

JAWS DAYS 2016 - Move Up the Next Cloud Workshop #5

「S3 を限界まで使い倒せ！」



2016.03.12

Shinya TAKEBAYASHI

 @chi9rin

発表者について

- 竹林 信哉（たけばやし しんや）
- 普段の業務は Java のサポートや HTML5/JavaScript やネットワークパケット解析やコア解析や . . .
- AWS は触り始めて数年. 8 割趣味, 2 割仕事.

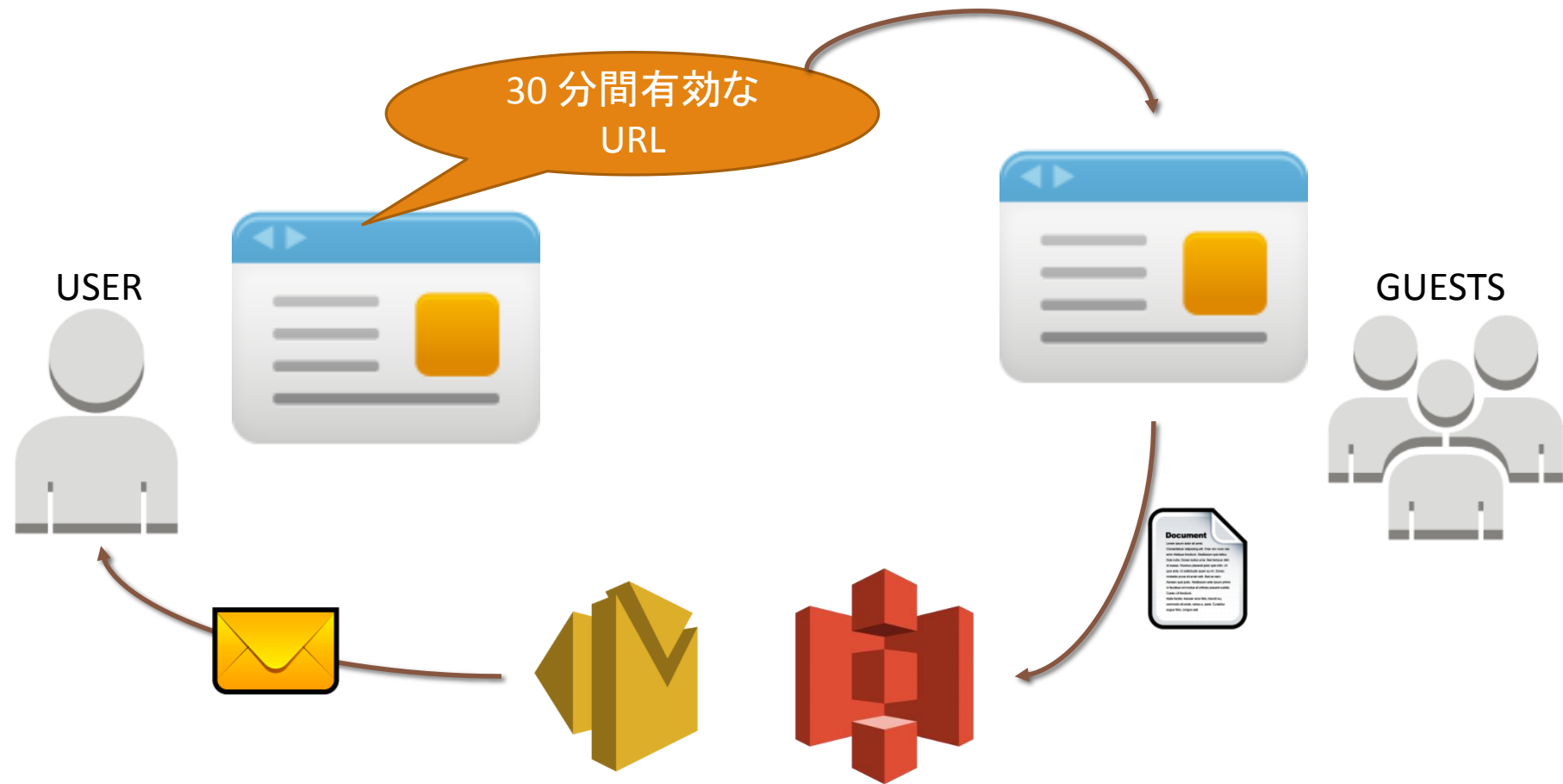
 @chi9rin

本ハンズオンの内容

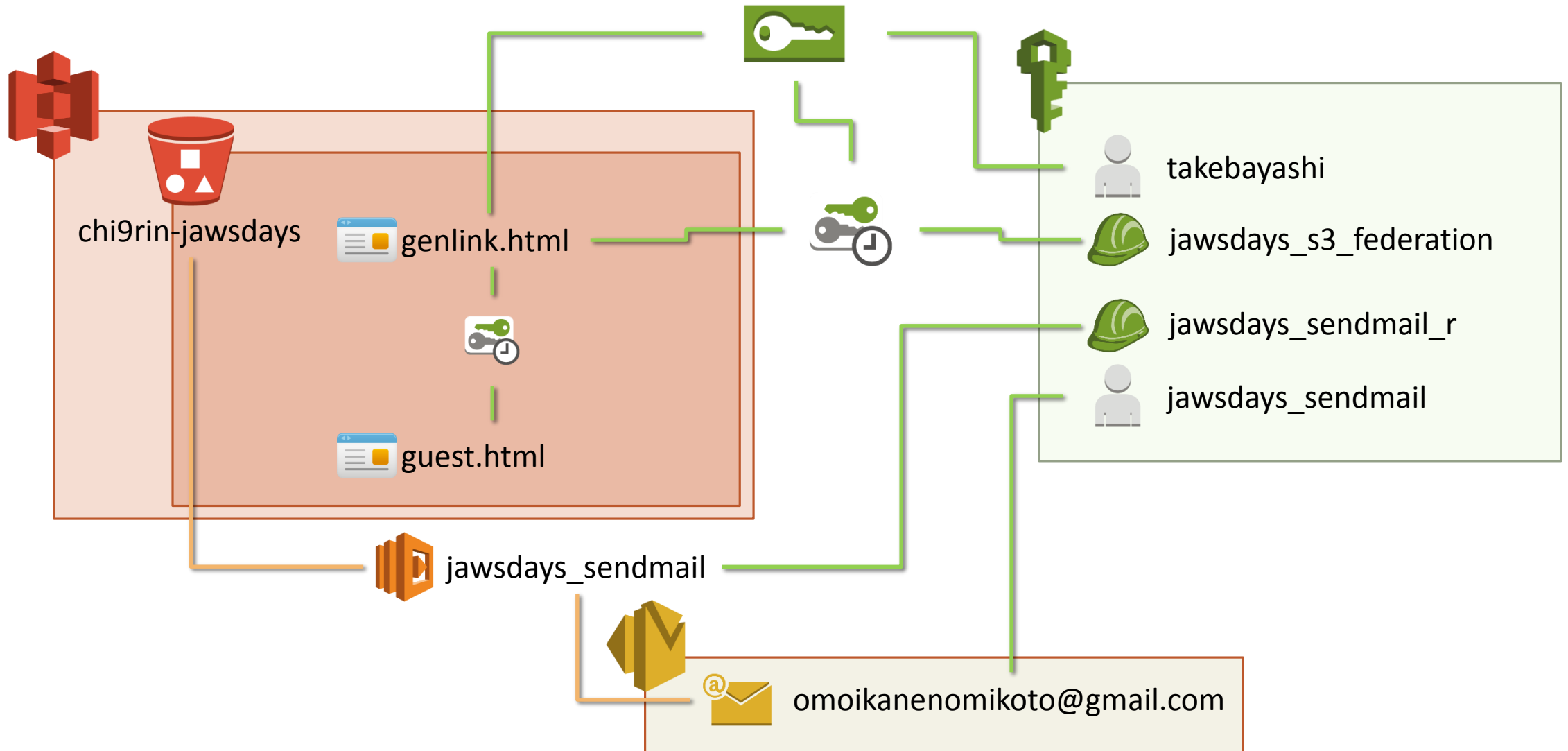
- ファイルアップローダを作ります.
- ユーザが, 30 分間有効なアップロード URL を生成します.
 - STS を用いて発行から 30 分の時間制限付きアクセス ID を作成します.
 - そのアカウント情報を含む URL をゲストに送付する, という想定で進めます.
- ゲストがその URL を開いて (一時トークンを使って) S3 にファイルをアップロードします.
- S3 にファイルが届くと, S3 バケットに設定した Lambda 関数により SES から「ファイルが届いたよ」というメールを送ります.
- 本ハンズオンの手順や資材は, 下記の場所で公開しています.
 - https://github.com/chi9rin/jawsdays2016_hands_on_chi9rin

本ハンズオンの内容

絵にしました



各サービスの連携



事前準備

ダウンロードしておいていただきたいもの

- https://github.com/chi9rin/jawsdays2016_hands_on_chi9rin/tree/master/materials

調べておいていただきたいもの

- AWS マネジメントコンソールへのログインに使用している IAM ユーザに関する下記の情報
 - ARN (チートシート [1])
 - アクセスキー ID (チートシート [2])
 - シークレットアクセスキー (チートシート [3])

メモ帳アプリケーション

- 上記 ZIP ファイルの中に「cheat_sheet.txt」があります.
- アカウント情報やハンズオン中に出てくる情報のメモにお役立てください.
- ←このアイコンのところでお使いください.



手順

- 【IAM】 SES で使用するユーザーの登録
- 【SES】 メールアドレスの登録
- 【Lambda】 Lambda 関数の作成
- 【SES】 Lambda 関数からのメール送信の許可設定
- 【S3】 バケットの作成, アクセス制御, Lambda 関数との関連づけ
- 【S3】 ユーザインタフェイス用の HTML/JavaScript の作成, アップロード
- 動作確認

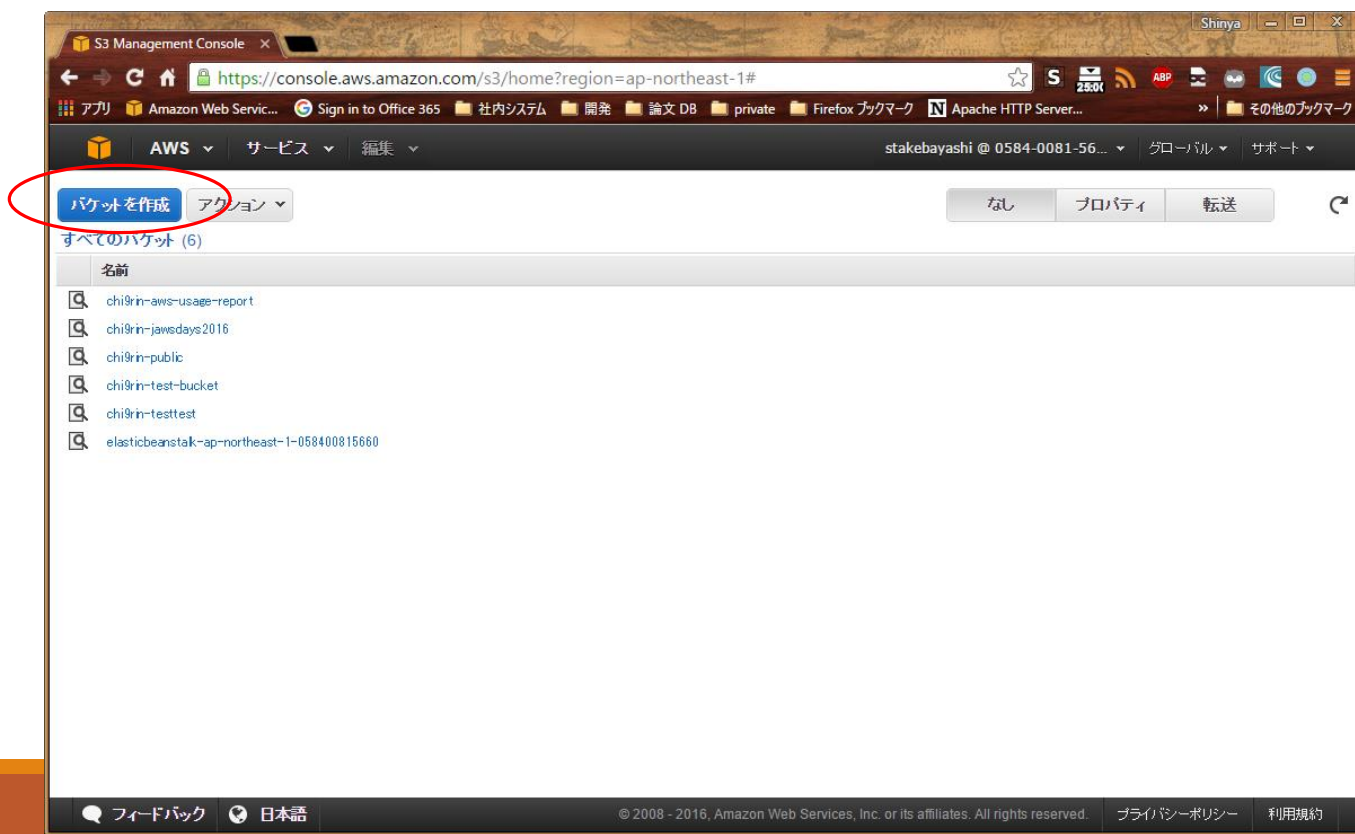
S3 バケットの作成

ファイルを保管するバケットを作成します。



- S3 の画面を開き「バケットを作成」を押下。
- バケット名: 任意 (例: chi9rin-jawsdays) (ARN は arn:aws:s3:::バケット名 です)
- [4][5] ◦ リージョン: オレゴン

※ Lambda 関数を作成するリージョンと合わせる必要があります。



バケットの作成 - バケット名とリージョンの選択

キャンセル

バケットとは、Amazon S3 に格納されるオブジェクトのコンテナです。バケットを作成する場合、レイテンシーを最適化し、コストを最小限に抑えて規制要件に対応できるリージョンを選ぶことが可能です。バケットの命名ルールに関する詳細については、『[Amazon S3 ドキュメント](#)』を参照してください。

バケット名

chi9rin-jawsdays

リージョン:

Oregon

ログ記録のセットアップ >

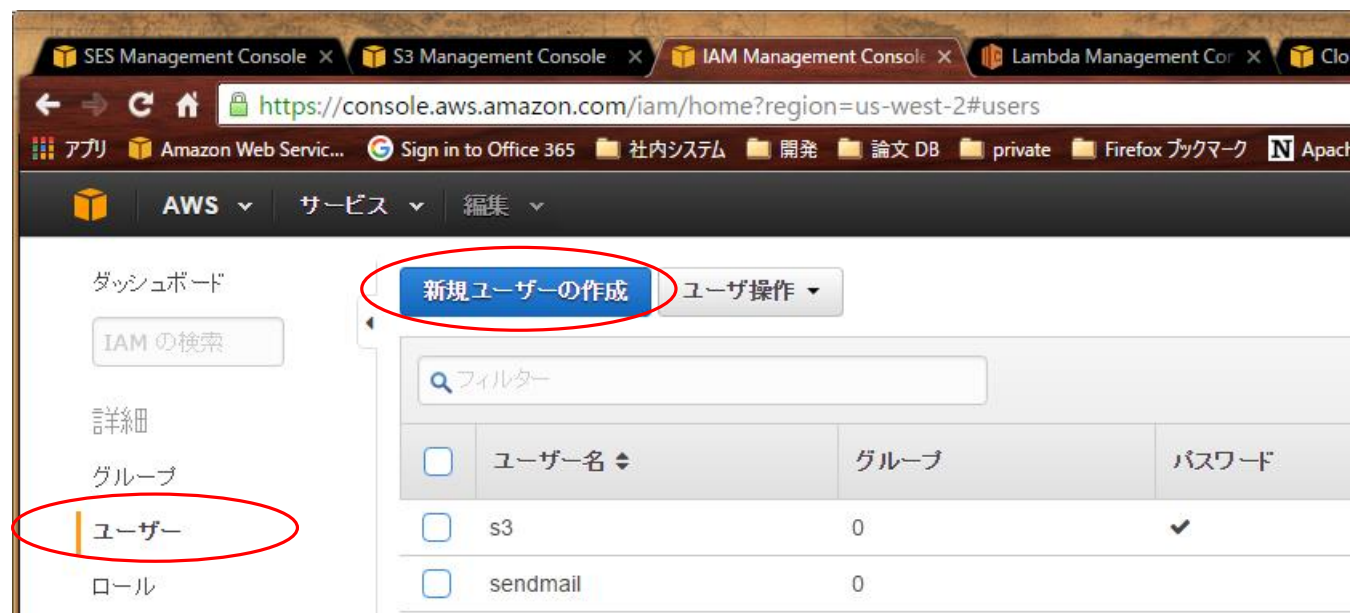
作成

キャンセル

IAM の設定 – SES 用アカウントの作成

IAM コンソールにて, SES にて E メールを送信するためのアカウントを作成します.

IAM コンソールの「ユーザー」を選択し, 「新規ユーザーの作成」を押下します.



IAM の設定 – SES 用アカウントの作成

ユーザ名: 任意 (例: jawsdays_sendmail)

- 「ユーザーごとにアクセスキーを生成」にチェックしてください。
- 「アクセスキー ID」と「シークレットアクセスキー」は後で使いますので、忘れずに写しておいてください。(特にシークレットアクセスキーはこの画面以降二度と参照できません)



Create User

ユーザー名の入力:

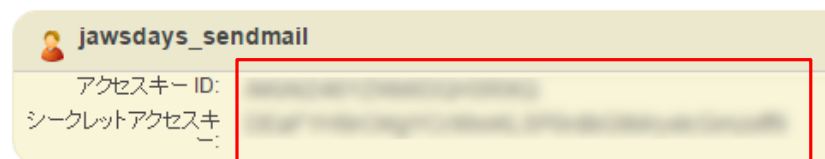
1. jawsdays_sendmail
- 2.
- 3.
- 4.
- 5.

最大でそれぞれ 64 文字

☒ ユーザーごとにアクセスキーを生成

☑ 1 ユーザーが正常に作成されました。
これは、これらのユーザーセキュリティ認証情報をダウンロードできる最後の機会です。
これらの認証情報はいつでも管理および再作成できます。

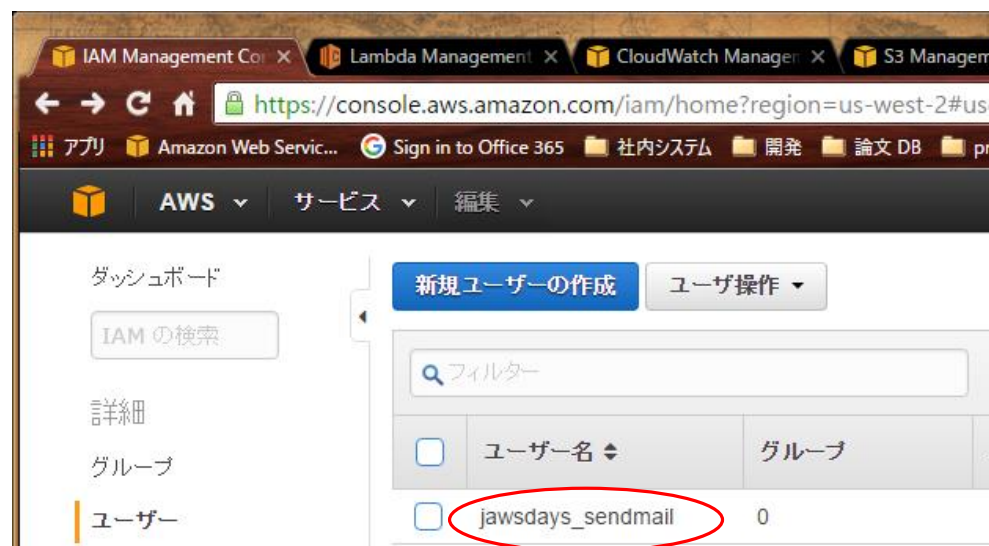
▼ ユーザーのセキュリティ認証情報を非表示



IAM の設定 – SES 用アカウントの作成

SES を用いてメールを送信できるように、ポリシーを割り当てます。

ユーザー一覧から作成したユーザを選択し、「アクセス許可」タブの「ポリシーのアタッチ」を押下します。



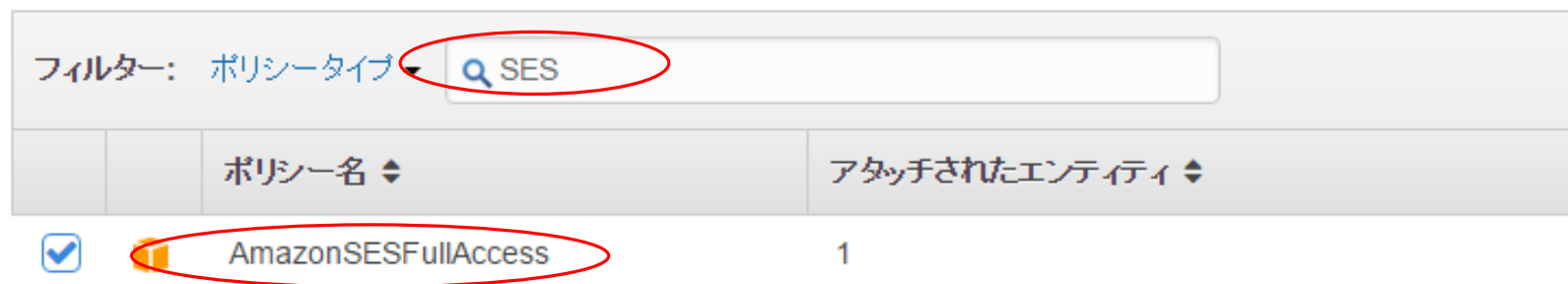
IAM の設定 – SES 用アカウントの作成

SES に対する FullAccess を許可します。

- フィルター欄に「SES」を入力するとリストの内容を絞り込めます。
- AmazonSESFullAccess にチェックを入れ、「ポリシーのアタッチ」を押下します。

ポリシーのアタッチ

アタッチするポリシーを 1 個以上選択してください。ユーザー は、それぞれ 10 個までのポリシーをアタッチできます。



フィルター: ポリシータイプ

	ポリシー名	アタッチされたエンティティ
<input checked="" type="checkbox"/>	AmazonSESFullAccess	1

IAM の設定 - STS 用 IAM ロールの作成

STS で払い出す一時アカウントに対する権限を設定するロールを作成します。
IAM コンソールで「ロール」→「新しいロールの作成」を押下してください。

- ロール名: 任意 (例: jawsdays-s3-federation)
- ロールタイプ: AWS サービスロール – Amazon EC2
- ポリシーのアタッチ: 未選択で「次のステップ」
- 確認画面: 「ロールの作成」

※ この後、手動でポリシーを設定するため「ロールタイプ」や「ポリシーのアタッチ」では
適当なものを選んでいただいて大丈夫です。

IAM の設定 - STS 用 IAM ロールの作成



作成した一時アカウント用の IAM ロールを設定していきます。

ここでロールの ARN をメモし、ローラー一覧から作成したロールを選択します。

ダッシュボード

IAM の検索

詳細

グループ

ユーザー

ロール

新しいロールの作成

ロールのアクション ▼

フィルター

ロール名 ◆

☒ jawdays_s3_federation

☐ jawdays_sendmail_r

▼ 概要

ロール ARN arn:aws:iam::**[redacted]**:role/jawdays_s3_
インスタンスプロフィールの ARN arn:aws:iam::**[redacted]**:instance-profile/ja
パス /
作成時刻 2016-02-27 01:42 UTC+0900

アクセス許可

信頼関係

アクセスアドバイザー

管理ポリシー

この ロール にアタッチされている管理ポリシーはありません。

ポリシーのアタッチ

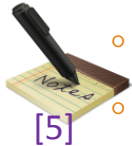
インラインポリシー

表示するインラインポリシーはありません。作成するには、[ここをクリックしてください](#)。

IAM の設定 - STS 用 IAM ロールの作成

Policy Generator の「選択」を押下し、下記の通り設定していきます。

- 効果: 許可
- AWS サービス: Amazon S3
- アクション: *
- Amazon リソースネーム: arn:aws:s3:::<バケット名>/, arn:aws:s3:::<バケット名>/*



[5]

IAM の設定 - STS 用 IAM ロールの作成

STS を用いて権限を払い出すためのアカウントを設定します。

ここで設定したアカウントをもとにして、一時的なアクセス権を入手します。



「信頼関係の編集」にある ARN（下図ぼかし部分）は、ご自身の IAM ユーザの ARN を設定してください。

▼ 概要

ロール ARN	arn:aws:iam::[redacted]:role/jawsdays
インスタンスプロファイルの ARN	arn:aws:iam::[redacted]:instance-profi
パス	/
作成時刻	2016-02-27 01:42 UTC+0900

アクセス許可 信頼関係 アクセサドバイザー

ロールと、ロールのアクセス条件を引き受けることができる信頼されたエンティティを表示でき

信頼関係の編集



信頼関係の編集

以下のアクセスコントロールポリシードキュメントを編集して、信頼関係をカスタマイズできます。

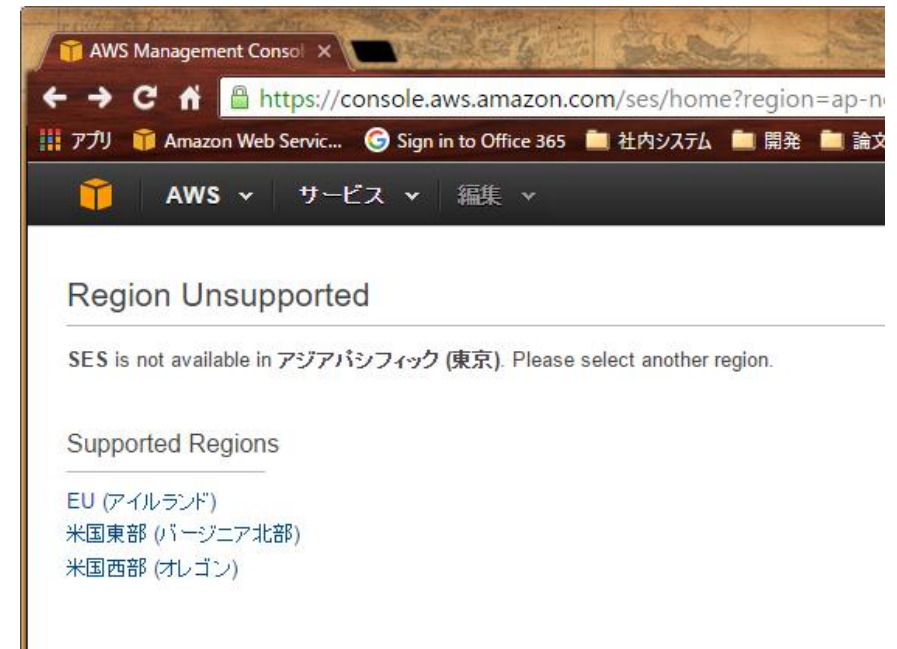
ポリシードキュメント

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Principal": {
7         "Service": "ec2.amazonaws.com"
8       },
9       "Action": "sts:AssumeRole"
10    },
11    {
12      "Effect": "Allow",
13      "Principal": {
14        "AWS": "arn:aws:iam::[redacted]:user/[redacted]"
15      },
16      "Action": "sts:AssumeRole"
17    }
18  ]
19 }
```


SES の設定

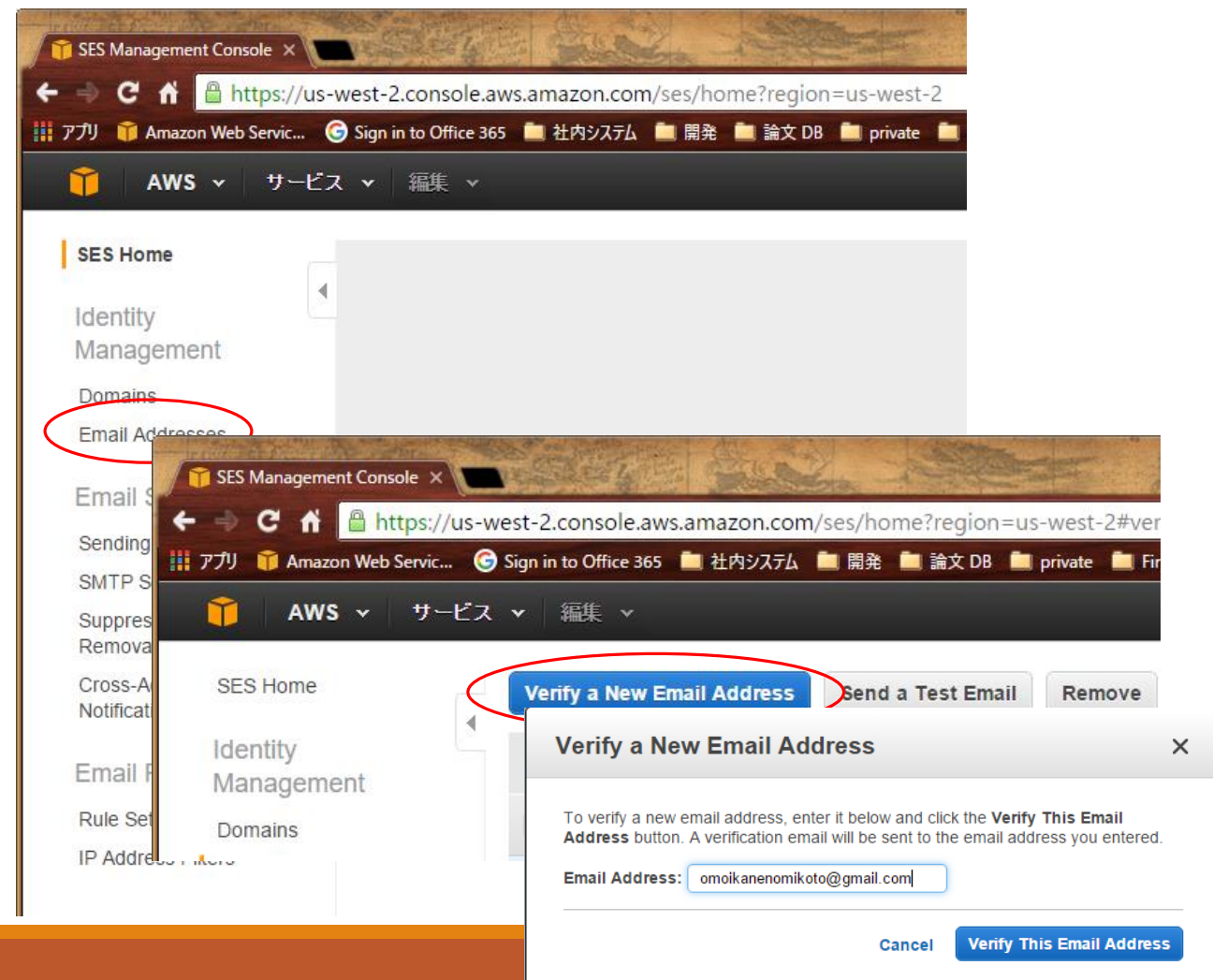
SES でメールを送信するための、メールアドレスのアカウントを設定します。

- ここで登録したメールアドレスを送信元として、ファイルが届いた時にメールを送信します。
- SES の画面を開いた際に右の画面が表示された場合は、米国西部（オレゴン）を選択してください。



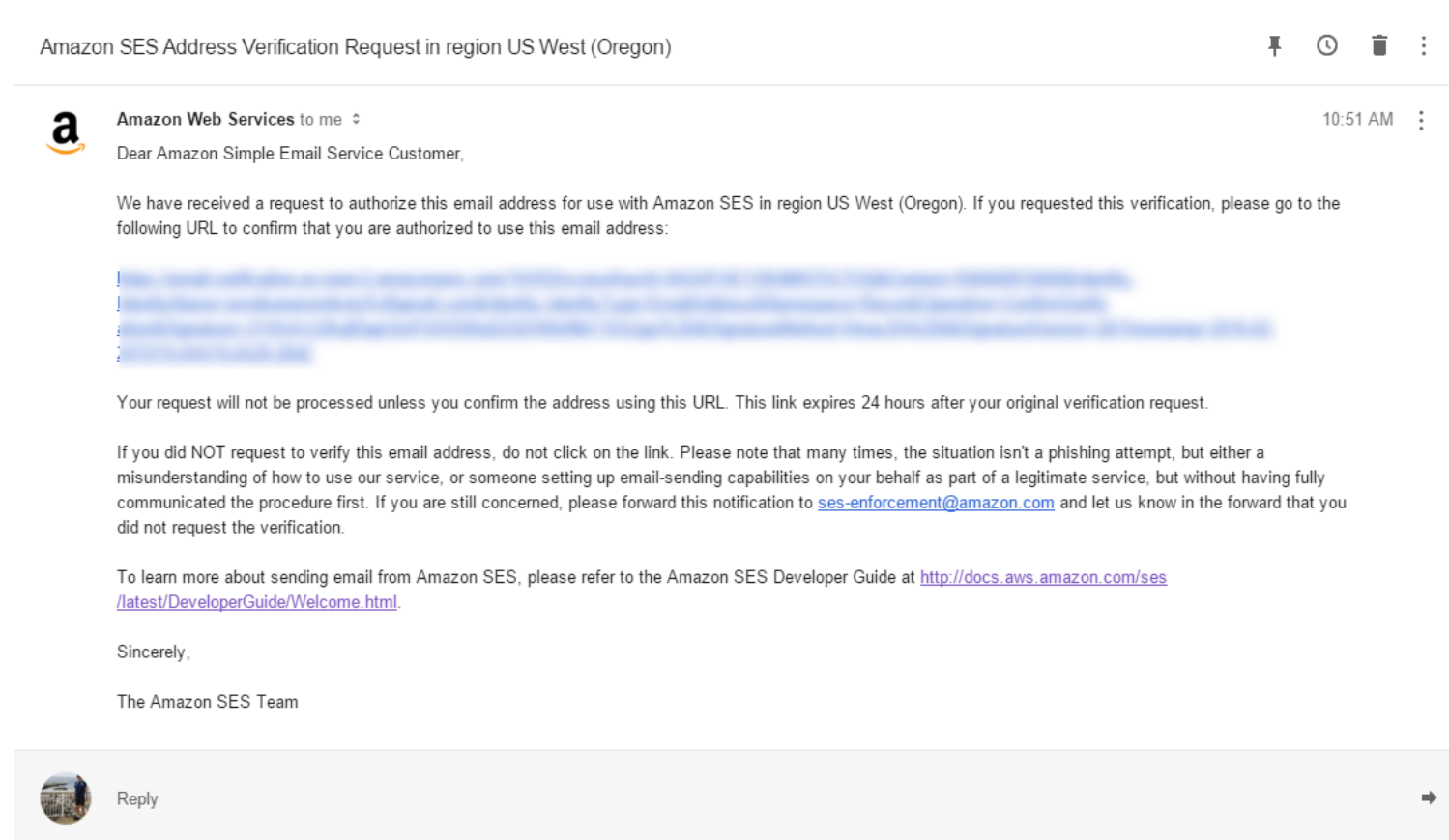
SES の設定 - メールアドレスの登録

- ① SES の Email Addresses を開きます
- ② Verify a New Email Address を開きます
- ③ 登録するメールアドレスを入力します



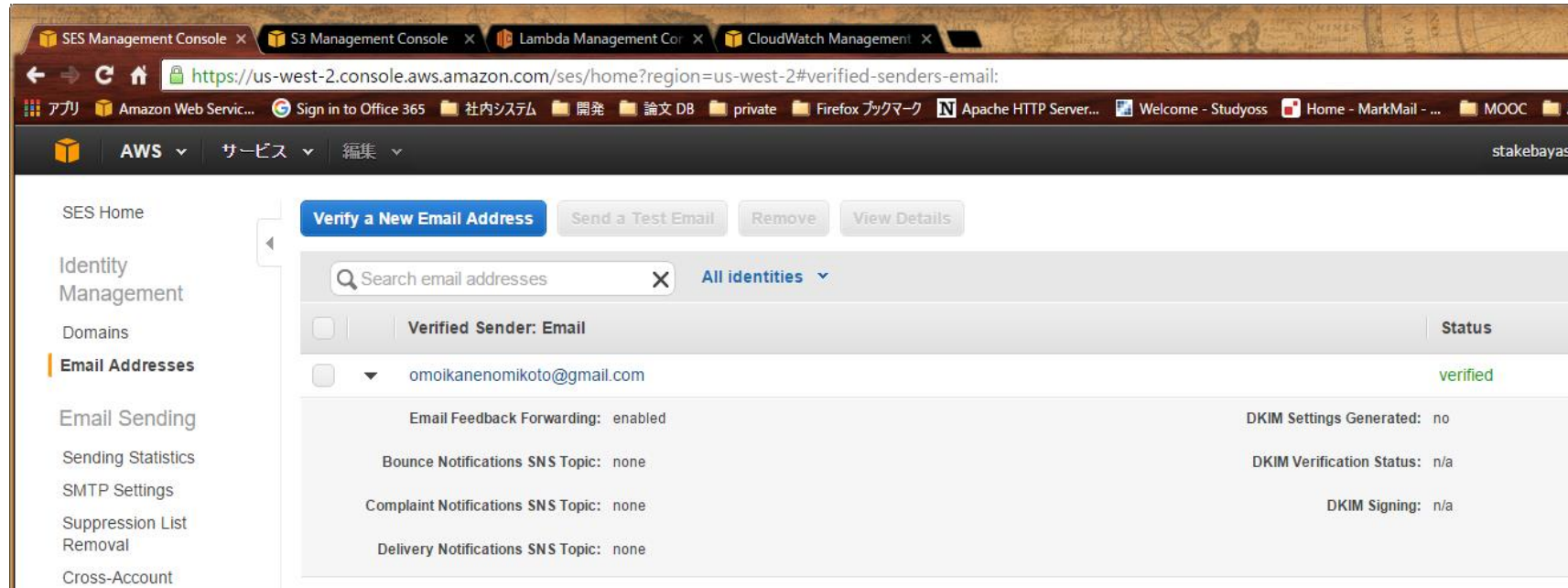
SES の設定 - メールアドレスの登録

下記のメールが届いたら、リンクを開いてメールアドレスの認証を完了します。



SES の設定 - メールアドレスの登録

SES のページリロードして, Status が **verified** になっていれば OK です.



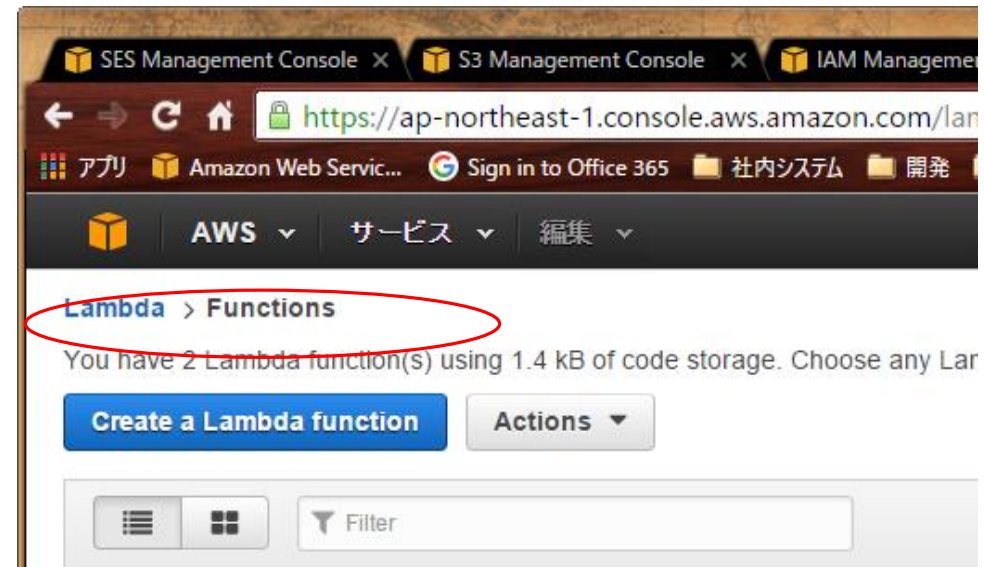
Lambda の設定

S3 のバケットにファイルがアップロードされた際に実行される Lambda 関数を作成します。

S3 のバケットと同じリージョンで作成する必要があります。

Create a Lambda function を押下し新しく Lambda 関数を作ります。

ボタン押下後， blueprint を選ぶ画面が表示されますが，スキップします。



Lambda の設定 – 初期設定



設定項目が多いため、スクリーンショットは省きます。

- Name: 任意 (例: jawsdays_sendmail)
- Description: 任意
- Runtime: Node.js
- Lambda function code: memo/lambda.txt



SES オブジェクトを作成する時に使う accessKeyId および secretAccesskey は、メール送信用ユーザを作成したときのものを指定してください。

また、送信用メールアドレスも、SES の設定で入力したメールアドレスを設定してください。

- Handler: index.handler
- Role: S3 execution role → [次ページへ](#)
- Memory(MB): 128
- Timeout: 30 sec
- VPC: No VPC

Lambda の設定 – 許容する操作の設定

Role の欄でプルダウンボックスから「S3 execution role」を選択すると、次の画面に移ります。ここで S3 の情報を収集することと、CloudWatch にログを出力することを許容されたロールを簡単に作成することができます。

- IAM ロール: 新しい IAM ロールの作成
- ロール名: 任意（例：jawsdays_sendmail_r）

▼ 詳細を非表示

ロールの概要 ?

ロールの説明 Lambda execution role permissions

IAM ロール

新しい IAM ロールの作成 ▼

ロール名

jawsdays_sendmail_r

▶ ポリシードキュメントを表示

Lambda の設定 – ロール ARN の確認



IAM コンソールから、作成したロールの ARN を確認し、メモします。

ダッシュボード

IAM の検索

詳細

グループ

ユーザー

ロール

IAM > ロール > jawsdays_sendmail_r

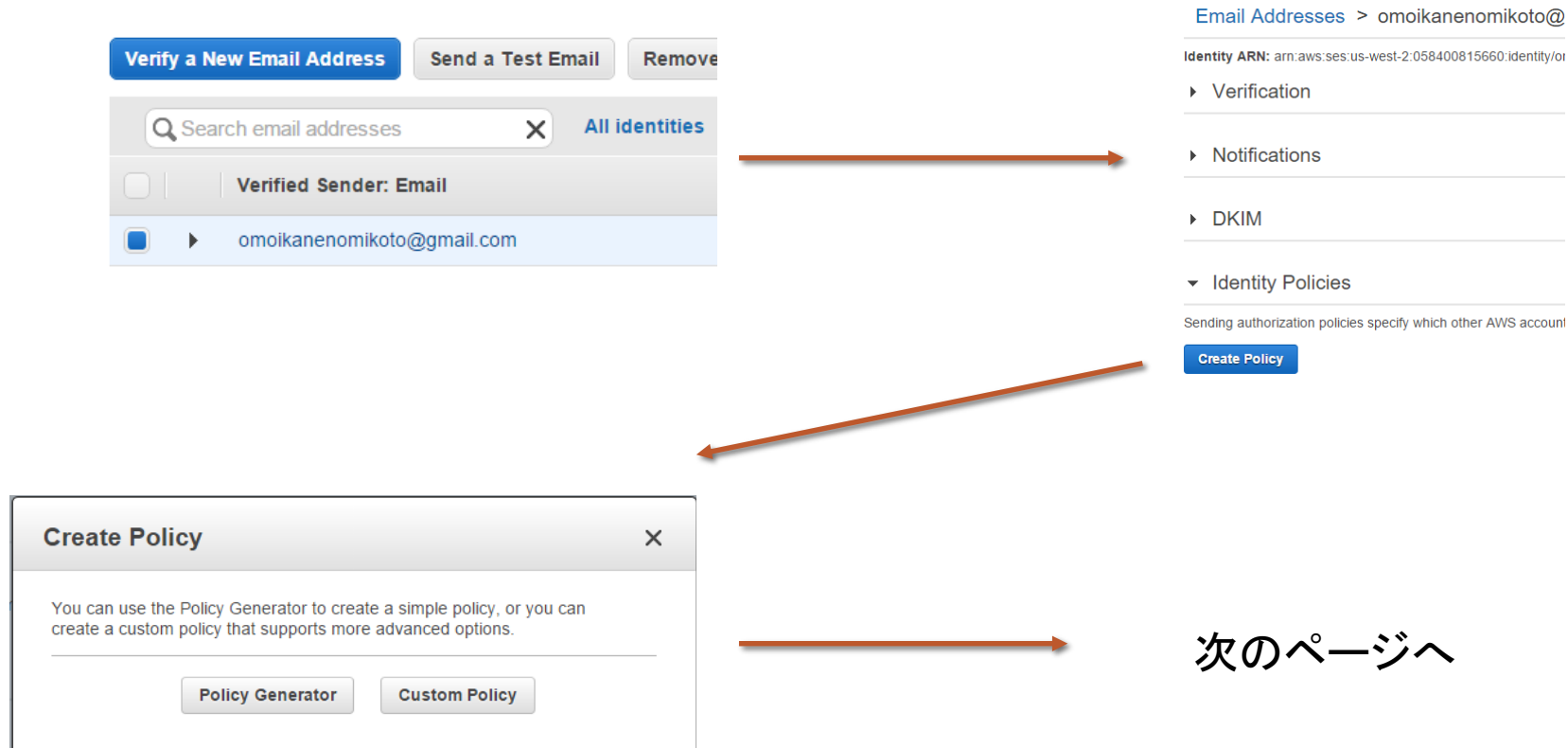
▼ 概要

ロール ARN	arn:aws:iam::058400815660:role/jawsdays_sendmail_r
インスタンスプロファイルの ARN	
パス	/
作成時刻	2016-02-28 14:03 UTC+0900

Lambda の設定 – SES の Lambda 関数から呼び出し許可

Lambda 実行ロールが作成できたら、SES がこのロールからの呼び出しを許可するように設定する必要があります。

SES のコンソールから先ほど追加した E メールアカウントを開き、ポリシーを設定します。



次のページへ

Lambda の設定 – SES の Lambda 関数から呼び出し許可



Principal に Lambda 関数実行用ロールの ARN を入力し, SendEmail と SendRawEmail の両方を許可します.

Policy Generator

With this tool, you can create a basic sending authorization policy by generating simple conditions in your policy, you can edit the policy later.

Identity omoikanenomikoto@gmail.com ⓘ

Effect ☒ Allow ☐ Deny ⓘ

Principals* ⓘ

Actions* ☒ ses:SendEmail ⓘ
☒ ses:SendRawEmail

Fields marked with asterisk (*) are required.

[Add Conditions \(optional\)](#)

[Add Statement](#)

S3 バケットの設定

作成した S3 のバケットを開き下記の設定を施します.

- アクセス権の設定
- Lambda 関数の呼び出し設定
- HTML ファイルの配置

S3 バケットの設定 - アクセス権の設定

外部の JavaScript から S3 にアクセスできるように、CORS の設定を変更します。



S3 バケットの設定 - アクセス権の設定

デフォルトでは GET のみ許可されていますが、PUT（ファイル作成）を追加します。

CORS 構成エディター

キャンセル

バケットの CORS 設定: "chi9rin-jawsdays"

CORS(Cross-Origin Resource Sharing)を使用して、他のドメインで実行しているウェブアプリケーションに、Amazon S3 バケットのコンテンツへのアクセスを選択的に許可することができます。各 CORS ルールには、そのオリジンに対して許可するオリジン/ドメインと HTTP メソッドのセットを含める必要があります。オプションで、ユーザーがリクエストに設定したり、レスポンスでアクセスしたりできるヘッダーと、飛行前のレスポンスをキャッシュする必要がある期間を指定することもできます。

下のテキストエリアでこのバケットの既存の CORS 設定を編集します。

```
<?xml version="1.0" encoding="UTF-8"?>
<CORSConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <CORSRule>
    <AllowedOrigin>*</AllowedOrigin>
    <AllowedMethod>GET</AllowedMethod>
    <AllowedMethod>PUT</AllowedMethod>
    <AllowedHeader>*</AllowedHeader>
  </CORSRule>
</CORSConfiguration>
```

CORS 設定例

保存 削除 閉じる

S3 バケットの設定 – Lambda 関数呼び出しの設定

「イベント」タブに下記を設定します。

- 名前: 任意（未記入でも OK）
- イベント: Put
- プレフィックス: 空欄
- サフィックス: 空欄
- 送信先: Lambda 関数
- Lambda 関数: 作成した Lambda 関数
（例: jawsdays_sendmail）

※ 一覧に表示されない場合は、
リージョンを再確認してください。



▶ ログ記録

▼ イベント

イベント通知によって、アラートの送信やワークフローのトリガーを実行できます。通知は、[Amazon Simple Service\(SNS\)](#)や [Amazon Simple Queue Service\(SQS\)](#)経由で送信したり、[Lambda 関数宛](#)に送信ができます（バケットの場所によって決まります）。

名前	<input type="text" value="例: MyEmailNotificationsForPut"/>	
イベント	<input type="text" value="Put"/>	
プレフィックス	<input type="text" value="e.g. images/"/>	
サフィックス	<input type="text" value="e.g. jpg"/>	
送信先	<input type="radio"/> SNS トピック <input type="radio"/> SQS キュー <input checked="" type="radio"/> Lambda 関数	
Lambda 関数の	<input type="text" value="jawsdays_sendmail"/>	

S3 バケットの設定 - HTML ファイルの編集と転送

リンク作成用のページ, ファイルアップロード用のページを編集します.



- genlink.html - リンク作成用ページ
 - 12 行目付近のユーザのアクセスキー ID およびシークレットアクセスキー
 - 20 行目付近の STS ロール ARN



- guest.html – ファイルアップロード用ページ
 - 30 行目付近のバケット名

JavaScript ライブラリを S3 のバケットに転送します.

転送後, 各 HTML ファイルと JavaScript ファイルのアクセス権を「全員」に対して「開く／ダウンロード」が可能になるように設定します.

S3 バケットの設定 - HTML ファイルの編集と転送

正しく設定されると
鍵アイコンではなくなる

「全員」が「開く／ダウンロード」
できるようにする

名前	ストレージクラス	サイズ	最終更新日時
genlink.html	スタンダード	2.6 KB	Sun Feb 28 12:04:14 GMT+900 2016
guest.html	スタンダード	1.5 KB	Sun Feb 28 12:05:16 GMT+900 2016
js	--	--	--

オブジェクト

バケット: chi9rin-jawsdays
名前: genlink.html
リンク: <https://s3-us-west-2.amazonaws.com/chi9rin-jawsdays/genlink.html>
サイズ: 2669
最終更新日時: Sun Feb 28 12:04:14 GMT+900 2016
所有者: omoikanenomikoto
ETag: ed84112b5ed68f7152726bcea272920
有効期限: なし
有効期限ルール: 該当なし

詳細

アクセス許可

アクセスポリシーを使い、バケットとそのコンテンツへのアクセスを制御できます。詳細はこちら。

被与者: omoikanenomikoto	<input checked="" type="checkbox"/> 開く/ダウンロード	<input checked="" type="checkbox"/> アクセス許可の表示	<input checked="" type="checkbox"/> アクセス許可の編集	×
被与者: 全員	<input checked="" type="checkbox"/> 開く/ダウンロード	<input type="checkbox"/> アクセス許可の表示	<input type="checkbox"/> アクセス許可の編集	×

さらにアクセス許可を追加する

保存 キャンセル

メタデータ

動作確認

genlink.html を開き、適当なユーザ名いれて「URL 発行」ボタンを押すと、30 分間有効なアップロード用ページが開きます。



動作確認

リンクを開いたらファイルを選択してアップロード開始ボタン。

「アップロード完了」のメッセージが表示されることを確認します。

S3 バケットにもファイルが入っていることを確認してみましょう。



本日やったこと

S3 を時間制限つきファイルアップローダとして使う方法を紹介しました.

- STS を使って一時的なアクセスキーを取得する
- S3 から Lambda 関数で SES を使って E メールを送信する