



SUMMIT
ONLINE
JAPAN

ナビタイムジャパンのAWSコスト最適化の秘訣

～ミックスインスタンスグループ有効活用およびコスト最適化手法～

小泉 亮輔

株式会社ナビタイムジャパン
ACTSインフラグループ マネージャー

田中 一樹

株式会社ナビタイムジャパン
ACTSインフラグループ

本日のアジェンダ

第一部

- ・ AWS全体コスト最適化

第二部

- ・ Amazon EC2費用削減

名前：小泉 亮輔

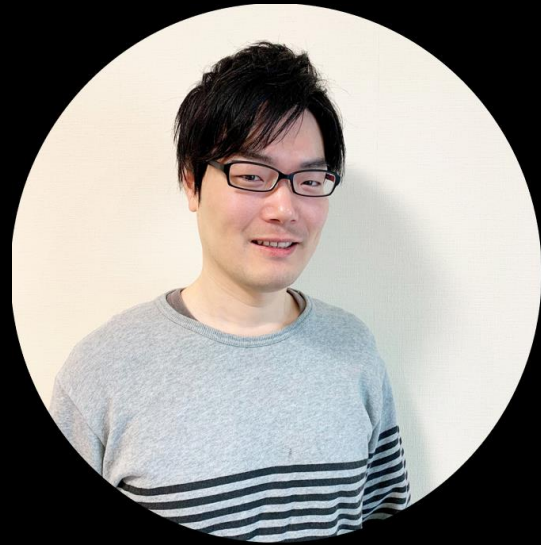
役割：インフラ(オンプレミス/クラウド)マネージャ

経歴：2010年 入社

主にデータベースエンジニアとして業務を開始し

2017年 よりインフラグループのマネージャとして従事

好きなAWSサービス：Amazon Aurora ,
Amazon route 53



会社紹介

- 株式会社 ナビタイムジャパン
- 設立 2000 年 3 月 1 日
- 業務内容
 - ナビゲーションサイト・アプリの運営開発
 - 通信カーナビゲーション事業
 - 経路探索エンジンのライセンス事業 など
- 従業員数 約 540 名 (エンジニア 80% 以上)



サービス紹介

公共交通



NAVITIME



乗換NAVITIME



バスNAVITIME



こみれぽ

ドライブ



カーナビタイム



ドライブサポーター



トラックカーナビ



バスカーナビ

ツーリング



ツーリングサポーター



自転車NAVITIME

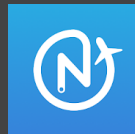
Travel/Inbound



JapanTravel



NAVITIME Transit



NAVITIME Travel



Plat

ヘルスケア



ALKOO

PC/SP Web



サービス紹介

ビジネスナビタイム
交通費精算パッケージ



ビジネスナビタイム
動態管理ソリューション



広告/アライアンス



オウンドメディア



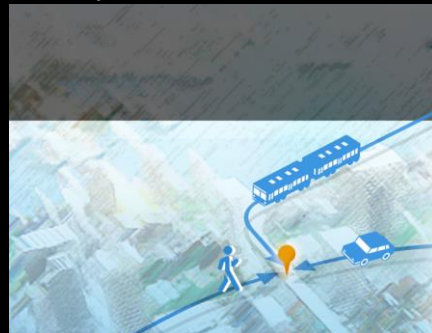
交通コンサルティング



電鉄/バス事業者向け
ソリューション



API / SDK



テレマティクスサービス



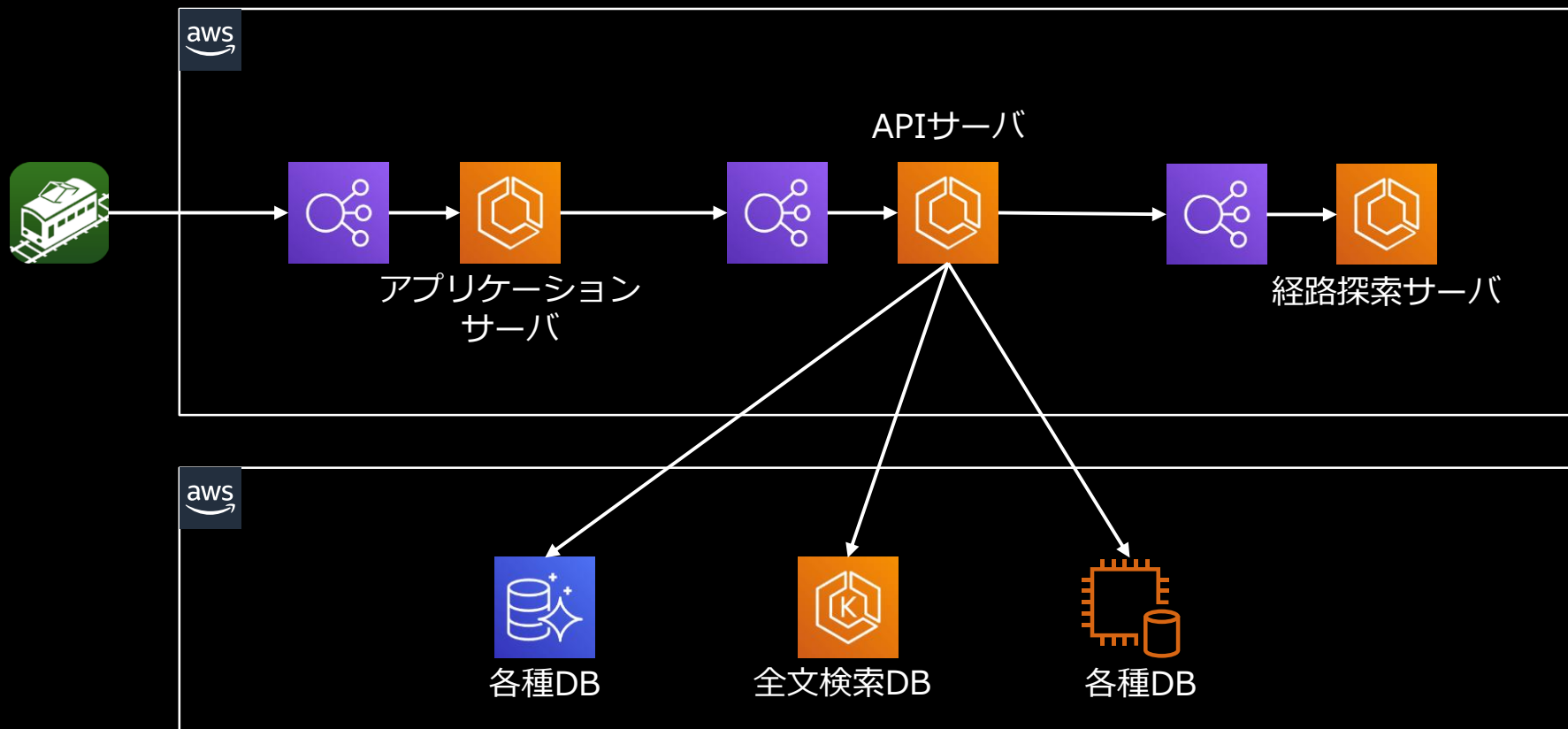
AWS 全体コスト最適化の取り組み

まずはナビタイムの代表的なサービスのシステム構成をご紹介します

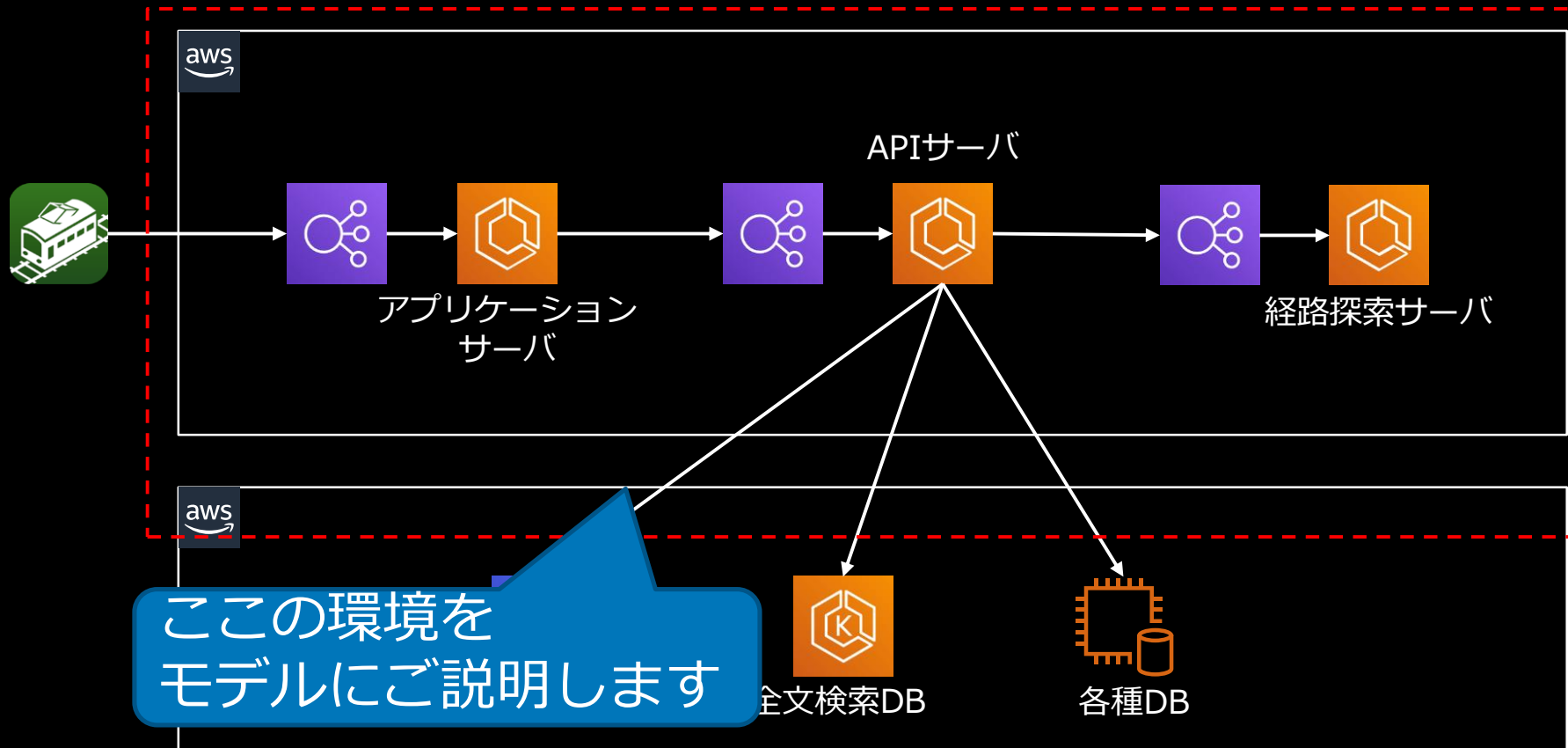
(AWS移行の経緯等は過去のaws summit資料をご参照ください)

<https://d1.awsstatic.com/events/jp/2017/summit/slide/D2T6-1.pdf>

ナビタイムサービスシステム構成



ナビタイムサービスシステム構成

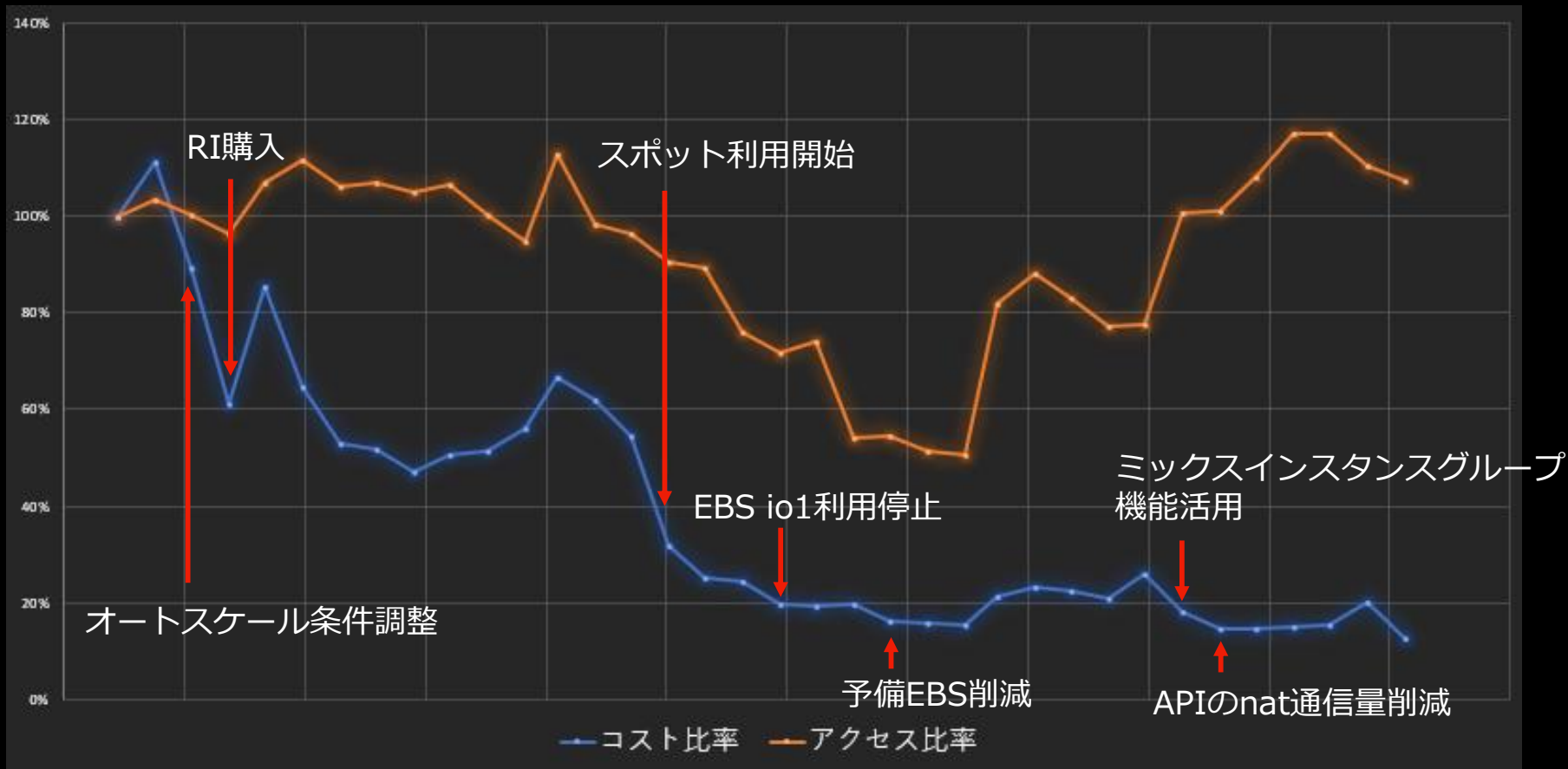


乗換ナビタイム専用環境のコスト最適化の推移はこちら

乗換NAVITIME ～AWS移行後のコスト推移～



乗換NAVITIME ～AWS移行後のコスト推移～



その成果は?

- クラウド移行後のコスト最適化の成果

(トータルでの)リクエスト120%増

※途中リクエスト削減策を行ったため参考



その成果は?

- クラウド移行後のコスト最適化の成果

(トータルでの)リクエスト120%増

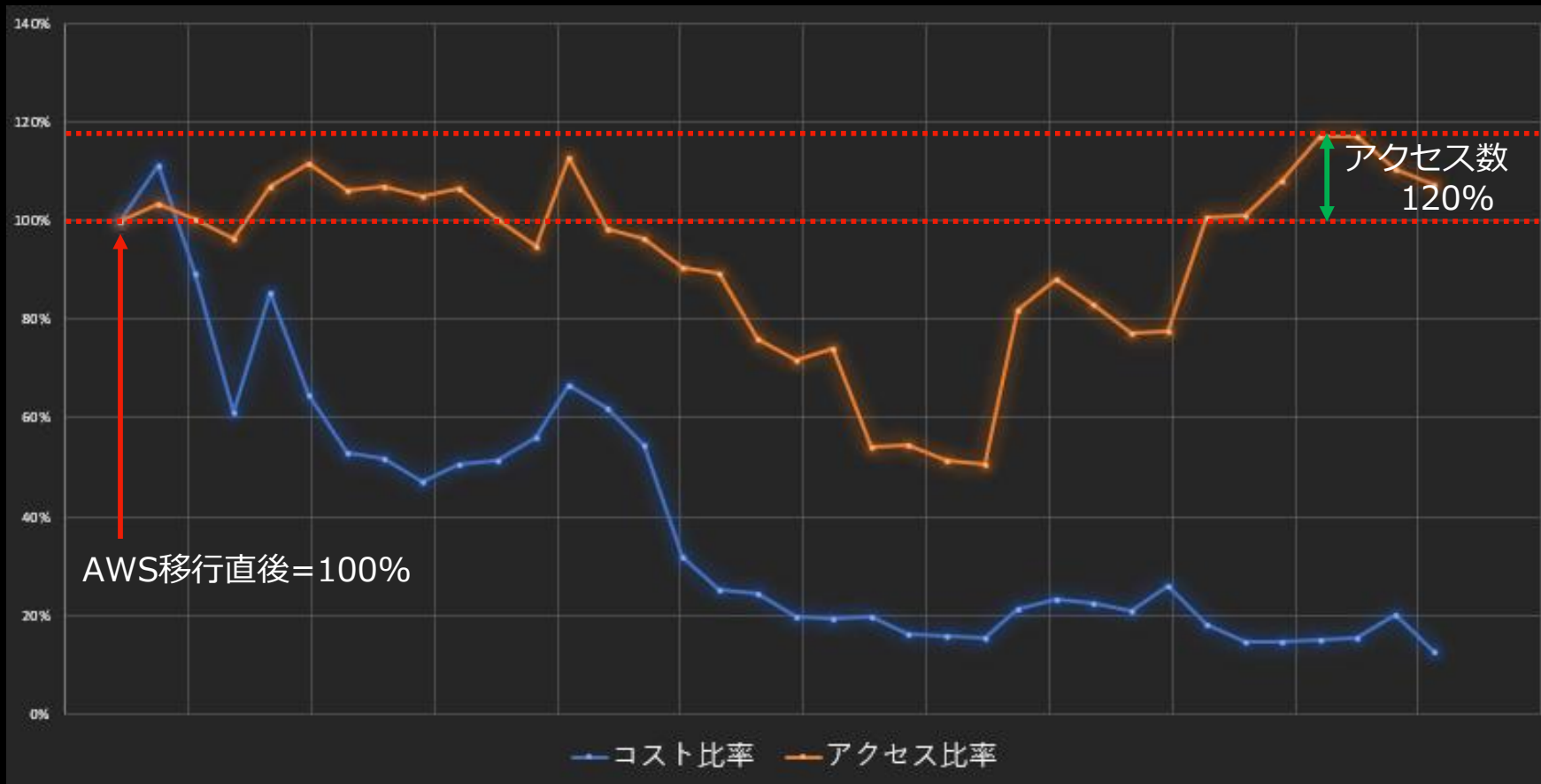
※途中リクエスト削減策を行ったため参考



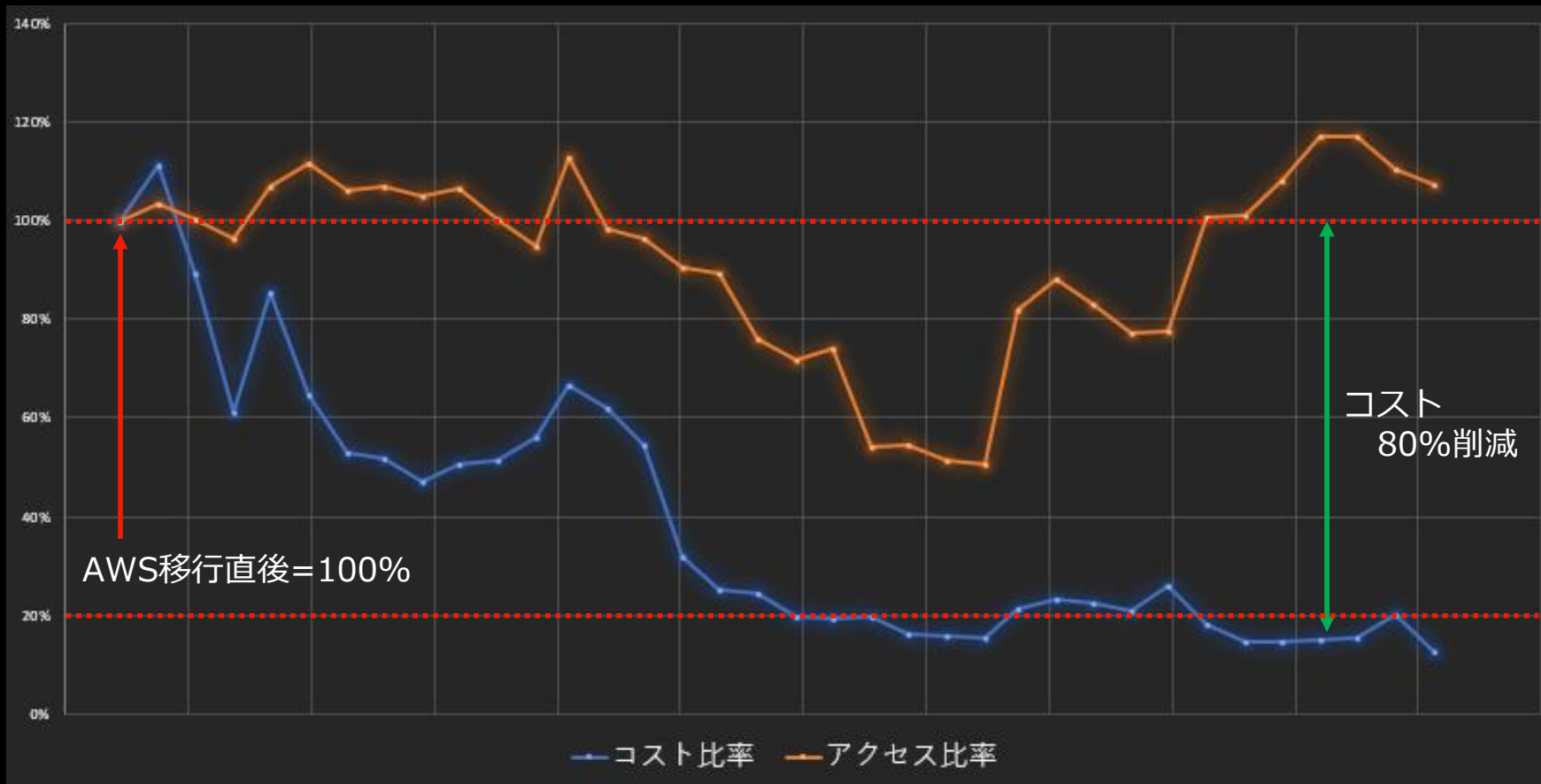
コスト **80%削減**

※ポイント：リクエスト増に対するコスト増の割合

乗換NAVITIME ～AWS移行後のコスト推移～



乗換NAVITIME ～AWS移行後のコスト推移～

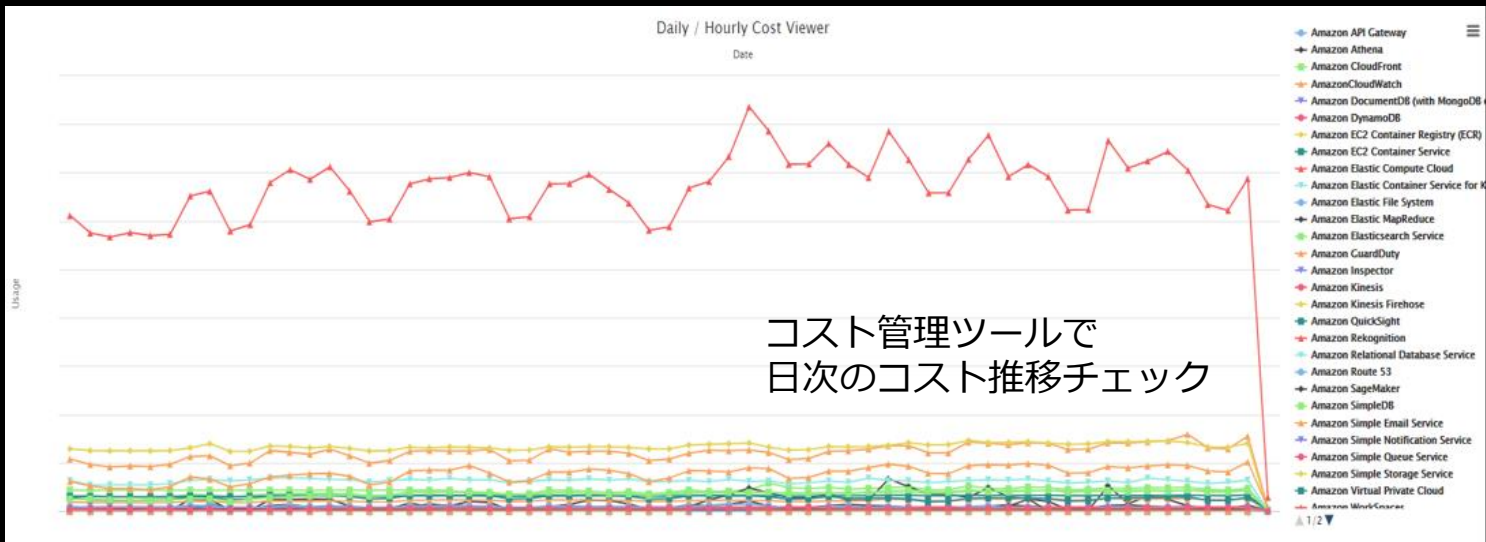
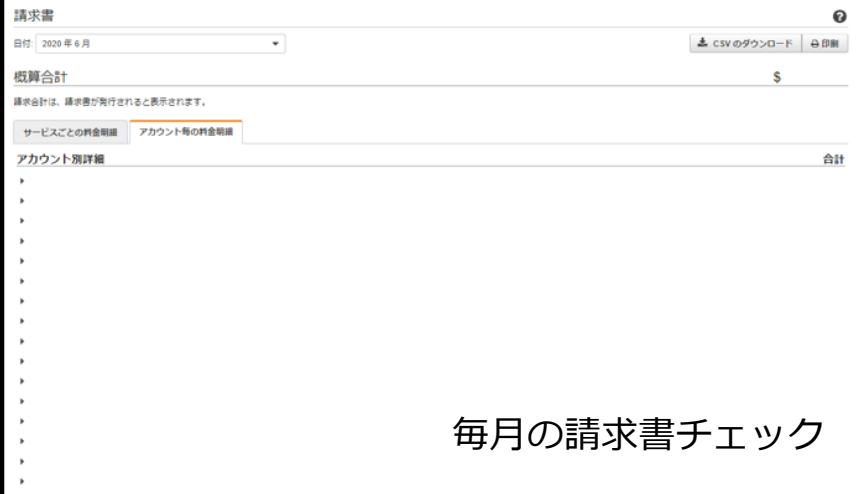


乗換NAVITIME ～AWS移行後のコスト推移～



どのようにコストカットの道筋をつけたのか

毎月ちゃんと請求書(請求内訳)を見る



コスト削減ポイント

コスト種別	要因	対応策
Amazon EC2	オンデマンド起動が多い	1. RI購入&スポットインスタンス活用 2. ミックスインスタンスグループ活用
EBS io1	性能を使いこなしていない	1. gp2へ移行しio1の利用停止 => 性能は容量確保で担保
EBS容量	スケールアウト上限までの予備EBS準備 (オートスケール高速化用)	1. データ量の容量削減 2. データ圧縮方式変更(zstd)によるダウンロード処理高速化
NAT Gateway	APIの外部通信費用	1. 同一ECSサービスへの複数ターゲットグループ登録を利用してint/extの分離 (2019年7月末サポート開始/8月導入)

コスト削減ポイント

後編で詳細にご紹介

コスト種別	要因	対応策
Amazon EC2	オンデマンド起動が多い	<ol style="list-style-type: none">1. RI購入&スポットインスタンス活用2. ミックスインスタンスグループ活用
EBS io1	性能を使いこなしていない	<ol style="list-style-type: none">1. gp2へ移行しio1の利用停止 => 性能は容量確保で担保
EBS容量	スケールアウト上限までの予備EBS準備 (オートスケール高速化用)	<ol style="list-style-type: none">1. データ量の容量削減2. データ圧縮方式変更(zstd)によるダウンロード処理高速化
NAT Gateway	APIの外部通信費用	<ol style="list-style-type: none">1. 同一ECSサービスへの複数ターゲットグループ登録を利用してint/extの分離 (2019年7月末サポート開始/8月導入)

コスト削減ポイント

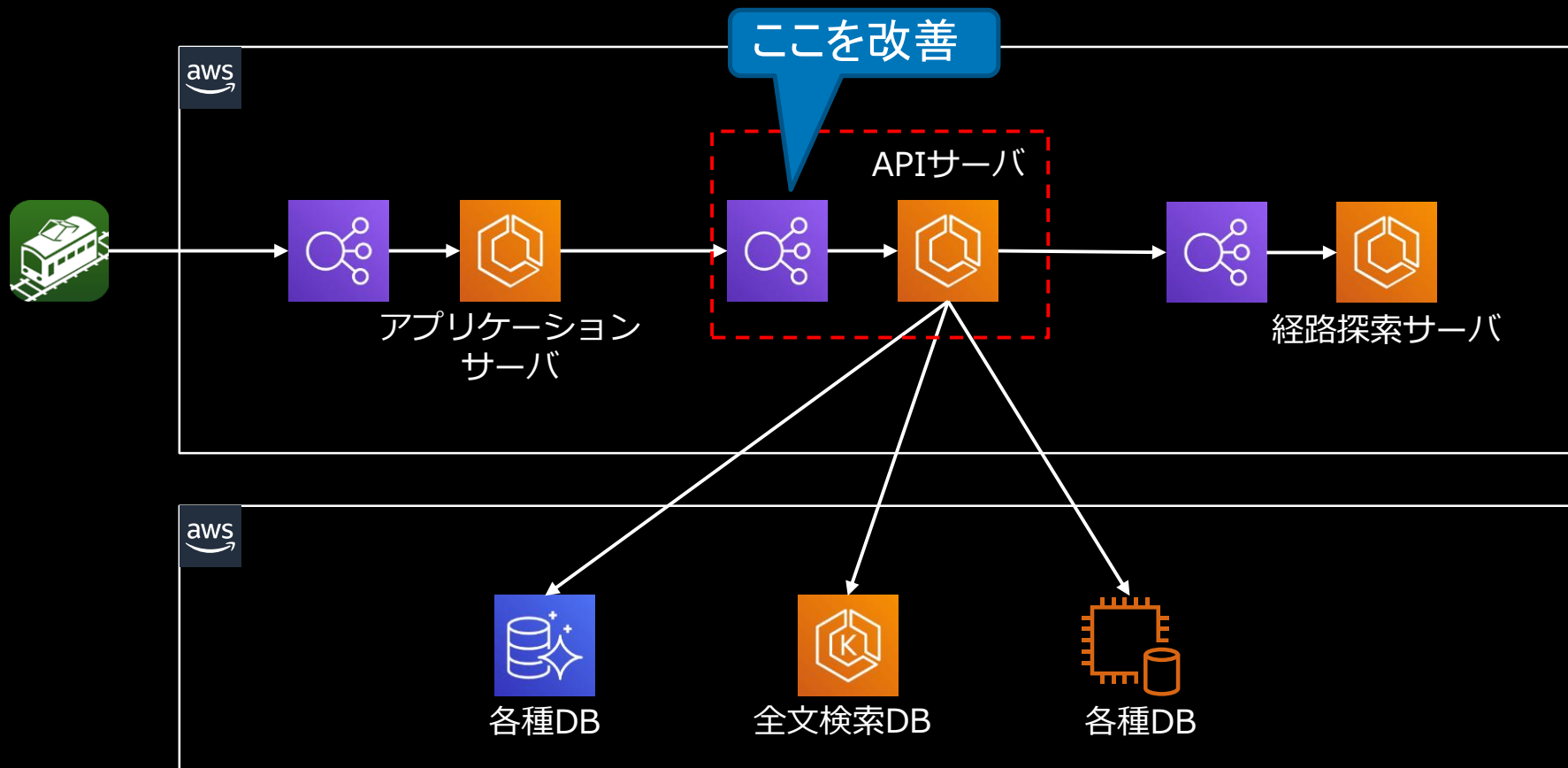
業務改善

コスト種別	要因	対応策
Amazon EC2	オンデマンド起動が多い	1. RI購入 & スポットインスタンス活用 2. ミックスインスタンスグループ活用
EBS io1	性能を使いこなしていない	1. gp2へ移行しio1の利用停止 => 性能は容量確保で担保
EBS容量	スケールアウト上限までの予備EBS準備 (オートスケール高速化用)	1. データ量の容量削減 2. データ圧縮方式変更(zstd)によるダウンロード処理高速化
NAT Gateway	APIの外部通信費用	1. 同一ECSサービスへの複数ターゲットグループ登録を利用してint/extの分離 (2019年7月末サポート開始/8月導入)

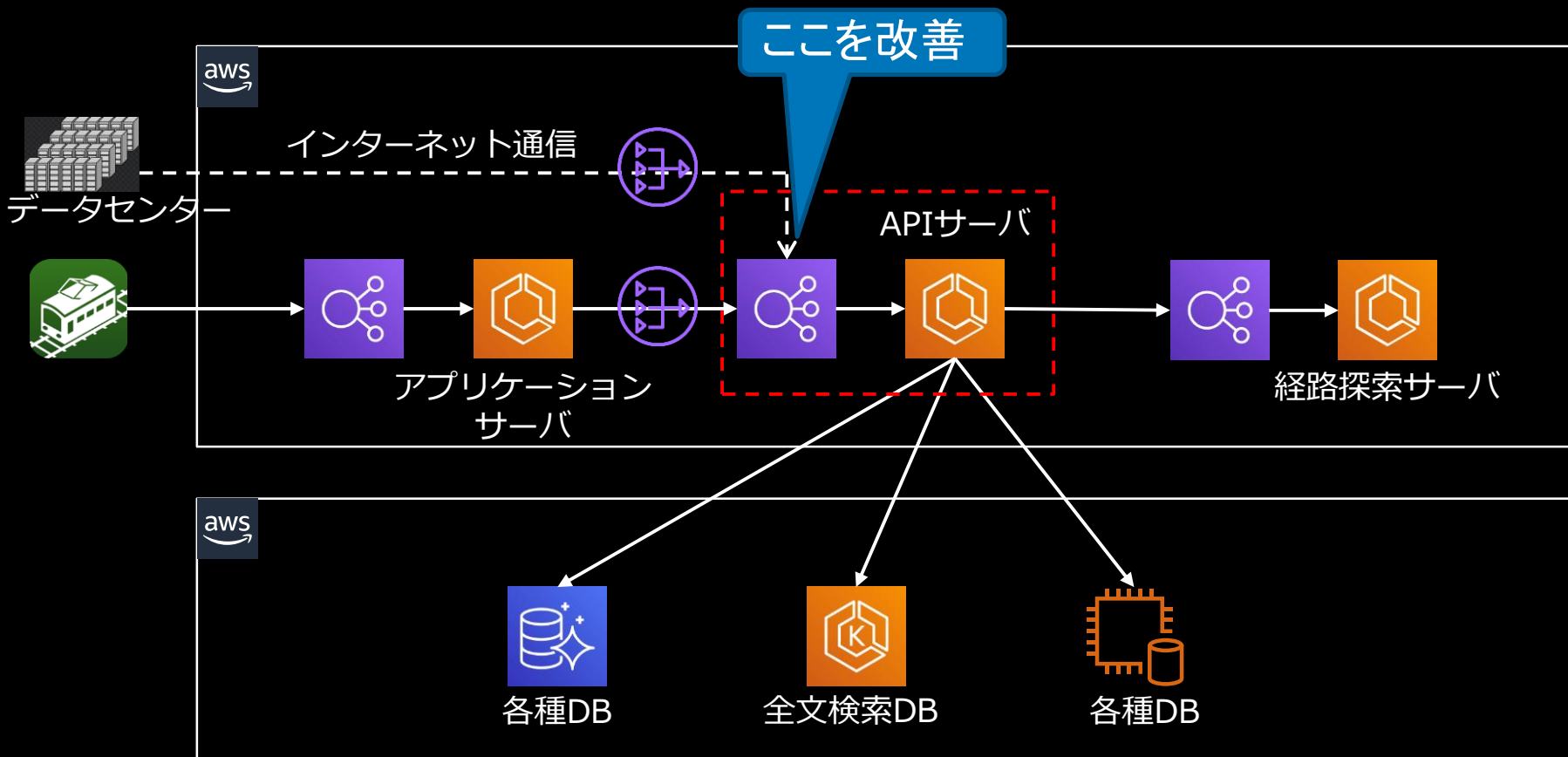
コスト削減ポイント

コスト種別	要因	対応策
Amazon EC2	オンデマンド起動が多い	<ol style="list-style-type: none">1. RI購入&スポットインスタンス活用2. ミックスインスタンスグループ活用
EBS io1	性能を使いこなしていない	<ol style="list-style-type: none">1. gp2へ移行しio1の利用停止 => 性能は容量確保で担保
EBS容量	スケールアウト上限までの予備EBS準備 (オートスケール高速化用)	<p>1. ...の容量判定 2. ...の処理高速化</p> <p>次ページで簡単にご説明</p>
NAT Gateway	APIの外部通信費用	<ol style="list-style-type: none">1. 同一ECSサービスへの複数ターゲットグループ登録を利用してint/extの分離 (2019年7月末サポート開始/8月導入)

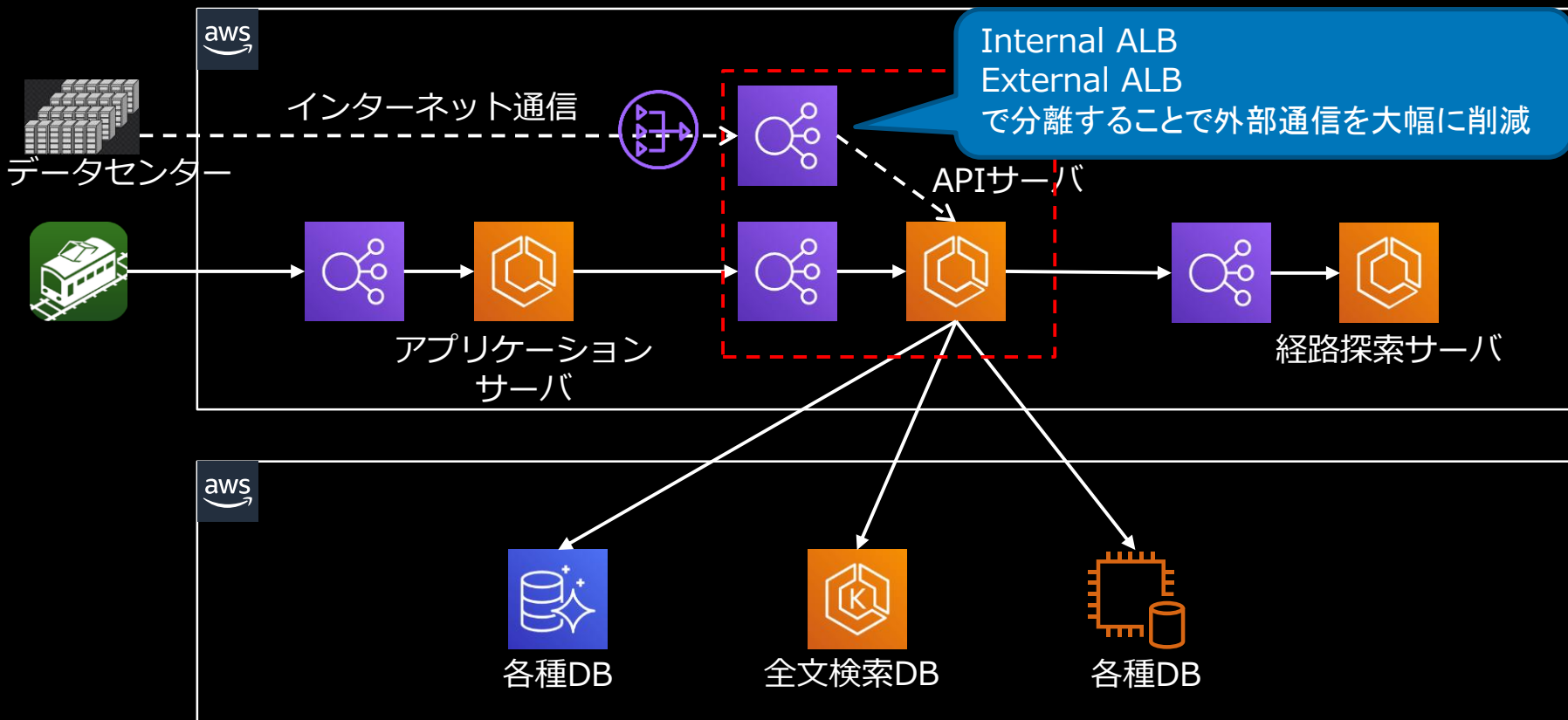
ナビタイムサービスシステム構成



ナビタイムサービスシステム構成(実は)



ナビタイムサービスシステム構成(実は)



その成果は?

再掲

- クラウド移行後のコスト最適化の成果

(トータルでの)リクエスト120%増

※途中リクエスト削減策を行ったため参考



コストは **80%削減**

※ポイント：リクエスト増に対するコスト増の割合

その成果は?

- オンプレの頃と比較してどうか

(トータルでの)リクエスト120%増

※途中リクエスト削減策を行ったため参考



その成果は?

- オンプレの頃と比較してどうか

(トータルでの)リクエスト120%増

※途中リクエスト削減策を行ったため参考



コストは **30%削減**

※移行直後はオンプレ時の300%増でした(汗)

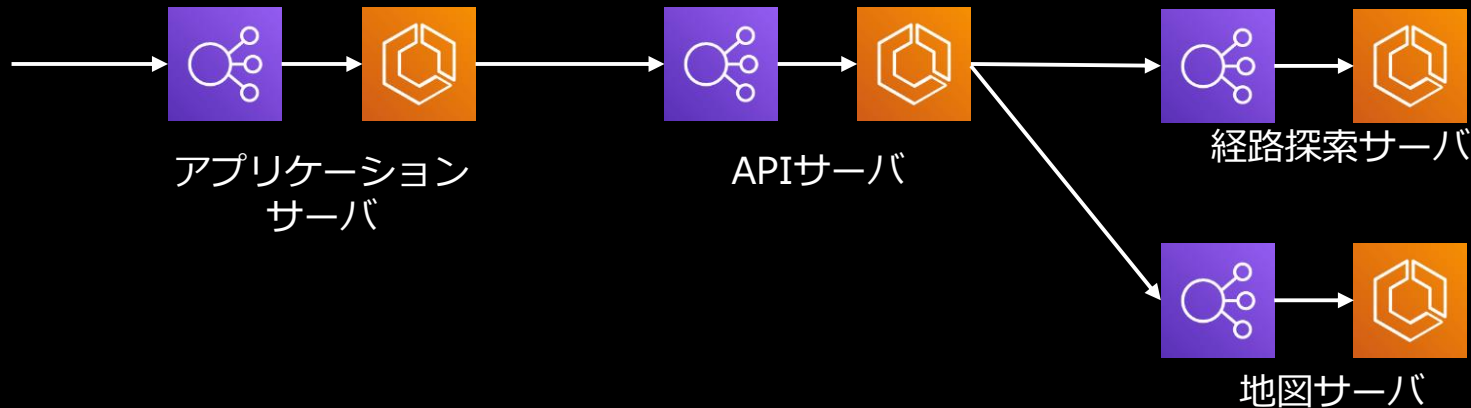
(前半の)最後に
その他の全体コスト削減への
チェックポイントのご紹介

その他 全体コストのチェックポイント

- ・クラウドネイティブな設計を意識する

Amazon API Gateway+Lambdaで賄える処理はほとんどの場合、Amazon EC2より安価。

参考) レガシー資産で変更が難しい場合(向き先変更含めて)

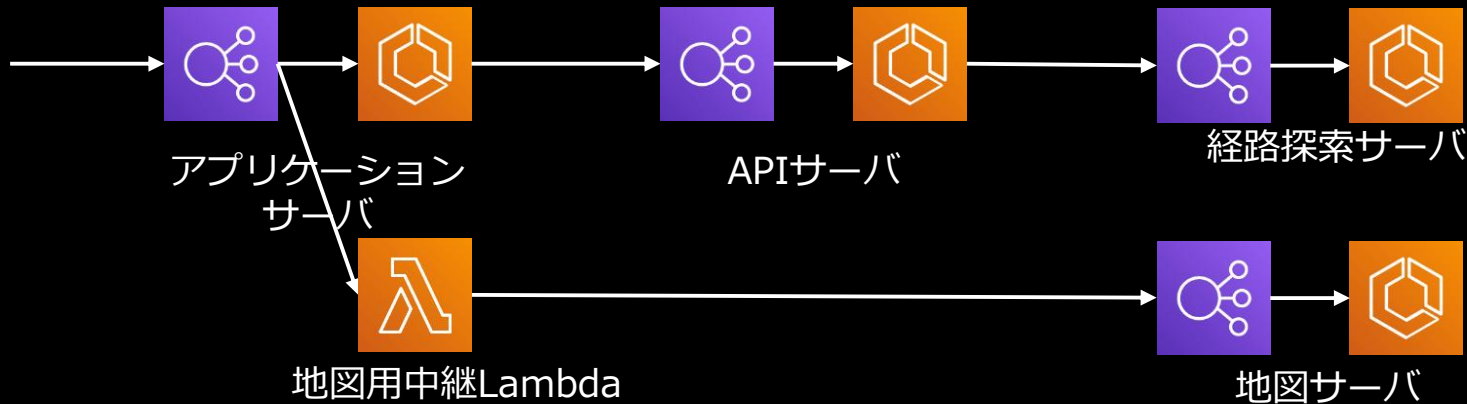


その他 全体コストのチェックポイント

- ・クラウドネイティブな設計を意識する

Amazon API Gateway+Lambdaで賄える処理はほとんどの場合、Amazon EC2より安価。

参考) レガシー資産で変更が難しい場合(向き先変更含めて)

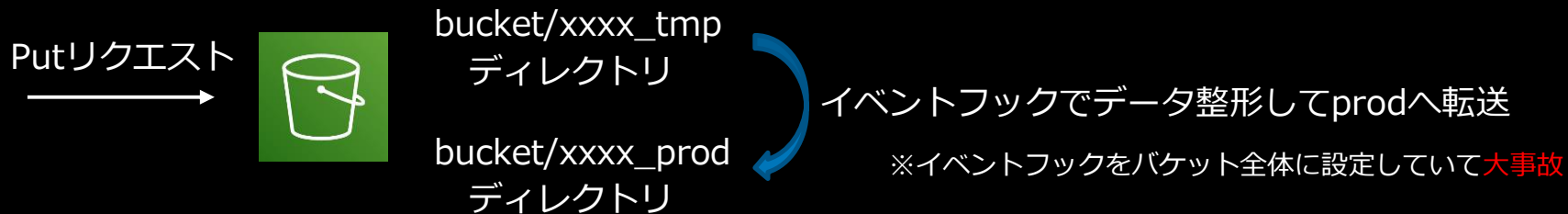


パスベースルーティングで
アプリケーションサーバ/APIサーバをスキップ

その他 全体コストのチェックポイント

- Amazon S3の利用料金

- 無限リクエストで**1日数十万**(Amazon S3+Lambda+CloudWatchlogs)



- Amazon S3の容量チェック

リリース用データ/バックアップ用データ
=> ライフサイクル設定の徹底

ログ系データ(長期保存データ)

=> Amazon Glacier/Intelligent-Tieringの有効活用 ※早期削除料金に注意

その他 全体コストのチェックポイント

- ・ CloudWatchLogsに注意

障害調査のためデバッグログをCloudWatchlogsに転送。

バグによる無限ログ出力/転送され **1日数十万**

その他 全体コストのチェックポイント

- ・ CloudWatchLogsに注意

障害調査のためデバッグログをCloudWatchlogsに転送。

バグによる無限ログ出力/転送され **1日数十万**



突然のコスト増を検知したい

その他 全体コストのチェックポイント

- ・ CloudWatchLogsに注意

障害調査のためデバッグログをCloudWatchlogsに転送。

バグによる無限ログ出力/転送され **1日数十万**



突然のコスト増を検知したい

AWS Cost and Usage Reports (AWS CUR)を利用

S3へデータを転送しAthenaで日次監視し

前日比較で2倍以上コストが上がっている課金タイプはアラート通知



BILLING APP 12:00

BILLING ALARM(2020-03-06)

以下のアカウント/サービスで前日より2倍以上の費用が加算されています

Amazon Simple Storage Service(アカウント番号): \$ 金額

Amazon Elastic MapReduce(アカウント番号): \$ 金額

Amazon Elastic Compute Cloud(アカウント番号): \$ 金額

その他 全体コストのチェックポイント

- ・ NAT Gatewayの費用削減
 - 外部通信しない環境でも意外と使っている
 - 特にAmazon S3, Amazon Kinesis, CloudWatchLogsなど
 - 費用削減のためにVPCエンドポイントを使うとコストが増える可能性がある
るので注意

その他 全体コストのチェックポイント

- ・ VPCエンドポイント
 - ゲートウェイエンドポイント
 - Amazon S3, Amazon DynamoDB
 - インターフェイスエンドポイント
 - Amazon API Gateway, CloudWatchLogs, Amazon Kinesis, Amazon ECR, ...

その他 全体コストのチェックポイント

- ・ VPCエンドポイント
 - ゲートウェイエンドポイント
 - 追加料金不要
 - インターフェイスエンドポイント
 - 起動時間課金+転送量課金

その他 全体コストのチェックポイント

- ・ VPCエンドポイント

- ゲートウェイエンドポイント

- 作成すればNATGWの通信費を抑えられる

- インターフェイスエンドポイント

- 起動時間課金+転送量課金 > NATGWの通信費となる可能性がある

- エンドポイントを3AZで起動する場合の損益分岐点は転送量600GB/月

全体的なコスト削減策については
以上となります

続きますして

「コスト削減ポイント：Amazon EC2のコスト削減」
について掘り下げてご紹介いたします。

Amazon EC2費用削減

名前：田中 一樹

役割：Cloud Architect

経歴：2013年新卒入社。2017年～クラウド系の作業に従事

好きなAWSサービス：CloudFormation, Lambda

キャンプ好きエンジニア



Amazon EC2の購入オプションはみなさんご存知でしょうか？

Amazon EC2

- 主な購入オプション

On-Demand
Instances

Spot Instances

Savings Plans

Reserved Instances

Scheduled
Reserved Instances

Amazon EC2

- 例) m5.large(東京リージョン)

On-Demand
Instances

\$0.124/h

Spot Instances

\$0.035/h

Savings Plans
(1年全額前払い)

\$0.073/h

Reserved Instances
(1年全額前払い)

\$0.073/h

Scheduled
Reserved Instances

※価格は2020/5時点

Amazon EC2

- 例) m5.large(東京リージョン)

On-Demand
Instances

\$0.124/h

Spot Instances

\$0.035/h

Savings Plans
(1年全額前払い)

\$0.073/h

Reserved Instances
(1年全額前払い)

\$0.073/h

Scheduled
Reserved Instances

※価格は2020/5時点

Spot Instancesは脅威の**71%OFF**

リスクは伴うものの、Spot Instancesを使わない手はない



Spot Instancesを制するものがAWSのコストカットを制す

Amazon EC2コストカットへの道①

- Spot Instancesだけでいけるのか！？

Amazon EC2コストカットへの道①

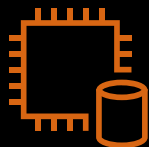
- Spot Instancesだけでいけるのか！？

- Spot Instances使えそうなもの

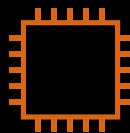


コンテナ系

- Spot Instances使えなさそうなもの



DB on EC2



コンバータ類



一定数のOn-Demand Instancesが必要

Amazon EC2コストカットへの道①

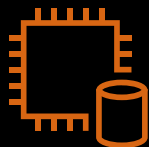
- Spot Instancesだけでいけるのか！？

- Spot Instances使えそうなもの

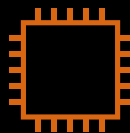


コンテナ系

- Spot Instances使えなさそうなもの



DB on EC2



コンバータ類



Spot+Reservedで稼働させる

Amazon EC2コストカットへの道②

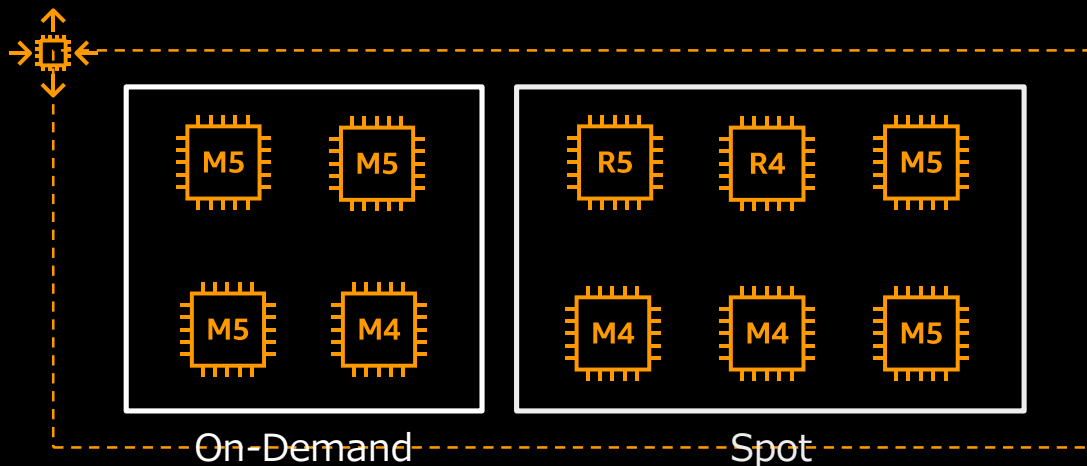
- スケールアウトはどうするのか？

Amazon EC2コストカットへの道②

- スケールアウトはどうするのか？
 - ミックスインスタンスグループ機能を使う

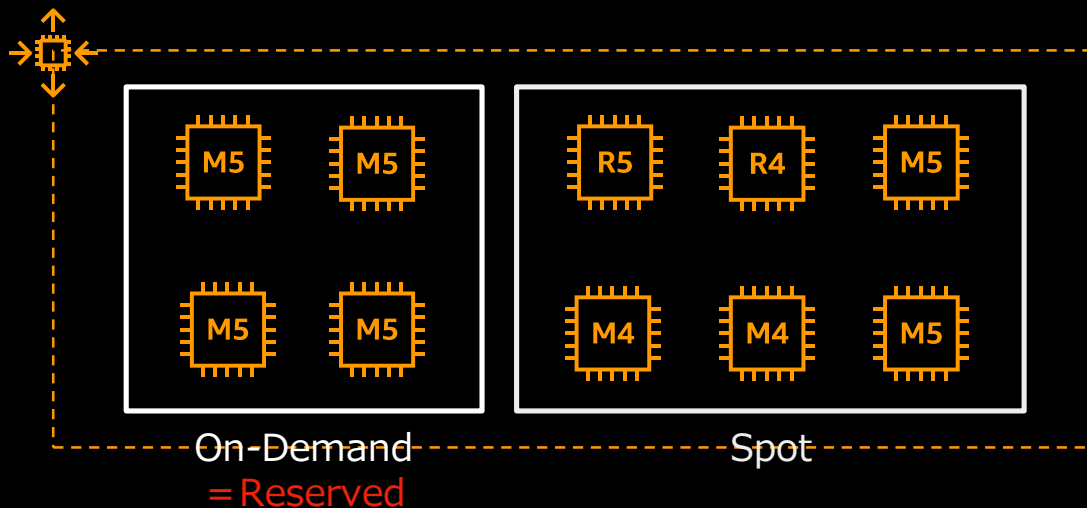
Amazon EC2コストカットへの道②

- ミックスインスタンスグループ機能
 - 1度のリクエストでOn-Demand/Spotの起動が可能
 - 複数のインスタンスタイプが指定可能



Amazon EC2コストカットへの道②

- ミックスインスタンスグループ機能
 - On-DemandをReservedで購入したインスタンスが起動するようにする
- 例) m5.largeをReservedで購入



Amazon EC2コストカットへの道②

● ミックスインスタンスグループ機能

起動テンプレート ⓘ	[REDACTED]	アベイラビリティゾーン ⓘ	ap-northeast-1a, ap-northeast-1c, ap-northeast-1d
起動テンプレートのバージョン ⓘ	1	サブネット ⓘ	[REDACTED]
起動テンプレートの説明 ⓘ	-	Classic ロードバランサー ⓘ	
インスタンスタイプ ⓘ	m5.large, m5d.large, r5.large, m4.large, r4.large	ターゲットグループ ⓘ	
スポット割り当て戦略 ⓘ	最低料金	ヘルスチェックのタイプ ⓘ	EC2
スポットインスタンスプール ⓘ	5	ヘルスチェックの猶予期間 ⓘ	180
オプションのオンデマンドベース ⓘ	6	インスタンスの保護 ⓘ	
オンデマンド割合 ⓘ	0%	終了ポリシー ⓘ	Default
希望するキャパシティ ⓘ	14	停止したプロセス ⓘ	
最小 ⓘ	6	Max Instance Lifetime ⓘ	
最大 ⓘ	36	プレイスメントグループ ⓘ	
		デフォルトのクールダウン ⓘ	300

Amazon EC2コストカットへの道②

- 設定の詳細は先日の発表資料をご覧ください

ECS×スポットインスタンス活用の秘訣

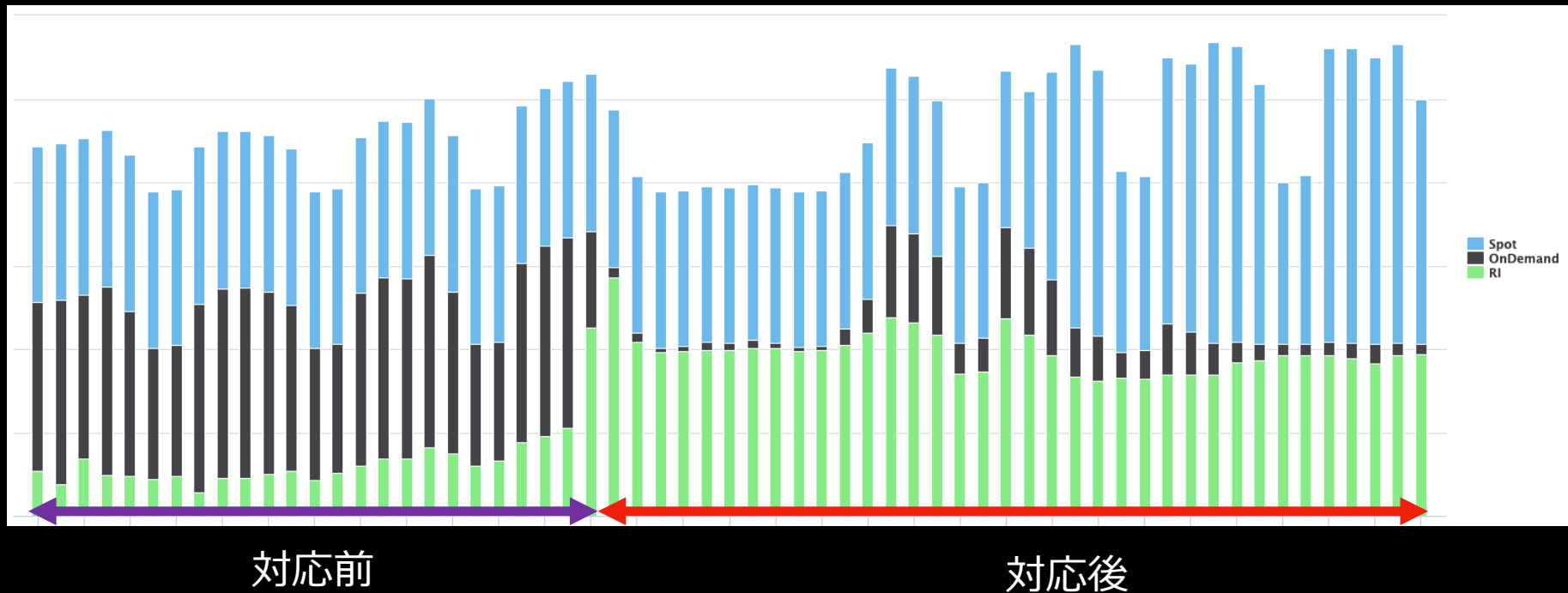
<https://aws.amazon.com/jp/blogs/news/container-ec2-spot/>

結果どうなったか

結果

- 起動割合

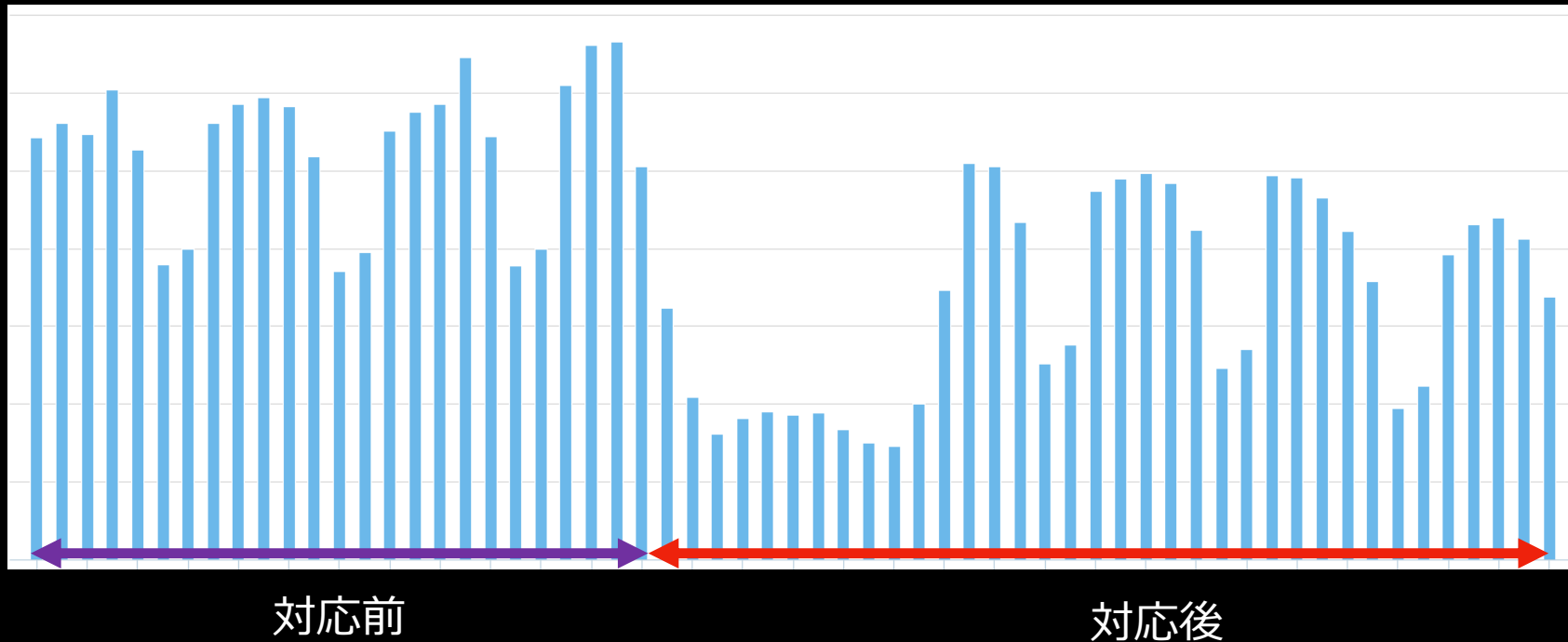
On-Demand起動時間約90%削減



結果

約30%～費用削減

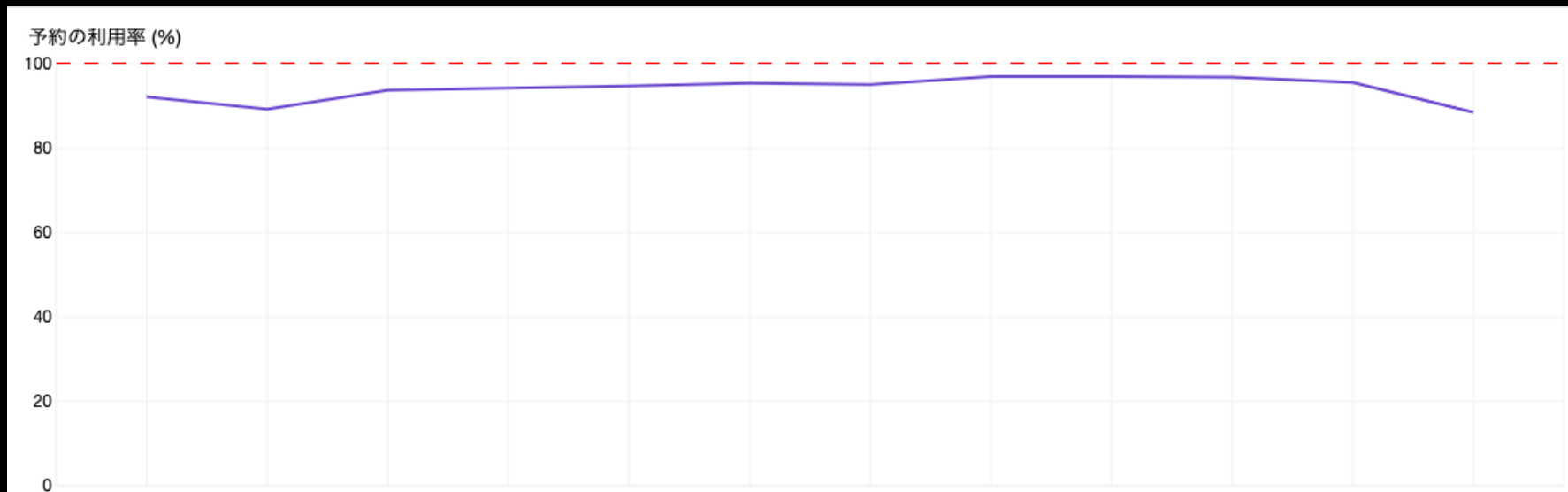
- コスト推移



Reserved Instancesの利用状況は？

結果

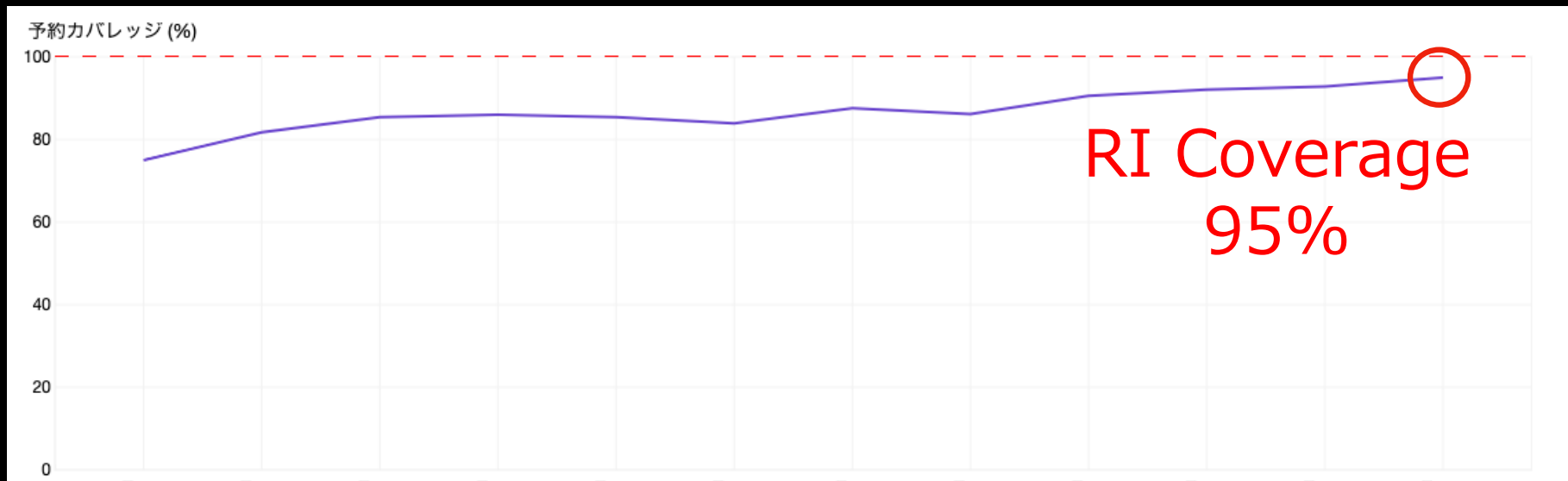
- RI利用状況



RI Usage 95%を堅持

結果

- RI利用状況



結果

- Amazon EC2インスタンス費用
 - 全インスタンスをOn-Demandで起動した場合
 - 今回の戦略(Spot+RI+一部On-Demand)にした場合を1年間の利用状況で比較

結果

- Amazon EC2インスタンス費用
 - 全インスタンスをOn-Demandで起動した場合
 - 今回の戦略(Spot+RI+一部On-Demand)にした場合を1年間の利用状況で比較

On-Demandだけのときよりも**約50%**安くEC2を利用

結果

- とあるアカウントでSpot+Reservedを使うとEC2利用費用が30%削減できた
- 全アカウントで購入したReservedは95%以上利用できた
- 全アカウントでReservedのカバレッジは95%以上達成
 - オンデマンドインスタンスは全インスタンスの5%

この戦略の問題点と機能要望

問題点

- Reservedで購入したインスタンスタイプが起動しないことがある
- Spotが確保できないことがある
- On-Demandインスタンスの割合は結局0にできない
- Savings Plansが使えない
- 複数AMIが指定できない

問題点

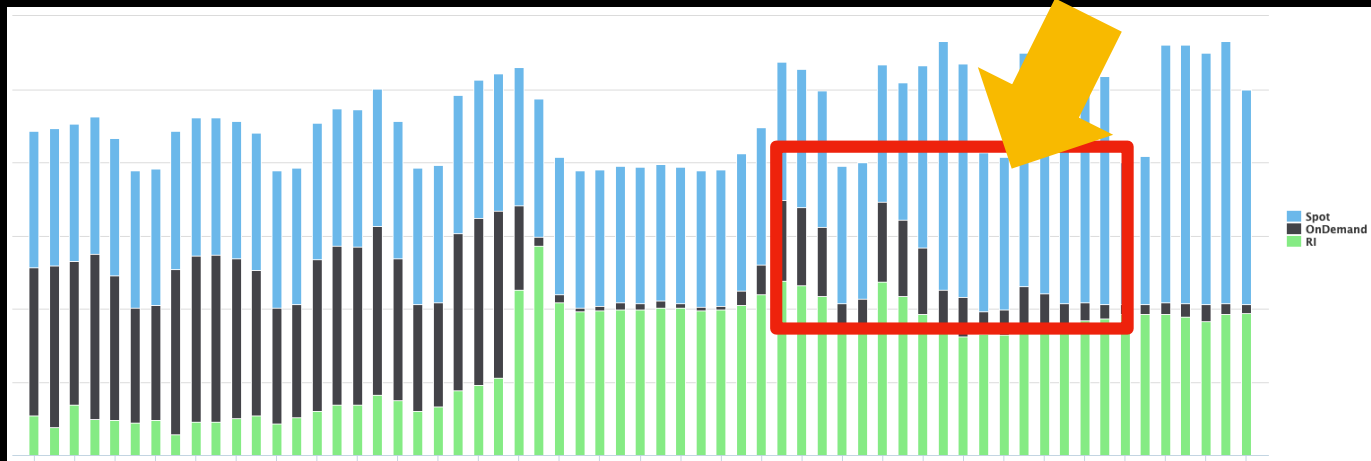
- Reservedで購入したインスタンスタイプが起動しないことがある

問題点

- Reservedで購入したインスタンスタイプが起動しないことがある
 - ミックスインスタンスグループ機能で指定したインスタンスタイプの先頭から順に起動を試みる
 - 予定外のインスタンスタイプがOn-Demandで起動することがある

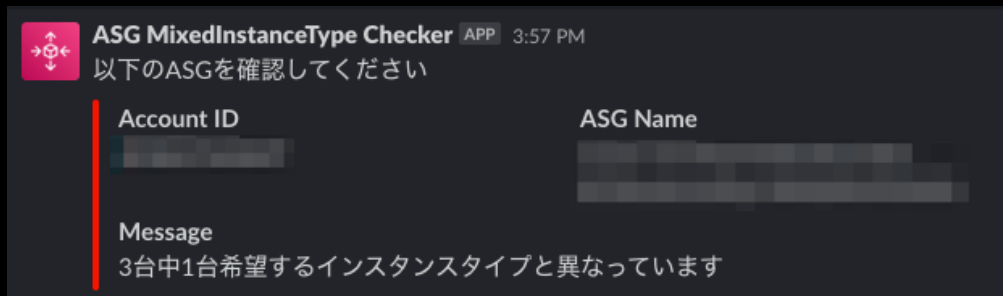
問題点

- Reservedで購入したインスタンスタイプが起動しないことがある
 - ミックスインスタンスグループ機能で指定したインスタンスタイプの先頭から順に起動を試みる
 - 予定外のインスタンスタイプがOn-Demandで起動することがある



問題点

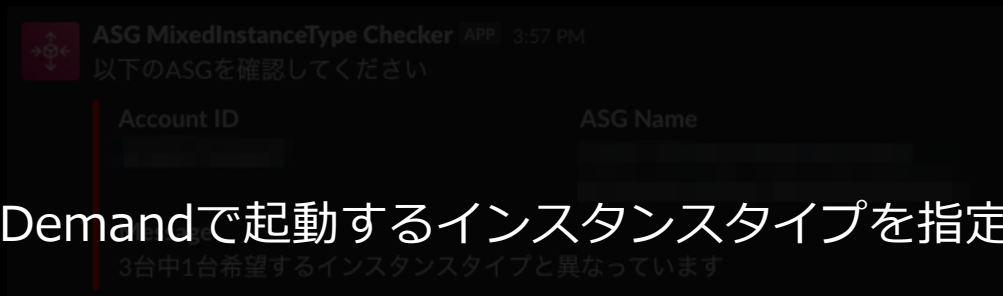
- Reservedで購入したインスタンスタイプが起動しないことがある
 - 予定外のインスタンスタイプが起動してきたら通知



- 対象のインスタンスをシャットダウンし、希望するインスタンスタイプで起動することを祈る

問題点

- Reservedで購入したインスタンスタイプが起動しないことがある
 - 予定外のインスタンスタイプが起動してきたら通知



On-Demandで起動するインスタンスタイプを指定したい

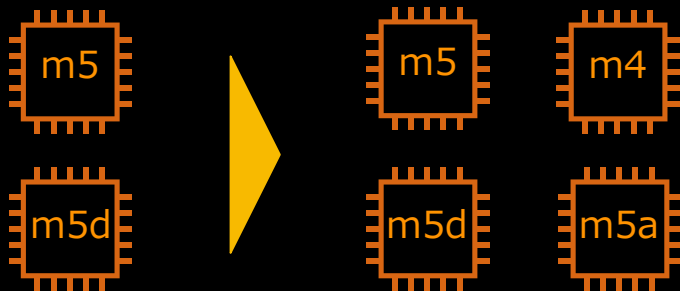
- 対象のインスタンスをシャットダウンし、希望するインスタンスタイプで起動することを祈る

問題点

- Spotが確保できないことがある

問題点

- Spotが確保できないことがある
 - ミックスインスタンスグループ機能で指定するインスタンスタイプを複数にする



- たいてい、旧世代のインスタンスタイプは起動する

問題点

- Spotが確保できないことがある
 - EC2 Fleetで指定するインスタンスタイプを複数にする



- たいてい、旧世代のインスタンスタイプは起動する

問題点

- On-Demandインスタンスの割合は結局0にできない

問題点

- On-Demandインスタンスの割合は結局0にできない
 - 落ちたら困るコンバータなどのインスタンスが存在
 - そういったインスタンスは日中/夜間/土日など使っていないことも多々ある

問題点

- On-Demandインスタンスの割合は結局0にできない
 - 落ちたら困るインスタンスが存在
 - 日中/夜間/土日など使っていないことも多々ある

東京リージョンにScheduled Reserved Instancesを・・・

問題点

- Savings Plansが使えない

問題点

- ~~Savings Plansが使えない~~
- 今の運用が確立されすぎて、Savings Plansが使えない

問題点

- ~~Savings Plansが使えない~~
- 今の運用が確立されすぎて、Savings Plansが使えない
 - Savings Plansを使うと今のCoverageが維持できない
 - RI Coverage/RI Usage/コスト割引率を維持するためにはRIの方が確実
 - インスタンスタイプは半年に1回しか原則、見直さない

問題点

- 複数AMIが指定できない

問題点

- 複数AMIが指定できない
 - ARMとAMDの異なるAMIが指定できない
 - ARMとAMDを共存する場合はAutoScaling Groupを複数作るしかない

問題点

- 複数AMIが指定できない
 - ARMとAMDの異なるAMIが指定できない
 - ARMとAMDを共存する場合はAutoScaling Groupを複数作るしかない

AutoScaling Group1つで複数AMI

もしくは

AMI1つでマルチアーキテクチャに対応可能にしてほしい

今後の展望

- API Call数の削減や通信量の削減をさらに実施予定
 - そのためのログやツールが潤沢になることを期待
- Amazon EC2の単価を見ながらコストパフォーマンスの高いインスタンスタイプをこれからも積極採用

NAVITIME



SUMMIT
ONLINE
JAPAN