



AWS IoT Device Management

AWS Black Belt Online Seminar

Kazunari Hattori

Solutions Architect

2023/03

AWS Black Belt Online Seminarとは

- 「サービス別」「ソリューション別」「業種別」などのテーマに分け、アマゾン ウェブ サービス ジャパン合同会社が提供するオンラインセミナーシリーズです
- AWSの技術担当者が、AWSの各サービスやソリューションについてテーマごとに動画を公開します
- 動画を一時停止・スキップすることで、興味がある分野・項目だけの聴講も可能、スキマ時間の学習にもお役立ていただけます
- 以下のURLより、過去のセミナー含めた資料などをダウンロードすることができます
 - <https://aws.amazon.com/jp/aws-jp-introduction/aws-jp-webinar-service-cut/>
 - <https://www.youtube.com/playlist?list=PLzWGOASvSx6FIwIC2X1nObr1KcMCBBBlqY>

内容についての注意点

- 本資料では2023年2月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(<https://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おります。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

自己紹介

名前：服部 一成

所属：エンタープライズ技術本部
ソリューションアーキテクト

趣味：自宅配線

好きなAWSサービス：

AWS IoT Device Management , AWS IoT Core



本セミナーの対象者

- IoT デバイスやサービスの開発者
- IoT デバイスやサービスを運用、管理するご担当者
- AWS IoT Device Management をご利用予定の方

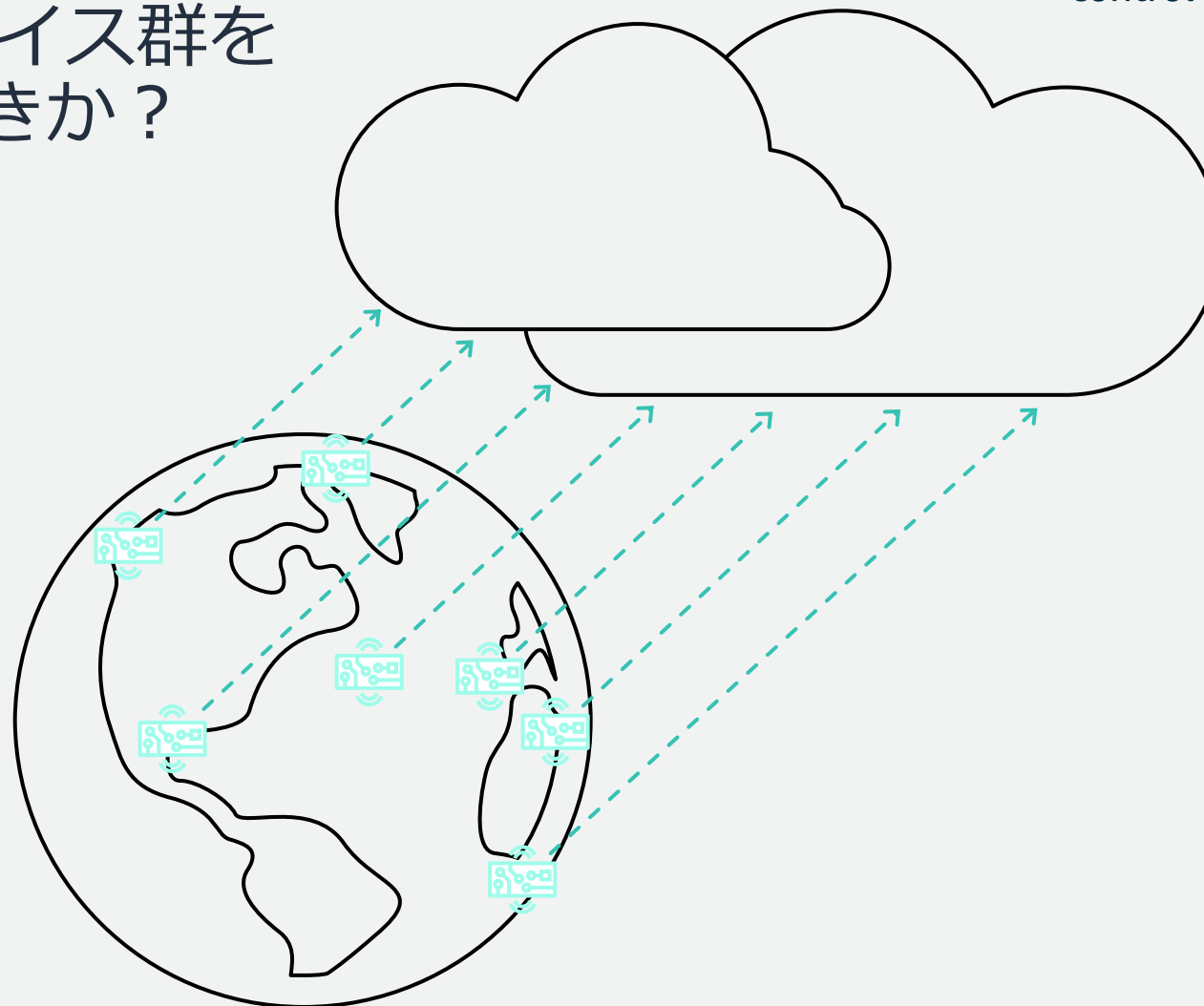
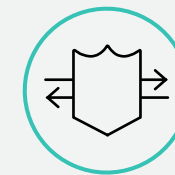
アジェンダ

1. IoT デバイスの管理における課題
2. AWS IoT Device Management とは
3. AWS IoT Device Management の料金体系

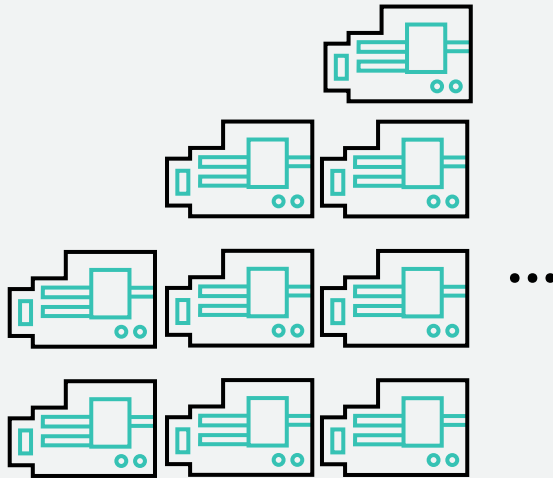
IoT デバイスの管理における課題

増え続ける IoT デバイス群を
どのように管理すべきか？

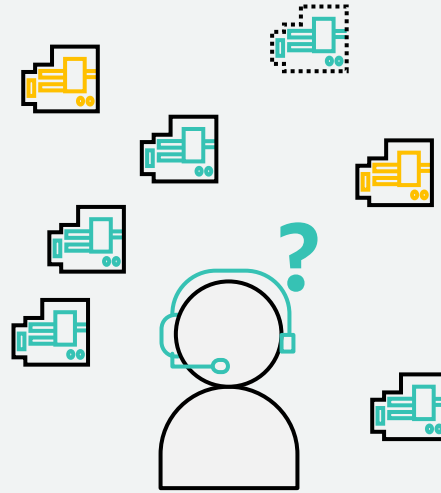
Connectivity and
control services



IoT デバイスの管理におけるお困りごと



増え続けるデバイスの
導入と設定



多数のデバイスの
運用や管理



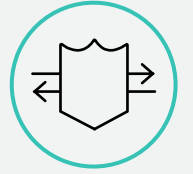
遠隔管理の煩雑さ

AWS IoT Device Management とは

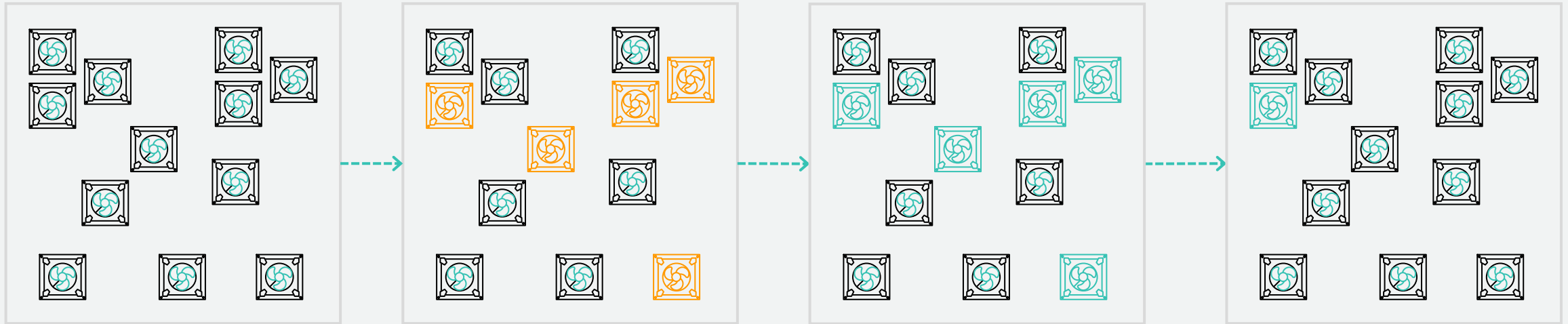


AWS IoT Device Management

Connectivity and control services



AWS IoT Device Management は増加するデバイスの導入、整理、監視、遠隔管理を支援します



迅速でスケーラブルな
デバイスの導入

リアルタイムなデバイスの
インデックス化と検索

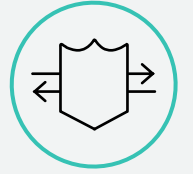
モニタリングと
デバイスのアップデート

個々のデバイスへの
セキュアなアクセス



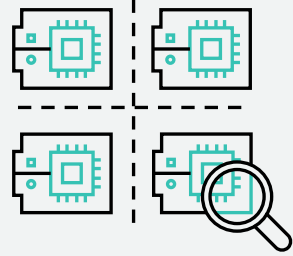
AWS IoT Device Management の機能

Connectivity and control services



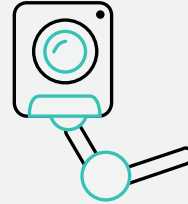
モノの一括登録

数百万のデバイスにあるアプリをデプロイし管理する



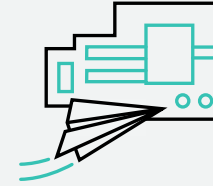
デバイス群のインデックス化と検索

全デバイスのステータスと状態を可視化



デバイスロギングとモニタリング

デバイスログを収集して問題を素早く特定、修正



ジョブ実行

グループ化されたデバイスへジョブを実行し管理する



セキュアトンネリング

ファイアウォール越しのデバイスにセキュアにアクセス

モノの一括登録

数回のクリックで、接続されているデバイスをまとめて素早く導入する

デバイス全体のメタデータ、証明書、ポリシーなどデバイス情報を登録

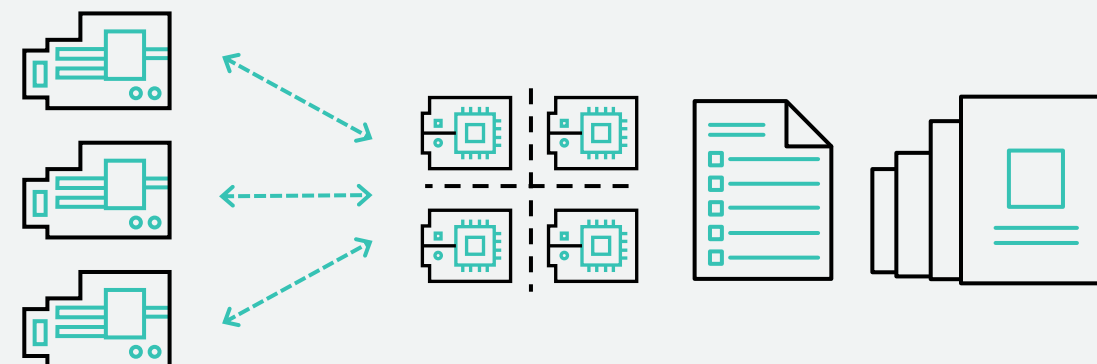
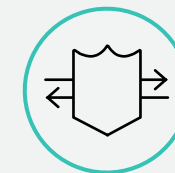
全デバイスを登録するには、コンソールでのアップロード機能かStartThingRegistrationTask APIを利用できる

デバイス登録の進捗はトラッキング可能で、完了したタスクのレポートはダウンロード可能

新しいデバイスの登録および既存デバイスの再登録が可能（例：証明書のローテーション）

ユースケースとしては製品製造時に利用するほか、既存システムからの移行で活用可能

Connectivity and control services



手動で Provisioningを行う場合

- モノの作成
 - `aws iot create-thing`
- 秘密鍵と証明書を発行
 - `aws iot create-keys-and-certificate`
- ポリシーの作成
 - `aws iot create-policy`
- ポリシーを証明書にアタッチ
 - `aws iot attach-policy`
- モノに証明書にアタッチ
 - `aws iot attach-thing-principal`

<https://docs.aws.amazon.com/cli/latest/reference/iot/>

AWS IoT Device Management で利用できる登録方法

- 一括登録
 - プロビジョニングテンプレートとパラメータファイルを利用して、ファイルに記載されているデバイスを[start-thing-registration-task](#)を使って一括で登録する
- フリートプロビジョニング
 - クレーム証明書を使用したデバイスのプロビジョニング
 - 信頼されるユーザによるプロビジョニング
- 事前登録されたCA証明書を使用したデバイスのプロビジョニング
 - Just in Time Provisioning(JITP)
 - Just in Time Registration(JITR)

参考 デバイスへの認証情報のプロビジョニング



<https://www.youtube.com/watch?v=gcJwNEQ2eLY>

https://pages.awscloud.com/rs/112-TZM-766/images/EV_ iot-deepdive-aws2_Sep-2020.pdf

テンプレートを用いて一括登録する Demo

1. テンプレートファイルの確認
2. 一括登録の実行
3. 実行結果の確認
4. 証明書の保存
5. 疎通確認

フリートインデックス

デバイス群の正常性と状態を可視化

デバイス属性の任意の組み合わせに基づいてデバイス群から目的のデバイスを見つけ出す

例：現在接続されているデバイスの中からファームウェアバージョン1.2かつ2013年以降に製造された全てのデバイスを検索

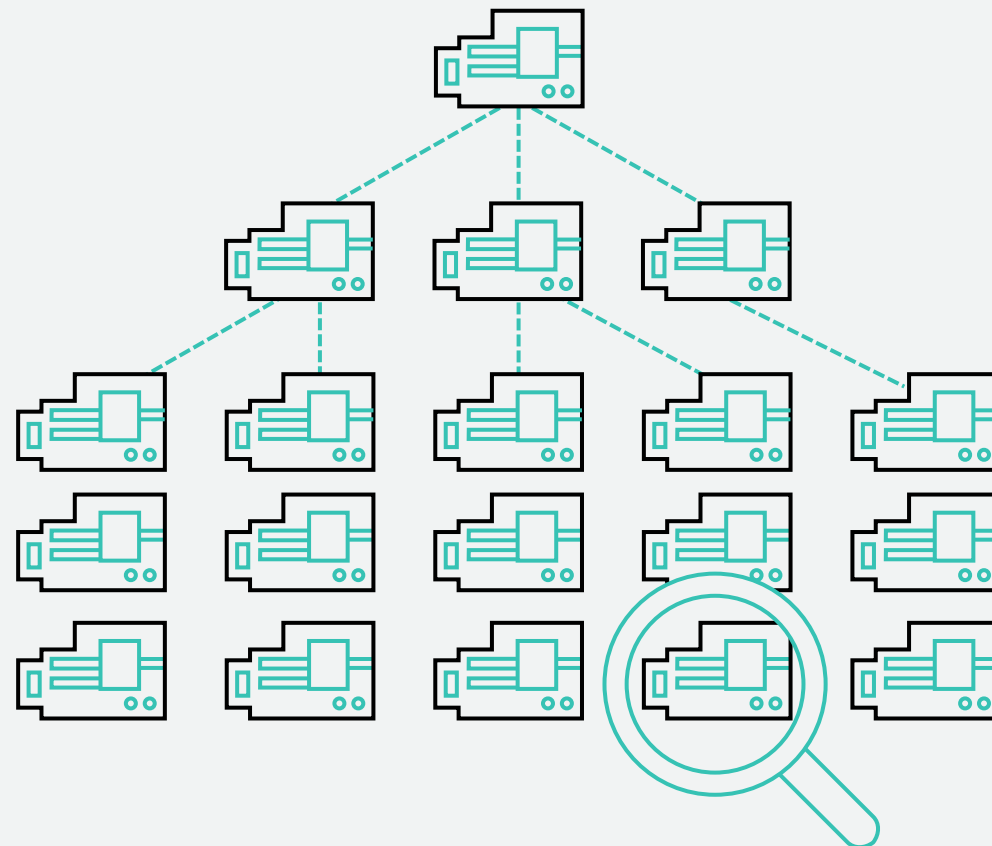
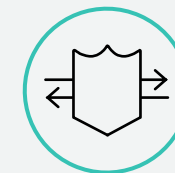
クエリに基づいてデバイスのグループを動的に更新し、デバイスの編成を自動化

例：東京にある全てのハードウェアバージョン1.1の電球をグループにする

Amazon CloudWatch を使用して、デバイス群の集計情報を定期的にモニタリング

コンソールからワンクリックの使いやすさ

Connectivity and control services



フリートインデックスの機能

- デバイス情報のインデックス化
- デバイス群の横断検索
- デバイス群の統計値の取得
- フリートメトリクスを利用した統計値のモニタリング

事前準備 デバイス情報のインデックス化

フリートのインデックス作成の管理 情報

クリック

モノのインデックス作成

モノの名前、説明、属性に基づいてモノを検索します。追加の設定を含めるには、モノのインデックス作成を有効にする必要があります。

有効化済み

設定

フリートのインデックス作成で、インデックスを作成するデータソースを選択します。

名前なし (クラシック) シャドウの追加

デバイスの名前なしシャドウファイルから reported、desired、hasDelta の各フィールドに基づいてモノを検索します。

名前付きシャドウの追加

デバイスの名前付きシャドウファイルから reported、desired、hasDelta の各フィールドに基づいてモノを検索します。

対象を選択

名前付きシャドウの選択 (4/4)

モノのインデックス作成にシャドウ名を追加します。名前付きシャドウを削除するには、リストからそれらを選択し、**【削除】**を選択します。

削除

モノでシャドウ名を参照

シャドウ名を追加

< 1 >

シャドウ名

location

building

更新

OR

```
aws iot update-indexing-configuration ¥
```

```
--thing-indexing-configuration thingIndexingMode=REGISTRY_AND_SHADOW,thingConnectivityIndexingMode=STATUS ¥
```

```
--thing-group-indexing-configuration thingGroupIndexingMode=ON
```



デバイス群を横断検索

GUI

AWS IoT > 管理 > モノ > モノの高度な検索

フリーインデックス作成で名前付きシャドウの選択が可能に
名前付きシャドウを選択して、フリーのインデックス作成設定に追加できます。 [詳細はこちら](#) フリーインデックス作成を管理

モノの高度な検索 情報

高度な検索では、Lucene の検索クエリを使用してモノを見つけます。フリーのインデックス作成を設定して、モノのプロパティ、Device Shadow データ、接続状態、Device Defender 違反のインデックスを作成できます。検索結果を、詳細統計、パーセントイルグループ、または濃度を表示するかどうかを選択します。

フリーインデックス作成を管理

クエリ 情報

検索するモノについて返します。クエリの入力を開始すると、予測されるプロパティがリストで表示されます。また、このリストに表示されないモノの属性を入力することもできます。

保存されたクエリを使用 クエリを保存

attributes.city:desiredValue

プロパティ よび thingName:* および thingTypeName:*

クエリをクリア キャンセル 検索

プロパティ	モノのタイプ	属性
connectivity.connected		
connectivity.disconnectReason		
connectivity.timestamp		
deviceDefender.*.inViolation		
deviceDefender.*.lastViolationTime		
deviceDefender.*.metricName		
deviceDefender.version		
deviceDefender.violationCount		
shadow.hasDelta		
thingGroupNames		
thingId		
thingName		
thingTypeName	xe4669	-
TestFleetPrefix_9541		

Luceneエンジンベースでの検索機能

shadow.reported.firmware:1.0.0

や

*thingName:test**

などのクエリをかけて最大12個のクエリ句でモノを検索することができる

検索条件で利用できるのは、接続状況、モノの名前、属性、モノのタイプ、グループ名、device shadow(*1)など

CLI

```
aws iot search-index --index-name "AWS_Things" --query-string "shadow.reported.firmware:1.1.0"
```

* 通常Shadowは暗号化されて保管されているがIndexでshadowを対象にすると暗号化されなくなります。

<https://docs.aws.amazon.com/iot/latest/developerguide/managing-fleet-index.html>

デバイス群を横断検索

```
$ aws iot search-index ¥
--index-name "AWS_Things" ¥
--query-string "thingName: 2022*"
{
  "things": [
    {
      "thingName": "20220202_SSMTest",
      "thingId": "31cfae7d-63f6-4bc5-8f3d-e5a61e7ea52b",
      "thingGroupNames": [
        "20220202_SSMTest_Group"
      ],
      "connectivity": {
        "connected": false
      }
    }
  ]
}
```

デバイス群の統計値の取得機能

- **GetStatistics**
 - クエリに一致するモノの総数、フィールドの値の平均、合計、最大、最小、二乗和、分散、標準偏差を取得
- **GetPercentiles**
 - クエリに一致するモノのフィールドの値の、任意のパーセンタイルにおける近似値を取得
- **GetCardinality**
 - クエリに一致するモノのフィールドの値、ユニークな数を取得
- **GetBucketsAggregation**
 - 同じ値の件数(group byのような集計)

使い方- Get Statistics

ThingName	Temp Sensor0	Temp Sensor1	Temp Sensor2	Temp Sensor3	Temp Sensor4	Temp Sensor5	Temp Sensor6	Temp Sensor7	Temp Sensor8	Temp Sensor9
temperature	70	71	72	73	74	75	47	97	98	99
rackId	Rack1	Rack1	Rack2	Rack2	Rack3	Rack4	Rack5	Rack6	Rack6	Rack6
stateNormal	true	true	true	true	true	true	false	false	false	false

```
$ aws iot get-statistics --aggregation-field "attributes.temperature" --query-string "thingName:TempSensor*"
```

```
{
  "statistics": {
    "count": 10,
    "average": 77.6,
    "sum": 776.0,
    "minimum": 47.0,
    "maximum": 99.0,
    "sumOfSquares": 62578.0,
    "variance": 236.04000000000013,
    "stdDeviation": 15.36359332968691
  }
}
```

temperature の統計値

使い方- Get Percentiles

ThingName	Temp Sensor0	Temp Sensor1	Temp Sensor2	Temp Sensor3	Temp Sensor4	Temp Sensor5	Temp Sensor6	Temp Sensor7	Temp Sensor8	Temp Sensor9
temperature	70	71	72	73	74	75	47	97	98	99
rackId	Rack1	Rack1	Rack2	Rack2	Rack3	Rack4	Rack5	Rack6	Rack6	Rack6
stateNormal	true	true	true	true	true	true	false	false	false	false

```
$ aws iot get-percentiles --aggregation-field "attributes.temperature"  
--query-string "thingName:TempSensor*" --percents 10 25 50 75 90
```

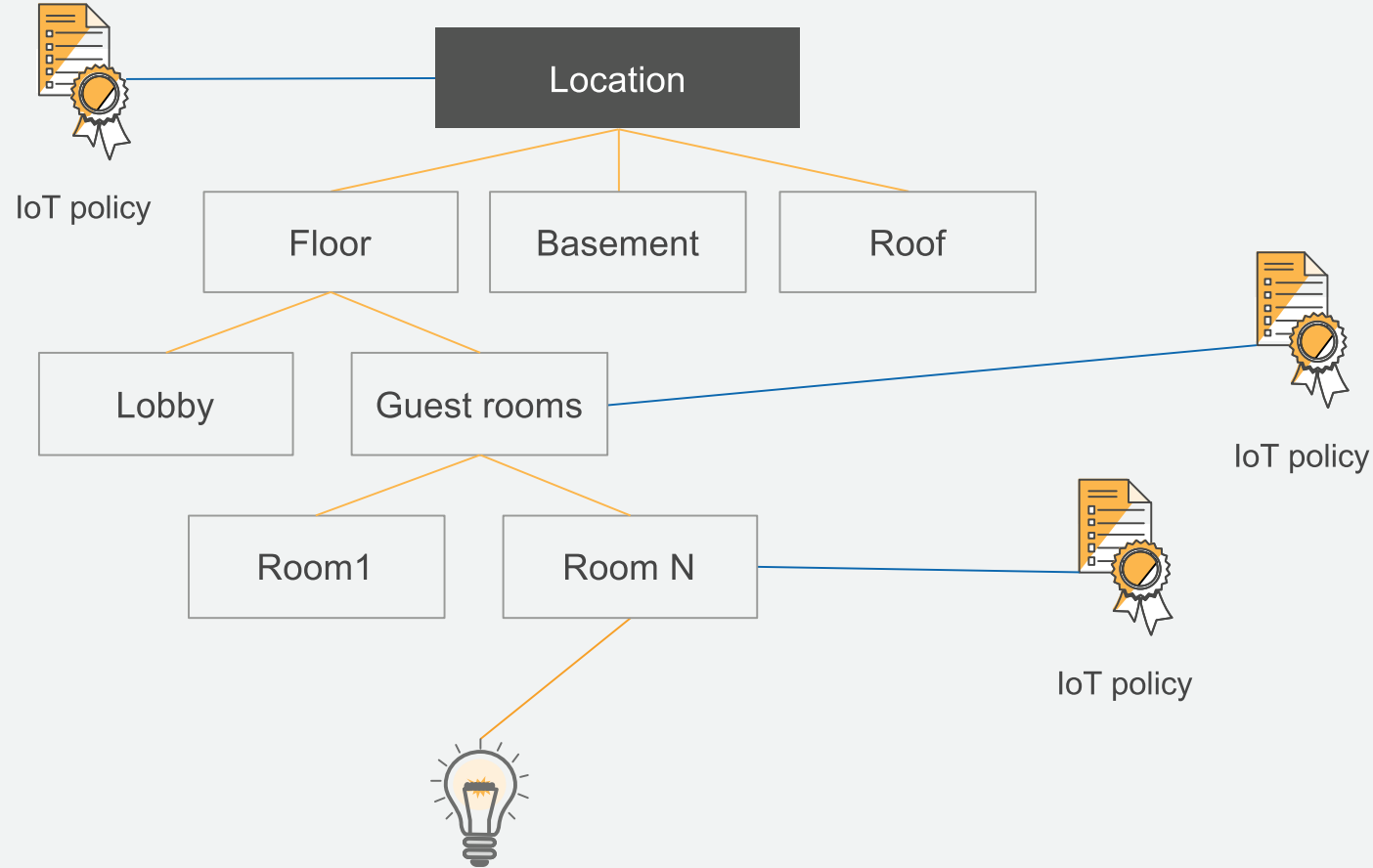
```
{  
  "percentiles": [  
    { "percent": 10.0, "value": 67.7 },  
    { "percent": 25.0, "value": 71.25 },  
    { "percent": 50.0, "value": 73.5 },  
    { "percent": 75.0, "value": 91.75 },  
    { "percent": 90.0, "value": 98.1 }  
  ]  
}
```

指定された Percentile での
temperature の値

モノのグループ

- グループ化することで複数のモノを一度に管理できる
- 静的(Static)グループと動的(Dynamic)グループがある
- グループごとにジョブを実行可能

モノの静的グループ



モノをグループとして、論理表現できる。グループ単位でのポリシー設定も可能。
グループは親子関係を持てるので、グループ間で階層設計ができる
例 ビル => フロア => 部屋

モノの静的グループ


AWS IoT > 管理 > モノのグループ > Location

Location 情報

編集

削除

モノのグループの詳細

名前 Location	親グループ -	作成日 February 24, 2023, 14:23:14 (UTC+0900)
ARN  arn:aws:iot:ap-northeast-1:327038145508:thinggroup/Location	説明 -	タイプ 静的

属性 | 子グループ | モノ | ポリシー | ジョブ | セキュリティプロファイル | タグ

子グループ (3) 情報

モノのグループは、1つの階層に整理できます。このリストの子グループは、このモノのグループを親グループとして持つグループです。



削除

子グループを作成

🔍 子のモノのグループをフィルタリング

< 1 > ⚙️

モノのグループ ▲

Basement

Floor

モノをグループとして、論理表現できる。グループ単位でのポリシー設定も可能。
グループは親子関係を持てるので、グループ間で階層設計ができる
例 ビル => フロア => 部屋

モノの動的グループ

- 条件を指定したグループを作成できる
- フリーインデックスが必要

```
aws iot create-dynamic-thing-group --thing-group-name  
"FirmwareV2" --query-string "attributes.fw_version=2.0"
```

- 登録した条件(`attributes.fw_version=2.0`)に一致すると、モノが自動的にグループに所属する
- これにより、`ファームウェアのバージョンが2.0`のデバイスだけに対してジョブを実行するといった事ができる
- 条件に合致しないデバイスはグループから抜ける
- `--query-string`で指定できる要素は以下を参照
 - https://docs.aws.amazon.com/ja_jp/iot/latest/developerguide/query-syntax.html

フリートメトリクスを利用した統計値のモニタリング

- フリートインデックスを有効化後、Amazon CloudWatch と連動してデバイスの統計情報を定期的に監視
- 集計フィールドには属性、モノの名前、接続状況、シャドウといった値を指定
- 集計タイプは、統計、濃度、パーセンタイルを選択
- データを送信する頻度は1分から最長24時間まで設定可能

フリートメトリクスを編集 情報

クエリとプロパティを指定して、CloudWatch に値を報告するためのフリートインデックスを作成し、そのインデックスを使用してデバイスの統計情報を作成します。

レポートのプロパティ
クエリが収集したモノからデータを集計する方法について説明します。

インデックス名
このレポートに集約するインデックスを選択します。
AWS_Things

集計フィールド
評価する属性またはデータを選択します。
connectivity.connected

集計タイプ
表示する統計的な要約を選択します。
統計

データを選択
統計のために集約するデータを選択
 数 最大 最小
 合計 平均

デバイスレポート

数	13	平均	0.154
合計	2	最小	-
最大	1	平方和	2
差異	0.131	標準偏差	0.361

フリートインデックス作成で名前付きシャドウの選択が可能
名前付きシャドウを選択して、フリートのインデックス作成の詳細はこちら [🔗](#)
フリートインデックス作成を管理

クエリ 情報
評価するモノについて記述します。クエリの入力を開始すると、予測される結果が表示されます。また、このリストに表示されないモノの属性を入力することもできます。
保存されたクエリを使用 **クエリを保存**

クエリをクリア

レポートをプレビュー

きめ細やかなデバイスの ロギングとモニタリング

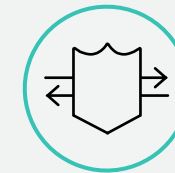
デバイスログを収集し問題を特定し
修復する

モノのグループ、Client Id, Source Ip, Principal
Id, Default ごとにロギングレベルを設定

トラブルシューティングのため、誤作動している
一部のデバイスの診断レベルを選択的に上げる
ことが可能

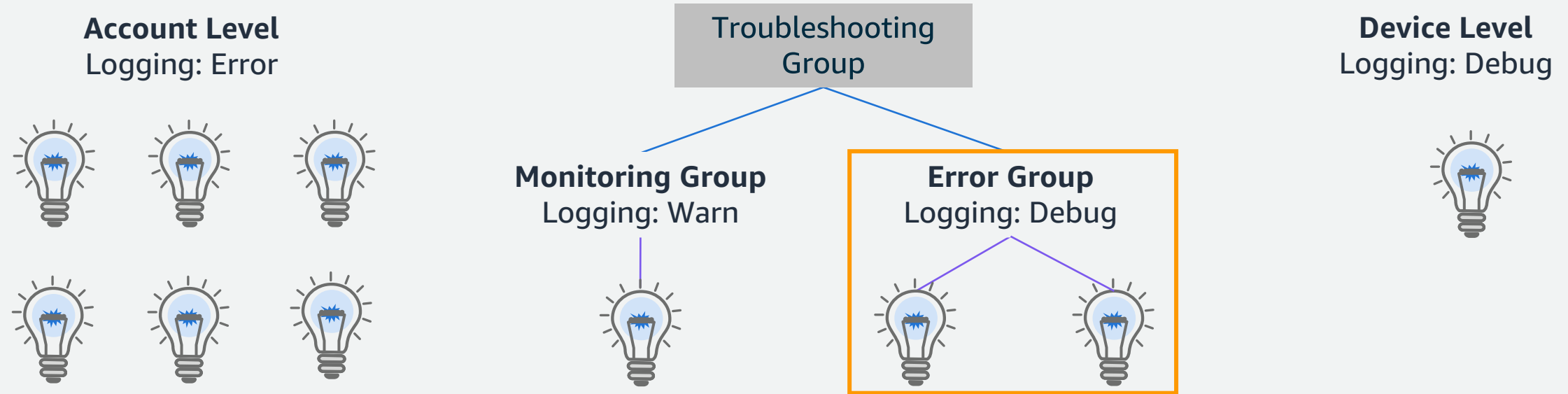
Amazon CloudWatch を使用してアラームを設定
& ログを検索

Connectivity and
control services



対象に応じたログレベルの変更

モノのグループ, Client Id, Source Ip, Principal Id の単位で指定可能



```
#Error Groupに対してログレベルをDEBUGに設定  
aws iot set-v2-logging-level --log-level DEBUG ¥  
--log-target "{¥"targetType¥": ¥"THING_GROUP¥", ¥"targetName¥": ¥"ERRORGROUP¥"}"
```

```
# ログのレベルを確認します  
aws iot list-v2-logging-levels
```

<https://docs.aws.amazon.com/cli/latest/reference/iot/set-v2-logging-level.html>

CloudWatch LogsでAWS IoTのログを参照

CloudWatch > ログのインサイト

ログのインサイト

ロググループを選択してクエリを実行するか、[サンプルクエリを選択](#)

ロググループを選択

AWSIoTLogsV2

```
1 fields @timestamp, @message, @logStream, @log , status
2 | filter @message like /ERROR/
3 | sort @timestamp desc
4 | limit 10000
```

クエリの実行 キャンセル 保存 履歴

クエリは最大 60 分間実行できます。

ログ 可視化

結果をエクスポート ダッシュボードに追加

1,009 の 1009 の一致したレコードの表示

80,821 レコード (36.0 MB) が 7.2s @ 11,206 records/s (5.0 MB/s) でスキャンされました

ヒストグラムを非表示

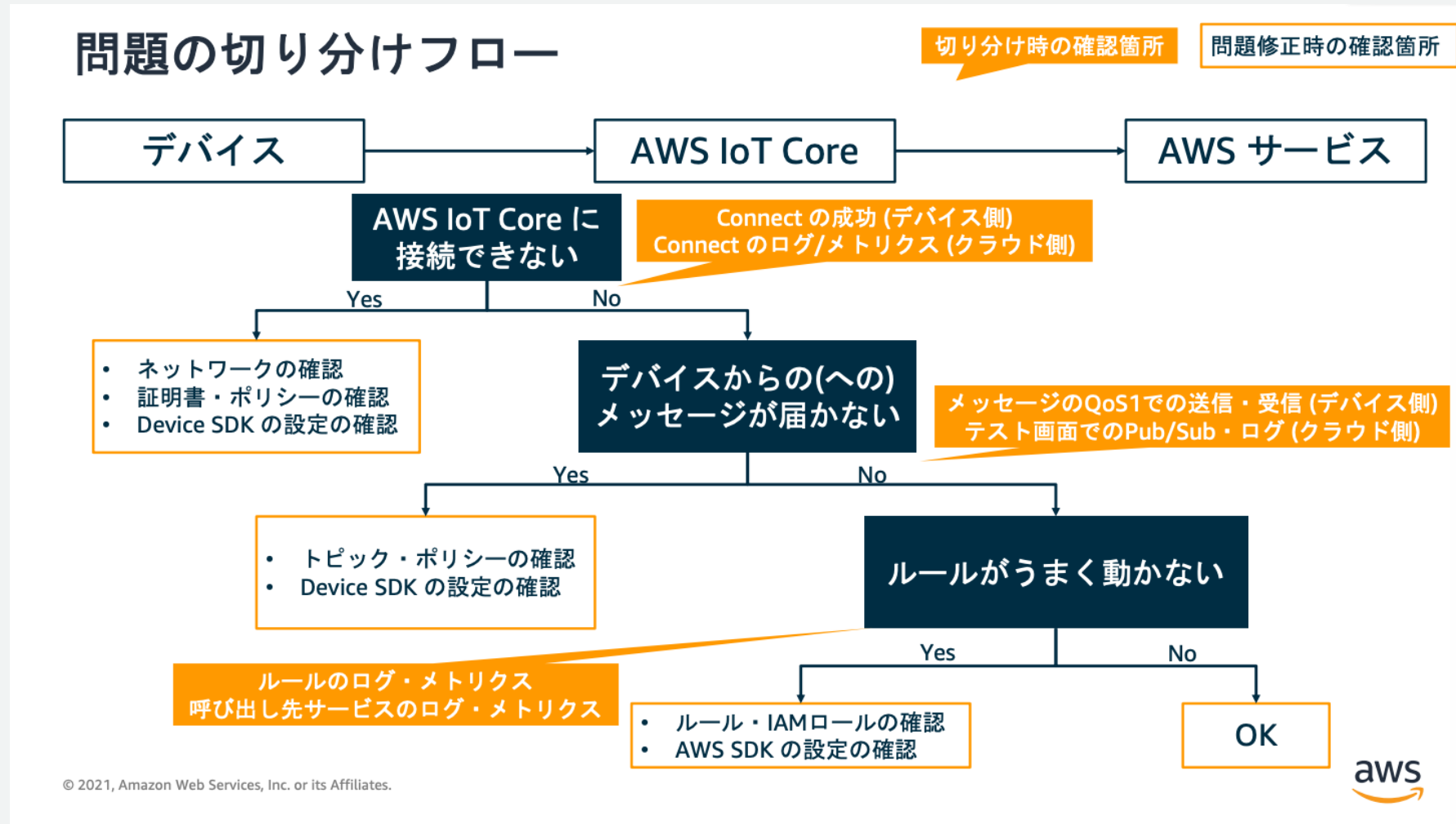
#	@timestamp	@message	@logStream	@log	status
▼ 1	2023-02-25T06:22:44...	{ "timestamp": "2023-02-24 21:22:44.050", "logLevel": "ERROR", "accountI...	0f05db04-692e-4af3-86b1-c4d8d5dece90_327038145508_...	327038145508:AWSIoTLog...	Failure

フィールド 値

- @ingestionTime 1677273774090
- @log 327038145508:AWSIoTLogsV2
- @logStream 0f05db04-692e-4af3-86b1-c4d8d5dece90_327038145508_0
- @message {"timestamp": "2023-02-24 21:22:44.050", "logLevel": "ERROR", "accountId": "327038145508", "status": "Failure", "eventType": "UpdateJobExecution", "protocol": "MQTT", "clientId": "bulky-AWS-IoT-Devicemanagement-Demo-Prod-1"}
- @timestamp 1677273764050
- accountId 327038145508
- clientId bulky-AWS-IoT-Devicemanagement-Demo-Prod-1
- clientToken fd306faa-3868-463c-9e08-786d92d14063

https://docs.aws.amazon.com/ja_jp/iot/latest/developerguide/cloud-watch-logs.html

参考 AWS IoT におけるデバック方法



https://pages.awscloud.com/rs/112-TZM-766/images/EV_aws-iot-deep-dive-3-310-topic1_Mar-2021.pdf
https://www.youtube.com/watch?v=8Q0cc_Mq64Q

ジョブ

グループに登録されているデバイスや個々のデバイスへジョブを実行し管理する

個々のデバイスのステータス更新を受信して、更新が実行されるたびに監視する

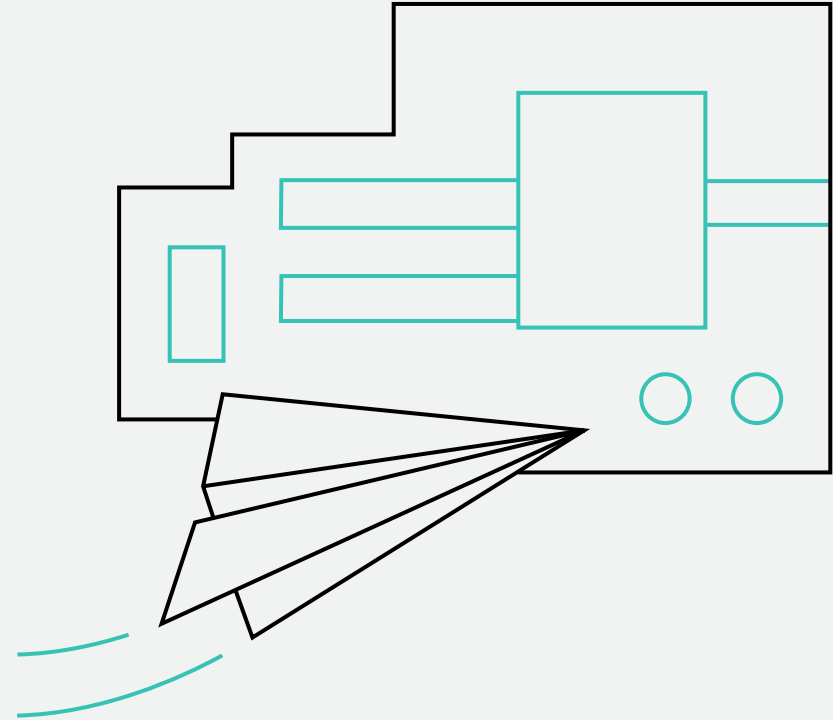
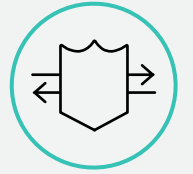
対象とするデバイスグループを一括更新または特定のデバイスを特定して更新できる

ジョブのロールアウトレートを設定したり、ジョブの成功失敗の閾値を設定することで、不具合の範囲を最小限に留めたりジョブを中止することが可能

デバイスに送信される前にジョブにデジタル署名することで、デバイスが危険にさらされるのを防ぐ

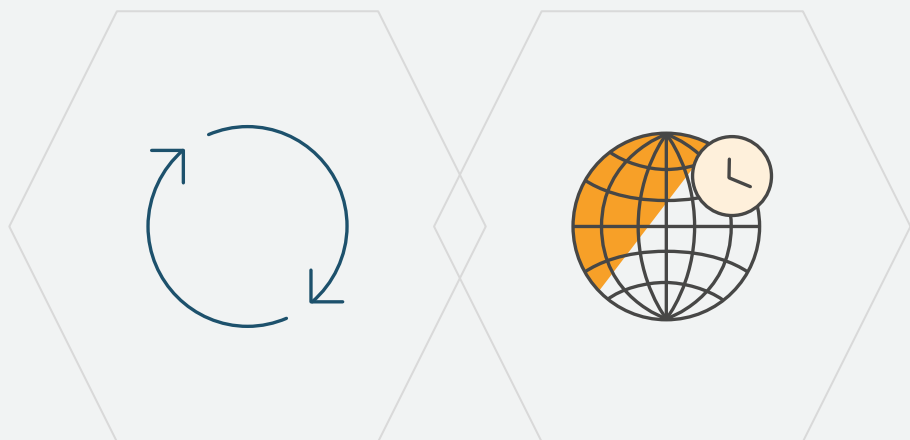
頻繁なリモートアクションをテンプレートとして保存し、ターゲットを選択することでそのアクションをオンデマンドで再利用可能

Connectivity and control services



ジョブの種類

ジョブの種類



Continuous

Snapshot

- **Snapshot:**
ジョブを作成したタイミングに存在するモノが対象。一度作るとジョブ通知の対象は変更されない。
- **Continuous:**
ジョブを作成した後も配信対象となるグループにモノが追加されると通知が行われる。

ジョブのコマンド例

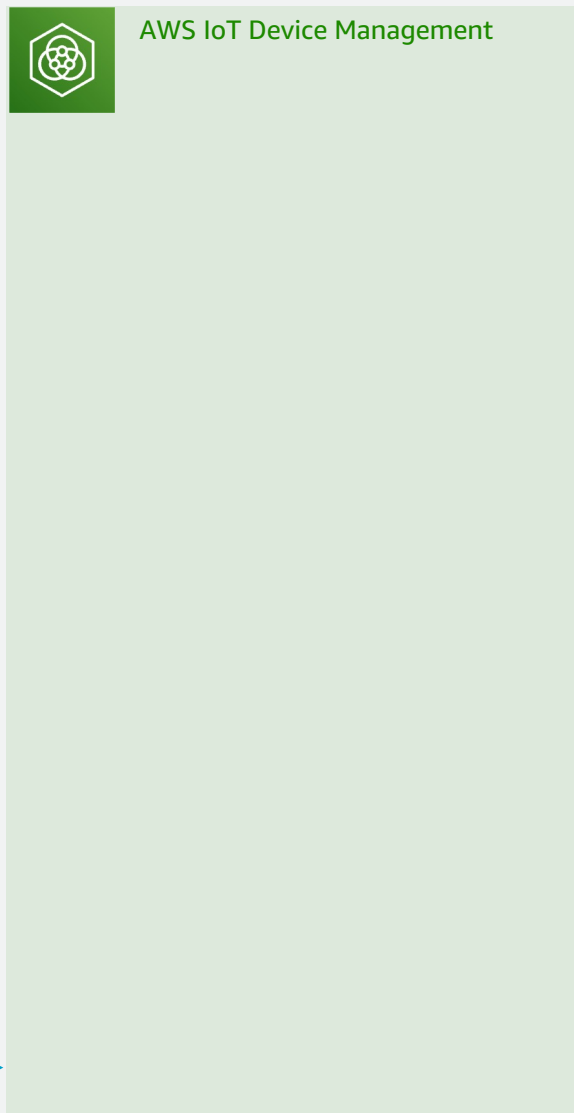
コマンド送信

```
{  
  "operation": "sys-info",  
  "sys-info": "uptime",  
  "topic": "sys/info"  
}
```

コマンドとデータの送信 (署名済み Amazon S3 URL に置き換え)

```
{  
  "operation": "install",  
  "data": "${aws:iot:s3-presigned-url:https://s3.amazonaws.com/job-test-bucket/datafile}"  
}
```

ジョブを利用する流れ



(1)※インデックスを使って対象を検索
(shadow.reported.firmware <= 1.0.0)

(2)※検索結果から更新用のグループ作成 ※オプション

(3) (2)で作成したグループ向けにジョブ生成

Details | **Job executions** | Job document | Job targets | Tags

Execution overview
When canceling and deleting execution refresh this page if the summary does not match.

Succeeded	Failed	Canceled	Rejected
1	0	0	0
Queued	In progress	Removed	Timed out
0	0	0	0

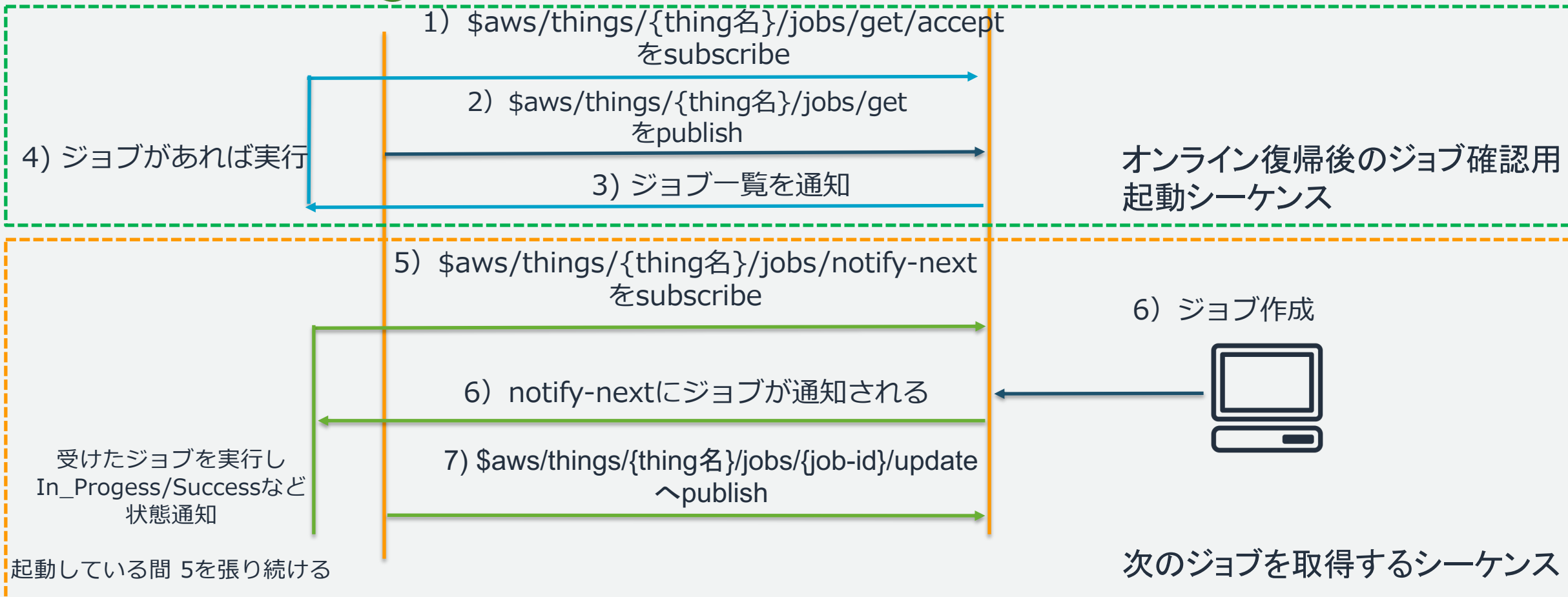
Job executions (1) Info Refresh Cancel execution Delete execution
Devices currently processing this job.

Find job executions All job executions (1) < 1 > ⚙

<input type="checkbox"/>	Thing name	Retry attempts	Retries remaining	Last updated ...	Queued at	Status
<input type="checkbox"/>	HomeGWV2	-	Retries not set	August 27, 2022, ...	August 27, 2022, ...	🟢 Succeeded

統計用API

ジョブを受けるデバイスのシーケンス



ジョブの作成

- ジョブの作成には「カスタムジョブ作成」「FreeRTOS OTA更新ジョブ作成」「Greengrass V1 Core 更新ジョブ作成」の3つがある。
- ジョブのターゲットはモノやモノのグループを指定可能
- ジョブの中身の定義はS3にアップロードしたジョブファイルまたはテンプレートから指定可能
- テンプレートは「AWS管理型テンプレート」「カスタムテンプレート」を利用可能

ジョブドキュメント - 新規 情報

ジョブドキュメントは、AWS IoT に接続されているデバイス上で送信および実行するリモートアクションを指定します。よく使用されるジョブは、デプロイを高速化するためジョブテンプレートに変換できます。AWS では、ジョブテンプレートの下にパブリックテンプレートが用意されており、実装を加速できます。

ファイルから
S3 にあるジョブファイルを指定します。このジョブは、後で再利用できるようにジョブテンプレートに変換できます。

テンプレートから
ジョブテンプレートを選択して、ジョブドキュメントとジョブ設定を再利用します。デプロイ前にファイルとその設定をカスタマイズできます。

ジョブテンプレート

テンプレートタイプ

AWS 管理型テンプレート

テンプレート

AWS-Start-Application

A managed job template for starting one or more system services.
環境: LINUX 最新バージョン: 1.0

AWS 管理型

表示 

AWS-Start-Application バージョン

1.0 - 現在
LINUX

services

A space separated list of system services to start.

`cron httpd bluetooth`

使用するシステムサービスのリストを入力します。このリストには文字と数字のみを含み、スペースで区切ります。

runAsUser - オプション

Execute handler as another user. If not specified, then handler is executed as the same user as device client.

`user1`

別のユーザーとしてジョブハンドラーを実行する場合は、256 文字以下の値を入力します。

pathToHandler - オプション

Path to handler on the device. If not specified, then device client will use the current working directory.

`/path/to/handler/script`

ジョブハンドラーへのパスを 4096 文字以下で入力します。

ジョブのロールアウト

- 通常だと1~1000個/分 の速度でモノに対してジョブが配信
- Exponential rate を指定することで
 - 一定の増加量でジョブをモノに配信可能
 - 一定の倍数で増加量を増やしジョブをモノに配信可能
 - 増加量を増やす条件を指定可能

追加設定 - オプション

▼ ロールアウト設定

レート

一定レート エクスポネンシャルレート

1分あたりの最大値 (1~1000)

1000

ジョブのロールアウト

1分あたりの基本レート (1~1000)

割増率 (1.2~5.0)

1分あたりの最大値 (1~1000)

レート増加の基準

成功したデバイスの数 ▼

ロールアウトレートのプレビュー

最初の 個のバッチのロールアウトレート。

カテゴリ	バッチ 1	バッチ 1	バッチ 1	バッチ 1
ロールアウトレート (デバイス/分)	10	20	40	80
成功したデバイスの数	5	10	15	20

開始時のレート

増加レート

1分あたりの最大数

レート増加させるために、ジョブが成功していなければならない台数

この設定のプレビュー

ジョブの中止設定

- 中止条件を設定することでカナリアデプロイのようなことが可能

•失敗のタイプ(Failed, Rejected, Time Out, All)

•ジョブを受け取ったモノから報告されたステータス

•Time Outは1分～7日の期間を指定でき、in progress になってからその期間を経過するとTime Outとなる

•失敗のタイプに該当するモノの割合の閾値

•中止するまでにジョブを実行するモノの最低数

▼ ジョブ実行タイムアウトの設定

ジョブの実行時間。

タイムアウト

タイムアウトなし (デフォルト)
ジョブはキャンセルまたは削除されるまで実行します。

タイムアウトを有効化
タイムアウトを1分～7日の間で指定できます。

日 時間 分

▼ ジョブ実行の再試行の設定

1つのデバイスでジョブの実行が失敗した場合、デバイスごとに最大10回の再試行を、組み合わせたすべての失敗タイプに対して実行できます。使用する再試行回数は、ジョブ作成時にのみ指定できます。ジョブが作成されると、再試行回数は編集できず、設定を削除することしかできません。

失敗のタイプ 再試行回数

9すべての失敗タイプの残りを再試行します。

タイムアウトの設定を1つ追加できます。

▼ 中止の設定

ジョブの失敗数がしきい値の割合に達し、ジョブがデバイスの最小数にデプロイされた場合、ジョブはキャンセルされます。

失敗のタイプ しきい値の割合 デプロイされた最小数のデバイス

0～100%

障害タイプごとに1つの設定を追加できます。

ジョブのスケジュール設定

- ジョブの開始と終了の日時を設定可能に
- ジョブ終了時の動作として「ジョブのロールアウトを停止」「キャンセル」「強制キャンセル」から選択
- ユーザの少ない夜間などの時間やサービスレベルに準じた停止時間に乗っ取ったジョブの実行時間を設定可能に

▼ スケジュール設定 - 新規

将来の日時でジョブをスケジュールします。表示される時間は、現在のシステムのタイムゾーンに基づいています。ただし、これらのタイムゾーンはシステムで UTC に変換されます。 [情報](#)

ジョブの開始 - オプション

このジョブを開始させる日時を指定します。[Job start] フィールドが空の場合、ジョブは作成直後に開始されます。startTime は、最初のジョブ作成日時から 1 年以内である必要があります。



24 時間形式を使用してください。

ジョブの終了 - オプション

このジョブを終了させる日時を指定します。endTime は、最初のジョブ作成日時から 2 年以下である必要があります。この値は、スケジュールされたすべてのジョブに対するオプションです。



24 時間形式を使用してください。

ジョブ終了時の動作 - オプション

ジョブの終了時間に達したときに、すべての未完了のジョブ実行の終了動作を選択します。デフォルトでは、ロールアウトは停止されます。

ジョブのロールアウトを停止 ▲

ジョブのロールアウトを停止

新しいジョブ実行は処理されませんが、以前にキューに入れられたキューと進行中の実行は 続行できます。

キャンセル

以前にキューに入れられた実行はキャンセルされますが、進行中の実行は続行できます。

強制キャンセル

以前にキューに入れられた実行はキャンセルされ、進行中の実行もキャンセルされます。

フリートハブ

AWS IoT に接続されたデバイス群を
管理するためのフルマネージド型
ウェブアプリケーション

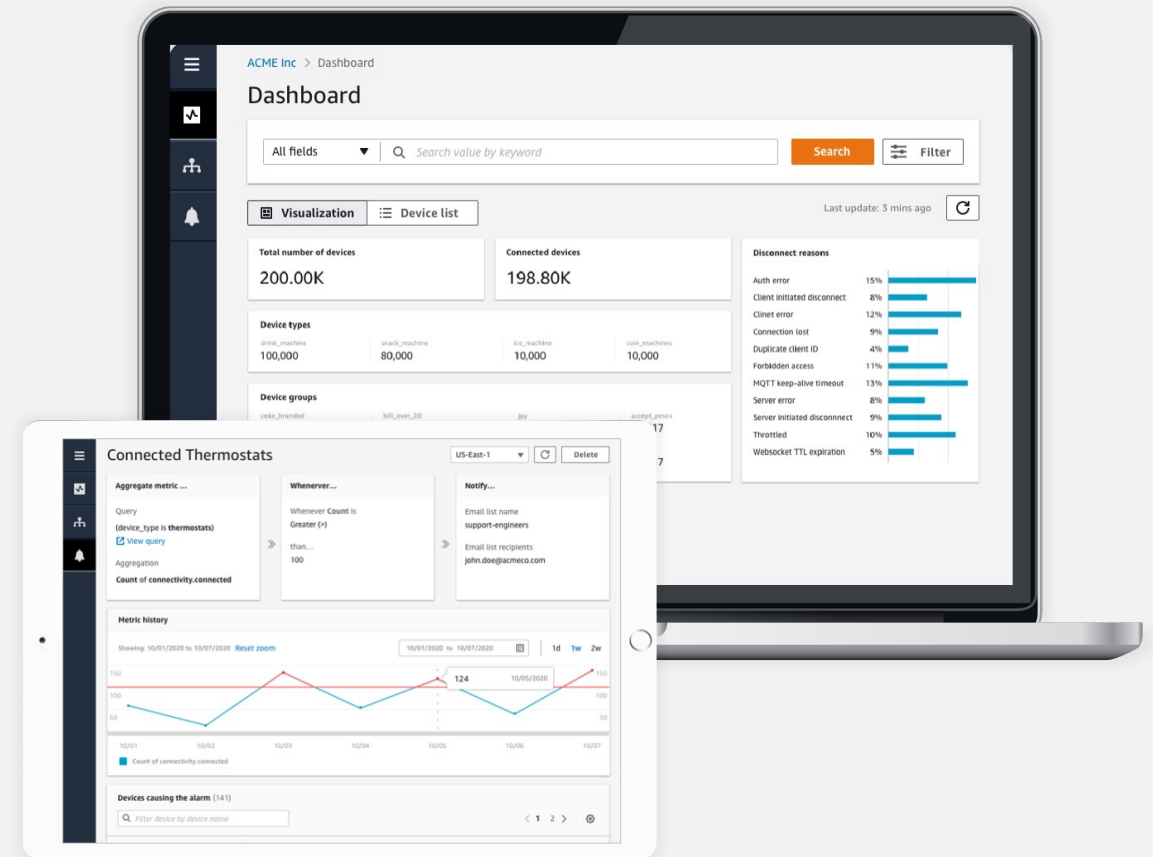
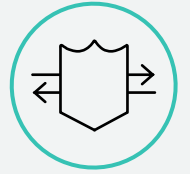
シンプルなインターフェースでデバイスを一覧、
フィルタ、アラームを設定し、アクションを実施

接続問題や異常な挙動、OTAの失敗などのデバイスの
状態をオペレータがカスタマイズして監視可能

フルマネージドなアプリケーションなのでビルドや
メンテナンスが不要

AWS IoT のコンソールページから数分で起動可能

Connectivity and
control services



フリートハブの Demo

1. デバイスの状態を一覧で確認する
2. 動的グループに所属するデバイスを確認する
3. 特定のグループのデバイスのみジョブを実行する
4. 接続デバイスが減った時に発報するアラートを作成する

セキュアトンネリング

アクセス制限のあるファイアウォール越しのデバイスに安全にアクセスする

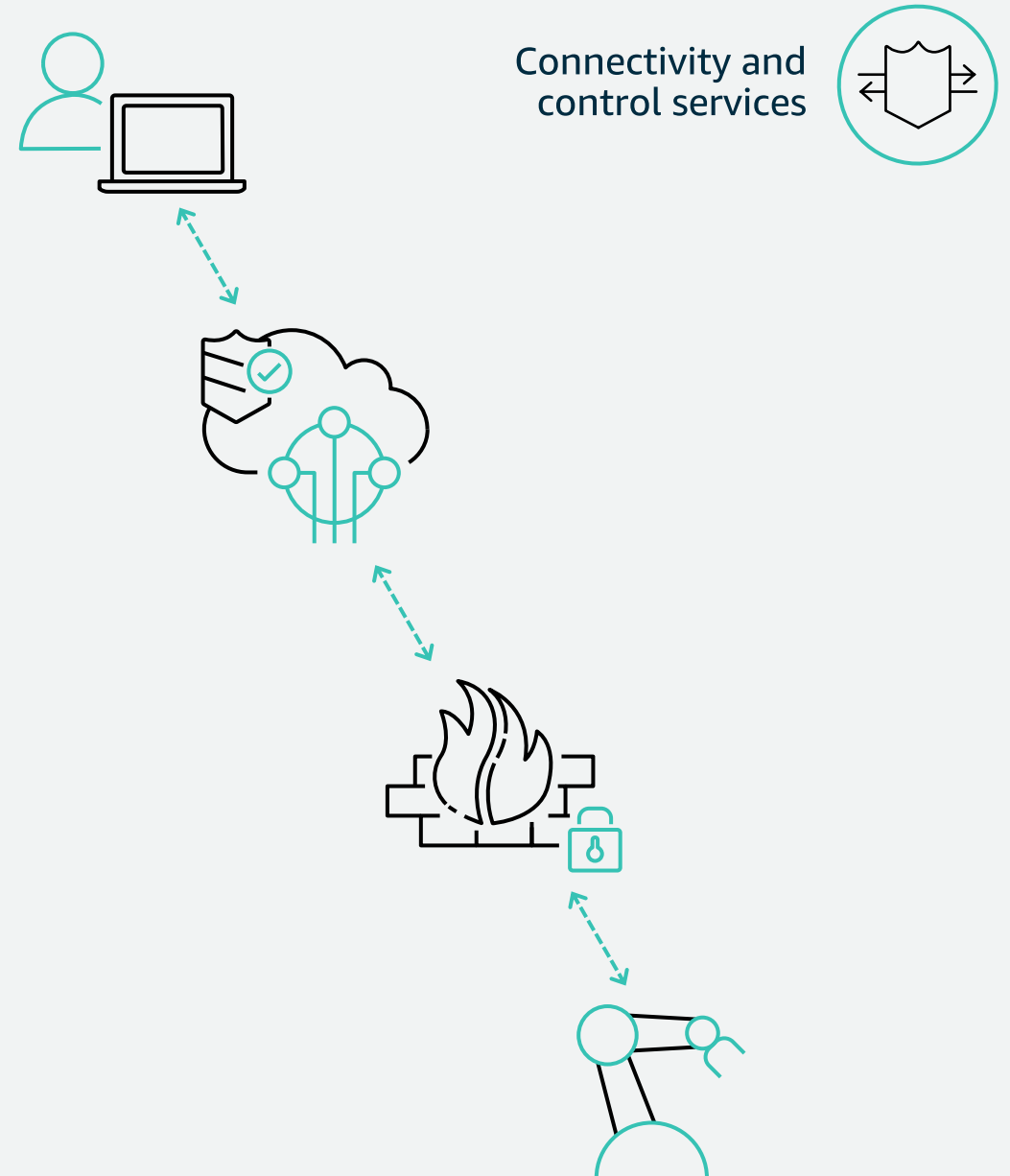
API またはコンソールを使用してトンネルを安全に開/閉できる

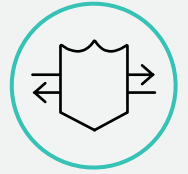
リモートシェルまたはリモートデスクトップ操作を使用して個々のデバイスにアクセスする

IAM 権限と最大 12 時間の設定可能なタイムアウトを使用して、各トンネルセッションのデバイス認証を管理する

MQTT メッセージを通じて各デバイスのアクセストークンを自動的に受け取る

ワンタイムトークンとトークンローテーション機能によって安全に再接続可能





セキュアトンネリング

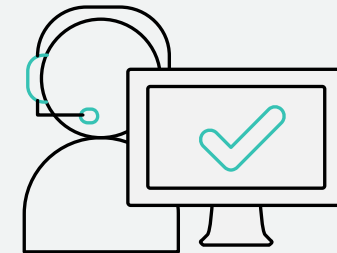
数回クリックするだけで個々のデバイスに安全に接続でき、問題を診断して解決するためのアクションを実行できます



隔離されたネットワーク上または
ファイアウォールの背後にある
デバイスへのリモートアクセスを実現



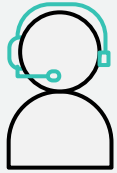
社内のセキュリティポリシーに
準拠した信頼できる接続を確立



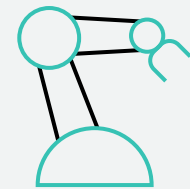
デバイスの問題をより迅速かつ費用対効果の
高い方法でトラブルシューティングし
解決することで、エンドユーザの体験を維持



セキュアトンネリング



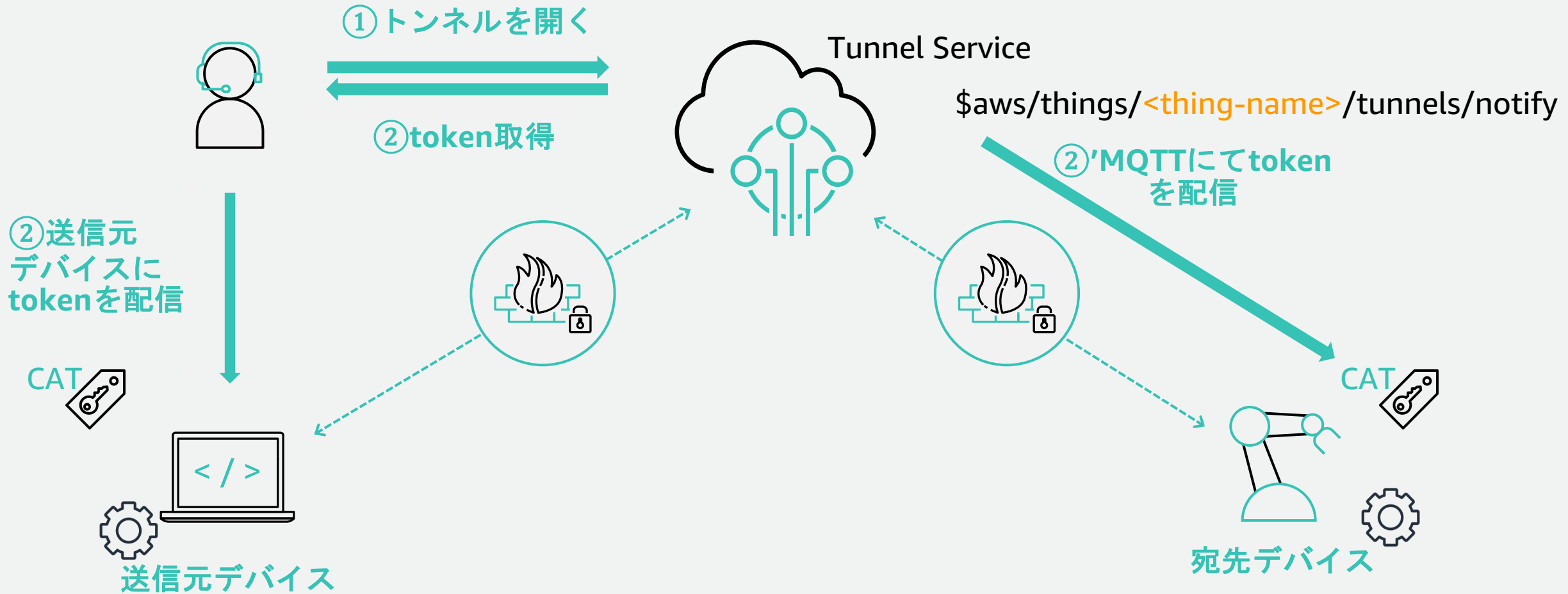
① ローカルプロキシ
をセットアップ



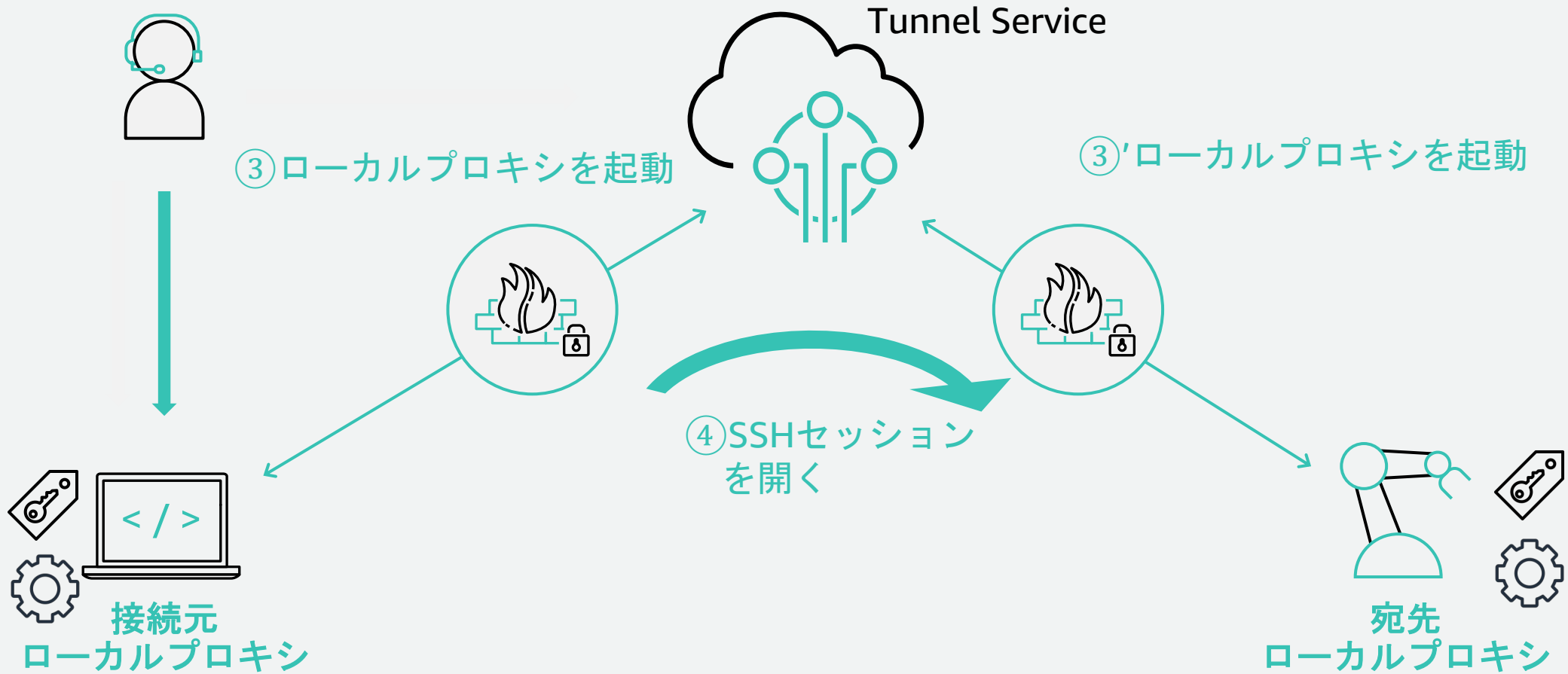
① ローカルプロキシ
をセットアップ



セキュアトンネリング

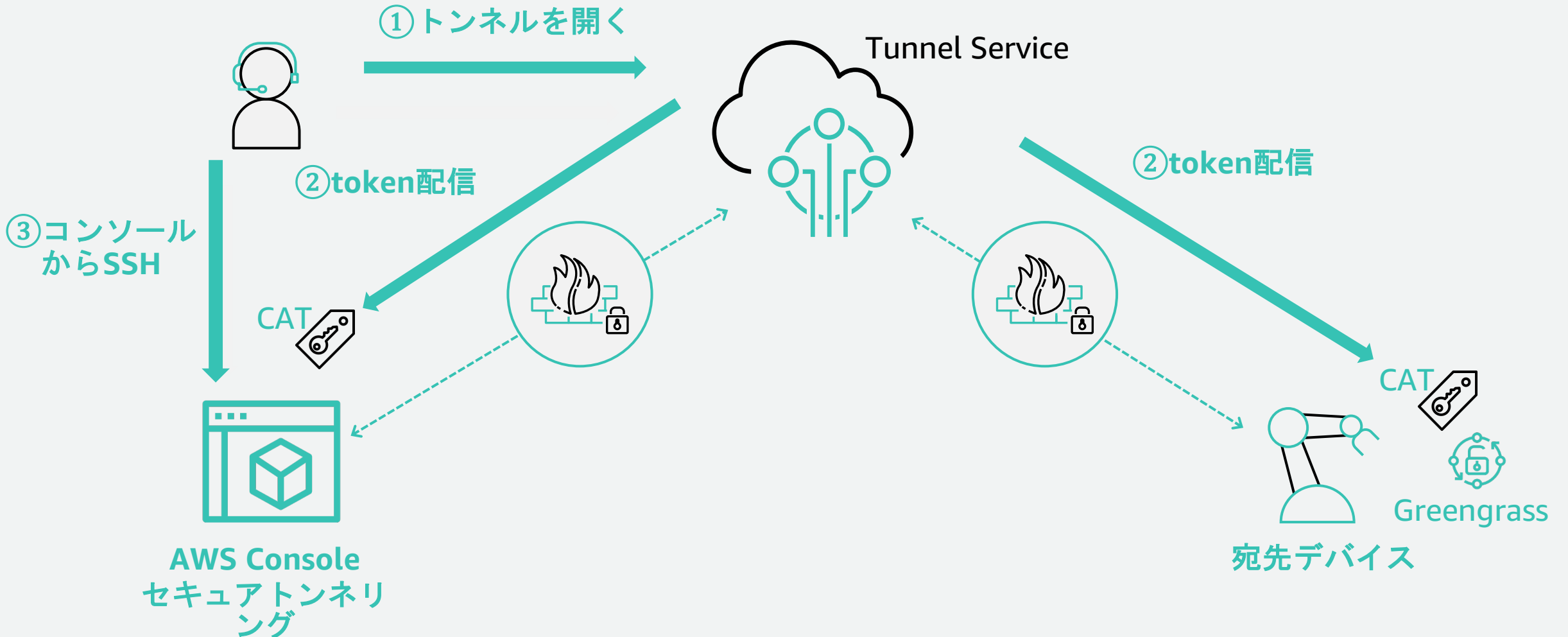


セキュアトンネリング





セキュアトンネリング ブラウザベースのSSH接続



セキュアトンネリングの Demo

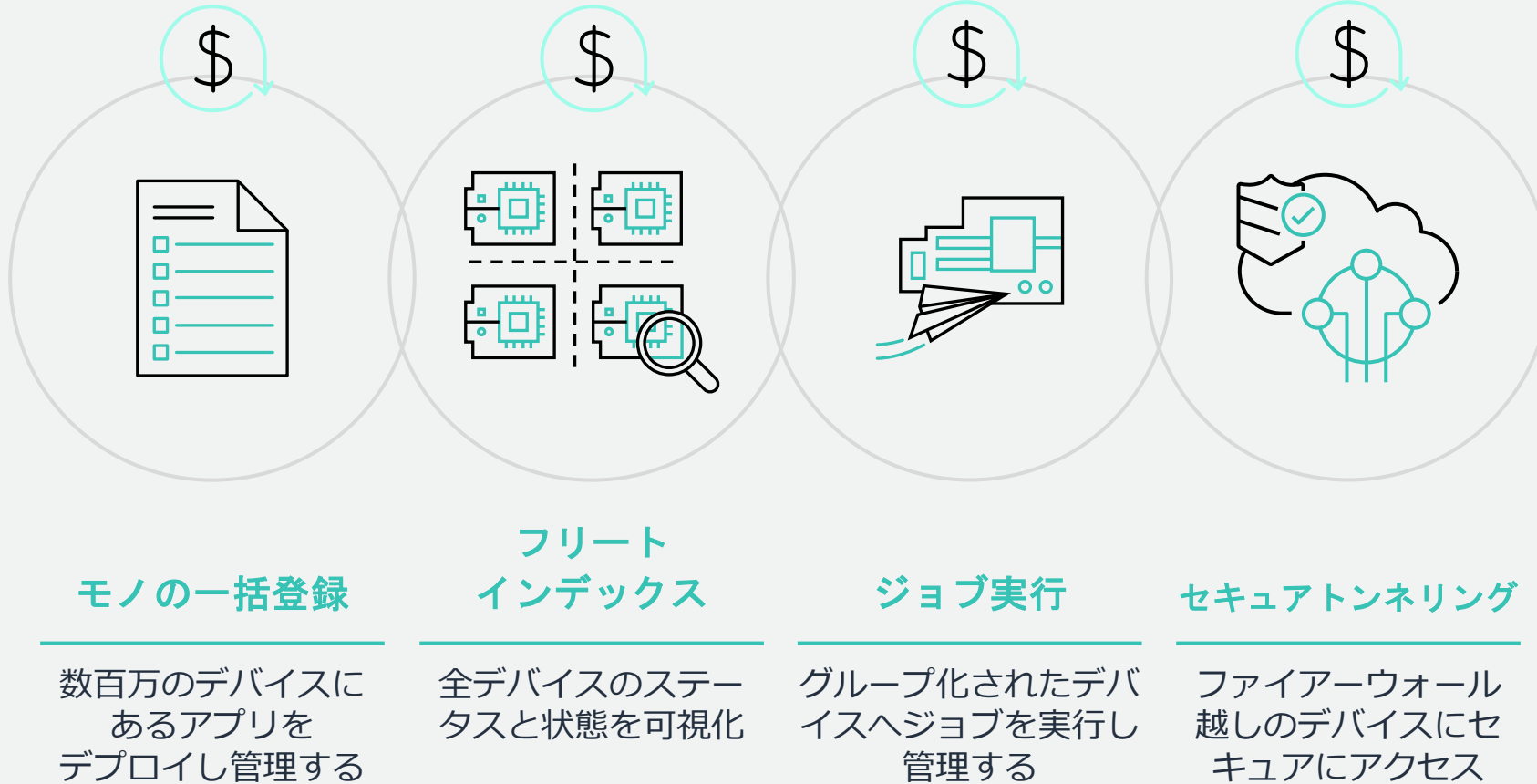
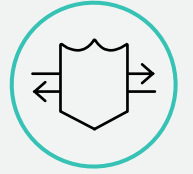
1. コンソールからトンネルを開く
2. クライアントアクセストークンのペアを確認
3. SSH セッションを開く
4. トンネルを閉じる

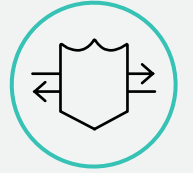
料金体系



AWS IoT Device Management の課金対象

Connectivity and control services





一括登録の価格

一括登録では、接続されているデバイスをまとめて登録できます。一括登録の費用は、登録するデバイスの数によって異なります。



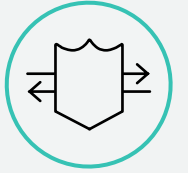
\$

- 登録されたモノ 1,000個ごとに料金が発生します。

例：10,000台を一括登録した場合、 $10,000/1,000 * \$0.12 = \1.20

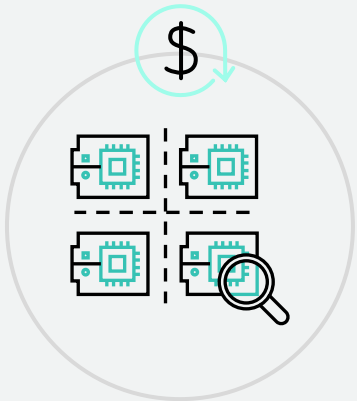
- また、AWS IoT Device Management にすでに登録されているデバイスに新しいデバイス属性を追加したり、デバイス証明書をローテーションしたりするなど、登録ワークフローを非登録タスクに使用する場合にも課金されます。5,000 台の登録済みデバイスのデバイス証明書をローテーションすると、登録された 5,000 個のデバイスに対して課金されます。

※この例は2023/02月時点の東京リージョンの計算です。最新の情報は [AWS IoT Device Management Pricing](#) を参照してください。



フリートインデックスの価格

フリートのインデックス作成と検索は、インデックスの更新回数と実行された検索クエリの数に基づいて計算されます。

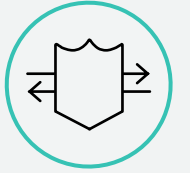


- インデックスの更新は1kb単位で料金が発生します
- インデックスの更新は、レジストリ、デバイスシャドウ、デバイスのライフサイクルイベント更新の際に行われます

例：10,000 個のセンサーそれぞれに、1 か月 1.5 KB のサイズのインデックス更新が 300 回実行された場合のコストは以下の通りです。

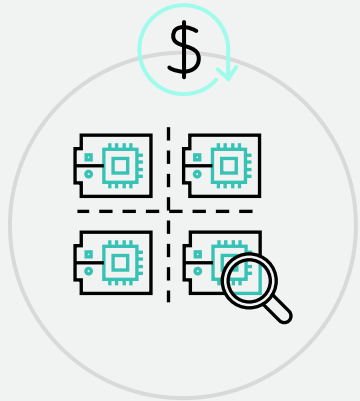
- インデックス更新サイズが 1 KB を超過しているため、超過した KB の分で 2 回のインデックス更新 (2 KB) として計算されます。
- インデックス更新料金の合計 = デバイス 10,000 台 * インデックス更新 300 回 * インデックス更新 2 回 * 2.70 USD/インデックス更新 1,000,000 回 = 16.20 USD/月

※この例は2023/02月時点の東京リージョンの計算です。最新の情報は [AWS IoT Device Management Pricing](#) を参照してください。



フリートインデックスの検索の価格

フリートのインデックス作成と検索は、インデックスの更新回数と実行された検索クエリの数に基づいて計算されます。

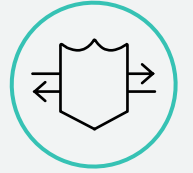


- 検索は10,000回あたりに料金が発生します

例：1か月あたり500,000回の検索を実行した場合、コストは以下の通りです。

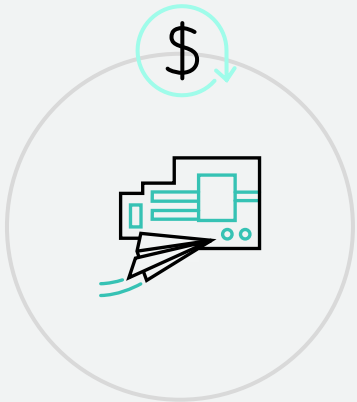
- フリートのインデックス検索料金の合計 = (検索 500,000 回 / 10,000 回) * 0.06
= 3.00 USD/月

※この例は2023/02月時点の東京リージョンの計算です。最新の情報は
[AWS IoT Device Management Pricing](#) を参照してください。



ジョブの価格

デバイスジョブは、デバイスに送信されたリモートアクションの数に基づいて計算されます。

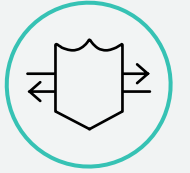


- リモートアクション毎に料金が発生します

例：ファームウェアの更新を 10,000 台のデバイスに実行した場合、コストは以下の通りです。

- デバイスジョブ料金の合計 = リモートアクション 10,000 回 * リモートアクション 1 回あたり 0.0036 USD = 36 USD
- 250,000アクションを超える場合、0.0018/アクション。
- 年間1,200万回以上のリモートアクションを行う場合の単価はお問い合わせください。

※この例は2023/02月時点の東京リージョンの計算です。最新の情報は [AWS IoT Device Management Pricing](#) を参照してください。



セキュアトンネリングの価格

セキュアトンネリングを使用すると、ファイアウォールの設定を変更することなく、制限付きファイアウォールの背後に展開できる個々のデバイスへの安全なトンネルを作成できます。

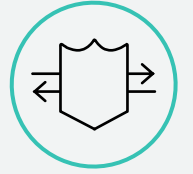
- 開いたトンネルごとに料金が発生します。

例：ローカルプロキシとトンネリングエージェントに 10,000 台のデバイスゲートウェイをすでに設置しています。今月、10 のゲートウェイに問題が発生し、自身のチームが 30 のトンネルを作成してこれらのデバイス上の問題をトラブルシューティングします。この場合、料金は以下のように計算されます。

- セキュアトンネリングの料金の合計 = 作成された 30 のトンネル * 作成されたトンネル 1 個につき \$1.20 = \$36.00
- 閉じられた、あるいは削除されたトンネルは、再び開くことはできなくなります。
- 接続上の問題があるときは、開いたトンネルを一時的に切り離すことができます。
- すでに開いているトンネルに再接続する際に課金は発生しません。

※この例は2023/02月時点の東京リージョンの計算です。最新の情報は [AWS IoT Device Management Pricing](#) を参照してください。





フリートハブの価格

Fleet Hub は、一元化されたインターフェイスを通じて既存の AWS IoT Device Management を拡張し、コーディングなしでデバイス群を管理できるようにします



- フリートハブを利用するための追加料金は発生しません
- フリートハブから利用されるサービスの利用料金がかかります

※この例は2023/02月時点の東京リージョンの計算です。最新の情報は [AWS IoT Device Management Pricing](#) を参照してください。

まとめ

- AWS IoT Device Management を利用することで、**大量のデバイスを効率よく登録、運用、監視、管理**することが出来る
- 配布されたデバイスを**継続的にアップデート**することが簡単に実現できる
- 遠方にあるデバイスの**トラブルシューティング**が可能になる

参考資料

- [AWS IoT Device Management Workshop](#)
- [AWS IoTにおけるデバイスへの認証情報のプロビジョニング](#)
- [AWS IoTにおけるデバッグ方法](#)
- [プロビジョニングテンプレート](#)
- [サンプルスクリプト](#)

本資料に関するお問い合わせ・ご感想

技術的な内容に関しましては、有料のAWSサポート窓口へお問い合わせください

<https://aws.amazon.com/jp/premiumsupport/>

料金面でのお問い合わせに関しましては、カスタマーサポート窓口へお問い合わせください（マネジメントコンソールへのログインが必要です）

<https://console.aws.amazon.com/support/home#/case/create?issueType=customer-service>

具体的な案件に対する構成相談は、後述する個別相談会をご活用ください



ご感想はTwitterへ！ハッシュタグは以下をご利用ください
#awsblackbelt

その他コンテンツのご紹介

ウェビナーなど、AWSのイベントスケジュールをご参照いただけます

<https://aws.amazon.com/jp/events/>

ハンズオンコンテンツ

<https://aws.amazon.com/jp/aws-jp-introduction/aws-jp-webinar-hands-on/>

AWS 個別相談会

AWSのソリューションアーキテクトと直接会話いただけます

<https://pages.awscloud.com/JAPAN-event-SP-Weekly-Sales-Consulting-Seminar-2021-reg-event.html>



Thank you!