エンジニア必見！SREへの第一歩

この資料を書いた、中の人紹介

株式会社スリーシェイク　Sreake事業部部長

手塚 卓也

自治体やデータベースマーケティング会社でのインフラ設計 /構築/運用を主に経験し、2018 年 10 月に 3-Shake に Join。

3-Shake Join 後は Google Cloud / AWS / kubernetes / ServiceMesh など様々な技術的アプローチを駆使し、
大手からベンチャー等規模を問わず様々な組織に対して SREのコンサルティングや実践を行っている。

趣味はボクシング観戦
※ 日曜日の昼間は一人で WOWOW 見ながら一人で熱狂してます ...

twitterでも情報発信しています！
@tt0603ぜひフォローをお願いいたします

アジェンダ

1. About 3-shake
2. SRE とは?
3. Sreake 流 SRE 実践
4. まとめ

こちらの資料の内容は動画でもご覧いただくことが可能です。
是非合わせてご活用ください https://youtu.be/s1x6p7WF_1Y

1. About 3-Shake

About 3-Shake

3-SHAKE

Sreake

SRE 支援 / 技術支援サービス

Sreake

Sreake は金融・医療・動画配信・AI・ゲームなど技術力が求められる領域で豊富な経験を持つ SRE が集まったチームによる技術支援サービスです。戦略策定から設計・構築・運用、SaaS 提供まで、幅広い領域をサポートします。

Sreake Security

セキュリティ脆弱性診断サービス

Sreake Security

Sreake Security は経験豊富なセキュリティ専門家が貴社の課題に合わせたセキュリティ対策を支援するサービスです。

Reckoner

クラウド型 ETL / データ パイプライン サービス

Reckoner

Reckoner はクラウド型 ETL / データ パイプライン サービスです。

データベース・ストレージ・アプリケーションなど、あらゆるデータを統合・連携することで、データを活用したビジネス変革に貢献します。

Relance

ハイスキル フリーランス紹介サービス

Relance

Relance は、プロの現役エンジニア集団が最適なエンジニアを紹介するフリーランス エンジニア紹介サービスです。

技術支援や、1on1 での定期フォローなど、参画後も高いパフォーマンスを維持し続けられる体制を提供しています。

そのほかの事業についてや、事業のより詳細な内容については
こちらの事業紹介ページをご覧ください。

Sreake のエンジニアによる技術支援

技術戦略から設計、構築、運用までワンストップ支援する　技術支援サービス

- Multi Cloud や Cloud Native な先進的技術及び大規模なサービス運用に強みを持つエンジニアによる技術支援
- ベンダー的な役割ではなく「お客様の チームメンバー」という立ち位置で最新技術の提案から運用支援までをトータルご支援

Sreake SRE 導入 / 定着化支援

Enterprise を中心に様々な業種/業界でSRE を実践してきたナレッジを活かしたSRE チームの導入及び定着化を行います

- SRE の組成/SRE 文化の定着化の為の支援
- お客様の組織に適合した SRE の導入を支援

本日お話する内容

3-SHAKE

話すこと

SRE についてさくっとしたお話

私達が展開している SRE の実践方法について

話さないこと

SRE の概念に関する詳しいお話

詳細なシステムアーキテクチャなど

SRE の基本について知りたい方は以下がおすすめです

SREとはなにか [サイト リライアビリティ エンジニアリング]

- **sre.google に掲載されているSRE Book**
  https://sre.google/sre-book/table-of-contents/
  https://sre.google/workbook/table-of-contents/

- **SRE をわかりやすく理解出来るコンテンツ**
  Google Cloud Days SRE の基本と組織への導入 ～ サービスレベル目標やエラー予算などサービスの信頼性に対する考え方～
  https://cloudonair.withgoogle.com/events/google-cloud-day-digital-21?talk=d3-infra-01

  3-shake SRE Tech Talk #2 (Google Cloud 篠原様による登壇)
  https://youtu.be/g-WMeetjoRM?t=218

  SREとはなにか[サイト リライアビリティ エンジニアリング]
  https://sreake.com/blog/what-is-sre/

2. SRE とは？

SRE とは ~ SRE 本の序章より ~

3-SHAKE

What exactly is Site Reliability Engineering, as it has come to be defined at Google?
My explanation is simple:

SRE is what happens when you ask a software engineer to design an operations team.

By Betsy Beyer

Googleで定義されるようになった Site Reliability Engineeringとは一体何なのか？
私の説明は簡単です。
SRE とは、ソフトウェアエンジニアに運用チームの設計を依頼したときにできあがるものです 。

https://sre.google/sre-book/introduction/

説明が簡単でも...理解が難しい..

3-SHAKE

Ops の現場で抱えている問題

煩雑で繰り返しの多い運用作業

日々追われる障害対応とメンバーへの過負荷

守りを重視するが故にリリーススピードが遅い

3-SHAKE

Principles of SRE

- リスクの受容
- SLO の定義
- 分散システムのモニタリング
- Toil の削減
- 自動化の推進
- 適切なリリースエンジニアリング

参照: https://sre.google/sre-book/part-II-principles/

SRE の原則について

3-SHAKE

Principles of SRE

- **リスクの受容**
- **SLO の定義**
- 分散システムのモニタリング
- Toil の削減
- 自動化の推進
- 適切なリリースエンジニアリング

100% の可用性を目指さず、SLO を元に適切なリスクマネジメントと業務ハンドリングを行う。

そのためにリスクの評価、管理、およびエラーバジェットの使用などを実施していく。

参照: https://sre.google/sre-book/part-II-principles/

Principles of SRE

- リスクの受容
- SLO の定義
- **分散システムのモニタリング**
- Toil の削減
- 自動化の推進
- 適切なリリースエンジニアリング

**長期的なトレンドの把握や適切なアラートによる問題解決の修復等を行うために、
各種のモニタリングやアラート設定を行っていく。**

参照: https://sre.google/sre-book/part-II-principles/

Principles of SRE

- リスクの受容
- SLO の定義
- 分散システムのモニタリング
- **Toil の削減**
- **自動化の推進**
- 適切なリリースエンジニアリング

サービスの成長に比例して拡大する永続的な価値を提供しない、ありふれた反復的な運用作業を自動化して削減していく。

参照: https://sre.google/sre-book/part-II-principles/

Principles of SRE

- リスクの受容
- SLO の定義
- 分散システムのモニタリング
- Toil の削減
- 自動化の推進
- **適切なリリースエンジニアリング**

多くの障害は人の手が加わることによって発生する。

その為、適切な構成管理やリリースエンジニアリングの仕組みを構築する必要がある。

参照: https://sre.google/sre-book/part-II-principles/

SRE の原則について

3-SHAKE

Principles of SRE

- リスクの受容
- SLO の定義

これらの原則を Software Engineering を通じて実現させる運用を行う職務こそが SRE

※ 独自解釈

- シンプルさの追求

参照: https://sre.google/sre-book/part-II-principles/

3. Sreake 流 SRE 実践

Sreake 流 SRE Roadmap

組織組成するとき/組織改革するときに失敗するパターン

SRE チームに限らずあらゆる組織を立ち上げる時は様々な苦労や困難な壁にぶつかります。

重要なのは**組織組成や改革は当事者や周りが冷めてしまったら終わり**だということです。

その為に生まれたての組織は尚更慎重に活動していく必要があります。

- 組織の役割が曖昧でメンバーに目的が浸透していない
- 活動が他のチームや上層部から理解が得られない
- 現実と折り合いを付けずに到達困難な高すぎる目標を立ててしまう
- 新しい活動に割けるリソースがなく、コミット量が少ない

class SRE implements DevOps

3-SHAKE

class SRE は DevOps を実装するとありますが ...

※SRE には、必ずしもDevOps インターフェースの一部ではない、追加のプラクティスや推奨事項が含まれている場合があります。

DevOps	SRE
Reduce organization silos 組織のサイロ化を無くす	Share ownership with developers by using the same tools and techniques across the stack 同じツールやテクニックを使用して、開発者とオーナーシップを共有する
Accept failure as normal 失敗を当たり前のように受け入れる	Have a formula for balancing accidents and failures against new releases 事故や失敗と新製品とのバランスをとるための公式を持つ
Implement gradual change 徐々に変化させる	Encourage moving quickly by reducing costs of failure 失敗したときのコストを減らすことで、迅速な行動を促します。
Leverage tooling & automation ツールと自動化の活用	Encourages "automating this year's job away" and minimizing manual systems work to focus on efforts that bring long-term value to the system システムに長期的な価値をもたらす取り組みに集中するために、「今年の仕事を自動化する」ことを奨励し、手作業によるシステム作業を最小限に抑える
Measure everything すべてを測定する	Believes that operations is a software problem, and defines prescriptive ways for measuring availability, uptime, outages, toil, etc. オペレーションはソフトウェアの問題であると考え、アベイラビリティー、アップタイム、アウテージ、トイルなどの測定方法を規定しています。

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

SRE はあらゆるレイヤーに密接に関わります

DevOps の「実装者」としての役割をもつ SRE は当然
開発 や これまでの運用といったものと密接に関わります。
また、SLO の定義や SLA の定義を行うためには当然 SRE のみで終止するものでは無いため、ビジネスサイドとも密接に関わります。

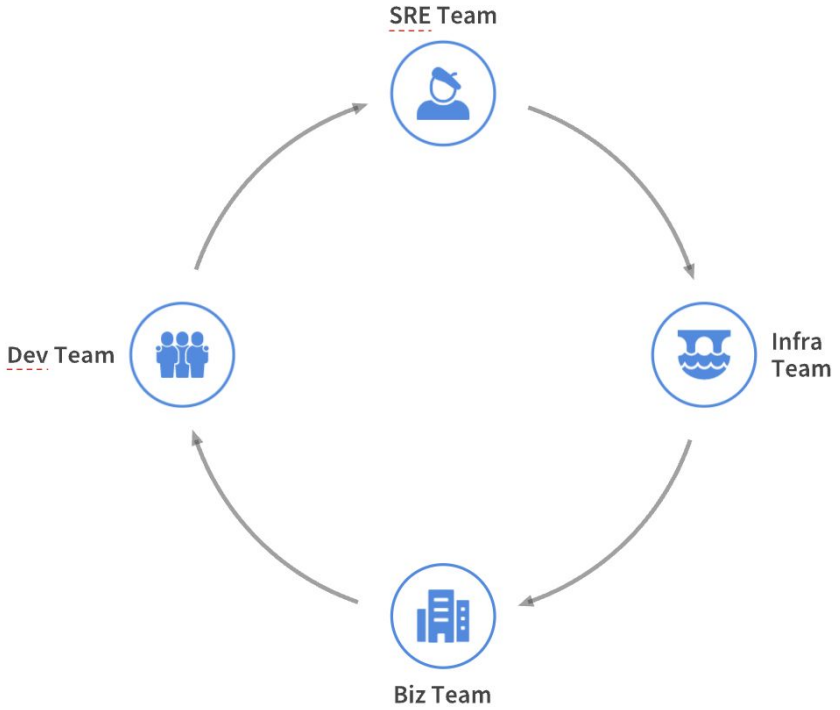
SRE は単一組織として自分たちのタスクに終始すれば良いものではなく、様々な組織や役割と密接に連携し合いながら実践していく必要があると考えています。

他チームとの協働の重要性

SRE チームの役割を明確にして理解を得よう

SRE は SRE チームだけで成立するものではありません。他チームとの協働があってこそ成立するものです。
そのためにも SRE の役割を明確にして有機的に協働する必要があります。

また、Google の SRE のカタチに囚われすぎない事も大事かもしれません。
今の組織やシステム構成にあわせた形で SRE の定義を行わなければ理想だけで上手く行かないと考えています。



Sreake 流 SRE Roadmap

SRE の実践内容 (例)

- 監視基盤導入
  - Logging
  - Monitoring
  - APM
- SLI / SLO の定義
- 運用体制整備
  - インシデント対応・管理
  - 効果的なアラート
- IaC (Infrastructure as Code)
  - 構成管理の品質チェック
  - GitOps
- CI/CD 導入
  - デプロイの自動化
  - コード品質・脆弱性の検査
  - DevSecOps
- パフォーマンス分析
  - 分散トレーシング
  - 負荷試験
  - カオスシナリオ試験

SRE の実践内容 (例)

- 監視基盤導入
  - Logging
  - Monitoring
  - APM
- SLI / SLO の定義

Cloudwatch

Cloud Operations

SRE の実践内容 (例)

- 運用体制整備
  - インシデント対応・管理
  - 効果的なアラート

運用体制構築

Slack 等の Chat

PagerDuty

SRE の実践内容 (例)

- IaC (Infrastructure as Code)
  - 構成管理の品質チェック
  - GitOps

SRE の実践内容 (例)

- CI/CD 導入
  - デプロイの自動化
  - コード品質・脆弱性の検査
  - DevSecOps

SRE の実践内容 (例)

- パフォーマンス分析
  - 分散トレーシング
  - 負荷試験
  - カオスシナリオ試験

SRE の実践内容 (例)

- 監視基盤導入
  - Logging
  - Monitoring
  - APM
- SLI / SLO の定義
- 運用体制整備
  - インシデント対応・管理
  - 効果的なアラート
- IaC (Infrastructure as Code)
  - 構成管理の品質チェック
  - GitOps
- CI/CD 導入
  - デプロイの自動化
  - コード品質・脆弱性の検査
  - DevSecOps
- パフォーマンス分析
  - 分散トレーシング
  - 負荷試験
  - カオスシナリオ試験

ひとつのサービスから始める

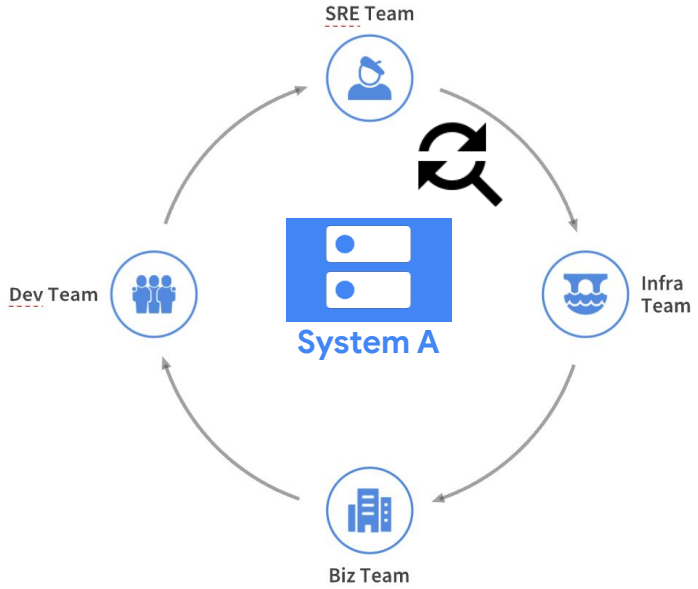
小さく始めてSRE チームとして成功体験を得よう

そのためには、多数のシステムを一気に対象としないこと。
まずはひとつのサービスを対象にSRE を始めて徐々に広げていきましょう。

なお且つ、出来れば新規サービスが望ましい。

理由

- ➢ 新しくSLI / SLO 設定する元となる情報を取得する為の基盤導入が難しい。
- ➢ 既存の運用がある為、新しい活動への導入が難しいケースがある
- ➢ そもそも手を加える前提で作られていない …



ひとつのサービスと
それに関わるチームからSRE の活動を始める

あらゆる情報を収集し始めよう

SRE を始めるにあたっては、元となるあらゆるデータが揃っていることが前提です。

結局は**データドリブンな世界**にする必要があり、データがない状態では SLI / SLO の設定はそもそも不可能です。

具体的には監視基盤の見直し等を通じてとにかくまずはデータを収集し、可視化出来るところまで持っていきましょう。

闇雲に戦う世界から

データを元に意思決定が出来る世界へ

Sreake 流 SRE Roadmap

SLI・SLO・SLA について

3-SHAKE

- **SLI (Service Level Indicator) サービスレベル指標**
  - 何をもとにシステムの良し悪しを判断するかの指標となるもの
  - SLO を定義する時に利用する
- **SLO (Service Level Objective) サービスレベル目標**
  - サービスレベルに対する内的な目標値
- **SLA (Service Level Agreement) サービスレベル保証**
  - サービスレベルに対する対外的な保証値

SLI・SLO・SLA について

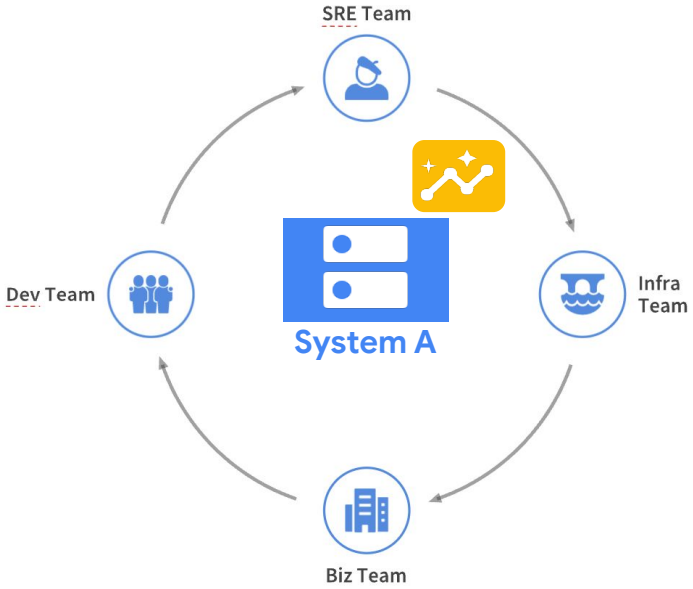
3-SHAKE

- **SLI (Service Level Indicator) サービスレベル指標**
  - 何をもとにシステムの良し悪しを判断するかの指標となるもの
  - SLO を定義する時に利用する

上質な SLO を設定するためにもまずは あらゆる指標となりうるあらゆるメトリクスを収集し、可視化ができるようにしよう

SLO の設定について

- CUJ (クリティカルユーザージャーニー)を検討していく事から始める
- ただ最初の SLO 設定はエイヤーで決めてしまうのも一つ。
  - SLO はサービスの成長に応じて変わっても良い。
  - ただし、SLO 100% はアンチパターンなので、そういう設計はしない事
- それより大事なのが、**Biz / Dev を置き去りにせず共通認識の獲得を意識しながら設定していくこと**
  - 定期的に振り返りながら組織一丸となって、一緒に高品質なプロダクトを目指していく事が大事。



ポストモーテムの実施

ポストモーテムとは、インシデント、その影響、インシデントを軽減または解決するために取られたアクション、根本原因、およびインシデントの再発を防止するためのフォローアップアクションを書面で記録することです。

- 振り返りから学ぶ文化を作る為にポストモーテムを行います。
- その為には障害解析に対して追跡が出来るようにしておかなければなりません。
- フォーマットについてはまずは SRE 本にて提示されている Example を流用するのがおすすめ。
- それと何よりも大切なのが **特定個人を非難しない文化** であることです。
  ※ あくまで、システムや仕組みの問題であると考えます。

参考: https://sre.google/sre-book/postmortem-culture/
https://sre.google/sre-book/example-postmortem/

PagerDuty の組み合わせで SRE を加速させる

オンコール機能
- ❖ 適切なオンコール体制を作る為のスケジューリング機能

実践的な Alerting
- ❖ EventRule を始めとしたインテリジェンスなアラート制御

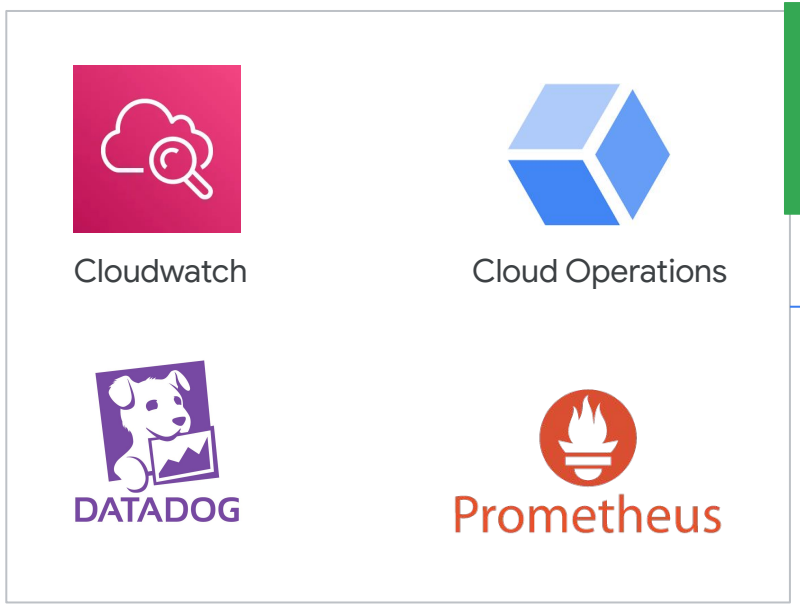
インシデント管理
- ❖ 様々なインテグレーションに対応しているので、包括的にインシデント管理が可能

障害発生時の対応の追跡
- ❖ ユーザーのアクションを自動で記録

ポストモーテム
- ❖ 組み込みでポストモーテムの機能があり、失敗から学ぶ文化をすぐに作れる

PagerDuty

Alerting & Incident Management



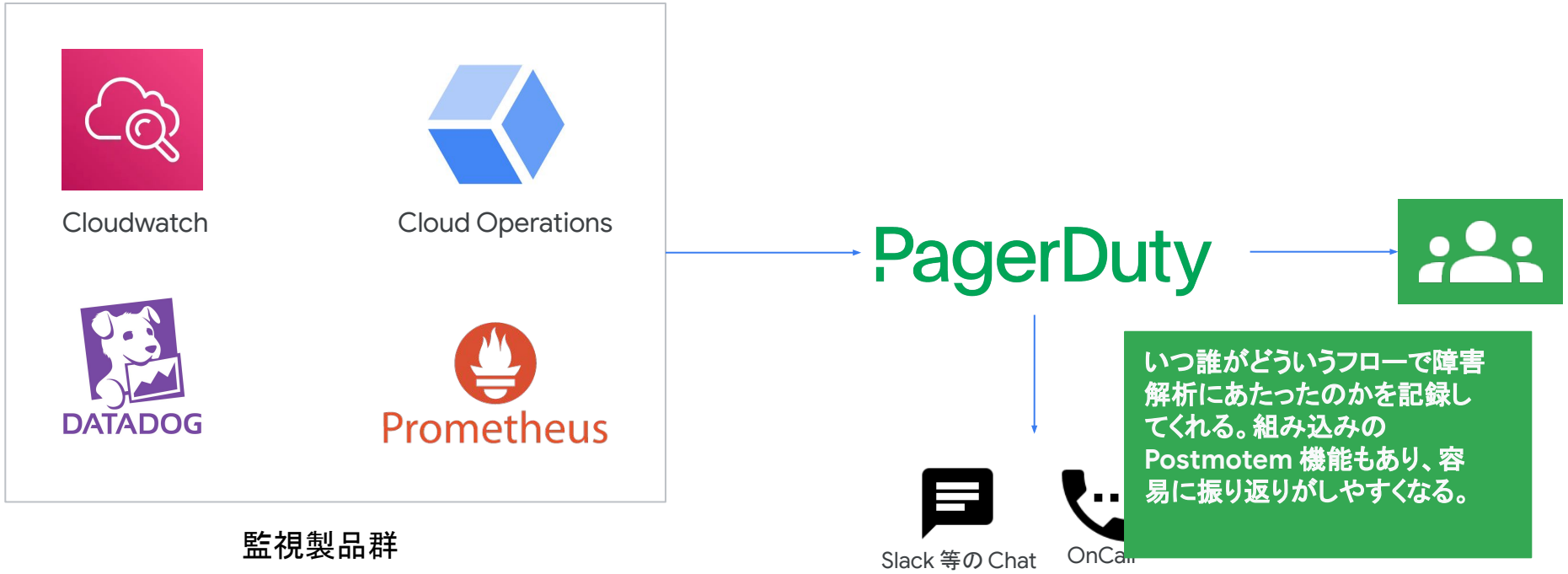
監視製品群

各監視製品群と連携機能が備わっているため、容易に設定が可能

PagerDuty

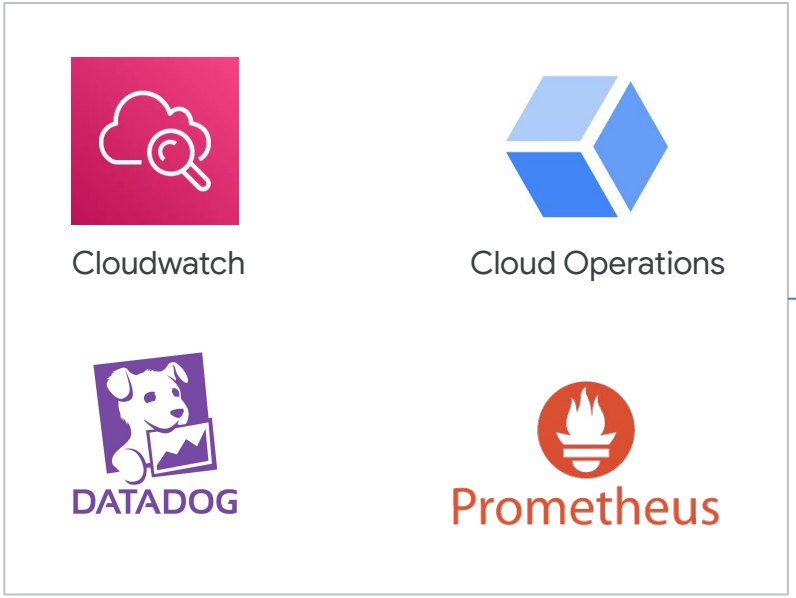
Alerting & Incident Management

Alerting & Incident Management



Alerting & Incident Management

3-SHAKE



監視製品群

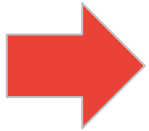
Jira 等のチケット管理ツールともインシデント連携可能。スクラムや SRE の活動との連携もしやすいため、Toil の測定やエラー予算の管理にも役立つ

Toil の削減

3-SHAKE

Toil とは、手作業、繰り返される、自動化が可能、戦術的、長期的な価値がない、サービスの成長に比例して増加する、といった特徴を持つ作業です。

- Dev Team からの依頼対応
- 障害対応
- 割り込みタスク
- SRE としての定型業務
- 定期的に発生する
  メンテナンス作業



まずはこれらを
すべて測定することから

どういう作業がどの程度の頻度発生し、どれだけの工数をかけてどういう効果が得られたのかを取得する

参考: https://sre.google/sre-book/eliminating-toil/

Toil 削減をするために

測定、振り返り、自動化のサイクルを回していく

測定

Jira 等のプロジェクト管理ツールを駆使して定形業務や割り込みタスクを測定する。

振り返り

SRE チーム内外含めて定期的な振り返りを通じてToil 特定のを行う

自動化

振り返りによって特定されたToil を自動化によって削減していく

Sreake 流 SRE Roadmap

SREの発展と継続

3-SHAKE

- SRE の文化を浸透させていく
  - ひとつの成功体験を得られたら他のサービスにも展開していく
  - 定期的な振り返りを通じてさらなる改善を行っていく
  - 計測したモノからToil 削減の為の自動化の促進を進めていこう

- Error Budget を元に業務をコントロールする
  - そのためには十分な体制やSLI・SLO の定義が必要
  - また、サービスのビジネス的な意味合いによるところもあるので、現実的にはこの運用は相当難しい...

4. まとめ

まとめ

- Google の SRE は素晴らしい見本になる。でも、SRE はウェットな部分も多いのが現実。　自分たちの組織やメンバー構成を考えながら組織にあった最適な　SRE を
- SRE を実現するにあたって「協働」は不可欠。他のチームと有機的に協働しながら　SRE を実現していこう
- SRE を始める時はなるべく小さく、手を付けやすいところから始めよう

Sreake サービスのご紹介

3-SHAKE

Sreake

Sreake（スリーク）は、金融・医療・動画配信・AI・ゲームなど、
技術力が求められる領域で豊富な経験を持つSREの専門家が集まったチームです。
戦略策定から設計・構築・運用、SaaS提供まで、幅広い領域をサポートします。

SRE導入/定着化支援

SRE導入のためにはシステム・アーキテクチャのみならず、組織全体としての取り組みが必要になります。
大手からベンチャーまで様々な組織規模でのSRE導入実績が豊富なSreakeのSREによるSRE組織の組成/SRE文化の定着化の為の支援を行います。

技術支援

Kubernetesを始めとしたCloudNativeな技術領域に強みを持ったエンジニアによる技術支援を行っております。

設計/構築/運用など様々なフェーズで知見が豊富なエンジニアによる技術支援を行うことが可能です。

サービス販売/導入支援

SREを最短距離で実現する為の製品導入支援

- Google Cloud
- Anthos
- PagerDuty
- RUNDECK
- PrismaCloud
- DataDog
- MetricFire
- Looker
- etc...

サービス詳細や料金についてのご質問・ご相談など
お気軽にお問い合わせください

詳細・お問い合わせ は
こちら

Thank you