Government Updates via SMS

Deployment Guide
Alexandre Levret (levret@amazon.com)

March 2020



Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

© 2020 Amazon Web Services, Inc. or its affiliates. All rights reserved.

Contents

[Introduction 1](#_Toc36722576)

[Deployment Overview 2](#_Toc36722577)

[Deployment – CovidUpdate.yml 3](#_Toc36722578)

[IAM Roles, Permissions and Bucket Policies Created 4](#_Toc36722579)

# Introduction

This document describes the necessary steps to deploy a solution which creates a notification system to send you a SMS on your phone number with the latest COVID-19 updates in your country. This solution will use API Gateway to get update notification from the end user and it will trigger a Lambda function to publish the update message to the SNS topic. Then the SNS will send these update notifications to all its subscribers.



# Deployment Overview

The deployment process utilizes CloudFormation to define infrastructure-as-code. This solution consists of one CloudFormation templates:

* **covidUpdate.yml** – Creates the S3 bucket, Lambda functions. IAM roles, CloudWatch events, SNS and Pinpoint.

**Important**: this solution only works in the following regions:

* US East
	+ us-east-1 (N. Virginia)
	+ us-west-2 (Oregon)
* GovCloud
	+ us-gov-west-1 (US-West)
* Europe
	+ eu-central-1 (Frankfurt)
	+ eu-west-1 (Ireland)
* Asia Pacific
	+ ap-northeast-1 (Tokyo)
	+ ap-south-1 (Mumbai)
	+ ap-southeast-1 (Singapore)
	+ ap-southeast-2 (Sydney)

# Deployment – CovidUpdate.yml

1. Navigate to the CloudFormation console.
2. Create a new stack using the ‘CovidUpdate.yml’ file.
3. Enter a stack name. Example: ‘Notification solution”.
4. Enter values for the parameters:
	* **Website’s URL:** Website URL to check for Covid-19 updates
	* **Name of the country:** Depending on the country where the solution will be deployed, it will change the language of the text message the citizen will receive. (Supports US – English and Fr - French)
	* **Lambda function name to Register citizen’s phone number:** Name of Lambda function used to register citizen phone number as subscriber into the SNS topic.
	* **Lambda function name to check for website updates:** Name of Lambda function used to check if there is a difference between old page (object stored in S3) and current page every 5 minutes (CW Events is behind) - if so, publish to SNS topic that there is an update.
	* **Lambda function name to publish to SNS topic:** Name of Lambda function triggered by an API (Government employee) to publish to SNS topic that there is an update.



# IAM Roles, Permissions and Bucket Policies Created

1. Below are roles created by this solution
	1. rRegisterCellPhoneSNSRole

 Type: "AWS::IAM::Role"

 DeletionPolicy: Delete

 Properties:

 RoleName: !Sub "rRegisterCellPhoneSNSRole-${AWS::AccountId}-${AWS::Region}"

 AssumeRolePolicyDocument:

 Version: "2012-10-17"

 Statement:

 - Effect: "Allow"

 Principal:

 Service:

 - lambda.amazonaws.com

 Action:

 - sts:AssumeRole

* 1. rCovidLambdaRole

 Type: "AWS::IAM::Role"

 DeletionPolicy: Delete

 Properties:

 RoleName: !Sub "rCovidLambdaRole-${AWS::AccountId}-${AWS::Region}"

 AssumeRolePolicyDocument:

 Version: 2012-10-17

 Statement:

 - Effect: "Allow"

 Principal:

 Service:

 - lambda.amazonaws.com

 Action:

 - "sts:AssumeRole"

 Path: /service-role/

 Policies:

 -

 PolicyName: "ReadAndWriteToBucket"

 PolicyDocument:

 Version: "2012-10-17"

 Statement:

 -

 Effect: "Allow"

 Action: "s3:\*"

 Resource: [

 !Sub "arn:aws:s3:::covidupdatebucket-${AWS::Region}-${AWS::AccountId}",

 !Sub "arn:aws:s3:::covidupdatebucket-${AWS::Region}-${AWS::AccountId}/\*"

 ]

 -

 Effect: "Allow"

 Action: "logs:CreateLogGroup"

 Resource: !Sub "arn:aws:logs:${AWS::Region}:${AWS::AccountId}:\*"

 -

 Effect: "Allow"

 Action: [

 "logs:CreateLogStream",

 "logs:PutLogEvents"

 ]

 Resource: [

 !Sub "arn:aws:logs:${AWS::Region}:${AWS::AccountId}:log-group:/aws/lambda/CovidUpdateSNS:\*"

 ]

 ManagedPolicyArns:

* 1. rUpdatedWebsiteLambdaRole

 Type: "AWS::IAM::Role"

 DeletionPolicy: Delete

 Properties:

 RoleName: !Sub "rUpdatedWebsiteLambdaRole-${AWS::AccountId}-${AWS::Region}"

 AssumeRolePolicyDocument:

 Version: 2012-10-17

 Statement:

 - Effect: "Allow"

 Principal:

 Service:

 - lambda.amazonaws.com

 Action:

 - "sts:AssumeRole"

 Path: /service-role/

 Policies:

 -

 PolicyName: "LogsAndSns"

 PolicyDocument:

 Version: "2012-10-17"

 Statement:

 -

 Effect: "Allow"

 Action: "logs:CreateLogGroup"

 Resource: !Sub "arn:aws:logs:${AWS::Region}:${AWS::AccountId}:\*"

 -

 Effect: "Allow"

 Action: [

 "logs:CreateLogStream",

 "logs:PutLogEvents"

 ]

 Resource: [

 !Sub "arn:aws:logs:${AWS::Region}:${AWS::AccountId}:log-group:/aws/lambda/rUpdatedWebsiteLambda:\*"

 ]