

このコンテンツは公開から3年以上経過しており内容が古い可能性があります
最新情報については[サービス別資料](#)もしくはサービスのドキュメントをご確認ください

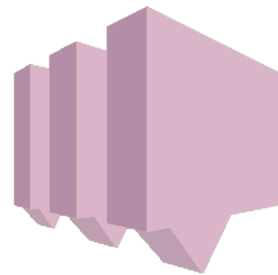
Amazon SNS Mobile Push

AWS Black Belt Tech Webinar 2015 (旧マイスターシリーズ)

アマゾン データサービス ジャパン 株式会社

ソリューション アーキテクト 清水 崇之

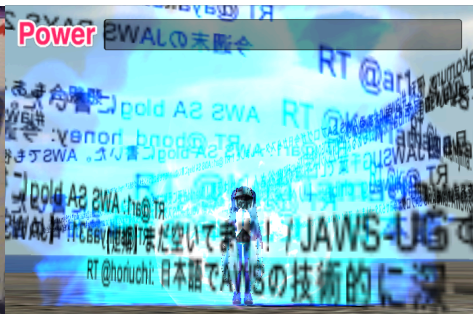
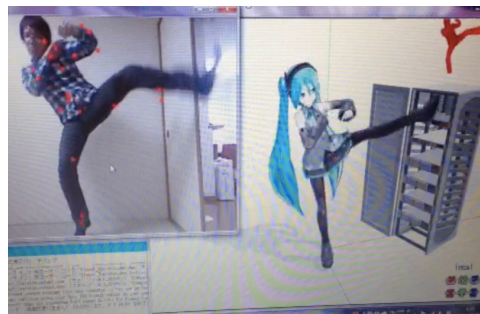
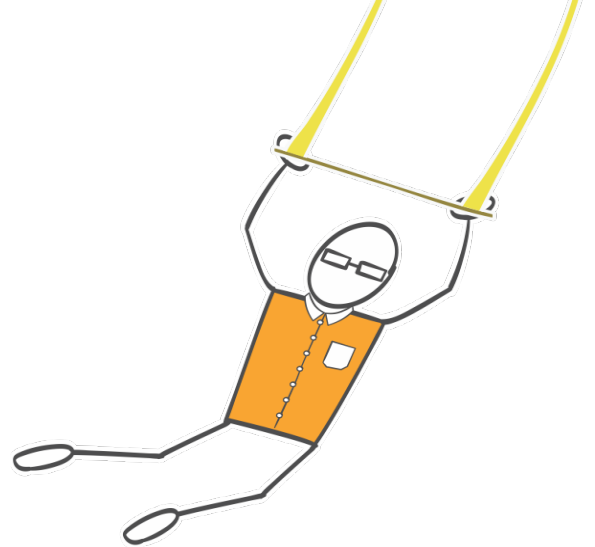
2015.03.11



自己紹介

• 清水 崇之

- ソリューション アーキテクト (WEST 担当)
- 大阪のお客様にもプライム対応で参上します
- Web サービス全般、モバイル、ゲーム など
- AWS 芸人 (詳しくは slideshare)



Agenda

- AWS モバイルサービスのご紹介
- Amazon SNS Mobile Push
 - モバイルプッシュ通知とは
 - 機能概要
 - SDK, IDE
 - 料金・無料使用枠
- Amazon SNS Mobile Push の使い方
 - マネージメントコンソールから
 - プログラムから
- Q&A

Agenda

Agenda

- **AWS モバイルサービスのご紹介**
- Amazon SNS Mobile Push
 - モバイルプッシュ通知とは
 - 機能概要
 - SDK, IDE
 - 料金・無料使用枠
- Amazon SNS Mobile Push の使い方
 - マネージメントコンソールから
 - プログラムから
- Q&A

モバイルアプリ開発の課題

ユーザ認証

ユーザの管理や
IDプロバイダとの連携

アクセスの認可

クラウドリソースへの
セキュアなアクセス

データの同期

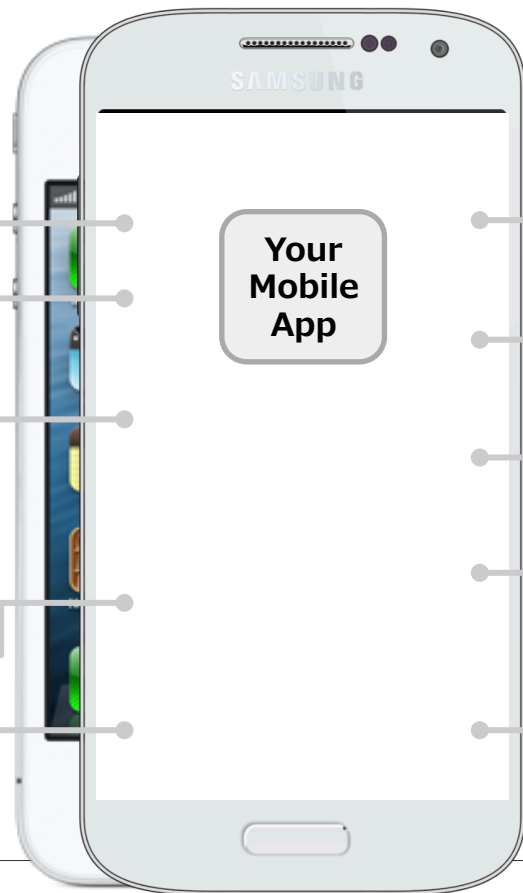
ユーザ設定等の複数
デバイス間での同期

ユーザの行動分析

アクティブユーザや
エンゲージメントの追跡

保持率の追跡

ファンネルやキャンペーン
効果の管理



メディアの管理

ユーザが投稿した写真やその他
メディアの保存と共有

メディアの配信

モバイルデバイスの自動識別と
素早く、グローバルなコンテンツの配信

プッシュ通知の送信

ユーザをアクティブに保つための信頼性の高い
メッセージ送信

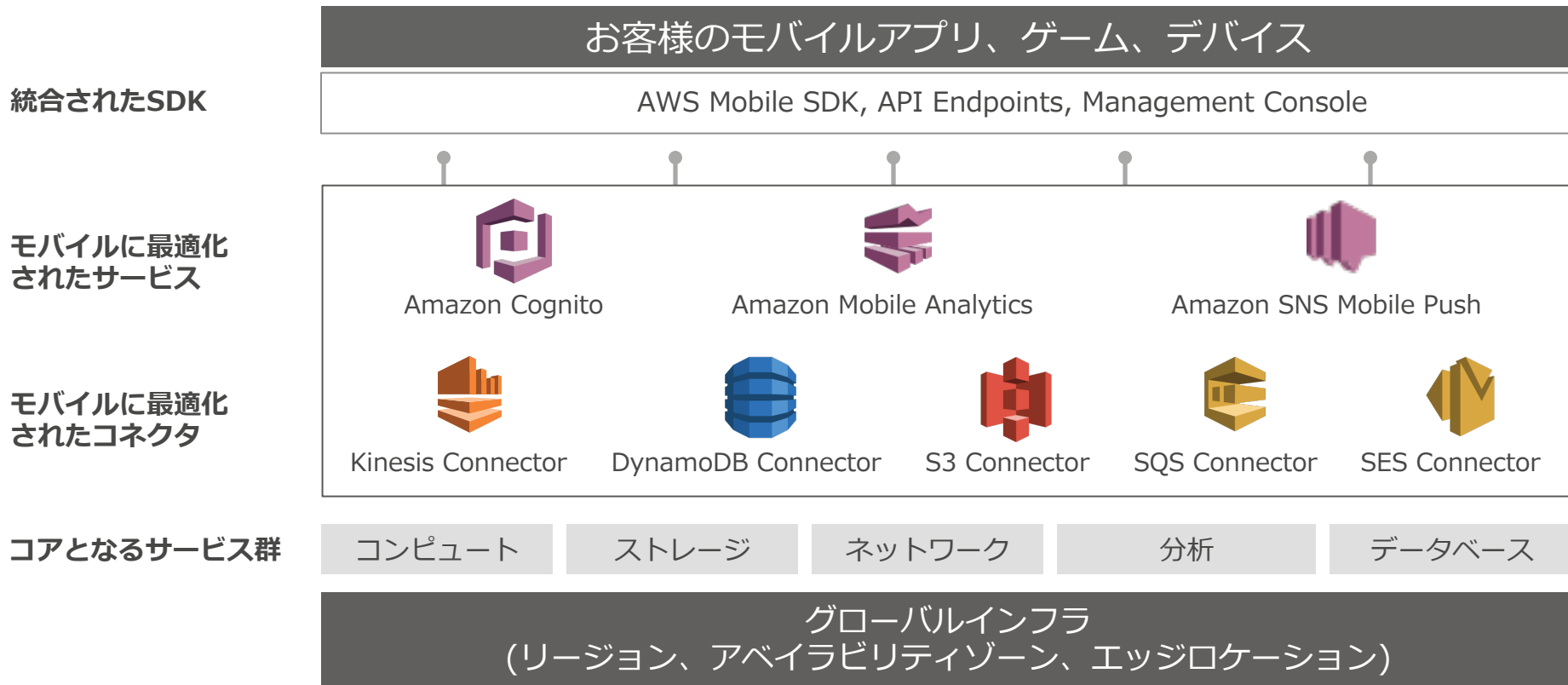
共有データの保存

複数ユーザ、デバイス間における共有データの
保存と高速な検索

データのリアルタイム解析

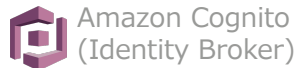
リアルタイムなクリックストリームやログ等の
収集とそれに対する素早いアクションの実施

AWS モバイルサービスのご紹介



モバイルアプリ開発の課題

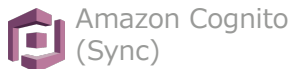
ユーザ認証



アクセスの認可



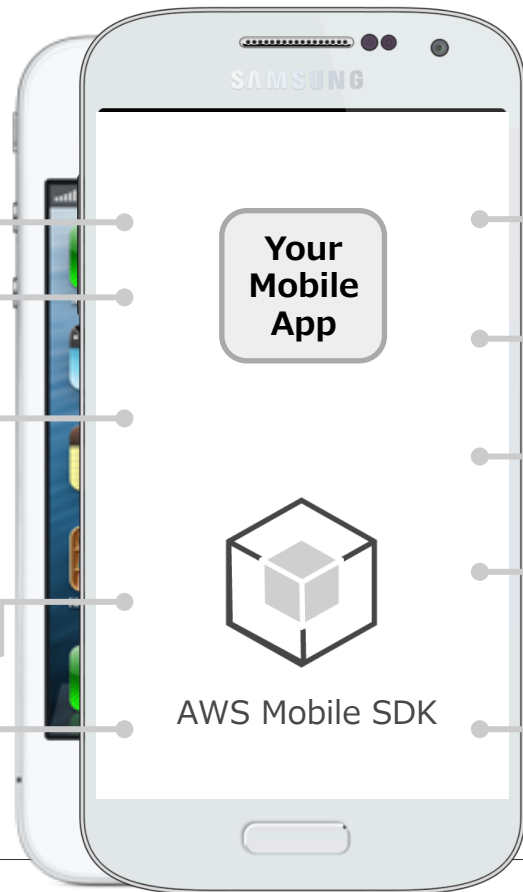
データの同期



ユーザの行動分析



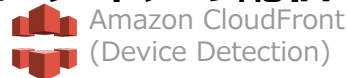
保持率の追跡



メディアの管理



メディアの配信



プッシュ通知の送信



共有データの保存



データのリアルタイム解析



モバイルに最適化されたサービス

おさらし



Amazon Cognito

- ・ ソーシャルログインによる認証
- ・ オフラインでも動作する柔軟なデータ保存／同期機能



Amazon Mobile Analytics

- ・ アプリの利用状況に関するデータの収集と分析
- ・ グラフィカルなレポートニングとデータ連携



Amazon SNS Mobile Push

- ・ クロスプラットフォームなモバイルプッシュ通知
- ・ 簡単なトークン管理

Agenda

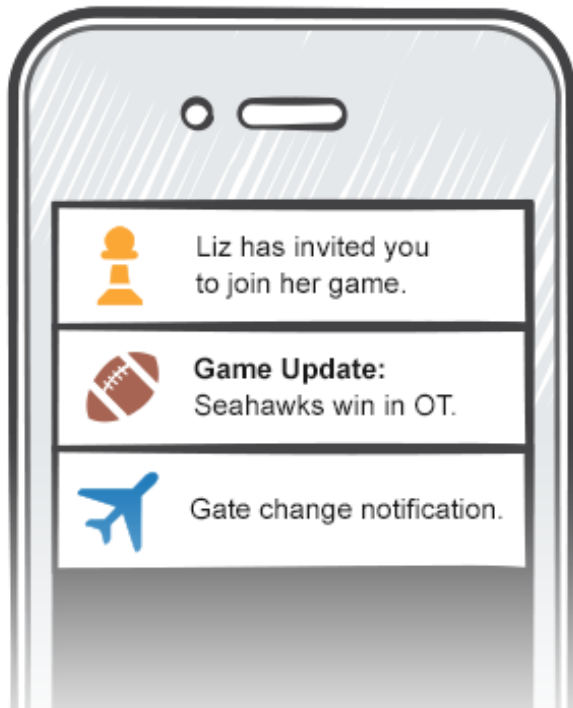
Agenda

- AWS モバイルサービスのご紹介
- **Amazon SNS Mobile Push**
 - モバイルプッシュ通知とは
 - 機能概要
 - SDK, IDE
 - 料金・無料使用枠
- Amazon SNS Mobile Push の使い方
 - マネージメントコンソールから
 - プログラムから
- Q&A

モバイルプッシュ通知とは

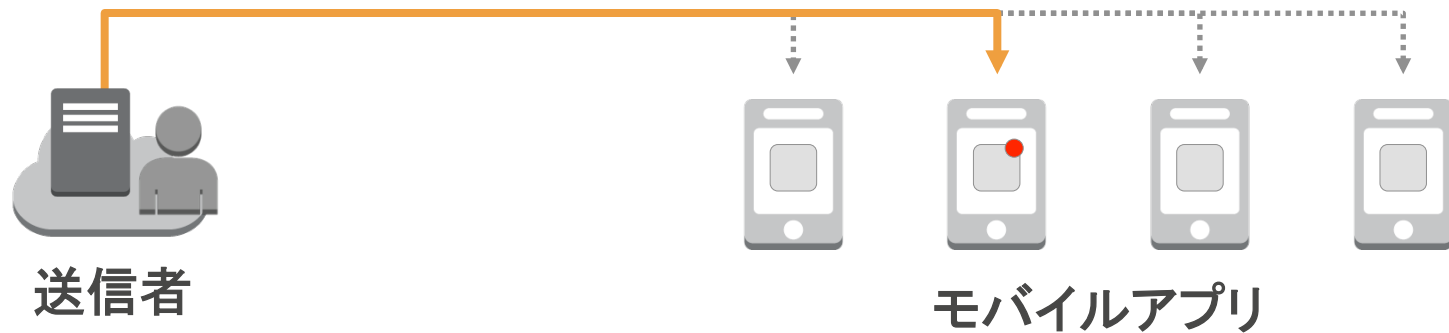


モバイルプッシュ通知ならアプリが起動していなくても通知が可能

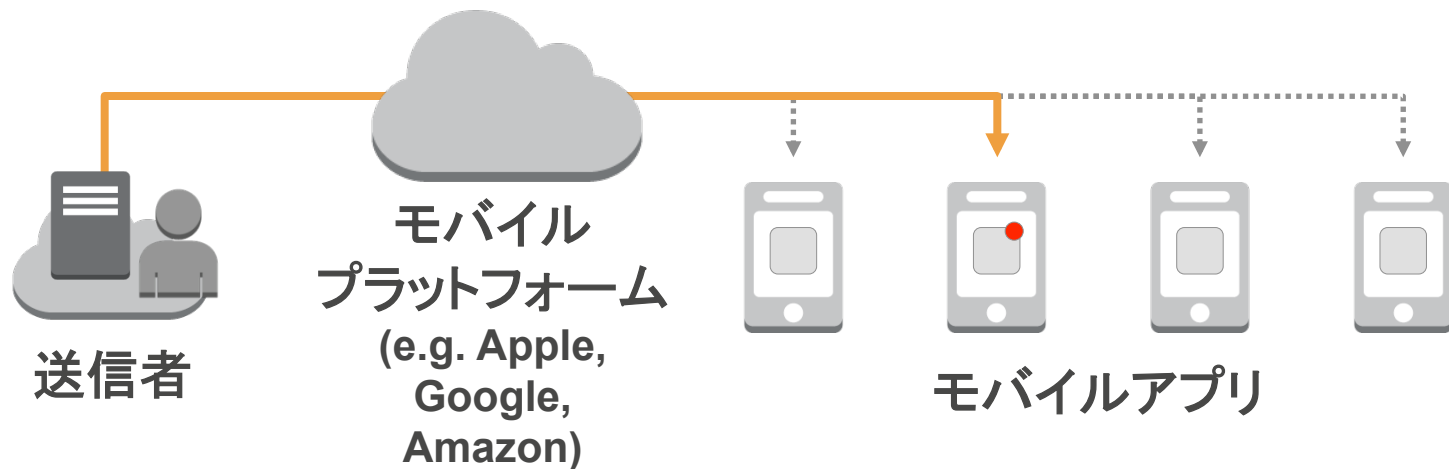


- ユーザは通知を受け取るか否か設定可能
- 特定のデバイス上の特定のアプリにのみ通知する事も可能
- 送るのは短いメッセージ：アプリを起動していただくきっかけ

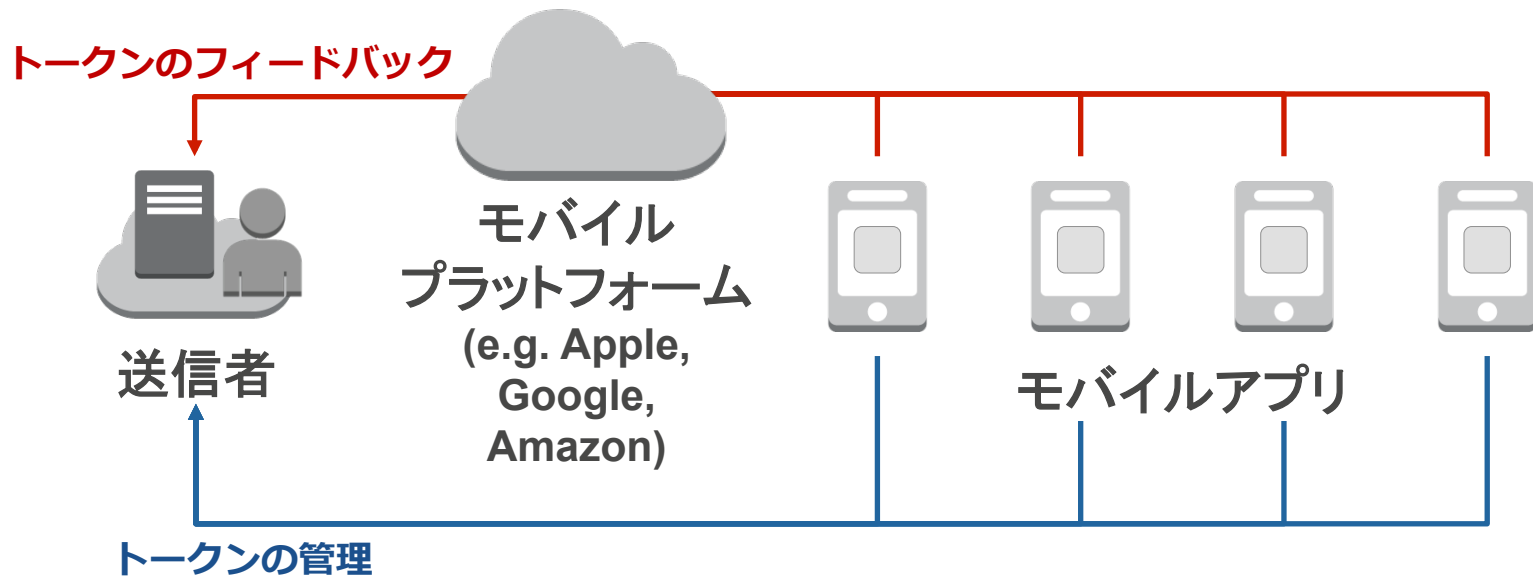
特定のデバイス上の特定のアプリに通知



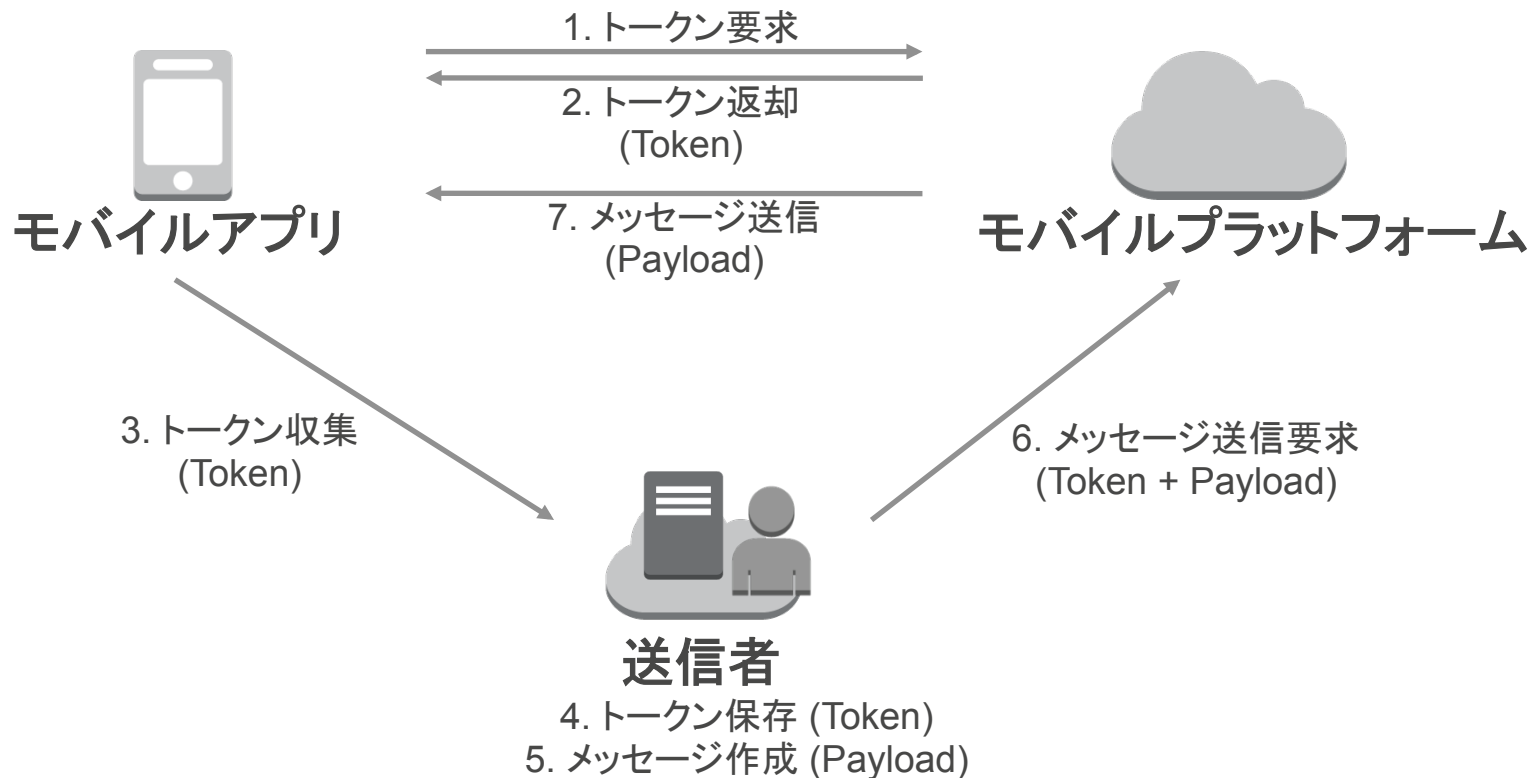
通知は各プラットフォーム独自の仕組みで送信



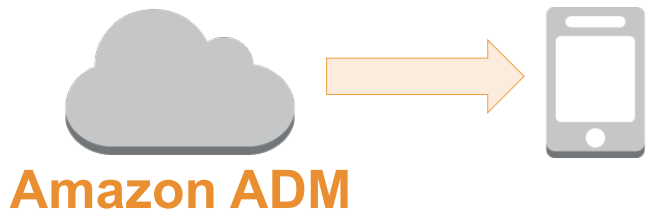
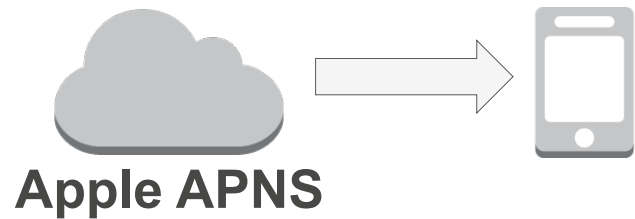
送信者側でデバイスごとのトークンおよびユーザーの受信設定を管理



メッセージを通知するまでのフロー

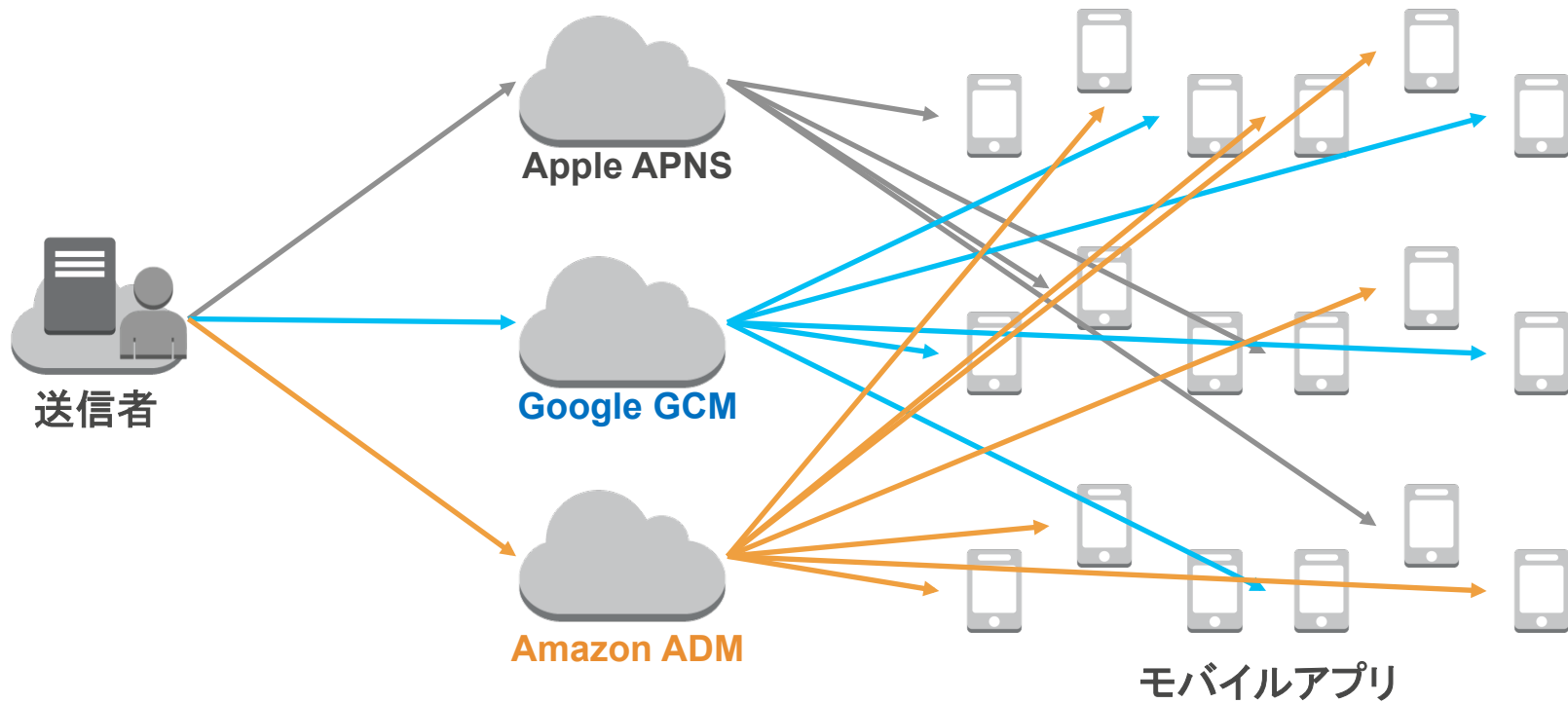


プラットフォームごとに通知の方法は異なる

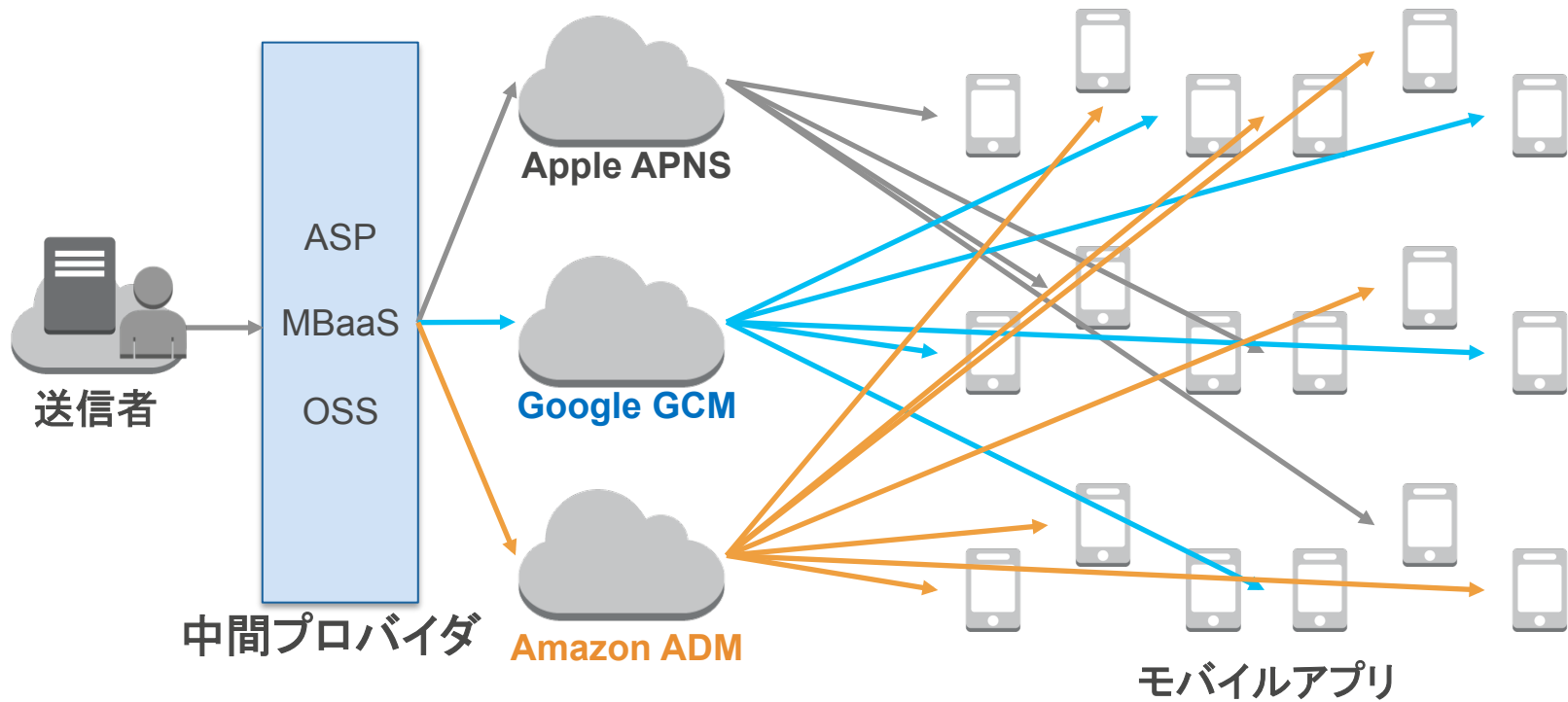


- 異なるAPI
- 異なる機能
- 異なるフィードバック機構

大量のデバイスへの通知の管理運用は大きな問題に



プラットフォームごとのAPIを抽象化した中間プロバイダを利用する



プラットフォームを抽象化することの メリット/デメリット

メリット

- ✓ どのプラットフォームのアプリに対しても簡単に送信できる
- ✓ 信頼性のある通知の管理ができる

デメリット

- ✗ 運用の複雑化
- ✗ スケーラビリティの確保
- ✗ 堅牢性^{けんろうせい}の確保
- ✗ コスト - プラットフォーム側が無料だとしても

プラットフォームを抽象化することの メリット/デメリット

メリット

- ✓ どのプラットフォームのアプリに対しても簡単に送信できる
- ✓ 信頼性のある通知の管理ができる

デメリット

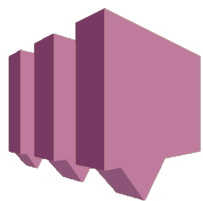
- ~~× 運用の複雑化~~
- ~~× スケーラビリティの確保~~
- ~~× けんろうせい 堅牢性の確保~~
- ~~× コスト - プラットフォーム側が無料だとしても~~

Amazon SNS Mobile Push なら 大丈夫

Amazon SNS Mobile Push はプラットフォーム横断型プッシュ通知のための中間マネージドサービス



送信者



Amazon SNS
Mobile Push

Apple Devices



Google Devices



Amazon Kindle Fire Devices



> その他のプラットフォームにも対応

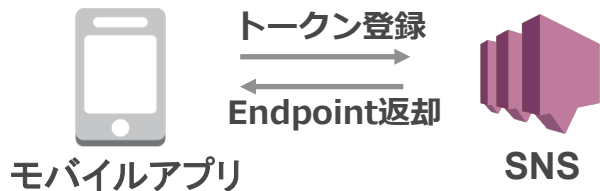
対応しているプラットフォームは以下のとおり

- ✓ Apple Push Notification Service (APNS)
- ✓ Google Cloud Messaging (GCM)
- ✓ Amazon Device Messaging (ADM)
- ✓ Baidu Cloud Push (Baidu)
- ✓ Microsoft Push Notification Service for Windows Phone (MPNS) : Windows 8 ~, Windows Phone 8.1 ~
- ✓ Windows Push Notification Services (WNS) : Windows Phone 7 ~

Endpoint と Topic について用語説明

Endpoint

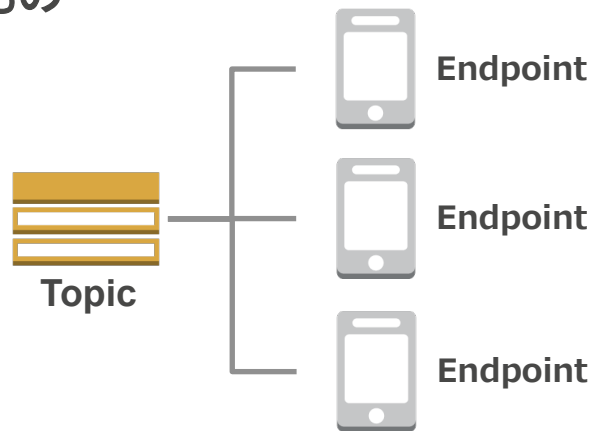
デバイスを SNS に登録すると発行される SNS 上でのデバイス識別子



トークン	Endpoint
1ebf21e3e1c56d061be2c520f448.....	arn:aws:sns:ap-northeast-1:xxxxxxx:endpoint/APNS_SANDBOX/MobilePushTest/xxxx-xxxx-xxxx-xxxx1
1d3a8db9eba038315385cdb892ab.....	arn:aws:sns:ap-northeast-1:xxxxxxx:endpoint/APNS_SANDBOX/MobilePushTest/xxxx-xxxx-xxxx-xxxx2
2abc3273330ea22b8f95503b93ee3.....	arn:aws:sns:ap-northeast-1:xxxxxxx:endpoint/APNS_SANDBOX/MobilePushTest/xxxx-xxxx-xxxx-xxxx3

Topic

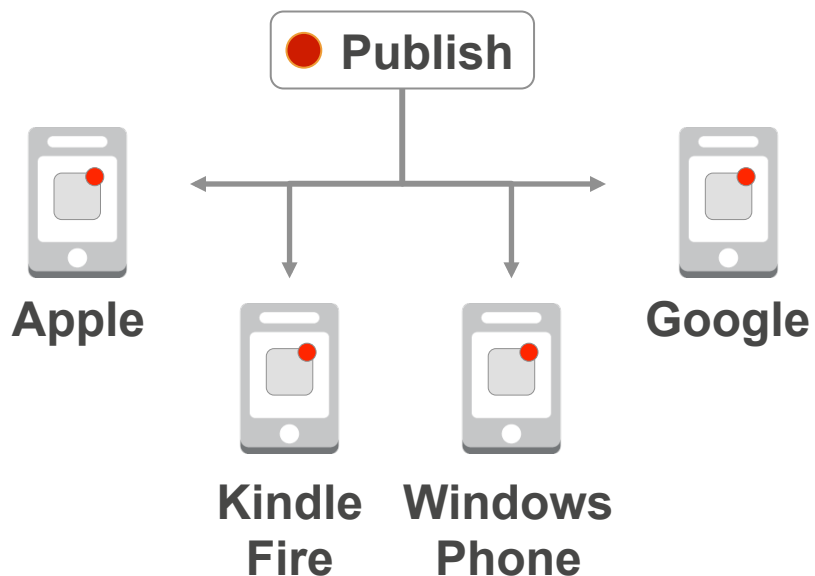
複数の Endpoint を束ねてグルーピングしたもの



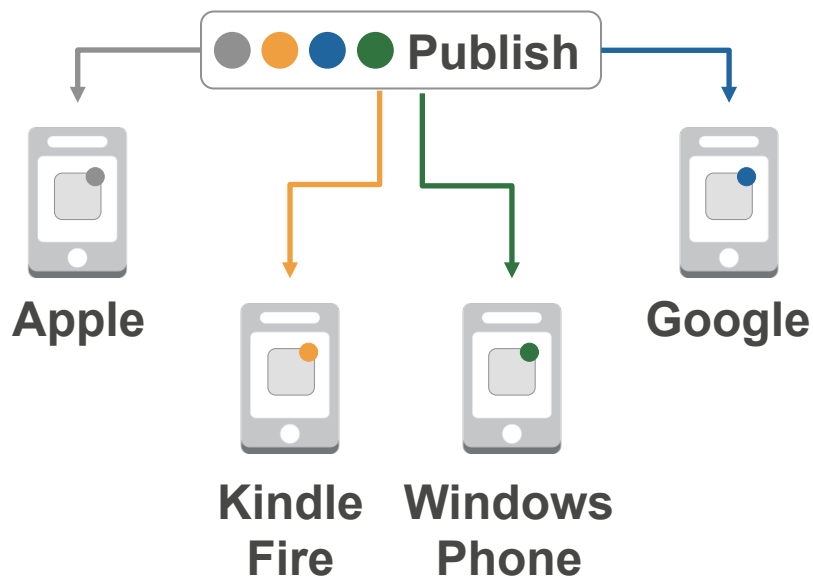
1トピックあたり
最大1,000万サブスクリプション(緩和可)
3,000トピックまで作成可

プラットフォーム特有のメッセージペイロードを通知することも可能

一度のPublish指示で同じメッセージを送信

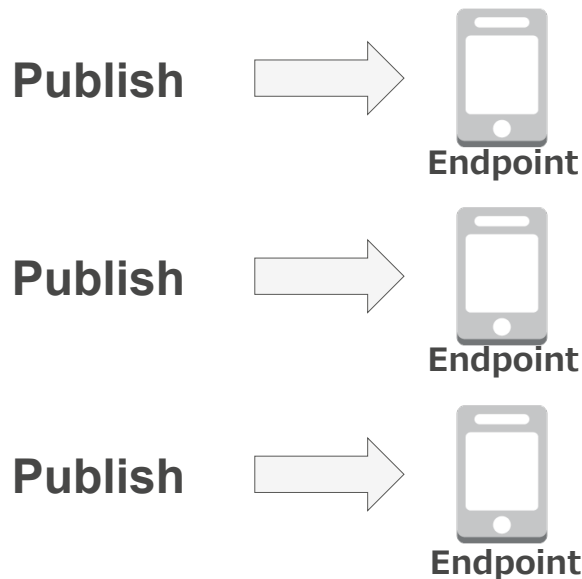


プラットフォームそれぞれに異なるペイロードを送信

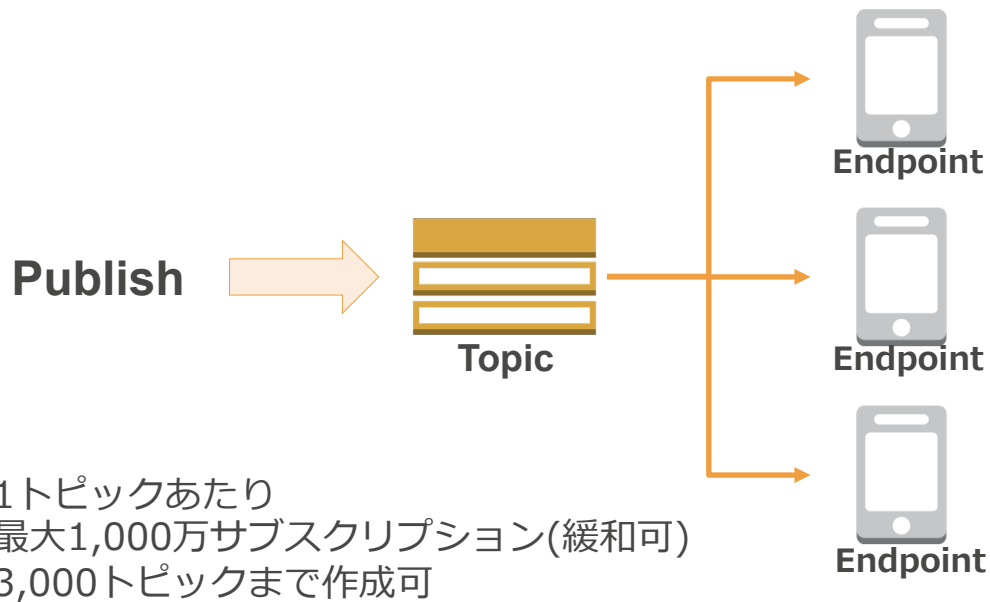


デバイスごとにメッセージを通知することも、グループに一斉通知することも可能

デバイスごとに直接通知



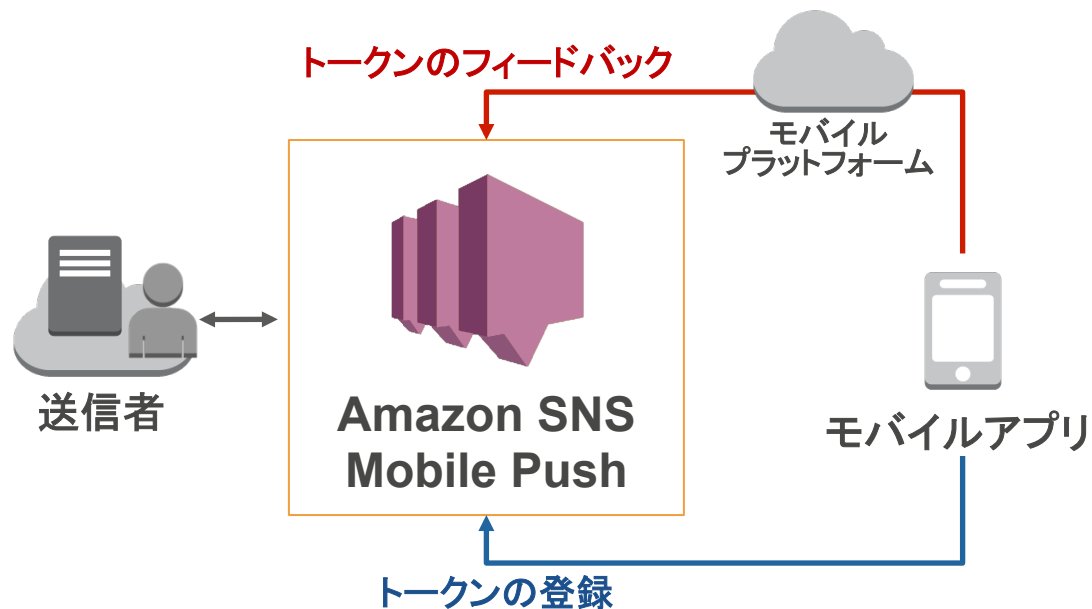
Topic による一斉通知



メッセージサイズの上限はプラットフォームごと

- **Apple Push Notification Service (APNS)**
 - 256 bytes(~iOS7), 2KB(iOS8)
 - <https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/Chapters/ApplePushService.html>
- **Google Cloud Messaging (GCM)**
 - 4096 bytes
 - <http://developer.android.com/intl/ja/google/gcm/adv.html>
- **Amazon Device Messaging (ADM)**
 - 6144 bytes
 - <https://developer.amazon.com/sdk/adm/sending-message.html>
- **Baidu Cloud Push (Baidu)**
 - 256 bytes
 - http://docs.aws.amazon.com/ja_jp/sns/latest/dg/SNSMobilePushBaiduPublish.html
- **Microsoft Push Notification Service for Windows Phone (MPNS)**
 - 3072 bytes
 - <https://msdn.microsoft.com/ja-jp/library/ff402558%28v=vs.92%29.aspx>
- **Windows Push Notification Services (WNS)**
 - 5000 bytes
 - <https://msdn.microsoft.com/ja-jp/library/windows/apps/hh465435.aspx#WNSResponseCodes>

各プラットフォームからのフィードバックを基に トークンの入れ替えや無効化を自動実行



- デバイスごとに安定したエンドポイントを提供
- トークンフィードバックを参照可能

多くの言語とプラットフォームをカバーするSDKとIDE

お好みの言語やプラットフォームのSDKを選択



iOS



Android



JavaScript



Java



Python



PHP



.NET



Ruby



Node.js

お好みのIDEに統合



Eclipse



Visual Studio



CLI



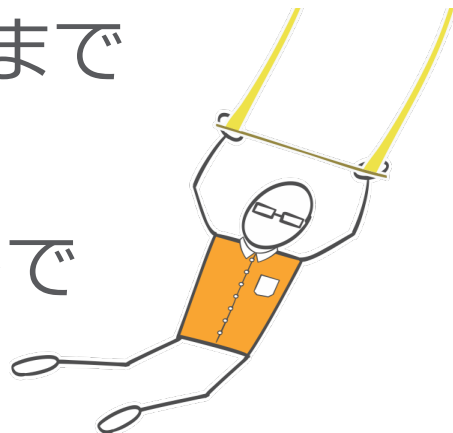
Powershell

コマンドラインから

Amazon SNS Mobile Push のスケーラビリティならアプリが必要とする分だけの通知をいつでも送信可能

- 数通のメッセージから億単位のメッセージまで
- 数個のデバイスから億単位のデバイスまで

...これらを設定 / 運用管理の手間なしで
けんろうせい
...堅牢性や信頼性を心配せず



料金・無料使用枠

ひと月あたり 100 万リクエストまで無料

- 100 万の Publish (アプリ→ SNS)
- 100 万のモバイルプッシュ通知 (SNS →各プラットフォーム)

以降は 100 万リクエストあたり \$1.00

- \$0.50 で 100 万 publish (アプリ→ SNS)
- \$0.50 で 100 万モバイルプッシュ通知 (SNS →各プラットフォーム)

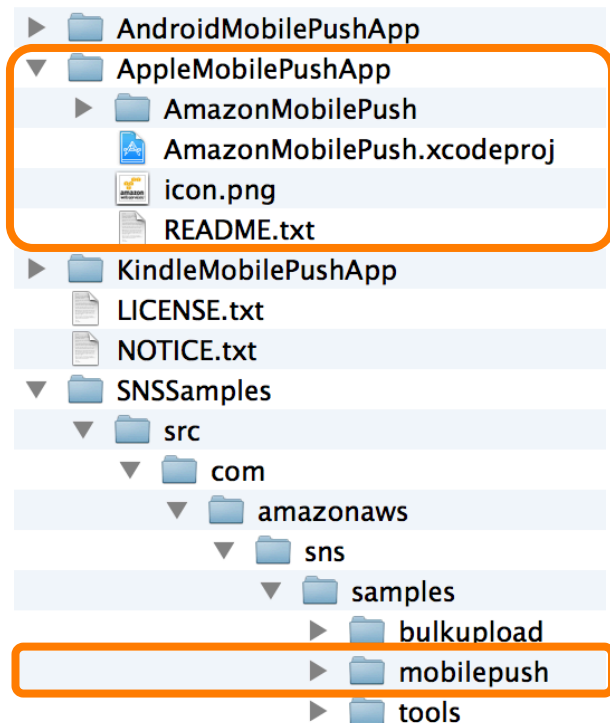
Agenda

Agenda

- AWS モバイルサービスのご紹介
- Amazon SNS Mobile Push
 - モバイルプッシュ通知とは
 - 機能概要
 - SDK, IDE
 - 料金・無料使用枠
- **Amazon SNS Mobile Push の使い方**
 - マネージメントコンソールから
 - プログラムから
- Q&A

サンプルアプリケーション

<https://s3.amazonaws.com/codesamples/sns/latest/snsmobilepush.zip>



iOS 受信サンプルアプリ
(~iOS7.1対応)

Java サンプルアプリ

iOS 受信サンプルアプリ

The screenshot shows the Xcode IDE with the AppDelegate.m file open. The code is as follows:

```
#import "AppDelegate.h"

@implementation AppDelegate

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions{
    application.applicationIconBadgeNumber = 0;
    [[UIApplication sharedApplication] registerForRemoteNotificationTypes:
    (UIRemoteNotificationTypeBadge | UIRemoteNotificationTypeSound | UIRemoteNotificationTypeAlert)];
    if(launchOptions!=nil){
        NSString *msg = [NSString stringWithFormat:@"%@", launchOptions];
        NSLog(@"%@",msg);
        [self createAlert:msg];
    }
    return YES;
}

- (void)application:(UIApplication*)application didRegisterForRemoteNotificationsWithDeviceToken:(NSData*)deviceToken{
    NSLog(@"deviceToken: %@", deviceToken);
}

- (void)application:(UIApplication*)application didFailToRegisterForRemoteNotificationsWithDeviceToken:(NSData*)deviceToken:(NSError*)error{
    NSLog(@"Failed to register with error : %@", error);
}

- (void)application:(UIApplication *)application didReceiveRemoteNotification:(NSDictionary *)userInfo {
    application.applicationIconBadgeNumber = 0;
    NSString *msg = [NSString stringWithFormat:@"%@", userInfo];
    NSLog(@"%@",msg);
    [self createAlert:msg];
}

- (void)createAlert:(NSString *)msg {
    UIAlertView *alertView = [[UIAlertView alloc] initWithTitle:@"Message Received" message:[NSString stringWithFormat:@"%@", msg]
    delegate:self cancelButtonTitle:@"OK" otherButtonTitles:nil];
    [alertView show];
}

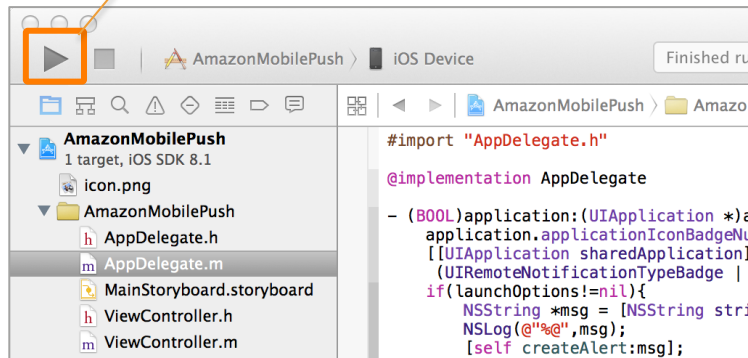
@end
```

Callouts in the image explain the following code sections:

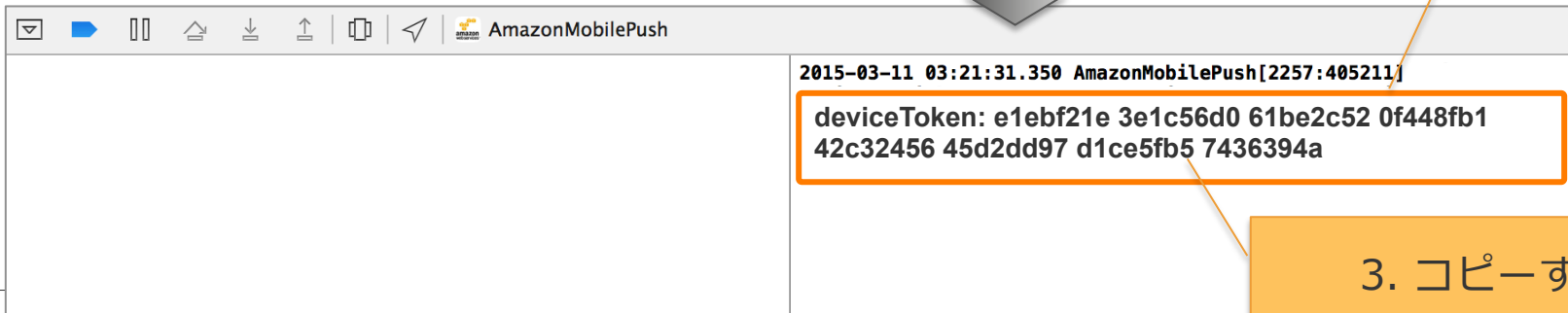
- didFinishLaunchingWithOptions** ハンドラで Device Token を APNS に要求する
- didRegisterForRemotoNotification...** ハンドラで Device Token を取得できたらログに書き出す
- didRecieveForRemotoNotification...** ハンドラでメッセージを受信したらアラートダイアログを表示する

iOS 受信サンプルアプリ

1. [▶] をクリック



2. コンソールに
トークンが表示される



3. コピーする

アプリケーションを作成

The screenshot shows the Amazon SNS console interface. On the left is a 'Navigation' sidebar with a 'Create and Add' dropdown, a 'Refresh' button, a 'Topic Filter' input, and a 'Clear' button. Below these are links for 'SNS Dashboard', 'Apps (0)', 'Subscriptions', and 'Topics (0)'. The main area is the 'Dashboard' with a 'Getting Started' section. It contains a text block about SNS, a sub-header 'Add a New App to start using Mobile Push. Create a New Topic to notify multiple recipients on any protocol.', and two buttons: 'Add a New App' and 'Create New Topic'. The 'Add a New App' button is highlighted with an orange border and a callout arrow pointing to an instruction box at the bottom. The instruction box contains the text '1. [Add a New App] をクリック'.

Navigation

Create and Add ▾ Refresh

Topic Filter Clear

- SNS Dashboard
- Apps (0)
- Subscriptions
- Topics (0)

Dashboard

Getting Started

Amazon Simple Notification Service (SNS) is a fast, flexible, fully managed push messaging service. SNS makes it simple and cost-effective to push messages to mobile devices such as iPhone, iPad, Android, Kindle Fire, and Internet-connected smart devices, as well as pushing to other distributed services. Besides pushing directly to mobile devices, SNS can also deliver notifications by SMS text message or email to Amazon Simple Queue Service (SQS) queues or to any HTTP endpoint.

Add a New App to start using Mobile Push. Create a New Topic to notify multiple recipients on any protocol.

[Add a New App](#) [Create New Topic](#)

Your resources will be created in the Asia Pacific (Tokyo) region.

1. [Add a New App] をクリック

アプリケーションを作成

Application Details

Application Actions

Add Your Application to Amazon SNS

Enter application name and select the push platform for your application ([learn more](#))

Application Name: MobilePushTest

Push Platform: Apple Push Notification Service Sandbox (APNS_SANDBOX)

Enter the credentials your app uses to connect to the selected push platform. By uploading these credentials to Amazon SNS, you are indicating you have the right to use these credentials, allowing Amazon SNS to use them on your behalf.

Apple Push Notification Service Sandbox (APNS_SANDBOX) Credentials:

Use this form to load your Apple credentials from a P12 file, or copy and paste them into the Certificate and Private Key text boxes below.

Choose P12 File: Certificates.p12

Enter Password:

Your Apple credentials should appear in the following text boxes in PEM format.

Certificate:

Private Key:

1. [Application Name] を入力
2. [Push Platform] は [APNS_SANDBOX] を選択

3. [Browse] をクリックして [証明書] を選択
4. [Enter Password] にパスワードを入力
5. [Load Credentials from File] をクリック

6. [Add New App] をクリック

Navigation

Create and Add

Topic Filter

- SNS Dashboard
- Apps (1)
 - MobilePushTest (APNS_SANDBOX)

作成完了！

※アプリケーションが作成される

トークンを登録

1. [Add Endpoint]
をクリック

Add Endpoints Cancel

Application: MobilePushTest (APNS_SANDBOX)

Endpoints represent instances of your app installed on particular devices. Endpoints correspond to the device tokens generated by each push platform. You can create endpoints from existing tokens one at a time or upload them in bulk ([learn more](#)).

Manually add a single token

Device Token:

User Data:

Upload multiple tokens in a CSV file ([Download sample CSV](#)).

4. [Add Endpoint] をクリック

2. [先ほどコピーした
トークン] を入力

3. [User Data] を入力

※CSV で一括登録も可能
(最大2MB)

※User Data は
任意のデータであり
SNS では使用されない
(最大2KB, UTF8)

トークンを登録

※トークンが登録される

	Token	EndpointArn
<input checked="" type="checkbox"/>	e1ebf21e3e1c5	arn:aws:sns:ap- /MobilePushTest/293c8942-faf5-3202-8076-86189013682a
<input type="checkbox"/>	1d3a8db9eba0:	arn:aws:sns:ap- /MobilePushTest/de8d5669-eff0-3178-be1e-a9226f83f3a1

登録完了！

メッセージを送信

- Endpoint ごとに送信する

1. [Endpoint] を
1件選択

2. [Publish] をクリック

Endpoint Actions	Add Endpoints	Publish	Clear	Endpoint Filter
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="Publish"/>	<input type="button" value="Clear"/>	<input type="text" value="Endpoint Filter"/>
Token	EndpointArn			
<input checked="" type="checkbox"/>	e1ebf21e3e1c5 arn:aws:sns: /MobilePushTest/293c8942-faf5-3202-8076-86189013682a			
<input type="checkbox"/>	1d3a8db9eba0 arn:aws:sns: /MobilePushTest/de8d5669-eff0-3178-be1e-a9226f83f3a1			



Application: MobilePushTest (APNS_SANDBOX)

Endpoint: 293c8942-faf5-3202-8076-86189013682a

Message:

3. [Message] を入力

4. [TTL] を入力

Time to Live (TTL):

TTL is the number of seconds since the message
When you use TTL, messages that remain undelivered for the specified time will expire.

※TTL(秒) を超えて
受信されないメッセージ
は破棄される

5. [Use text format] を選択

Use text format

Use platform specific JSON message dictionaries

※プラットフォーム独自
フォーマットにも対応

6. [Publish Message] をクリック

送信完了!

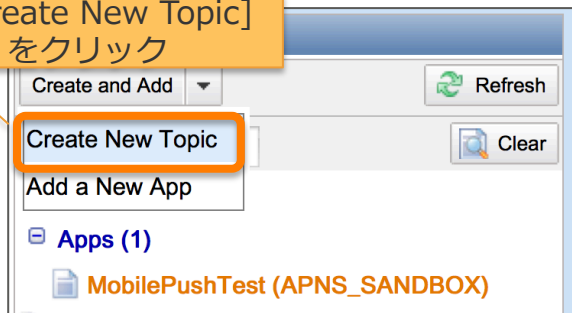
スマートフォンアプリでメッセージを受信



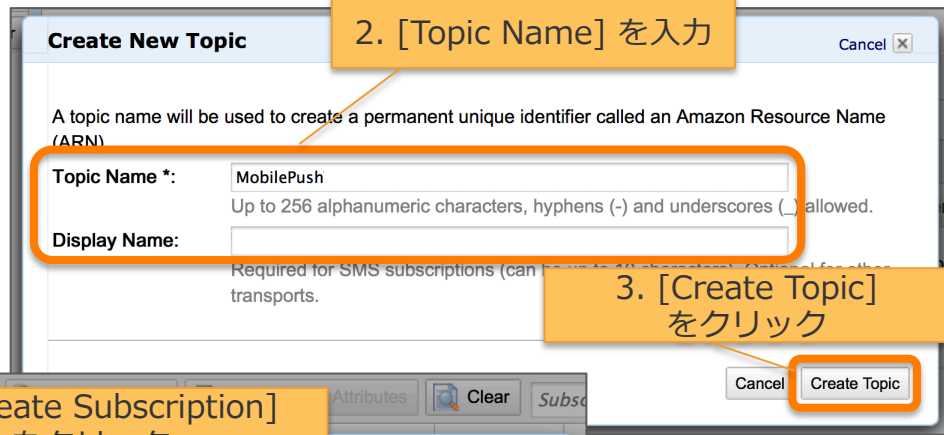
メッセージを一斉送信

・ Topic を作成して Endpoint を Subscribe する

1. [Create New Topic]
をクリック

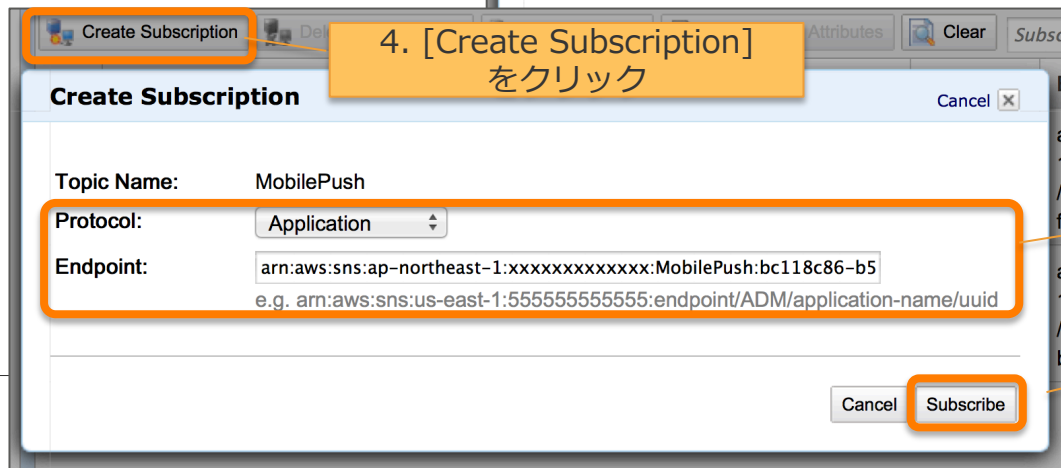


2. [Topic Name] を入力



3. [Create Topic]
をクリック

4. [Create Subscription]
をクリック



5. [Application] を選択
6. [Endpoint] を入力

7. [Subscribe]
をクリック

メッセージを一斉送信

- ・ Topic で一斉送信する

1. [Publish] をクリック

2. 適宜入力

3. [Publish Message] をクリック

※プラットフォーム独自フォーマットにも対応

送信完了!

```
Topic Name: MobilePush
Subject: Hello World
Message: こんにちは
Time to Live (TTL): 100
Use same message body for all protocols
Use different message body for different protocols
```

```
Topic Name: MobilePush
Subject:
Message: {
  "default": "ENTER YOUR MESSAGE",
  "email": "ENTER YOUR MESSAGE",
  "sqs": "ENTER YOUR MESSAGE",
  "http": "ENTER YOUR MESSAGE",
  "https": "ENTER YOUR MESSAGE",
  "sms": "ENTER YOUR MESSAGE",
  "APNS": "{ \"aps\": { \"alert\": \"ENTER YOUR MESSAGE\", \"sound\": \"default\" } }",
  "GCM": "{ \"data\": { \"message\": \"ENTER YOUR MESSAGE\" } }",
  "fcm": "{ \"notification\": { \"title\": \"ENTER YOUR MESSAGE\", \"body\": \"ENTER YOUR MESSAGE\" } }"
}
```

複数のスマートフォンアプリでメッセージを受信



トークンを Amazon SNS Mobile Push へ登録する方法

- モバイルアプリ → SNS (2-tier)

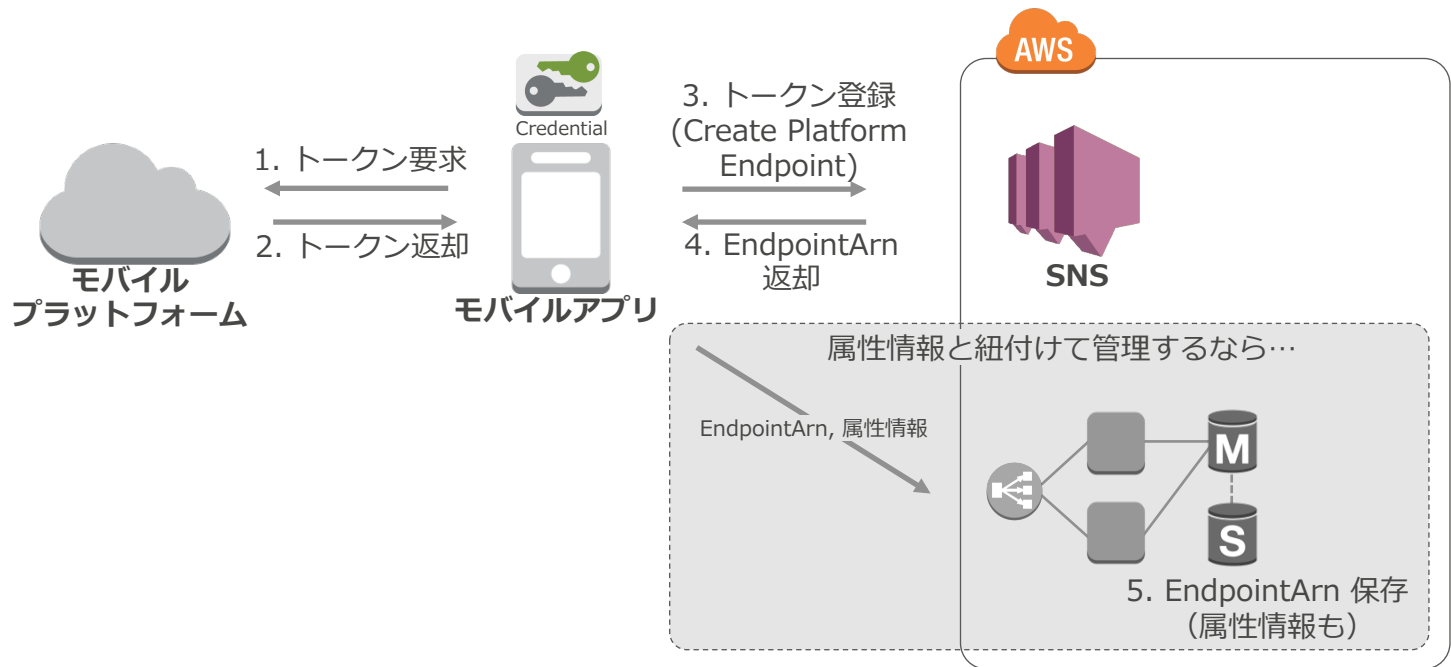
- ダメ** • ~~Credential をモバイルアプリに埋め込んでトークンを SNS へ登録~~
- Token Vending Machine で Temporary credential を発行してトークンを SNS へ登録

- シンプル** • Amazon Cognito で Temporary credential を発行してトークンを SNS へ登録

- モバイルアプリ → ウェブアプリ → SNS (3-tier)

- 多機能** • 独自構築ウェブアプリの API 経由でトークン SNS へ登録

Credential をモバイルアプリに埋め込んで トークンを SNS へ登録



Credential をモバイルアプリに埋め込んで トークンを SNS へ登録



メリット

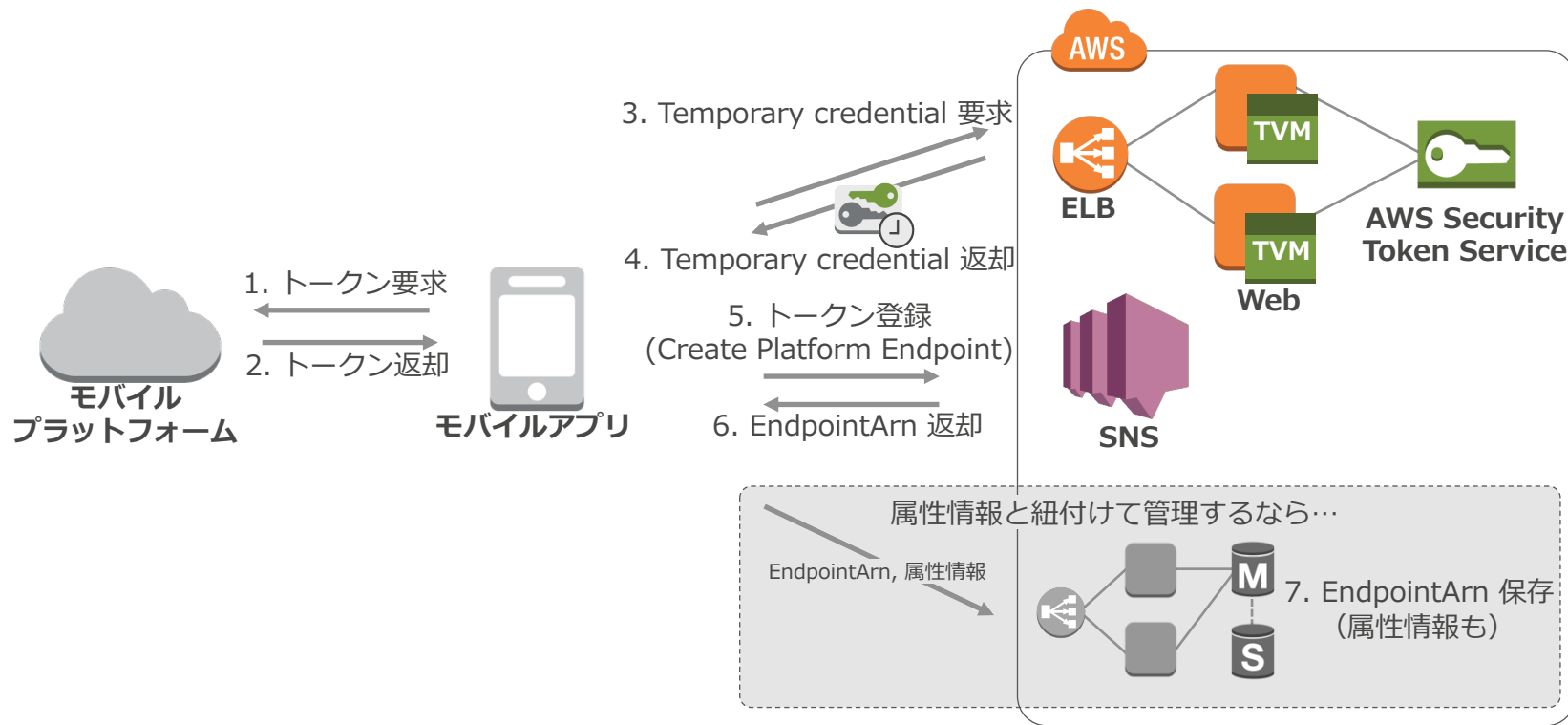
- ✓ 構築が非常に簡単
- ✓ 低コスト

デメリット

- ✗ Credential 漏洩の可能性
- ✗ 権限の絞込み必須
- ✗ 変更が必要な場合にアップデートするのが大変（ユーザがアップデートしてくれない、審査に時間がかかる）
- ✗ 複雑な通知を実現するには別途ウェブアプリが必要

アンチパターン！

Token Vending Machine で Temporary credential を発行してトークンを SNS へ登録



Token Vending Machine で Temporary credential を発行してトークンを SNS へ登録

メリット

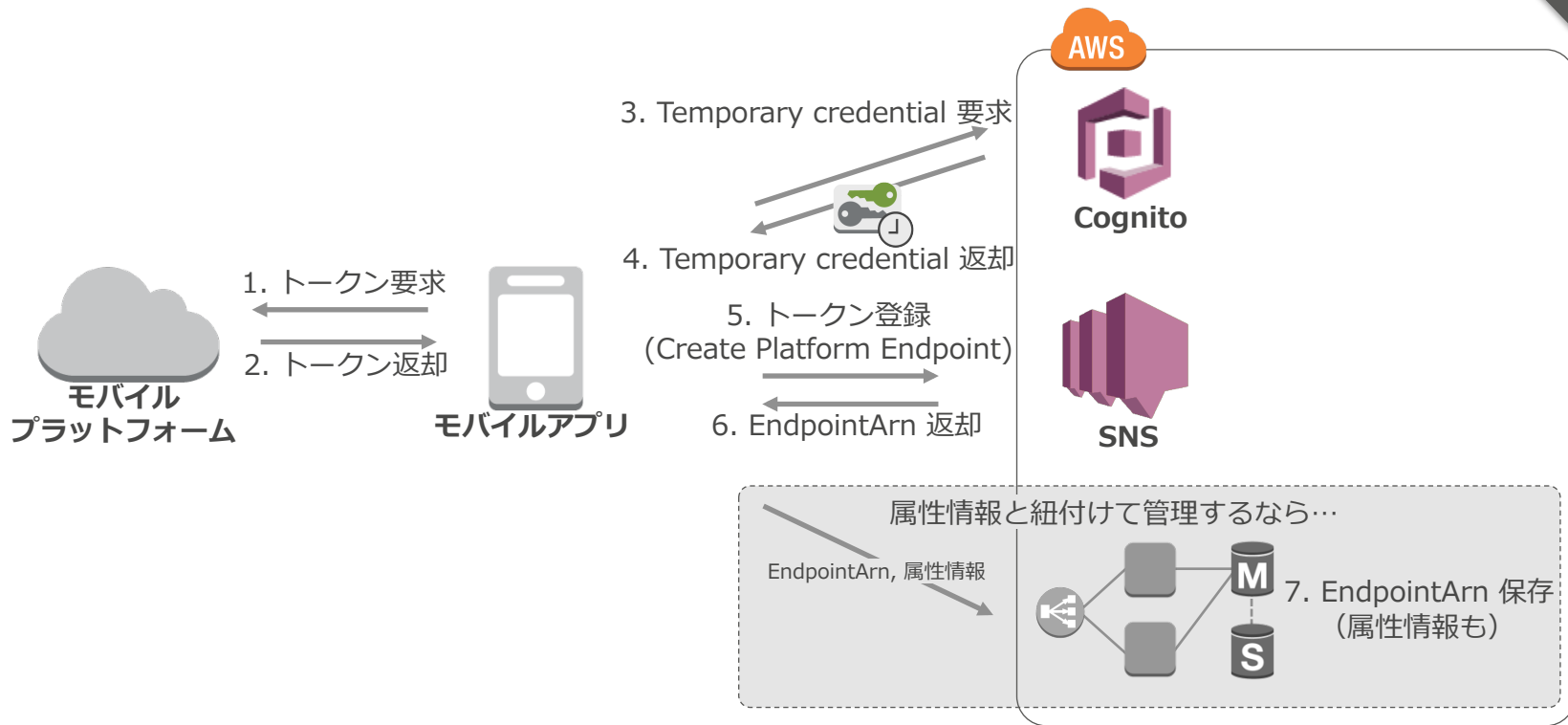
- ✓ Credential 漏洩の回避
- ✓ 変更が必要な場合にアップデートするのが容易

デメリット

- ✗ TVM の構築と運用
- ✗ スケーラビリティの確保
- ✗ けんろうせい 堅牢性の確保
- ✗ 複雑な通知を実現するには別途ウェブアプリが必要

Amazon Cognito で Temporary credential を発行してトークンを SNS へ登録

おすすめ



Amazon Cognito で Temporary credential を発行してトークンを SNS へ登録

おすすめ

メリット

- ✓ 構築が非常に簡単
- ✓ Credential 漏洩の回避
- ✓ スケーラビリティの確保
- ✓ 堅牢性の確保

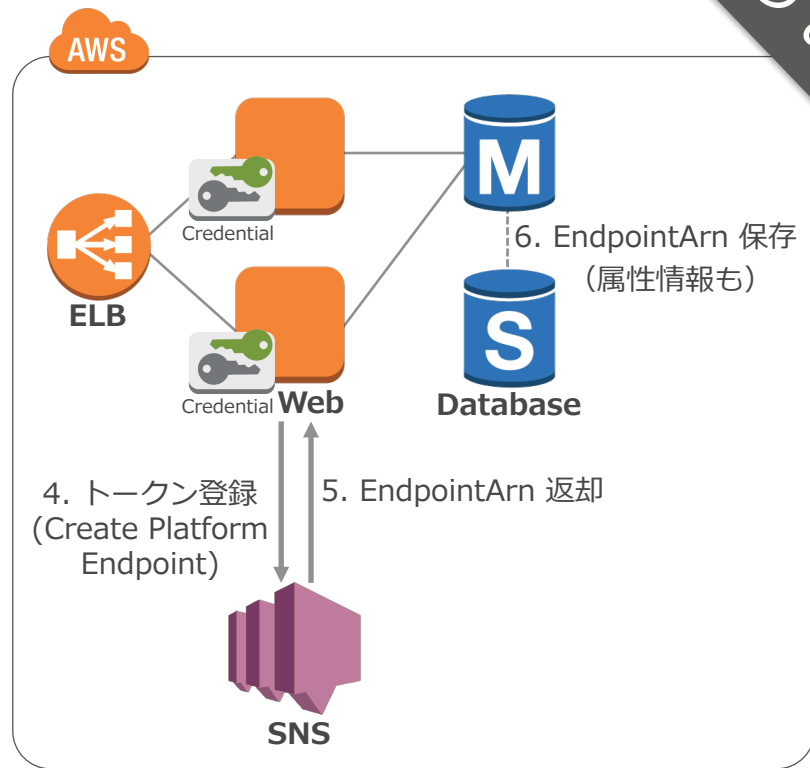
デメリット

- ✗ 複雑な通知を実現するには別途ウェブアプリが必要

シンプルな要件にマッチ！

独自構築ウェブアプリの API 経由で トークンを SNS へ登録

おすすめ



独自構築ウェブアプリの API 経由で トークンを SNS へ登録

おすすめ

メリット

- ✓ 属性情報とトークンを紐付けて柔軟に管理できる
- ✓ 検索やグルーピングによって複雑な通知を実現できる

デメリット

- ✗ ウェブアプリの構築と運用
- ✗ スケーラビリティの確保
- ✗ 堅牢性^{けんろうせい}の確保
- ✗ コスト

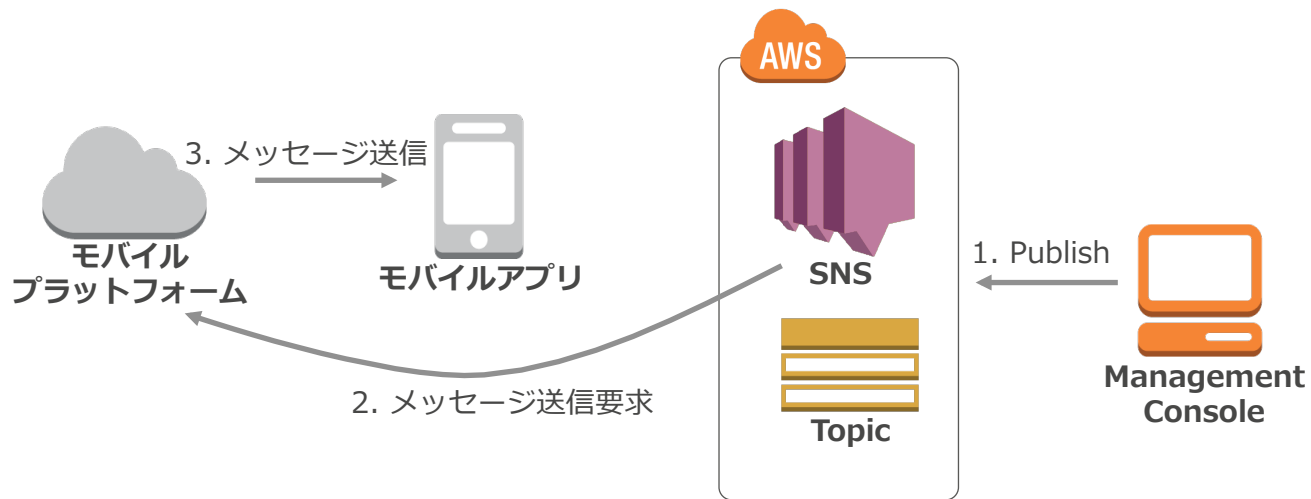
高機能的要件にマッチ！

Amazon SNS Mobile Push からメッセージを送信する方法

- マネージメントコンソールから送信
→ 前述のとおり、1件送信(Endpoint) or 一斉送信(Topic)
- プログラムからトークンと属性情報を紐付けて検索のうえ送信

マネージメントコンソールから送信

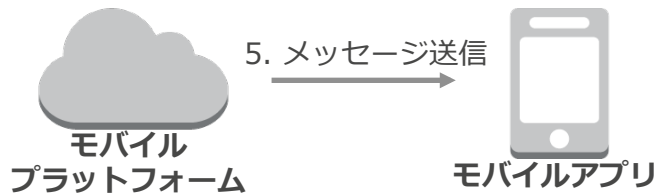
おすすめ



シンプルな要件にマッチ！

プログラムからトークンと属性情報を紐付けて検索のうえ送信

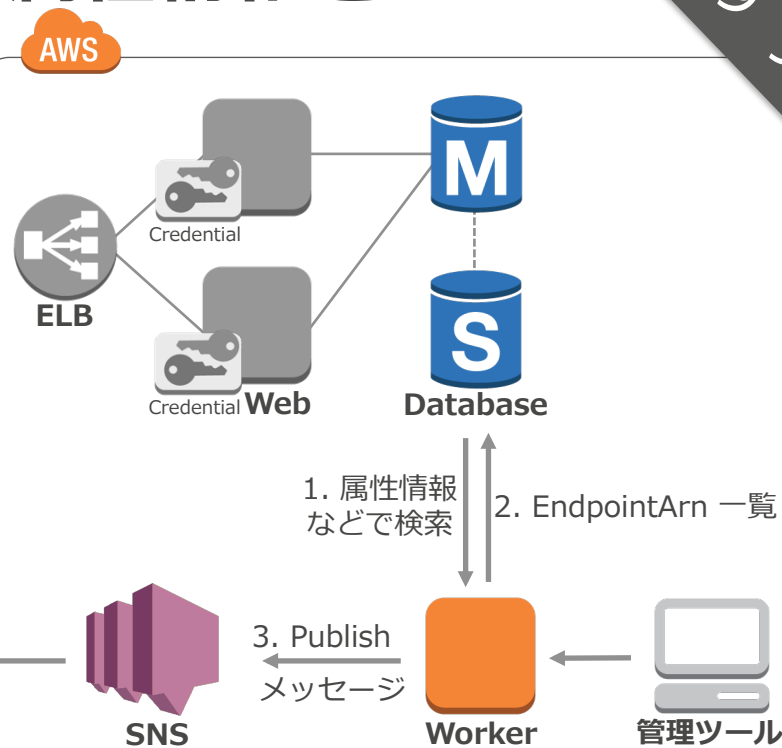
おすすめ



モバイルプラットフォーム

モバイルアプリ

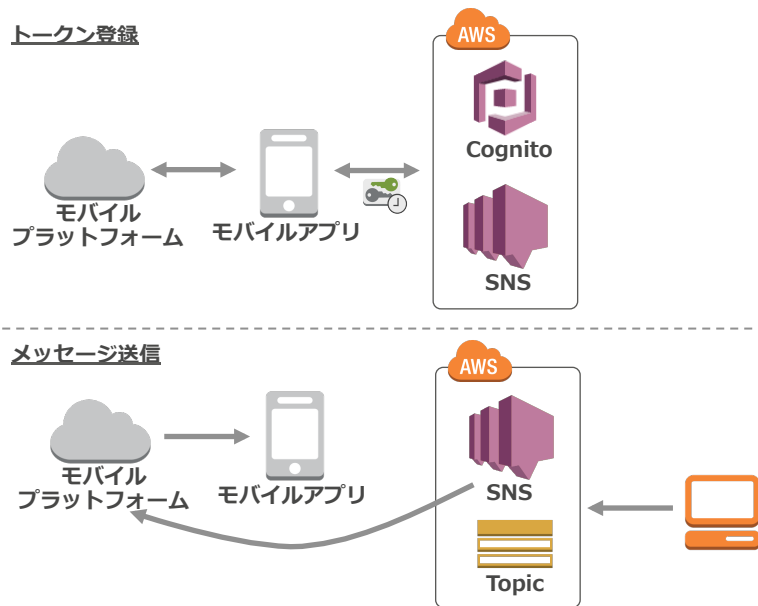
4. メッセージ送信要求



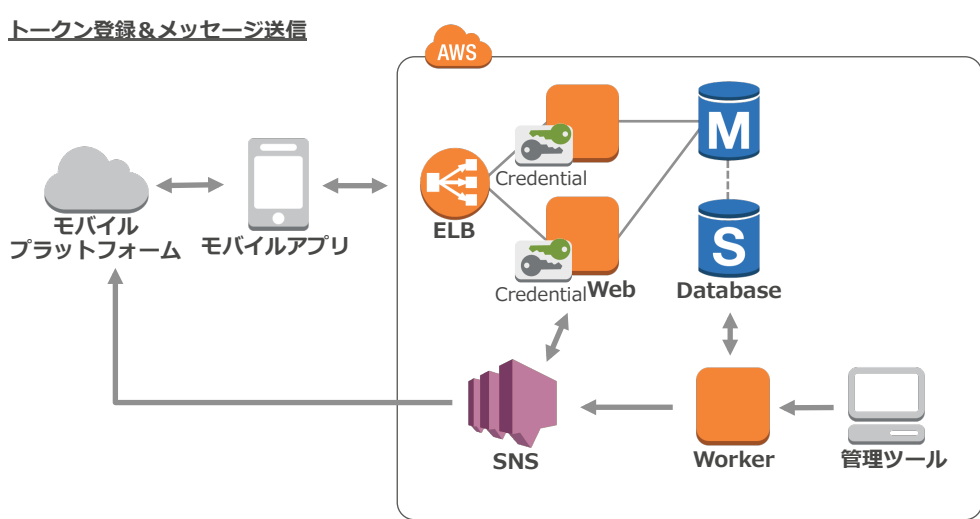
高機能な要件にマッチ！

トークン登録／メッセージ送信のまとめ

シンプル



高機能



トークンを登録して Endpoint を生成する Java サンプル

```
String applicationArn = platformApplicationResult.getPlatformApplicationArn();  
  
CreatePlatformEndpointRequest platformEndpointRequest =  
    new CreatePlatformEndpointRequest();  
  
platformEndpointRequest.setCustomUserData(customData,,  
platformEndpointRequest.setToken(platformToken);  
platformEndpointRequest.setPlatformApplicationArn(applicationArn);  
  
CreatePlatformEndpointResult platformEndpointResult =  
    snsClient.createPlatformEndpoint(platformEndpointRequest);
```

端末から取得したトークン

アプリケーションARN

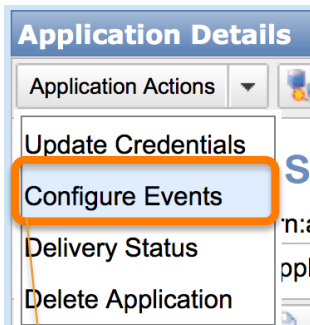
Endpoint にメッセージを送信する Java サンプル

```
PublishRequest publishRequest = new PublishRequest();  
  
Map<String, String> messageMap = new HashMap<String, String>();  
messageMap.put(platform.name(), getPlatformSampleMessage(platform));  
publishRequest.setTargetArn(platformEndpointResult.getEndpointArn());  
publishRequest.setMessageStructure("json");  
  
message = jsonify(messageMap);  
publishRequest.setMessage(message);  
  
PublishResult publishResult = snsClient.publish(publishRequest);
```

メッセージを設定

Publish でメッセージ送信

Event 管理の設定



1. [Configure Events]
をクリック



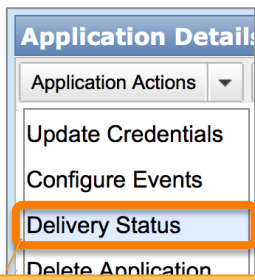
2. Endpointの追加 / 削除 / 更新 / 送信失敗 という
Eventが発生した場合にSNSで検知が可能

3. [Save Configuration]
をクリック

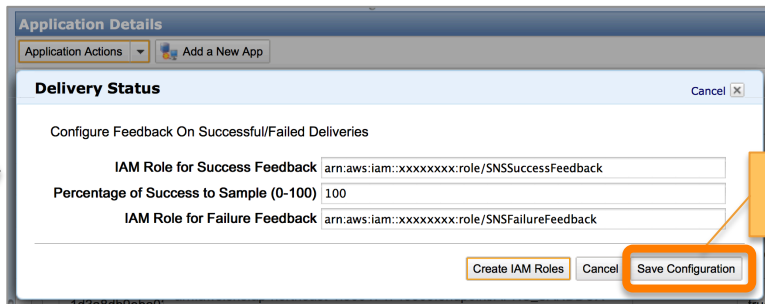
4. Event 発火時に
以下の様な通知を受信

```
{
  "EndpointArn": "arn:aws:sns:ap-northeast-1:xxxxxxxxxx:endpoint/APNS_SANDBOX/MobilePushTest/xxxx-xxxx-xxxx-xxxx",
  "EventType": "EndpointCreated",
  "Resource": "arn:aws:sns:ap-northeast-1:xxxxxxxxxx:app/APNS_SANDBOX/MobilePushTest",
  "Service": "SNS",
  "Time": "2015-03-11T03:45:22.545Z",
  "Type": "EndpointCreated"
}
```

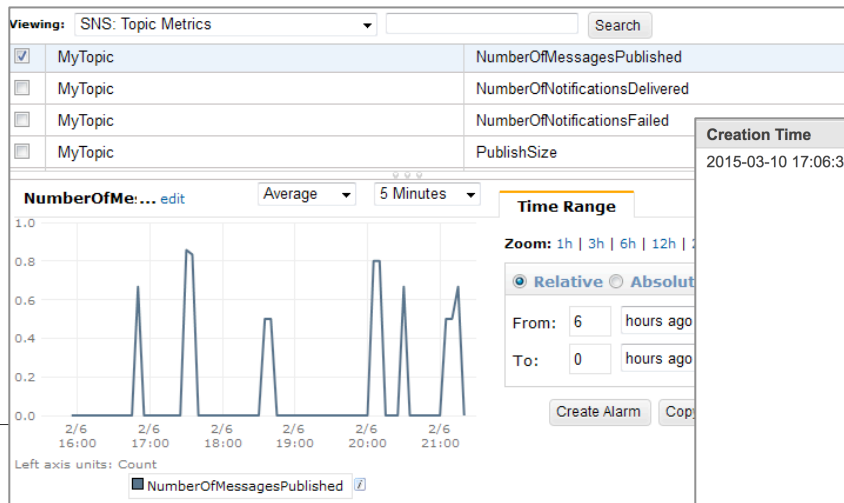

CloudWatch によるモニタリング



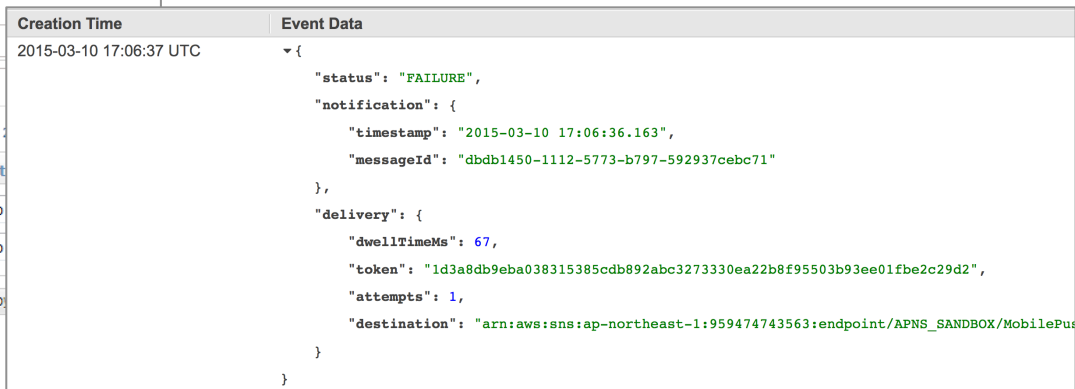
1. [Delivery Status]
をクリック



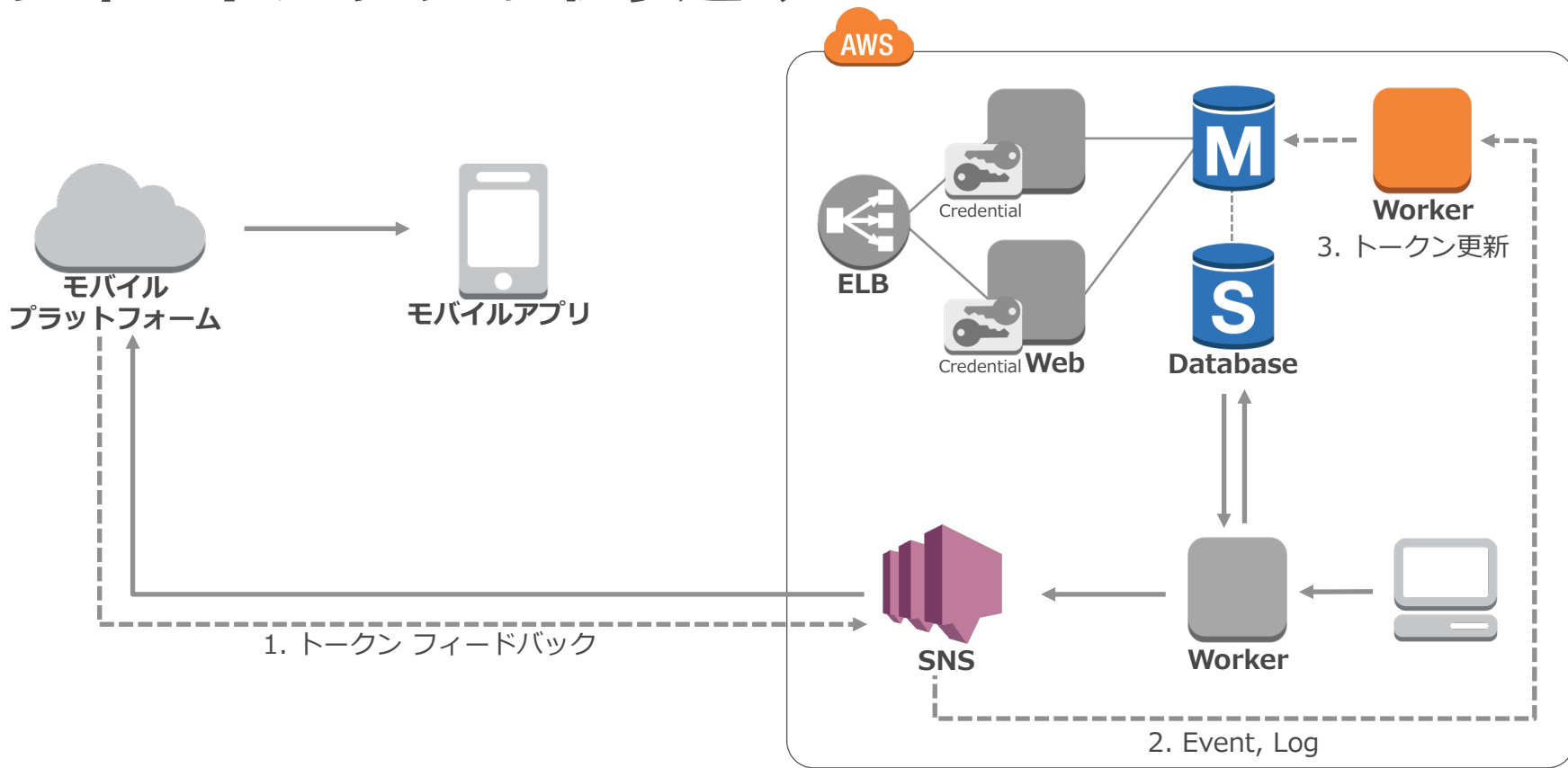
2. [Save Configuration]
をクリック



3. CloudWatch でログを
確認

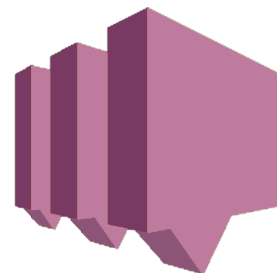


フィードバックの取り込み



Amazon SNS Mobile Push のまとめ

- クロスプラットフォームのモバイルプッシュ通知
(Apple, Google, Amazon, Baidu, Windows)
- ひと月 100 万リクエスト無料、以降は 100 万ごとに \$1.00
- けんろうせい 堅牢性と信頼性とスケーラビリティを兼備
- 簡単なトークン管理



Agenda

Agenda

- AWS モバイルサービスのご紹介
- Amazon SNS Mobile Push
 - モバイルプッシュ通知とは
 - 機能概要
 - SDK, IDE
 - 料金・無料使用枠
- Amazon SNS Mobile Push の使い方
 - マネージメントコンソールから
 - プログラムから
- **Q&A**

Q&A



Webinar資料の配置場所

- AWS クラウドサービス活用資料集

- <http://aws.amazon.com/jp/aws-jp-introduction/>

プロダクト別：				
Amazon S3		AWSマイスターシリーズ Re:Generate Amazon Simple Storage Service (S3)	Slideshare	PDF
Amazon Glacier		AWSマイスターシリーズ Reloaded Amazon Glacier Amazon Glacierのご紹介 機能編	Slideshare (Reloaded) Slideshare (機能編)	PDF (Reloaded) PDF (機能編)
Amazon Route 53		AWSマイスターシリーズ Re:Generate	Slideshare	PDF

公式Twitter/Facebook AWSの最新情報をお届けします



@awscloud_jp



検索



もしくは

<http://on.fb.me/1vR8yWm>

最新技術情報、イベント情報、お役立ち情報、お得なキャンペーン情報などを
日々更新しています！

ご参加ありがとうございました。