

AWS Health Event 통합하고 실시간 채널로 주요 이벤트 확인 하기



© 2024

Dark Mode



BespinGlobal

심선보

2024.07.25

구현의 배경



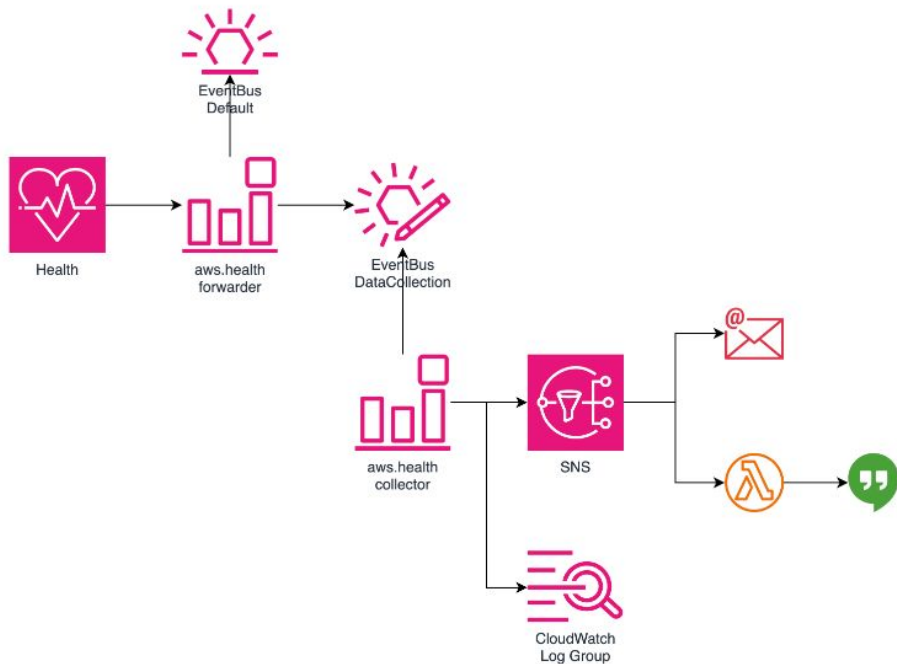
회사에서 운영하는 Cloud 규모가 커지면서 다양한 아키텍처, 너무나 많은 워크로드를 운영 관리해야 합니다.

리소스는 언제나 상태가 변하기 마련입니다.

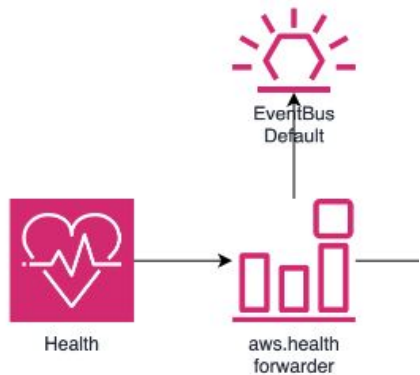
AWS 메인テナンス, 개발팀의 애플리케이션 변화, 뜻하지 않은 장애 등 다양한 상황에 놓이게 됩니다.

AWS Health 이벤트를 통합하고 AWS Cloud 에서 서비스 Health 관련 실시간 알림을 받음으로써 이 문제를 어느 정도 완화할 수 있습니다.

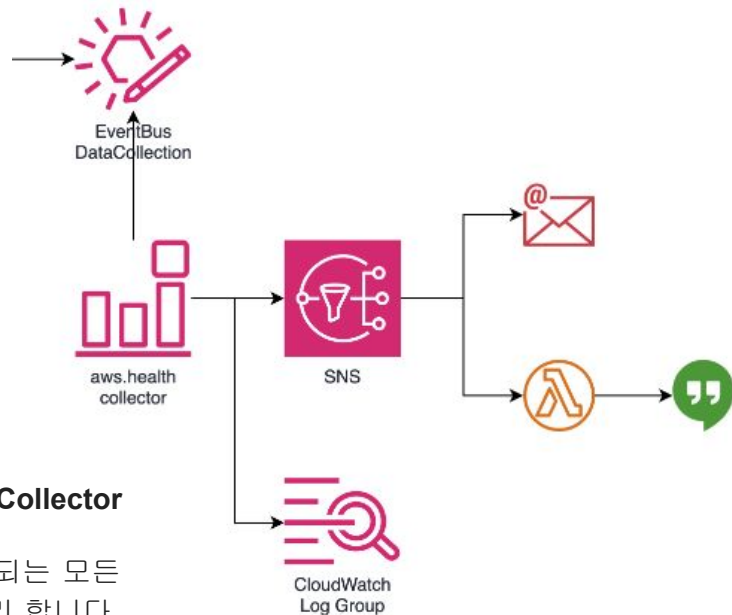
아키텍처



- **Data Collector:** Organizations 에 가입된 AWS 계정을 대상으로 Health 이벤트를 수집합니다.
- **Lambda Consumer:** Data Collector Event Bus 로부터 전달받은 Health 이벤트를 Google Hangout 채널로 실시간 전송 합니다.
- **Event Forwarder:** Organizations 에 가입된 AWS 계정이 Health 이벤트를 전송합니다.

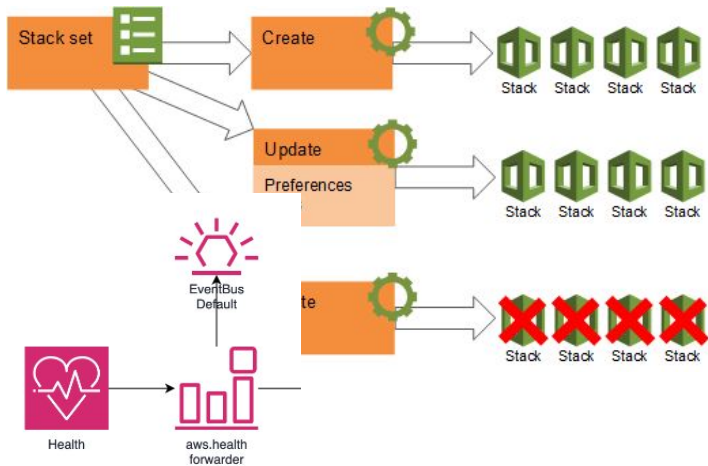


Event Forwarder
AWS Health 이벤트를
전송 합니다.



Data Collector
EventHub 로 유입되는 모든
이벤트를 처리 합니다.

스택 구현 전략



한 곳으로 보내고,

여러 AWS 계정에 일관되게
프로비저닝 및 운영이 가능한 건?

CloudFormation StackSet

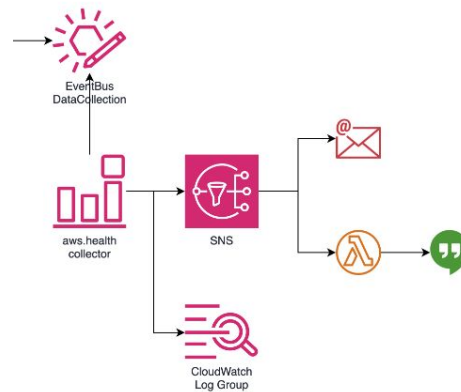
서비스간 유기적으로 잘
통합되고,
배포 및 운영을 쉽게 하는건?

CloudFormation template

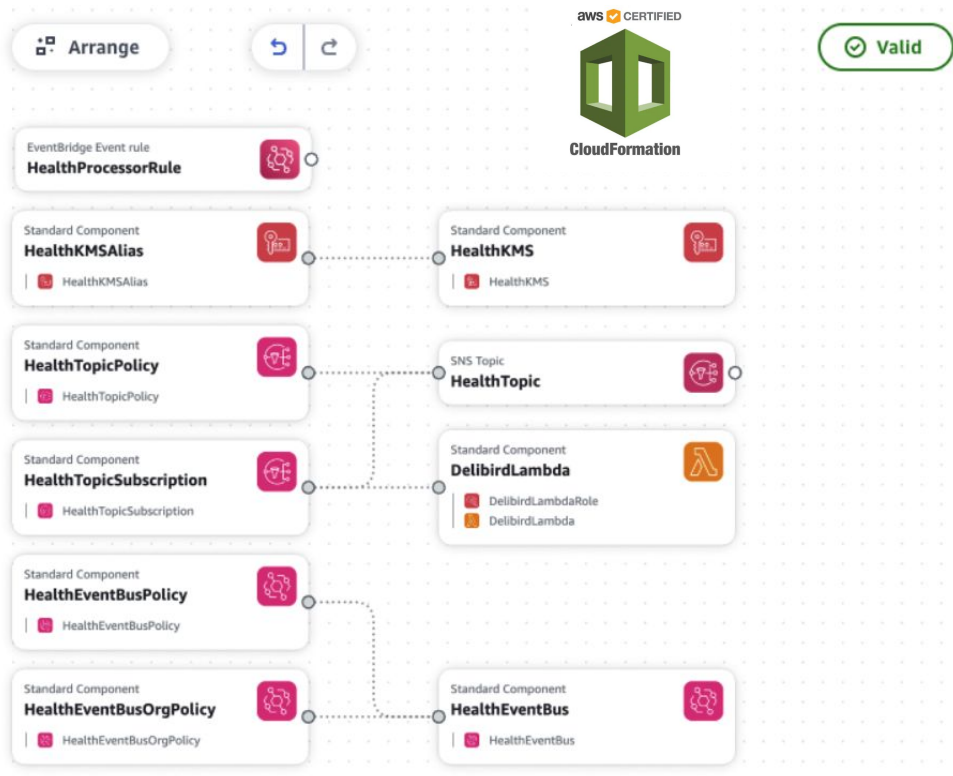
aws CERTIFIED



CloudFormation



Data Collector 컴포넌트



아래 고려 요소들로 여러번의 트러블 슈팅을 할 수 밖에 없습니다.

- SaaS 서비스간 통합이 다소 존재
- 서비스간 제한된 액세스 정책 적용
- KMS 를 통한 보안 강화

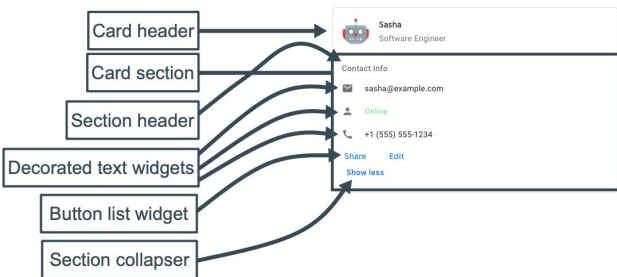
최초 완성 모델이 나오기까지 오류가 나더라도 롤백-비활성화 옵션을 통해 진행하는걸 추천 합니다.

하지만, 커스텀파라미터 추가, 리소스 추가가 빈번하게 이루어진다면 역시 고단한 트러블 슈팅이 일어납니다.

서비스 내장형 IAM 정책을 놓치지 말기

- KMS
 - events.amazonaws.com
- SNS
 - events.amazonaws.com

aws-health-delibird - Lambda 구현



실험적으로 람다에서 아직 적용하지 않은 Kotlin 을 해보자.

- Java-21 Corretto
- ARM 프로세서
- Kotlin 최신버전
- AWS 최신 라이브러리 채택
- Arrow 를 통한 함수형 프로그래밍을 제대로 (Functor, Monad, Applicative 등 다양한 함수형 프로그래밍 기법을 적용 가능)
- Maven 으로 빌드하고 하나의 Jar 로 통합 (Why? 없는것 같음)
- JSON 을 SQL 처럼 쿼리해 올 수 없을까?
- Google Card-V2 메시지 모델 장난아니게 복잡한데 <보기좋은 메시지 레이아웃으로 주요 정보 확인>
- Lambda 를 Image 타입으로 서비스 하자

Json-SQL 잠깐 소개

```
[
  {
    "id": "668ff44842e4428cc95c5929",
    "index": 1,
    "guid": "db73d088-d98c-4a28-aef1-a3d83874ff2b",
    "isActive": false,
    "balance": "$2,080.82",
    "age": 25,
    "eyeColor": "blue",
    "name": "Bartlett Monroe",
    "gender": "male",
    "company": "LIQUICOM",
    "email": "bartlettmonroe@liquicom.com",
    "phone": "+1 (980) 487-3574",
    "address": "105 Wyckoff Street, Kipp, FederatedS",
    "registered": "2022-05-19T01:25:14 -09:00",
    "others": {
      "code": "101",
      "genius": 10,
      "level": 2
    }
  },
  {
    "id": "668ff4482965ffc5acb09c08",
    "index": 2,
    "guid": "dc9bb694-3a10-4005-9d91-c89bdca03ea3",
    "isActive": true,
    "balance": "$3,746.41",
    "age": 20,
    "eyeColor": "green",
    "name": "Manuela Olson",
    "gender": "female",
```

```
val sql = """
select id, guid, isActive, balance, age,
       eyeColor, name, gender, company,
       email, phone, address, registered

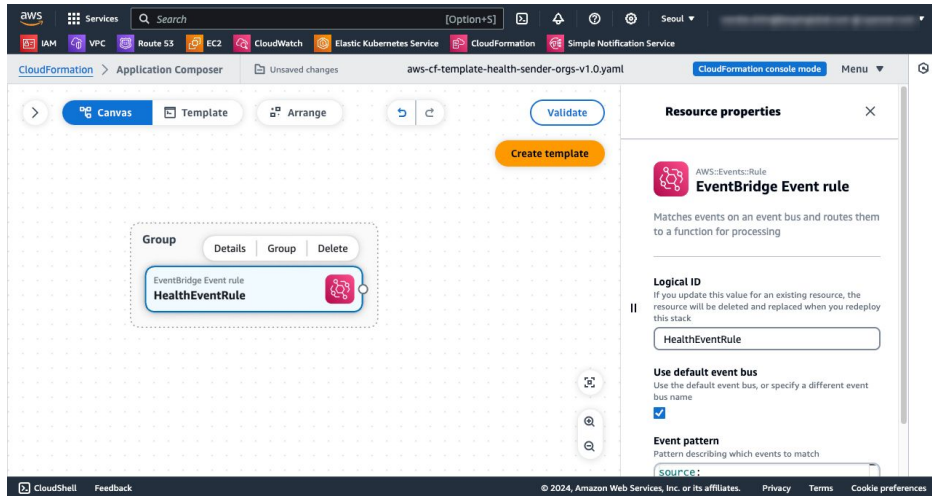
from member

where gender = :gender
and age <= :age
and eyeColor = :eyeColor
"""
```

```
val list = sqlSession.queryForList(sql, mapOf(
    "gender" to "female",
    "age" to "30",
    "eyeColor" to "blue"))
```

```
val data = list.firstOrNull()
val age = data.getInt("age")
val name = data.getString("name")
val registered = data.getDate("registered")
```


Event Forwarder 컴포넌트



Event Bus 에서 aws.health 소스를 대상으로 Collector 에 전달만 하면 됩니다.

너무 빈약해서 할 말이 없네요.

CloudFormation StackSet을 통해 자알 프로비저닝 되기를

배포 순서

3 단계로 전체 스택을 구성할 수 있습니다. CloudFormation을 통해 One-Step 자동화 하려면 **Lambda** 이미지를 사전에 **ECR** 저장소에 **Push** 해 두어야 합니다.

1. **SNS Subscriber** 역할로 실시간 통보를 담당하는 **aws-health-delibird** 컨테이너 이미지로 ECR 저장소에 업로드 합니다.
2. **Data Collector** 스택을 **aws-cf-template-health-collector-v1.0.yaml** 템플릿 으로 배포합니다.
3. **Event Forwarder** 스택을 **aws-cf-template-health-sender-orgs-v1.0.yaml** Stack-Sets 으로 배포합니다.

배포 - 1 health-delibird-lambda ECR Push

`aws-health-ecr.sh` 셸 파일을 이용하여, `symplesims/aws-health-delibird:1.0.0` 도커 이미지를 로컬에 내려받고 ECR 저장소를 생성하고 업로드를 합니다.

```
#!/bin/bash

REGION="ap-northeast-2"
ECR_NAME="cops-health-delibird-lambda-ecr"
ECR_TAG="1.0.0"
KMS_ALIAS_NAME="aws/ecr"
LAMBDA_IMAGE="symplesims/aws-health-delibird:1.0.0"

create_ecr() {}

upload_image() {}

echo "1. ECR 저장소 생성"
create_ecr

echo "2. ECR 업로드 실행"
upload_image
```

REGION

ECR 이미지를 배포할 리전 입니다.

ECR_NAME

ECR 저장소 이름입니다.

ECR_TAG

ECR 이미지 태그 입니다.

KMS_ALIAS_NAME

AWS 관리형 KMS 키를 사용하세요.

LAMBDA_IMAGE

docker hub 에 공개된 람다 이미지입니다.

배포 - 2 Data Collector 스택 배포

aws Services [Option+S] Seoul

IAM VPC Route 53 EC2 CloudWatch Elastic Kubernetes Service CloudFormation Simple Notification Service

Step 1
Create stack

Step 2
Specify stack details

Step 3
Configure stack options

Step 4
Review and create

Create stack

Prerequisite - Prepare template

Prepare template
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

- Choose an existing template**
Upload or choose an existing template.
- Use a sample template**
Choose from our sample template library.
- Build from Application Composer**
Create a template using a visual builder.

Specify template Info

A template is a JSON or YAML file that describes your stack's resources and properties.

Template source
Selecting a template generates an Amazon S3 URL where it will be stored.

- Amazon S3 URL**
Provide an Amazon S3 URL to your template.
- Upload a template file**
Upload your template directly to the console.
- Sync from Git - new**
Sync a template from your Git repository.

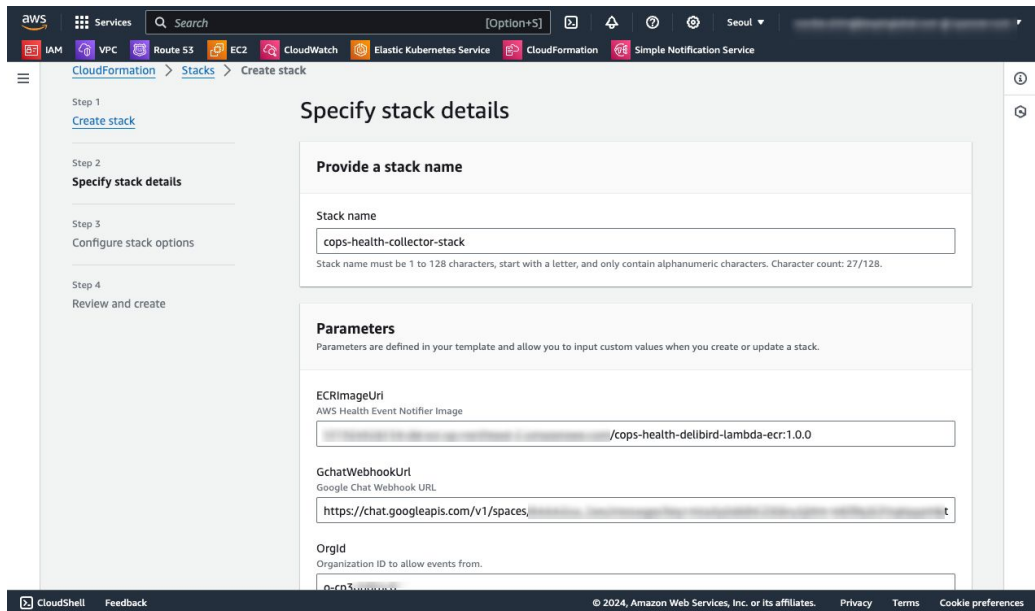
Amazon S3 URL

https://[redacted].ap-northeast-2.amazonaws.com/aws-cf-template-health-collector-v1.0.yaml

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

CloudFormation > Stacks

배포 - 2 Data Collector 스택 배포



Project

프로젝트 코드입니다. 리소스 네임 및 태깅 속성을 통한 일관성을 유지하기 위한 코드입니다.

Region

스택이 배포될 리전 입니다.

ECRImageUri

111122223333.dkr.ecr.<your-region>.amazonaws.com/cops-health-delibird-lambda-ecr:1.0.0

GchatWebhookUrl

메시지를 전송 할 채널로 Google Hangout WebHook 주소입니다.

OrgId

Organizations 조직 아이디 입니다. OrgId 에서오는 모든 AWS Health 이벤트를 수신받기 위함입니다.

배포 - 2 Data Collector 스택 배포

The screenshot shows the AWS CloudFormation console interface. The top navigation bar includes the AWS logo, 'Services', a search bar, and the user's profile 'seonbo.shim@bespinglobal.com @ opsnow-com'. The main content area is divided into a left sidebar and a main panel. The sidebar shows the 'Stacks (2)' section for the 'cops-health-collector-stack', with a filter for 'Active' and 'View nested' options. The main panel displays the 'Events (22)' for this stack, with a search bar and a 'Detect root cause' button. The events table lists the following:

Timestamp	Logical ID	Status	Detailed status
2024-07-25 00:02:28 UTC+0900	DelibirdLambda	CREATE_COMPLETE	-
2024-07-25 00:02:59 UTC+0900	DelibirdLambda	CREATE_IN_PROGRESS	-
2024-07-25 00:02:58 UTC+0900	DelibirdLambda	CREATE_IN_PROGRESS	-
2024-07-25 00:02:51 UTC+0900	DelibirdLambdaRole	CREATE_COMPLETE	-
2024-07-25 00:02:48 UTC+0900	HealthKMS	CREATE_COMPLETE	-
2024-07-25 00:02:35 UTC+0900	HealthEventBusOrgPolicy	CREATE_COMPLETE	-
2024-07-25 00:02:35 UTC+0900	HealthEventBusPolicy	CREATE_COMPLETE	-
2024-07-25 00:02:35 UTC+0900	HealthEventBusOrgPolicy	CREATE_IN_PROGRESS	-

The bottom of the console shows the footer with 'CloudShell', 'Feedback', and copyright information: '© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences'.

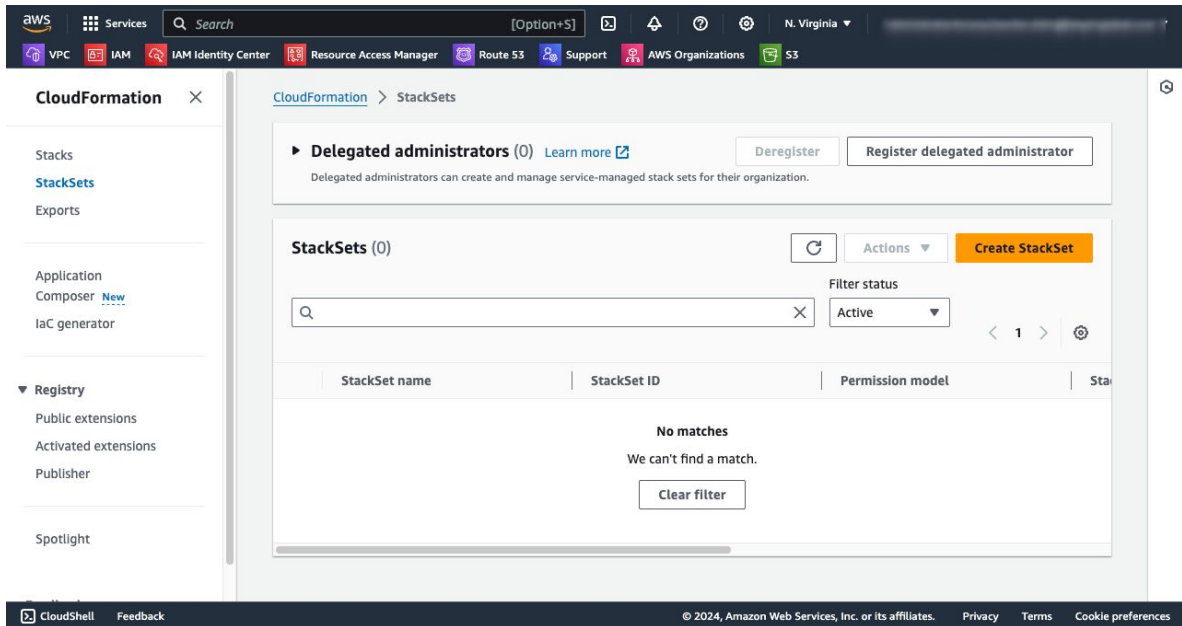
최종 리뷰를 마쳤다면
프로비저닝 Checked
하고 Submit 합니다.

캡처된 이미지와 같이
깔끔하게 성공하기까지

!느낌엔 수십번 트러블
슈팅한 것 같습니다.

리소스 타입에 없는 속성,
리소스 참조 오류 등

배포 - 3 Event Forwarder 스택 배포



CloudFormation > StackSets - Create StackSet 으로 진행됩니다.

조직 마스터 계정에서 프로비저닝 해야 합니다.

StackSet는 리전을 선택하여 프로비저닝 하게 되므로 **가급적 us-east-1 버지니아 리전**에서 관리하는걸 추천 합니다.

배포 - 3 Event Forwarder 스택 배포

The screenshot displays the AWS CloudFormation console during the deployment of a StackSet. The left sidebar shows the progress through five steps: Step 2 (Specify StackSet details), Step 3 (Configure StackSet options), Step 4 (Set deployment options), Step 5 (Review), and a final Review step. The main content area is divided into three sections:

- Permissions:** A section titled "Permissions" with the instruction "Choose an IAM role to explicitly define how CloudFormation will manage your target accounts." Two options are available: "Service-managed permissions" (selected) and "Self-service permissions".
- Prerequisite - Prepare template:** A section titled "Prerequisite - Prepare template" with the instruction "Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack." Two options are available: "Template is ready" (selected) and "Use a sample template".
- Specify template:** A section titled "Specify template" with the instruction "A template is a JSON or YAML file that describes your stack's resources and properties." Under "Template source", two options are available: "Amazon S3 URL" (selected) and "Upload a template file". Below this, the "Amazon S3 URL" field contains the text "https://[redacted].amazonaws.com/aws-cf-template-health-sender-orgs-v1.0.yaml".

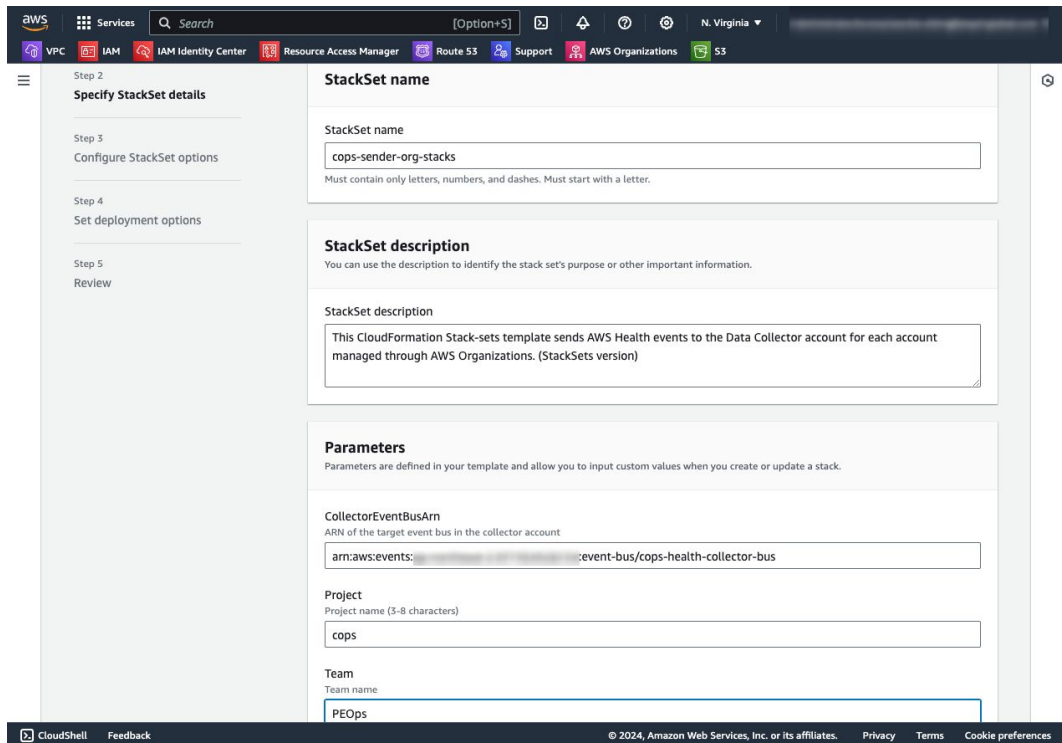
CloudFormation > StackSets

Service-Managed 퍼미션을 선택하세요

- AWS Organizations 과의 통합
- 프로비저닝 관련하여 IAM 권한

문제가 해소됩니다.

배포 - 3 Event Forwarder 스택 배포



StackSet 주요 사용자 정보를 기입합니다 .

Project

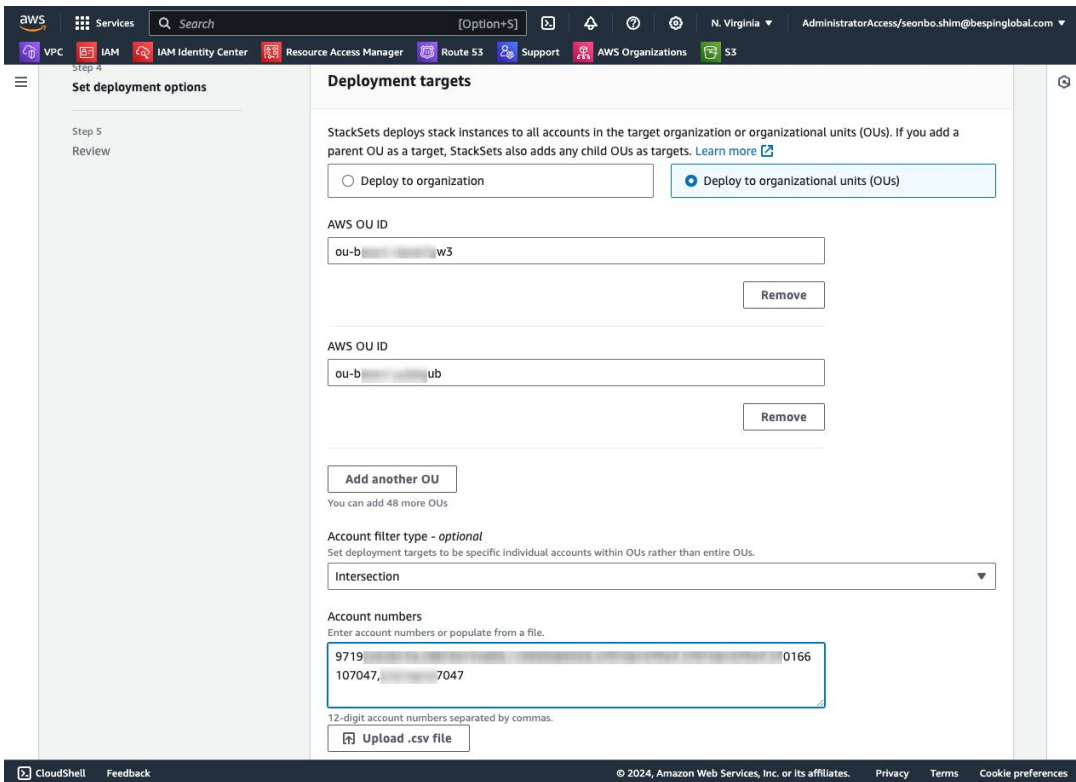
프로젝트 코드입니다. 리소스 네임 및 태깅 속성을 통한 일관성을 유지하기 위한 코드입니다.

CollectorEventBusArn

앞서 프로비저닝한 Data Collector의 Event Bus ARN 을 입력 합니다.

Ex) arn:aws:events:::event-bus/cops-health-collector-bus

배포 - 3 Event Forwarder 스택 배포



Member AWS 계정을 타게팅 합니다.

ORG 전체 vs 선택적 OU

전체 ACCOUNT vs 선택적 ACCOUNT

리전

배포 - 3 Event Forwarder 스택 배포

The screenshot displays the AWS IAM console interface for configuring a StackSet. The top navigation bar shows the AWS logo, 'Services', a search bar, and the current region 'N. Virginia'. Below the navigation bar, the 'Regions' section shows a search bar and a list of regions with 'ap-northeast-2' selected. The 'Deployment options' section contains a table with the following settings:

Maximum concurrent accounts	Failure tolerance
1	0
Region concurrency	Concurrency mode
SEQUENTIAL	STRICT_FAILURE_TOLERANCE

The 'Capabilities' section features a blue information box with the following text:

The following resource(s) require capabilities: [AWS::IAM::Role]
This template contains Identity and Access Management (IAM) resources. Check that you want to create each of these resources and that they have the minimum required permissions. In addition, they have custom names. Check that the custom names are unique within your AWS account. [Learn more](#)

I acknowledge that AWS CloudFormation might create IAM resources with custom names.

최종 리뷰 및 Submit

전략적으로 같은 리전을 대상으로 AWS 멤버 계정을 타게팅 하는 것이 훨씬 유리합니다

스택의 이름은 ue1-<project> 형식으로 배포하는 것이 좋습니다.

StackSet-ue1-<project>-<uuid> 의 네이밍으로 배포 됩니다.

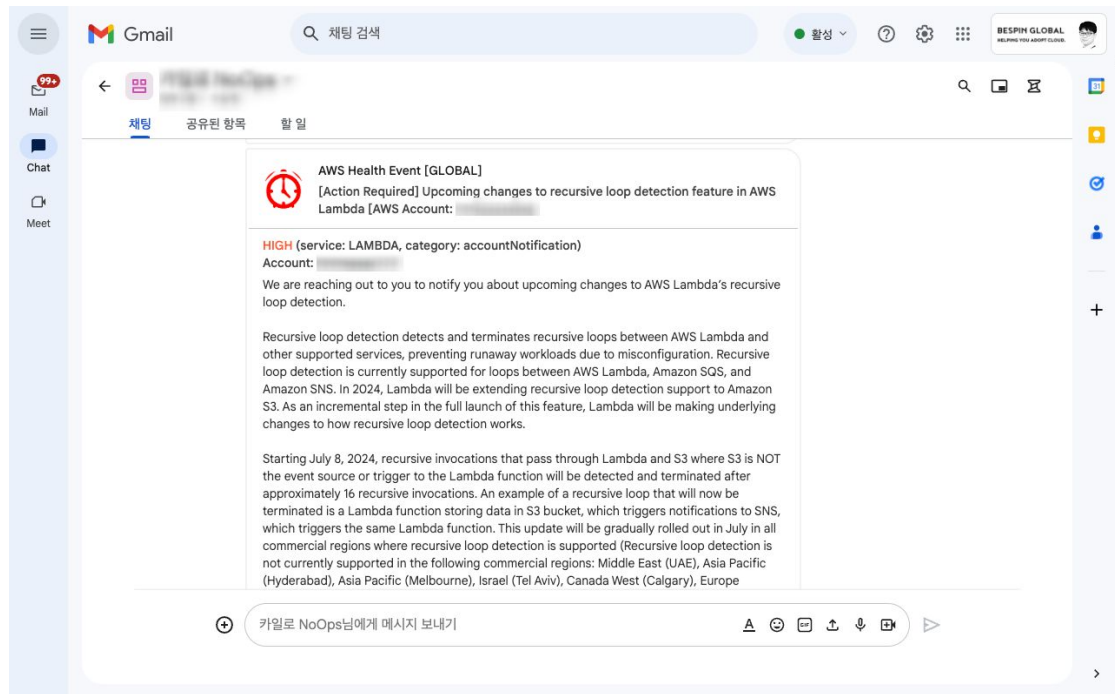
배포 - 3 Event Forwarder 스택 배포

The screenshot shows the AWS CloudFormation console for a StackSet named 'cops-sender-org-stacks'. The 'Stack instances' tab is selected, displaying a table of 5 instances, all with a 'SUCCEEDED' status. The table columns are AWS account, AWS region, Stack ID, and Detailed status.

AWS account	AWS region	Stack ID	Detailed status
1293 [redacted]	ap-northeast-2	arn:aws:cloudformation:ap-northeast-2:1293-74ef-401b-9e7f-478f6668bf49/06f517e0-45[redacted]	✓ SUCCEEDED
3701 [redacted]	ap-northeast-2	arn:aws:cloudformation:ap-northeast-2:3701-81a32511-756c-4f18-8351-be0a76f92d88/1[redacted]	✓ SUCCEEDED
5578 [redacted]	ap-northeast-2	arn:aws:cloudformation:ap-northeast-2:5578-f93-48c3-9529-967b37f667b3/3ce40230-4[redacted]	✓ SUCCEEDED
5901 [redacted]	ap-northeast-2	arn:aws:cloudformation:ap-northeast-2:5901-ef0f-4329-985d-8235ab31dd34/984314a0-[redacted]	✓ SUCCEEDED
[redacted]	[redacted]	arn:aws:cloudformation:ap-northeast-2:0710[redacted]	[redacted]

타게팅된 AWS 멤버 계정에
프로비저닝 현황

Demo

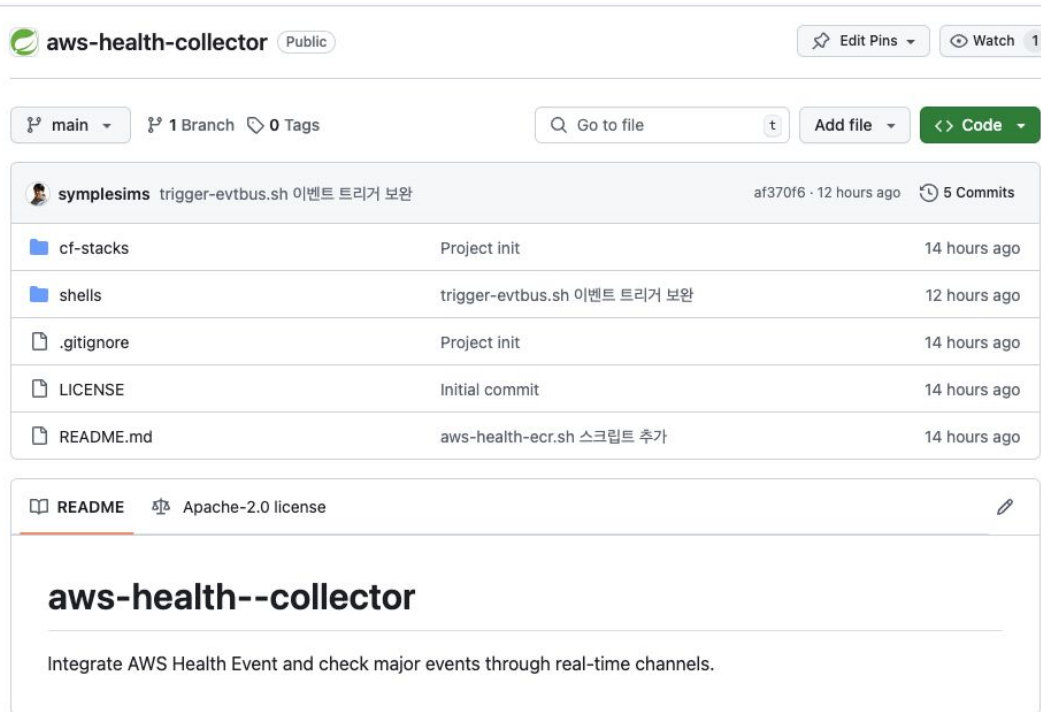


타게팅된 AWS 멤버 계정으로부터

이벤트 알림을 받을 수 있게 되었습니다.

^.^!

누구나 다 마구마구 이용해 주세요~

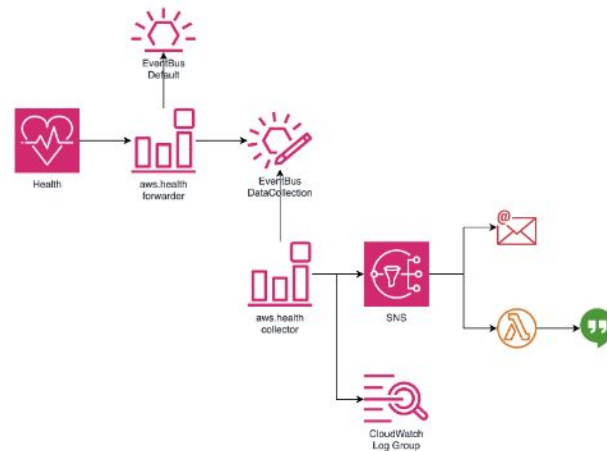


The screenshot shows the GitHub repository page for 'aws-health-collector'. At the top, it indicates the repository is public and has 1 watch. Below the repository name, there are options for 'main' branch, 1 branch, and 0 tags. A search bar and 'Add file' button are visible. The commit history table shows the following entries:

File	Commit Message	Time
cf-stacks	Project init	14 hours ago
shells	trigger-evtbus.sh 이벤트 트리거 보완	12 hours ago
.gitignore	Project init	14 hours ago
LICENSE	Initial commit	14 hours ago
README.md	aws-health-ecr.sh 스크립트 추가	14 hours ago

The README section is partially visible, showing the title 'aws-health--collector' and the description 'Integrate AWS Health Event and check major events through real-time channels.'

<https://github.com/simplydemo/aws-health-collector>



감사합니다.