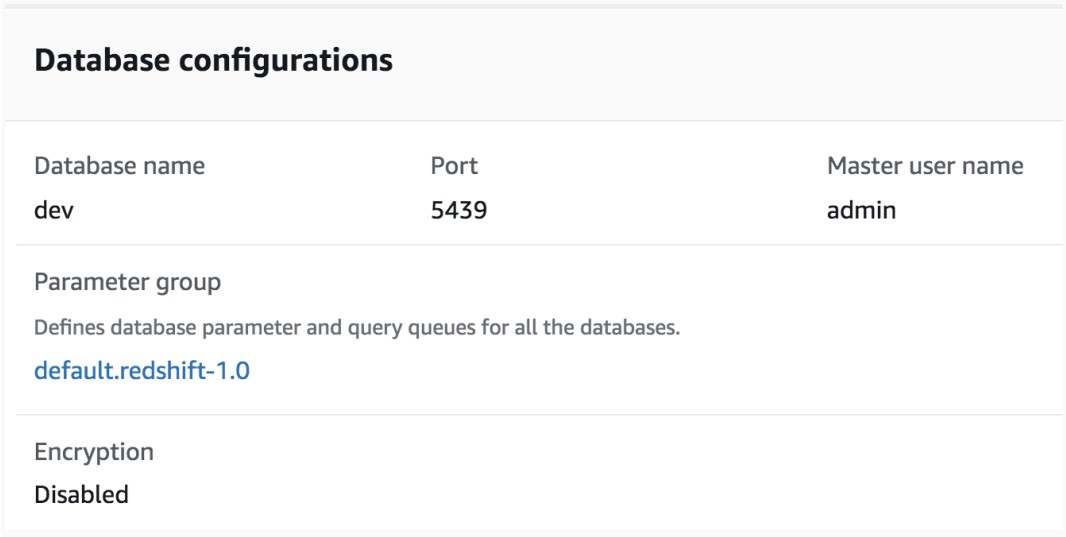
Redshift Cluster 설정하기

이 글에서는 Redshift 클러스터를 만들고 여기에 실습에 사용했던 스키마와 테이블들을 다시 만드는 방법에 대해 설명한다. 하나 중요한 부분은 Redshift 클러스터를 만들때 외부 액세스가 가능하도록 설정을 해야한다는 점(1번 Redshift를 외부에서 접근할 수 있게 하기)인데 외부 액세스가 불가능하게 한번 생성한 후에는 다시 오픈하는 것이 불가능하기 때문이다.

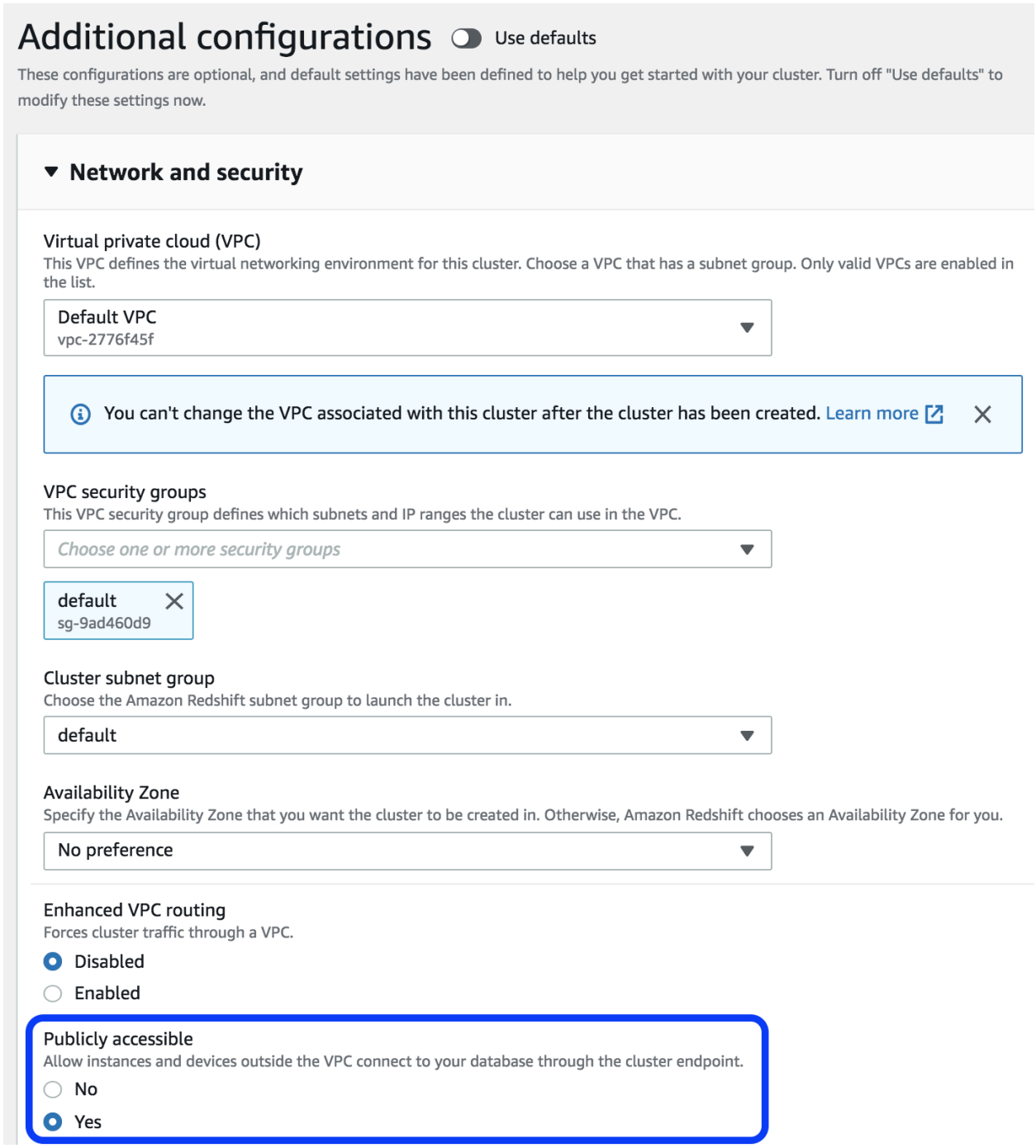
목차**:**

Redshift cluster의 이름이 grepp-data라고 가정. 그리고 해당 클러스터의 호스트 이름과 포트 번호와 데이터베이스와 사용자ID와 패스워드를 기록해두어야 함 (이는 Redshift 클러스터의 Properties에서 Database configurations를 선택하면 볼 수 있음):

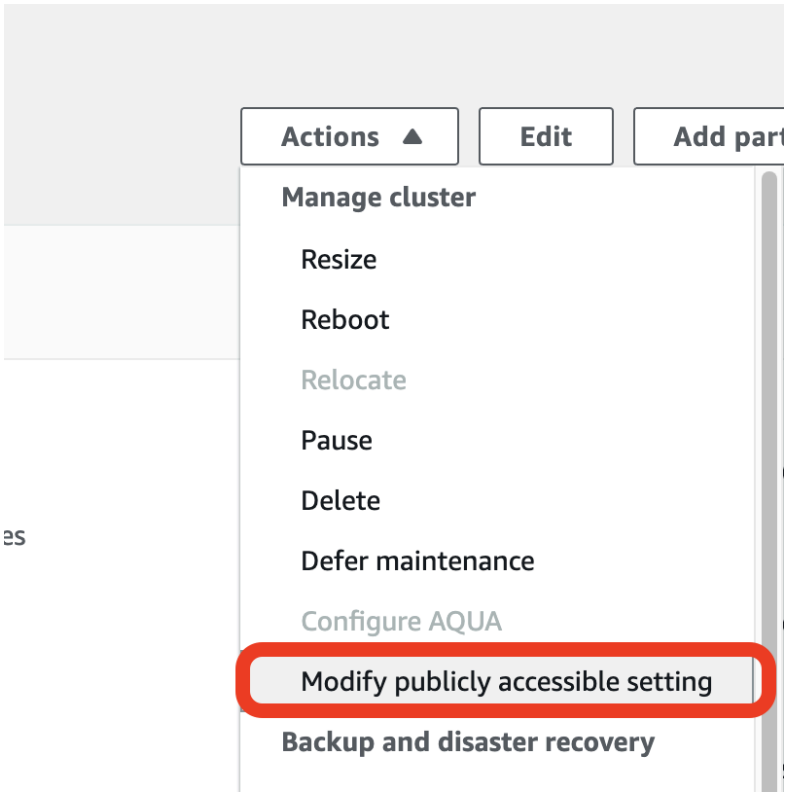


1. Redshift를 외부에서 접근할 수 있게 하기

    - 이는 Redshift를 처음 생성할 때 정해주어야 한다. Additioanl Configuration 섹션의 Network and security에서 Publicly accessible의 값으로 Yes를 선택한다.

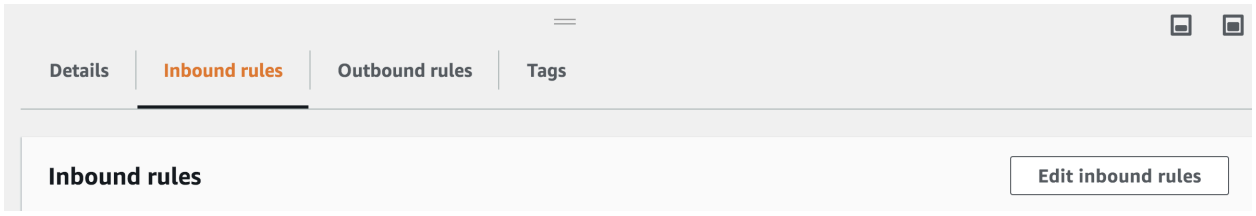


만일 위의 Publicly accessible가 안 보이면 해당 Redshift 인스턴스의 세부 페이지에서 오른쪽 상단의 Actions 메뉴를 보면 아래와 같은 메뉴(Modify publicly accessible setting)가 보이는지 확인해보세요:

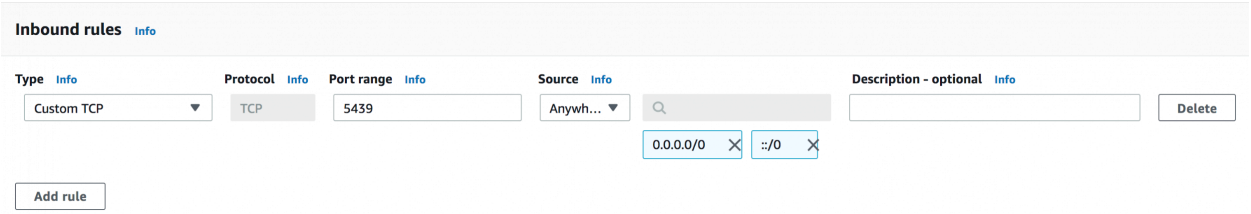


2. Redshift의 TCP 포트번호를 열어주기

- 다음은 Security Group의 설정을 변경해서 TCP port number 5439를 오픈해야 한다. 위에서 이미 사용했던 “Network and Security” 섹션을 다시 방문해서 보면 VPC Security Group 링크가 존재한다.
- 그 링크를 클릭하면 Security Groups 페이지로 이동하는데 거기서 “Edit Inbound Rules” 버튼을 클릭한다.



- 여기서 아래와 같이 입력하고 Add Rule 버튼을 누른다. 이 경우는 외부에서 모든 액세스를 가능하게 하기위해서 Source로 Anywhere를 선택한 경우이다. 지금 이 콘솔을 액세스하고 있는 컴퓨터의 IP에게만 액세스를 주고 싶다면 My IP를 선택한다.



3. SQL 에디터에서 Redshift를 admin 계정으로 액세스한 후에 스키마와 그룹들을 생성

https://github.com/keeyong/data-engineering/blob/main/sql/admin.md

스키마 생성

```
CREATE SCHEMA raw_data;
CREATE SCHEMA analytics;
CREATE SCHEMA adhoc;
```

그룹 생성과 권한 지정

```
CREATE GROUP analytics_users;
GRANT USAGE ON SCHEMA analytics TO GROUP analytics_users;
GRANT USAGE ON SCHEMA raw_data TO GROUP analytics_users;
GRANT SELECT ON ALL TABLES IN SCHEMA analytics TO GROUP analytics_users;
GRANT SELECT ON ALL TABLES IN SCHEMA raw_data TO GROUP analytics_users;
GRANT ALL ON SCHEMA adhoc to GROUP analytics_users;
GRANT ALL ON ALL TABLES IN SCHEMA adhoc TO GROUP analytics_users;
```

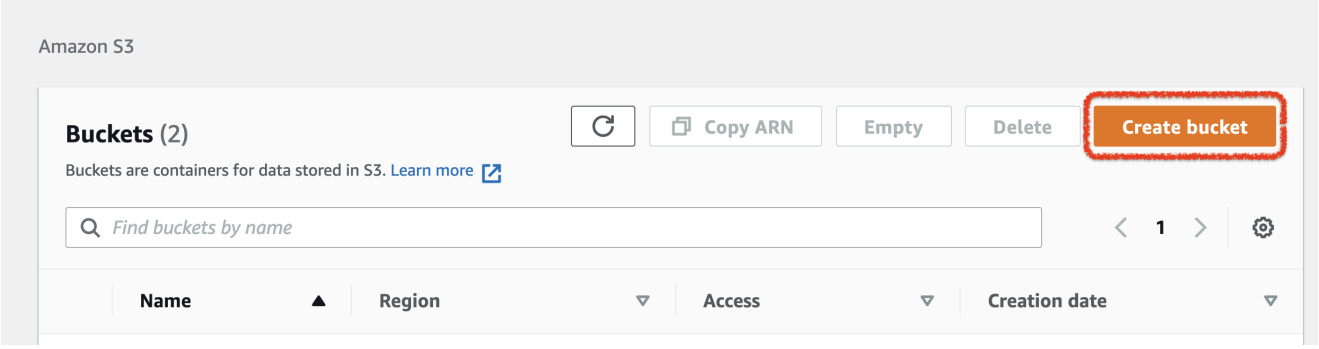
일반 사용자 생성과 권한 지정

```
CREATE USER keeyong PASSWORD '...';
ALTER GROUP analytics_users ADD USER keeyong;
```

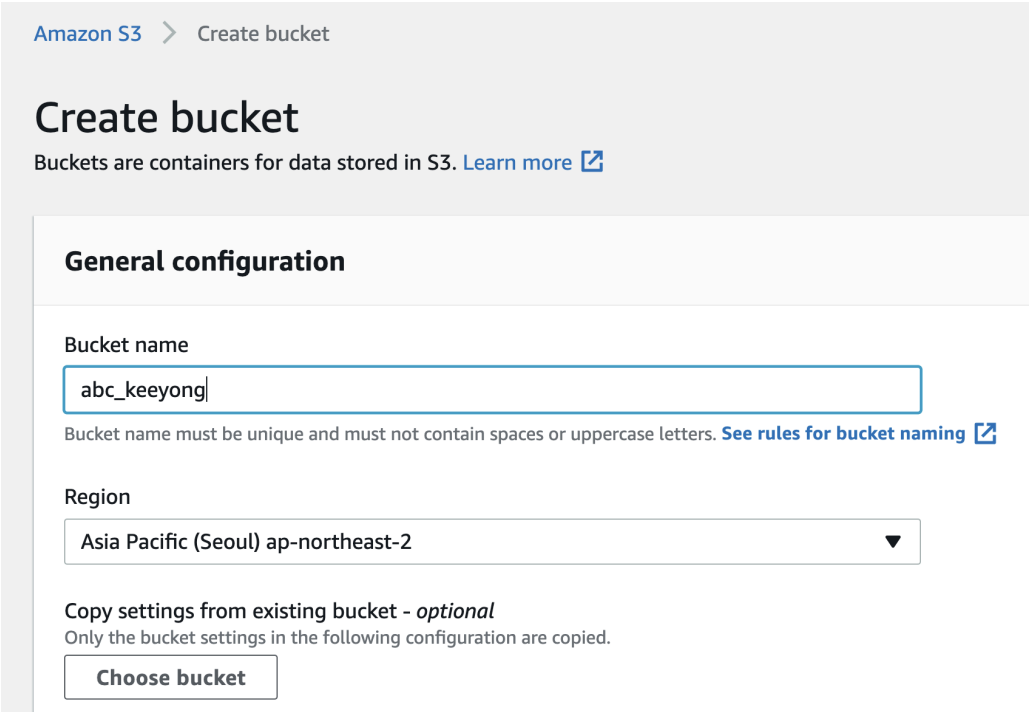
4. raw_data.zip 파일의 압축을 풀기
   a. 압축을 풀면 거기 4개의 CSV파일들이 있다
   b. 이 4개 모두 raw_data 스키마 밑에 각각의 테이블로 존재하게 됨 (강의에서 사용했던 테이블들이기에 이름이 친숙해야함)

5. S3 bucket을 하나 만들어야함.
   a. 이 4개의 파일들을 COPY SQL로 복사해야하기 때문에 먼저 S3로 업로드해야하며 이 때문에 S3 버켓을 만들어야 함.

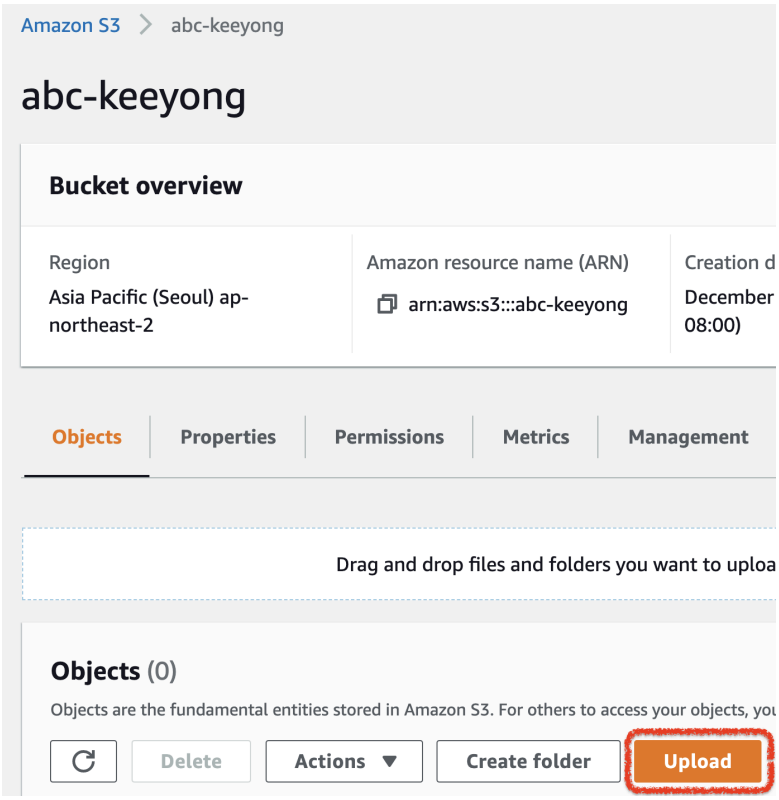
b. 이를 위해 AWS console에서 S3 서비스로 이동: https://s3.console.aws.amazon.com/s3/home?region=ap-northeast-2
c. 여기서 Create bucket 버튼을 클릭해서 적당한 이름의 버켓을 생성.



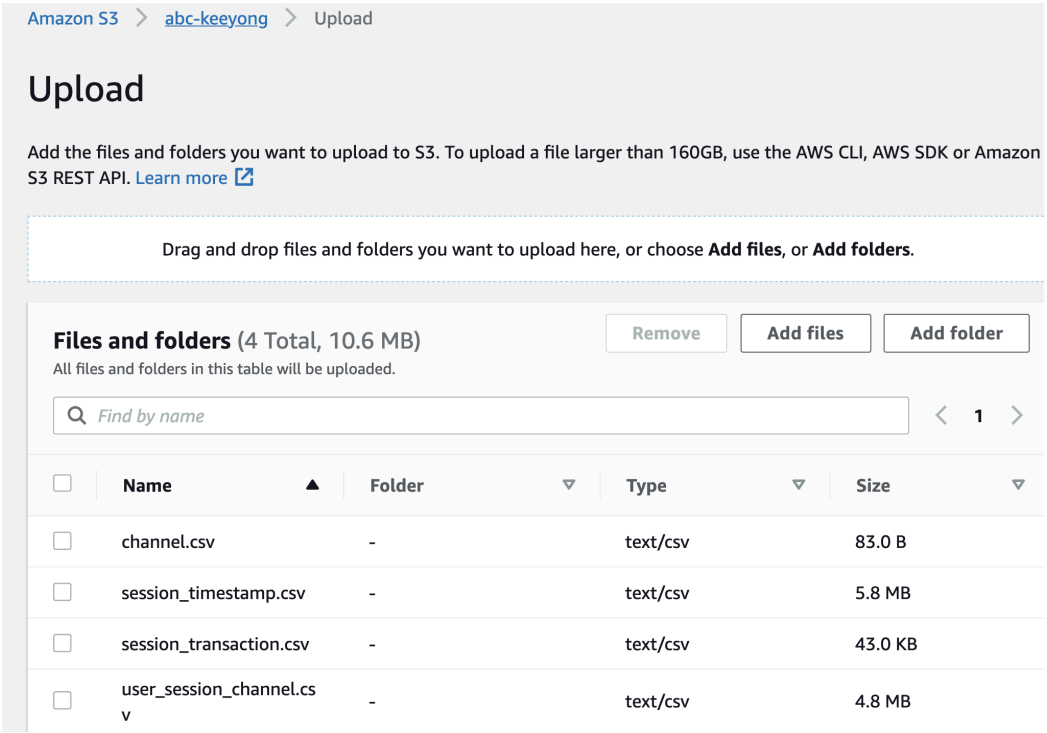
d. **abc-keeyong**이라고 해보겠다. 각자 유일한 이름을 하나 골라서 사용하기를 ... 밑으로 스크롤해서 Create Bucket 버튼을 클릭



6. 이제 앞서 4개의 CSV 파일들을 S3로 업로드
   a. 위에서 만든 abc-keeyong이란 이름의 S3 bucket으로 복사한다.
   b. S3 콘솔에서 abc-keeyong을 선택하고 **Upload** 버튼을 선택해서 앞의 4개의 CSV 파일들을 업로드 한다.



c. 그리고나면 아래와 같이 4개의 CSV 파일이 보여야 한다.



이 4개의 CSV 파일들의 S3 링크는 아래와 같다:

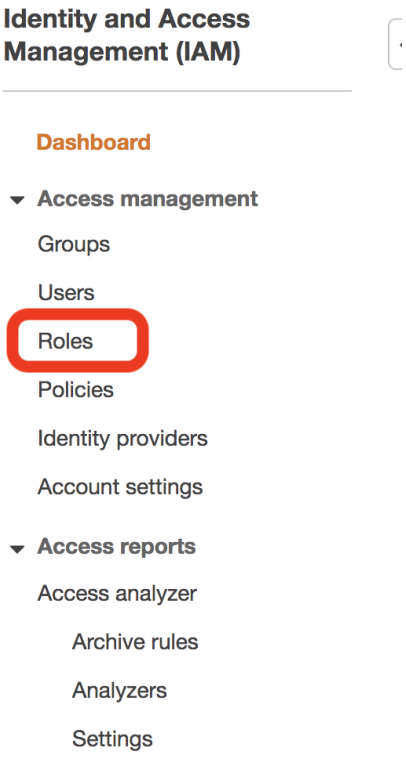
- s3://abc-keeyong/channel.csv
- s3://abc-keeyong/session_timestamp.csv

- s3://abc-keeyong/session_transaction.csv
- s3://abc-keeyong/user_session_channel.csv

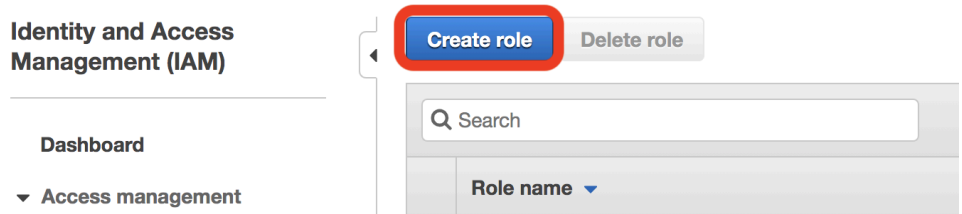
7. Redshift에게 위 S3 버켓에 대한 액세스 권한 지정이 필요

최종적으로 COPY 명령을 실행하기 위해서는 S3 액세스 권한을 Redshift에 지정해주어야 한다. 이를 위해서 AWS IAM(Identity and Access Management)을 이용해 이에 해당하는 Role을 만들고 이를 Redshift에 grant해야 한다. 주의할 점은 IAM을 사용할 때는 반드시 어드민 권한을 갖고 있는 사용자로 로그인한 상태이어야 한다 (root 사용자).

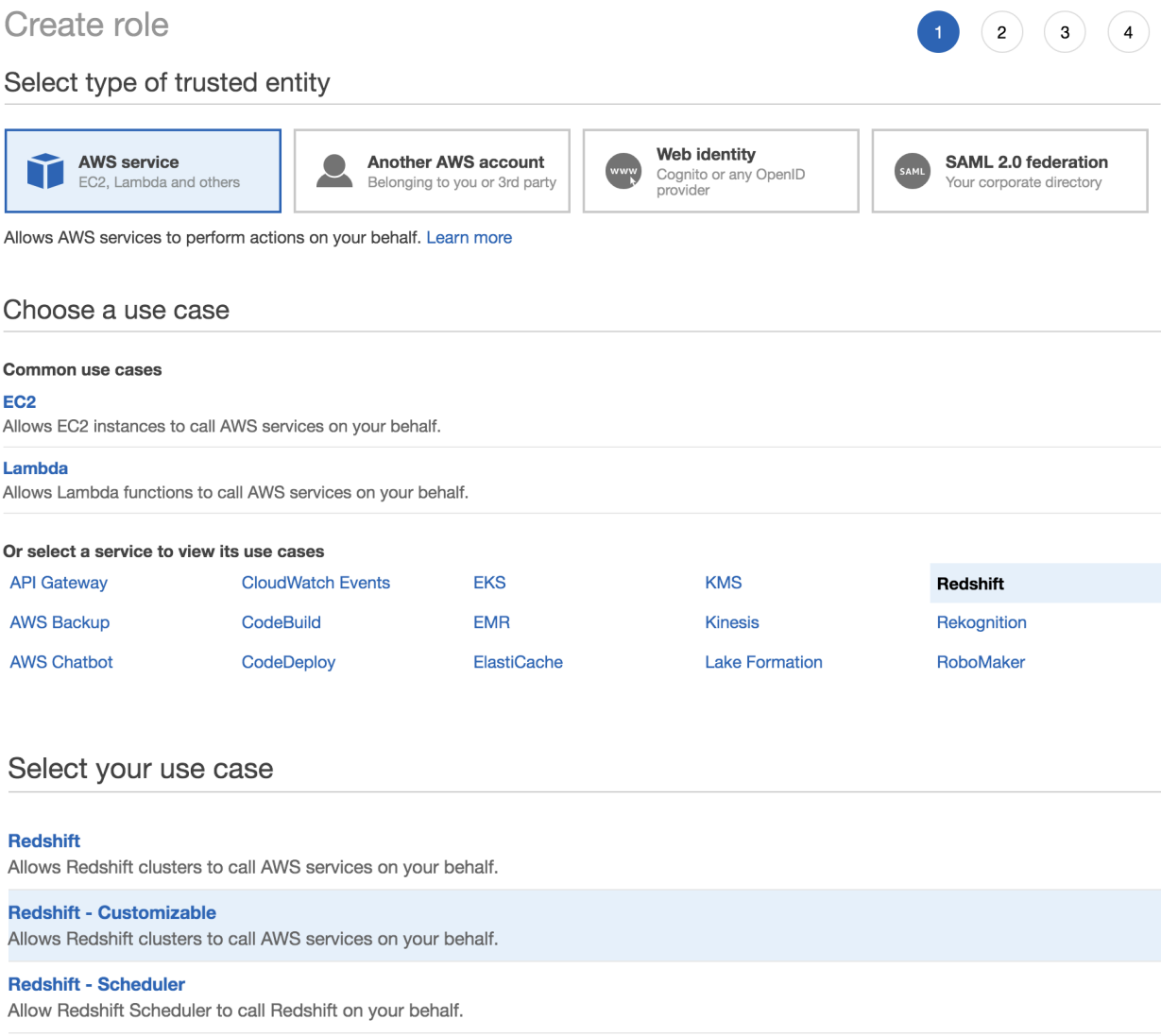
1. https://console.aws.amazon.com/iam/home을 방문한다.
2. 왼쪽 메뉴에서 Role을 선택한다



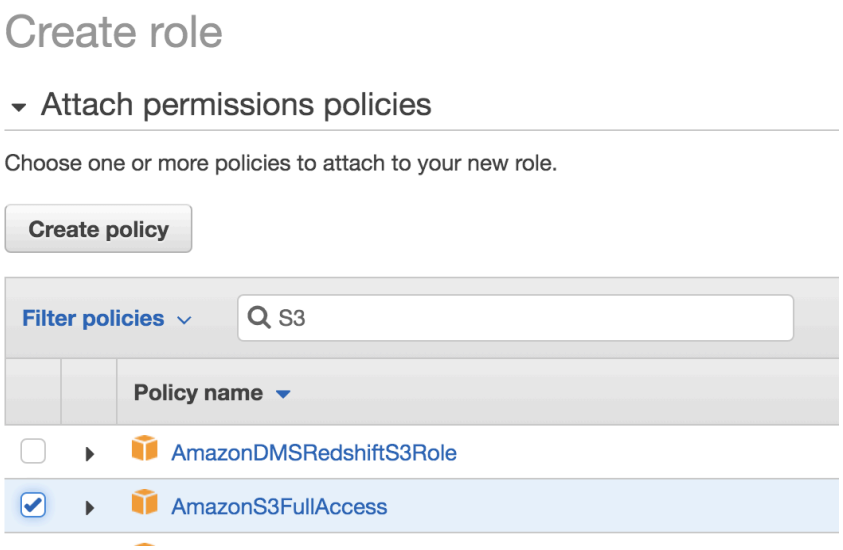
c. Create Role을 선택한다



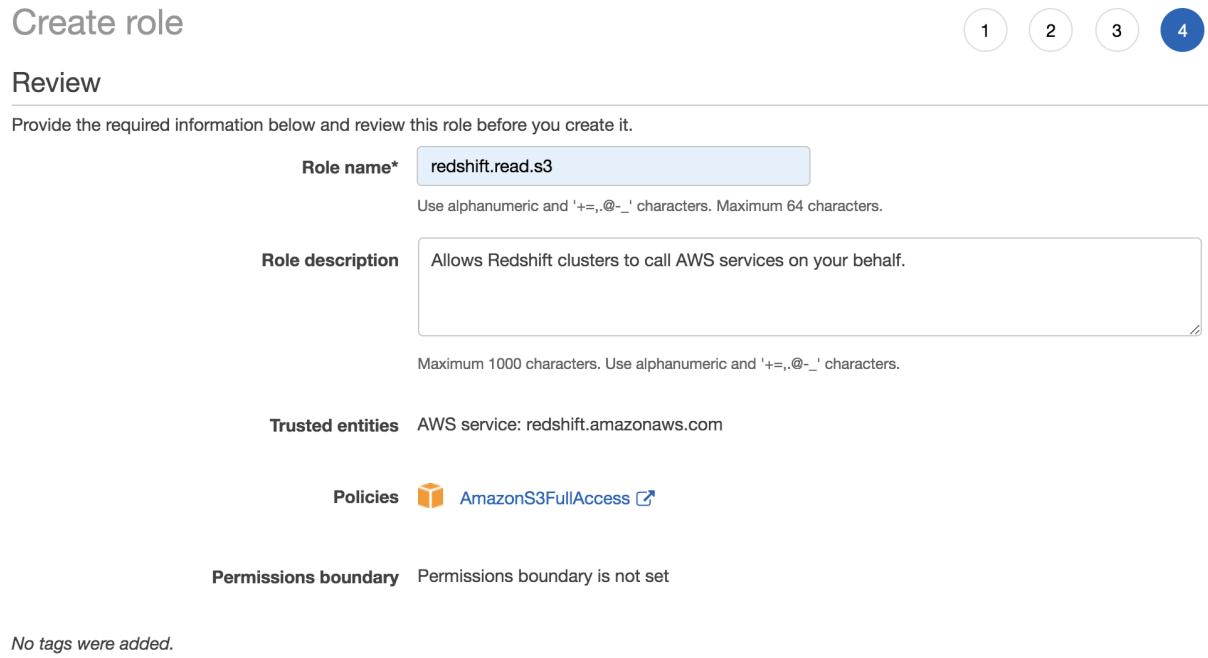
d. AWS service를 선택하고 아래 리스트에서 Redshift를 선택하고 이어서 Redshift - Customizable (Allows Redshift to call AWS services on your behalf)를 선택하고 Next Permissions 버튼을 클릭한다.



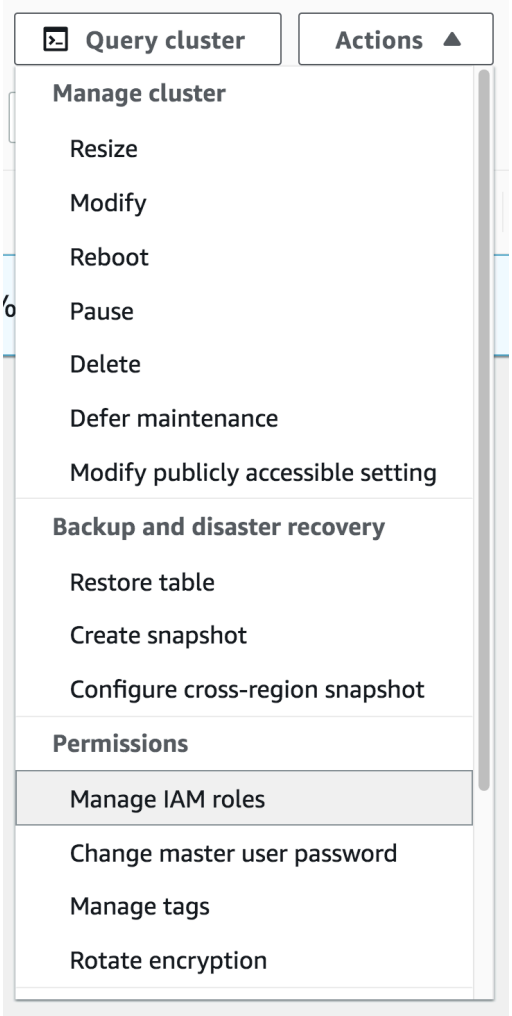
5. 다음 Policies 박스에서 AmazonS3FullAccess를 찾아서 왼쪽의 체크박스를 선택하고 Next: Tags 버튼을 클릭한다



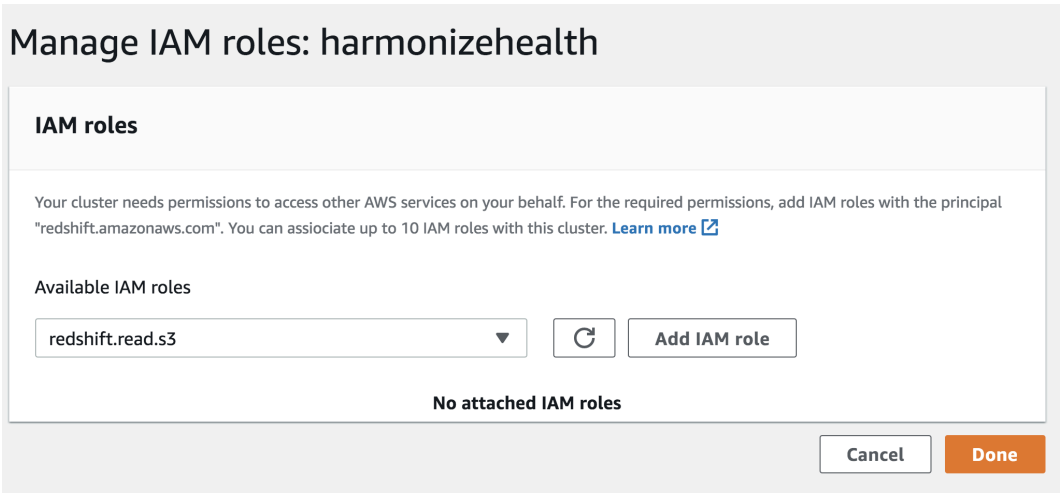
6. Next: Review 버튼을 클릭하고 최종 이름으로 redshift.read.s3을 지정하고 최종 생성한다.



7. 마지막으로 이 권한을 Redshift cluster에 지정해야 한다. Redshift console로 돌아가 해당 클러스터를 선택한 후 Action 메뉴에서 “Manage IAM roles”를 선택한다.

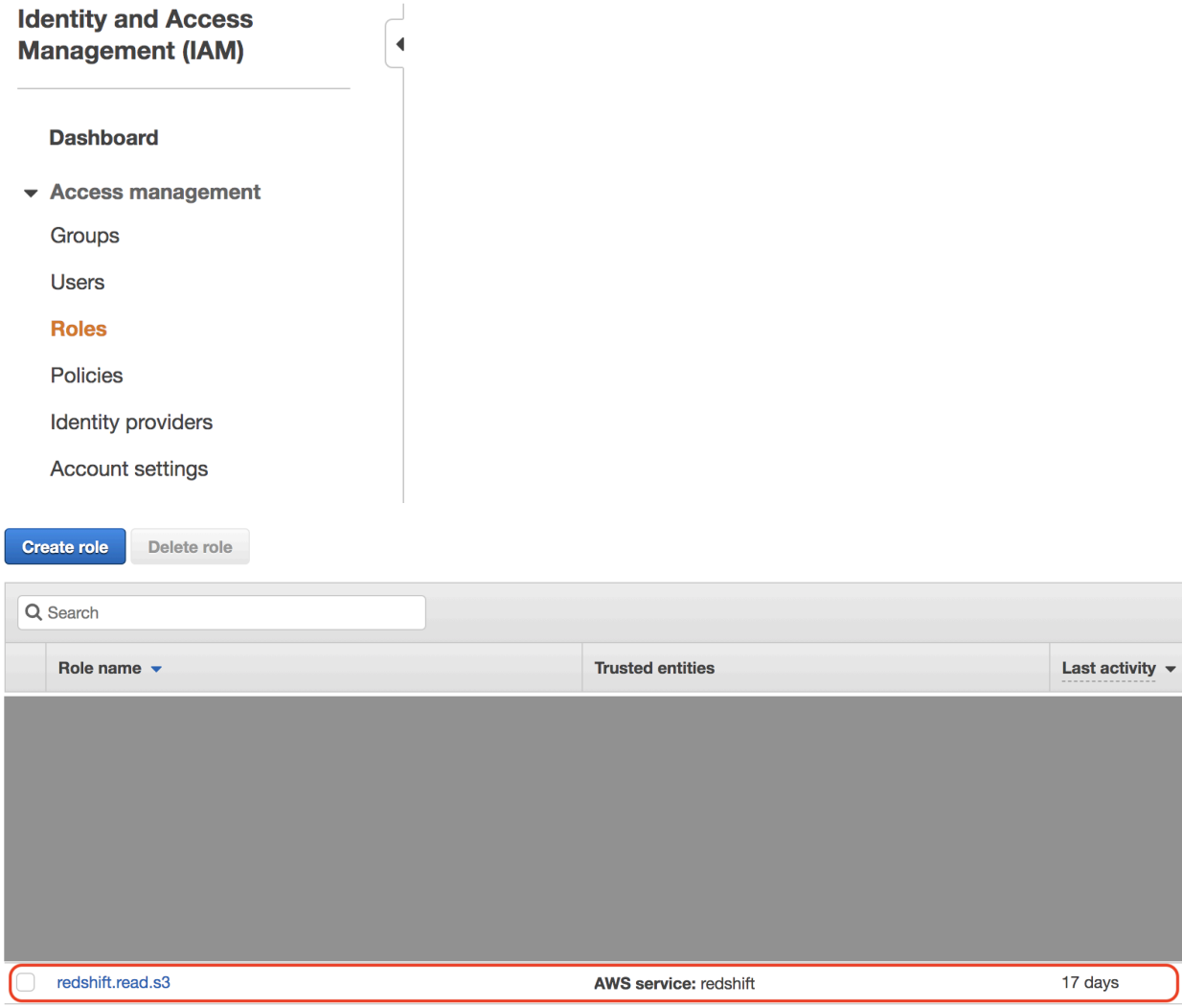


8. 여기서 앞서 만든 redshift.read.s3 권한을 지정하고 “Add IAM role” 버튼을 클릭한다.

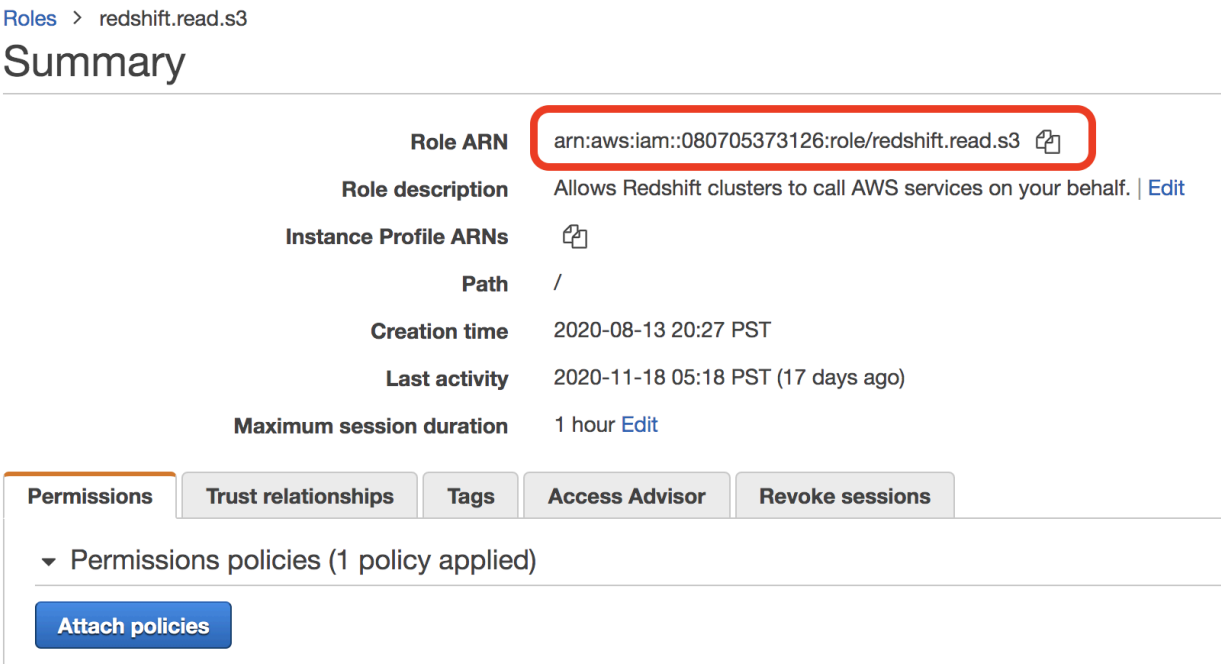


9. 이제 Redshift는 S3 버켓의 액세스가 가능해졌는데 이번 권한의 ID를 COPY 명령을 실행할때 인자 중의 하나로 주어야 한다. IAM

콘솔(https://console.aws.amazon.com/iam/home)로 돌아가 Roles 링크를 다시 클릭하고 거기서 redshift.read.s3 role을 선택한다.



redshift.read.s3 Summary 화면에서 Role ARN의 값을 복사하여 기억해둔다. 이 값을 최종 COPY 커맨드에서 사용해야한다.



7. COPY 명령을 실행하여 S3의 CSV 파일들을 각각 Redshift 테이블들로 복사.

   a. 먼저 Redshift raw_data 스키마에 아래와 같이 4개의 테이블을 생성한다

```
CREATE TABLE raw_data.channel (
    channelname character varying(32)
);
CREATE TABLE raw_data.session_timestamp (
    sessionid character varying(32),
    ts timestamp without time zone
);
CREATE TABLE raw_data.session_transaction (
    sessionid character varying(32),
    refunded boolean,
    amount integer
);
CREATE TABLE raw_data.user_session_channel (
    userid integer,
```

```
    sessionid character varying(32),
    channel character varying(32)
);
```

b. 다음으로 Redshift SQL 클라이언트에서 아래 4개의 COPY SQL을 생성한다:
   i. CSV 파일이기에 delimiter로는 콤마(,)를 지정한다. 그리고 CSV 파일에서 문자열이 따옴표로 둘러싸인 경우 이를 제거하기 위해 removequotes를 지정한다.
   ii. 그리고 CSV 파일의 첫번째 라인(헤더)을 무시하기 위해 “IGNOREHEADER 1”을 지정해준다
   iii. COPY SQL의 문법은 https://docs.aws.amazon.com/redshift/latest/dg/r_COPY.html

```
COPY raw_data.channel
FROM 's3://abc-keeyong/channel.csv'
credentials 'aws_iam_role=arn:aws:iam::080705373126:role/redshift.read.s3' delimiter ','
IGNOREHEADER 1 removequotes;


COPY raw_data.session_timestamp
FROM 's3://abc-keeyong/session_timestamp.csv'
credentials 'aws_iam_role=arn:aws:iam::080705373126:role/redshift.read.s3'
delimiter ',' dateformat 'auto' timeformat 'auto' IGNOREHEADER 1
removequotes;

COPY raw_data.session_transaction
FROM 's3://abc-keeyong/session_transaction.csv'
credentials 'aws_iam_role=arn:aws:iam::080705373126:role/redshift.read.s3'
delimiter ',' dateformat 'auto' timeformat 'auto' IGNOREHEADER 1
removequotes;

COPY raw_data.user_session_channel
FROM 's3://abc-keeyong/user_session_channel.csv'
credentials 'aws_iam_role=arn:aws:iam::080705373126:role/redshift.read.s3'
delimiter ',' dateformat 'auto' timeformat 'auto' IGNOREHEADER 1
removequotes;
```

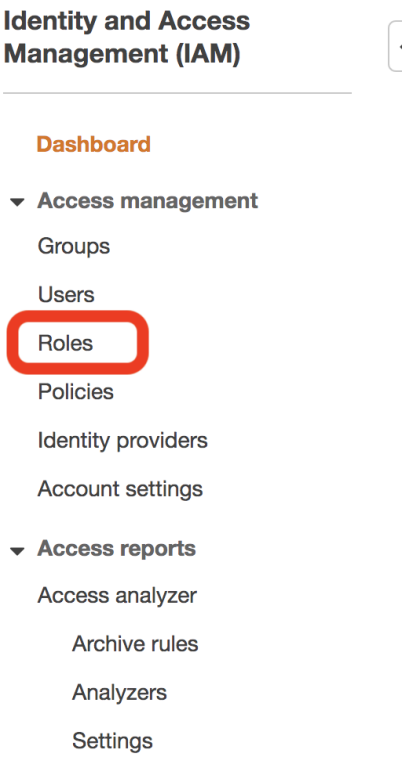
만일 COPY 명령 실행 중에 에러가 나면 stl_load_errors 테이블의 내용을 보고 확인한다:

```
SELECT * FROM stl_load_errors ORDER BY starttime DESC;
```

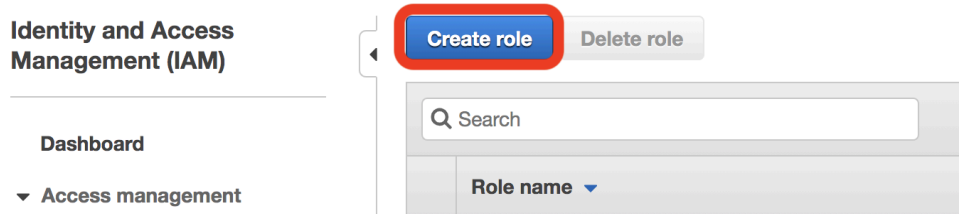
8. EC2 서버에 S3 접근 권한을 주기

Redshift와 직접적인 관련은 없지만 만일 Airflow를 EC2 서버에서 설치했다면 설정해야하는 IAM Role이 하나 더 있다. 이는 S3 접근 권한을 EC2 서버에 지정해주는 IAM Role로 이를 생성한 후 접근이 필요한 서버에 아래와 같이 명시적으로 권한을 부여해주어야 한다. 이는 Airflow가 S3를 문제없이 접근할 수 있게 함이다.

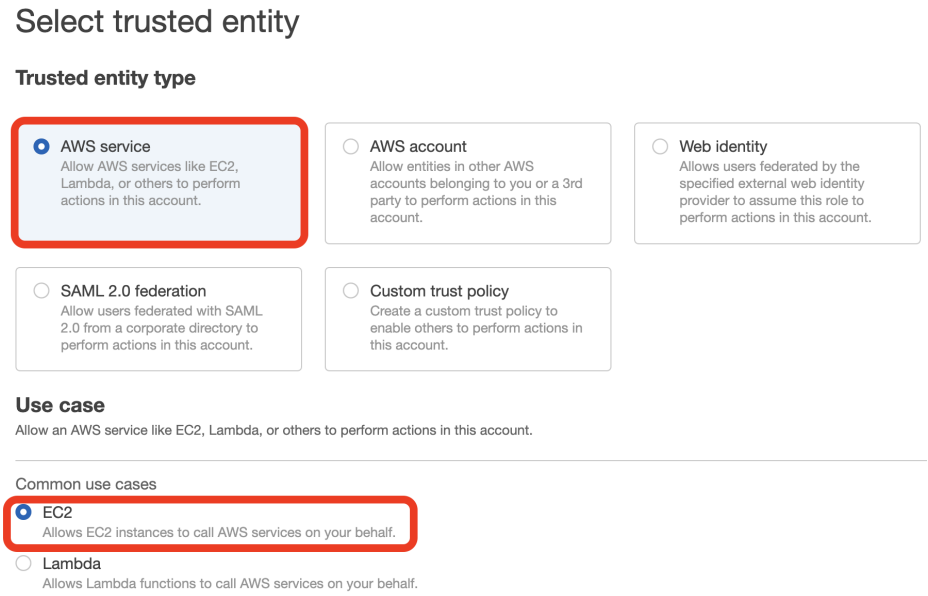
a. https://console.aws.amazon.com/iam/home을 방문한다.
b. 왼쪽 메뉴에서 Role을 선택한다



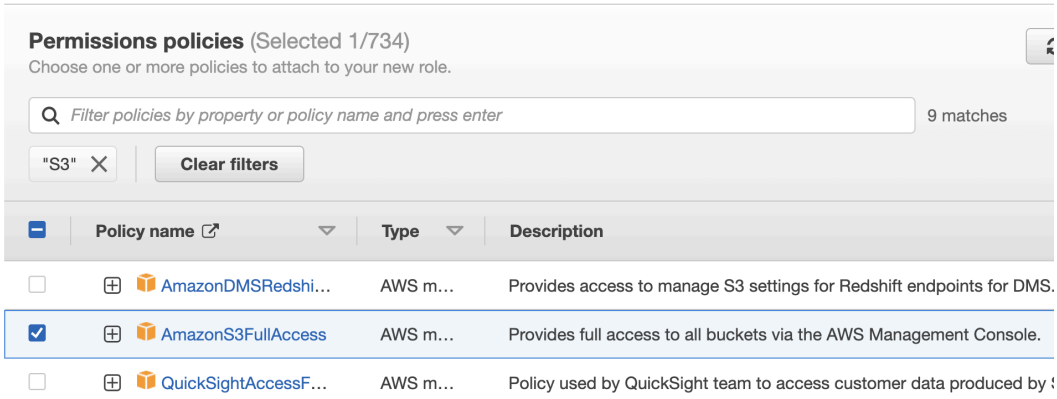
d. Create Role을 선택한다



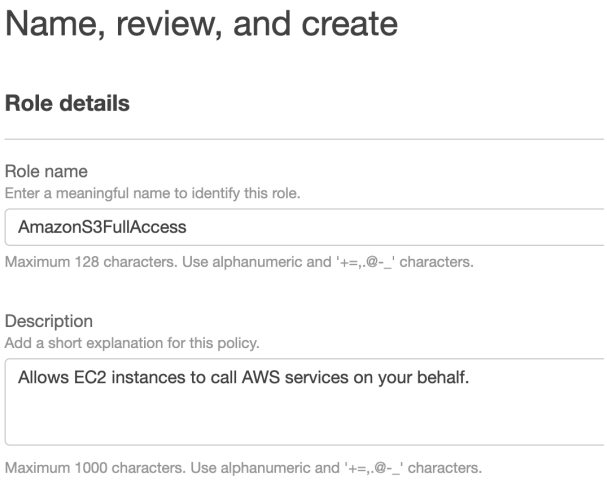
e. AWS service를 선택하고 아래 리스트에서 EC를 선택한다. Next를 누른다



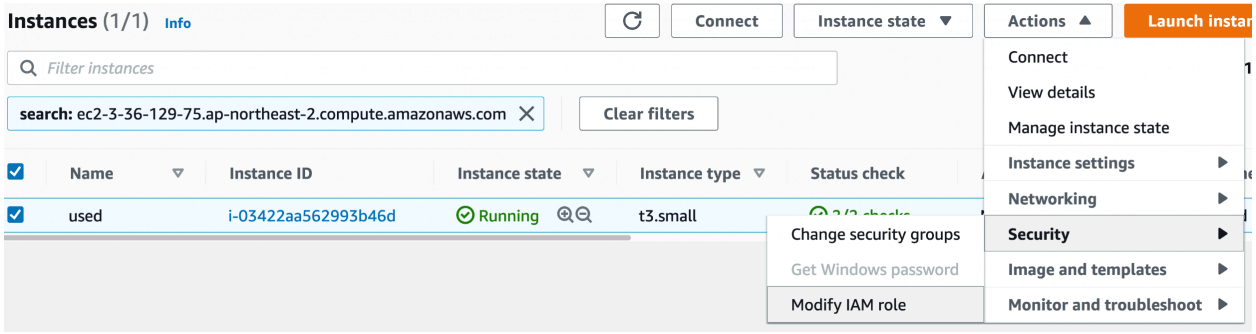
f. Add Permissions에서 S3를 검색하고 AmazonS3FullAccess를 선택하고 Next를 클릭한다



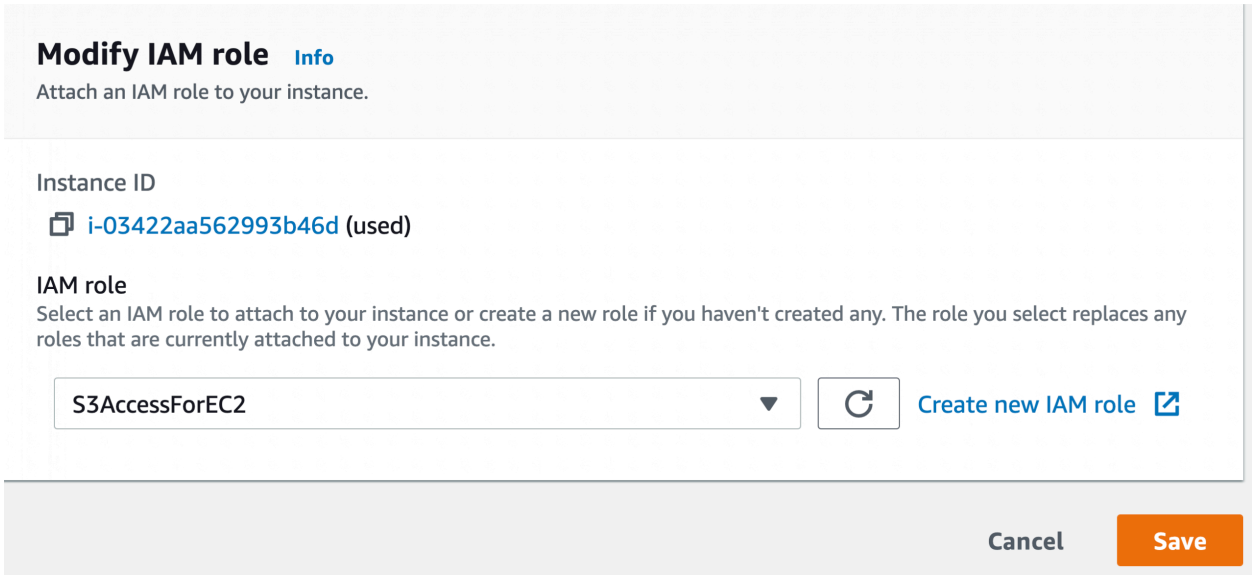
g. 마지막 페이지에서 Role 이름으로 **AmazonS3FullAccess**를 주고 “Create Role” 버튼을 클릭한다.



h. EC2 콘솔에서 이 접근 권한이 필요한 서버 선택하고 Actions 메뉴에서 Security 선택하고 최종적으로 Modify IAM role을 선택한다.



c. 여기서 해당 S3 접근 권한을 지정해주는 IAM Role(**AmazonS3FullAccess**)을 선택한다



9. EC2 서버가 아닌 환경에 S3 접근 권한을 주기

- 이는 보통 도커로 만들어진 개인컴퓨터등의 환경에 해당되는데 이 경우 가장 편리한 방법은 S3 액세스 권한을 가진 IAM 사용자를 만들고 그 사용자의 Access Key와 Secret Access Key를 S3 Connection 설정을 할때 사용하는 방식이다. 이 경우 중요한 포인트는 해당 IAM 사용자가 불필요한 권한을 가지면 안된다는 점이다 (외부로 노출되었을 때 보안 이슈)

10. guest 유저 만들고 권한 주기

```
CREATE USER guest CREATEUSER PASSWORD 'Guest1234' ;
```

```
GRANT USAGE ON SCHEMA raw_data TO guest;
GRANT SELECT ON ALL TABLES IN SCHEMA raw_data TO guest;
GRANT USAGE ON SCHEMA analytics TO guest;
GRANT SELECT ON ALL TABLES IN SCHEMA analytics TO guest;
```