|  |
| --- |
| **[600003] 클라우드컴퓨팅** |
| **실습 #13 문제 및 보고서** |

|  |  |
| --- | --- |
| **이름** | 곽영주 |
| **학번** | 20175105 |
| **소속**  **학과/대학** | 빅데이터 |
| **분반** | 01 (담당교수: 김태운) |

## <주의사항>

* 개별 과제 입니다. (팀으로 진행하는 과제가 아니며, 모든 학생이 보고서를 제출해야 함)
* 파일명에 본인의 이름과 학번을 입력하세요.
* 각각의 문제 바로 아래에 답을 작성 후 제출해 주세요.
  + 소스코드/스크립트 등을 작성 한 경우, 본 문서에 코드를 붙여 넣거나 또는 별도의 파일로 첨부해서 제출하세요. 별도의 파일로 제출하는 경우 해당 파일의 이름도 적어주세요.
* SmartLEAD 제출 데드라인:
  + 화요일 분반: 다음 다음 실습 시간 전날까지(월요일 까지) // 2주 과제
  + 목요일 분반: 다음 다음 실습 시간 전날까지 (수요일 까지) // 2주 과제
  + 데드라인을 지나서 제출하면 24시간 단위로 20%감점(5일 경과 시 0점)
  + 주말/휴일/학교행사 등 모든 날짜 카운트 함
  + 부정행위 적발 시, 원본(보여준 사람)과 복사본(베낀 사람) 모두 0점 처리함
  + 예외 없음
* SmartLEAD 에 아래의 파일을 제출 해 주세요
  + 보고서(PDF 파일로 변환 후 제출을 권장하나. 워드 문서 제출도 가능)
  + 보고서 파일명에 이름과 학번을 입력 해 주세요.
  + 소스코드, 스크립트, Makefile 등을 작성해야 하는 경우, 작성한 모든 파일을 본 문서에 붙여 넣기 하거나 또는 하나의 첨부파일(zip 압축파일)로 제출

## <개요>

이번 과제는 Amazon 의 클라우드 컴퓨팅 서비스인 AWS(Amazon Web Services)를 사용하는 내용입니다.

## <실습 과제: 기본>

|  |
| --- |
| **[Q 0] 요약 [10점]**  이번 과제에서 배운 내용 또는 과제 완성을 위해서 무엇을 했는지를 3문장 이상으로 요약하세요. |

답변: 가이드 없이 이번 과제를 하면서 헷갈렸던 부분들을 다시 복습할 수 있었습니다.

또한, recognition에 다양한 서비스를 사용해볼 수 있는 좋은 기회였습니다.

Boto3 api 문서가 과제를 완성하는데, 큰 도움이 되었고, 마지막 문제는 구글과 유튜브를 찾아보면서 사용법을 익혔습니다.

이번 과제에서는 Python용 AWS SDK(Software Development Kit)인 boto3를 사용합니다. SDK를 사용하면 프로그래밍을 이용하여 손쉽게 AWS 제어/관리가 가능합니다. Python 언어를 사용할 수 있는 학생은 아래의 과제를 그대로 수행하면 되고, Python 언어를 사용할 수 없는 학생은 다른 언어로 제공되는 SDK를 사용하여 아래의 과제를 수행해도 됩니다.

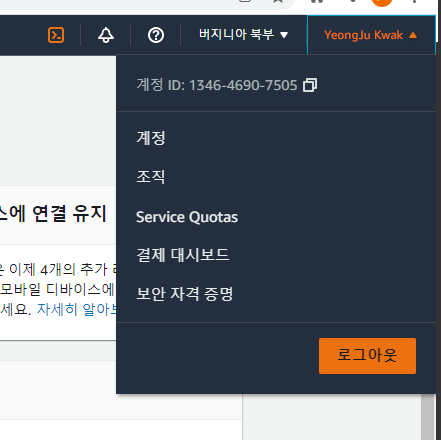
* 프로그래밍 언어별 AWS SDK 찾아보기: <https://aws.amazon.com/ko/tools/>
* AWS SDK지원 언어: C++, Go, Java, JavaScript, .NET, Node.js, PHP, Python, Ruby
* 'boto3’
  + API 문서: <https://boto3.amazonaws.com/v1/documentation/api/latest/index.html>
  + Github repo: <https://github.com/boto/boto3>
* 과제 완료 후, 모든 리소스를 해제/반납하고 청구 금액을 반드시 확인하세요

|  |
| --- |
| **[Q 1] Amazon AWS SDK boto3 사용하기 [10점] ... Python을 이용하는 경우**  이번 과제에서는 Python용 AWS SDK인 boto3를 사용하여, 프로그래밍을 이용한 AWS 제어/관리에 대해 알아보겠습니다.  다음 문서를 참고하여 과제를 진행하세요:  <https://github.com/overegoz/cloud-computing/blob/main/W13_Q1_boto3.pdf>  **<boto3 설치하기>**  <https://aws.amazon.com/ko/sdk-for-python/> 페이지를 참고하여 Boto3를 설치하세요. 설치 시, 문제가 생기면 Boto3 Docs 의 'Quickstart’ 섹션에서 ‘Installation’을 참고하세요.  <https://boto3.amazonaws.com/v1/documentation/api/latest/guide/quickstart.html>  **<Boto3 설치 확인 및 간단한 예제 실행 1: 계정 정보 조회>**  Boto3 가 정상적으로 설치 되었는지를 확인하기 위해, 아래와 같이 간단한 예제를 실행합니다. 아래의 예제가 정상적으로 실행되면, 숫자로 표시된 사용자 계정 ID가 출력됩니다.    **[과제 1]** 위의 예제를 실행하고, 터미널 화면을 캡처해서 첨부하세요. 또한, AWS Console 웹 페이지에서 우측 상단의 계정을 클릭하고 팝업되는 화면에서 본인의 계정 아이디를 캡처해서 첨부하세요. (두 개의 첨부 화면 예시는 [예시 화면 1]을 참고하세요)  **<Boto3 설치 확인 및 간단한 예제 실행 2: S3 버킷 조회>**  Boto3 가 정상적으로 설치 되었는지를 확인하기 위해, 아래와 같이 간단한 예제를 실행합니다. 아래의 예제가 정상적으로 실행되면, 본인 계정에서 생성한 S3 버킷 목록이 표시됩니다. 참고로, 본인 계정에서 생성한 S3 버킷이 없는 경우 아무런 목록이 출력되지 않을 수도 있습니다.    **[과제 2]** 위의 예제를 실행하고, 터미널 화면을 캡처해서 첨부하세요. 또한, AWS Console 웹 페이지에서 ‘Amazon S3’ 페이지로 이동하여 현재 계정에서 생성한 S3 버킷 목록을 캡처해서 첨부하세요. (두 개의 첨부 화면 예시는 [예시 화면 2]를 참고하세요). |

답변 1 (계정 정보 조회):

텍스트이(가) 표시된 사진

자동 생성된 설명



답변 2 (S3 버킷 목록 조회):

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

|  |
| --- |
| **[Q 2] SDK를 이용한 EC2 가상 머신 관리 [20점]**  이번 문제는 Free Tier에서 무료로 사용할 수 있는 Ubuntu 18.04 + t2.micro 인스턴스를 사용합니다. Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0279c3b3186e54acd (64비트 x86) 템플릿을 t2.micro 유형으로 사용하면 무료로 EC2 인스턴스를 생성할 수 있습니다.    이번 문제는 별도의 안내문서가 제공되지 않습니다. <https://boto3.amazonaws.com/v1/documentation/api/latest/index.html> 에서 필요한 API 함수를 검색해서 사용하세요 (+ 구글 검색 찬스를 활용하세요!).  **[과제 1]** EC2 인스턴스 ID와 인스턴스의 상태를 조회하고 터미널에 출력하는 코드를 작성하세요. 코드를 실행하면, 아래와 같이 터미널 출력을 해야 합니다. 아래의 화면은 2개의 EC2 인스턴스가 생성된 상태에서의 출력이며, 본인 계정에 생성된 EC2 인스턴스 수에 따라 출력 개수가 달라질 수 있습니다. 본인 계정에 생성된 EC2 인스턴스가 없는 경우, AWS 웹 Console 에서 Ubuntu Server 18.04 LTS (HVM), SSD Volume Type (타입은 t2.micro로 설정) 이미지로 가상 머신을 하나 또는 그 이상 만들고, 코드를 다시 실행하세요.    코드를 캡처하여 아래에 첨부하고, 실행 결과 터미널 화면도 캡처하여 아래에 첨부하세요.  **[과제 2]** EC2 인스턴스를 생성하는 코드를 작성하세요. 인스턴스 생성 시, 이미지 ID는 ‘ami-0279c3b3186e54acd’로, 인스턴스 Type은 't2.micro'로 설정하세요. 코드를 실행하고, 코드를 캡처하여 아래에 첨부하세요. 이번 코드는 아무런 터미널 출력을 하지 않아도 됩니다.  **[과제 3]** 위의 [과제 2]를 실행한 후, [과제 1]에서 작성한 코드를 다시 실행하세요. 터미널 출력 화면을 캡처하여 아래에 첨부하세요.  **[과제 4]** 현재 running 상태인 EC2 인스턴스 중 하나를 정지(stop)하는 코드를 작성하세요. 코드를 실행 후, [과제 1]에서 작성한 코드를 실행하여 EC2 인스턴스의 상태를 조회하세요. 아래와 같이 하나의 EC2 인스턴스의 상태가 stopped 인 것으로 나와야 합니다. 코드와 터미널 출력을 아래에 첨부하세요    **[과제 5]** 현재 stopped 상태인 EC2 인스턴스를 삭제 (terminate)하는 코드를 작성하세요. 코드를 실행 후, [과제 1]에서 작성한 코드를 실행하여 EC2 인스턴스의 상태를 조회하세요. 아래와 같이 stopped 였던 EC2 인스턴스의 상태가 terminated 인 것으로 나와야 합니다. 코드와 터미널 출력을 아래에 첨부하세요.    \*\* [Q2] 문제가 끝났습니다. 사용하지 않는 EC2 인스턴스를 모두 제거/terminate 하세요. |

답변 1 (코드):

텍스트이(가) 표시된 사진

자동 생성된 설명

답변 1 (터미널 출력):

텍스트이(가) 표시된 사진

자동 생성된 설명

답변 2 (코드):

텍스트이(가) 표시된 사진

자동 생성된 설명

답변 3 (터미널 출력):

텍스트이(가) 표시된 사진

자동 생성된 설명

답변 4 (코드):

텍스트이(가) 표시된 사진

자동 생성된 설명

답변 4 (터미널 출력):

텍스트이(가) 표시된 사진

자동 생성된 설명

답변 5 (코드):

텍스트이(가) 표시된 사진

자동 생성된 설명

답변 5 (터미널 출력):

텍스트이(가) 표시된 사진

자동 생성된 설명

|  |
| --- |
| **[Q 3] SDK를 이용한 S3 버킷 관리 [20점]**  AWS Free Tier를 사용하면 5GB까지 S3버킷을 무료로 사용할 수 있습니다.  이번 문제는 별도의 안내문서가 제공되지 않습니다. <https://boto3.amazonaws.com/v1/documentation/api/latest/index.html> 에서 필요한 API 함수를 검색해서 사용하세요 (+ 구글 검색 찬스를 활용하세요!).  [문제 1] 위의 [Q 1]에서 사용한 버킷 목록 조회 코드를 실행하고, 현재 계정에 있는 버킷 목록을 터미널에 출력하고 캡처하여 아래에 첨부하세요.  [문제 2] SDK 를 사용하여 새로운 S3 버킷을 만드는 코드를 작성하세요 (코드를 캡처하여 아래에 첨부). 작성한 코드를 실행해서 버킷을 생성하고, [문제 1]에서 사용한 코드를 실행해서 버킷 목록을 터미널에 출력하고 캡처하여 아래에 첨부하세요.  [문제 3] ‘hello-world.txt’라는 이름의 파일을 만들고, 파일에는 “Hello world.”라는 문자열을 입력합니다. AWS 웹 Console을 사용해서 [문제 2번]에서 생성한 버킷에 ‘hello-world.txt’ 파일을 업로드 하세요. SDK를 사용해서 해당 버킷 내에 저장된 모든 파일명을 출력하는 코드를 작성하세요 (코드를 캡처하여 아래에 첨부). 코드를 실행하고 터미널 화면을 캡처하여 아래에 첨부하세요. 아래와 같은 형식으로 출력하면 됩니다.    [문제 4] SDK를 사용해서, 로컬 PC에 저장된 파일을 S3 버킷에 업로드 하는 코드를 작성하세요 (코드를 캡처하여 아래에 첨부). 아래의 두 개의 파일을 로컬 PC에 저장하고, S3에 업로드 한 후, [문제 3]에서 작성한 코드를 실행해서, S3에 저장된 파일 목록을 출력하세요 (터미널 출력 화면 캡처 후 첨부)  (사진 1) <https://github.com/overegoz/cloud-computing/blob/main/actor.jpg>  (사진 2) <https://github.com/overegoz/cloud-computing/blob/main/actress.jpg>  아래와 같은 형식으로 출력하면 됩니다.    \*\* 여기서 생성한 버킷과, 업로드한 사진은 다음 문제에서 사용합니다. 아직 삭제하지 마세요. |

답변 1 (현재 버킷 목록 조회):

텍스트이(가) 표시된 사진

자동 생성된 설명

답변 2 (버킷 생성 코드):

텍스트이(가) 표시된 사진

자동 생성된 설명

답변 2 (현재 버킷 목록 조회):

텍스트, 장치, 측정기이(가) 표시된 사진

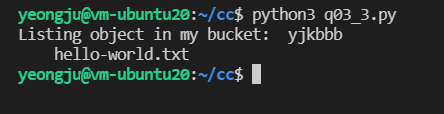
자동 생성된 설명

답변 3 (버킷에 저장된 파일 목록 조회 코드):

텍스트이(가) 표시된 사진

자동 생성된 설명

답변 3 (버킷에 저장된 파일 목록 조회):



답변 4 (로컬 파일 업로드하는 코드):

텍스트이(가) 표시된 사진

자동 생성된 설명

답변 4 (버킷에 저장된 파일 목록 조회):

텍스트이(가) 표시된 사진

자동 생성된 설명

|  |
| --- |
| **[Q 4] Rekognition [20점]**  이번 문제는 별도의 안내문서가 제공되지 않습니다. <https://boto3.amazonaws.com/v1/documentation/api/latest/index.html> 및 <https://docs.aws.amazon.com/rekognition/latest/dg/get-started-exercise.html> 에서 필요한 API 함수를 검색해서 사용하세요 (+ 구글 검색 찬스를 활용하세요!).  AWS의 Rekognition은 딥 러닝 기반의 시각 분석 (= 이미지, 동영상 분석) 서비스 이고, 사진 및 동영상 파일을 손쉽게 분석할 수 있습니다. [Q3] 문제에서 S3 버킷에 업로드한 actor.jpg와 actress.jpg 파일을 사용하고, 필요시 추가 사진을 사용하세요. Amazon Rekognition은 Free Tier를 사용하는 경우 이미지 분석을 월별 5,000건 까지 무료로 사용할 수 있습니다.    [문제 1] S3 버킷에 저장된 사진으로 유명 인사 얼굴 탐지하기  SDK를 사용해서 S3 버킷에 업로드한 ‘actor.jpg’ 사진에서 유명인사를 검색하는 프로그램을 작성하세요 (코드를 캡처하여 아래에 첨부). 프로그램 실행 후, 탐지한 유명인사의 이름을 터미널에 출력해야 합니다 (터미널 출력 화면 캡처 후 첨부). 아래와 같은 형식으로 출력하면 됩니다.    [문제 2] 위의 [문제 1]과 같은 방식으로, S3 버킷에 업로드한 ‘actress.jpg’ 사진에서 유명인사를 탐지하고, 결과를 터미널에 출력하세요. 아래와 같은 형식으로 출력하면 됩니다. (터미널 출력 화면 캡처 후 첨부).    [문제 3] 로컬 PC에 저장된 사진으로 유명 인사 얼굴 탐지하기  위의 [문제 1]에서 작성한 코드를 수정하여, S3 버킷이 아닌, 로컬 PC에 저장된 사진에서 유명인사를 탐지하는 코드를 작성하세요 (코드를 캡처하여 아래에 첨부). 이때, 아래의 ko-actor.jpg 사진을 사용하세요.  <https://github.com/overegoz/cloud-computing/blob/main/ko-actor.jpg>  탐지한 결과를 터미널에 출력하세요. 아래와 같은 형식으로 출력하면 됩니다. (터미널 출력 화면 캡처 후 첨부).    [문제 4] 본인의 사진을 이용해서 [문제 3]을 반복하세요. 터미널 결과를 캡처하여 아래에 첨부하세요. 참고로, 제 사진으로 테스트 한 결과는 아래와 같습니다. Zhu Yilong...??    [문제 5] 이번 문제는 사진에서 텍스트를 검출하여 터미널에 출력하는 과제이며, S3에 업로드 된 사진을 사용해도 되고, 또는 로컬 PC에 저장된 사진을 사용해도 됩니다. 단, 사진은 다음의 사진을 사용해야 합니다:<https://github.com/overegoz/cloud-computing/blob/main/img-text.jpg>  사진에서 텍스트를 검출하는 코드를 작성하세요 (코드를 캡처하여 아래에 첨부). 탐지한 결과를 터미널에 출력하세요. 아래와 같은 형식으로 출력하면 됩니다. (터미널 출력 화면 캡처 후 첨부).    \*\* 버킷은 더 이상 사용하지 않습니다. 버킷 내의 모든 파일을 삭제하고, 버킷도 삭제하세요. |

답변 1 (유명인사 탐지 코드/S3):

텍스트이(가) 표시된 사진

자동 생성된 설명

답변 1 (터미널 출력):

텍스트이(가) 표시된 사진

자동 생성된 설명

답변 2 (터미널 출력):

텍스트이(가) 표시된 사진

자동 생성된 설명

답변 3 (유명인사 탐지 코드/로컬 파일):

텍스트이(가) 표시된 사진

자동 생성된 설명

답변 3 (터미널 출력):

텍스트이(가) 표시된 사진

자동 생성된 설명

답변 4 (터미널 출력):

텍스트이(가) 표시된 사진

자동 생성된 설명

답변 5 (텍스트 검출 코드):

텍스트이(가) 표시된 사진

자동 생성된 설명

답변 5 (터미널 출력):

텍스트이(가) 표시된 사진

자동 생성된 설명

|  |
| --- |
| **[Q 5] Rekognition + Polly [20점]**  이번 문제는 별도의 안내문서가 제공되지 않습니다. <https://boto3.amazonaws.com/v1/documentation/api/latest/index.html> 및 <https://docs.aws.amazon.com/polly/latest/dg/getting-started.html> 에서 필요한 API 함수를 검색해서 사용하세요 (+ 구글 검색 찬스를 활용하세요!).  Amazon Polly는 텍스트를 현실에 가까운 음성으로 변환하는 서비스 입니다. Rekognition을 사용해서 사진에서 텍스트를 추출하고, 그 결과를 Polly를 통해 음성으로 읽어주는 서비스를 개발합니다. AWS Free Tier를 사용할 경우, 매월 500만건의 음성 변환을 무료로 사용할 수 있습니다.    다음의 이미지를 다운받아 로컬 PC 또는 S3 버킷에 저장하세요.  <https://github.com/overegoz/cloud-computing/blob/main/alright.jpg>  [문제] ‘alright.jpg’ 이미지에서 텍스트를 추출하고, 추출한 텍스트를 Polly를 이용해 음성으로 변환한 뒤, 음성을 ‘speech.mp3’ 파일로 로컬 PC/노트북에 저장하는 코드를 작성하세요. 사진에서 텍스트를 추출할 때, 중복되어 추출되는 텍스트가 없도록 코딩하세요. 즉, 텍스트 추출 결과는 “everything's gonna be alright” 이어야 합니다. 같은 단어가 두 번 또는 그 이상 추출되면 안됩니다. 아래와 같이 출력하도록 프로그램을 작성하세요.    다음으로, 탐지된 텍스트 끝에 본인의 이름을 추가하세요:  detected\_sentence += ' Taewoon' # ‘Taewoon’ 대신 본인의 이름 사용  “탐지된 텍스트 + 본인의 이름”을 음성으로 변환하세요.  소스코드를 캡처하여 아래에 첨부하고, 터미널 출력을 캡처하여 아래에 첨부하세요. 또한, 생성되는 mp3 파일을 별도로 첨부하여 업로드 하세요.  결과 mp3 파일 예시: <https://github.com/overegoz/cloud-computing/blob/main/speech-taewoon.mp3>  모든 과제가 종료되었습니다. EC2 인스턴스, 버킷 등 이번 과제에서 생성/업로드한 모든 것을 삭제하세요. |

답변 (소스코드):

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

답변 (터미널 출력):

텍스트이(가) 표시된 사진

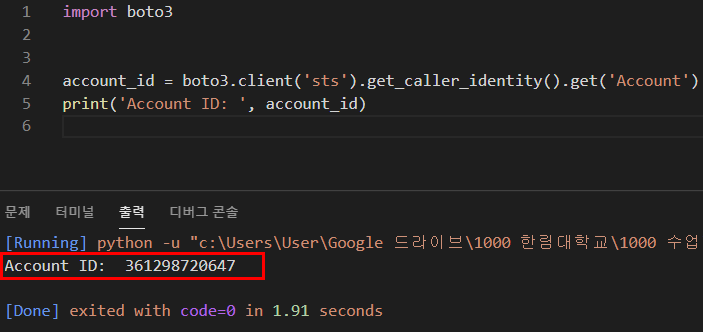
자동 생성된 설명

끝! 수고하셨습니다.

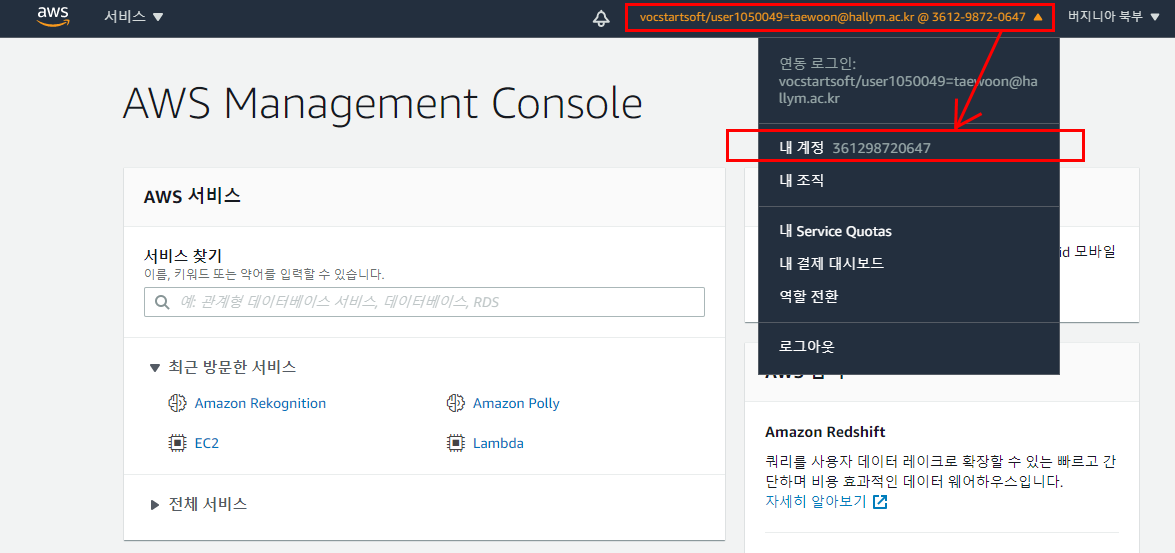
\*\* 더 이상 사용하지 않는 자원은 모두 중지 & 제거해 주세요. 과제 완료 후 [대금 및 비용 관리 대시보드]에서 청구된 요금을 확인하세요. 이번 과제는 모두 무료인 자원을 사용했으며, 과제 종료 후 아무런 요금이 청구되어 있지 않아야 합니다.

**[예시 화면 1] 사용자 계정 ID 조회**

1) boto3 SDK를 이용하여 계정 ID 조회하기

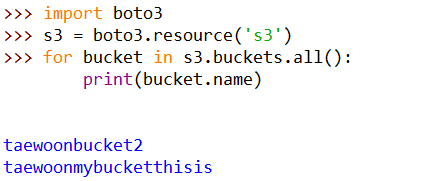


2) AWS Console에서 계정 ID 조회하기

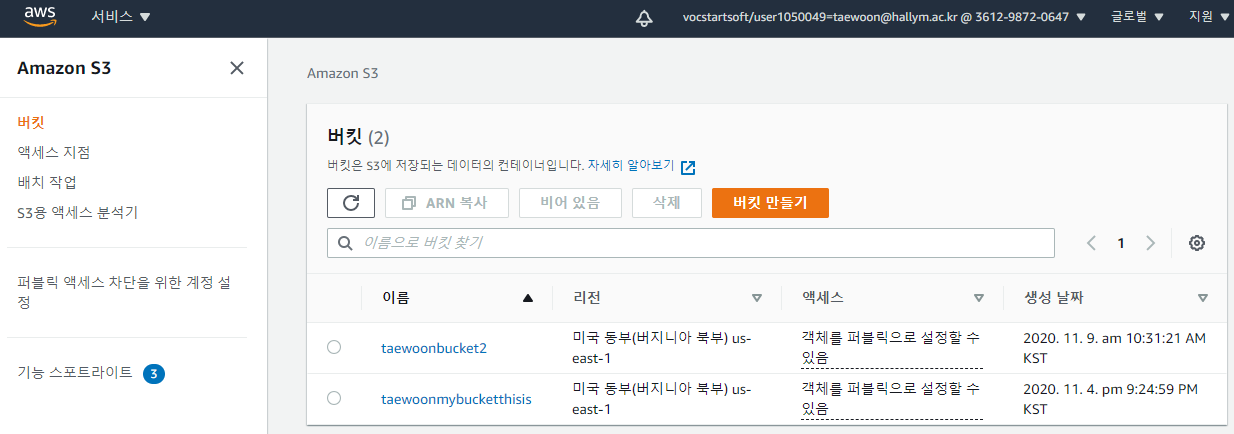


**[예시 화면 2] S3 버킷 목록 조회**

1) boto3 SDK를 이용하여 버킷 목록 조회하기: 버킷이 있는 경우 (참고로, 버킷이 없는 상태에서 조회하고, 그 결과를 캡처해도 됩니다)



2) AWS Console에서 버킷 목록 조회하기



**끝! 수고하셨습니다 ☺**