導入支援サービスの現場から見た、企業でのクラウド導入の勘所

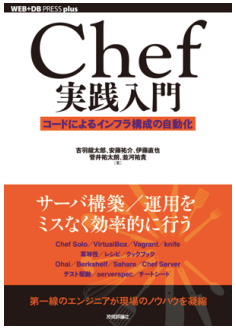
TC-04 テクノロジートラック

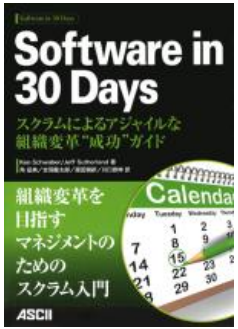
アマゾン データサービス ジャパン
プロフェッショナルサービス本部/部長
吉羽 龍太郎

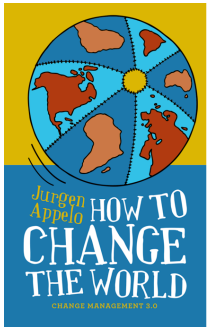
amazon web services

自己紹介

- 吉羽龍太郎
  - アマゾン データ サービス ジャパン株式会社
  - プロフェッショナルサービス本部 部長
  - クラウドコンピューティング、アジャイル開発、自動化を軸にして企業のアジリティの向上を支援







amazon web services

本日のアジェンダ

- AWS Professional Servicesとは
- クラウド導入範囲の選択肢
- All-inの進め方と勘所

本日のアジェンダ

- AWS Professional Servicesとは
- クラウド導入範囲の選択肢
- All-inの進め方と勘所

導入支援サービス？

AWSではお客様のAWS利用を支える3つのサービスを用意

#1 AWS Support

- 日本語でのサポートをご提供しております
  http://aws.amazon.com/jp/premiumsupport/

	ベーシック	デベロッパー	ビジネス	エンタープライズ
サポートフォーラム	利用可能	利用可能	利用可能	利用可能
サポートへのコンタクト	EC2の健全性エラーが発生した場合	コンタクトフォーム	電話、チャット、コンタクトフォーム	電話、チャット、コンタクトフォーム
最速初回応答時間	不可	12時間以内（営業時間内）	1時間以内	１５分以内
連絡先登録		1	5	無制限
24/365対応	なし	なし	あり	あり
上級サポートエンジニアへの直接ルーティング	なし	なし	あり	あり
専任スタッフ	なし	なし	なし	あり
特別サポート	なし	なし	なし	あり
料金（月額）	無料	4,900円	AWS利用総額の 0円~100万円:　10% 100万円~800万円: 7% 800万円~2,500万円: 5% 2,500万円~　3% （最低10,000円）	AWS利用総額の 0円~1,500万:　10% 1,500万~5,000万: 7% 5,000万~10,000万: 5% 10,000万~　3% （最低150万円）

※1ドル100円換算

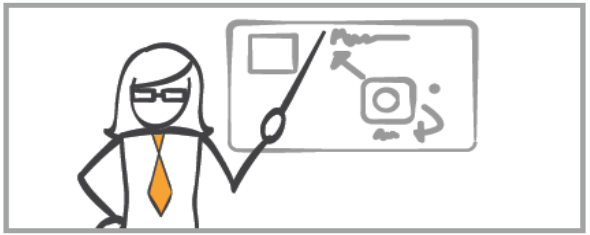
#2 AWS トレーニングと認定制度

- AWSサービスの知識とスキルをお客様に提供するための、教育と認定のプログラムです
- 技術者のレベルや経験に合わせて、複数のコースをご用意しています

セルフペースドラボ

自習（ハンズオン）を行うことで、AWSサービスに慣れ、さらに新しい知識を吸収し、AWS経験値を上げる。

トレーニング



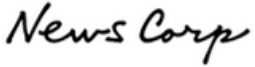
自信を持ってAWS上で設計、開発、運用ができるようになるAWS知識やスキルを習得する。

認定制度

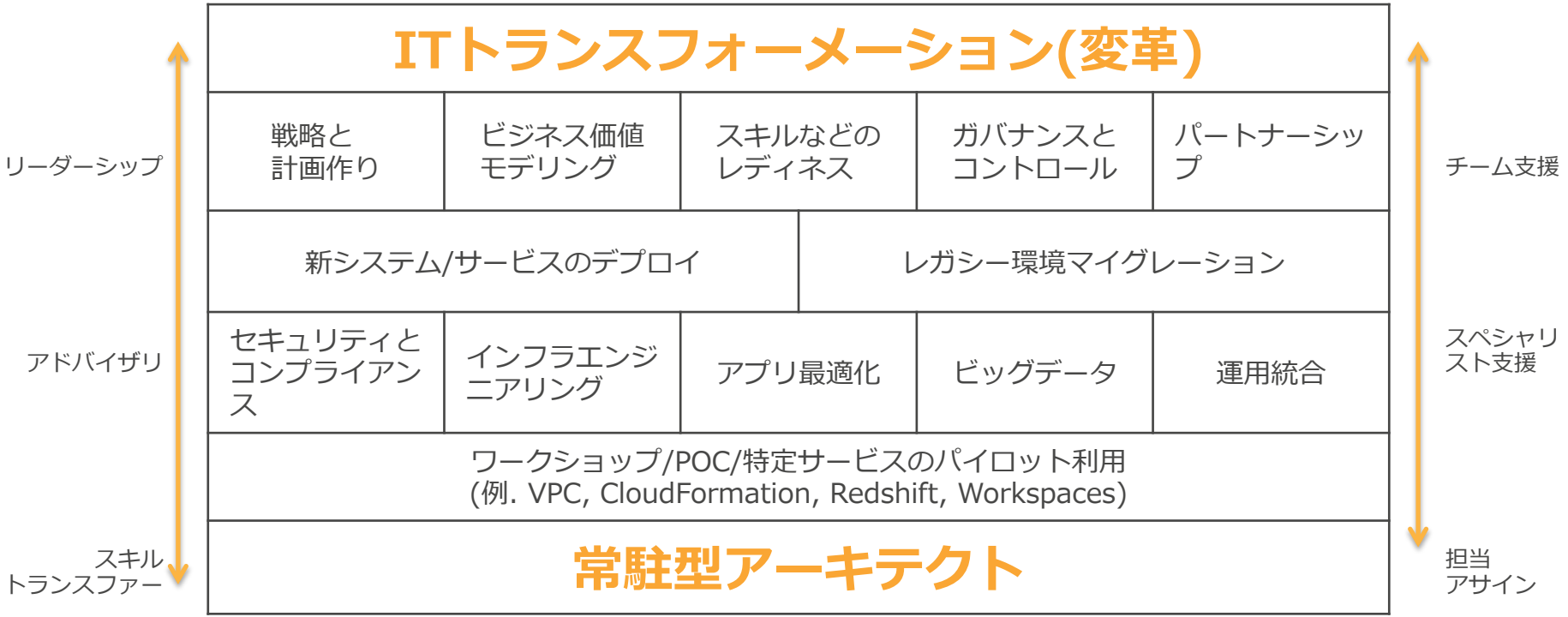
AWSの知識レベルの証明
・ソリューションアーキテクト
・デベロッパー
・SysOps

#3 AWS プロフェッショナルサービス

- お客様のクラウド導入をご支援・加速するための
  有償コンサルティング/アドバイザリーサービス
- エンタープライズ、政府機関、それらのお客様に従事するSI/ISV様
  にご提供
- AWSの技術領域に高度に特化
- プロジェクトベースでご支援。期間は短期～1年以上
- タイムアンドマテリアル型で毎月稼働時間分をご請求



#3 AWS プロフェッショナルサービス



日本通運株式会社様

- プライベートクラウド環境からの更なる進化（コスト効果の最適化・柔軟性の向上）を目指し、パブリッククラウドを活用したサービス展開を検討
- 2013年3月の企画フェーズから、実際の移行におけるQA対応・インフラ設計・検証支援においてAWSのプロフェッショナルサービスを活用
- 2018年までにプライベートクラウドからパブリッククラウドにオールインすることに決定

“クラウドベンダー決定後からサービスインまでの準備期間は約３ヶ月であったが、プロフェッショナルサービスを活用することにより、サービス設計及び、システム導入設計を行う中で日々発生する問題点、課題を早期に解決することができた。
当サービスの利用によって準備期間を短縮でき、計画通りのサービス開始が可能となった。“
日通情報システム株式会社　代表取締役社長　永瀬 裕伸様　より

株式会社NTTドコモ様

- 社内でのAWS利用増加に伴って、ガバナンス確保、リテラシーの向上、開発・運用の効率化が急務に。その対応の1つとして、社内ツールの開発・展開を検討

- ツールの開発におけるQA対応・資料作成・検証支援においてAWSプロフェッショナルサービスを活用

- 短期間で、AWSを利用する場合のガイドラインやセキュリティガバナンスのための基礎ツールを開発し、社内展開を推進

本日のアジェンダ

- AWS Professional Servicesとは
- クラウド導入範囲の選択肢
- All-inの進め方と勘所

クラウドコンピューティングとは？

初期投資が不要

低額な変動価格

実際の使用分
のみ支払い

セルフサービスな
インフラ

スケールアップ、
ダウンが容易

市場投入時間と
アジリティの改善

amazon web services

クラウドコンピューティングとは？

初期投資が不要

セルフサービスな
インフラ

低額な変動価格

スケールアップ、
ダウンが容易

実際の使用分
のみ支払い

市場投入時間と
アジリティの改善

どの特性を重要視するかは
お客様によって異なる

amazon web services

【勘所】クラウド適用範囲と目的を明確化

適用の範囲

全社レベル

部門レベル

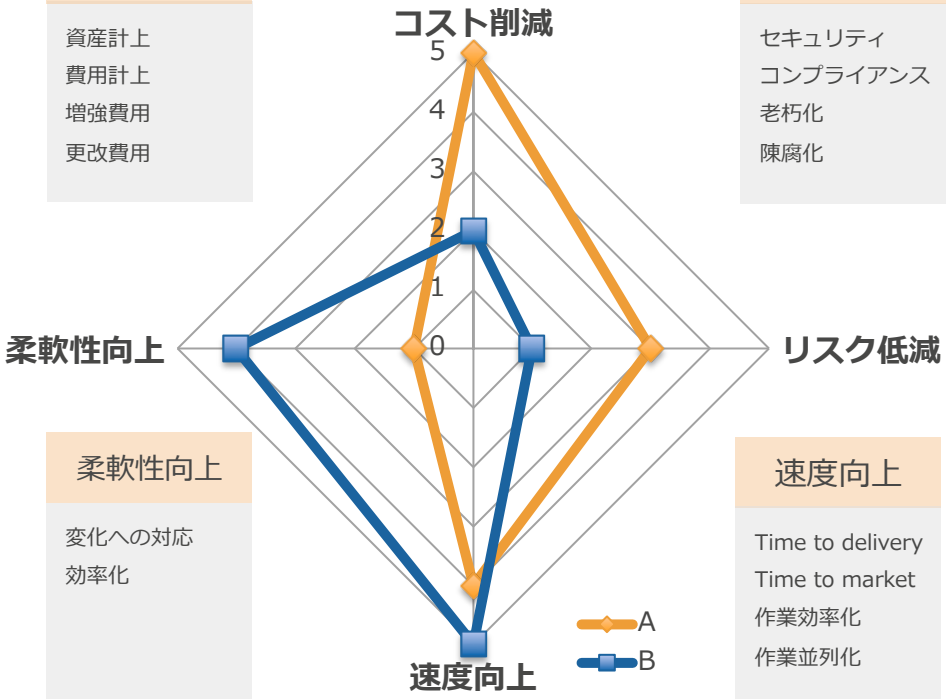
プロジェクトレベル

コスト削減

資産計上
費用計上
増強費用
更改費用

リスク低減

セキュリティ
コンプライアンス
老朽化
陳腐化



柔軟性向上

変化への対応
効率化

速度向上

Time to delivery
Time to market
作業効率化
作業並列化

【勘所】クラウド適用範囲と目的を明確化

適用の範囲

プロジェクトレベル

コスト削減

資産計上
費用計上
増強費用
更改費用

リスク低減

セキュリティ
コンプライアンス
老朽化
陳腐化

変化への対応
効率化

Time to delivery
Time to market
作業効率化
作業並列化

これによってアプローチが変化

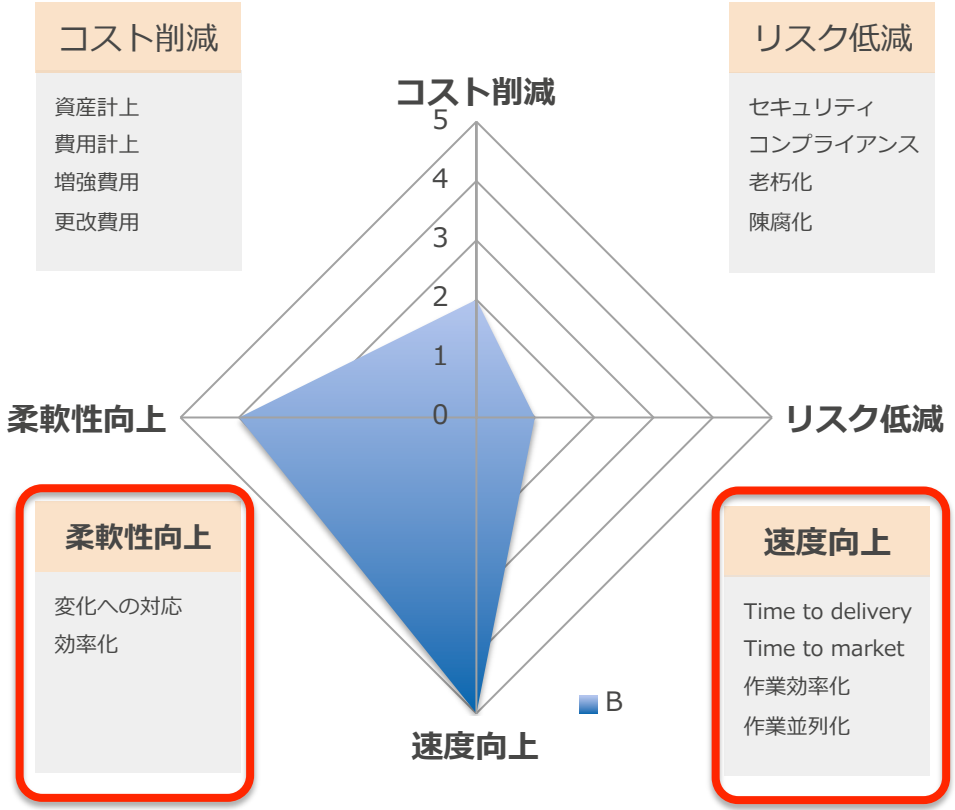
(例) 個別システム最適化

適用の範囲

全社レベル

部門レベル

システムレベル



(例) データセンター移行

適用の範囲

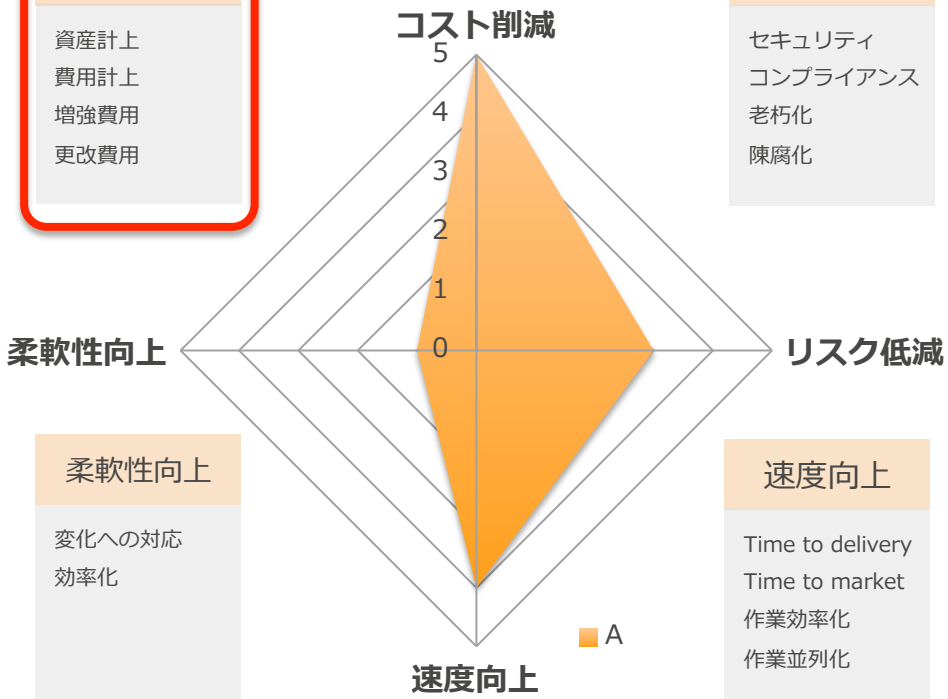
全社レベル

部門レベル

システムレベル

コスト削減

資産計上
費用計上
増強費用
更改費用



リスク低減

セキュリティ
コンプライアンス
老朽化
陳腐化

柔軟性向上

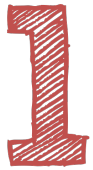
変化への対応
効率化

速度向上

Time to delivery
Time to market
作業効率化
作業並列化

【勘所】どのように利用するかのスコープを決める

実験・検証



開発やテスト

まず開発環境やテスト環境をオフロード

限定範囲での利用

本番環境

新規アプリの構築や既存アプリのマイグレーションマーケティングサイトなど

広い範囲での活用

ミッション
クリティカル

ミッションクリティカルなアプリの新規構築やマイグレーション

社内標準としての利用

オールイン

社内標準

amazon web services

#1 開発・検証環境

Oracle

開発・テスト環境
5年で9割のコスト削減
固定資産管理からの解放

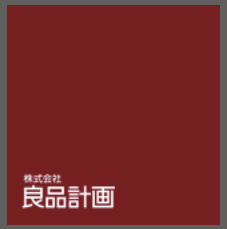
ERP

ERPアプリケーションの開発環境
調達のスピードとコスト削減

HPC

柔軟な開発環境を手に入れ
待ち行列がゼロ
ビジネススピード向上

#2 本番環境での利用



データ解析基盤

Amazon Redshiftで
安価なBI環境を実現し、
顧客のニーズや購買行動を
迅速、正確に把握可能に

コンテンツ配信

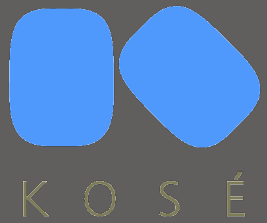
メディアからのトラフィック
流入に対して安定的にコンテ
ンツ配信

25%のレスポンス改善

モバイルメディア

システム構築期間の短縮
運用負荷と費用を削減
規模に合わせた
システム拡張の柔軟性

#3 ミッションクリティカル



顧客情報
管理・検索システム

個人情報を扱うため、セキュリティーが最優先事項

機会損失がないためIT部門が経営に貢献

SAP環境

SAP環境をAWSに移行。プライベートクラウドに比べ50-60%コスト削減

AWSは調べれば調べるほどセキュアな環境との判断

高負荷なバッチ処理

４時間かかっていた処理を20分に短縮
EDI、ワークフロー、人事、WebサイトのAWS化をその後完了

#4 All-in ― 全てのITをクラウドで

Marubeni

LAWSON

500台規模のプライベートクラウドを丸ごとAWSへ

業務システムの全インフラを数年かけてAWSに移行

1000台以上のサーバーを順次クラウドへ移行

5年間で40%のコスト削減

基幹システムをAWSに移行

その他の業務システムも順次AWSに移行

amazon web services

本日のアジェンダ

- AWS Professional Servicesとは
- クラウド導入範囲の選択肢
- All-inの進め方と勘所

プロセスの流れ

戦略 → **分析** → **設計** → **移行** → **運用** → **改善**

- 戦略や方向性の設定
- 現状の調査(アセスメント)と分析
- 必要に応じプラットフォーム評価
- TCOの算出
- 体制の確立、必要に応じてスキルトランスファー
- これら全体計画の評価

- 早い段階でのPoC実施
- サービスカタログ作成
- アーキテクチャの標準化
- 移行方式の類型化
- 開発プロセスの最適化

などを繰り返しながら順次移行を進めていく

- 既存環境との運用統合
- 運用方法の見直しと省力化
- 障害対応の自動化、運用指摘事項のサービスカタログへの反映
- フィードバックループによる継続的な運用改善

戦略や方向性、ゴールの設定

- クラウド移行がうまくいったかを判断するためには明確なゴールと評価のメトリクスが必要
- ゴールには優先順位をつける
- 定期的に測定し続ける

成功の基準	現行	移行後目標	測定方法
コスト(CapEx)	1億円	3000万円	今後2年間でCapExを60%削減できたかどうか
コスト(OpEx)	2000万円	1000万円	スタッフ1人あたりのサーバ管理台数が倍になったかどうか。4つの保守契約を解約できたかどうか
ハードウェア調達効率	7ヶ月で10台	5分で100台	リソースの調達を30倍の速さにできたかどうか
Time to market	9ヶ月	1ヶ月	新規製品・サービスの立ち上げ時間を80%短縮できたかどうか
信頼性	不明	冗長化	ハードウェア関連のサポート問い合わせを40%削減できたかどうか
可用性	不明	最低99.9%の稼働	運用関連のサポート問い合わせを20%削減できたかどうか
柔軟性	固定	柔軟	特定のハードウェアなどにロックインされていないかどうか
新規システムLaunch数	10個の待ち行列	さらに追加で5個	3ヶ月で25個のプロジェクトをLaunchできたかどうか

現状分析

サーバーインフラ	サーバ機種、CPU、アーキテクチャ、メモリ、特殊デバイス有無、負荷状況、利用状況の傾向、調達時期、購入価格、保守費用…
ネットワークインフラ	ネットワーク構成図、帯域、回線キャリア、スイッチ、F/W、ロードバランサ、地理、ブロードキャスト有無、マルチキャスト有無、費用…
ストレージインフラ	ディスク容量、キャッシュ容量、アクセス傾向、IOPS、変化率、バックアップ方式、データ増加頻度、購入費用、保守費用…
OS	OS種別、バージョン、パッチ適用ポリシー…
ライセンスやサポート契約	インストール済みミドルウェア、ライセンス調達元、初期費用、サポート費用、契約年数、バージョンアップ費用…
人	ステークホルダー、ネットワーク管理者・システム管理者・サポート要員・アプリ開発者などの人数やコスト、稼働状況…
個々のアプリケーション	アプリケーションスタック、ユーザ数、パフォーマンス、成長度合い、収益、ビジネス上の重要度、格納データの機密度、初期構築費用、保守費用…
アプリケーションの依存関係	他システムとのデータ連携の有無、ストレージやDB共有の有無、処理の依存関係や前後関係…
コスト	サーバインフラ、ネットワーク、ストレージ、ライセンス、人的コストなど各項目における初期コストやランニングコスト、増加率…

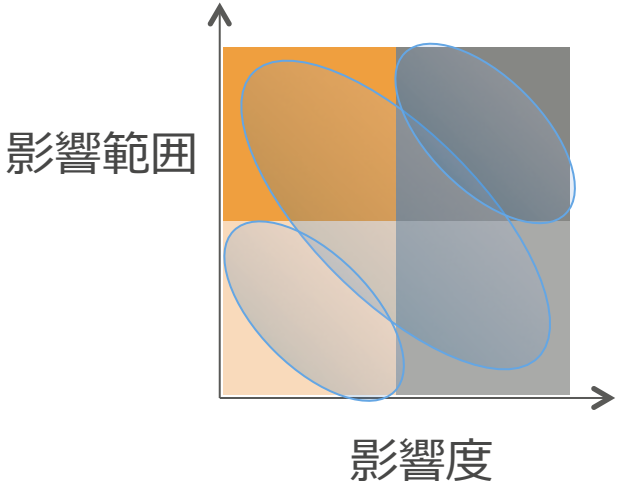
現状分析

サーバーインフラ	サーバ機種、CPU、アーキテクチャ、メモリ、特殊デバイス有無、負荷状況、利用状況の傾向、調達時期、購入価格、保守費用…
ネットワークインフラ	ネットワーク構成図、帯域、回線キャリア、スイッチ、F/W、ロードバランサ、地理、ブロードキャスト有無、マルチキャスト有無、費用…
ストレージインフラ	ディスク容量、キャッシュ容量、アクセス傾向、IOPS、変化率、バックアップ方式、データ増加頻度、購入費用、保守費用…
人	アプリ開発者などの人数やコスト、稼働状況…
個々のアプリケーション	アプリケーションスタック、ユーザ数、パフォーマンス、成長度合い、収益、ビジネス上の重要度、格納データの機密度、初期構築費用、保守費用…
アプリケーションの依存関係	他システムとのデータ連携の有無、ストレージやDB共有の有無、処理の依存関係や前後関係…
コスト	サーバインフラ、ネットワーク、ストレージ、ライセンス、人的コストなど各項目における初期コストやランニングコスト、増加率…

戦略やゴールによってどこまで分析すべきかは変化する

現状分析：移行対象リストと影響度判断

■評価マトリクス例



■評価表例

		アプリケーション			
		A	B	C	…
影響範囲	ユーザ種別	1	1	5	
	ユーザ数	3	2	5	
	アクセス数	3	1	3	
影響度	業務影響	5	2	3	
	原本性	5	1	5	
	得点合計	17	7	21	

影響範囲として利用する評価項目例：

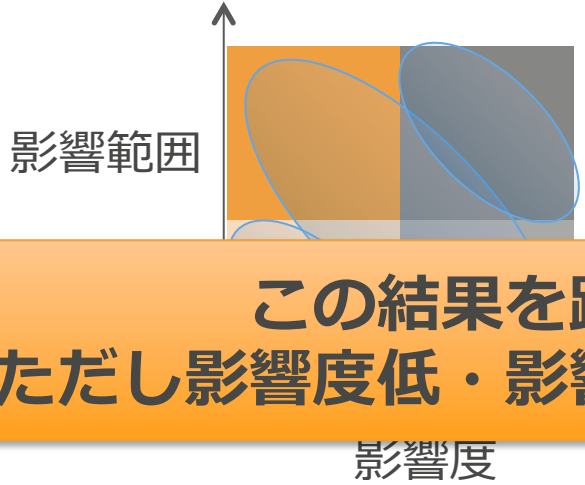
- ユーザ種別（社内、パートナー、顧客）
- ユーザ数（部門内、事業部内、全社、消費者）
- アクセス数（～数十、～数百、～数千、数万～）

影響度として利用する評価項目例：

- 停止時のビジネスインパクト（～円）
- 停止時の業務影響（個人影響、特定業務停止、特定部門麻痺、全業務停止）
- 原本性（取引ログ、マスター、情報系）

現状分析：移行対象リストと影響度判断

■評価マトリクス例



■評価表例

		アプリケーション			
		A	B	C	…
影響範囲	ユーザ種別	1	1	5	
	ユーザ数	3	2	5	
	アクセス数	2	1	3	

この結果を踏まえて移行順序を決める。
ただし影響度低・影響範囲低のものが最初とは限らない

影響範囲として利用する評価項目例：

- ユーザ種別（社内、パートナー、顧客）
- ユーザ数（部門内、事業部内、全社、消費者）
- アクセス数（～数十、～数百、～数千、数万～）

影響度として利用する評価項目例：

- 停止時のビジネスインパクト（～円）
- 停止時の業務影響（個人影響、特定業務停止、特定部門麻痺、全業務停止）
- 原本性（取引ログ、マスター、情報系）

現状分析：移行効果の分類による移行順序決定

- 移行効果と移行容易性の2軸で分類すると、どの順に移行すべきががおおよそ分かる
- 移行効果はクラウド利用の目的にアラインして考える

プラットフォームの評価 (例：SLA)

項目		本番：サービスレベル高	本番：サービスレベル低	検証開発
サービス時間	サービス提供時間	24時間365日	月-土 9:00-22:00	月-金9:00-22:00
	計画停止	1週間前通知で夜間6時間	数日前通知で数時間	随時
可用性	障害時停止許容時間	1-2時間	24時間	24時間
	障害時リカバリーポイント	直前	1日前	1日前
	障害時性能	100%	50%以上	なし
	災害対策RPO	1日前	なし	なし
拡張性	システム追加	n/a	n/a	1時間以内
	リソース増強	1営業日以内	1営業日以内	数分
監視		あり	なし	なし
性能		（別途定義）		
セキュリティ		（別途定義）		

プラットフォームがビジネスに必要な要求を満たせるか評価する

プラットフォームの評価 (例：セキュリティ)

プラットフォームの評価 (例：セキュリティ)

ベストプラクティス

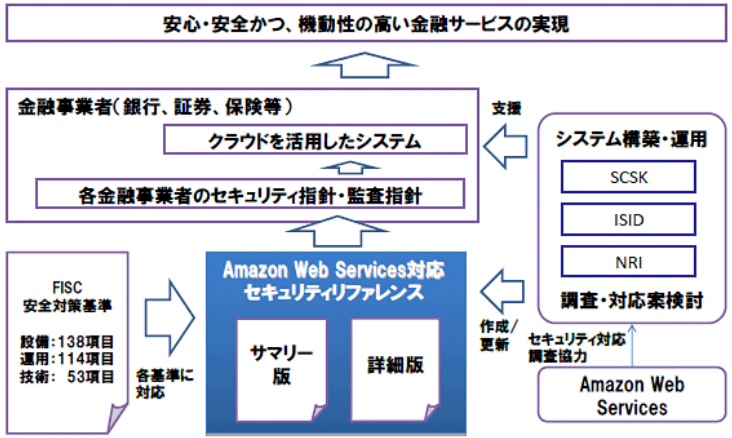
- 一番最初にセキュリティの評価に時間をかけすぎない
- 実際のシステムを例にとって評価する

コイニー様

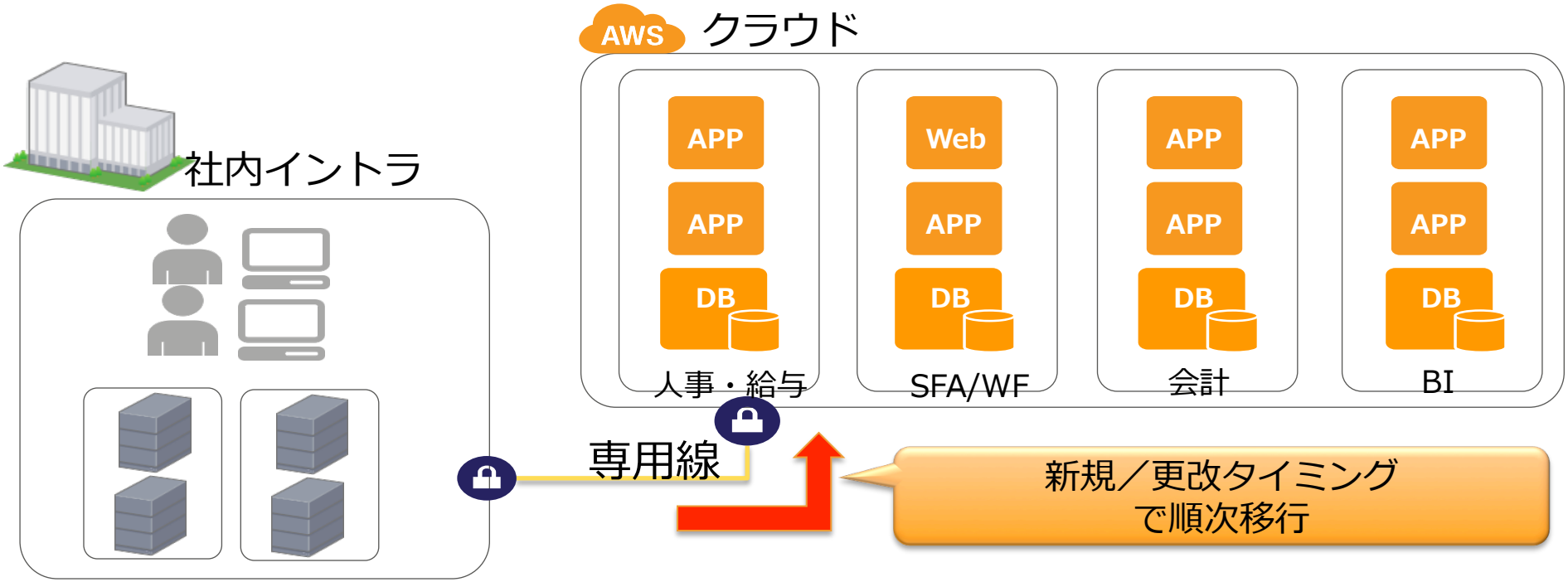
- スマートフォンを使ったクレジットカードでの決済サービスを提供
- “AWSを利用すれば、弊社のような設立１年に満たないスタートアップでも容易にPCI DSS準拠を取得することが可能に”

FISC安全対策基準へのAWSの準拠状況調査

- FISC安全対策基準（第8版追補）へのAWSの準拠状況を調査した資料を一般公開
- システムインテグレーター/パッケージベンダーが7社が共同で調査
- AWSと利用者で責任分担することで、基準を満たせるとの見解
- https://aws.amazon.com/jp/aws-jp-fisclist/



セキュリティ：専用線とVPCによる閉域網の構築

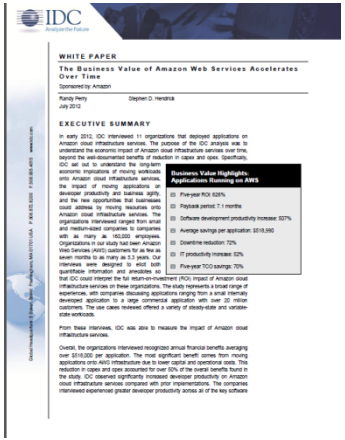


TCO分析

オンプレミスからAWSクラウドへの変更で、5年間で70%のTCO削減

（2012/7 IDC White Paper）

- ✓ AWS利用の企業11社へのヒアリング結果
- ✓ ソフト開発の生産性507%増加
- ✓ アプリケーション辺り平均約5000万円の削減
- ✓ ダウンタイムを72%減少
- ✓ IT全体の生産性を52%増加


IDC

WHITE PAPER

The Business Value of Amazon Web Services Accelerates Over Time

Sponsored by: Amazon

Randy Perry
July 2012

Stephen D. Hendrick

EXECUTIVE SUMMARY

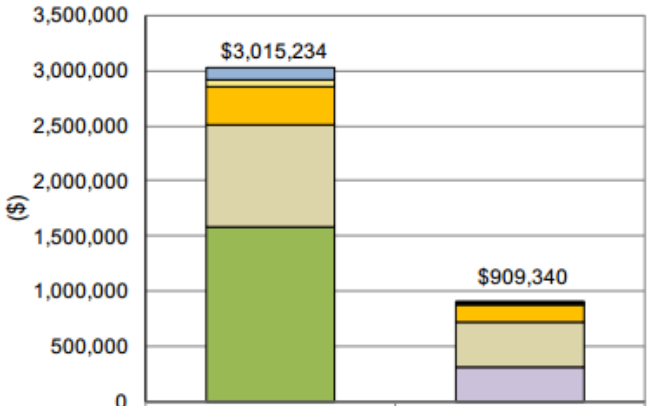
In early 2012, IDC interviewed 11 organizations that deployed applications on Amazon cloud infrastructure services. The purpose of the IDC analysis was to understand the economic impact of Amazon cloud infrastructure services over time, beyond the well-documented benefits of reduction in capex and opex. Specifically, IDC set out to understand the long-term economic implications of moving workloads onto Amazon cloud infrastructure services, the impact of moving applications on developer productivity and business agility, and the new opportunities that businesses could address by moving resources onto Amazon cloud infrastructure services. The organizations interviewed ranged from small and medium-sized companies to companies with as many as 160,000 employees. Organizations in our study had been Amazon Web Services (AWS) customers for as few as seven months to as many as 3.3 years. Our interviews were designed to elicit both quantifiable information and anecdotes so that IDC could interpret the full return-on-investment (ROI) impact of Amazon cloud infrastructure services on these organizations. The study represents a broad range of experiences, with companies discussing applications ranging from a small internally developed application to a large commercial application with over 20 million customers. The use cases reviewed offered a variety of steady-state and variable-state workloads.

From these interviews, IDC was able to measure the impact of Amazon cloud infrastructure services.

Overall, the organizations interviewed recognized annual financial benefits averaging over $516,000 per application. The most significant benefit comes from moving applications onto AWS infrastructure due to lower capital and operational costs. This reduction in capex and opex accounted for over 60% of the overall benefits found in the study. IDC observed significantly increased developer productivity on Amazon cloud infrastructure services compared with prior implementations. The companies interviewed experienced greater developer productivity across all of the key software

Business Value Highlights: Applications Running on AWS

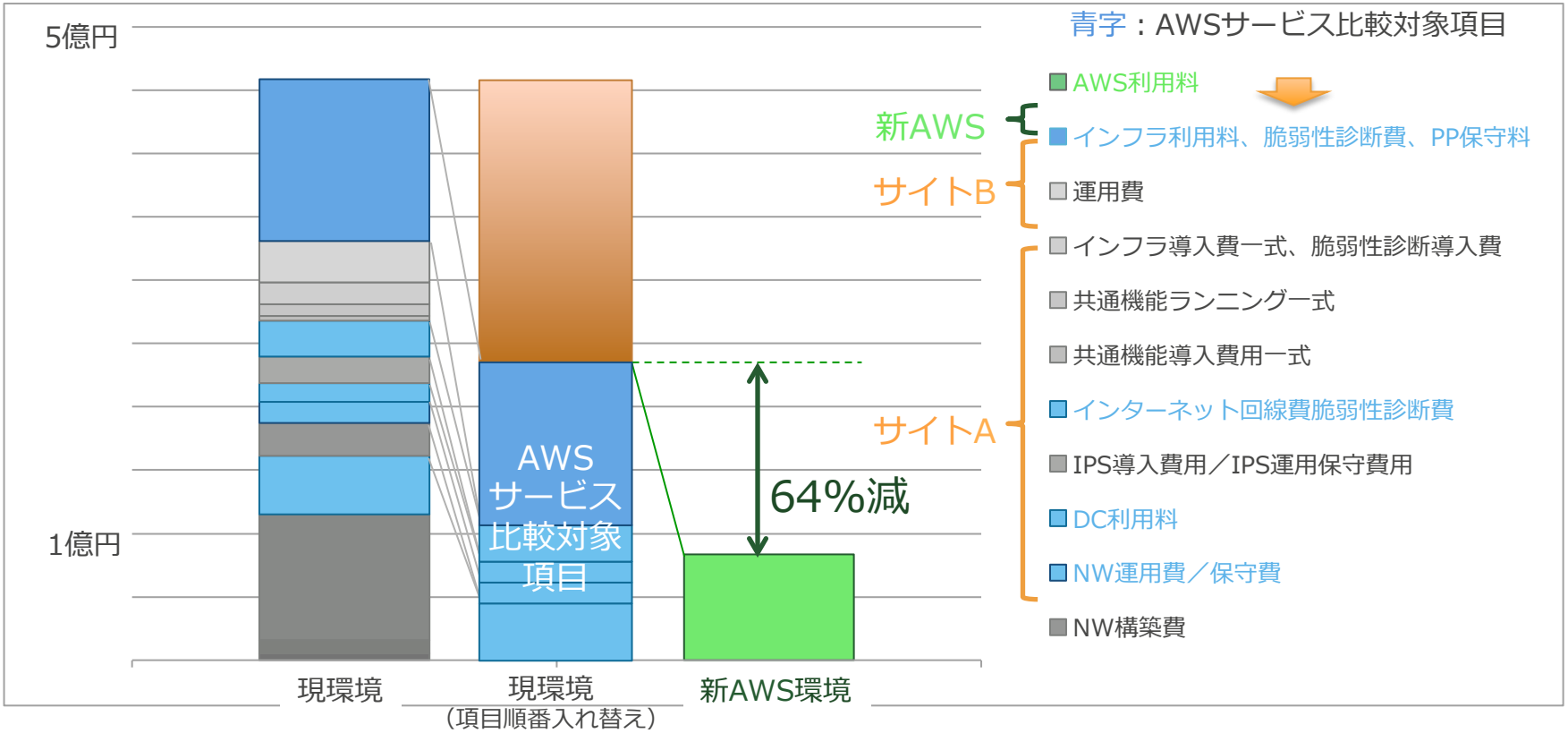
- Five-year ROI: 626%
- Payback period: 7.1 months
- Software development productivity increase: 507%
- Average savings per application: $516,990
- Downtime reduction: 72%
- IT productivity increase: 52%
- Five-year TCO savings: 70%



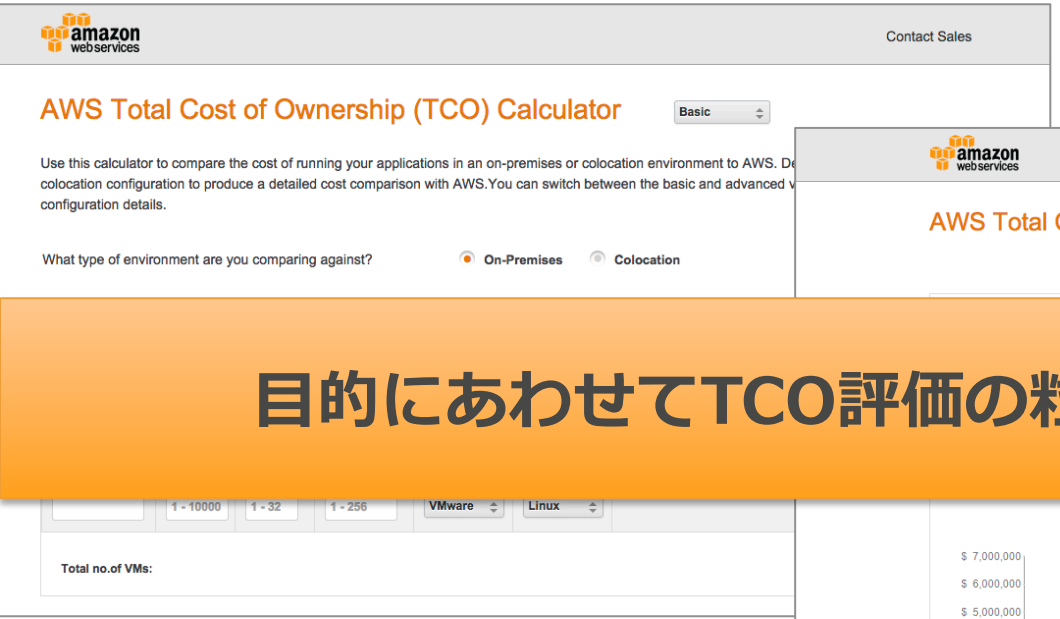
	Other	AWS
Development	$109,638	$20,714
Deployment	$54,432	$11,664
Application management	$346,920	$166,208
Infrastructure management	$920,664	$408,962
Infrastructure	$1,583,579	$-
AWS costs	$-	$301,792

amazon web services

TCOの評価（実例）



TCOの評価



目的にあわせてTCO評価の粒度は変えた方が良い

http://awstcocalculator.com/

体制


CREATIVITY
AMBITION
ENEGRGY
MOTIVATION
LEADERSHIP
STRATEGY
VISION
MANAGEMENT

体制づくりのベストプラクティス

- ナレッジを自社に蓄積できるようにする
  - 必要なスキルセットのベースラインを確保
  - システムのブラックボックス化を防ぎ、自社でITのコントロールを
- 各システムの移行や開発でAWSパートナーを活用

日本のコンサルティングパートナー

日本のテクノロジーパートナー

PoCやパイロットプロジェクト

実際に触ってみる	各AWSのサービスを触ってみることで、机上評価ではなく、実体験として、メリットや自社での適合性を確認する
パイロットプロジェクト実施	実際にAWS上に小規模なアプリを構築し評価を行う
既存アプリをクラウドで動かす	既存アプリをクラウド上にポーティングし既存との違いを明らかにする

マネジメントコンソールの使い方　性能　セキュリティ機能　ネットワーク　BYOL　プロビジョニング時間　バックアップ&リカバリ

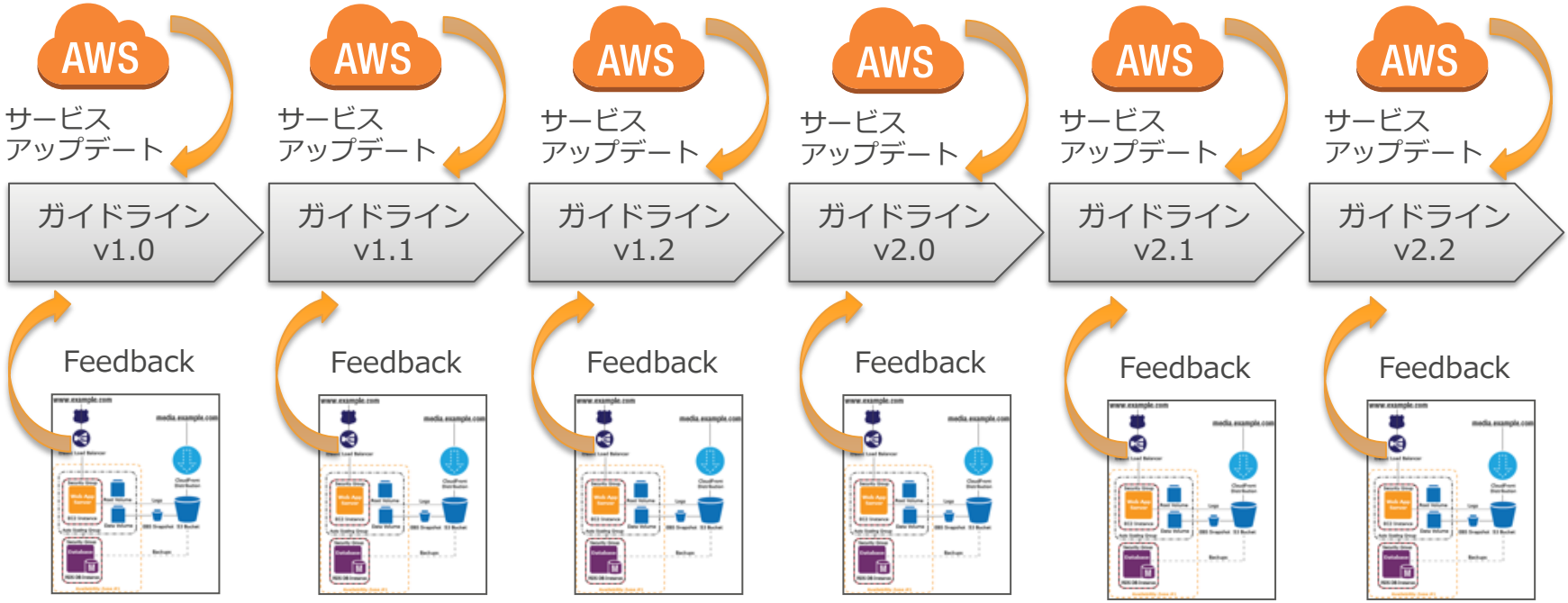
アーキテクチャ

AWSの各種サービス

PoCによって多くの疑問は解決可能
評価したい項目に優先順位をつけて実際に試す

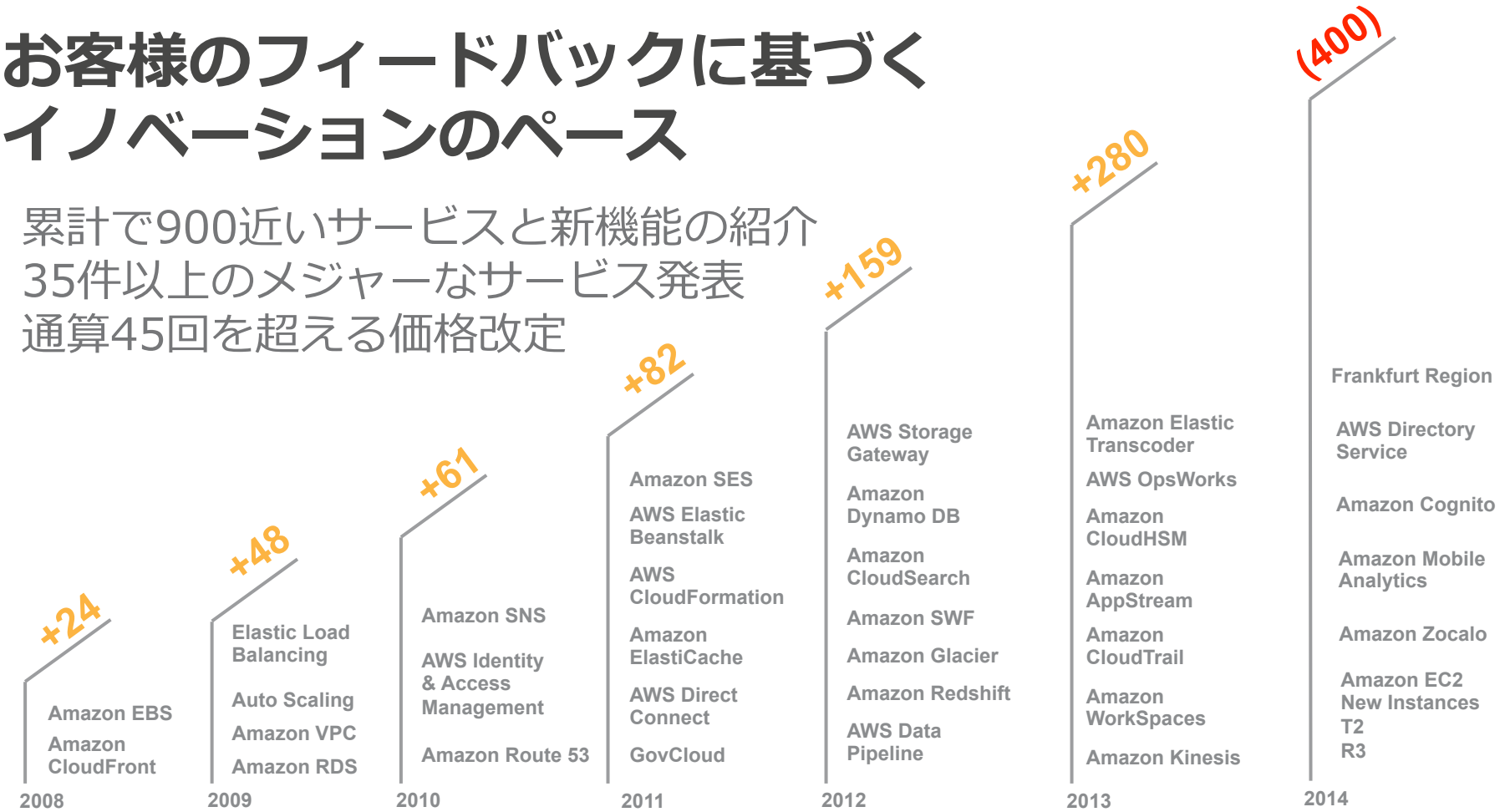
標準化やガイドライン

- クラウド移行を加速させるとともにリスクを低減するために標準化やガイドラインを整備する



お客様のフィードバックに基づくイノベーションのペース

- 累計で900近いサービスと新機能の紹介
- 35件以上のメジャーなサービス発表
- 通算45回を超える価格改定

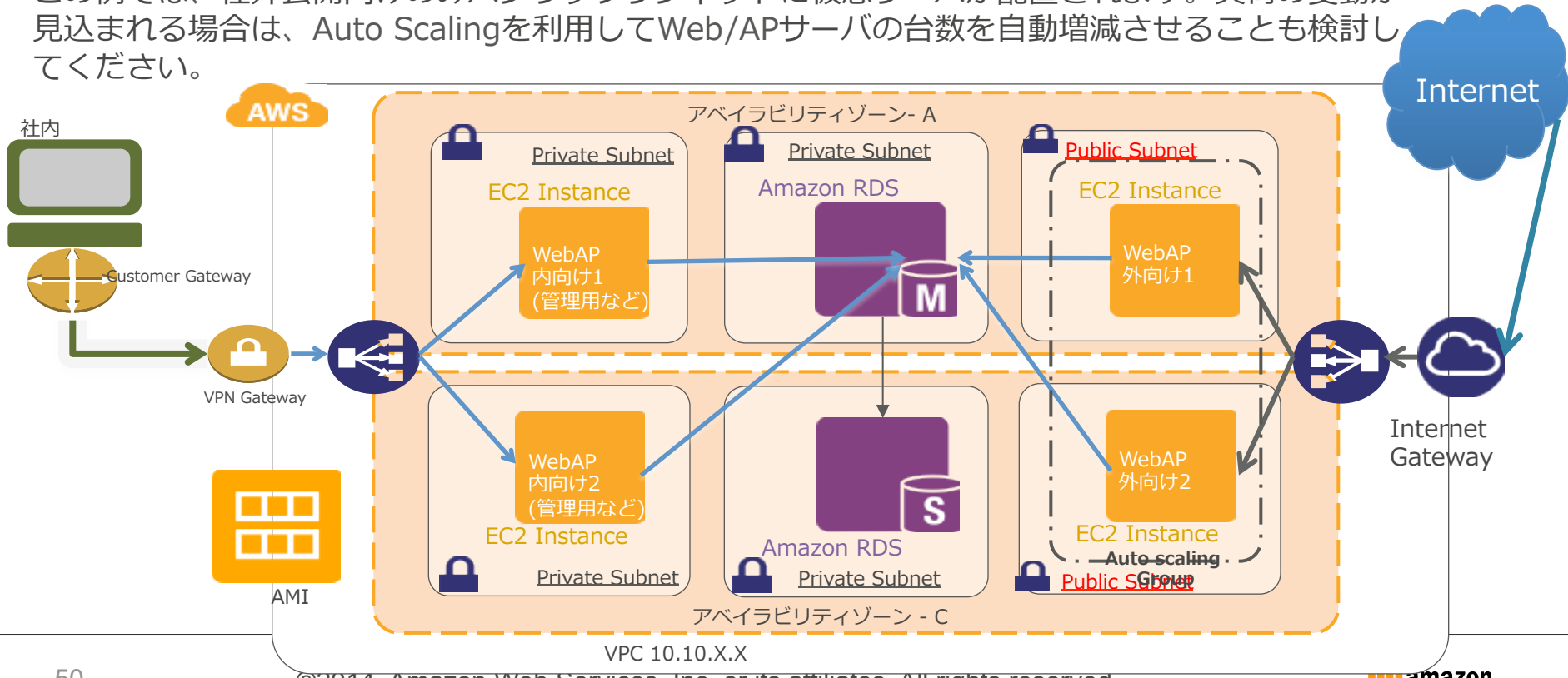


標準化ガイドラインのベストプラクティス

- 定期的な更新を続ける
- 利用者が誰なのかによって記述範囲やボリュームは変わる
- CloudFormationなどで自動的に実現できるように

標準化ガイドライン (例)

高可用性モデルを利用した社外公開型Webアプリケーションのアーキテクチャー例です。
この例では、社外公開向けのみパブリックサブネットに仮想サーバが配置されます。負荷の変動が見込まれる場合は、Auto Scalingを利用してWeb/APサーバの台数を自動増減させることも検討してください。



標準化ガイドライン (例)

故障のための設計

- ❑ あらゆるものはいつでも故障する前提で設計。単一障害点を避ける
- ❑ サーバコピーを取得しいつでも起動可能に
- ❑ スナップショットでのバックアップ
- ❑ 障害時の自動復旧のためにAuto Scalingを活用
- ❑ 複数アベイラビリティゾーンによる冗長化

疎結合なコンポーネント

- ❑ コンポーネントを独立させ、各コンポーネントはブラックボックスとして設計
- ❑ コンポーネント間を疎結合に
- ❑ アプリケーションをステートレスに。この実現のためにELBを活用し、スティッキーセッションを避ける
- ❑ セッションデータは各サーバではなくデータベース(RDS)に保存

弾力性の実装

- ❑ コンポーネントの健全性や配置場所を決めつけない
- ❑ いつでもリブート可能になるように設計
- ❑ 動的なコンフィグレーション(起動時に自動でアプリケーションやライブラリ等をインストールする)の活用
- ❑ 設定済みAMIの活用による時間短縮

全レイヤでセキュリティ担保

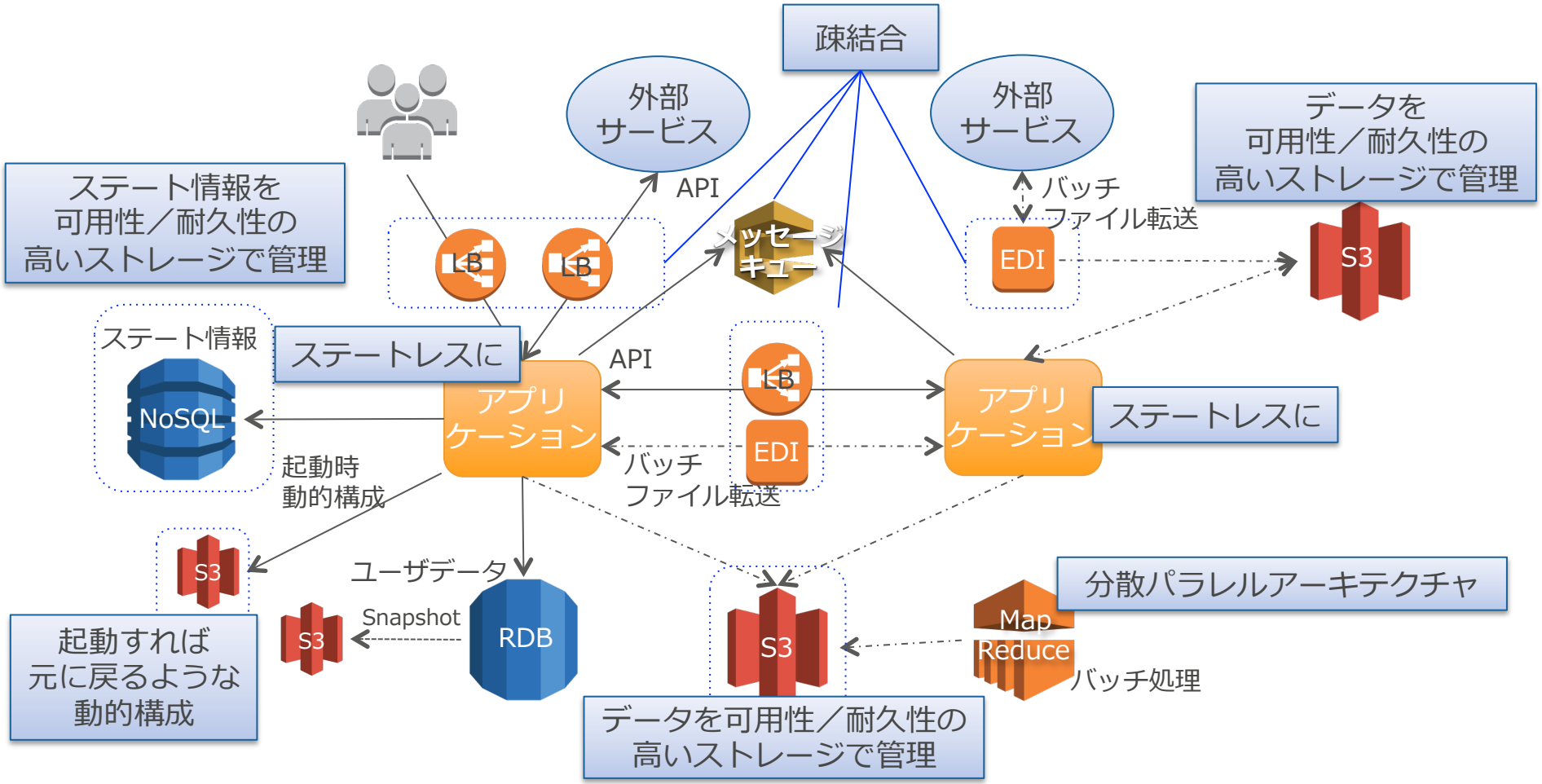
- ❑ AWSのセキュリティは責任分担モデル。OS以上の層やセキュリティグループの設定、データ暗号化、秘密鍵の管理等は利用者側の責任
- ❑ OSの層とアプリの層、ネットワークの層など各所でセキュリティを担保すること。必要に応じてアプリケーションのペネトレーション試験も推奨

並列処理の実装

- ❑ AWSの特性は時間単位でリソースを使用可能な点。これを活かしてアプリケーションやバッチ処理の並列化、マルチスレッド化を検討する
- ❑ ELBやAutoScalingを活用して並列処理、分散処理を実装

異なるストレージの使い分け

- ❑ EBS(仮想マシンで利用するディスク)や、データベース、S3(インターネットストレージ)等データを保存するストレージには多種多様なものがある
- ❑ 読み書きの特性(読み書きの比率や頻度)や、データ喪失の可能性、ステートレスの実現可否を踏まえて適切なストレージを選択する



標準化ガイドラインをセルフチェックする

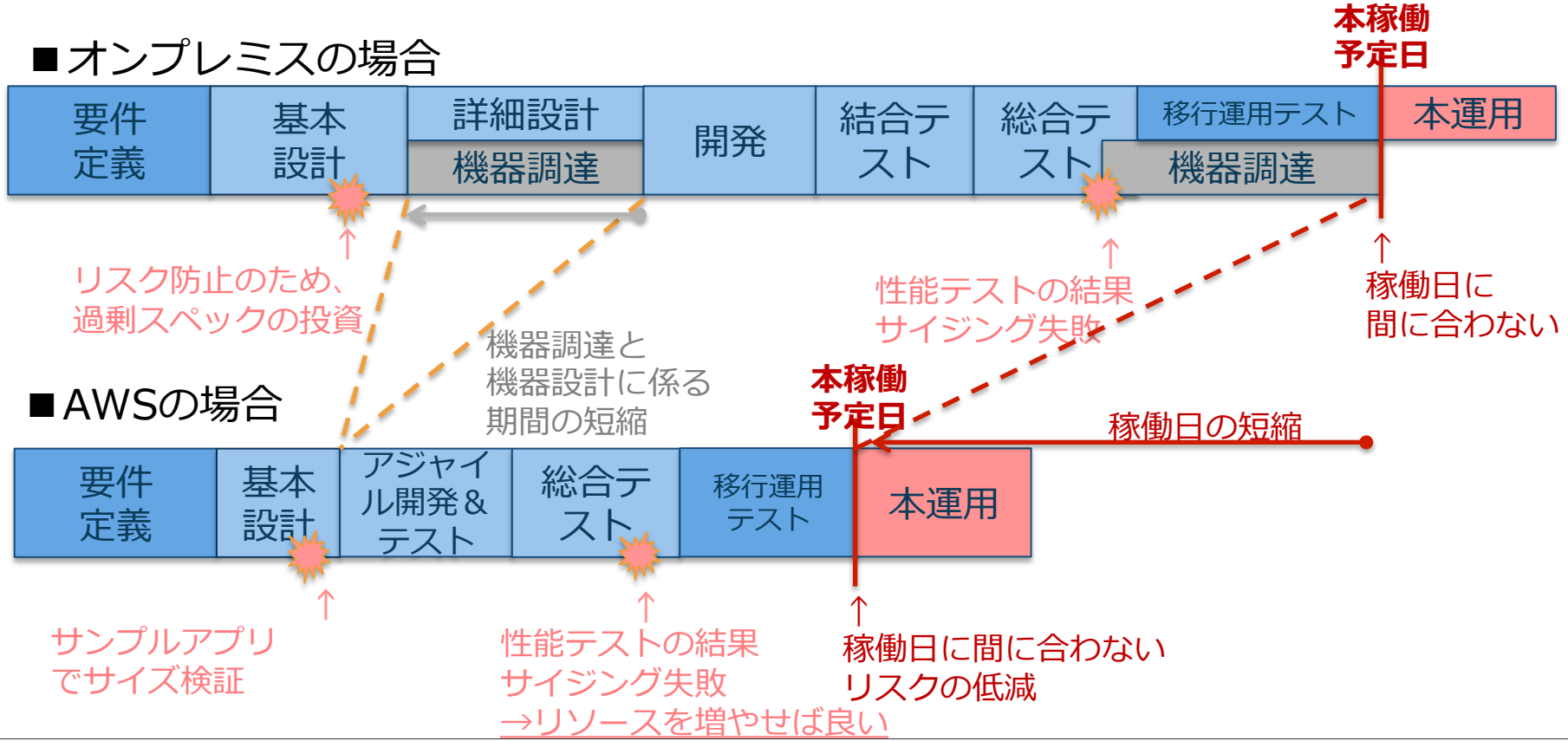
EC2	MUST	インスタンスにNameタグを設定して各インスタンスの役割が明確になっているか	・情報資産管理の不備
	MUST	適切なインスタンスタイプの選択ができているか。 キャパシティの測り方が適切に検討されているか。	・必要以上の課金の発生 ・資源の有効利用への影響
	MUST	PIOPSを利用しているときはEBS最適化が可能なインスタンスを選択しているか	・パフォーマンスに対する影響
	BETTER	自身で別タグを設定して各インスタンスの役割を明確にできているか。Key:Env Value:(Develop|Production|staging)など	・情報資産管理の不備 ・課金情報の集計・自動化等の仕組み作りに対する影響
	MUST	適切なIPアドレッシングが行われているか。不必要なEIPがアサインされていないか等	・情報資産管理の不備 ・適切でないネットワーク設計による障害の発生やセキュリティインシデントの発生
	SHOULD	本番用インスタンスにおいてTermination Protectionを有効にしているか	・オペレーションのミスによる本番環境の毀損
RDS	MUST	ローカルのファイルシステムアクセスしなければいけない要件があるか	・設計の変更 ・RDS以外のソリューションの選択
	MUST	適切なIOPSを設定しているか	・パフォーマンスに対する影響
	MUST	必要となるキャッシュ機能についての検討はされているか	・パフォーマンスに対する影響
DynamoDB	SHOULD	スループットに関する要件は確認できているか	・パフォーマンスに対する影響
	SHOULD	どのようなインデックスが必要になるか確認ができているか	・パフォーマンスに対する影響
	SHOULD	フルスキャンを多用する設計にしていないか	・パフォーマンスに対する影響
Redshift	MUST	DataのロードとSyncのプロセスは確認できているか	・設計の変更
	MUST	BIアプリケーションのコンパチビリティは確認をしているか	・設計の変更
	MUST	複数のアプリケーションで利用する際にコネクションの制限が問題にならないか確認をしているか	・上限緩和の必要性

ガイドラインや過去の経験を踏まえて
ベストプラクティスを蓄積しセルフチェックする

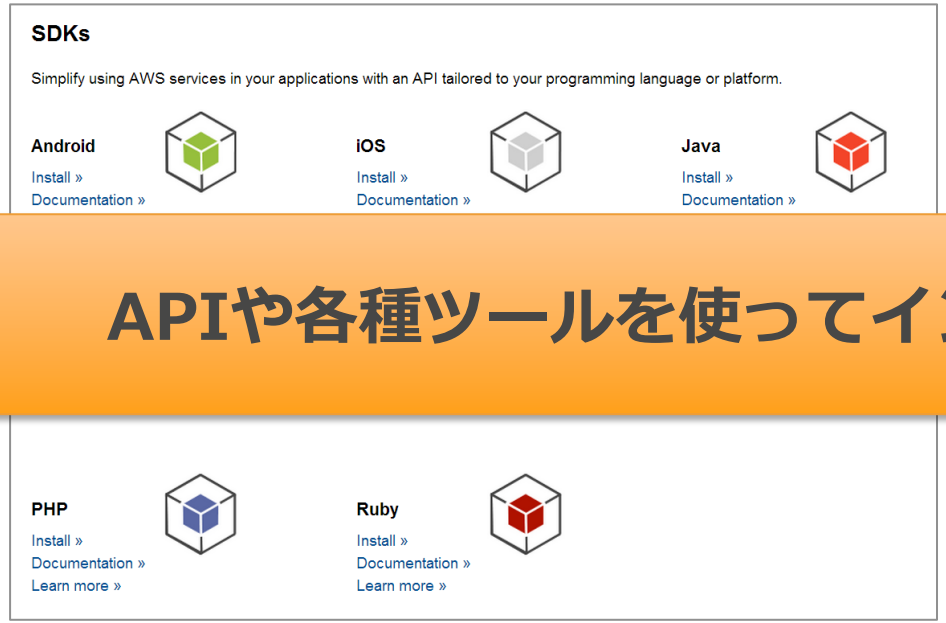
個別のプロジェクト計画もクラウドに最適化

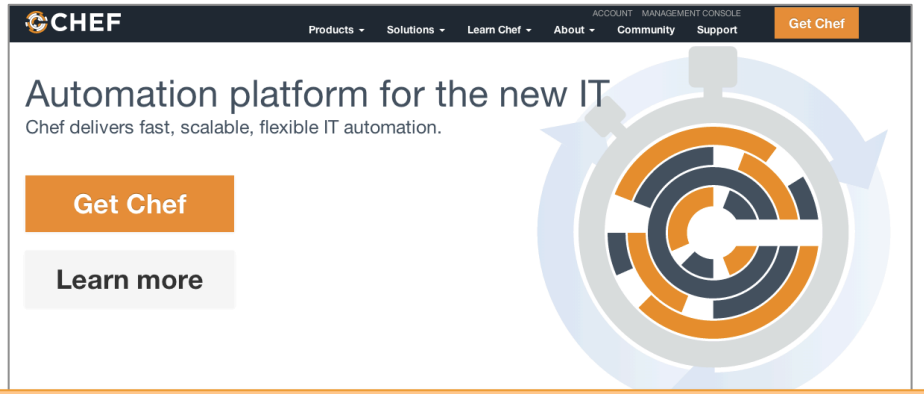
- 要件定義フェーズのキャパシティ計画に時間をかけるよりも、あとフェーズのパフォーマンス（性能）テストに時間をかけてキャパシティを決定していったほうがよい（パフォーマンステスト後サイズのUP/DOWNが可能）
- コア数課金のソフトウェアの購入タイミングは、パフォーマンステスト後サイジングが決まってから（事前に必要な場合は少なめに買っておく）
- プロジェクト初期段階でパフォーマンス上の課題がある場合は、一時テスト環境を作って実機を使ってのパフォーマンステストを実施する
- その他、プロジェクト期間中に出る技術的課題や新しい要求について、お試し環境や本番コピー環境を作って実機での確認を行う
- 他システム／プロジェクトのOS環境のコピー（バックアップ）を積極的に使う（構築期間の短縮かつ安定稼働環境を手に入れられる）
- 検証環境をアプリケーションバージョンごとに複数持つことができる（検証するときだけ立ち上げておけばよいので複数検証環境をもってもコストがかからない）
- 最終テスト済みの検証環境を本番環境に昇格させるかたちで切り替えてのサービスインが可能

開発プロセスの最適化

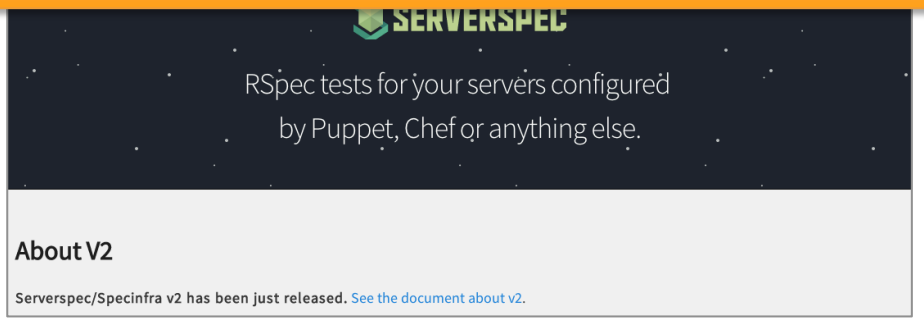


構築





APIや各種ツールを使ってインフラ構築の手作業を減らす



自動化の効果

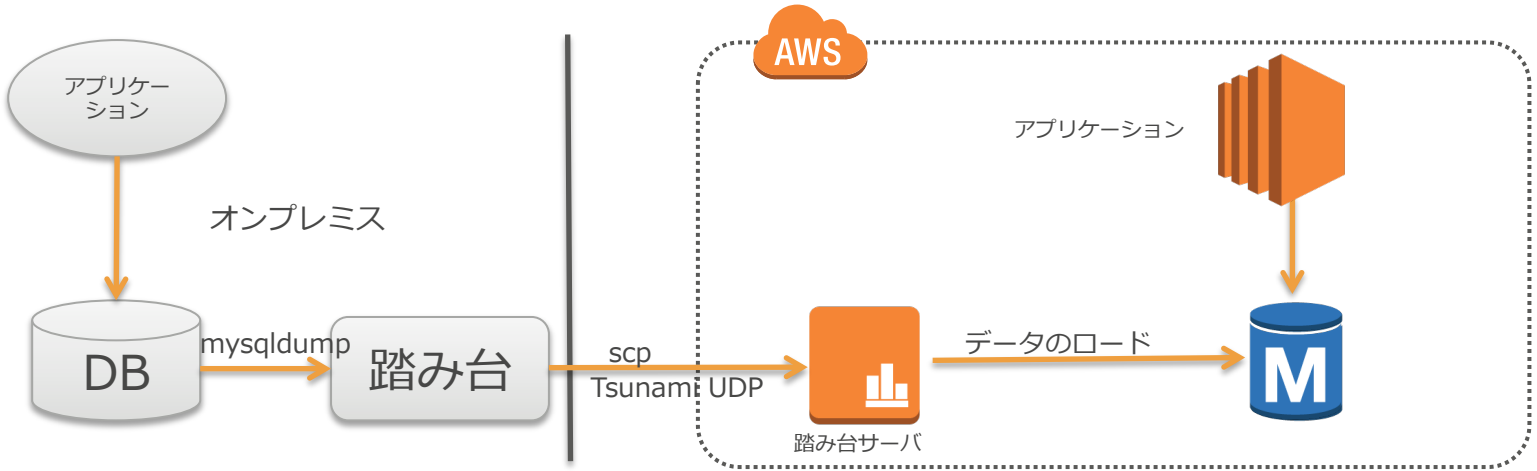
- 頻繁な変化を人手で行うことは限界がある
- いままでは手作業に多くの時間を
- テスト自動化、インフラ構築自動化へ
- 本質的に価値を生む箇所に時間を使う
- これは開発だけでなく運用も同じ
- 品質を作りこみ続ける

amazon web services

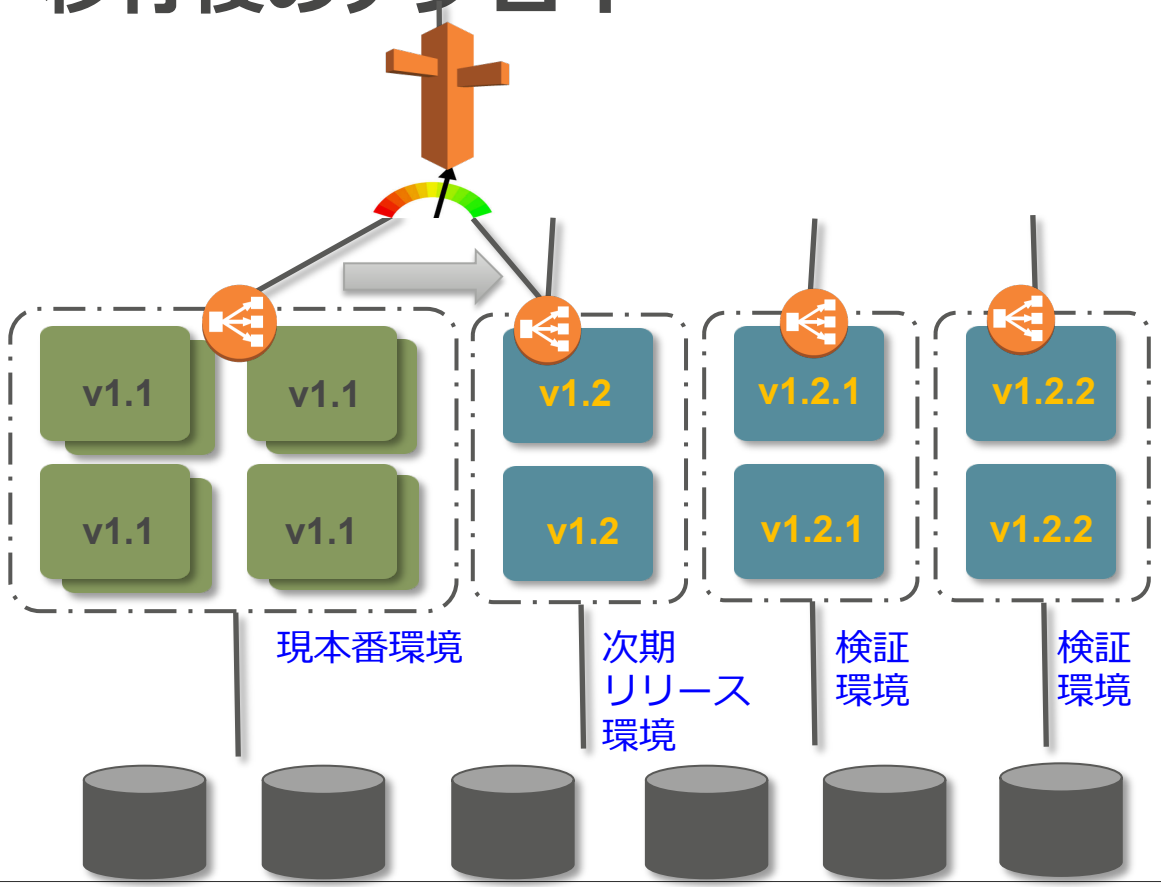
移行

多くの場合、データベースや各種アセットデータの移行が課題になる
PoCまたはプロジェクト初期段階で移行方式について検討を行うと良い

- データ量
- データ変化率
- 停止許容時間
- 回線速度
- 頻度

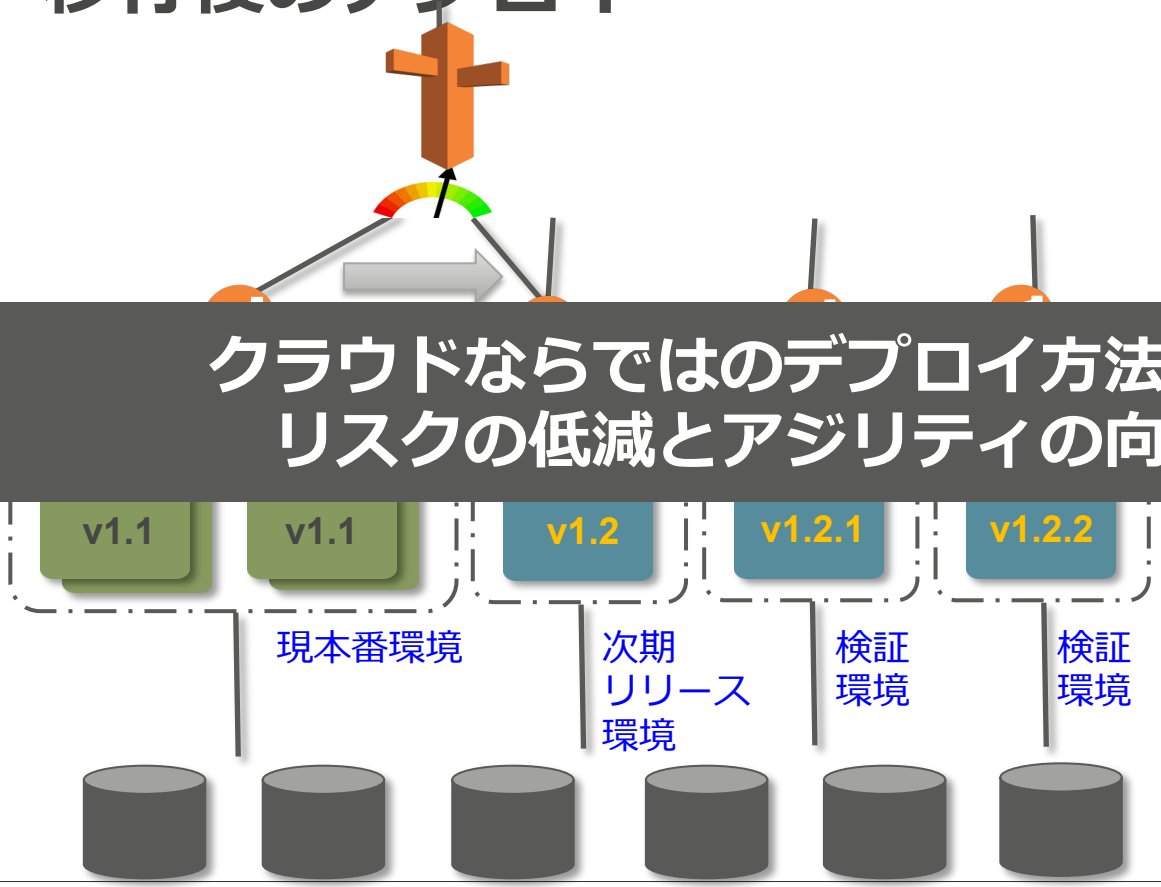


移行後のデプロイ



- アプリケーションバージョンごとに検証環境を作る
- 各検証環境は利用するときだけ起動する
- 次期リリース検証環境の最終テスト後、本場環境に昇格させて切り替える
- これらの切り替えを全てAPIの組み合わせで可能

移行後のデプロイ

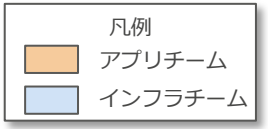


- アプリケーションバージョンごとに検証環境を作る

クラウドならではのデプロイ方法の活用によって
リスクの低減とアジリティの向上を実現する

せて切り替える
- これらの切り替えを全てAPIの組み合わせで可能

運用



運用／メンテ、監視／監査の分界点									
自動再起動	起動停止	バックアップ	ログ管理	インベントリ管理	監視	サービス監視	課金管理	セキュリティ管理	インシデント管理
自動再起動設計	ミドルウェア起動停止スクリプト開発	バックアップ設計	Fluentd設計構築	EC2インスタンス棚卸	監視設計	サービス監視設計	課金管理設計	セキュリティ管理監視設計	監視アラート設定
		スクリプト開発	キーピング開発		ト開発			ティ監査	ト対応
		Snapshot管理	ミドルウェアログ解析設計					OSアカウント管理	
		OSバックアップ取得（AMI作成）						OSパスワード変更	
								Webアクセスセキュリティログ監視設計	
								AWSログ監視設計	

クラウド化に伴い従来と責任分界点が変えた方がよい点があるかどうかを検討する

開発と運用の関係


Dev
オーバーヘッド
価値を生み出すのに使う時間
ゴールの共有
お互いの協力
ツールの活用
によって価値を生み出す時間を増やし、時間効率も改善
Ops
オーバーヘッド
価値を生み出すのに使う時間

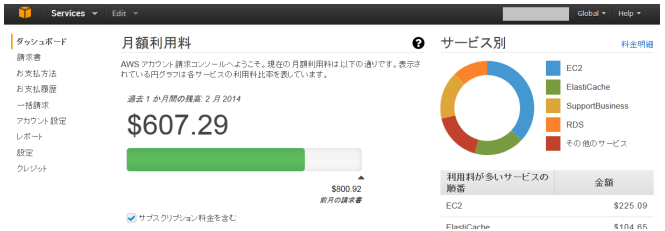
改善

- Just in Time
- かんばん
- ムダ
- 平準化
- アンドン
- ポカヨケ
- 自働化
- 改善
- 見える化

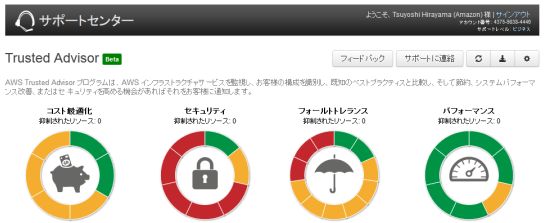
- 作りすぎのムダ
- 手待ちのムダ
- 運搬のムダ
- 加工のムダ
- 在庫のムダ
- 動作のムダ
- 不良をつくるムダ

改善

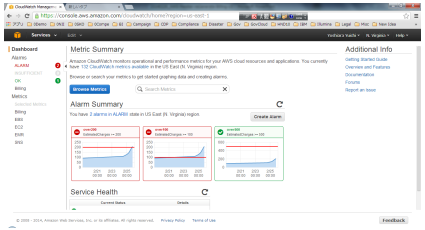
Billing Consoleで概要把握



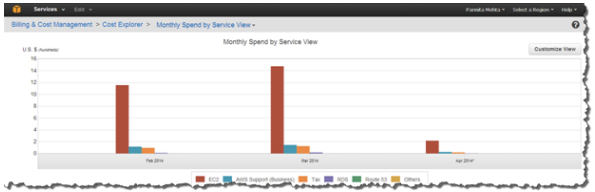
Trusted Advisorで コスト削減アドバイス



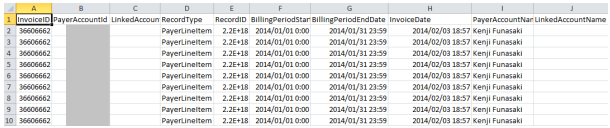
Billing Alertで 課金閾値アラート



Cost Explorerで オンラインでのコスト分析



Detailed Billing Reportで ローカルでのコスト分析



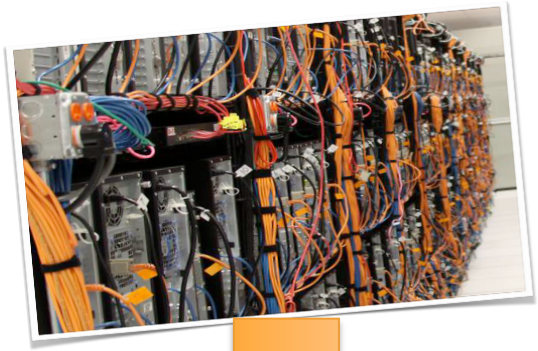
	A	B	C	D	E	F	G	H	I	J
1	InvoiceID	PayerAccountId	LinkedAccoun	RecordType	RecordID	BillingPeriodStar	BillingPeriodEndDate	InvoiceDate	PayerAccountNar	LinkedAccountName
2	36606662			PayerLineItem	2.2E+18	2014/01/01 0:00	2014/01/31 23:59	2014/02/03 18:57	Kenji Funasaki	
3	36606662			PayerLineItem	2.2E+18	2014/01/01 0:00	2014/01/31 23:59	2014/02/03 18:57	Kenji Funasaki	
4	36606662			PayerLineItem	2.2E+18	2014/01/01 0:00	2014/01/31 23:59	2014/02/03 18:57	Kenji Funasaki	
5	36606662			PayerLineItem	2.2E+18	2014/01/01 0:00	2014/01/31 23:59	2014/02/03 18:57	Kenji Funasaki	
6	36606662			PayerLineItem	2.2E+18	2014/01/01 0:00	2014/01/31 23:59	2014/02/03 18:57	Kenji Funasaki	
7	36606662			PayerLineItem	2.2E+18	2014/01/01 0:00	2014/01/31 23:59	2014/02/03 18:57	Kenji Funasaki	
8	36606662			PayerLineItem	2.2E+18	2014/01/01 0:00	2014/01/31 23:59	2014/02/03 18:57	Kenji Funasaki	
9	36606662			PayerLineItem	2.2E+18	2014/01/01 0:00	2014/01/31 23:59	2014/02/03 18:57	Kenji Funasaki	
10	36606662			PayerLineItem	2.2E+18	2014/01/01 0:00	2014/01/31 23:59	2014/02/03 18:57	Kenji Funasaki	

定期的にコスト状況を確認し、
ムダがないかもっと良い方法がないかを確認する

SUNCORP

- オーストラリアの金融グループ
  - SUNCORP BANK: オーストラリア第五位の銀行
  - SUNCORP Insurance: オーストラリア最大の保険契約数
  - その他14の金融ブランド
- 900万人の顧客、1万5千人の従業員
- ミッションクリティカルなものも合わせて、2000のアプリケーションを保有

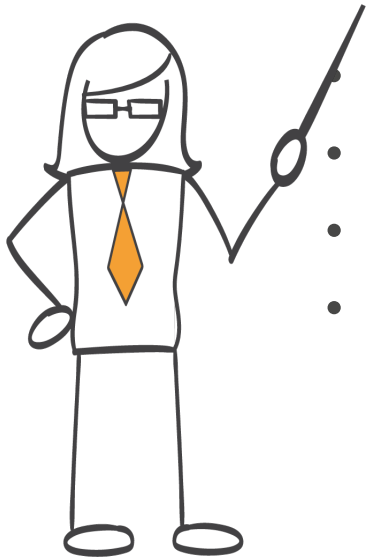
- **1ヶ月目**
  - 新技術調査
  - ビジネス継続性の検討
  - 広範囲の認可
- **2ヶ月目**
  - リスク、セキュリティ、リーガル面の確認
  - ガバナンスの作成
  - 規制各局とのブリーフィング
- **3ヶ月目**
  - オンラインアプリケーション、BIなどからマイグレーション開始
- **18ヶ月目**
  - ミッションクリティカルなものも含め、2000ある全てのアプリケーションをAWSに移行



amazon web services™

amazon web services

まとめ



- 組織におけるクラウド導入の範囲と目的を整理する
- クラウド移行では現状分析、計画づくり、体制確立、PoC、標準化が肝となる
- クラウドにあった開発の進め方があるはず
- データ移行は初期の段階から検討する
- 開発と運用の役割分担を考えなおし、アジリティを高める
- 継続的に改善しつづける

amazon web services