MWAA 란

MWAA(Amazon Managed Workflows for Apache Airflow)는 AWS에서 제공하는 Apache Airflow 용 관리형 오케스트레이션 서비스입니다. 따라서 확장성, 가용성 및 보안을 위해 기본 인프라를 관리할 필요 없이 Airflow 및 Python을 사용하여 워크플로를 생성할 수 있습니다.

1. 특징

- 자동 Airflow 설정

  > Amazon MWAA 환경을 생성할 때 Apache Airflow 버전 및 일부 설정값을 선택하여 Apache Airflow를 빠르게 설정 합니다.

  : 기존 Airflow 환경 구성 시 Airflow 설치를 위한 종속 라이브러리라 Plugin 설치 그 외의 Meta DB나 Broker들에 대한 추가 작업이 필요하였으나 Managed Service에서는 필요한 version 및 일부 설정값 선택으로 해당 작업들이 자동으로 진행됩니다.

- 자동 크기 조절

  > Managed Airflow는 Scale in-out이 자동으로 진행되어 worker의 최소 및 최대 수를 설정하여 Worker환경에서 실행됩니다.

  : Auto Scaling 기능을 제공하므로 탄력적인 사용이 가능합니다.

- 기본 인증

  > AWS Identity and Access Management(IAM)에서 액세스 제어 정책을 정의 하여 Apache Airflow webserver에 대한 역할 기반 인증 및 권한 부여를 활성화 합니다.

  : 기존 Airflow 환경에서는 Airflow에 대해 Role을 통한 계정 생성으로 접근이 가능하였으나 Managed에서는 Iam을 통해 권한이 있는 User들만 접근할 수 있도록 기본적인 인증 절차를 제공합니다.

- 기본 보안

  > Apache Airflow worker 및 scheduler는 Amazon MWAA의 Amazon VPC에서 실행됩니다 . 또한 데이터는 KMS(AWS Key Management Service)를 사용하여 자동으로 암호화됩니다.

  : 기본적으로 Scheduler와 Worker는 Managed Service에서 제공하는 VPC에 구성되고 데이터 암호화도 AWS KMS를 통해 수행할 수 있습니다.

- Public 또는 Private Access Mode

  > Public 또는 Private Access Mode를 사용하여 Apache Airflow webserver에 액세스합니다 . Public 또는 Private 네트워크 여부에 따라 인터넷 또는 VPC 내부에서 접근 가능합니다.

  : 기존 Airflow 환경도 최초 구성 시 어디에 Webserver를 구성하냐에 따라 똑같은 기능을 제공할 수 있지만 Container 형태의 Webserver를 제공하므로 보다 빠르고 쉽게 구성할 수 있습니다.

- 간소화된 업그레이드 및 패치

  > Amazon MWAA는 새로운 버전의 Apache Airflow를 주기적으로 제공합니다.

  : 관리자가 지속적인 업그레이드 작업을 진행할 필요가 없습니다.

- 워크플로 모니터링

> Amazon CloudWatch에서 Apache Airflow 로그 및 Apache Airflow 지표를 확인하여 Apache Airflow 작업 지연 또는 워크플로 오류를 식별합니다.

: Service에 대한 모든 Log는 CloudWatch Log group에서 관리 및 확인할 수 있고 지표를 통해 확인 가능합니다.

- AWS 통합

> Amazon MWAA는 Amazon Athena, AWS Batch, Amazon CloudWatch, Amazon DynamoDB, AWS DataSync, Amazon EMR, AWS Fargate, Amazon EKS, Amazon Kinesis Data Firehose, AWS Glue, AWS Lambda, Amazon Redshift, Amazon과의 오픈 소스 통합을 지원합니다.

- 작업자 플릿

> Amazon MWAA는 AWS Fargate의 Amazon ECS를 사용하여 요청 시 Fleet을 확장하고 스케줄러 중단을 줄이기 위해 컨테이너를 사용할 수 있는 지원을 제공합니다.

각 Amazon MWAA 환경에는 스케줄러, 웹 서버 및 1 작업자가 포함됩니다. 작업자는 시스템 로드에 따라 확장 및 축소됩니다. 환경의 로드를 모니터링하고 클래스를 언제든지 수정할 수 있습니다.

	DAG 용량*	스케줄러 CPU	작업자 CPU	웹 서버 CPU
mw1.small	최대 50	1 vCPU	1 vCPU	0.5 vCPU
mw1.medium	최대 250	2 vCPU	2 vCPU	1 vCPU
mw1.large	최대 1000	4 vCPU	4 vCPU	2 vCPU

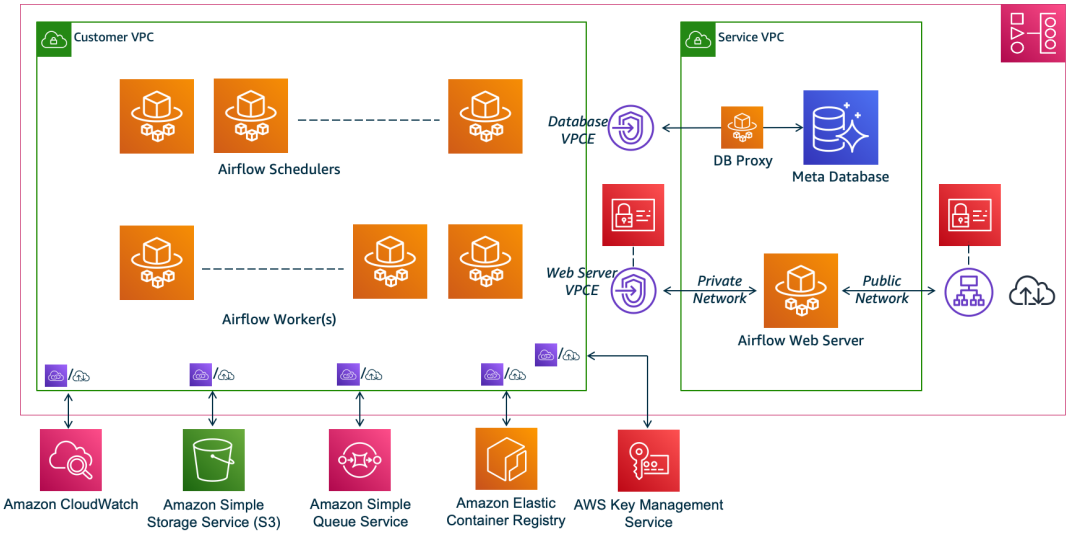
*일반적인 사용량에서

Fleet 이란 기본적으로 Instance를 Group 형태로 지원하는 것인데 단일 유형이 아닌 복수의 유형을 지원하는 것 [예) EMR Instance Fleet]

따라서 상단의 사양을 충족하는 Type이면 여러 유형에 거쳐서 사용이 가능하기 때문에 Resource에 대한 제한이 적어져 Scale out 시 중단 시간에 대한 제한 사항이 적어집니다.
(예를 들어 EC2를 t3.small type으로 10개를 증가 시킬 때 해당 리전에 Resource가 없으면 같은 사양의 다른 type으로 생성)

2. Architecture

Amazon MWAA Architecture

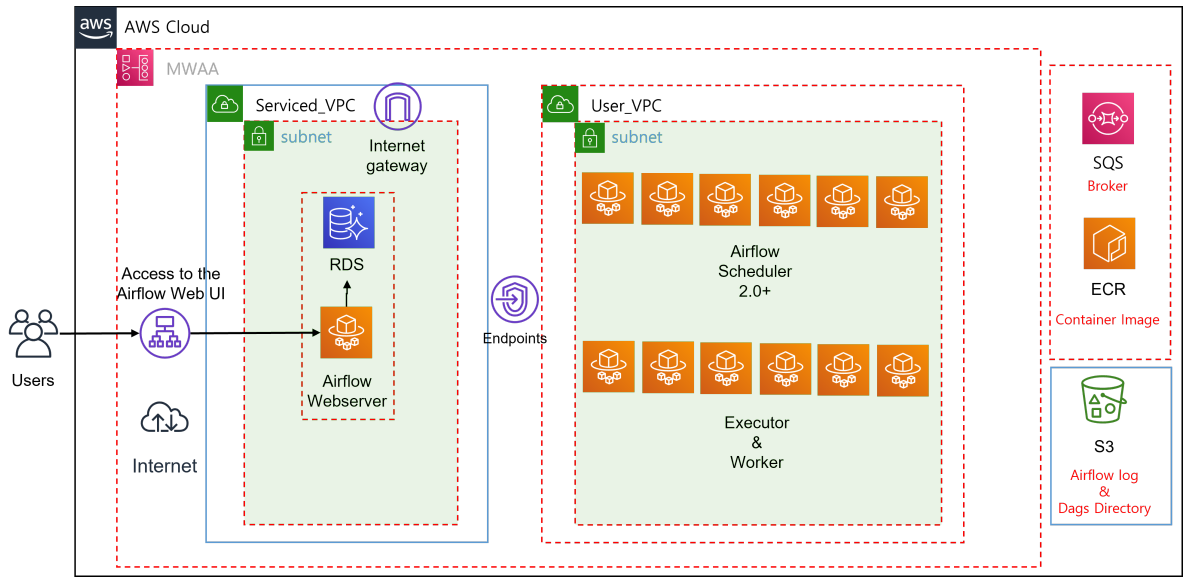


(Image_URL : https://docs.aws.amazon.com/mwaa/latest/userguide/images/mwaa-architecture.png)

상단의 이미지는 Amazon에서 제공하는 MWAA Architecture입니다.

- Apache Airflow 스케줄러 및 작업자는 환경에 대한 Amazon VPC의 프라이빗 서브넷에 연결하는 AWS Fargate(Fargate) 컨테이너입니다.
- 각 환경에는 비공개로 보호되는 VPC 엔드포인트를 통해 스케줄러 및 작업자 Fargate 컨테이너에 액세스할 수 있는 AWS에서 관리하는 자체 메타 데이터베이스를 Aurora로 구성합니다.
- Amazon CloudWatch, Amazon S3, Amazon SQS, Amazon ECR 및 AWS KMS는 Amazon MWAA와 별개 이며 Fargate 컨테이너 의 Apache Airflow 스케줄러 및 작업자에서 액세스할 수 있어야 합니다.
- Apache Airflow 웹 서버 는 공용 네트워크 Apache Airflow 액세스 모드 를 선택 하여 인터넷 을 통해 액세스하거나 사설 네트워크 Apache Airflow 액세스 모드 를 선택하여 VPC 내에서 액세스할 수 있습니다.
- 두 경우 모두 Apache Airflow 사용자에 대한 액세스는 AWS Identity and Access Management(IAM)에서 정의한 액세스 제어 정책에 의해 제어됩니다.

(자료 출처 : https://docs.aws.amazon.com/mwaa/latest/userguide/what-is-mwaa.html#architecture-mwaa)



상단의 아키텍쳐는 MWAA가 제공해주는 Resource와 사용자가 구성해야하는 Resource로 분류하여 다시 만들어진 것입니다.

공통적으로 MWAA에서 자체적으로 만들어진 Resource에 대해서는 관리자나 사용자 모두 접근하거나 변경할 수 없습니다.

사용자 구성 (파란색 박스)

- Airflow Webserver 및 MetaData Database가 위치해야할 네트워크를 선택합니다

  해당 VPC 액세스 모든에 따라 public 또는 private 서브넷이 구성되며 해당 서브넷 안에 AWS Fargate를 사용한 Webserver와 Aurora를 사용한 Meta DB가 생성됩니다.

  해당 Webserver는 MWAA에서 생성한 Load Balancer를 통해 접근됩니다.
- MWAA의 Dags, Plugins, 또는 추가 Pyhon library에 대해 정의한 requirements.txt를 정의한 File들을 저장할 수 있는 S3 Bucket 및 Folder를 선택합니다.
  - 해당 Bucket 및 폴더에서 DAG나 Plugin, Library 정보를 가지고 MWAA 구성합니다

MWAA로 자동 구성 (빨간색 박스)

- Worker 및 Scheduler 역할을 할 노드들을 AWS Fargate로 구성되며 해당 노드들은 MWAA가 구성한 Private한 네트워크에서 구성됩니다.
- Broker 역할을 할 SQS나 Conatiner 이미지를 제공할 ECR도 구성됩니다.

3. MWAA 환경 구성

1. 관련 서비스

   Amazon MWAA는 Amazon MWAA가 프로비저닝 중에 환경에서 사용하는 다른 AWS 서비스를 사용할 수 있도록 계정에 서비스 연결 역할을 생성합니다.

   - Amazon ECR : Apache Airflow 용 이미지가 호스팅됩니다.
   - CloudWatch Logs : Apache Airflow 로그를 저장합니다.
   - Amazon EC2 : 하단의 Resource 들이 생성됩니다.
     - Webserver, Scheduler, Worker 용 AWS Fargate
     - AWS Fargate 및 AWS 관리형 Amazon Aurora PostgreSQL 데이터베이스 클러스터용 Amazon VPC 엔드포인트 .
     - Apache Airflow 웹 서버에 대해 프라이빗 네트워크 옵션을 선택한 경우 웹 서버에 대한 네트워크 액세스를 활성화하는 추가 Amazon VPC 엔드포인트 .
     - Amazon VPC에서 호스팅되는 AWS 리소스에 대한 네트워크 액세스를 활성화하기 위한 Amazon VPC의 ENI.
2. 구성 권한

   Amazon MWAA가 사용자 환경에서 사용하는 다른 AWS 서비스를 사용할 수 있는 권한들이 필요합니다.

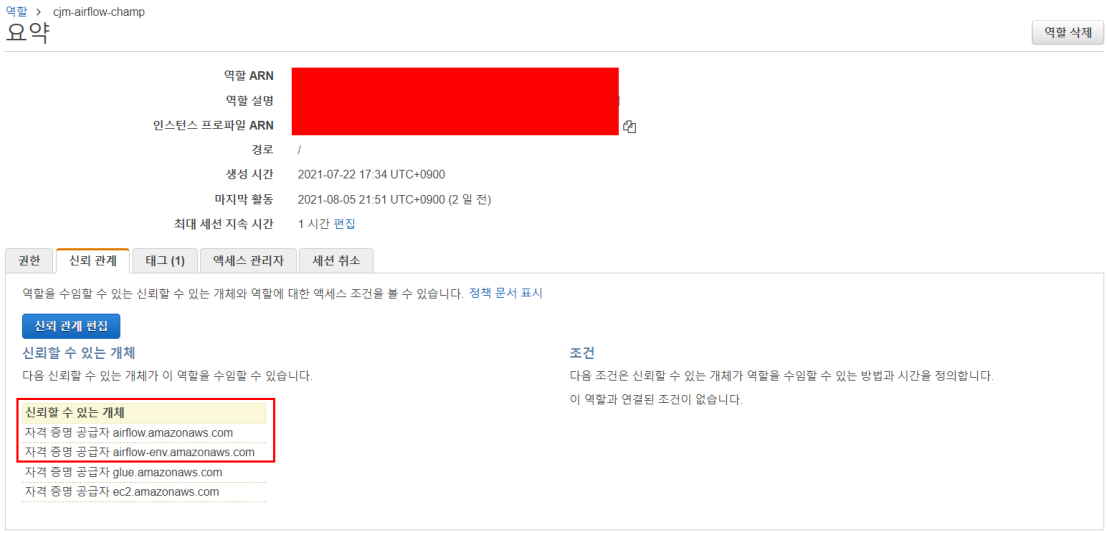
   하단의 권한들은 해당 서비스들을 이용할 시에 IAM Role에 추가 되어야 합니다.

   - CloudWatch – Apache Airflow 지표 및 로그를 전송
   - Amazon S3 – 환경의 DAG 코드 및 지원 파일(예: a `requirements.txt`) 을 구문 분석
   - Amazon SQS – 환경의 Apache Airflow 작업을 대기열에 넣습니다.
   - AWS KMS – 환경의 데이터 암호화용.

4. MWAA 환경 생성

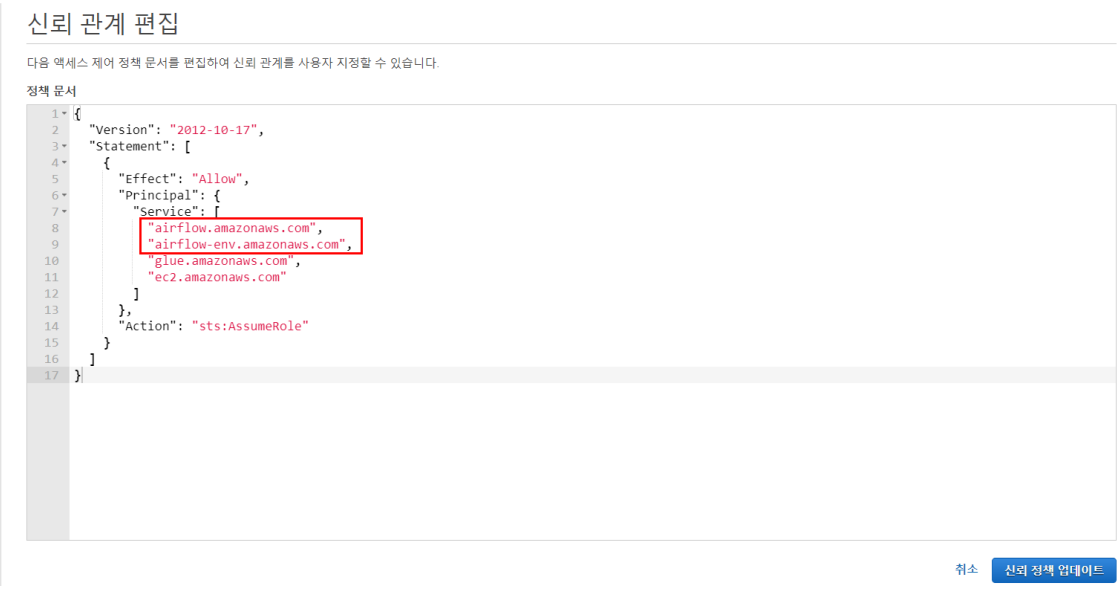
1. IAM 구성

기존에 EC2에 구성하던 IAM Role 권한과 신뢰관계를 추가함으로써 사용가능합니다.



상단 예시 이미지와 같이 `airflow`, `airflow-env`에 대한 신뢰 관계가 추가 되어야 합니다.

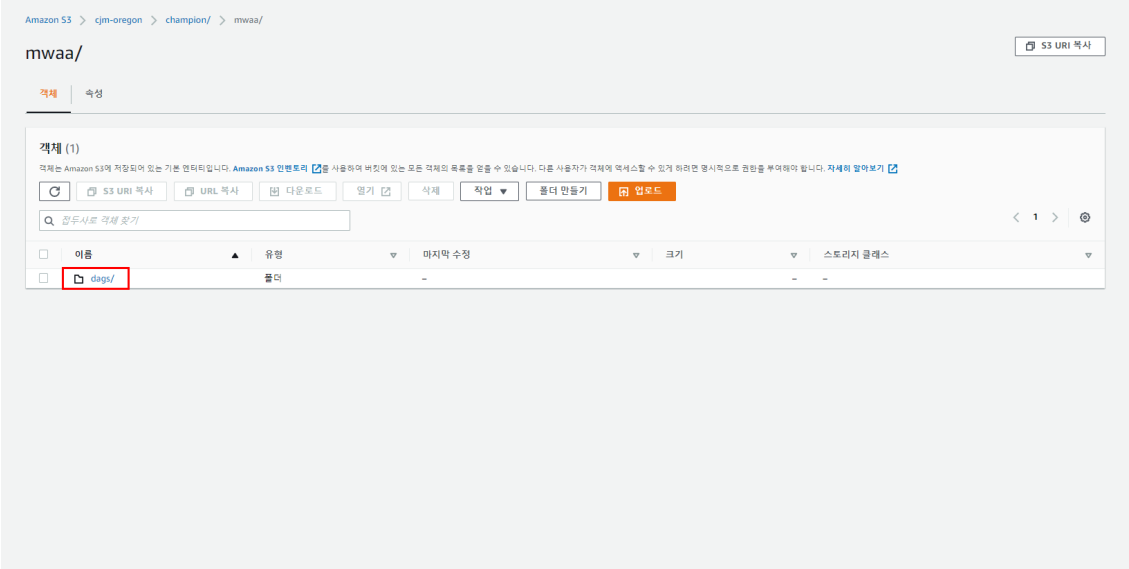
신뢰 관계는 신뢰 관계 편집에서 Json으로 수정 가능합니다.

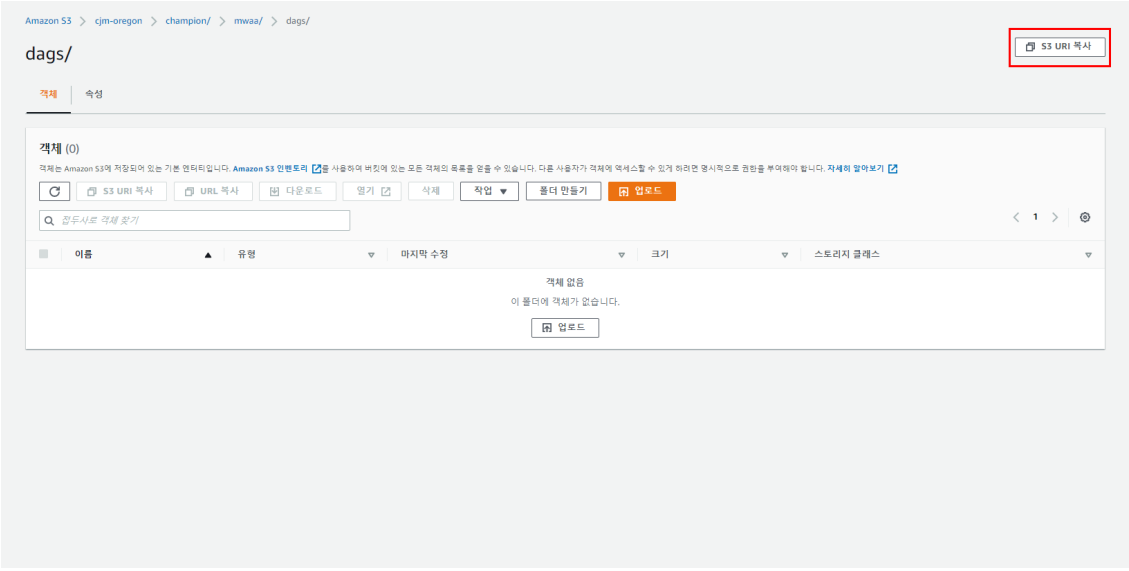


추가적으로 다른 서비스들에 대한 추가 권한을 부여합니다.

2. S3 구성

MWAA에서 사용할 Bucket과 Dag를 저장할 Folder를 구성합니다.





오른쪽 상단의 `S3 URI 복사`를 통해 S3 URI를 쉽게 복사할 수 있습니다.

상단의 두 이미지 외에도 Plugin이나 requirements.txt에 대한 Foler도 구성하면 사용 가능합니다.

3. VPC 및 SG

개인용 VPC를 하나 생성합니다.

SG는 하단과 같이 구성합니다.

1. MWAA에 모든 접근이 가능한 SG 생성

유형	규약	소스 유형	원천
모든 트래픽	모두	모두	0.0.0.0/0

2. 상단의 SG를 참조하는 SG가 필요합니다.
   상단의 SG는 MWAA와 외부 서비스들 간의 SG이기 때문에 내부 Resource들에 대한 설정은 상단에 생성한 SG들을 참조하는 SG를 생성하고 부착해줌으로써 설정 가능합니다.

유형	규약	소스 유형	원천
모든 트래픽	모두	모두	sg-0909e8e81919 / my-mwaa-vpc-security-group

3. Meta DB용 SG

   Aurora PG이므로 5432 포트에 대한 허용이 되는 SG를 생성합니다.
   2번과 같은 이유로 1번에서 생성한 SG를 참조합니다.

유형	규약	포트 범위	소스 유형	원천
HTTPS	TCP	5432	Custom	sg-0909e8e81919 / my-mwaa-vpc-security-group

위의 3개의 SG를 1개로 구성할 수 있습니다

자기참조 SG를 생성하므로써 1개의 SG에 구성가능합니다.

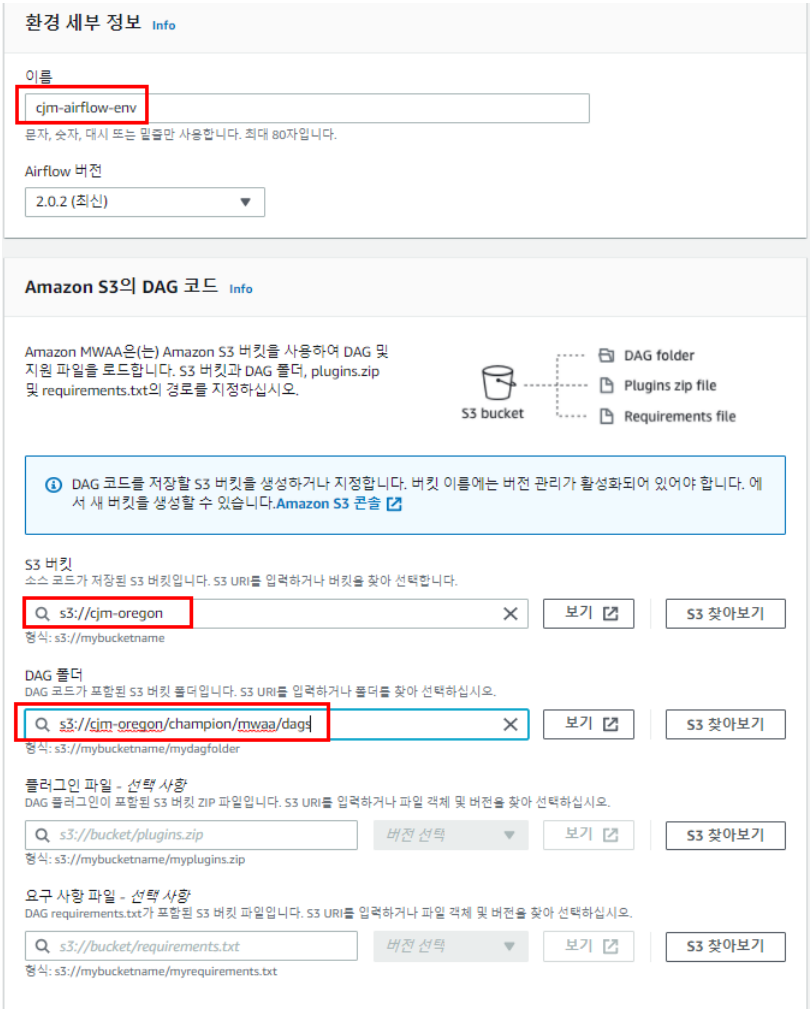
유형	규약	포트 범위	원천
모든 트래픽	모두	모두	sg-0909e8e81919 / my-mwaa-vpc-security-group
모든 트래픽	모두	모두	0.0.0.0/0

4. mwaa env 생성

1. 환경 세부 정보
   이름 : `cjm-airflow-env`
   Airflow 버전 : `2.0.2`

2. Amazon S3의 DAG 코드
   마지막 `/`는 포함되지 않습니다.

   S3 버킷 : `s3://cjm-oregon`
   DAG 폴더 : `s3://cjm-oregon/champion/mwaa/dags`
   플러그인 파일, 요구 사항 파일 은 선택사항 입니다.



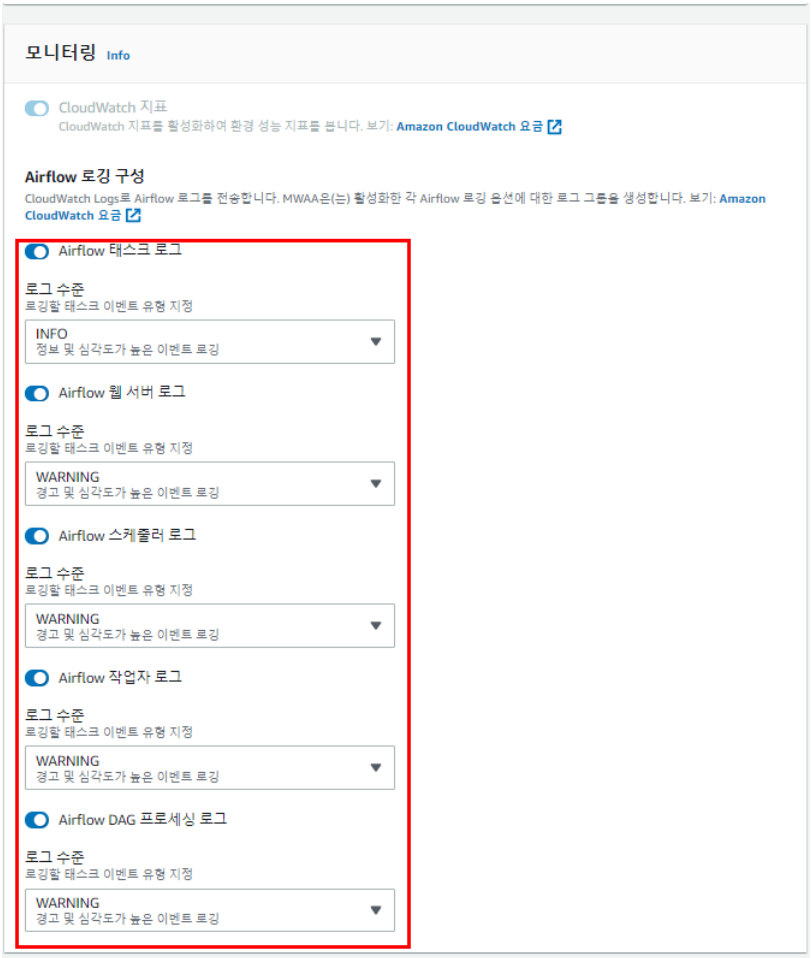
3. 네트워킹

   VPC : `개인 VPC`
   Subnet 1 : `VPC 내부 Private Subnet - c`
   Subnet 2 : `VPC 내부 Private Subnet - a`
   웹 서버 엑세스 : `Webserver 접근 방식을 설정`
   보안 그룹 : `위에서 생성한 SG`

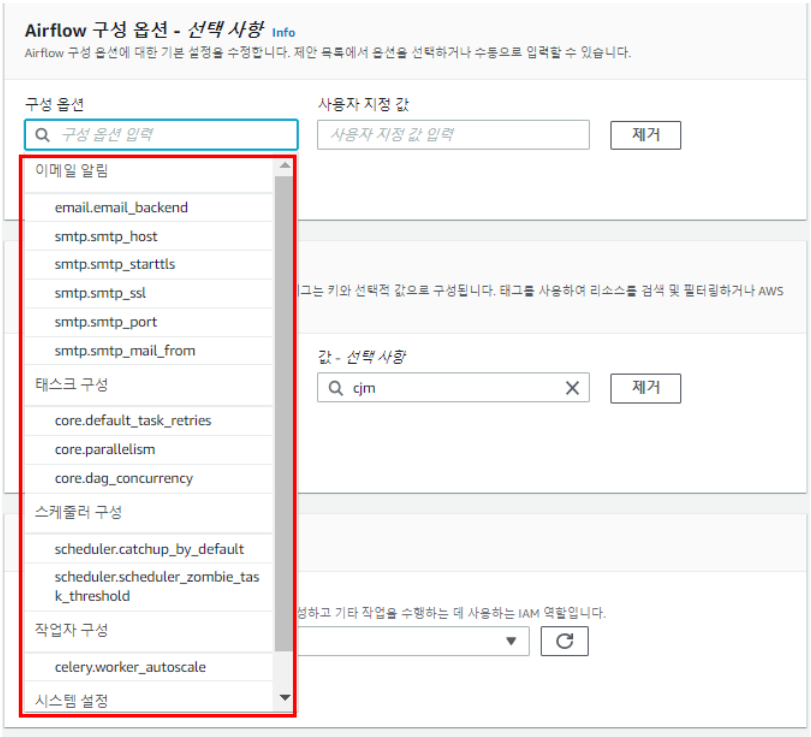
4. 환경 클래스

   Type 설정 : `mw1.small`, `mw1.medium`, `mw1.large` 중 선택
   최대 작업자 수 : `1 ~ 25`
   최소 작업자 수 : `1 ~ 최대 작업자 수`
   스케줄러 수 : `2 ~ 5`

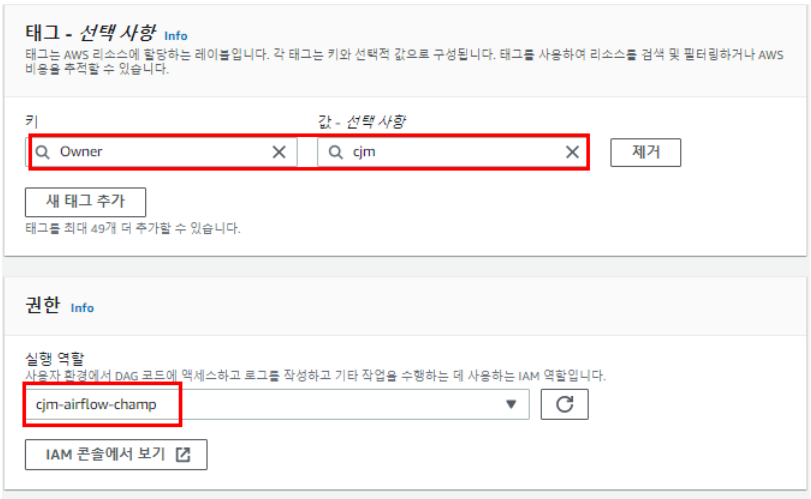
5. 로그 설정
   CloudWatch Log Groups에 적재할 Log들을 설정합니다.



6. Airflow 구성 옵션
   MWAA에서 제공하는 Option에 한해서 변경 가능합니다.



7. 권한 및 태그
   위에서 생성한 권한을 부여합니다.



상단의 일련의 과정 후 생성하면 생성까지 2 ~ 30분의 생성 시간 소요 후 생성됩니다.

과제

개인 Airflow 구성(AWS MWAA) 및 문서 작성