aws SUMMIT

3-H1-1-16

AWS の 運用入門

能仁 信亮
ソリューションアーキテクト アマゾンウェブサービスジャパン株式会社

自己紹介

能仁 信亮(のうにん しんりょう)

エンタープライズ ソリューション本部
ソリューション アーキテクト

普段の仕事
金融機関のお客様へクラウド活用のご支援をさせていただいております

好きなAWSのサービス
S3, Service Catalog

本日、お持ち帰り頂きたい内容

AWSの運用に関連する
サービスの概要

AWSの運用の
ベストプラクティス

AWSの運用に関する俯瞰的な知識

このセッションのアジェンダ

AWSの運用のベストプラクティス
AWSの運用に関連するサービスの概要

責任共有モデル

お客様自身で情報資産
に対する統制が可能
(Security “IN” the Cloud)

お客様のアプリケーション・コンテンツ

ネットワーク
サーバー
セキュリティ

インベントリ
・構成管理

アクセス
コントロール

データ
セキュリティ

AWSがクラウドの
セキュリティを統制
(Security "OF" the Cloud)

aws

AWS 基本サービス

コンピュート

ストレージ

データベース

ネットワーク

AWS
グローバル
インフラストラクチャ

アベイラビリティ
ゾーン

リージョン

エッジ
ロケーション

運用のプロセスを支えるサービス群

1 リソースの
プロビジョニング
- AWS CloudFormation
- AWS Service Catalog

2
構成管理
- AWS Systems Manager

3
監視
- Amazon CloudWatch

4
ガバナンスと
コンプライアンス
- AWS CloudTrail
- AWS Config

5
リソースの最適化
- AWS Trusted Advisor

このセッションのアジェンダ

AWSの運用のベストプラクティス
AWSの運用に関連するサービスの概要

AWS Well-Architected フレームワーク

セキュリティ

信頼性

パフォーマンス

コストの最適化

運用性

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

AWS Well-Architected フレームワーク

セキュリティ

信頼性

パフォーマンス

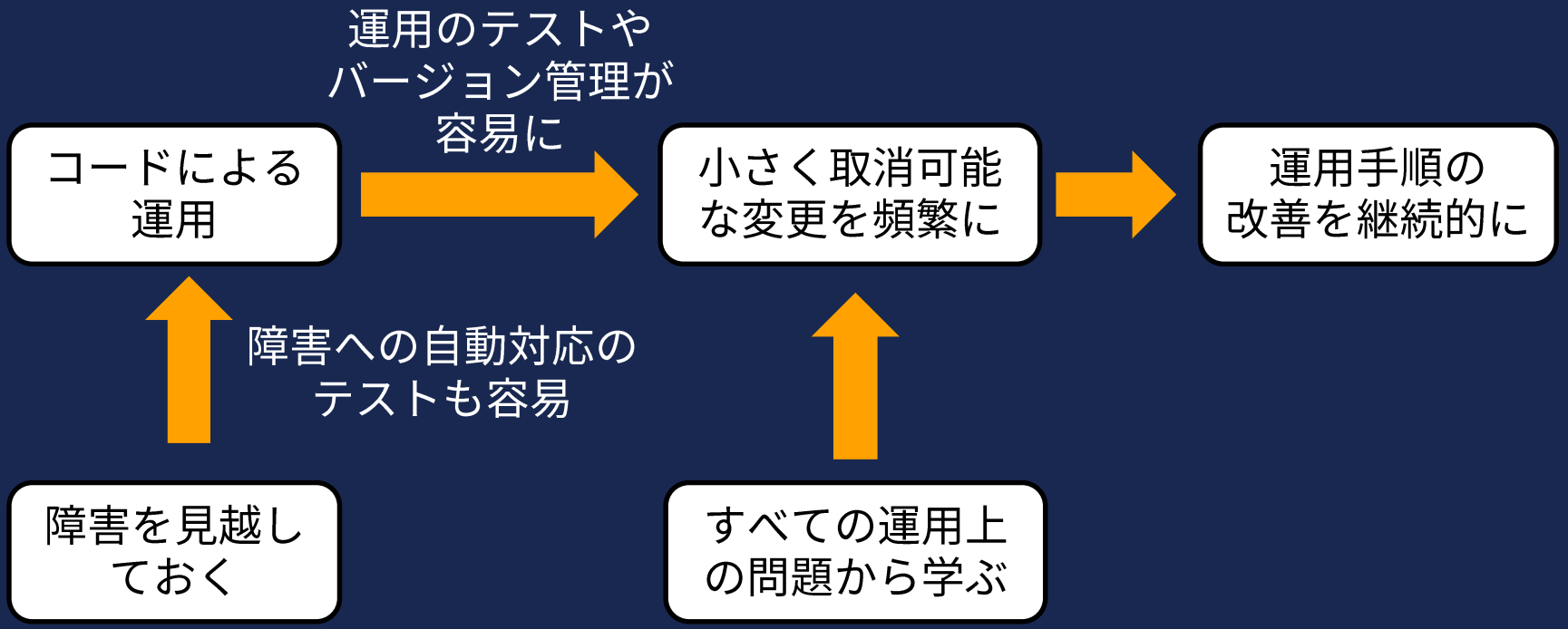
コストの最適化

運用性

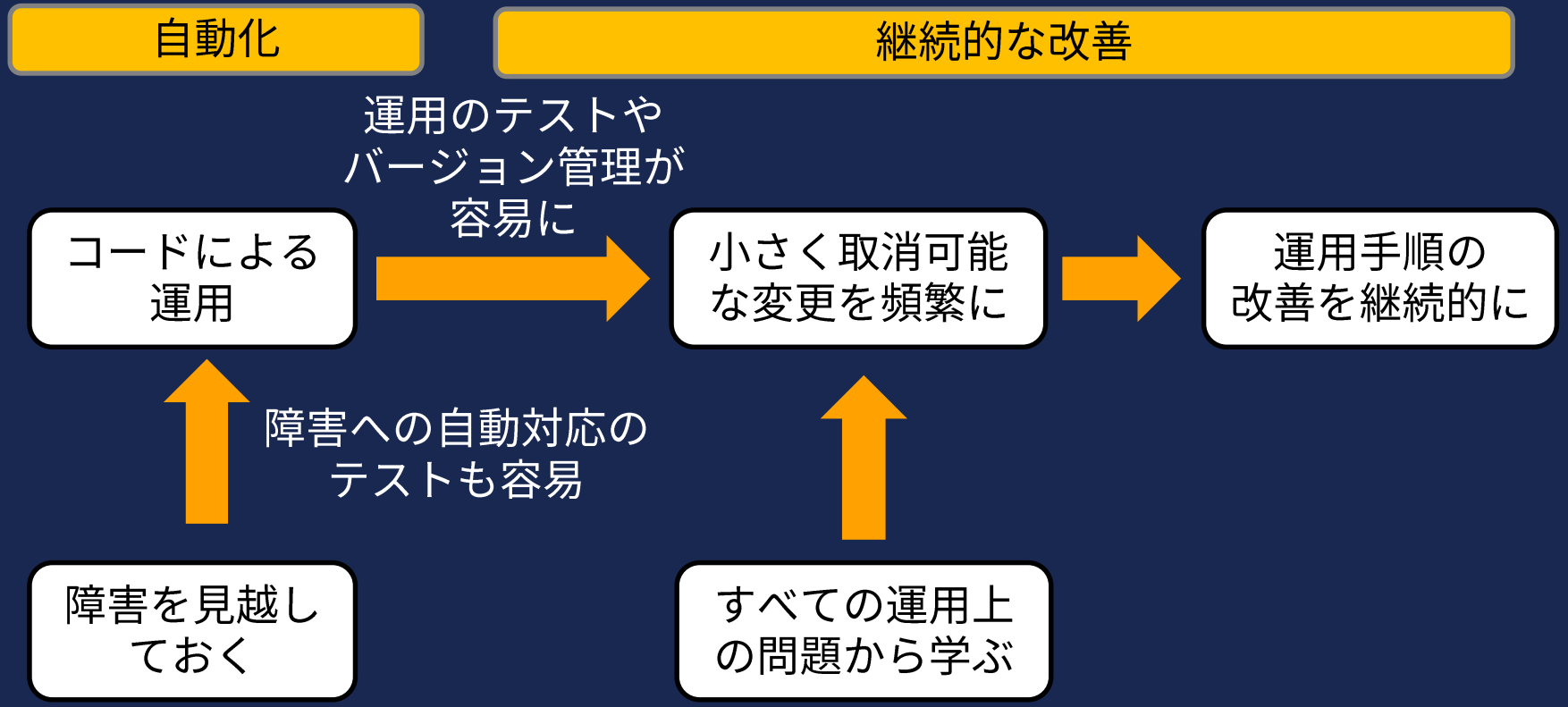
Well-Architected 運用性: 設計原則

- コードによる運用
- 小さく、取消可能な変更を頻繁に行う
- 運用手順の改善を継続的に行う
- 障害を見越しておく
- すべての運用上の問題から学ぶ

Well-Architected 運用性: 設計原則



Well-Architected 運用性: 設計原則



このセッションのアジェンダ

AWSの運用のベストプラクティス
AWSの運用に関連するサービスの概要

運用のプロセスを支えるサービス群

1 リソースの
プロビジョニング
- AWS CloudFormation
- AWS Service Catalog

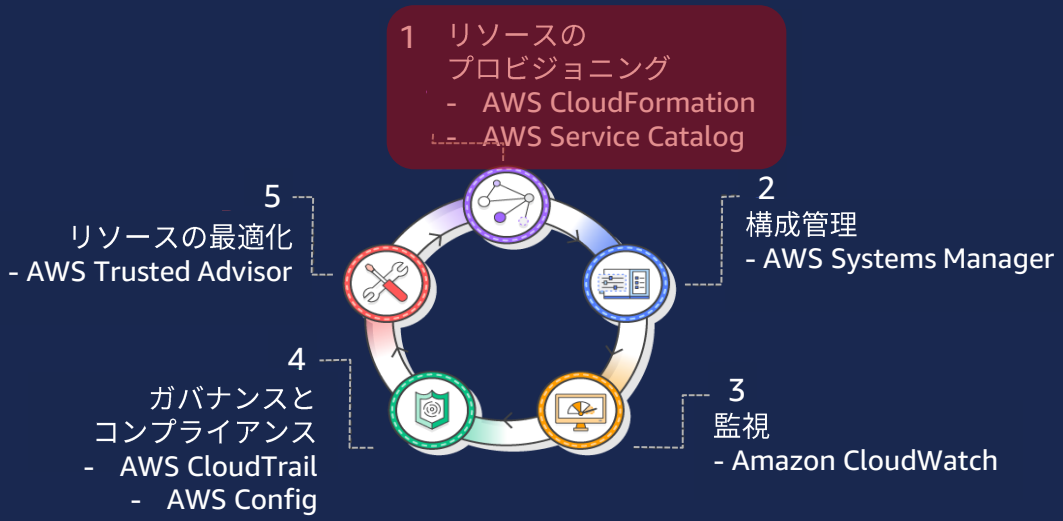
2
構成管理
- AWS Systems Manager

3
監視
- Amazon CloudWatch

4
ガバナンスと
コンプライアンス
- AWS CloudTrail
- AWS Config

5
リソースの最適化
- AWS Trusted Advisor

リソースのプロビジョニング



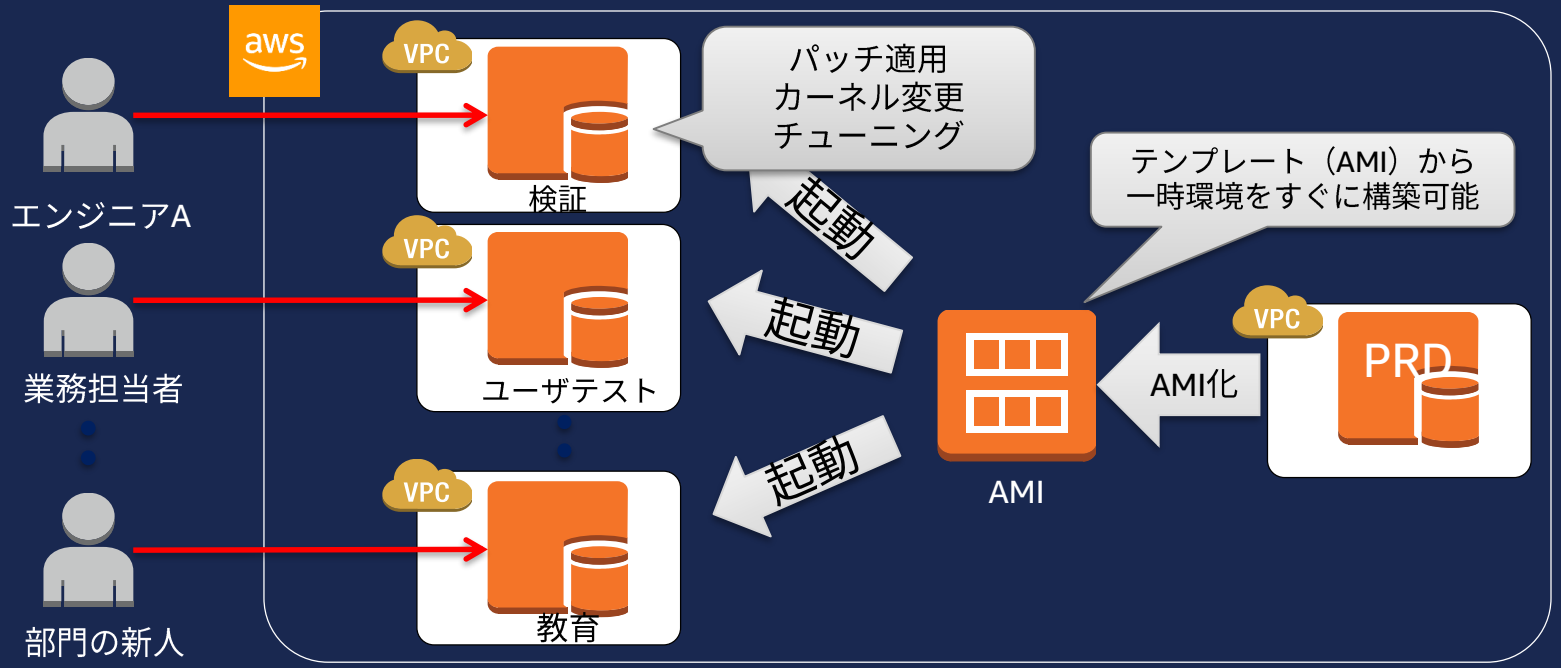
スタート地点

まずは

一台の仮想サーバーをプロビジョニング

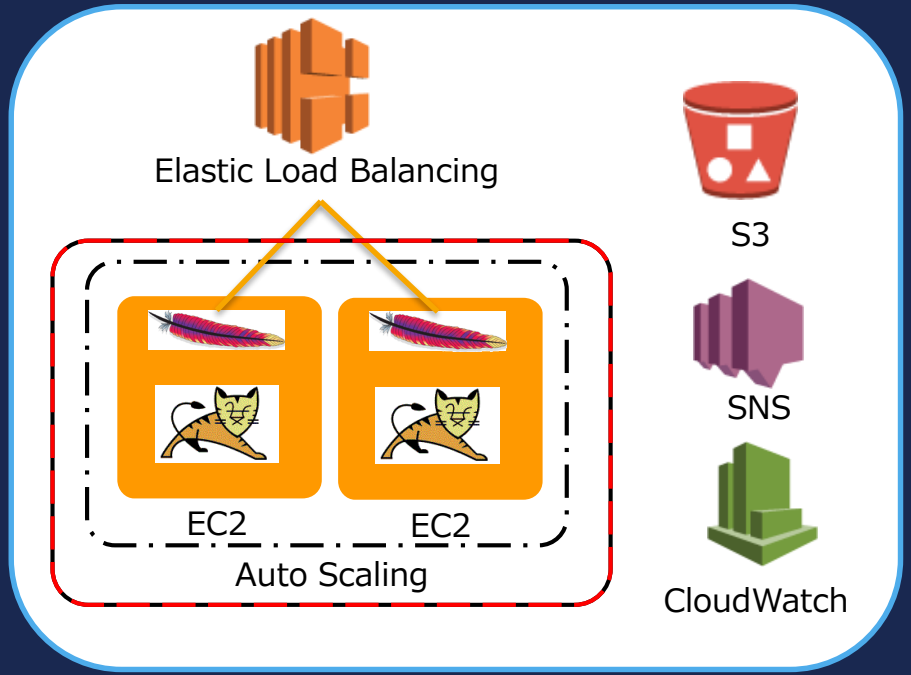
AMI(Amazon Machine Image)

EC2インスタンスを起動する際に指定するベースイメージ
EBSのスナップショットから独自のカスタマイズAMIを保存可能
AMIを利用することでセットアップ済みのサーバを容易に準備可能



現実のシステムはもっと複雑

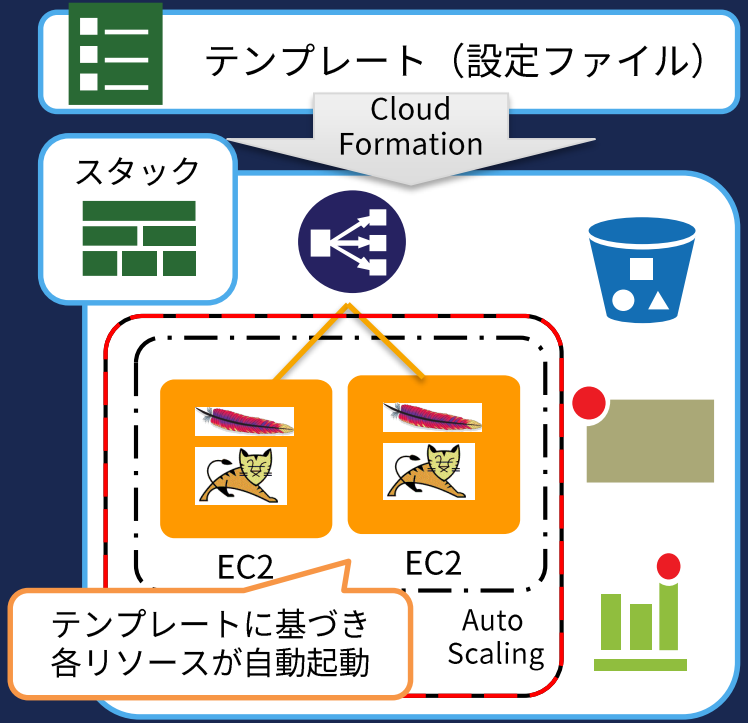
依存関係をもち複数のコンポーネントからなるシステムのプロビジョニングを自動化するにはどうすればよいか？



Amazon CloudFormation

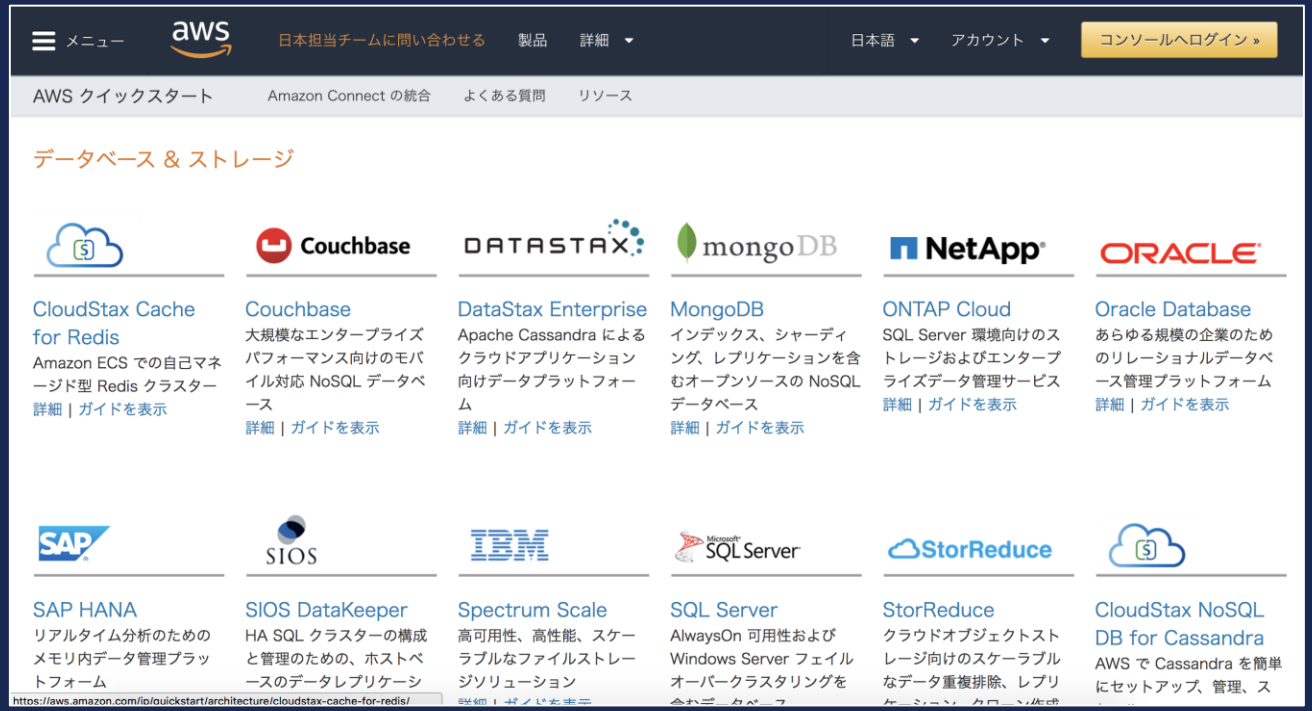
自動化

設定管理＆クラウドのオーケストレーション サービス



- テンプレートを元に、EC2やELBといったAWSリソースの環境構築を自動
- YAMLやJSONで、テンプレートを自由に記述可能
- Microsoft Windows Server や SAP HANA などのクイックスタートリファレンスを用意

クイックスタートリファレンス



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

環境構築の自動化で期待される効果

テスト済みのテンプレートを適用することにより環境構築のミスを排除

開発環境、テスト環境、障害再現環境など、必要な環境を何面でも、工数や時間をかけずに構築可能

システムの品質の向上

工数の削減
リードタイムの短縮

次のステップ

CloudFormationによる
プロビジョニングの自動化

次にやるべきことは？

次のステップ

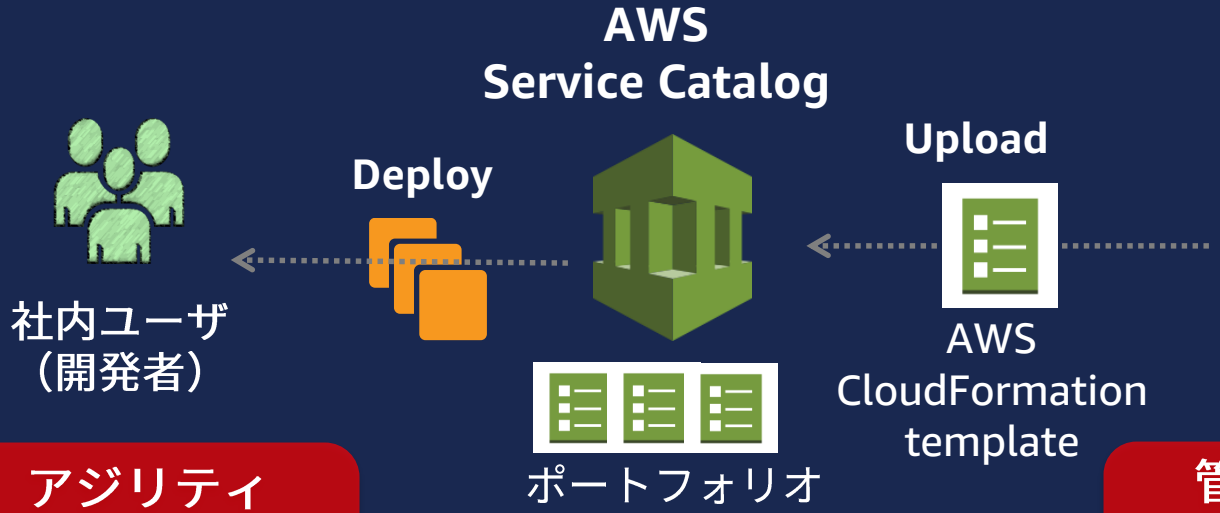
CloudFormationによる
プロビジョニングの自動化

プロビジョニングのセルフサービス化
定型的な構成はセルフサービスで
プロビジョニング

AWS Service Catalog

自動化

組織内サービスポータル提供サービス



アジリティ
セルフサービス
迅速な開発

管理の簡素化
可視性
コンプライアンス

構成管理

1 リソースの
プロビジョニング
- AWS CloudFormation
- AWS Service Catalog

2
構成管理
- AWS Systems Manager

3
監視
- Amazon CloudWatch

4
ガバナンスと
コンプライアンス
- AWS CloudTrail
- AWS Config

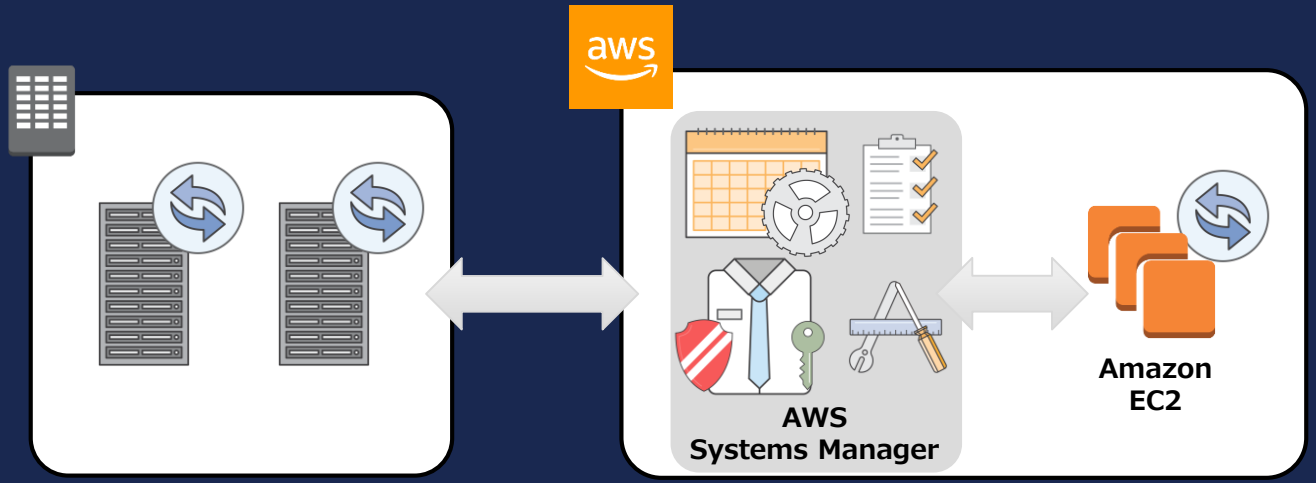
5
リソースの最適化
- AWS Trusted Advisor

大量のリソースをどのように管理していくか

構成の把握	構成の変更管理
例) 各サーバーにインストールされているソフトウェアを把握したい	例) 各サーバーのOSのパッチレベルをそろえる

AWS Systems Manager

Amazon EC2、またはオンプレミスで実行されるWindows、Linuxに対してシステムの自動構成と継続的な管理を可能にする一連の機能



AWS Systems Managerの機能

構成の把握

構成の変更管理

Inventory

Automation

Run Command

Patch Manager

State Manager

その他の機能に関しては → https://aws.amazon.com/jp/systems-manager/

Inventory

自動化　継続的な改善

ソフトウェアインベントリの情報収集

- EC2、オンプレの各種インベントリ情報を収集、管理
- AWSが定義する収集テンプレートを利用可能
- JSON形式で取得したいデータを定義する事でカスタマイズも可能
- AWS Configを有効にする事でインベントリ情報の変更履歴を追跡
- ソフトウェアのライセンス使用状況確認、ソフトウェアバージョンの管理が簡素化される事でのセキュリティ脆弱性の早期発見

Run Command

自動化

管理作業をリモートから実行

- リモートから任意のコマンド実行が可能
  - ソフトウェアのインストール、パッチング、アップデート
  - ユーザーの追加・削除、サービスの起動・停止、状態取得
- JSONベースのドキュメントでコマンド、タスクを定義
- 定義済みのドキュメントも提供、コミュニティ版もあり
- 実行結果はS3に保存可能、実行状態に合わせてSNSを使って通知
- SSH、RDPの接続ポートを閉じる事でセキュアに運用

Patch Manager

自動化

ベースラインを定義してWindows/Linuxのパッチを適用

- Patch Baselineを使ってカスタムパッチポリシーを定義
  - 例：クリティカルなパッチが提供された場合には1日後に適用
- パッチ適用は指定したMaintenance Window内で実施
- 実施されたパッチングの結果はレポートされる
  - インストールされたパッチ、スキップ、失敗したパッチ等
- 重要なアップデートやゼロデイ脆弱性への対応を自動化、時間を短縮

Automation

自動化　継続的な改善

シンプルなワークフローを使って一般的なタスクを自動化

- Amazon Machine Images(AMI)の作成と管理に最適
  - AMIからEC2を起動 → パッチ適用 → 更新されたAMIを作成
- JSONベースのドキュメントでワークフローを定義
- 企業で管理する「ゴールデンイメージ」管理をサポート

監視

1 リソースの
プロビジョニング
- AWS CloudFormation
- AWS Service Catalog

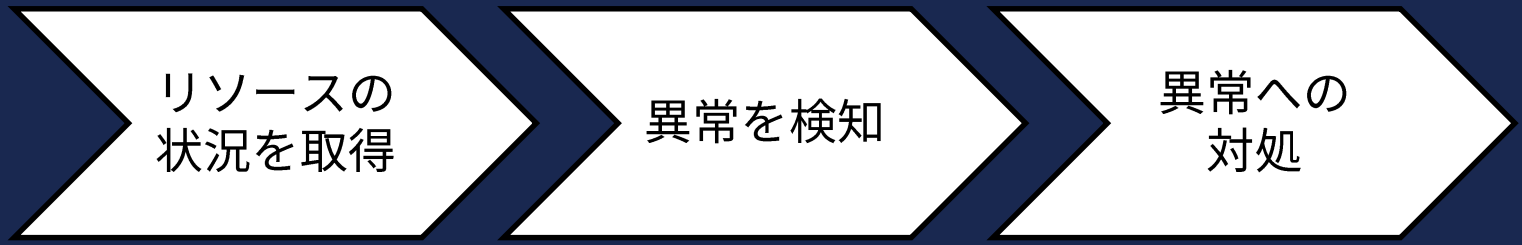
2
構成管理
- AWS Systems Manager

3
監視
- Amazon CloudWatch

4
ガバナンスと
コンプライアンス
- AWS CloudTrail
- AWS Config

5
リソースの最適化
- AWS Trusted Advisor

監視のながれ

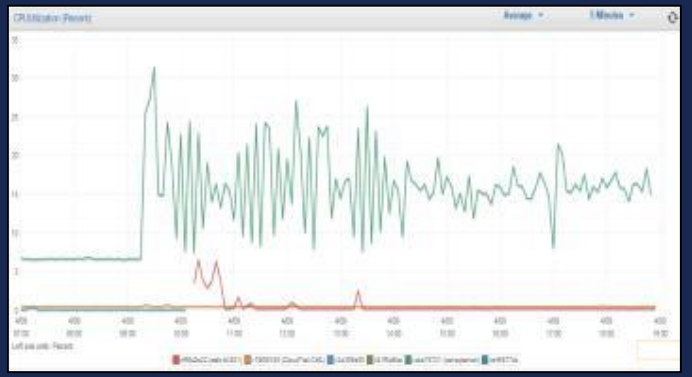


Amazon CloudWacth

CloudWatch

- ***CloudWatch***
  - AWS上で稼働するシステム監視サービス
    - ✓ 死活監視 / 性能監視 / キャパシティ監視
- ***CloudWatch Logs***
  - ログ管理プラットフォーム サービス
    - ✓ EC2上のOS, APPのログ
    - ✓ AWSのマネージドサービスのログ
- ***CloudWatch Events***
  - AWS上リソースの状態監視サービス
  - AWSリソースに対するイベントをトリガーにアクションを実行する機能

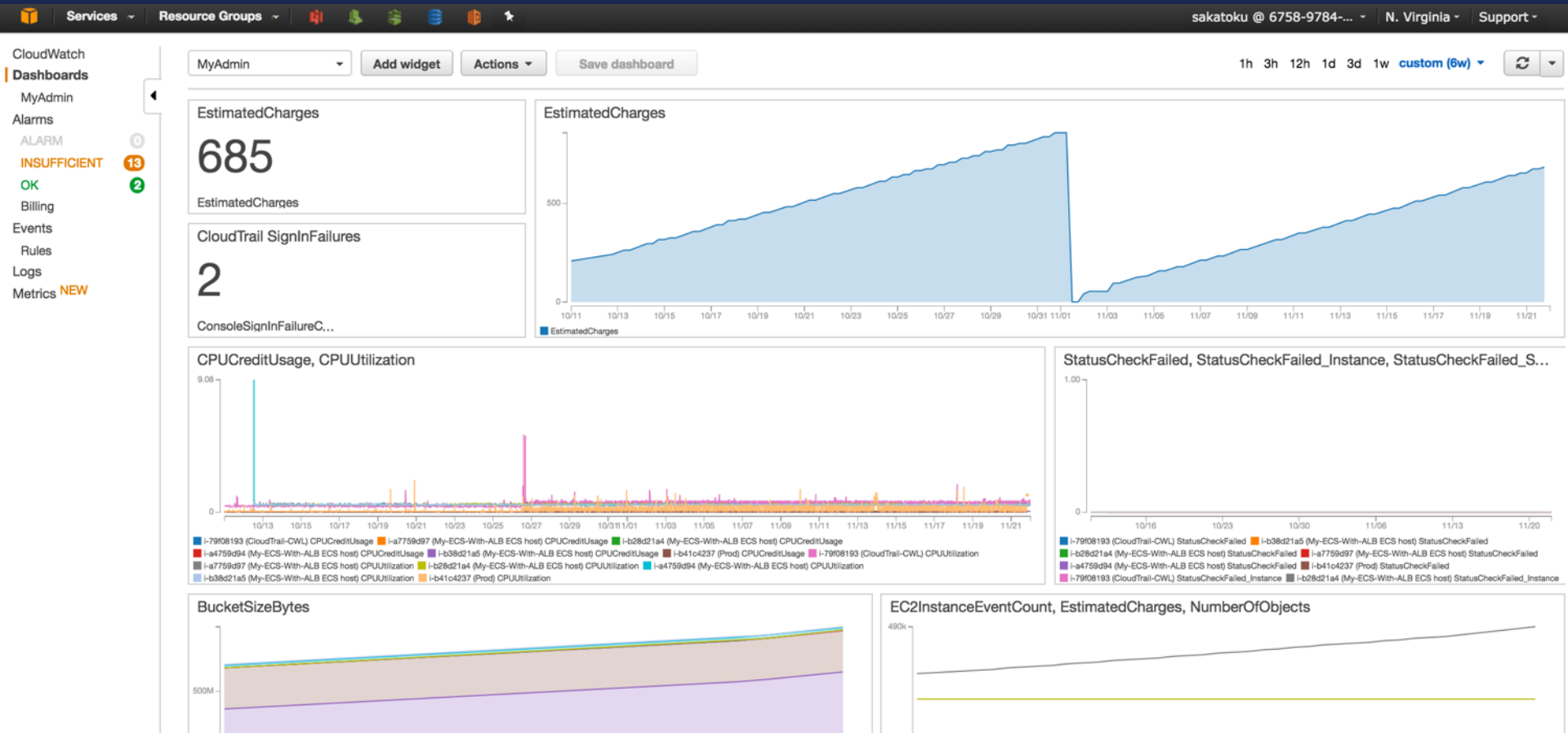
Amazon CloudWatch



AWSの各種リソースを監視

- 各種AWSリソースの状態・死活、性能、ログ監視 (監視)
- 取得メトリクスのグラフ化 (可視化)
- 各メトリックスをベースとしたアラーム(通知)、アクションの設定が可能
- カスタムメトリクスの作成が可能

CloudWatchによる環視の例

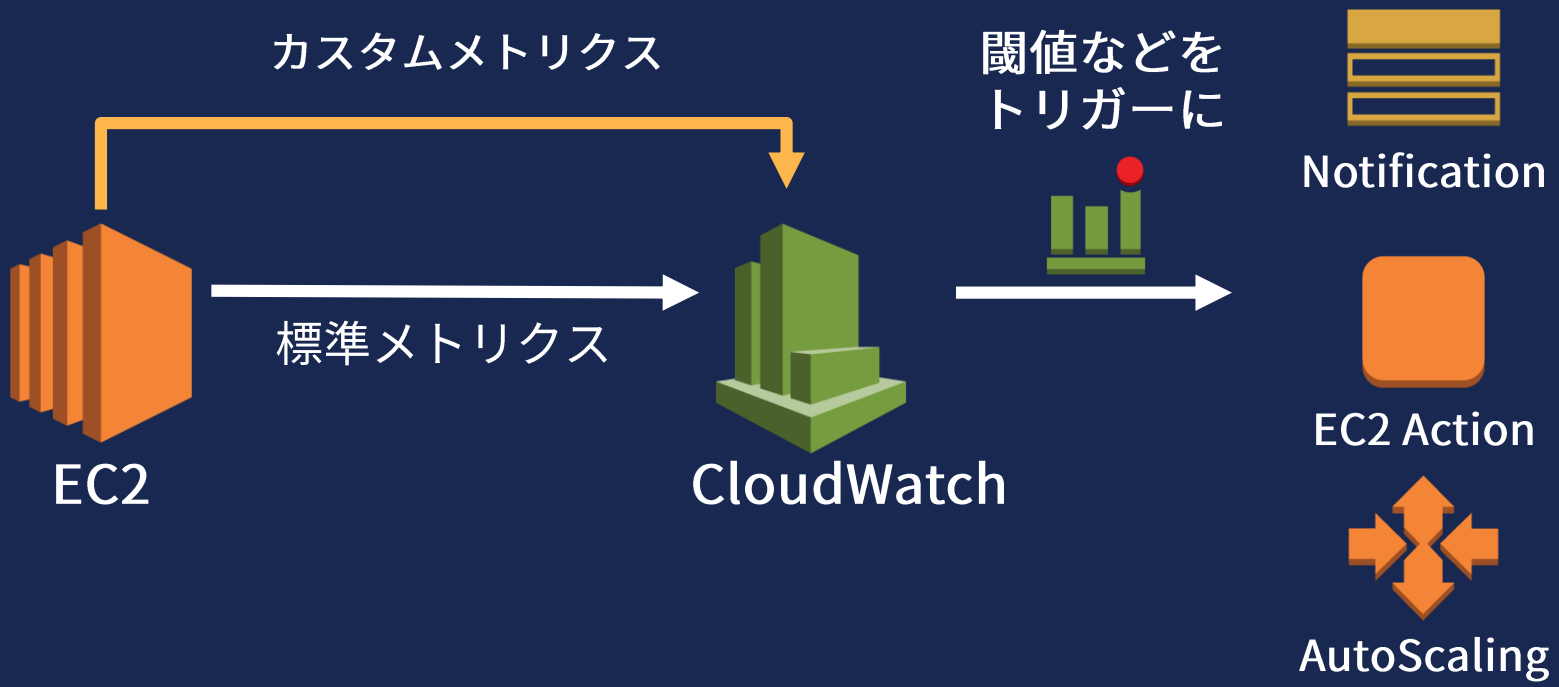


Amazon CloudWatch のアクション機能

自動化

モニタリング

アクション



CloudWatch Logs

自動化

各種ログを収集

CloudWatch Logs

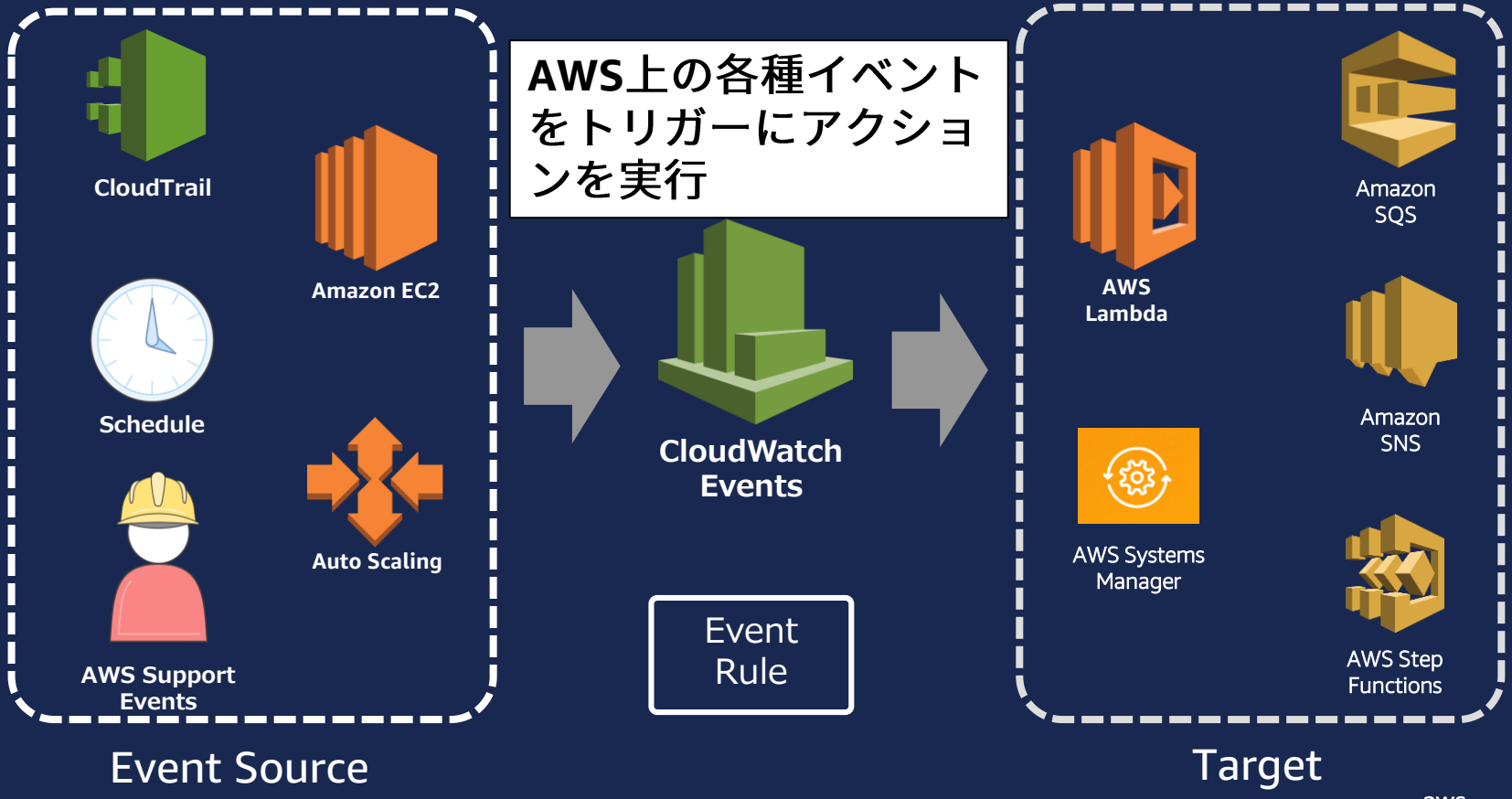
通知:
CloudWatch Alarm

エクスポート:
Amazon Kinesis Firehose

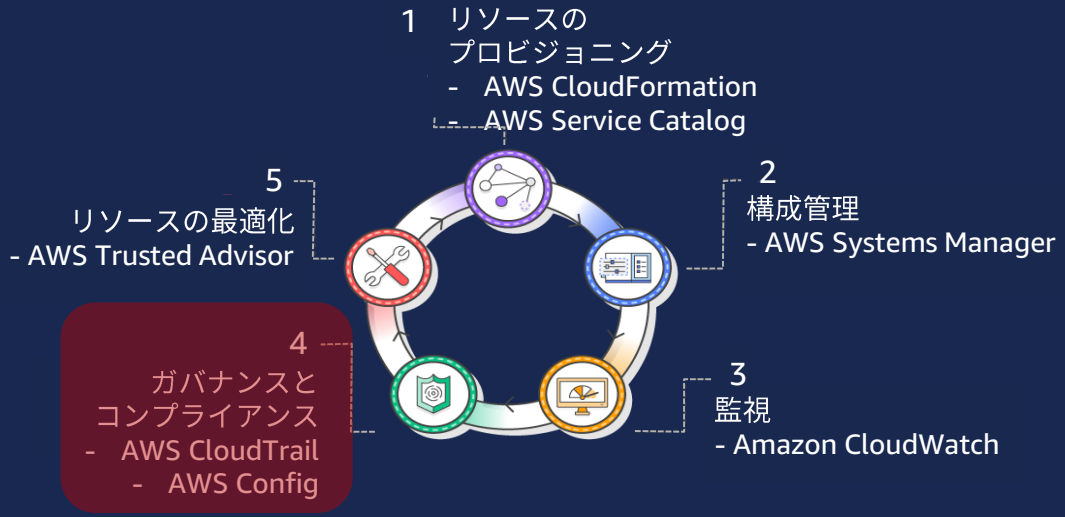
可視化:
Amazon Elasticsearh Service (Kibana)

CloudWatch Events利用イメージ

自動化



ガバナンスとコンプライアンス



内部統制

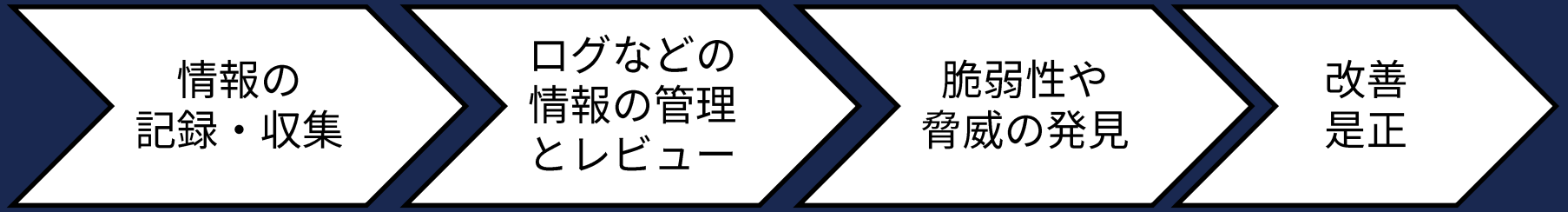
予防的統制	発見的統制
潜在的なリスクに対して 事前にリスクの発生を防止 例) 認証・認可によるアクセス可能な機能の制限	顕在化したリスクに対して 適切に是正・対応 例) ログによる不正な操作の検出と対応

内部統制

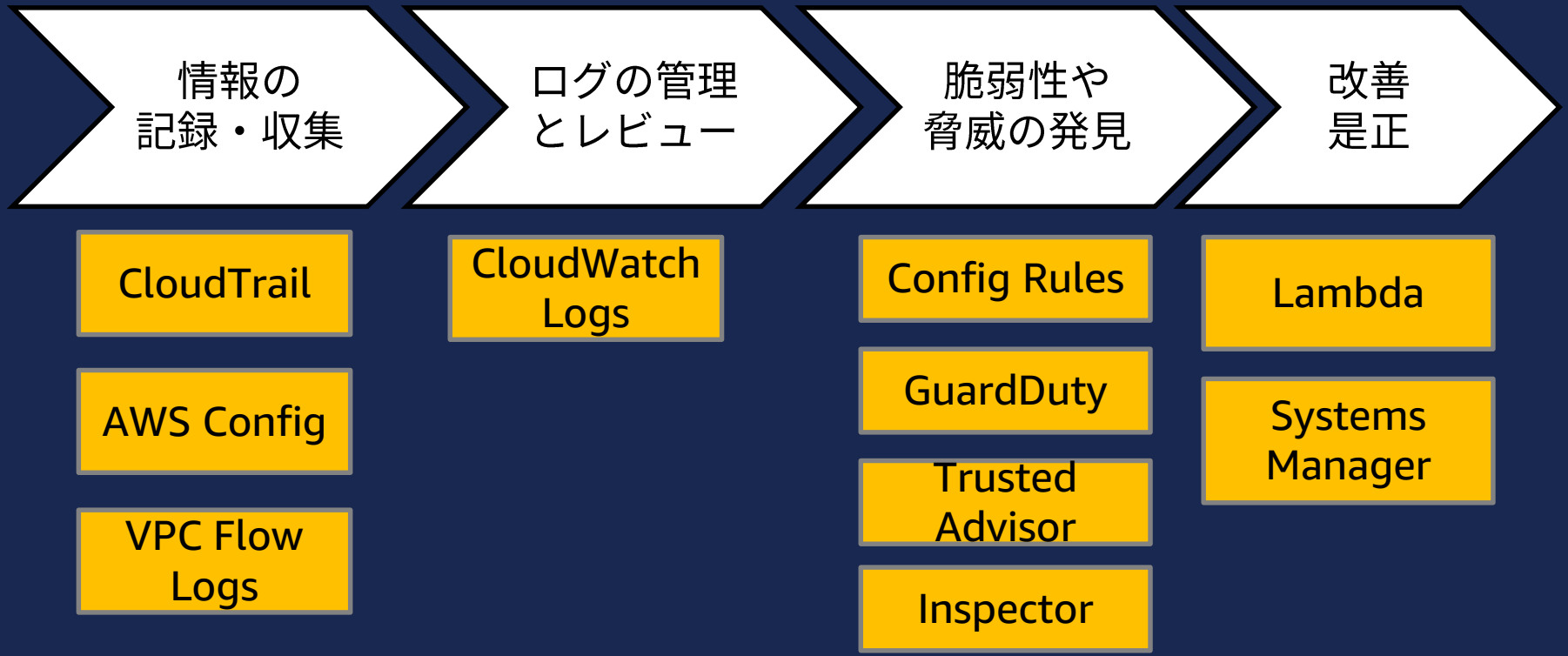
予防的統制	発見的統制
潜在的なリスクに対して 事前にリスクの発生を防止 例) 認証・認可によるアクセス可能な機能の制限	顕在化したリスクに対して 適切に是正・対応 例) ログによる不正な操作の検出と対応

本セッションでは運用時の
発見的統制に係わるサービスを
取り上げます

発見的統制



発見的統制 – 対応するAWSのサービス



AWS CloudTrail

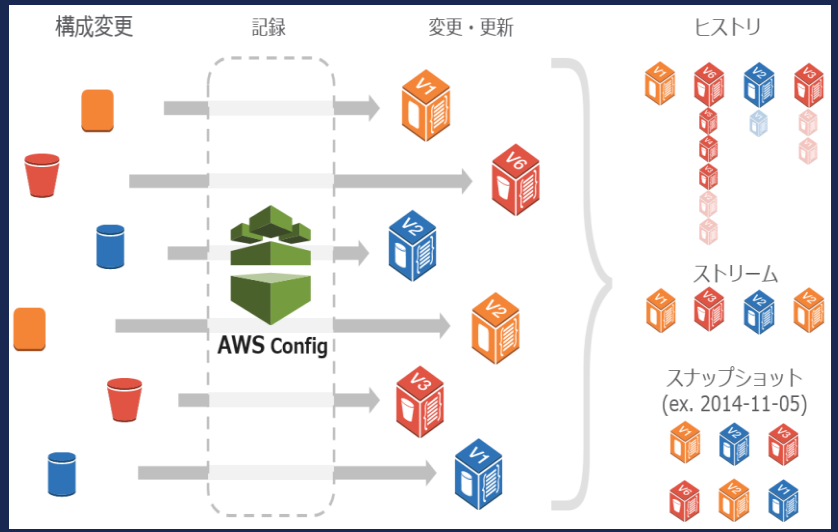
AWS上のAPI操作を記録するサービス

- AWSマネジメントコンソール、コマンドライン、サードパーティ製品等AWS APIの呼び出しを記録
- セキュリティの分析、リソース変更の追跡、およびコンプライアンスの監査に利用

AWS CloudTrail サポートサービス一覧
https://docs.aws.amazon.com/ja_jp/awscloudtrail/latest/userguide/what_is_cloud_trail_supported_services.html

AWS Config

構成変更の通知、構成履歴を記録する構成管理マネージドサービス

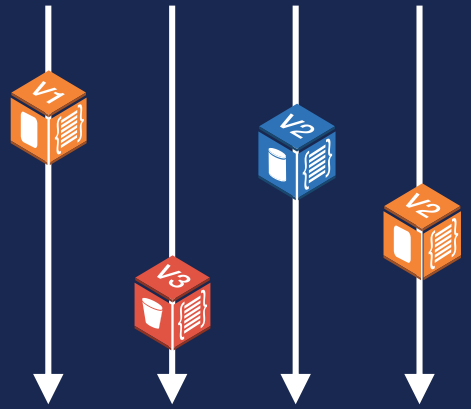


- AWSリソースの変更履歴、構成情報を管理
- アカウント内のAWSリソース間の関係をリレーションシップとして関連付ける

AWS Config

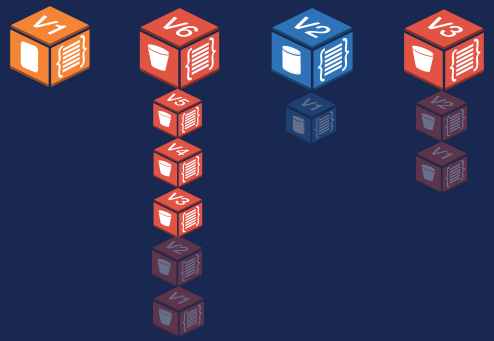
ストリーム
(Configuration Stream)

- リソースが作成/変更/削除されるたびに作成
- 構成ストリームに追加される
- SNSトピック連携可能



ヒストリー
(Configuration History)

- 任意の期間における各リソースタイプの構成要素の集合
- リソースの設定履歴を、指定したS3バケットに保存



スナップショット
(Configuration Snapshot)

- ある時点でのコンフィグレーションアイテムの集合
- 自動で定期的、あるいは変更トリガで作成され、指定したS3バケットに保存

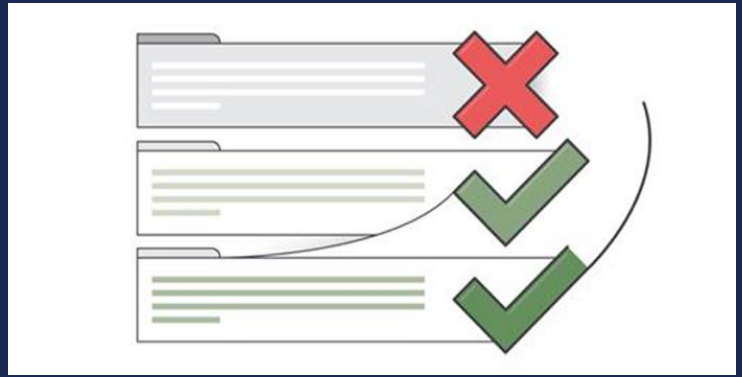
Snapshot @ 2017-06-02, 5:40pm

AWS Config Rulesによるポリシー適合の評価

自動化

準拠すべきルールを事前に設定し、その内容に沿った構成変更が行われているかを評価

- •全てのEBSボリュームが暗号化されているか
- •EC2インスタンスが適切にタグ付されているか等



AWS Managed Rules

- AWSにより定義・提供される
- AWSにより運用される
- 必要最低限のベーシック・ルール

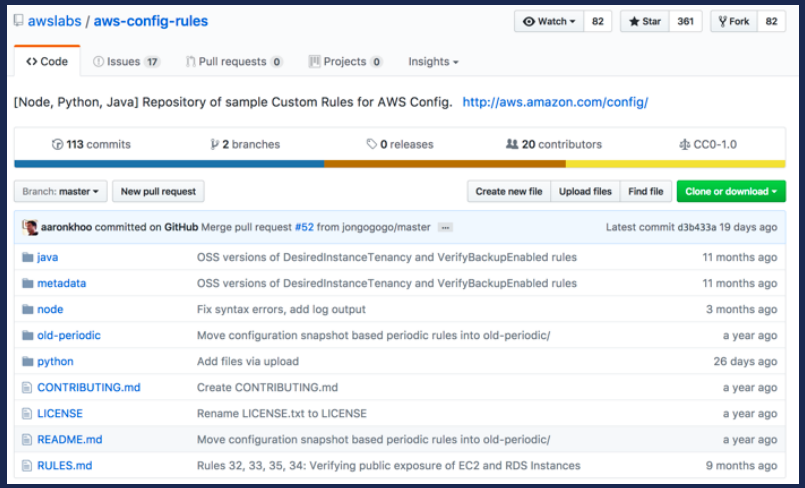
Customer Managed Rules

- 自分でAWS Lambdaをベースにルールを作成可能
- 管理自体は作成者 (自分) で実施

Customer Managed Rules

自動化　継続的な改善

- ルールをカスタマイズし、環境にあったベースラインを作成
- AWS Lambda ベース
- AWS Config Rules Repository
    - GitHub(awslabs/aws-config-rules)
        - IAMポリシー関連
        - IAM鍵のローテーション
        - MFAの有効化
        - ルートアカウントの無効化
        - VPC Flow Logの有効化
        - タグフォーマットの制御



https://github.com/awslabs/aws-config-rules/blob/master/RULES.md

リソースの最適化

1 リソースの
プロビジョニング
- AWS CloudFormation
- AWS Service Catalog

2
構成管理
- AWS Systems Manager

3
監視
- Amazon CloudWatch

4
ガバナンスと
コンプライアンス
- AWS CloudTrail
- AWS Config

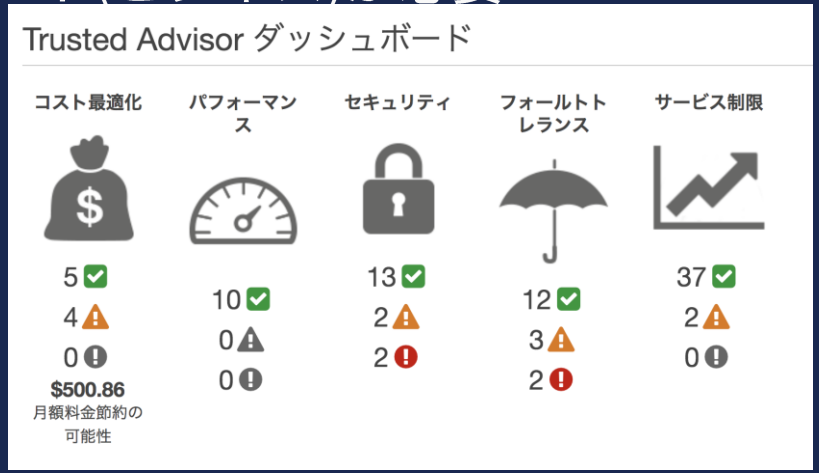
5
リソースの最適化
- AWS Trusted Advisor

Trusted Advisor

継続的な改善

利用実績や設定情報を元に自動的に、コスト最適化、パフォーマンス、セキュリティ、耐障害性の提案を行うツール

- 使用率の低いEC2, 利用頻度の低いEBS, 関連付けれれていないEIPなどを指摘
- すべての機能の利用にはAWSサポート(ビジネス)が必要



Trusted Advisorのヘルスチェックの例

継続的な改善

カテゴリ	チェックする内容	例
コスト最適化	コスト最適化の可能性がある項目に対する推奨事項	使用率の低いEC2インスタンス 利用頻度の低いEBSボリューム など
セキュリティ	お客様のシステムのセキュリティ弱体化につながる恐れのある設定	セキュリティグループ(無制限アクセス）、MFA設定　など
耐障害性	お客様システムのアプリケーションの可用性や冗長性を高めるためのベストプラクティスからの推奨事項	RDSのマルチAZ構成、EBSスナップショット、ELBのクロスゾーン設定　など
パフォーマンスの向上	アプリケーションの拡張性や応答性の改善、過剰なキャパシティのチェックなどパフォーマンス最適化のための推奨事項	サービス制限、高負荷なEC2インスタンス、CloudFrontのキャッシュヒット率チェック　など

各サービスのオンラインセミナー資料

AWS クラウドサービス活用資料集

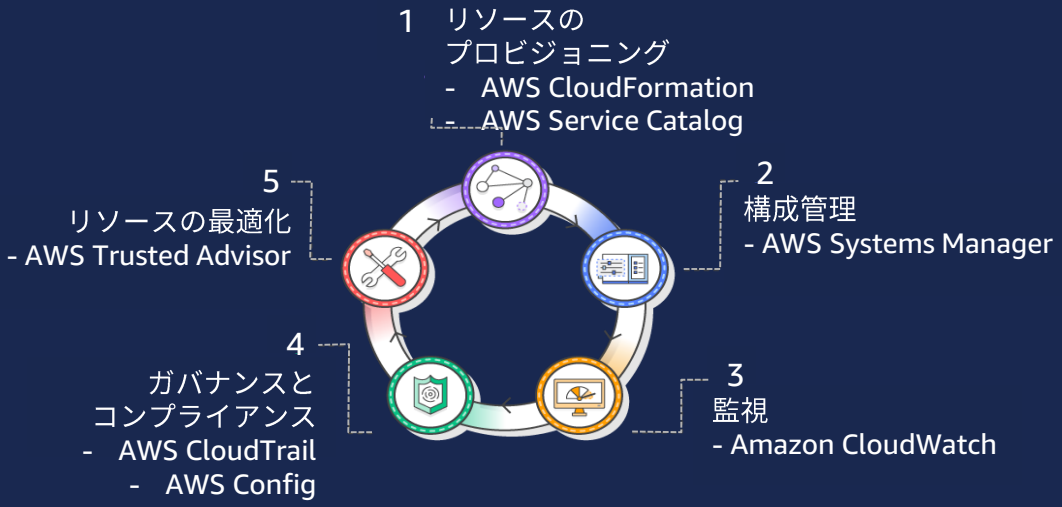
- https://aws.amazon.com/jp/aws-jp-introduction/

AWS Summit Tokyo 2018 関連セッション

- AWS Systems Managerによるシステム運用管理の実践
- AWS環境をコードで管理する ～コード化の開始から頻出パターンまで～
- AWSのオペレーション最適化の勘所
- AWSセキュリティ入門1 – リスク評価と保護
- AWSセキュリティ入門2 – 脅威検知と対応
- DevSecOps on AWS -AWSのモニタリング-
- AWSによるセキュリティ・オートメーションの実践

まとめ

- 運用の各フェーズで各種のAWSのサービスを活用できます
- これらのサービスを利用して運用の自動化と継続的な改善を実践することが可能です



自動化

継続的な改善

本セッションのFeedbackをお願いします

お手元のサミットガイドブックの表紙に記載している『QRコード』からご回答ください。
もれなく**素敵なAWSオリジナルグッズ**をプレゼントします。

プレゼントの引き換えは、パミール3F展示会場内アンケート確認エリア・受付エリアのいずれかにお越しください。