対応

# 6. クラウドを扱う際に取り組むべきセキュリティ対策（AWS編）

**アイデンティティ管理とアクセス管理**

- SEC 1 認証情報と認証をどのように管理していますか?
- SEC 2 人為的なアクセスをどのように制御していますか?
- SEC 3 プログラムによるアクセスをどのように制御していますか?

**発見的統制**

- SEC 4 セキュリティイベントをどのように検出し、調査していますか？
- SEC 5 新しいセキュリティ脅威に対してどのように防御していますか?

# 6. クラウドを扱う際に取り組むべきセキュリティ対策（AWS編）

**インフラストラクチャ保護**

- SEC 6 ネットワークをどのように保護していますか?
- SEC 7 コンピューティングリソースをどのように保護していますか?

**データ保護**

- SEC 8 データをどのように分類していますか?
- SEC 9 保管中のデータをどのように保護していますか?
- SEC 10 伝送中のデータをどのように保護していますか?

**インシデント対応**

- SEC 11 セキュリティインシデントにどのように対応していますか?

各項目の対応状況を定期的にチェックし
必要があれば今後の対応の優先度を検討する

# AWSを扱う際に気をつけるべき基本的なポイント

# AWSを扱う際に気をつけるべき基本的なポイント

## 内容について

- 記載の内容はどれも基本的なものです。
- AWSが公式に推奨しているものではありません。
- 今までの運用経験やAWSのドキュメントなどを参考にしています。
- 今後のAWSのアップデートによって、内容が変わる可能性があります。
- MUSTかSHOULDかどうかは自身にてご判断ください。

# 6. クラウドを扱う際に取り組むべきセキュリティ対策（AWS編）

## セキュリティサービス

- AWS CloudTrail、Amazon GuardDuty、AWS Configの有効化する。
- AWS Trusted Advisorで定期的にセキュリティチェックする。
- AWS Certificate Managerで発行したTLS証明書を使う。
- AWS Well-Architected レビューを受ける。
- Amazon ECRの脆弱性スキャンを使う。

## VPC、セキュリティグループ

- デフォルトVPC、デフォルトセキュリティグループは使わない。
- セキュリティグループの命名規則を設ける。
- パブリックサブネットとプライベートサブネットを分ける。
- TCP 22番ポートを使ったSSHアクセスをやめて、Session Managerを使ったSSHアクセスにする。

# 6. クラウドを扱う際に取り組むべきセキュリティ対策（AWS編）

## IAM

- ソースレポジトリ内にアクセスキー、シークレットキーを保持しない。
- マネージメントコンソールへログインできるIAMユーザーはMFAを設定する。
- 退職者/異動者など利用しなくなったIAMユーザー/アクセスキーは削除する。
- ルートユーザーにMFAを設定する。
- アクセスキー、シークレットキーを共有しない。
- 外部サービスにアクセスキー、シークレットキーを設定しない。
- AWSアカウントを環境ごとに分ける。
- スイッチロールする場合はMFAを必須とする。

# その他（1）

- 出所不明のパブリックAMIを使わない。
- できるだけ新しいバージョンのAMIを使う。
- S3バケットを直接インターネットに公開しない。
- CloudFrontキーペアファイルをソースコードレポジトリに含めない。
- CloudFrontのTLSセキュリティポリシーはTLSv1.2以降を使う。

# その他（2）

- RDSはインターネットからアクセスさせない。
- RDSのパッチを定期的に適用する。
- ECSコンテナエージェントを定期的にアップデートする。
- EKSクラスタを定期的にアップデートする。
- AWSアカウントを環境ごとに分ける。

# 6. クラウドを扱う際に取り組むべきセキュリティ対策（AWS編）

## AWSに限らずやったほうがよいこともまとめておくと…

- 責任共有モデルを理解する（クラウド内のセキュリティはユーザーの責任）。
- クラウドに用意されているセキュリティソリューションを活用する。
- ネットワークを分離する（インターネットからのアクセスは最小限に）。
- 不要なポートは空けない。
- 最小限のIAMポリシーを設定する。
- アップデートできるものは定期的にアップデートする。

# 7

# まとめ

# 7. まとめ

- SREがセキュリティに貢献できることはたくさんある。
- 信頼性（Reliability）とセキュリティはできるだけ両立させたい。
- インフラ、アプリケーションどちらも早い段階からセキュリティを考慮できることが理想的。
- セキュリティに関する課題や問題への対処はできるだけ早いほうが良い。
- 問題を発見しやすい環境づくりが大事。
- 手間がかかることは後回しにされやすいので、できる限り自動化する。
- クラウドで用意されているセキュリティソリューションを活用する。