Lab2. Using AWS Glue Workflows

1. 이전 Lab에서 생성했던 AWS Glue Table 삭제
   1)[서비스] > [분석] > [AWS Glue]
   2)[Data Catalog] > [Databases] > [Tables]
   3)모든 테이블 선택 > [Delete] 버튼 클릭
   4)[Delete tables] 팝업창에서 "Delete" 입력 후 [Delete] 버튼 클릭

2. AWS Glue Service를 사용하기 위한 Role 생성하기
   1)[서비스] > [보안, 자격 증명 및 규정 준수] > [IAM]
   2)좌측 메뉴의 [역할]을 클릭하여 해당 페이지로 이동
   3)[역할] 페이지에서
      -[역할 만들기] 클릭
      -[신뢰할 수 있는 엔터티 선택] 페이지에서,
         --[신뢰할 수 있는 엔터티 유형] : [AWS 서비스]
         --[사용 사례]
            ---[다른 AWS 서비스의 사용 사례]:"Glue"로 검색하여 [Glue] 선택
            ---[다음] 클릭
      -[권한 추가] 페이지에서
         --[권한 정책]에서 검색을 "AWSGlueServiceRole"로 검색하여 [AWSGlueServiceRole] 체크
         --[다음] 클릭
      -[이름 지정, 검토 및 생성] 페이지에서,
         --[역할 세부 정보] > [역할 이름] : {계정}-glue-role
         --[역할 생성] 클릭

3. AWS Glue Studio를 사용한 Job 생성하기
   1)[서비스] > [분석] > [AWS Glue]
   2)[Data Integration and ETL] > [AWS Glue Studio] > [Jobs]
   3)[Create job] 섹션에서
      -[Spark script editor] 선택
      -[Options] > [Create a new script with boilerplate code] 선택
      -[Create] click

   4)현재 Job의 이름이 "Untitled job"이어서 이름을 변경한다. --> {계정}-glue-job
   5)[Job details] 탭으로 이동
   6)[IAM Role] : 위에서 생성한 "{계정}-glue-role"을 목록에서 선택
   7)[Type] : 기본값 그대로 "Spark"
   8)[Glue version] : 기본값 그대로 "Glue 3.0 - Supports spark 3.1, Scala 2, Python 3"
   9)[Language] : 기본값 그대로 "Python 3"
   10)[Worker type] : 기본값 그대로 "G 1X"
   11)[Requested number of workers] : "3"으로 수정
   12)페이지 상단의 [Save] 버튼 클릭
   13)"Successfully updated job" 메시지가 나오면 [Run]을 클릭한다.
   14)[Runs] 탭에서 [Run status]가 "Running" -> "Succeeded" 확인

4. AWS Glue Workflows 생성하기
   1)[AWS Glue] > [Data Integration and ETL] > [Workflows(orchestration)] 이동
   2)[Workflows]페이지에서
      -[Add workflow] 클릭
      -[Add a new ETL workflow] 페이지에서,
         --[Workflow name] : {계정}-workflow
         --[Create workflow] 클릭

   3)새로 생성한 "{계정}-workflow" 선택
   4)페이지 아래 섹션에서 [Add trigger] 클릭
   5)[Add trigger] 팝업창에서,
      -[Add new] 탭 선택
      -[Name] : {계정}-start-trigger
      -[Trigger type] : "On demand"
      -[Add] 버튼 클릭

   6)"Incomplete" 즉 불완전 Trigger가 생성된다. 그 이유는 연결할 Job이 없기 때문이다.
   7)"Add node"를 클릭한다.
   8)[Add job(s) and crawlers(s) to trigger] 팝업창에서,
      -[Jobs] 탭 선택
      -[Name]이 위에서 생성한 "{계정}-glue-job"을 체크하고 [Add] 버튼 클릭

   9)방금 생성한 "{계정}-glue-job" 선택후, "Add trigger" 클릭
   10)[Add trigger] 팝업창에서,
      -[Add new] 탭
      -[Name] : {계정}-next-trigger
      -[Trigger type] : "Event"
      -[Add] 버튼 클릭

   11)새로 생성한 "{계정}-next-trigger"를 선택한 후, 이 트리거가 성공후 생성할 작업 즉, 크롤러를 추가하기로 한다.
   12)"Add node"를 클릭한다.
   13)[Add job(s) and crawlers(s) to trigger] 팝업창에서,
      -[Crawlers] 탭 선택
      -[Name]에서 "{계정}-crawler"를 선택 후 [Add] 클릭

5. Workflows 실행하기
   1)위에서 생성한 "{계정}-workflow" 페이지 상단에서 [Run workflow]을 클릭한다.
   2)[AWS Glue] > [Data Integration and ETL] > [AWS Glue Studio] > [Jobs] > "{계정}-glue-job"의 상세 페이지로 이동하여
   3)[Runs] 탭에서
      -[Run status]가 "Running"으로 확인됨. 끝나면 "Succeeded"로 변경됨.(보이지 않으면 Refresh 버튼 클릭)
      -[Trigger name]이 "{계정}-start-trigger"로 확인됨.

   4)[AWS Glue] > [Data Catalog] > [Crawlers] > "{계정}-crawler"로 이동
   5)앞의 Lab처럼 Run이 끝나면 [State]는 "Running" -> "Stopping" -> "Ready"로 변경되고, [Last run]은 "Succeeded"로, [Table changes from last run]은 "1 created"로 변경된다.
   6)모두 Running이 마치면, [Data Catalog] > [Databases] > [Tables] > "{계정}-output"로 이동
      -[View data] > "Table data" 링크 클릭
      -[View data] 팝업창에서 [Proceed] 버튼 클릭
      -앞의 Lab과 동일한 결과 확인