STARTUP DAY

[AWS Start-up ゼミ]

よくある課題を一気に解説！

～御社の技術レベルがアップする
　2018 春期講習～

Akihiro Tsukada, Senior Solutions Architect | 2018.03.12

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

AWSは日々多くのスタートアップを技術的に(も)支援

SmartNews
NEWS PICKS
Gunosy
Voicy
Piece of Cake, inc.
Speee
ペライチ
translimit
Money Forward
zaim
WEALTHNAVI
THEO
FINATEXT
One Tap BUY
metaps
freee
AnyPay
ORIGAMI
Kyash
BASE
Coiney
FOLIO
coincheck
MISOCA
scouty
ABEJA
SKYDISC
SORACOM
Moff
CrowdWorks
Lancers
enechange
SmartHR
WANTEDLY
BIZREACH
Green
Eight
RakSul
PIXTA
iRidge
YAMAP
Farmnote
akippa
ecbo cloak
CONCORE'S
JapanTaxi
SmartDrive
AXELSPACE
Talknote
ONETEAM
chatwork
TimeTree
mamari
MedPeer
MEDLEY
弁護士ドットコム
mercari
Timers
STORES
FRIL
ジモティー
Creema
CHANNEL
CAULIS
FiNC
VASILY
nana
SORA
TORETA
EmotionTech
EventRegist
MAMORIO
cookpad
ごちクル
Retty
dely
every.tv
Qrio
BONX
safie
KAIZEN PLATFORM
USERDIVE
from scratch
PLAID
IntimateMerger
FULLER
Repro
ietty
Smart Education
アオイゼミ
mikan
Studyplus
Progate
Qiita
nulab
RoomClip

[AWS Start-up ゼミ] よくある課題シリーズについて

- AWSスタートアップチームは、日々うけている技術相談から
  スタートアップにおける**技術的な課題と傾向**を把握
- このスライドは2018年春時点での
  「あるある」まとめ
- 詳細な解説を全てこの資料で行うのでなく、
  「こうしたい」に対して
  「どう考えてどの資料を見ればよいか」
  を示す逆引き辞典

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

参考: 2017 夏期講習

https://www.slideshare.net/AmazonWebServicesJapan/aws-startup-2017

参考: 2017 夏期講習

アジェンダ

よくある課題編

1. ユーザ動向を分析したい
2. CI/CDをちゃんとしたい
3. コンテナを使いたい
4. 運用監視ちゃんとしたい
5. システム負荷下げたい
6. （モバイルアプリの）Growth Hackしたい
7. コスト下げたい

お役立ち情報編

1. おさえておきたい技術トレンド
2. 要るの？要らないの？セキュリティ

#awsstartup amazon web services

https://www.slideshare.net/AmazonWebServicesJapan/aws-startup-2017

本日 2018 春期講習のアジェンダはこちら

1. WebSocket、チャット、リアルタイムな何かをしたい
2. 定時バッチ処理うまいことやりたい
3. 一斉プッシュ通知送信後にサーバ負荷上がる問題
4. セキュリティどこまでやればいいの

おまけ：AWSの技術情報を効率的にキャッチアップしたい

WebSocket、チャット、リアルタイムな何かをしたい

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

WebSocket、チャット、リアルタイムな何かをしたい

- よくいただく課題
  - 「リアルタイムに更新されるUIを作りたい」
  - 「チャット機能作りたい」
- 本当にしたいことは何？
  - リアルタイム性を高めて UX 向上したい
  - お金と運用のコスト抑えながらスケールするバックエンド作りたい
    （"リアルタイム"な機能は一般に複雑で高コストになりがち　　）
- 思考フロー
  1. マネージドサービス（AWS AppSync）でできるかな？
  2. ALBとWebSocketサーバならどうだろう
  3. それでもだめなら個別にWebSocketサーバ立てるか

WebSocket、チャット、リアルタイムな何かをしたい

1. マネージドサービス（AWS AppSync）でできるかな？

MQTT over WebSocket

```
type Todo {
    id: ID!
    name: String
    description: String
    priority: Int
    duedate: String
}
```

AWS AppSync

Amazon DynamoDB

AWS Lambda

Amazon Elasticsearch Service

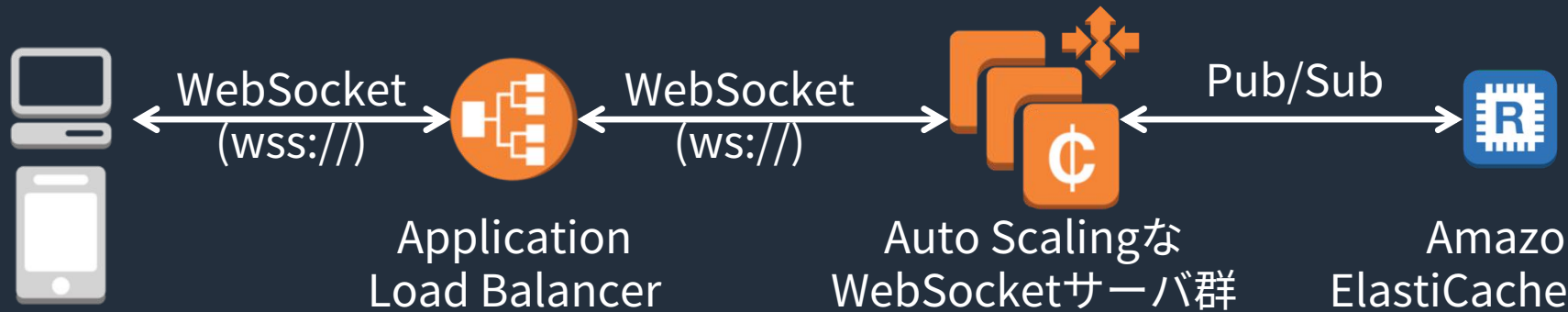
クライアントはAppSync SDKを通じて
データを読み書きするだけ
更新があれば通知される
オフラインやコンフリクトのサポートも

バックエンドは柔軟に選択可能
AppSyncがよしなにテーブルを作ったり
クエリしたりしてくれる

※2018.3.12現在、AWS AppSyncはパブリックベータです

WebSocket、チャット、リアルタイムな何かをしたい

2. ALBとWebSocketサーバならどうだろう



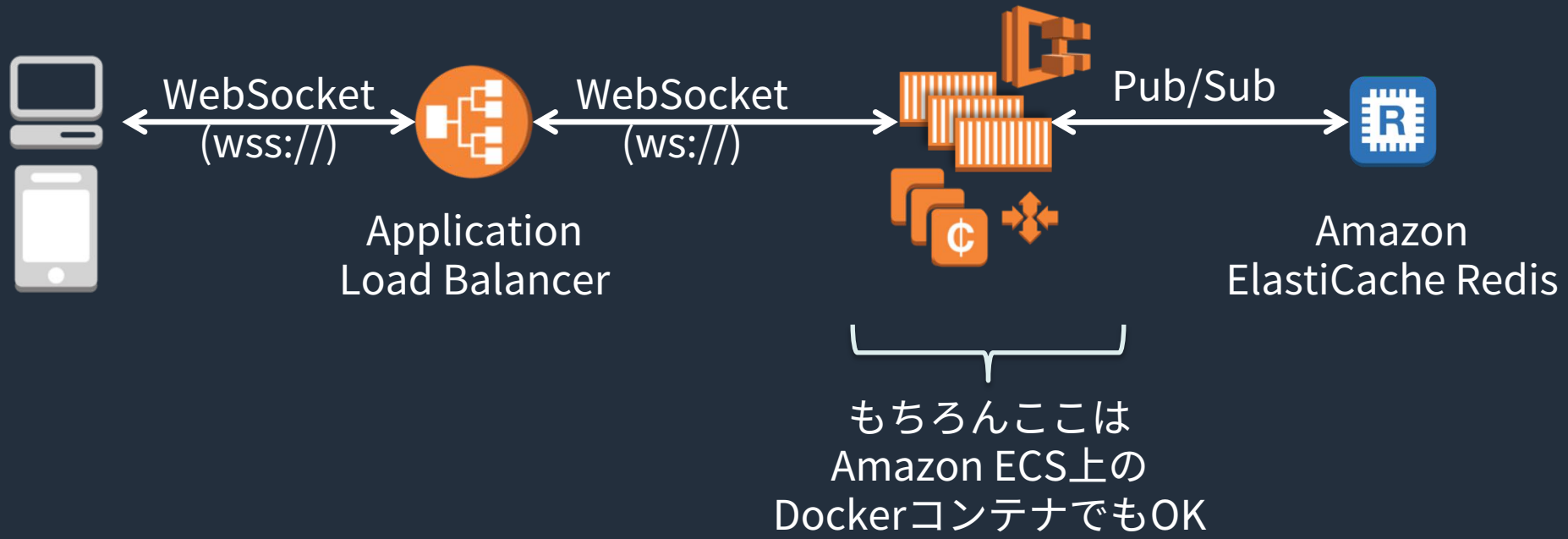
クライアントはシンプルに
WebSocketでALBに接続
切断時の再接続や
オフライン時のケアは
要実装

コネクション確立時に
ALBが分散してくれる
TLSの終端もALBで

ステートレスを保ち、
どのインスタンスに
どのクライアントが
接続してもいいように
チャンネルはRedisで管理

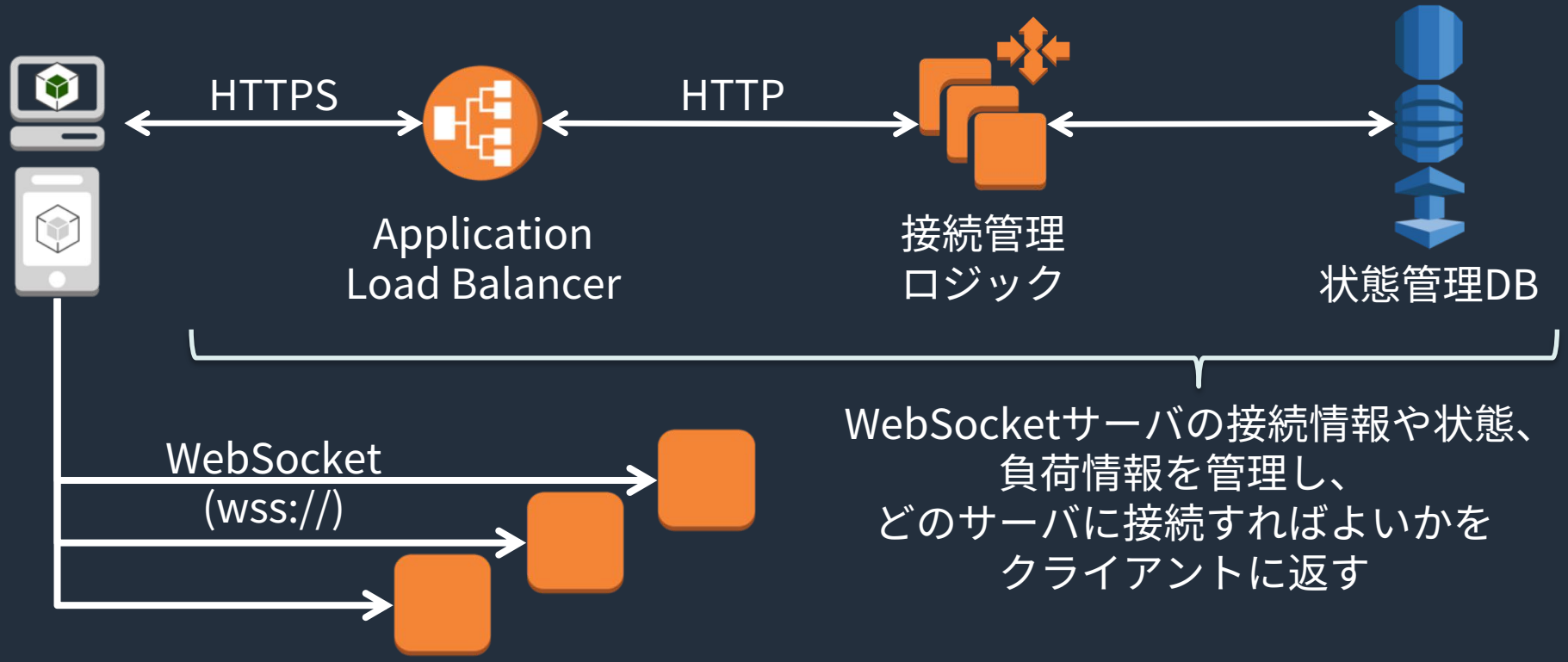
WebSocket、チャット、リアルタイムな何かをしたい

2. ALBとWebSocketサーバならどうだろう



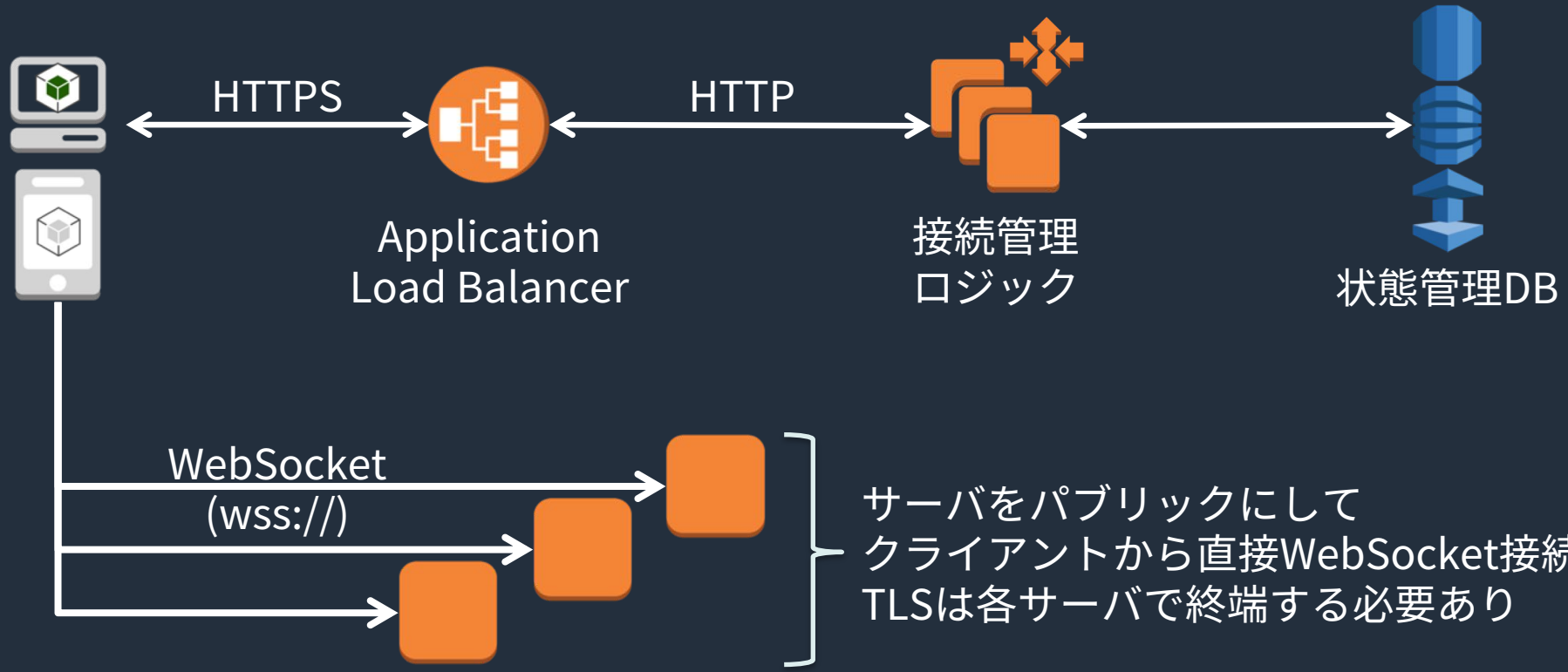
WebSocket、チャット、リアルタイムな何かをしたい

3. それでもダメなら（本当にダメか？）個別にWebSocketサーバ立てるか



WebSocket、チャット、リアルタイムな何かをしたい

3. それでもダメなら（本当にダメか？）個別にWebSocketサーバ立てるか



WebSocket、リアルタイムな何か – 実装時の参考資料

- AWS Black Belt Online Seminar資料＆動画
  - Introduction to AWS AppSync and GraphQL - AWS Online Tech Talks（資料|動画）
  - Elastic Load Balancing（資料|動画）
  - WebSocket on AWS（注意: 2012年の資料。思考フロー3の参考に）（資料）
- AWS Summit/Dev Day講演資料 （2016 | 2017）
  - [CyberZ] OPENREC.tvにおけるライブ動画およびメッセージ配信基盤のリプレース全貌 (2017)（資料 | 動画）
- 事例
  - AWS 導入事例：株式会社CyberZ
  - WebSocket対応した噂のALB (Application Load Balancer) を試してみた

定時バッチ処理
うまいことやりたい

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

定時バッチ処理うまいことやりたい

- よくいただく課題
  - 「今は管理用サーバでcron動かしてるんですが…」
  - 「Lambdaが1時間くらい連続で使えればいいのに」
- 本当にしたいことは何？
  - 安定したスケジューラが欲しい
  - 複雑な処理をシンプルに、大量な処理をスケーラブルにやりたい
- 思考フロー
  1. とりあえずcronやめてマネージドなスケジューラを活用しよう
  2. ちょっとした処理ならAWS Lambdaで
  3. 難しければ AWS Step Functions や Amazon ECS、AWS Batch で

定時バッチ処理うまいことやりたい

1. とりあえずcronやめてマネージドなスケジューラを活用しよう

cronの運用は割と大変

cronサーバが一台だと
SPOFになっちゃうし…

冗長化しようとして
二台にすると処理が二重に
走っちゃうし、
内容更新したときの
同期もしないと…

AutoScalingなクラスタ
とも切り離す必要があって
コスト効率悪し

定時バッチ処理うまいことやりたい

1. とりあえずcronやめてマネージドなスケジューラを活用しよう

AWSが提供するマネージドなスケジューラの例

CloudWatch Events

AWS Lambda

AWS Batch Job Queue

Amazon ECS Run Task

Amazon SNS

AWS Step Functions Workflow

AWS Systems Manager

・・・

柔軟なイベントスケジューラ

AWS Glue

ETL処理のためにデータを取得するcrawlerの起動タイミングをスケジューリングできる

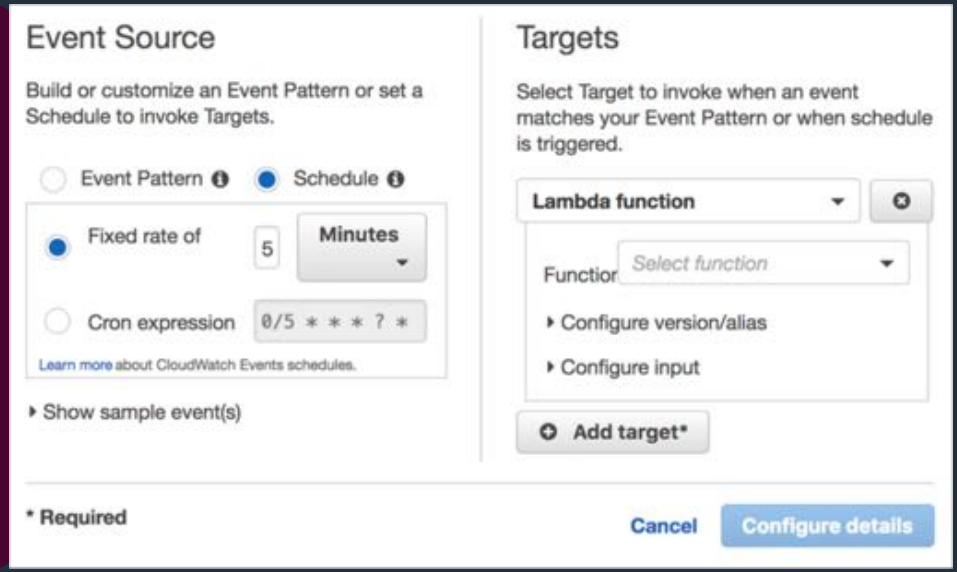
AWS Elastic Beanstalk Worker Tier

cron.yaml でのジョブ定義

定時バッチ処理うまいことやりたい

1. とりあえずcronやめてマネージドなスケジューラを活用しよう

AWSが提供するマネージドなスケジューラの例



柔軟なイベントスケジューラ

直感的な設定画面

定時バッチ処理うまいことやりたい

2. ちょっとした処理ならAWS Lambdaで

AWS Lambda

- シンプルにスクリプトを登録してスケジュールを設定するだけ
  - Node.js, Java, Python, Go, C#
- IAM Roleによる実行時の権限設定やAWS KMSによる環境変数の暗号化など便利な機能を提供
- 実行時の制限事項（メモリ容量、/tmp容量、最大実行時間 etc）に注意
- 制限事項に引っかかる大きな処理や、1ファンクションで表現しづらい複雑なものはStep FunctionsやECS, Batchの利用を検討

定時バッチ処理うまいことやりたい

3. 難しければ AWS Step Functions や Amazon ECS、AWS Batch で

AWS Step Functions - Workflow
Lambdaファンクションを始めとした複数の処理から成る
複雑なワークフローを定義し、実行スケジュールを設定できる

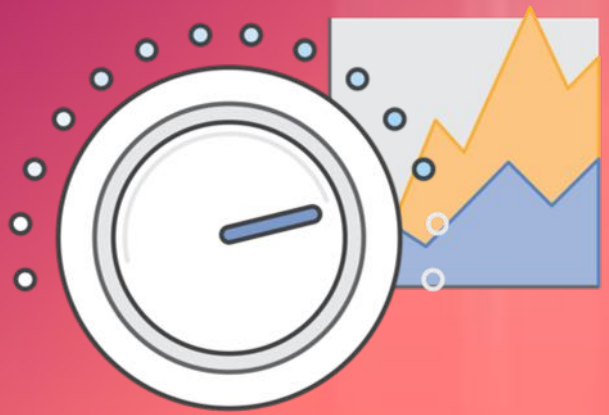
Amazon ECS - Run Task
任意のDocker コンテナを任意のコマンドで実行する Run Task をスケジューリングできる。Fargateを使えばEC2インスタンスの管理も不要に

AWS Batch - Job Queue
エンキューされたJobに応じて事前に定義されたコンピュート環境で大量のコンテナを起動し分散処理が可能。Array Jobなども実行でき、Spot Fleetとの統合もされている

定時バッチ処理うまくやる – 実装時の参考資料

- AWS Black Belt Online Seminar資料＆動画
  - Amazon CloudWatch（資料|動画）
  - AWS Step Funcations （資料|動画）
  - AWS Batch（資料|動画）
- AWS Summit/Dev Day講演資料 （2016 | 2017）
  - AWS Lambdaで変わる バッチの世界 〜CPUトータル100時間を10分で終わらせるには〜 (2017)（資料|動画）
  - AWS が支える Eight のリコメンデーションエンジンの裏側 (2017) （資料|動画）
- ドキュメント
  - AWS Lambda の制限
- コネヒト様 事例
  - Amazon ECS ScheduleTaskで実現するスマートなDockerベースのバッチ実行環境

一斉プッシュ通知送信後にサーバの負荷上がる問題



© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

一斉プッシュ通知送信後にサーバの負荷上がる問題

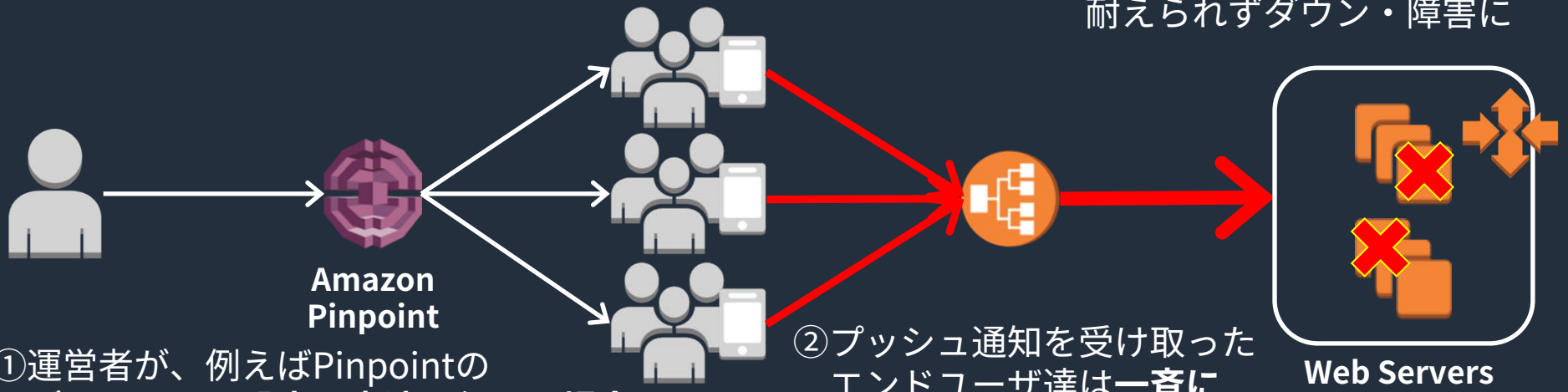
- よくいただく課題
  - 「プッシュ通知打つとアプリが一斉に起動してサーバ負荷が…」
  - 「プッシュの配信が速すぎて困るんでトピックをいくつかに分けて送ってるんです」
- 本当にしたいことは何？
  - スパイクアクセスに耐えるスケーラブルなバックエンドが欲しい
  - または、通知を適切な速さで届けたい
- 思考フロー
  1. そもそも速報性が不要なら Amazon Pinpoint で速度調整
  2. キャッシュが効くコンテンツなら Amazon CloudFront でさばこう
  3. それも難しければ普通に各所を負荷対策しよう

一斉プッシュ通知送信後にサーバの負荷上がる問題

1. そもそも速報性が不要なら Amazon Pinpoint で速度調整

速度調整なしの場合

③バックエンドサーバが
Auto Scalingだけでは負荷に
耐えられずダウン・障害に



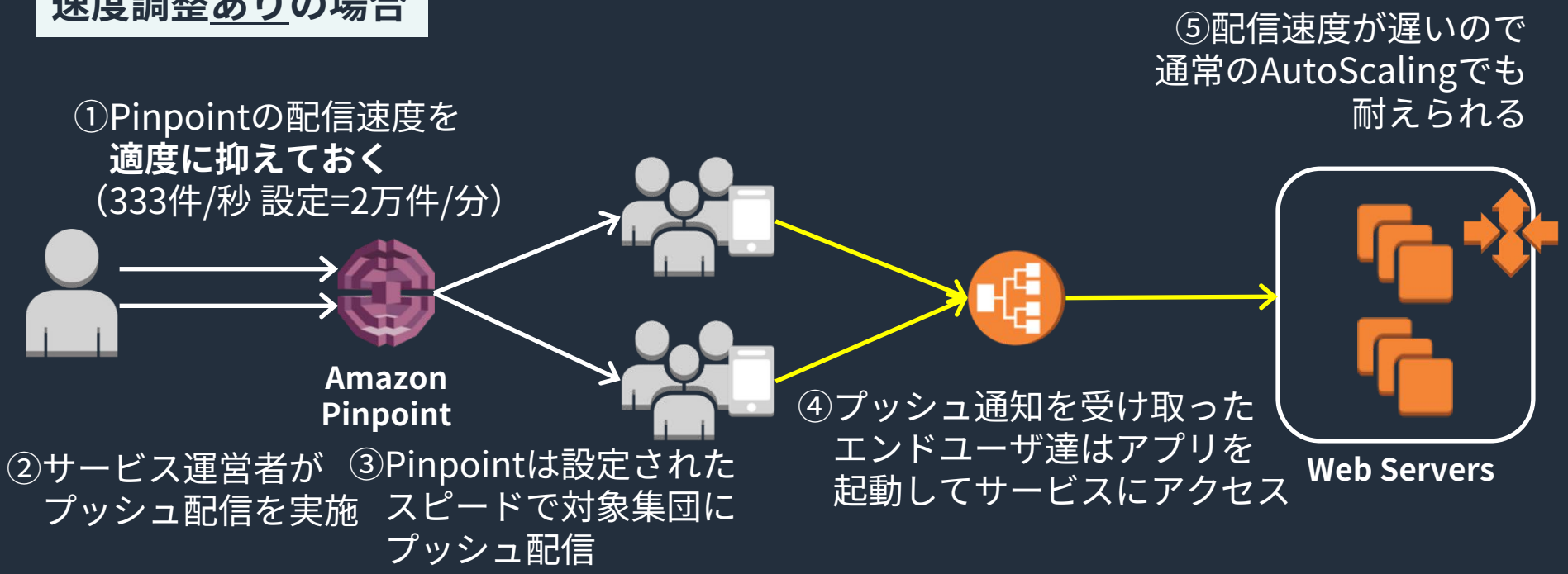
①運営者が、例えばPinpointの
デフォルトの設定で高速配信した場合は…
（16,666件/秒 設定=100万件/分）

②プッシュ通知を受け取った
エンドユーザ達は**一斉に**
アプリを起動してサービスに
アクセス、サーバは高負荷に

一斉プッシュ通知送信後にサーバの負荷上がる問題

1. そもそも速報性が不要なら Amazon Pinpoint で速度調整

速度調整ありの場合

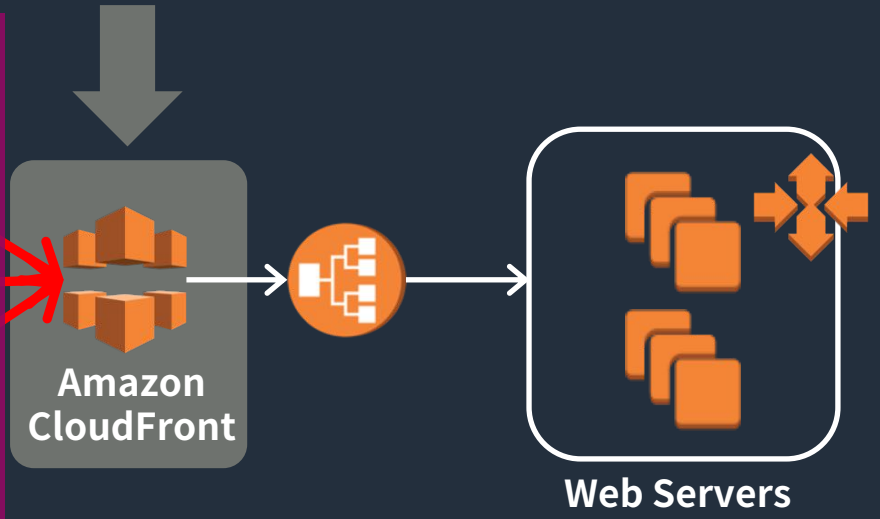


一斉プッシュ通知送信後にサーバの負荷上がる問題

2. キャッシュが効くコンテンツなら Amazon CloudFront でさばこう

CloudFrontありの場合

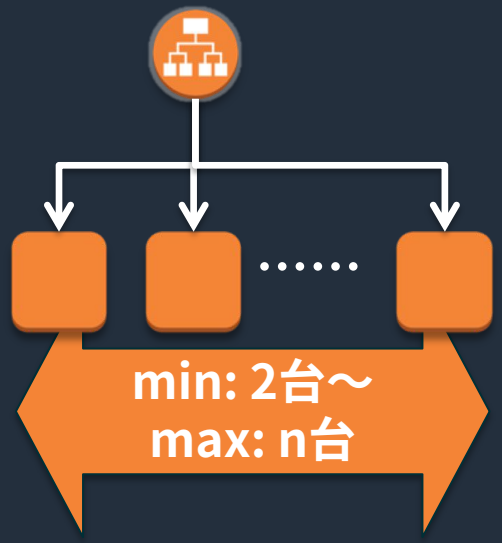
- CloudFrontならデフォルトでディストリビューションあたり100,000rps、40 Gbps と大容量配信
- 「内容的にキャッシュできないから…」施策の設計次第で実現できないでしょうか？
- S3でLanding Pageをホストとか無理でしょうか？



一斉プッシュ通知送信後にサーバの負荷上がる問題

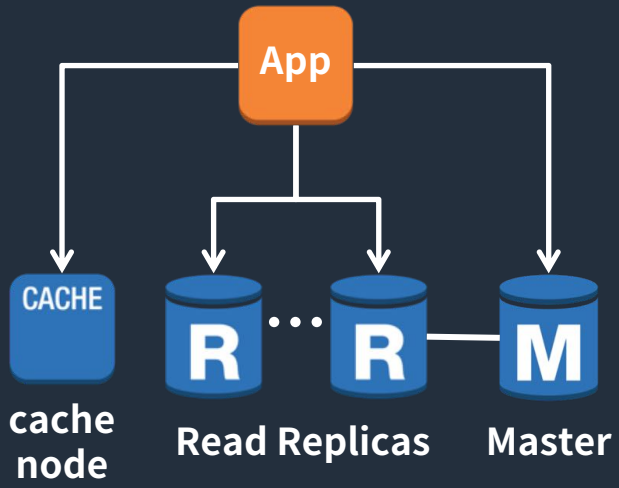
3. それも難しければ普通に各所を負荷対策しよう

Scale Out
（水平スケール）



Scale Up
（垂直スケール）

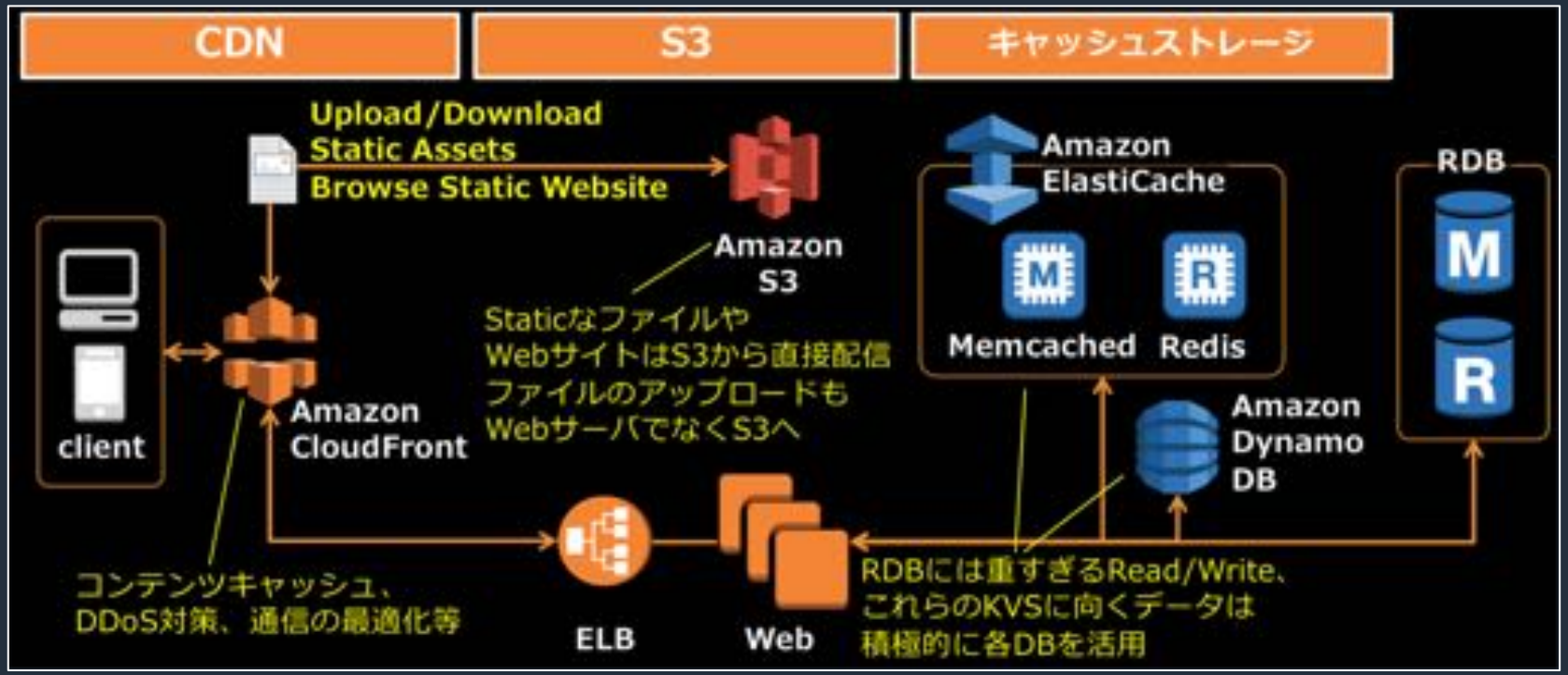
レプリケーション
＆キャッシュ



※2017 夏期講習「システム負荷下げたい」も参照のこと

一斉プッシュ通知送信後にサーバの負荷上がる問題

3. それも難しければ普通に各所を負荷対策しよう



※2017 夏期講習「システム負荷下げたい」も参照のこと

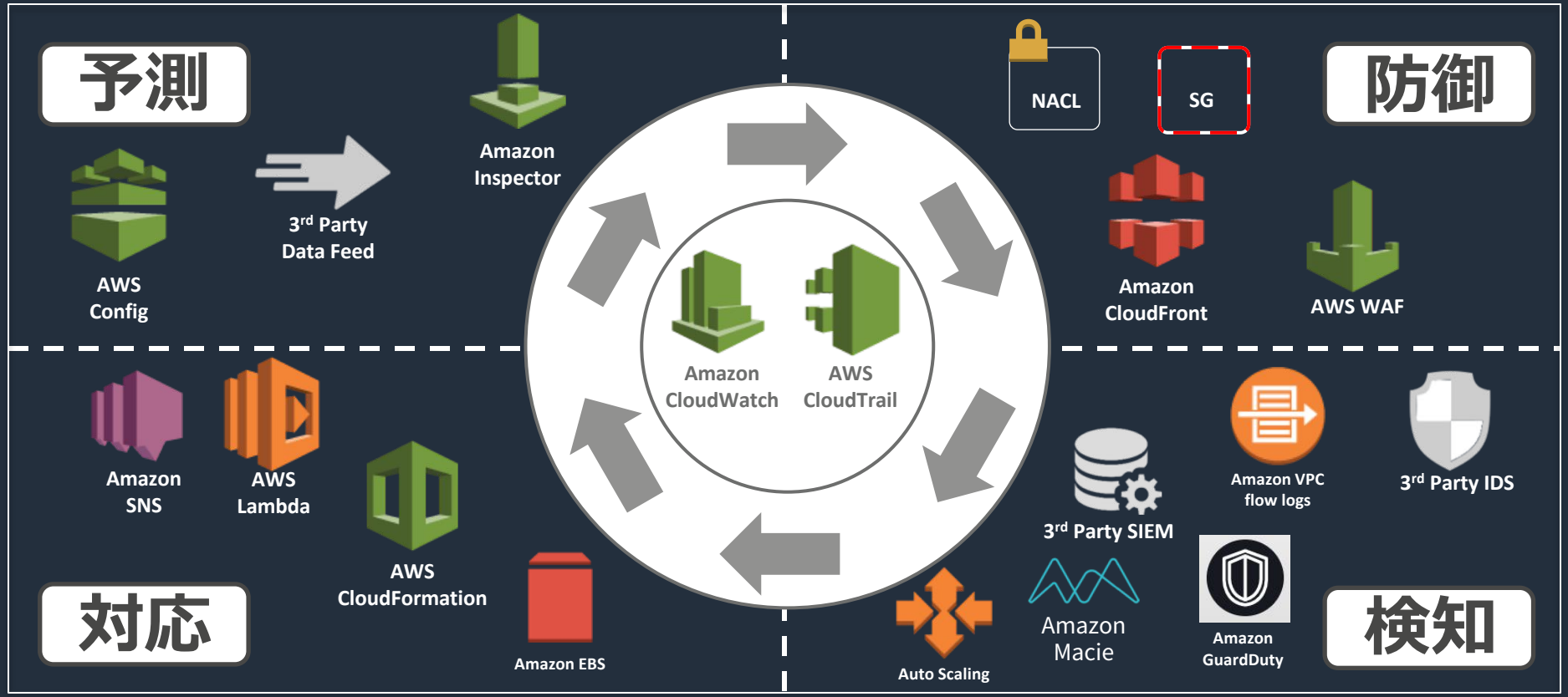
プッシュ通知後のサーバ負荷問題 – 実装時の参考資料

- AWS Black Belt Online Seminar資料＆動画
  - Amazon Pinpoint で始めるモバイルアプリのグロースハック(資料|動画)
  - Auto Scaling（資料|動画）
  - [Amazon RDS] Aurora （資料|動画）
  - Amazon CloudFront （資料|動画）
  - Amazon S3 （資料|動画）
  - 他関連資料多数 → https://aws.amazon.com/jp/aws-jp-introduction/
- [AWS Start-up ゼミ] 2017 夏期講習
  - システム負荷下げたい https://www.slideshare.net/AmazonWebServicesJapan/aws-startup-2017/40

セキュリティ
どこまでやればいいの

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

前提: ざっくり「セキュリティ」で考えること

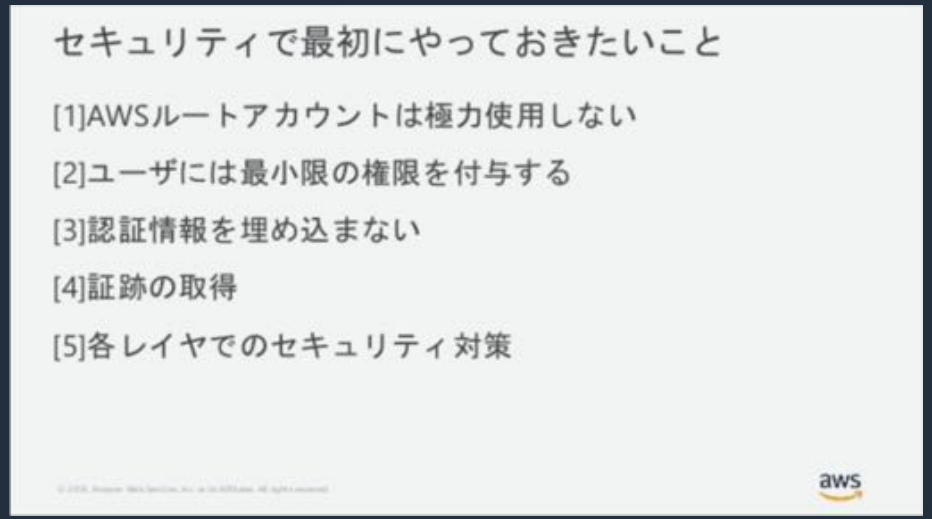


セキュリティどこまでやればいいの

- よくいただく課題
  - 「セキュリティってどうしても優先度下がりがちで…」
  - 「何からやればいいのか…」「やるにしてもキリがなくて…」
- 本当にしたいことは何？
  - 適切な労力で適切なセキュリティを得たい！
    （けど何が適切かわからない！）
- 思考フロー
  1. まず最低限のことはやろう
  2. Well-Architected Frameworkでリスクを洗い出して評価しよう
     （CAPDのCheck-Act）
  3. 従うべき法規制やガイドラインがあればそれを遵守しよう

セキュリティどこまでやればいいの

1. まず最低限のことはやろう



https://www.slideshare.net/AmazonWebServicesJapan/day-1-with-amazon-web-services-aws10/6

それで足りなくなったら…

セキュリティどこまでやればいいの

2. Well-Architected Frameworkでリスクを洗い出して評価しよう

AWS Well-Architectedとは？

- AWSが作成した**クラウド設計、構築、運用のベストプラクティス集**
- クラウドでの設計原則と**セキュリティ、信頼性、パフォーマンス効率、コストの最適化、運用性**について質問形式で記載

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

セキュリティどこまでやればいいの

2. Well-Architected Frameworkでリスクを洗い出して評価しよう

1) たとえばW-Aチェックの結果、以下の項目が要改善だったとする
   - ルートアカウントの保護
   - 保存されたデータの暗号化
   - 証跡の取得と管理

まずバックログとして
整理できることが大きな価値

2) それらをバックログに加え、重大度、緊急度、対応工数等で評価する
   - 「これが守られていないことで何が起こるか」
     「起こったときの事業インパクトはどの程度か」
     「どの程度の発生確率か」… etc

3) 他の機能追加タスク等と同様にイテレーションに取り込んでいく

セキュリティどこまでやればいいの

2. Well-Architected Frameworkでリスクを洗い出して評価しよう

- W-Aの実施および改善策の検討には**AWSの資料集**や**SA**を**有効活用**しよう！
  - それによって個々のアイテムの対応工数も変わってくるはず
- 例「ブルートフォースアタックの検知って
  どうやればいいの…？」
  SA「GuardDuty有効にすれば完了ですよ」
- AWSのセキュリティサービスを有効活用して、最小限の努力で最大限の成果を！

参考: Amazon GuardDutyによるインシデントレスポンス自動化

攻撃者

(1) EC2インスタンス
に総当り攻撃

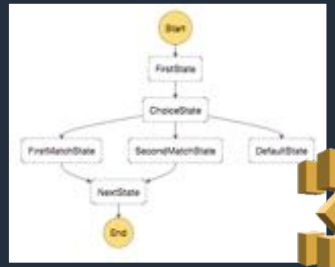
(2) 自動的に検出

Amazon GuardDuty

(3) 自動的に起動

AWS Lambda

(4) イベント内容に
応じたワークフロー
を呼び出し



AWS StepFunctions

(5) ワークフローに
従った、自動化された
インシデント対応の実行

セキュリティどこまでやればいいの

3. 従うべき法規制やガイドラインがあればそれを遵守しよう

- 金融、ヘルスケア、政府、国際連合などのコンプライアンス対応が必要な場合は、それぞれの要求を見て実装
  - スタートアップでもやることはやらねばなりません
- AWSでは、コンプライアンス対応のためのリソースをまとめたサイトにおいて、ホワイトペーパーやクイックスタート、リファレンスアーキテクチャを提供
- 対応に困ったらここでもまずはご相談ください！

コンプライアンスサイト

https://aws.amazon.com/jp/compliance/

コンプライアンス対応のためのリファレンス集

https://aws.amazon.com/jp/compliance/reference/

例: AWS FinTechリファレンス・アーキテクチャ[日本版]

FINTECH

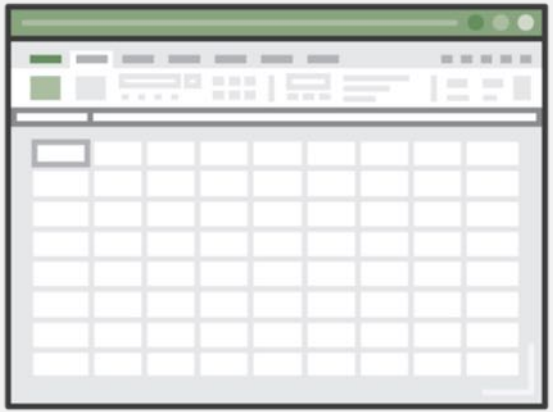
- AWS Japanでは2018年1月に FinTech リファレンス・アーキテクチャをリリース
- 金融事業で典型的に必要となる FISC, FISC API接続チェックリスト, PCI-DSS, ISO27001 の要求事項を網羅的に考慮し、対応内容をまとめた AWSオリジナルのリファレンス

https://aws.amazon.com/jp/compliance/fintech/

例: AWS FinTechリファレンス・アーキテクチャ[日本版]

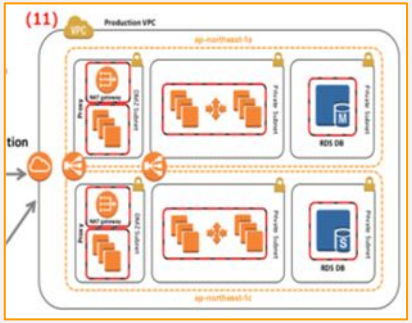
FinTechリファレンスアーキテクチャ

FinTechリファレンスガイド



検討事項と対応内容の
マッピング表

FinTechリファレンステンプレート



構成図

AWS CloudFormation
テンプレート

https://aws.amazon.com/jp/compliance/fintech/

例: FinTech - システムへの管理アクセス認証

大項目	中項目	小項目
インフラストラクチャー	システムのアクセス管理	システム管理のためのアクセス認証ポリシーを設け、システムの重要度とリスクに応じた強度のアクセス認証機能を実装する。

https://aws.amazon.com/jp/compliance/fintech/

AWSの該当事項	お客様の該当事項	推奨される追加の対応事項
AWSは、内部者による不適切なアクセスの脅威に対処するための統制を提供しています。取得している認証とサードパーティーによる証明で、論理アクセスの予防統制と発見的統制が評価されています。さらに、定期的なリスク評価時に、内部者によるアクセスの統制および監視方法を評価しています。 AWSは、所定の統制によってシステムとデータのアクセスを制限し、システムまたはデータに対するアクセスを制限および監視可能にしています。特権のあるユーザーアクセス制御は、AWS SOC 1、ISO27001、PCI、ITAR、およびFedRAMPsmの監査中に独立監査人によって....(略)	AWS環境においては、AWS IAMが利用可能です。Identity and Access Management (IAM)では、使用してユーザーIDの管理、セキュリティ認証情報の割り当て、ユーザーのグループ化による整理、ユーザーのアクセス許可の管理を一元的に行うことができます。ユーザーを複数人で共有せず、付与する権限は必要最低限のものにとどめてください。 認証情報には、パスワード、暗号キー、デジタル署名、および証明書が含まれます。また、AWSアカウントまたはIAMユーザーアカウントへのログインに多要素認証（MFA）を要求するオプションもあります。 IAMの詳細およびベストプラクティスについては、下記のURLを参照ください。 https://aws.amazon.com/iam/ 下記のURLを参照してください http://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/best-practices.html	個々の作業者に割り当てるIAMユーザについては別途作成するか、またはIAMのフェデレーションを用いて外部のIDプロバイダ（Active Directoryなど）と連携するなどして別途管理する必要があります。IAMでユーザを管理する場合、パスワードの条件やMFAなどの要件をIAMで設定できます。その他のIAMの設計、運用に関するベストプラクティスは下記のURLを参照ください。 http://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/best-practices.html プログラムやスクリプト内でID・パスワードなどの認証情報を扱う場合、AWS Systems ManagerのParameter Storeを用いることで安全に情報を管理することができ、ソースコードや設定ファイル内にそれらの情報をハードコーディングする必要がなくなります。Parameter Storeについては下記のURLを参照してください。 https://aws.amazon.com/jp/ec2/systems-manager/parameter-store/ また、AWSリソースへのアクセス時に必要な認証情報（Access Key や Secret Access Key）については、 .aws/credentials ファイルや環境変数を用いる方法の他に、EC2 のインスタンスプロファイルや AWS STS、Amazon Cognito を用いることで一時的な認証情報をその都度払い出すことができ、やはりハードコーディングを避けることができるようになります。一時的な認証情報の取得や活用方法については、下記のURLを参照してください。 http://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/id_credentials_temp.html これらの認証情報をソースコードに含めてバージョン管理ツール（Gitなど）にコミットしないように注意してください。AWSでは、誤った認証情報の公開を防ぐためのツールを提供しています。関連情報については、下記のURLを参照ください。 https://github.com/awslabs/git-secrets

例: AWS FinTechリファレンス・アーキテクチャ[日本版]

ホワイトペーパー、
AWS FinTech リファレンス・ガイド 日本版
を入手できます

↓「AWS FinTech」で検索↓

https://aws.amazon.com/jp/compliance/fintech/

セキュリティどこまでやればいいの – 参考資料

- AWS Black Belt Online Seminar資料＆動画
  - 利用者が実施する AWS 上でのセキュリティ対策(資料)
  - AWSにおけるセキュリティとコンプライアンス（資料|動画）
  - AWS re:Invent 2017 Recap: Security and Other （資料|動画）
  - 他関連資料多数 → https://aws.amazon.com/jp/aws-jp-introduction/
- AWS Summit/Dev Day講演資料 （2016 | 2017）
  - AWS Well-Architected フレームワークによるクラウド ベスト プラクティス (2017)（資料|動画）
  - DevSecOps on AWS - Policy in Code (2017) （資料|動画）
  - AWS セキュリティ入門（2017） （資料|動画）
  - AWS で実現するセキュリティ・オートメーション (2017)（資料|動画）

セキュリティどこまでやればいいの – 参考資料

- AWS のマネージドサービスを使ったセキュリティ強化のための自動化 by freee 坂井様
  https://speakerdeck.com/manabusakai/aws-security-automation
- コンテナを守る技術2018 by SUPINF 中丸様
  https://speakerdeck.com/pottava/container-security-20180310
- 東京海上日動&システムズ_サーバレスを活用した可視化分析の取り組み by 東京海上日動火災保険株式会社 村野様, 東京海上日動システムズ株式会社 畠山様
  https://s3-ap-northeast-1.amazonaws.com/20170808-awsentserverless/20170808-awsentserverless-1100-tokiomarine.pdf

おまけ：
AWSの技術情報を効率的にキャッチアップしたい

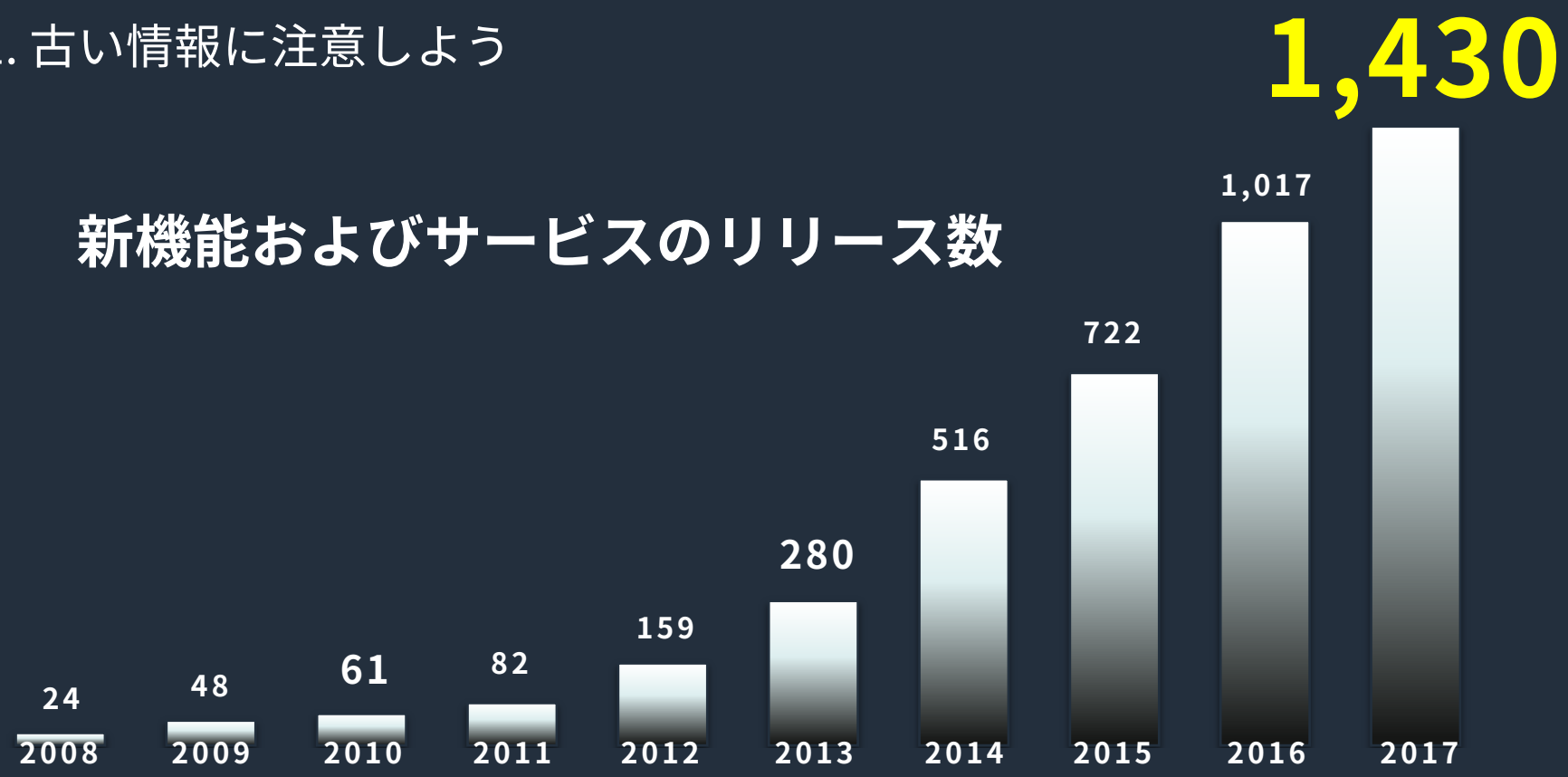
© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

AWSの技術情報を効率的にキャッチアップしたい

- よくいただく課題
  - 「AWSのサービスアップデート早すぎて勉強するのも大変」
  - 「そんなサービスあったんですね…（自分で作っちゃったよ」
- 本当にしたいことは何？
  - 自分のシステムに必要な機能を見逃して損したくない
  - 学習コストを抑えつつ動向を抑えておきたい
- 思考フロー
  1. 古い情報に注意しよう
  2. AWS BlackBelt Online Seminar を見よう
  3. AWSに相談しよう
  4. カンファレンス、コミュニティに参加しよう

AWSの技術情報を効率的にキャッチアップしたい

1. 古い情報に注意しよう



AWSの技術情報を効率的にキャッチアップしたい

1. 古い情報に注意しよう

- AWS Black Belt、AWS公式ブログ、Qiita、ブログ記事などの技術情報を漁るときは日付に注意
- あくまでも一次情報は公式ドキュメントに求めよう

AWSの技術情報を効率的にキャッチアップしたい

2. AWS BlackBelt Online Seminar を見よう

AWS Japan の Tech メンバーが無料でお届けする最新技術情報。

【火曜 12:00~13:00】→ お昼たべながら

主にAWSのソリューションや業界カットでの使いどころなどを紹介
(例：IoT、金融業界向け etc.)

【水曜 18:00~19:00】→ 夕食かコーヒーと一緒に

主にAWSサービスの紹介やアップデートの解説
(例：EC2、RDS、Lambda etc.)

AWS Black Belt Online Seminar

※開催曜日と時間帯は変更となる場合あり。最新の情報はWebで確認を！
オンラインセミナーのスケジュール&申し込みサイト
https://aws.amazon.com/jp/about-aws/events/webinars/

AWSの技術情報を効率的にキャッチアップしたい

2. AWS BlackBelt Online Seminar を見よう

AWS クラウドサービス活用資料集

AWS クラウドサービス活用資料集

まずは無料で始める »

日本担当チームへお問い合わせ

最新アップデート

働き方改革を実現するAWSのエンドユーザコンピューティングサービス【New!】	Slideshare	PDF	オンデマンドセミナー
Amazon QuickSight 【New!】	Slideshare	PDF	オンデマンドセミナー
AWS Lambda@Edge【New!】	Slideshare	PDF	オンデマンドセミナー
Amazon Container Services【New!】	Slideshare	PDF	オンデマンドセミナー

過去の資料、動画を閲覧可能

気になるトピックはぜひ一度動画でチェックすることをオススメ

「AWS 活用」で検索！

https://aws.amazon.com/jp/aws-jp-introduction/

AWSの技術情報を効率的にキャッチアップしたい

3. AWSに相談しよう

AWS では無料の技術相談を随時受け付けています

- 構成のレビューをしてほしい
- ○○のいい実現方法を教えて欲しい
- テーブル設計のレビューをしてほしい
- エンジニアのカルチャーってどうやってつくるの ...etc

お気軽にご相談ください！

AWSの技術情報を効率的にキャッチアップしたい

4. カンファレンス、コミュニティに参加しよう

AWS Summitでは今年も最新情報と事例セッションが目白押し

aws SUMMIT

AWS Summit Tokyo

2018 年 5 月 30 日（水）～ 6 月 1 日（金）

グランドプリンスホテル新高輪 | 来場無料

AWS Summit Osaka

2018 年 6 月 20 日（水）

グランフロント大阪 | 来場無料

AWSの技術情報を効率的にキャッチアップしたい

4. カンファレンス、コミュニティに参加しよう

実践的なノウハウを知るならやはりコミュニティで人の話を聞こう

AWS Startup
Tech Meetup

Meguro.dev

JAWS-UG

まとめ

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

2018 春期講習 まとめ

- 2018年春のあるある情報は以下の通りでした
  1. WebSocket、チャット、リアルタイムな何かをしたい
  2. 定時バッチ処理うまいことやりたい
  3. 一斉プッシュ通知送信後にサーバの負荷上がる問題
  4. セキュリティどこまでやればいいの
  5. おまけ: AWSの技術情報を効率的にキャッチアップしたい
- AWS スタートアップチームでは日々多くの技術相談を実施
- Start-upゼミシリーズ（2017夏期講習、2018春期講習）、AWS Black Belt Online Seminar 、カンファレンスやコミュニティなどなどで知識をひろげよう
- 困ったことがあればAWS スタートアップチームに相談しよう

Ask the AWS Start-up SAs!

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Thank You

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.