[AWS Black Belt Online Seminar] AWS Systems Manager

サービスカットシリーズ

Solutions Architect 石橋 香代子
2020/02/12

AWS 公式 Webinar
https://amzn.to/JPWebinar

過去資料
https://amzn.to/JPArchive

石橋 香代子 (いしばし かよこ)

ソリューションアーキテクト

- 流通・小売業界のエンタープライズ企業をサポート
- 運用系サービス

好きなAWSのサービス： **AWS Systems Manager**
Amazon CloudWatch

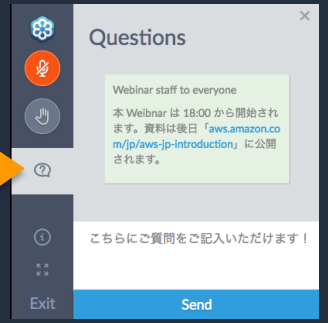
AWS Black Belt Online Seminar とは

「サービス別」「ソリューション別」「業種別」のそれぞれのテーマに分かれて、アマゾン ウェブ サービス ジャパン株式会社が主催するオンラインセミナーシリーズです。

質問を投げることができます！

- 書き込んだ質問は、主催者にしか見えません
- 今後のロードマップに関するご質問は
  お答えできませんのでご了承下さい

① 吹き出しをクリック
② 質問を入力
③ Sendをクリック



Twitter ハッシュタグは以下をご利用ください
#awsblackbelt

内容についての注意点

- 本資料では2020年02月12日時点のサービス内容および価格についてご説明しています。最新の情報はAWS公式ウェブサイト(http://aws.amazon.com)にてご確認ください。

- 資料作成には十分注意しておりますが、資料内の価格とAWS公式ウェブサイト記載の価格に相違があった場合、AWS公式ウェブサイトの価格を優先とさせていただきます。

- 価格は税抜表記となっています。日本居住者のお客様には別途消費税をご請求させていただきます。

- AWS does not offer binding price quotes. AWS pricing is publicly available and is subject to change in accordance with the AWS Customer Agreement available at http://aws.amazon.com/agreement/. Any pricing information included in this document is provided only as an estimate of usage charges for AWS services based on certain information that you have provided. Monthly charges will be based on your actual use of AWS services, and may vary from the estimates provided.

本セッションの目的

- AWS Systems Managerの全体像をご理解いただく。
- AWS Systems Managerの各機能の概要を掴んでいただき、どんなことができるのか、イメージを持っていただく。

本日お話ししないこと

- AWS Systems Managerの各機能の詳細

アジェンダ

1. AWS Systems Manager 全体像
2. AWS Systems Managerを使ってみよう
    1. 準備編
    2. リソースの”今”を把握しよう
    3. SSMで定型運用を実施しよう
    4. 非定型なインタラクティブ操作もSSMで
    5. アプリケーションの設定管理もSSMで
3. AWS Systems Managerのセキュリティーベストプラクティス
4. まとめ

アジェンダ

1. **AWS Systems Manager 全体像**
2. AWS Systems Managerを使ってみよう
    1. 準備編
    2. リソースの”今”を把握しよう
    3. SSMで定型運用を実施しよう
    4. 非定型なインタラクティブ操作もSSMで
    5. アプリケーションの設定管理もSSMで
3. AWS Systems Managerのセキュリティーベストプラクティス
4. まとめ

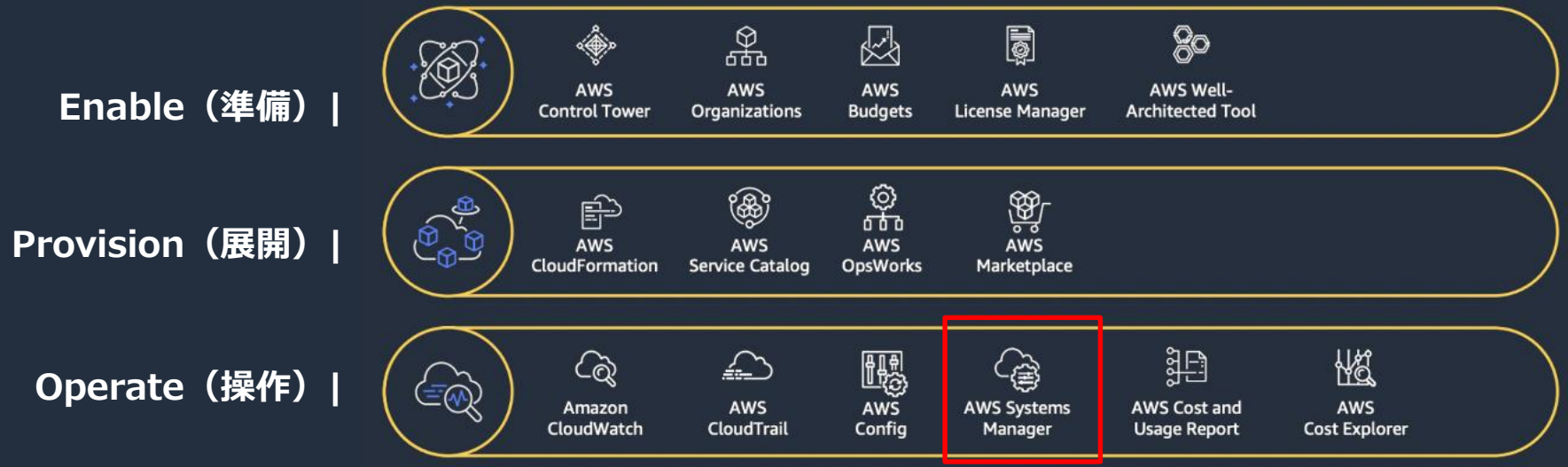
AWS マネジメント & ガバナンス サービス

AWS環境の運用管理を スケーラブルかつコスト効率よく行うサービス群

ビジネスアジリティとガバナンスコントロールの両立

AWS マネジメント & ガバナンス サービス

AWS環境の運用管理を スケーラブルかつコスト効率よく行うサービス群



ビジネスアジリティとガバナンスコントロールの両立

AWS Systems Manager (AWS SSM)

安全かつスケーラブルにAWS環境を運用するためのコックピット

グループ化
アプリケーションのリソース群をグループ化

可視化
アプリケーション運用上の洞察を可視化
多数のAWSリソースを1つのコンソールで

対応
安全性高いAWSのベストプラクティスで対応

AWSとオンプレミス
両方をサポート
クロスプラットフォーム対応
WindowsもLinuxも

Systems Manager = SSMと略します

aws

AWS SSM：Features (1/2)

全体

クイックセットアップ	インスタンスをSSMで管理するよう自動構成

オペレーションの管理

Explorer	運用アイテム情報のダッシュボード(XRXA*)
OpsCenter	運用アイテム（対応が必要なイベント）の管理

アプリケーションマネジメント

リソースグループ	タグによるサーバ群のグループ管理
AppConfig	アプリケーション設定（機能フラグ等）の管理
パラメータストア	設定パラメータの集中管理用データストア

アクションと変更

Automation	AWS環境全体に対する自動化処理の実行
Change Calendar	実行可否を制御するカレンダー
メンテナンスウィンドウ	自動化処理のスケジュールと順序の管理

*XRXA=CrossRegion CrossAccount

AWS SSM：Features (2/2)

インスタンスとノード

コンプライアンス	コンプライアンスの適合状態ダッシュボード
インベントリ	サーバ構成情報のインベントリを閲覧する
マネージドインスタンス	SSM管理対象のサーバ一覧
ハイブリッド アクティベーション	オンプレミスサーバをSSM管理下に入れる
セッションマネージャー	SSMを使ったサーバへリモートアクセスする
Run Command	サーバ群の上でコマンドを実行する
ステートマネージャー	サーバ群の構成を指定した状態に維持する
パッチマネージャー	サーバ群に指定ルールに基づきパッチを適用する
ディストリビューター	サーバ群にパッケージをインストールする

共有リソース

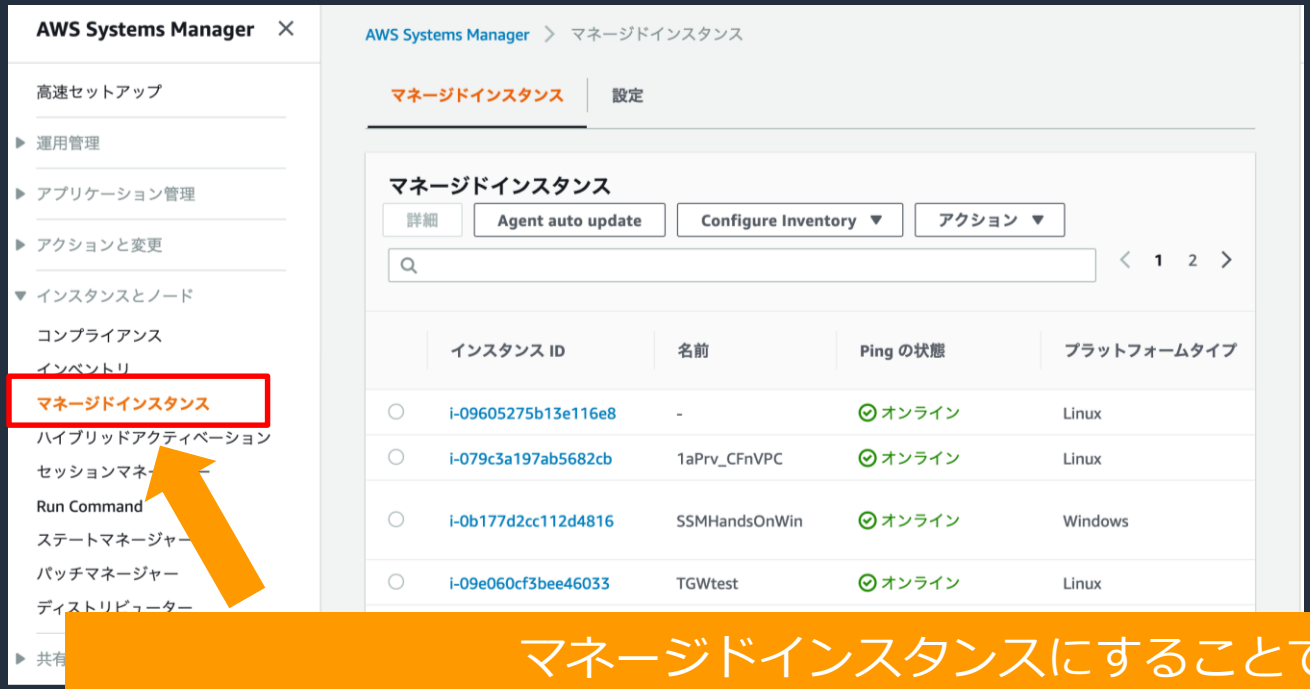
ドキュメント	SSMで実行する処理を記述したドキュメント

aws

アジェンダ

1. AWS Systems Manager 全体像
2. **AWS Systems Managerを使ってみよう**
    1. **準備編**
    2. リソースの”今”を把握しよう
    3. SSMで定型運用を実施しよう
    4. 非定型なインタラクティブ操作もSSMで
    5. アプリケーションの設定管理もSSMで
3. AWS Systems Managerのセキュリティーベストプラクティス
4. まとめ

Step1. まずは、マネージドインスタンスにしよう



マネージドインスタンス：

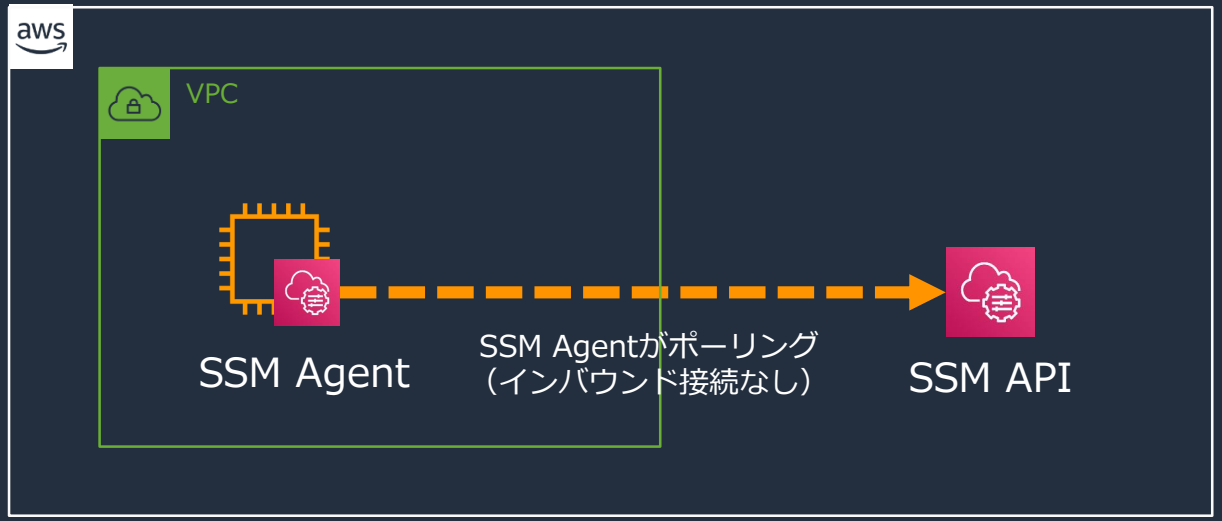
・SSM管理下のインスタンス群

・EC2インスタンスのほか、オンプレミスのインスタンスも含められる。

マネージドインスタンスにすることで、
オンプレミス/AWSハイブリッド環境のインスタンス管理が可能に

マネージドインスタンスにするために ①SSM Agentの導入

- SSM AgentがSSM APIと連携し各種操作、コントロールを行う。
- Amazon LinuxやWindows、 Ubuntu Serverのオフィシャルイメージには導入済み
  - それ以外のAMI、及びオンプレミスサーバは、手動でインストール
- 幅広い対応OS（WindowsServer2003～、RHEL6.0～、Ubuntu12.04～、Raspbian等）
  - https://docs.aws.amazon.com/ja_jp/systems-manager/latest/userguide/prereqs-operating-systems.html



詳細は、https://docs.aws.amazon.com/ja_jp/systems-manager/latest/userguide/ssm-agent.html

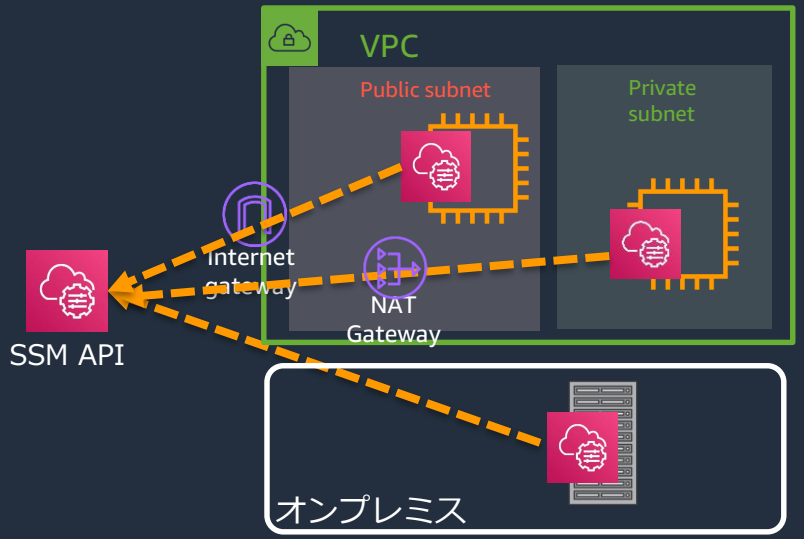
マネージドインスタンスにするために ②SSM APIへの経路確保

- 以下2パターンのどちらかで、SSM Agentからのアウトバウンド経路を確保する。

1, インターネット経由

・インバウンドアクセスは不要

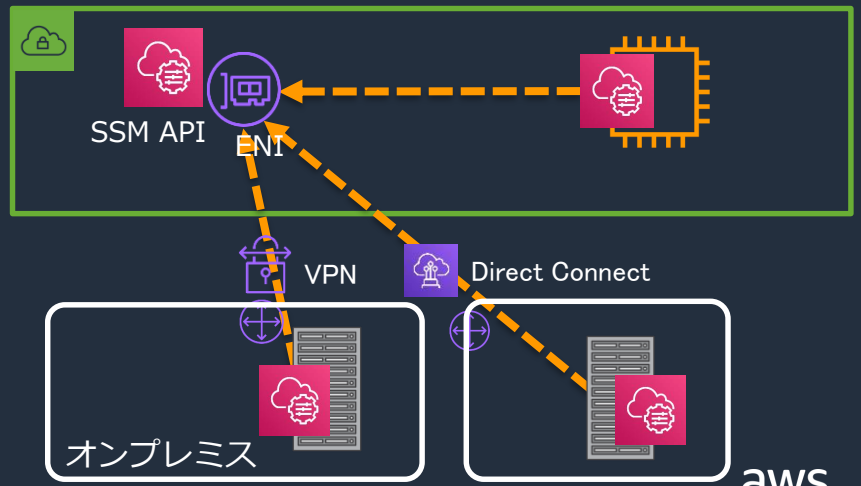
・パブリックサブネットやNAT Gatewayを使用



2, VPC エンドポイント経由

・プライベートネットワークによる接続が可能

・オンプレミスからもAWS Direct ConnectやVPN経由で閉域網経由のアクセスが可能



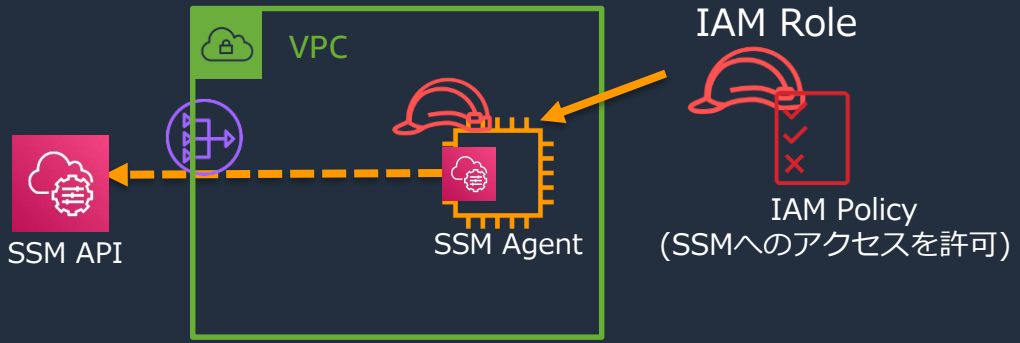
マネージドインスタンスにするために ③ IAMロール付与

- IAMロールを作成し、EC2にアタッチ
- IAMポリシー

  1, 「AmazonSSMManagedInstanceCore」 でコア機能をアタッチ(必須)

  2, 必要に応じて、S3などのポリシーをアタッチ(option)

  (※)以前からある「AmazonEC2RoleforSSM」ポリシーの使用も可能だが、
  権限が広いため、「AmazonSSMManagedInstanceCore」をベースに割り当てることを推奨



詳細は、https://docs.aws.amazon.com/ja_jp/systems-manager/latest/userguide/session-manager-getting-started-instance-profile.html

ここまでやれば、晴れてマネージドインスタンスに！

- マネージドインスタンスにするための手順の復習
  - 1, SSM Agentの導入
  - 2, SSM APIへの経路確保
  - 3, IAMロール付与

- しかし、ここで出てくるよくある悩み

全てのインスタンスで、
これを徹底できるかが不安

Agentは導入済みのものを使っているし、VPCエンドポイントは一度作れば問題ないけど、ロールは一つ一つに設定が必要だし・・・

➡ **クイックセットアップ**

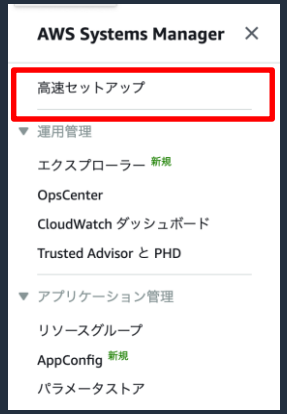
クイックセットアップ（高速セットアップ）

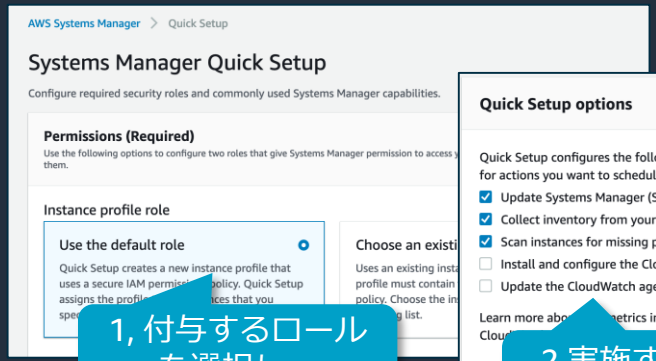
必要なセキュリティロールと一般的に使用される SSM機能をEC2インスタンスですばやく設定

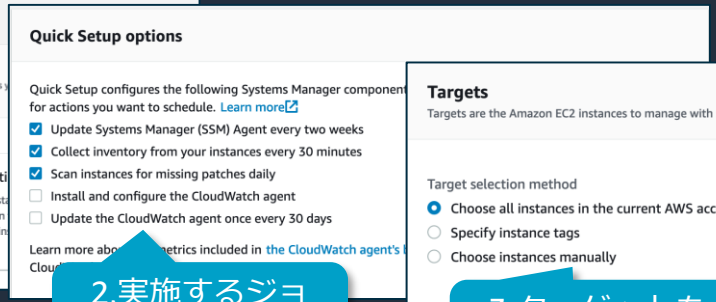
NEW

2019/08～

- インスタンスのSSM設定の自動構成ができる機能
  - SSM の IAMインスタンスプロファイルのロール
  - SSM Agent のスケジュールされた隔週ごとの更新
  - 30 分ごとにスケジュールされたインベントリメタデータの収集
  - 欠落しているパッチを特定するために、インスタンスを毎日スキャン
  - Amazon CloudWatch エージェントの 1 回限りのインストールと設定
  - CloudWatch エージェントのスケジュールに基づく毎月の更新







1, 付与するロールを選択し、

2,実施するジョブを選択し、

3,ターゲットを指定する。

クイックセットアップ（高速セットアップ）

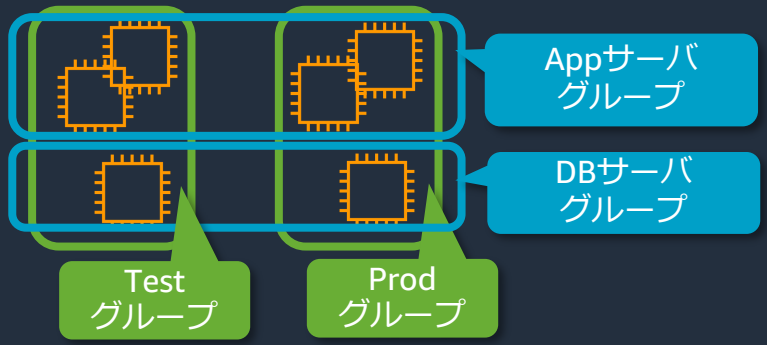
必要なセキュリティロールと一般的に使用されるSSM機能をEC2インスタンスですばやく設定

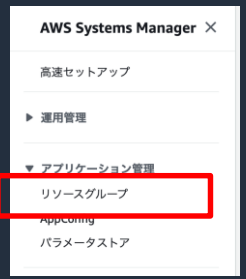
- クイックセットアップを使うと・・・
- （いいところ１）新規インスタンスも自動でマネージドインスタンスにすることが可能
  - ただし、SSM Agentが導入されていること、SSM APIへの経路確保されていることが前提
  - すでにロールが割り当てられている場合は、置き換えはしないので注意
- （いいところ２）SSMのベストプラクティスに則って管理できる。
  - 2週間毎のSSM Agentの自動更新、30分おきのインベントリー収集など

詳細は、https://docs.aws.amazon.com/ja_jp/systems-manager/latest/userguide/systems-manager-quick-setup.html

Step2. インスタンスをグループ化しよう – リソースグループ

- 「リソースグループ」は、AWS リソースをグルーピングすることで、整理・管理をしやすくする機能
  - タグベース or CloudFormationスタックベース で指定できる





- 一括アクションを行うターゲットとして、リソースグループを指定できる。

Run Commandの指定画面

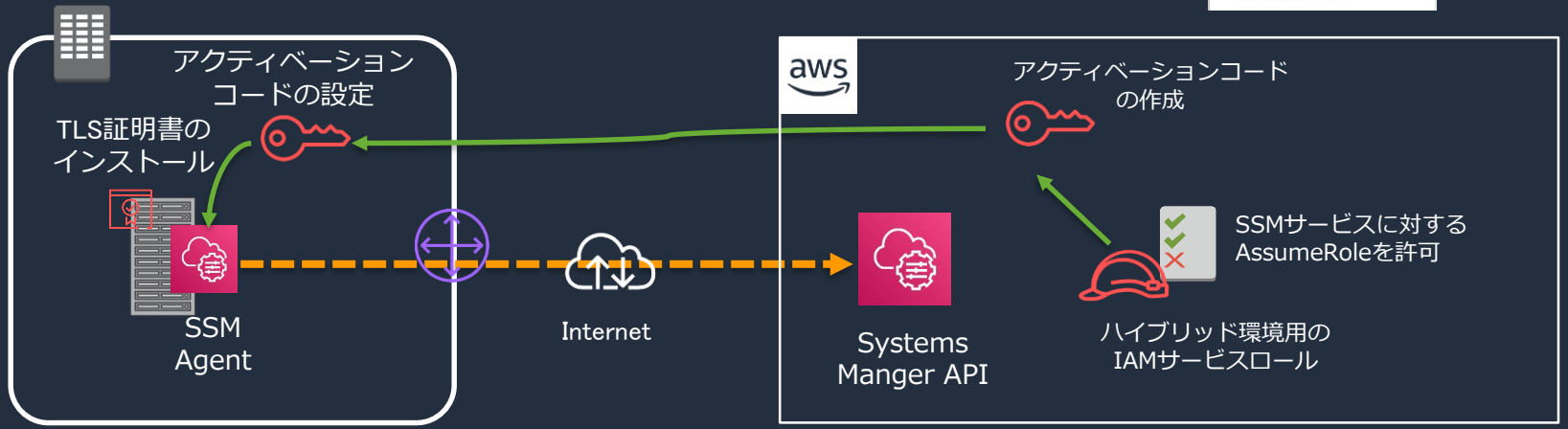
リソースグループを整理しておくと便利

(参考) タグ付けの便利機能　タグエディター

- タグエディターを使用することで、一度に複数のリソースのタグを追加・編集・削除が可能になる。

オンプレミスの場合

1. (Option) TLS証明書のインストール
2. ハイブリッド環境用のIAMロールを作成（初回のみ）
3. SSMでアクティベーションコードを生成
4. インスタンスにアクティベーションコードを設定



詳細は、https://docs.aws.amazon.com/ja_jp/systems-manager/latest/userguide/systems-manager-managedinstances.html

1. 準備編 まとめ

- SSMの管理下におくためは、マネージドインスタンスにする必要がある。
    - そのための３点セット
        - SSMエージェント
        - SSM APIへのアクセス経路
        - EC2ロール
- クイックセットアップでセットアップするのがオススメ
    - 管理されていないインスタンスの排除に有効
    - SSMベストプラクティスに則った管理が可能
- 一括実行の単位となるリソースグループを作成しておくと管理しやすい。
    - タグエディターをうまく使って、インスタンスにタグ定義を
- オンプレミスも管理できる。
    - SSMを使ってEC2もオンプレミスも同じように運用を

アジェンダ

1. AWS Systems Manager 全体像
2. **AWS Systems Managerを使ってみよう**
    1. 準備編
    2. **リソースの”今”を把握しよう**
    3. SSMで定型運用を実施しよう
    4. 非定型なインタラクティブ操作もSSMで
    5. アプリケーションの設定管理もSSMで
3. AWS Systems Managerのセキュリティーベストプラクティス
4. まとめ

2.リソースの”今”を把握しよう

1、**AWSリソース**に関する情報を把握するためのダッシュボード

ご紹介する機能
- SSM Explorer
- SSM OpsCenter
- コンプライアンス

2, **インスタンスの”中身”**を把握するための機能

ご紹介する機能
- SSM インベントリ

2.リソースの”今”を把握しよう

1、**AWSリソース**に関する情報を把握するためのダッシュボード

ご紹介する機能
- SSM Explorer
- SSM OpsCenter
- コンプライアンス

2, **インスタンスの”中身”**を把握するための機能

ご紹介する機能
- SSM インベントリ

AWSリソースに関する情報を把握する (デモ)

- みなさまが運用担当者なら・・・
    - 朝出社して、まずインスタンスの状況を確認
    - そして、何か問題が起きていないか、確認
    - 問題が起きているようだと、その詳細を確認
    - 必要に応じて修復のためのアクションを実施

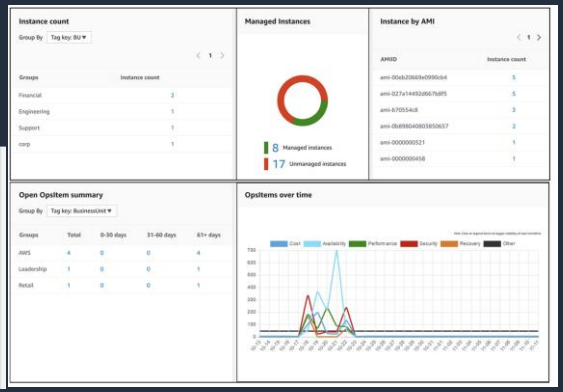
AWS SSM Explorer

AWSリソースに関する情報をレポートするオペレーションダッシュボード

NEW

2019/11～

- クロスアカウント、クロスリージョンで、”今”のリソース状況を可視化できる。
  - Explorerでは、一つ一つのオペレーションデータを”OpsData”と呼ぶ。
  - Explorerは、アカウントおよびリージョン全体のOpsDataの集約ビュー
  - クロスアカウントは AWS Organizationsが前提
- デフォルトのダッシュボードに表示されるOpsData
  - EC2情報
    - EC2インスタンス数
    - マネージドインスタンス数
    - AMI別インスタンス
  - OpsCenter OpsItems
  - パッチコンプライアンス



詳細は、https://docs.aws.amazon.com/ja_jp/systems-manager/latest/userguide/Explorer.html

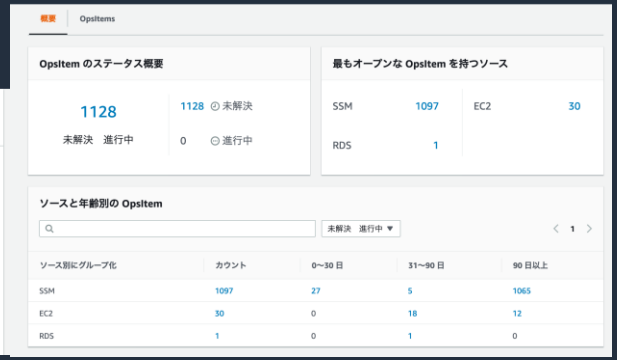
AWS SSM OpsCenter

AWSリソースに関連する運用作業項目 (OpsItems) を表示、調査、解決できるダッシュボード

NEW

2019/06～

- 運用作業項目 (OpsItems) を表示、調査、解決できるダッシュボード
  - サマリは、Explorerのダッシュボードにも表示される。
- OpsItemsに対して修復を行ったり、対応の完了を記録してクローズするなど、運用タスク管理に利用できる。
- CloudWatch Eventsのルールとして登録する。
  - デフォルトでEC2やRDS、SSMなどのイベントが登録済み
  - Amazon EventBridgeと連携でき、外部アラートも登録することが可能

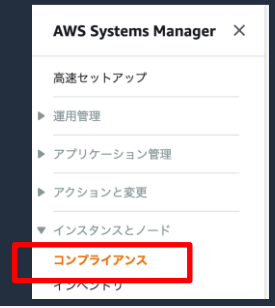


詳細は、https://docs.aws.amazon.com/ja_jp/systems-manager/latest/userguide/OpsCenter.html

AWS SSM コンプライアンス

コンプライアンスに準拠していないリソースを表示できるダッシュボード

- コンプライアンス準拠していないリソースを特定できるダッシュボード
- デフォルトでは、以下がコンプライアンスとして定義済み
  - パッチ適用状況(Patch)
    - SSM パッチマネージャーのScan結果を集計
    - SSM Explorerのダッシュボードにも
  - ステートマネージャの関連づけ状況(Association)
    - SSM ステートマネージャの稼働状況を集計
- カスタムコンプライアンスタイプの定義も可能
  - 例)ソフトウェアXのバージョン4.0以外のインスタンスは非準拠とする。



詳細は、https://docs.aws.amazon.com/ja_jp/systems-manager/latest/userguide/systems-manager-compliance.html

2.リソースの”今”を把握しよう

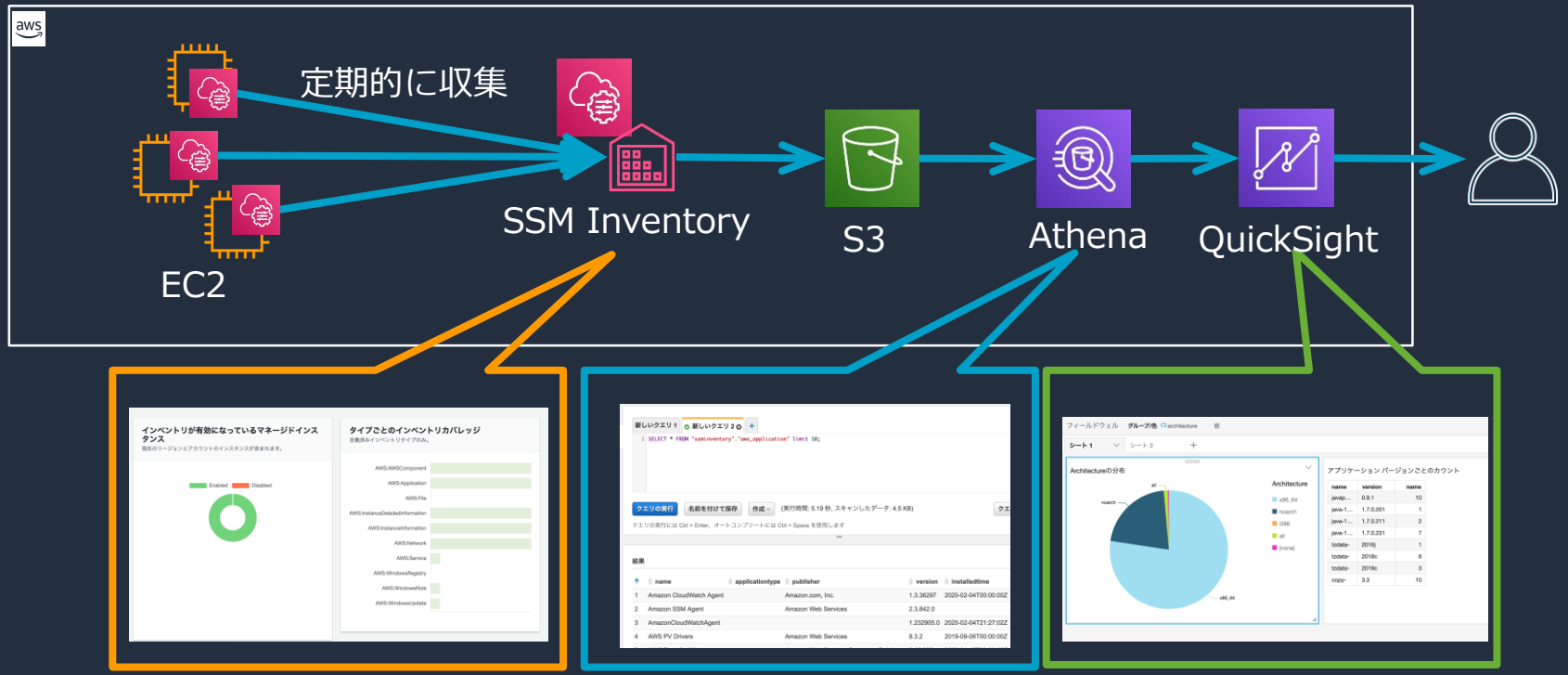
1、**AWSリソース**に関する情報を把握するためのダッシュボード

ご紹介する機能
- SSM Explorer
- SSM OpsCenter
- コンプライアンス

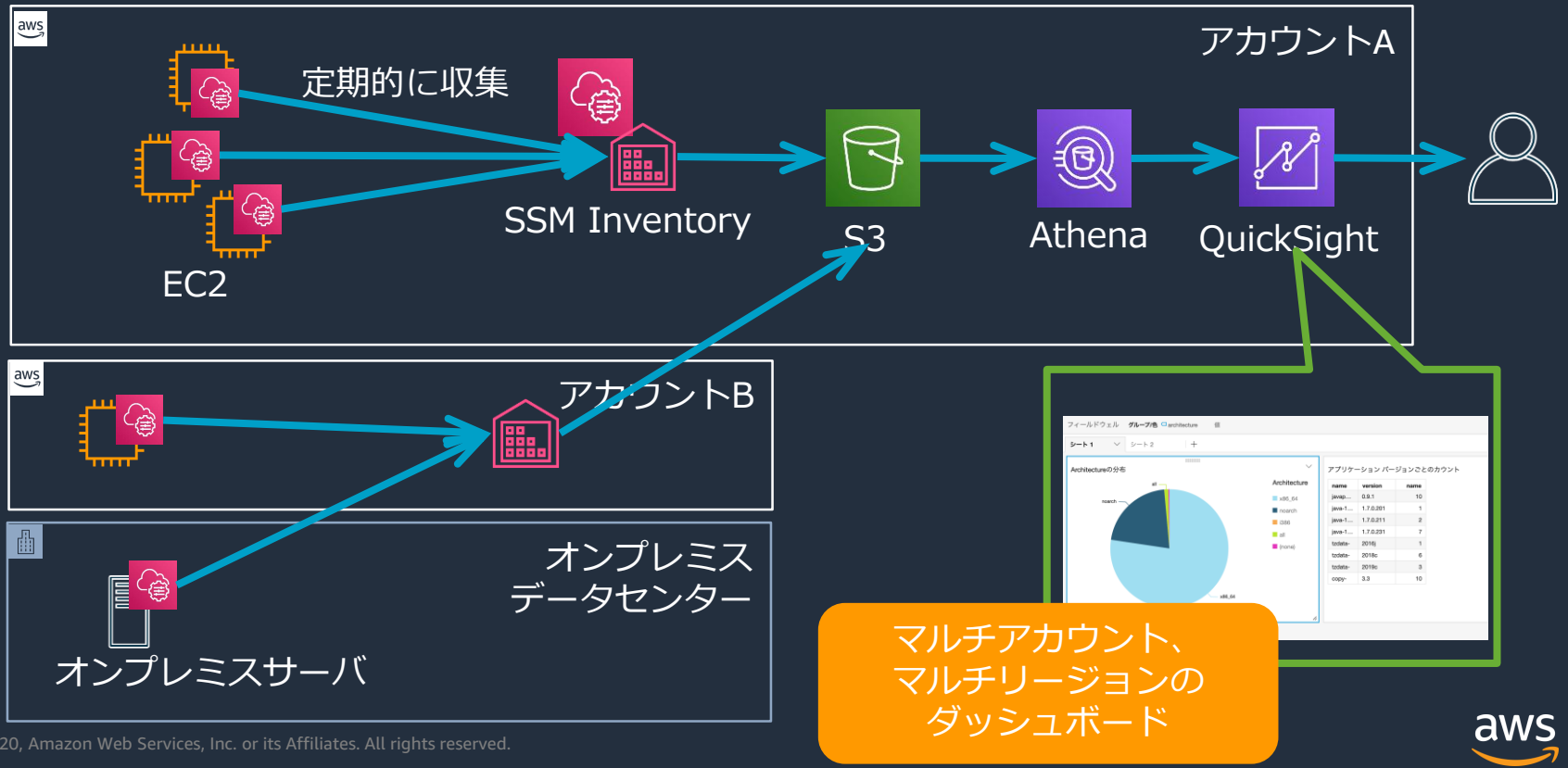
2, **インスタンスの”中身”**を把握するための機能

ご紹介する機能
- SSM インベントリ

デモの流れ



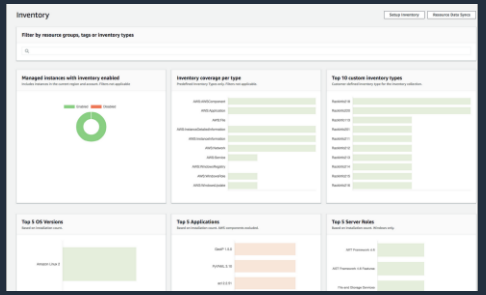
マルチアカウント/マルチリージョンのダッシュボード



AWS SSM インベントリ

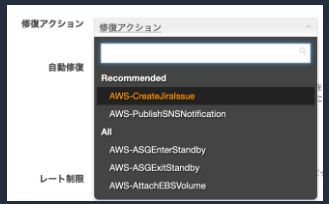
マネージドインスタンスからメタデータを収集し可視化

- OS上のアプリケーション一覧など構成情報を記録し、可視化する。
- ステートマネージャーを使用して定期的に収集
  - クイックセットアップにてセットアップできる
  - 構成情報データはS3バケットに保管
- Athena, QuickSightを用いて
  マルチアカウント/マルチリージョン横断分析も



- AWS Configに構成情報を送信し、
  インベントリ情報の変更追跡が可能

- Config Rulesで準拠状況をチェック
  - 修復アクションで自動対応も



詳細は、https://docs.aws.amazon.com/ja_jp/systems-manager/latest/userguide/systems-manager-inventory.html

AWS SSM インベントリ 

マネージドインスタンスからメタデータを収集し可視化

- インベントリで収集できるメタデータタイプの一覧

取得できる情報	詳細
アプリケーション	アプリケーション名、発行元、バージョンなど。
AWS コンポーネント	EC2 ドライバ、エージェント、バージョンなど
ファイル	名前、サイズ、バージョン、インストール日、変更および最新アクセス時間など パス(C:¥Program Files など)、パターン(*.exe, *.logなど)を指定し、再帰的に抽出できる
ネットワーク設定の詳細	IP アドレス、MAC アドレス、DNS、ゲートウェイ、サブネットマスクなど
Windows アップデート (Winのみ)	Windows Updateに関する情報 (Hotfix ID、インストール者、インストール日など)
インスタンスの詳細	OS名、OSバージョン、最終起動、DNS、ドメイン、ワークグループ、OS アーキテクチャなど
Windows サービス (Winのみ)	名前、表示名、ステータス、依存サービス、サービスのタイプ、起動タイプなど
タグ	インスタンスに割り当てられているタグ
Windows レジストリ (Winのみ)	レジストリキーのパス、値の名前、値タイプおよび値
Windows ロール (Winのみ)	名前、表示名、パス、機能タイプ、インストール日など
カスタムインベントリ	カスタムに割り当てられるメタデータ。例えばオンプレミスの各インスタンスのラック位置など。

- 上記のほか、SSM 設定コンプライアンスで取得されるパッチコンプライアンス、関連づけコンプライアンス情報も、インベントリとして保存される。

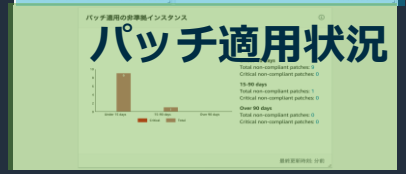
aws

2.リソースの”今”を把握しよう　まとめ

1、AWS リソースに関する情報は **AWS Explorer**で。
マルチアカウント/マルチリージョン

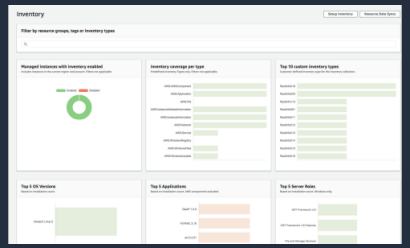
・インスタンス数
・マネージドインスタンス
・AMI別インスタンス

詳細は、**SSM OpsCenter**へ
・運用タスクリスト
・関連リソースの調査
・修復アクションの実行



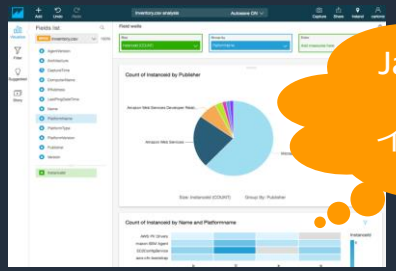
詳細は、**コンプライアンス**へ
・パッチ適用状況
・ステートマネージャ適用状況
・その他コンプラ定義準拠状況

2, インスタンスの”中身”は **SSM インベントリ** で。



・インストール済アプリ
・ファイル情報
・OS情報　など

Athena/QuickSightと連携し
マルチアカウント/マルチリージョン
分析が可能



Java 6がインストールされている
インスタンス一覧は？

アジェンダ

1. AWS Systems Manager 全体像
2. **AWS Systems Managerを使ってみよう**
    1. 準備編
    2. リソースの”今”を把握しよう
    3. **SSMで定型運用を実施しよう**
    4. 非定型なインタラクティブ操作もSSMで
    5. アプリケーションの設定管理もSSMで
3. AWS Systems Managerのセキュリティーベストプラクティス
4. まとめ

SSMでできる定型作業の整理

1、SSMでは、運用処理をSSMドキュメントにて定義し、実行する。

- 汎用的な処理は、事前定義されたドキュメントあり
- カスタマイズした処理を実現したい場合は、ドキュメントを自作する。

事前定義ドキュメント

自作ドキュメント

共有されたドキュメント

SSMドキュメント

実体はJSON or YAML

2、事前定義ドキュメントの中でも、需要が多く複雑な処理は、ドキュメントの実行フレームワークをSSMの機能として提供

	処理内容	実行するSSMドキュメント	実行フレームワーク
1	サーバの構成情報の収集	AWS-GatherSoftwareInventory	SSM インベントリ
2	パッチ適用プロセスの自動化	AWS-RunPatchBaseline	SSM パッチマネージャー
3	ソフトウェアパッケージの配布	AWS-ConfigureAWSPackage	SSM ディストリビューター

詳細は、https://docs.aws.amazon.com/ja_jp/systems-manager/latest/userguide/sysman-ssm-docs.html

SSMでできる定型作業の整理

1、SSMでは、運用処理をSSMドキュメントにて定義し、実行する。

- 汎用的な処理は、事前定義されたドキュメントあり
- カスタマイズした処理を実現したい場合は、ドキュメントを自作する。

事前定義ドキュメント

自作ドキュメント

共有されたドキュメント

SSMドキュメント

実体はJSON or YAML

2、事前定義ドキュメントの中でも、需要が多く複雑な処理は、ドキュメントの実行フレームワークをSSMの機能として提供

	処理内容	実行するSSMドキュメント	実行フレームワーク
1	サーバの構成情報の収集	AWS-GatherSoftwareInventory	SSM インベントリ
2	パッチ適用プロセスの自動化	AWS-RunPatchBaseline	SSM パッチマネージャー
3	ソフトウェアパッケージの配布	AWS-ConfigureAWSPackage	SSM ディストリビューター

詳細は、https://docs.aws.amazon.com/ja_jp/systems-manager/latest/userguide/sysman-ssm-docs.html

SSMドキュメントで運用を定義し実行する (デモ)

- 例えば・・・
  - あるタスクの日には、プロジェクトチームがやってくる
  - その日には、プロジェクトチーム用のEC2を立ち上げておきたい

- デモでやること
  1. タスクがある日をカレンダーで定義
  2. カレンダーで実施可否を確認後、EC2を立ち上げ、ステータスチェックをするSSMドキュメントを作成
  3. 作成したSSMドキュメントを実行

aws

デモの流れ

Change Calendar

Document Builder

SSMドキュメント（定義済み）

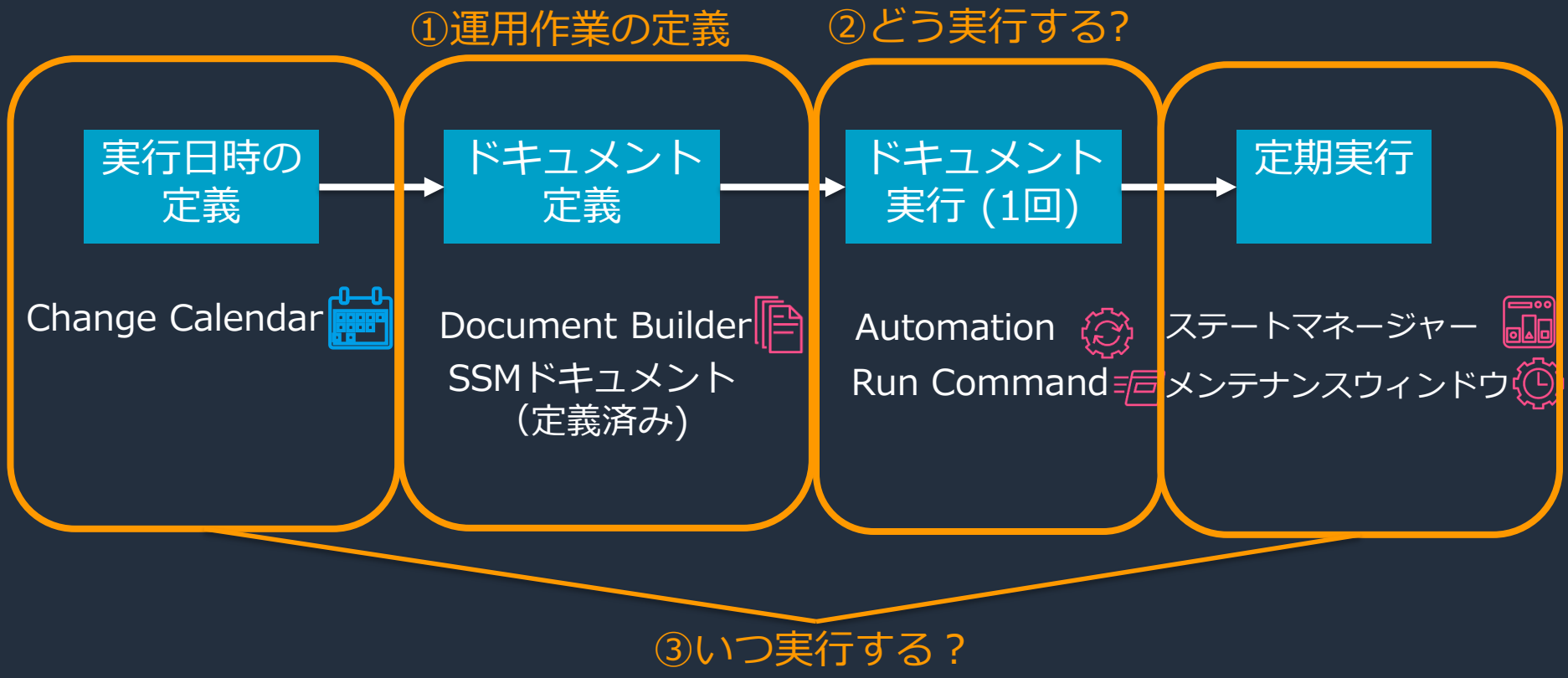
Automation

ステートマネージャー

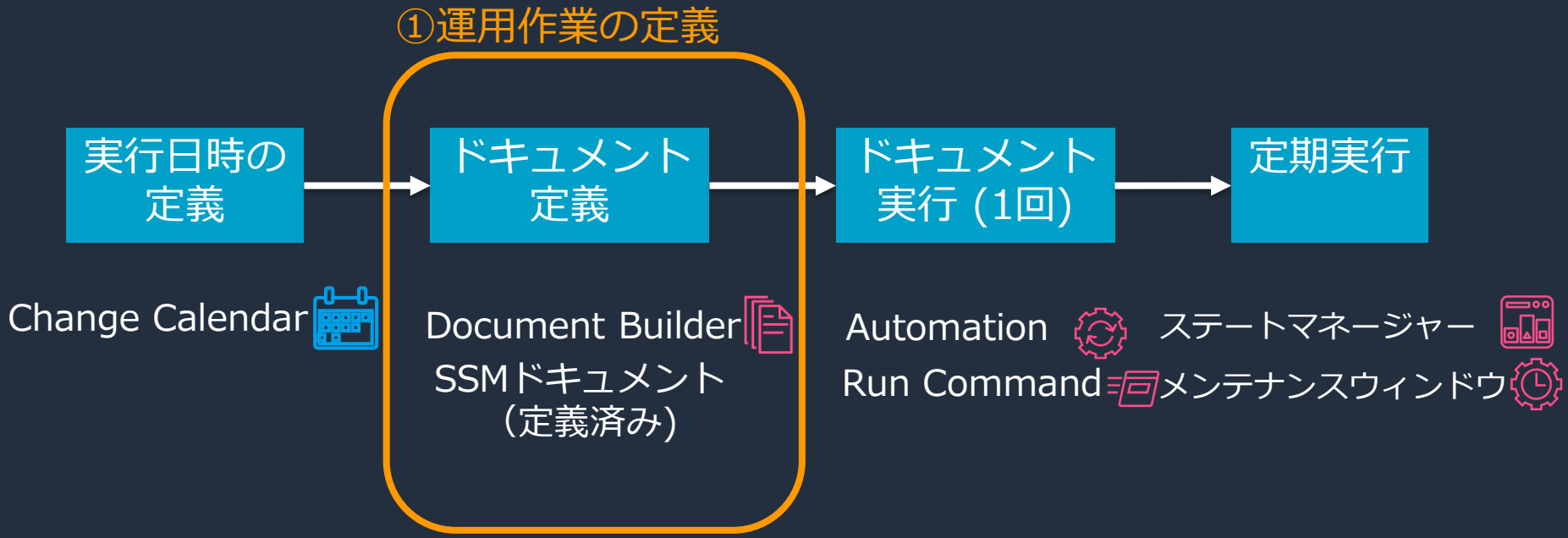
Run Command

メンテナンスウィンドウ

運用作業の定義・実行の流れ



運用作業の定義・実行の流れ

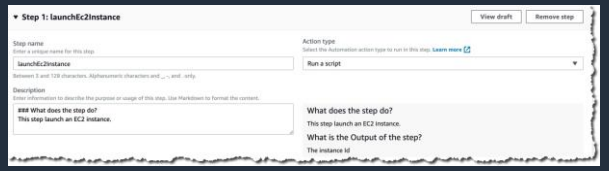


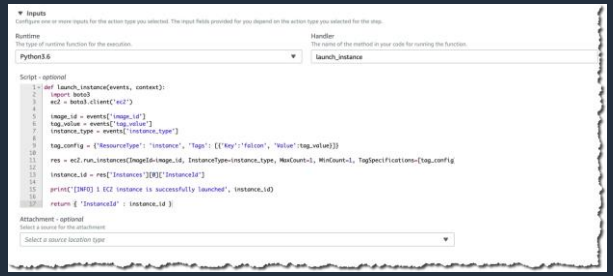
① まずは運用作業の定義 Document Builder

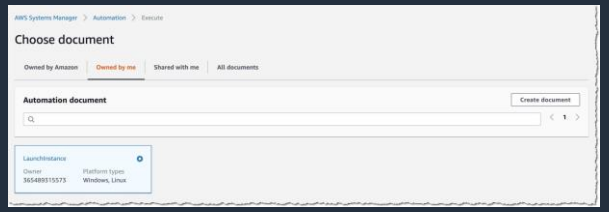
NEW

2019/11～

- 自動化ドキュメント (Automation Playbook) を作成するためのウィザード形式のツール
  - PythonやPowerShellのコードを直接記述することも可能
  - 使い方や目的の説明をMarkdown形式で残すことができる
- 条件分岐を使用した動的ワークフローも可能
- AWSの操作もOS上での操作もこれ一本で記述できるため、運用の自動化がさらに容易に



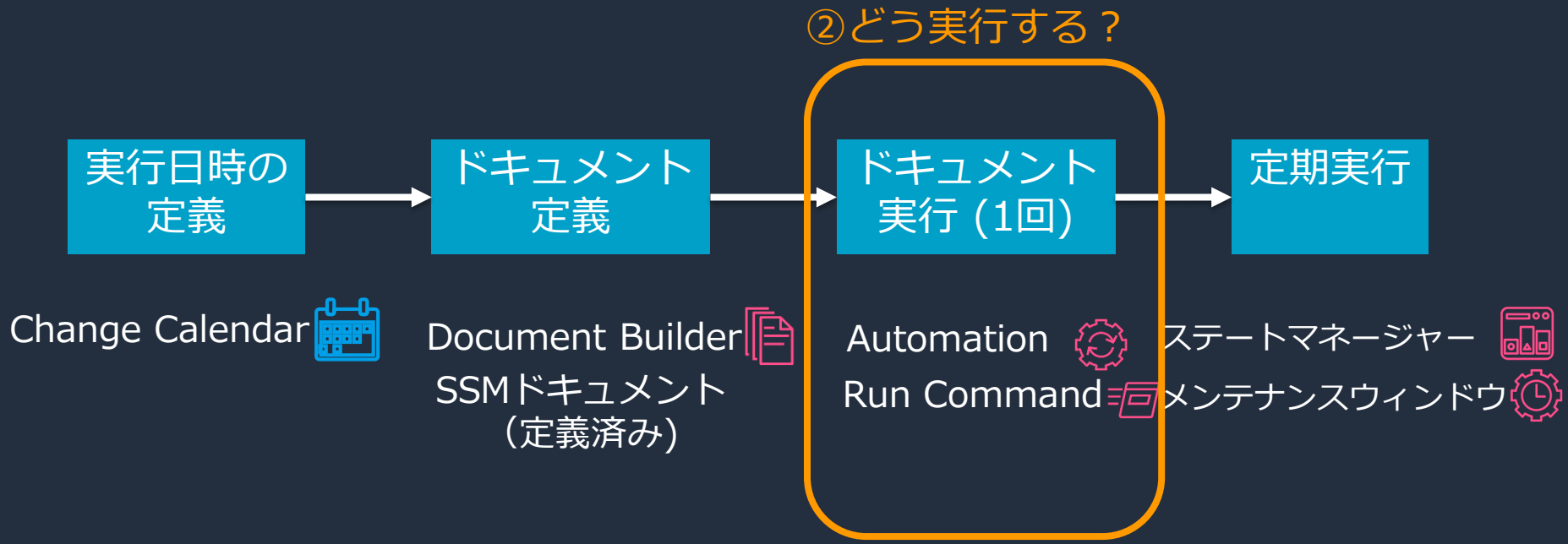




詳細は、https://docs.aws.amazon.com/ja_jp/systems-manager/latest/userguide/automation-document-builder.html

aws

運用作業の定義・実行の流れ

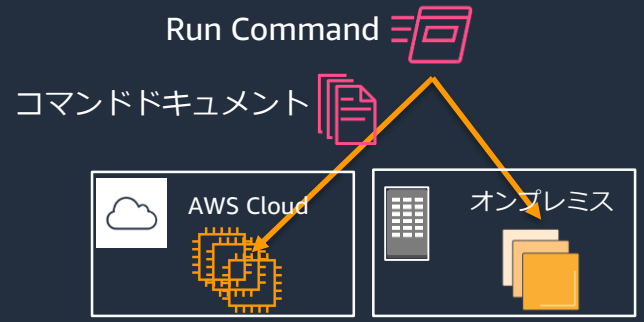


② 定義したもの（SSMドキュメント)をどう実行する？

- Run Command : **OS上でコマンドを実行**

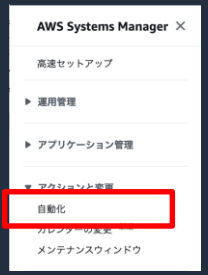
例) ShellScriptの実行、AnsiblePlaybookの実行

- コマンドドキュメントを実行する
- サーバログイン不要
- RDPやSSHのためのインバウンドポート開放不要

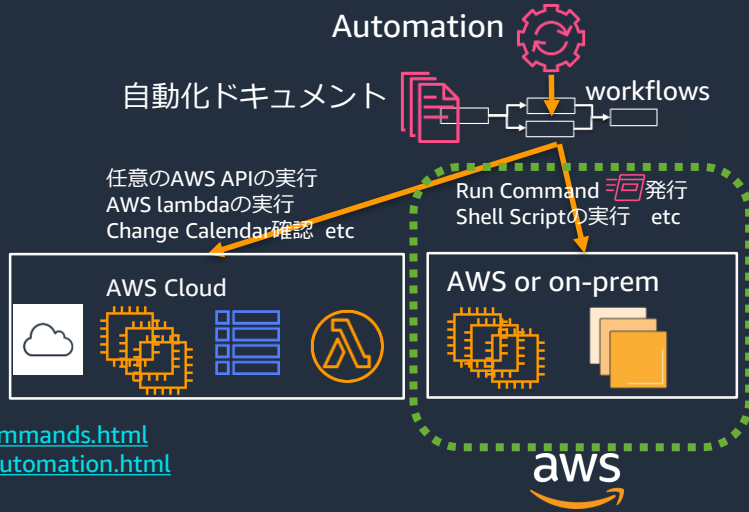


- Automation : **AWSサービス全体に渡ったワークフロー**

例) RDS Snapshot作成、任意のAWS API実行

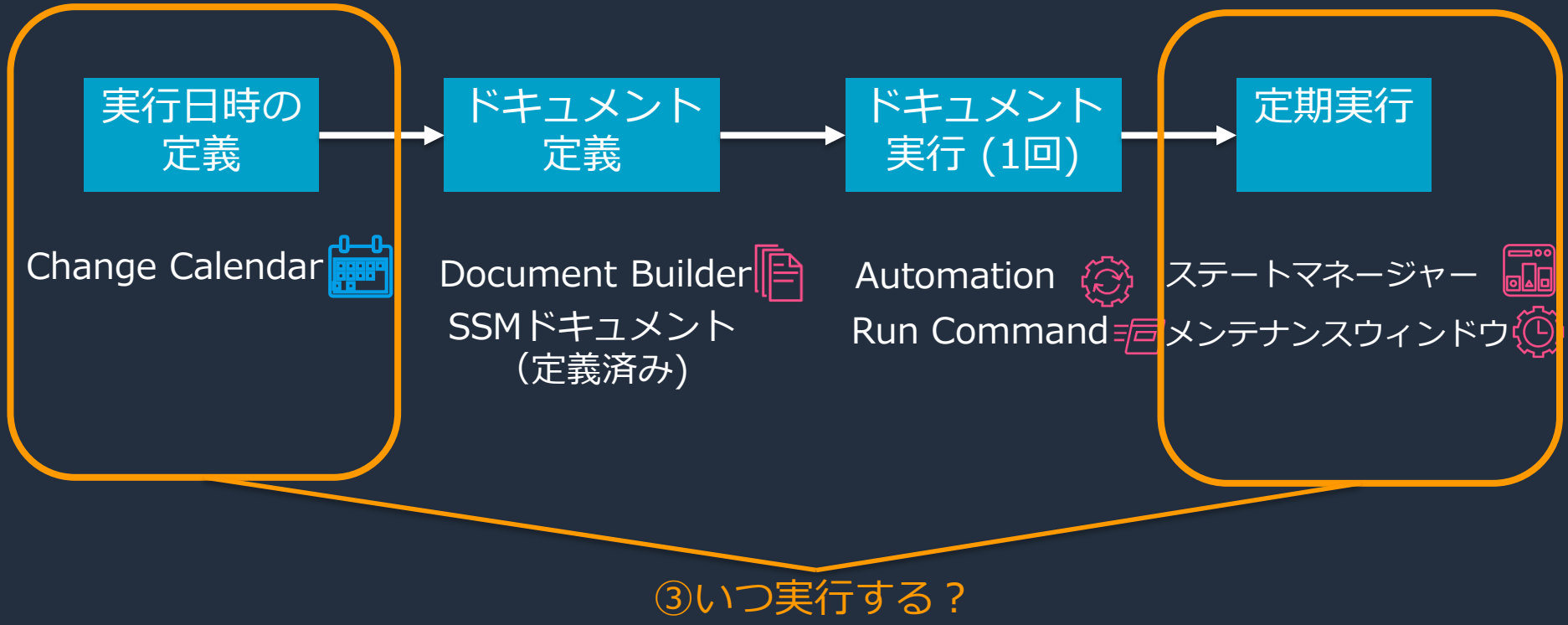


- 自動化ドキュメントを実行する
- AWSの操作もOS上での操作も



詳細は、https://docs.aws.amazon.com/ja_jp/systems-manager/latest/userguide/execute-remote-commands.html
https://docs.aws.amazon.com/ja_jp/systems-manager/latest/userguide/systems-manager-automation.html

運用作業の定義・実行の流れ



③ いつ実行する？

- １度きりの手動実行なら
  - Run Command をそのまま「実行」
  - Automation をそのまま「実行」

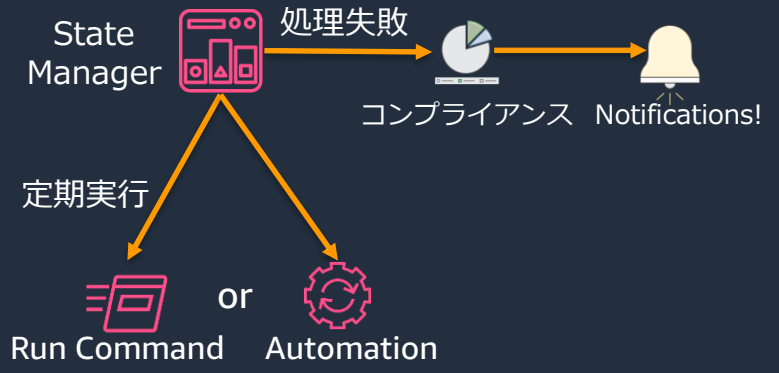
- 繰り返し実行したい定期実行なら
  - ステートマネージャー
  - メンテナンスウィンドウ

- (手動実行でも定期実行でも）実行できる日時を制御するなら
  - Change Calendar

SSM ステートマネージャー

定義された状態に保つプロセスを自動化

- サーバ群に対して定期的に処理を行うためのフレームワーク
  - サーバの状態を確認・是正するための定期的な処理に向く
  - 例) インベントリ収集、SSM Agentの定期更新
  - 処理が失敗すると、求める状態を維持できていないと判断され、コンプライアンスにレポートされる

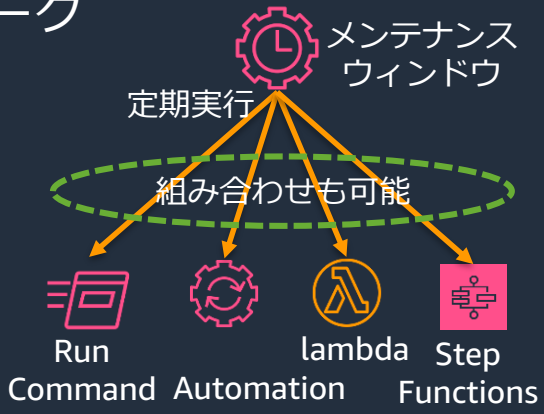


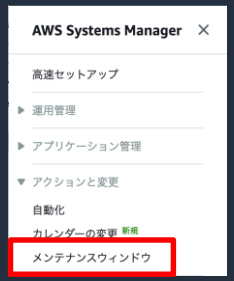
詳細は、https://docs.aws.amazon.com/ja_jp/systems-manager/latest/userguide/systems-manager-state.html

SSM メンテナンスウィンドウ

アクションを実行するスケジュールを定義

- サーバ群に対して定期的に処理を行うためのフレームワーク
  - サービス停止を伴うような比較的重い処理に向く
  - 例) OSパッチ適用、バックアップ取得
  - ステートマネージャーと比べ、精緻な制御が可能
    - 複数のタスク同士の関連性の定義
    - 残り時間がない場合は処理を起動しない etc
  - Lambda、Step Functionsも実行可能





何を：
タスクを複数定義でき
優先順位も指定可能

いつ：
ウィンドウの開始時間、長さ、
タスク実行開始を許可する
残り時間を指定可能

詳細は、https://docs.aws.amazon.com/ja_jp/systems-manager/latest/userguide/systems-manager-maintenance.html

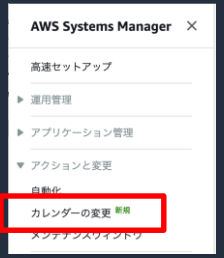
SSM Change Calendar

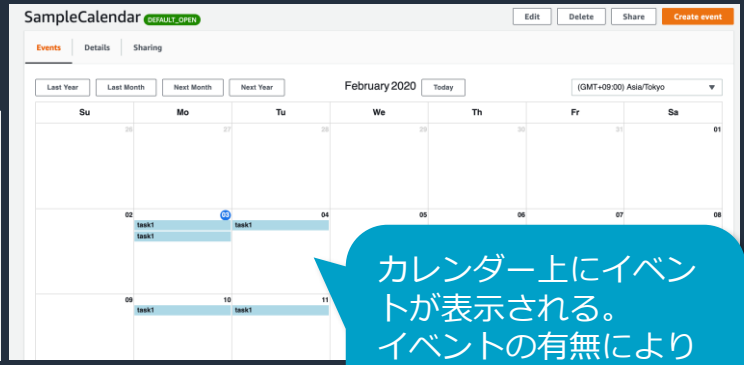
指定したアクションが実行できるまたはできない日付と時刻の範囲を設定

NEW

2019/12～

- システム内で利用するカレンダー情報を集中管理するサービス
- 実行可否の判定結果を提供する (Open/Closed)
- Calendarタイプは２種類
  - Open by default
  - Closed by default
- マルチアカウントでの共有が可能
- SSM Automationには統合ずみ
  他のサービスとも統合を予定



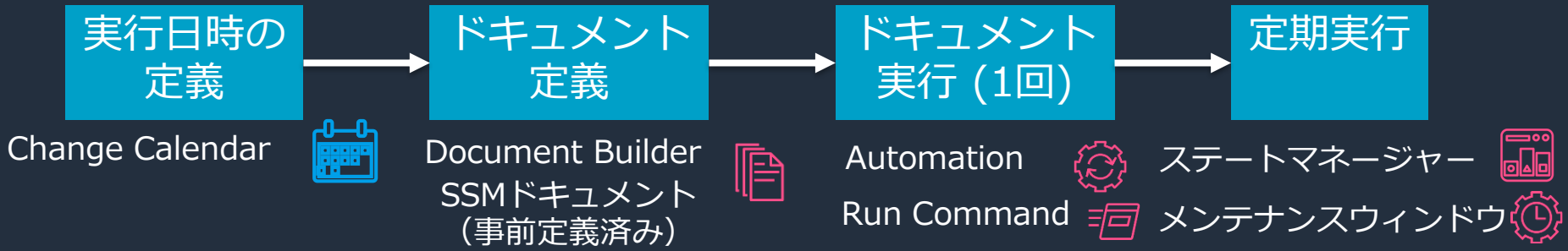


カレンダー上にイベントが表示される。
イベントの有無により「Open」「Closed」が判定される。

詳細は、https://docs.aws.amazon.com/ja_jp/systems-manager/latest/userguide/systems-manager-change-calendar.html

SSMでできる定型作業の整理

1、運用処理をSSMドキュメントにて定義し、実行する。



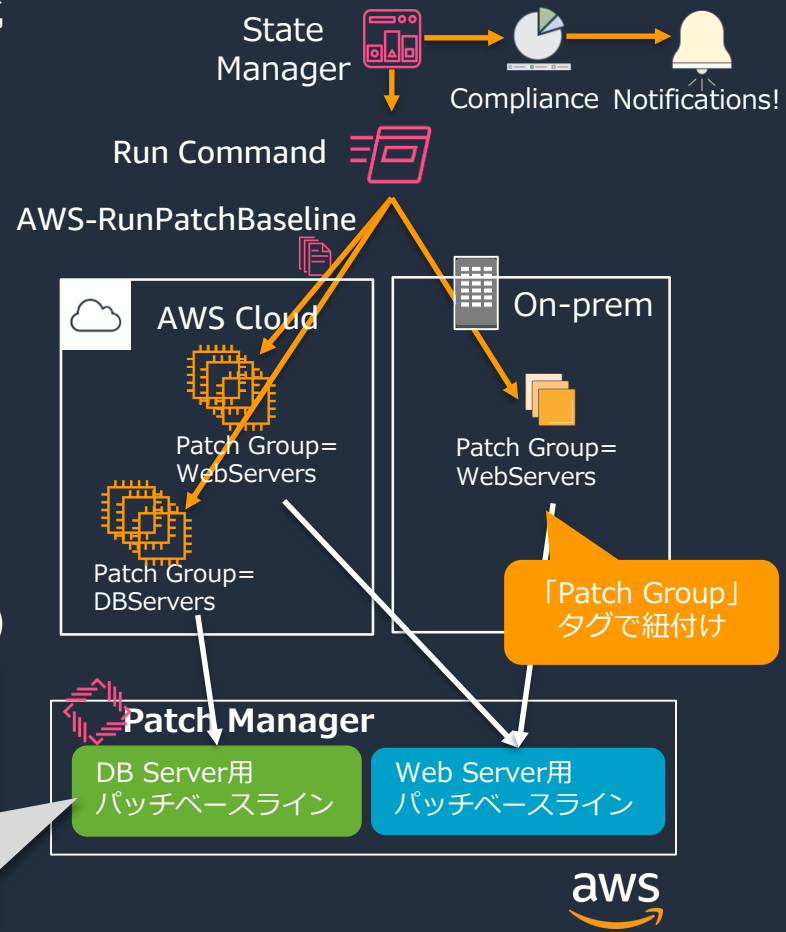
2、SSMの機能として、ドキュメントの実行フレームワークが提供されている処理を実行する。

	処理内容	実行するSSMドキュメント	実行フレームワーク
1	サーバの構成情報の収集	AWS-GatherSoftwareInventory	SSM インベントリ
2	パッチ適用プロセスの自動化	AWS-RunPatchBaseline	SSM パッチマネージャー
3	ソフトウェアパッケージの配布	AWS-ConfigureAWSPackage	SSM ディストリビューター

AWS SSM パッチマネージャー

マネージドインスタンスにパッチを適用するプロセスを自動化

- パッチルール準拠の確認、インスタンスへのパッチ適用が可能
- Scanのみ | Scan & Installの２通り
  - Scanの定期実行はクイックセットアップで設定され、結果を設定コンプライアンスにレポート
- パッチベースラインをOSの種類ごとに作成
  - パッチ適用ルール
  - OS+用途で分けるなど、複数作成可能
- パッチベースラインは「Patch Group」タグ(固定)で紐付け



パッチベースラインの例

OS: Windows
製品：Windows Server 2016
分類：Security Update
重要度：Critical
自動承認の遅延：7日
承認済みパッチ：KB111111
拒否済みパッチ：KB222222

AWS SSM パッチマネージャー

マネージドインスタンスにパッチを適用するプロセスを自動化

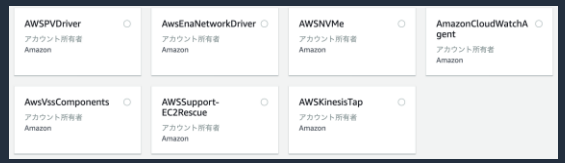
- パッチマネージャー その他
  - インスタンスに指定されたパッチダウンロードサイトへのアクセス経路の確保が必要
    - Windowsインスタンスは、Microsoft Windows Update サイトにアクセスできること
      - プライベートネットワーク内のWSUSサーバをレポジトリに構成することも可能
    - Linuxインスタンスは、インスタンスに設定されたリポジトリへ接続できること
  - パッチ適用後の再起動は、NoRebootオプションでタイミングを制御可能 NEW 2020/01～
  - パッチ自動承認のタイミング指定は以下の２通りから選択
    - パッチがリリースされてからX日経過したらパッチを承認する
    - 特定の日付までにリリースされたパッチを承認する NEW 2020/02～
  - パッチマネージャーがサポートするOSは、SSMサポートOSと異なるので注意
    https://docs.aws.amazon.com/ja_jp/systems-manager/latest/userguide/patch-manager-prerequisites.html

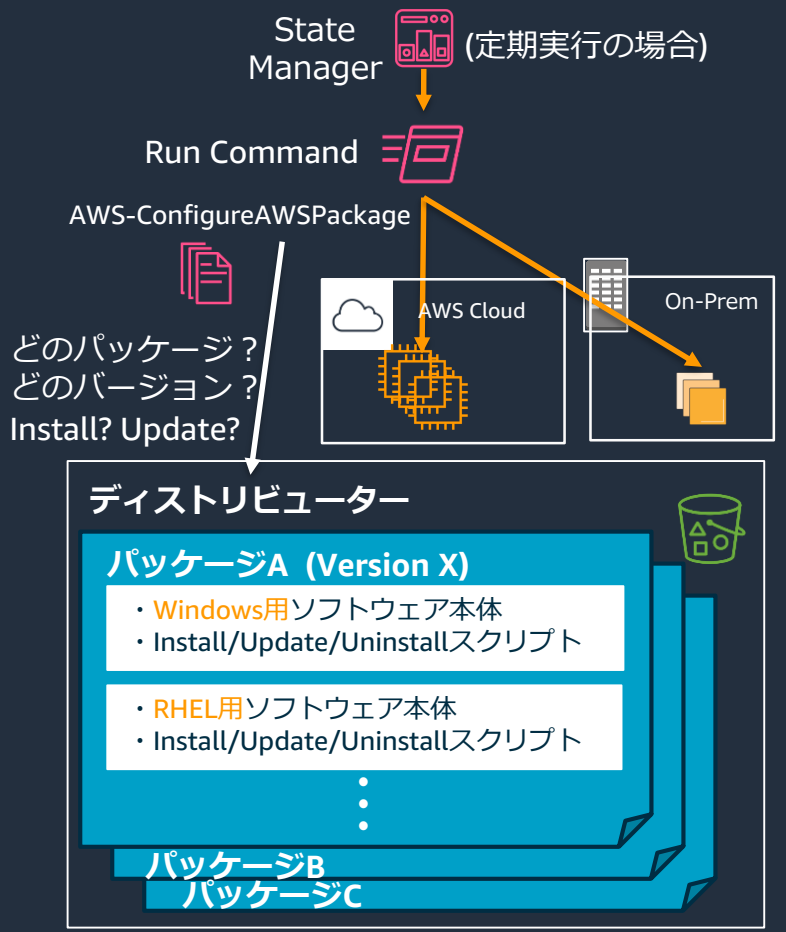
詳細は、https://docs.aws.amazon.com/ja_jp/systems-manager/latest/userguide/systems-manager-patch.html

AWS SSM ディストリビューター

ソフトウェアパッケージを安全に保存し配信

- 独自のソフトウェアパッケージの配布、インストールが可能
  - 指定したサーバ群への配布（一回・定期）
  - 複数のプラットフォームに対応
  - 配布パッケージのバージョン管理
- パッケージは他アカウントへ共有可能
- AWSの各種のパッケージが事前定義されておりその導入・更新にも有効



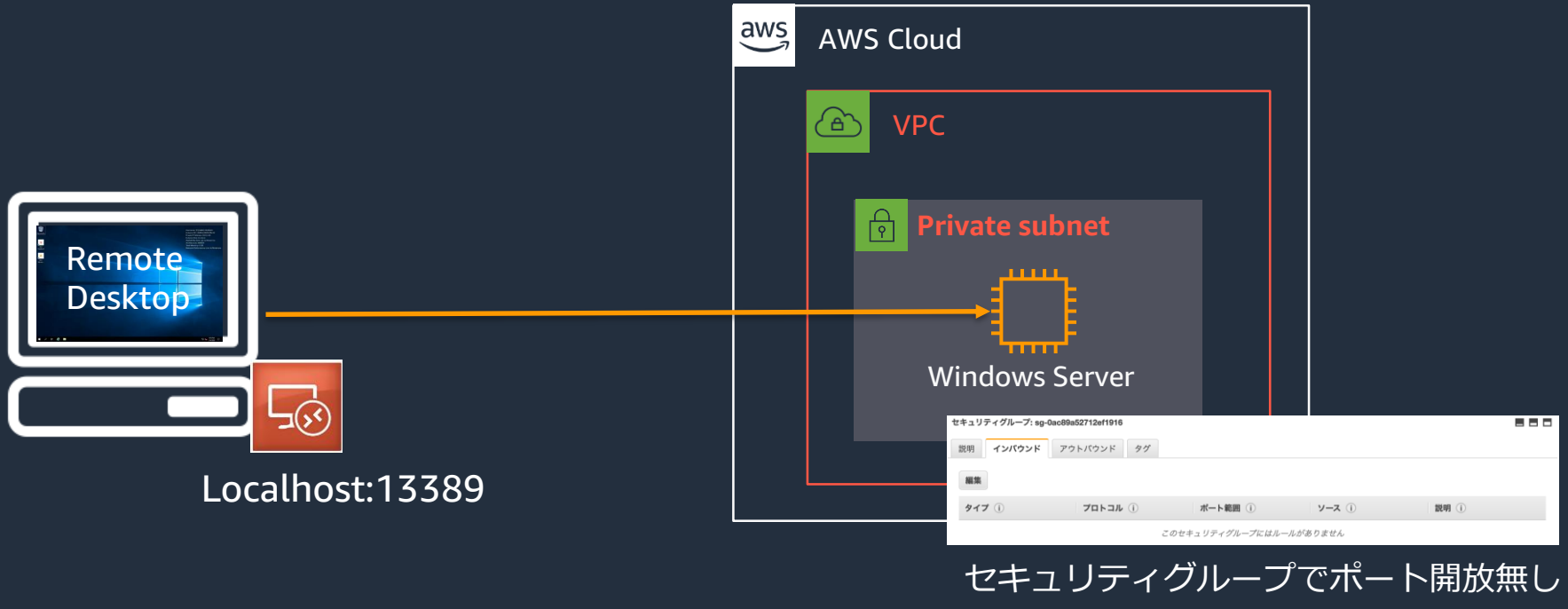


詳細は、https://docs.aws.amazon.com/ja_jp/systems-manager/latest/userguide/distributor.html

アジェンダ

1. AWS Systems Manager 全体像
2. **AWS Systems Managerを使ってみよう**
    1. 準備編
    2. リソースの”今”を把握しよう
    3. SSMで定型運用を実施しよう
    4. **非定型なインタラクティブ操作もSSMで**
    5. アプリケーションの設定管理もSSMで
3. AWS Systems Managerのセキュリティーベストプラクティス
4. まとめ

セッションマネージャー RDPアクセス



セキュリティグループでポート開放無し

SSM セッションマネージャー

インバウンドポートを開くことなく、インタラクティブなシェルアクセスを実現

- 通信ポートを開放せずにサーバへのシェルアクセスが可能
  - セキュリティグループでの通信ポートの穴あけ不要。インスタンスをセキュアに維持。
  - プライベートサブネットのインスタンスにもアクセス可能。踏み台サーバいらずに。
- アクセス制御はIAMユーザに対しIAM Policyで指定する。
- セッションマネージャーで用意されている接続手段

  1、SSM Agent 経由で直接アクセス

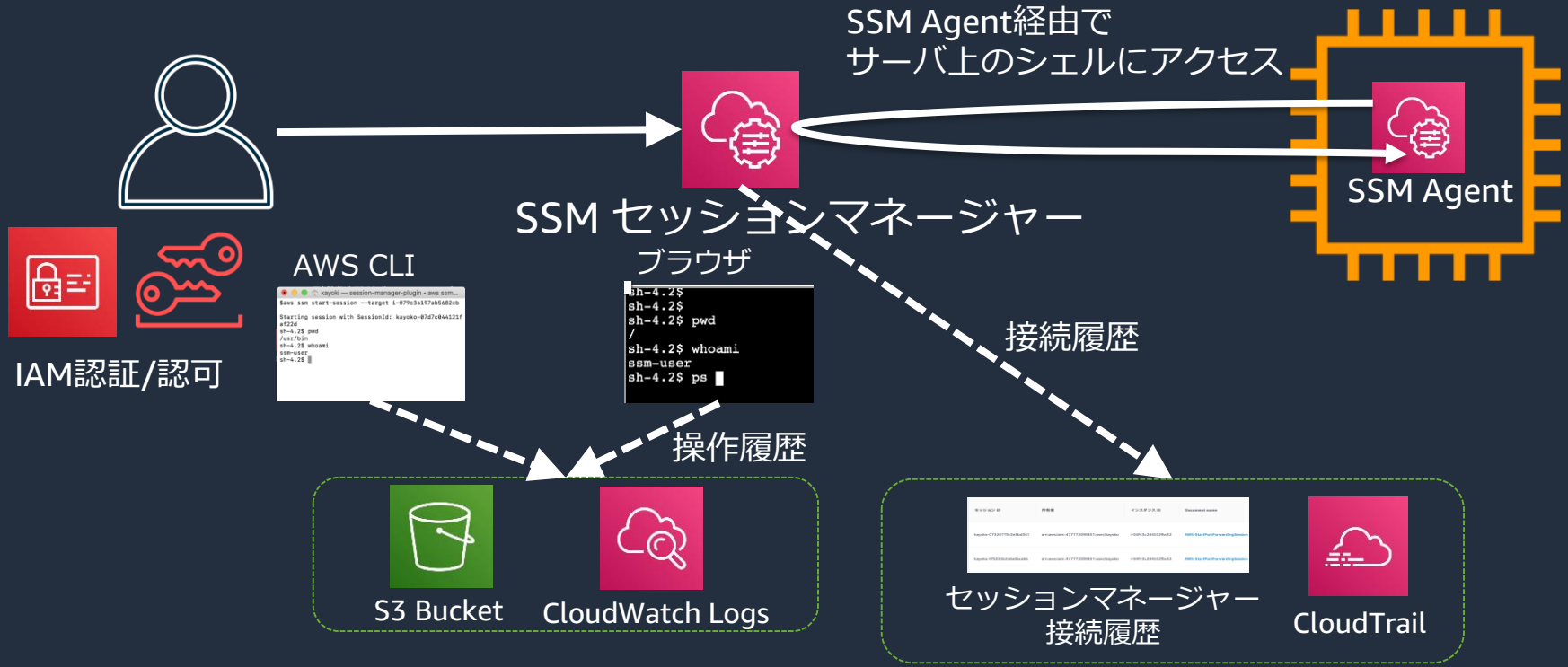
  2、SSM Agent でトンネルを作成してSSHなどでアクセス

NEW

2019/7

詳細は、https://docs.aws.amazon.com/ja_jp/systems-manager/latest/userguide/session-manager.html

セッションマネージャー　SSM Agent経由で直接アクセス



セッションマネージャー　SSM Agent経由で直接アクセス

- ブラウザのみでインタラクティブなシェルアクセスを実現可能
  - サーバのログイン情報（キーペアおよびID・パスワード）が不要（IAM認証）
  - Linux は bash、Windows は PowerShellが利用可能
- その他
  - 操作ログを CloudWatch Logs や S3 に保存。暗号化も可能。
  - セッションマネージャーでの接続履歴や、CloudTrailにて接続情報を追跡可能
  - Linuxはセッションを開始するOSユーザを設定可能。（デフォルトは ssm-user）
  - インスタンスでSSHを起動させる必要はない。ポート穴あけも不要。
  - AWS CLIからアクセスすることも可能。session-manager-pluginの導入要

```
$ aws ssm start-session --target i-079c3a197ab5682cb

Starting session with SessionId: kayoko-024e90a532f59ad5e
sh-4.2$
```

セッションマネージャー　トンネリングアクセス (RDP接続)

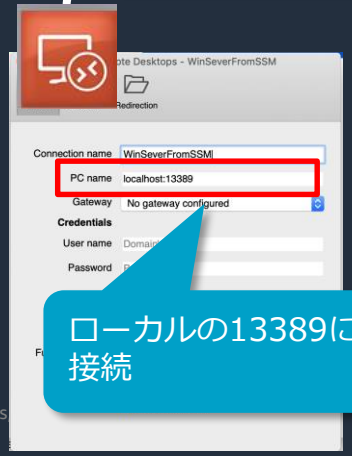
```
$ aws ssm start-session --target i-04f43c284532fbc32 --document-name AWS-StartPortForwardingSession --parameters
"portNumber=3389, localPortNumber=13389"

Starting session with SessionId: kayoko-0f66a98202044da39
Port 13389 opened for sessionId kayoko-0f66a98202044da39.
Connection accepted for session kayoko-0f66a98202044da39.
```

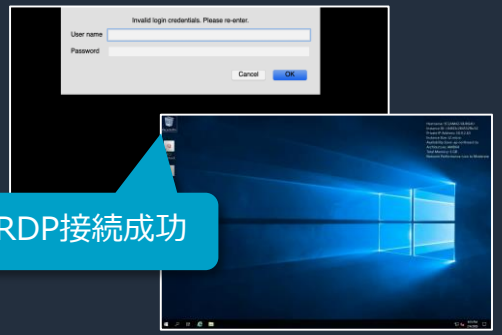
ローカル側の tcp13389 へのアクセスが
リモート側の tcp3389 に転送される

IAM認証/認可

OS認証



ローカルの13389に
接続



RDP接続成功

接続履歴

セッションマネージャー
接続履歴

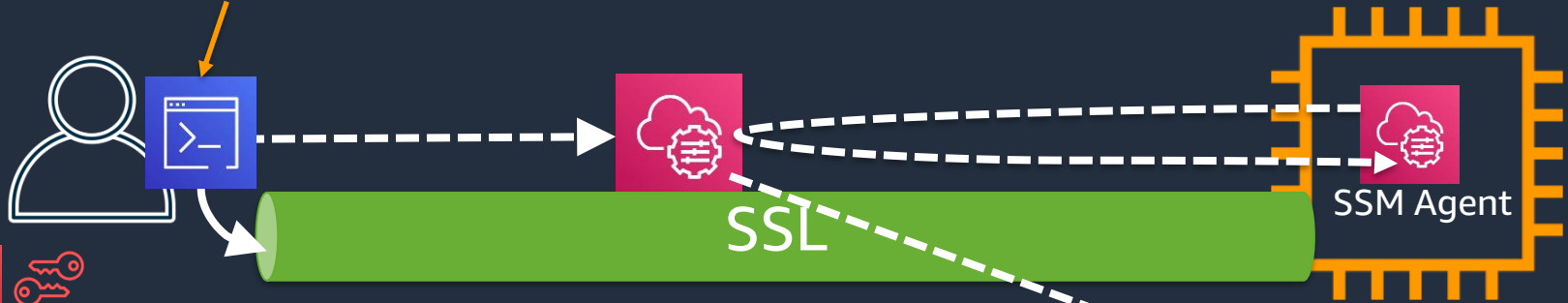
CloudTrail

※操作履歴は取得できない

セッションマネージャー　トンネリングアクセス（SSH接続）

```
$ cat ~/.ssh/config
# SSH over Seesion Manager
host i-* mi-*
ProxyCommand sh -c "aws ssm start-session --target %h --document-name AWS-StartSSHSession --parameters 'portNumber=%p'"
$
```

SSHクライアントにProxy設定を追加



IAM認証/認可

OS認証

```
$ ssh -i TokyoKey.pem ec2-user@i-079c3a197ab5682cb
Last login: Tue Feb  4 20:54:19 2020 from localhost

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|¥___|___|

https://aws.amazon.com/amazon-linux-2/
6 package(s) needed for security, out of 28 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-10-0-2-214 ~]$
```

接続履歴

セッションマネージャー
接続履歴

CloudTrail

※操作履歴は取得できない

セッションマネージャー　トンネリングアクセス

- 使い慣れたSSHクライアントから、SSH/SCPが実現可能
  - プロキシ設定で、AWS CLIのコマンド設定要
- AWS CLIを用いてポートフォワーディングが可能
  - プライベートサブネットにあるRDSに開発端末から接続
  - WindowsインスタンスにするRDP接続 etc

- その他
  - 操作ログは保管されない。従来通り、SSHクライアント側で取得する。
  - IAM認証に加え、サーバのログイン情報（キーペアおよびID・パスワード）が必要。
  - インスタンスで SSH/RDP が実行されている必要がある。
    ただしポート穴あけは不要
  - セッションマネージャーでの接続履歴や、CloudTrailにて接続情報を追跡可能。
  - 利用には、session-manager-pluginの導入が必要

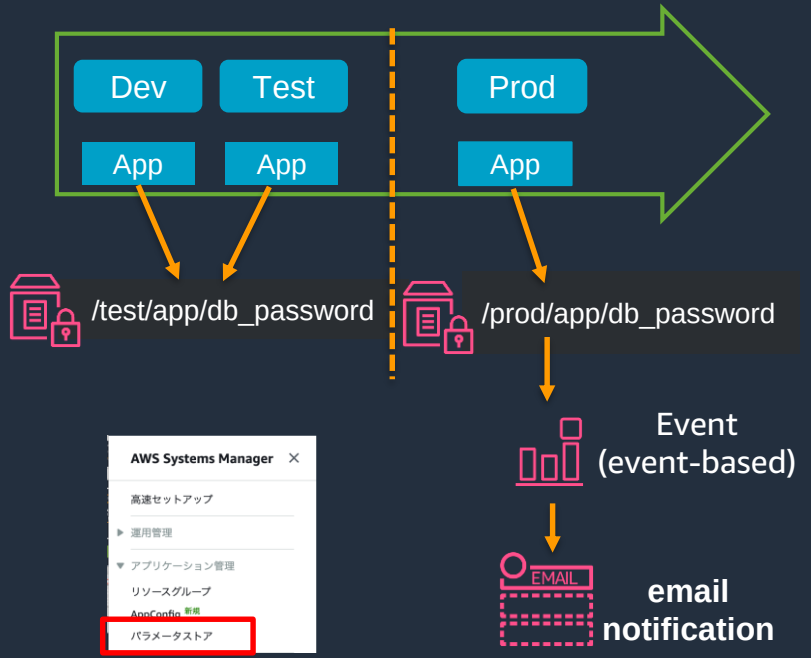
アジェンダ

1. AWS Systems Manager 全体像
2. **AWS Systems Managerを使ってみよう**
    1. 準備編
    2. リソースの”今”を把握しよう
    3. SSMで定型運用を実施しよう
    4. 非定型なインタラクティブ操作もSSMで
    5. **アプリケーションの設定管理もSSMで**
3. AWS Systems Managerのセキュリティーベストプラクティス
4. まとめ

SSM パラメータストア

構成や設定情報の管理のための安全な階層型ストレージ

- コンフィグレーションや設定値を権限別の階層型で保存
  - IAMによるアクセス制御
- パスワードなど機密情報をKMSで暗号化
- パブリックパラメータあり
  - AWSが提供するパラメータ
  - 例) AMI情報
- CloudFormation, Lambda, ECS, CodeBuild, CodeDeployなどのサービスと統合済み
  - 環境変数を渡す用途などに使用



詳細は、https://docs.aws.amazon.com/ja_jp/systems-manager/latest/userguide/systems-manager-parameter-store.html

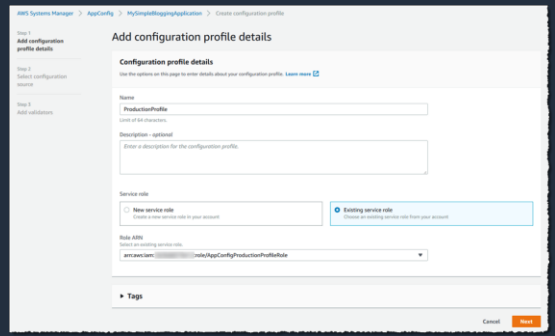
SSM AppConfig

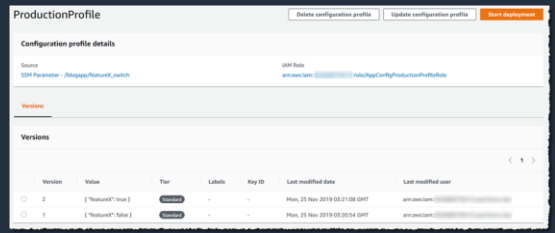
アプリケーション設定を作成、管理し、迅速なデプロイをサポート

NEW

2019/12～

- アプリケーションの設定情報を迅速に展開するための機能
  - EC2、コンテナ、Lambdaへスケーラブルかつアプリケーションの再起動なしに展開可能
- 開発や本番など環境毎に異なる設定情報をデプロイできる
  - 設定情報はパラメータストアもしくはSSMドキュメントとして保管
  - アプリケーションコードから AppConfig の GetConfiguration APIでパラメータを取得。その値で動作を変えるよう開発する。
- 展開前にバリデーションも実施できる
- デプロイ戦略を定義でき、カナリアリリースも可能





詳細は、Https://docs.aws.amazon.com/ja_jp/systems-manager/latest/userguide/appconfig.html

アジェンダ

1. AWS Systems Manager 全体像
2. AWS Systems Managerを使ってみよう
    1. 準備編
    2. リソースの”今”を把握しよう
    3. SSMで定型運用を実施しよう
    4. 非定型なインタラクティブ操作もSSMで
    5. アプリケーションの設定管理もSSMで
3. **AWS Systems Managerのセキュリティーベストプラクティス**
4. まとめ

SSMを使用する上でのセキュリティーベストプラクティス

「**Systems Manager のセキュリティのベストプラクティス**」をご参照ください

https://docs.aws.amazon.com/ja_jp/systems-manager/latest/userguide/security-best-practices.html

- 最小限の特権アクセスを実装する
  - ユーザのIAMポリシーは、該当リソース・特定アクションについてのみ有効に
  - 例えば”ssm.StartSession”をDenyすることで、セッションマネージャを使用しない設定が可能。
- VPCエンドポイントを使用可能
- 特別セキュアな処理が必要な場合はSession Managerに対話型コマンドのみを使用する
- AWSおよびSSMツールを最新に保つ
- CloudWatch / CloudTrail / AWS Configを使用

SSMの料金

- AWS Systems Manager の利用は基本的に無料
- 一部の機能は有料
  - OpsCenter (OpsItem の数とAPI コールの数に基づく課金)
  - Explorer (ダッシュボード表示の際のOpsCenter APIコールのみ課金)
  - パラメータストア（パラメータサイズが4KB以上、パラメータ数10000以上の場合）
  - ディストリビューターの独自パッケージ
  - Automation（ステップカウント、ステップ実行時間、プレイブックに対して課金）
  - AppConfig (APIコールの数とターゲットごとの構成更新の合計数に対してのみ課金)
  - オンプレミス管理のアドバンストインスタンスティア
- その他関連サービスの使用量に応じた料金
  - Athena + QuickSight / Config / CloudWatch （カスタムメトリクス、Logs）/ S3に格納したログデータ

詳細は、https://aws.amazon.com/jp/systems-manager/pricing/

(参考) オンプレミスインスタンス管理

- SSMでは、オンプレミスインスタンス管理用に2つのティアがある。

	標準インスタンスティア (デフォルト)	アドバンストインスタンスティア
課金	無料	インスタンス実行時間に基づく 従量課金
登録できる サーバ数	リージョン/アカウントごとに 最大1000まで	リージョン/アカウントごとに 1000を超えるサーバを登録可能
機能差異	・セッションマネージャーが 使えない ・パッチパネージャーで、Microsoftアプリケーションのパッチ管理ができない	・セッションマネージャーも使用可能 ・パッチパネージャーで、Microsoftアプリケーションのパッチ管理が可能

「マネージドインスタンス」 > 「設定」 > 「インスタンス枠」> 「アカウント設定の変更」からインスタンスティアを変更可能

詳細は、https://docs.aws.amazon.com/ja_jp/systems-manager/latest/userguide/systems-manager-managedinstances-advanced.html

アジェンダ

1. AWS Systems Manager 全体像
2. AWS Systems Managerを使ってみよう
    1. 準備編
    2. リソースの”今”を把握しよう
    3. SSMで定型運用を実施しよう
    4. 非定型なインタラクティブ操作もSSMで
    5. アプリケーションの設定管理もSSMで
3. AWS Systems Managerのセキュリティーベストプラクティス
4. **まとめ**

まとめ

- AWS SSMを用いることで、オンプレミス／AWS両環境で運用に必要な作業を、実施することができます。
    - リソース状況の可視化
    - 定型作業の実施
    - インタラクティブな操作
    - アプリケーションの設定管理
- 何か一つの機能から始めてみてはいかがでしょうか。

Q&A

お答えできなかったご質問については
AWS Japan Blog 「https://aws.amazon.com/jp/blogs/news/」にて
後日掲載します。

AWS の日本語資料の場所「AWS 資料」で検索

https://amzn.to/JPArchive

AWS Well-Architected 個別技術相談会

毎週”W-A個別技術相談会”を実施中

- AWSのソリューションアーキテクト(SA)に対策などを相談することも可能
- 申込みはイベント告知サイトから

(https://aws.amazon.com/jp/about-aws/events/)

AWS イベント で[検索]

ご視聴ありがとうございました

AWS 公式 Webinar
https://amzn.to/JPWebinar

過去資料
https://amzn.to/JPArchive