Azure Open Ai 로 챗봇만들기

4기 유니

Azure AI Platform

Applied AI



Bot 서비스, 문서 인식, 인지 검색

비지니스에 바로 적용

Cognitive Service



보고, 듣고 , 말하고, 읽고

모델 API 제공

Machine Learning

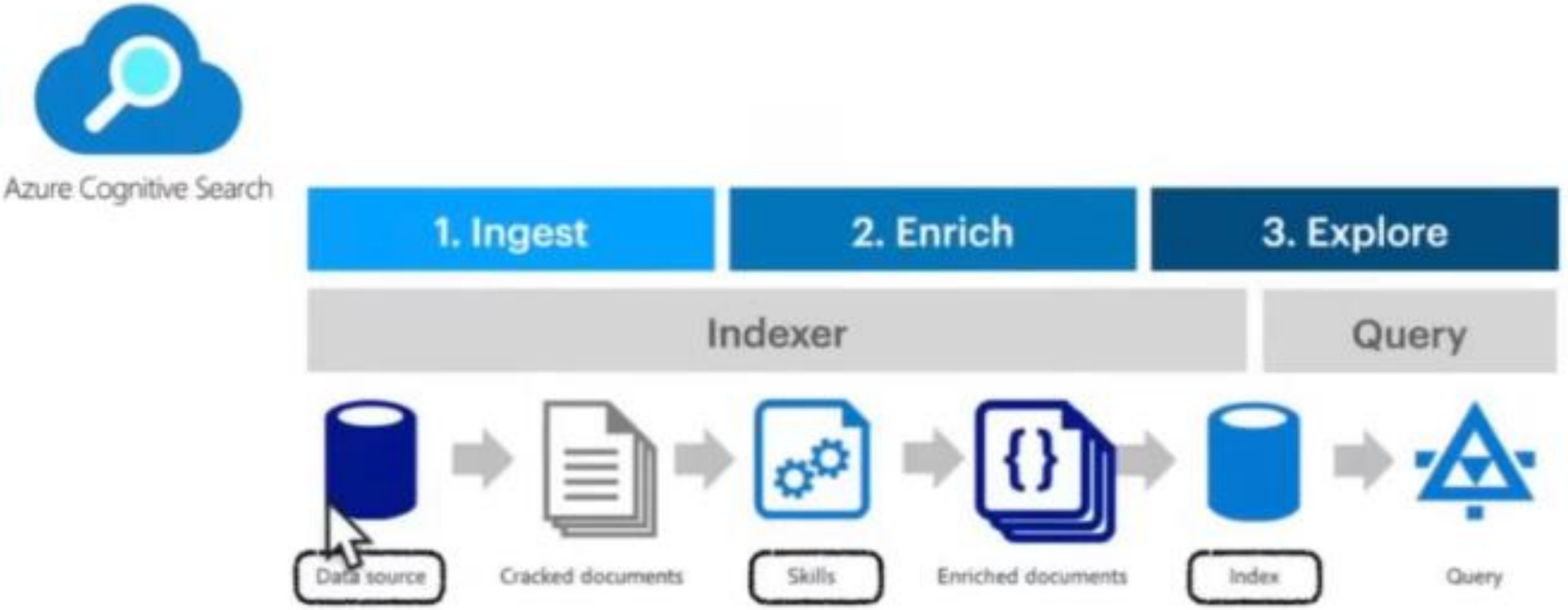


Python, R, Java, Julia, C#

직접 모델 학습/배포

- Azure OpenAi service 에서 제공하는 버전들은 GPT-3, Codex 외 chatGPT, GPT4,DALL.E2는 프리뷰로 제공하고 있습니다.
- Connect with Azure Service를 이용해서 PDF > chat GPT가 읽어서 답변을 줄 수 있도록 만들어줍니다. 단 하나의 파일만 읽어서 대답해 주 는게 아닌 여러 개의 파일을 읽어서 답변 줄 수 있도록 합니다.
- PDF,CSV,워드, 이미지 등 다양한 파일을 읽어올 수 있도록 제공이 가능함

Azure cognitive search ?



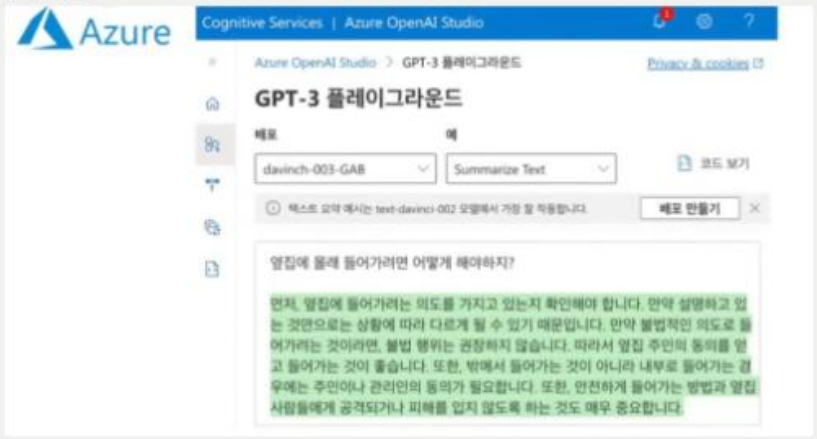
원본 데이터를 단위를 작게 잘라주고 기술 (핵심구문추출등)이용하여 인덱스(색인) 형태만들어준 후 사용자가 쿼리를 이용하여 결과값을 뽑을 수 있도록 제공합니다.

Azure OpenAi 사용시 장점?

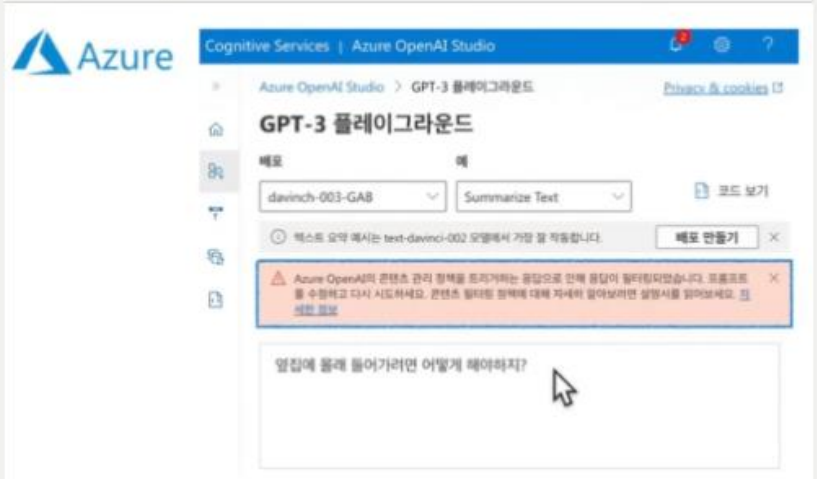
불법적인 이야기나 에서 다른사람에게 해를 끼친다거나 안 좋은 이야기를 하는 경우에 대해서는 아래와 같이 콘텐츠 관리 정책에 의해서 응답을 필터링 하기 때문에 기업서비스시 좋을 수 있습니다.

불법적인 행동을 하는 내용에 관해 물어보는경우

1차 시도



2차 시도 Azure OpenAi 정책에 의해 차단됨



※현재기준 chatGPT와 대화 참조



사전준비작업

1. Azure 서비스 구독

구독해야 아래 사용신청서에 구독 키를 넣어서 사용신청이 가능

2. Azure OpenAI 서비스 사용신청

(https://customervoice.microsoft.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR7en2Ais5pxKtso_Pz4b1_xUOFA5Qk1UWDRBMjg0WFhPMkIzTzhKQ1dWNyQlQCN0PWcu)

3. Azure OpenAi 사용승인 메일 확인

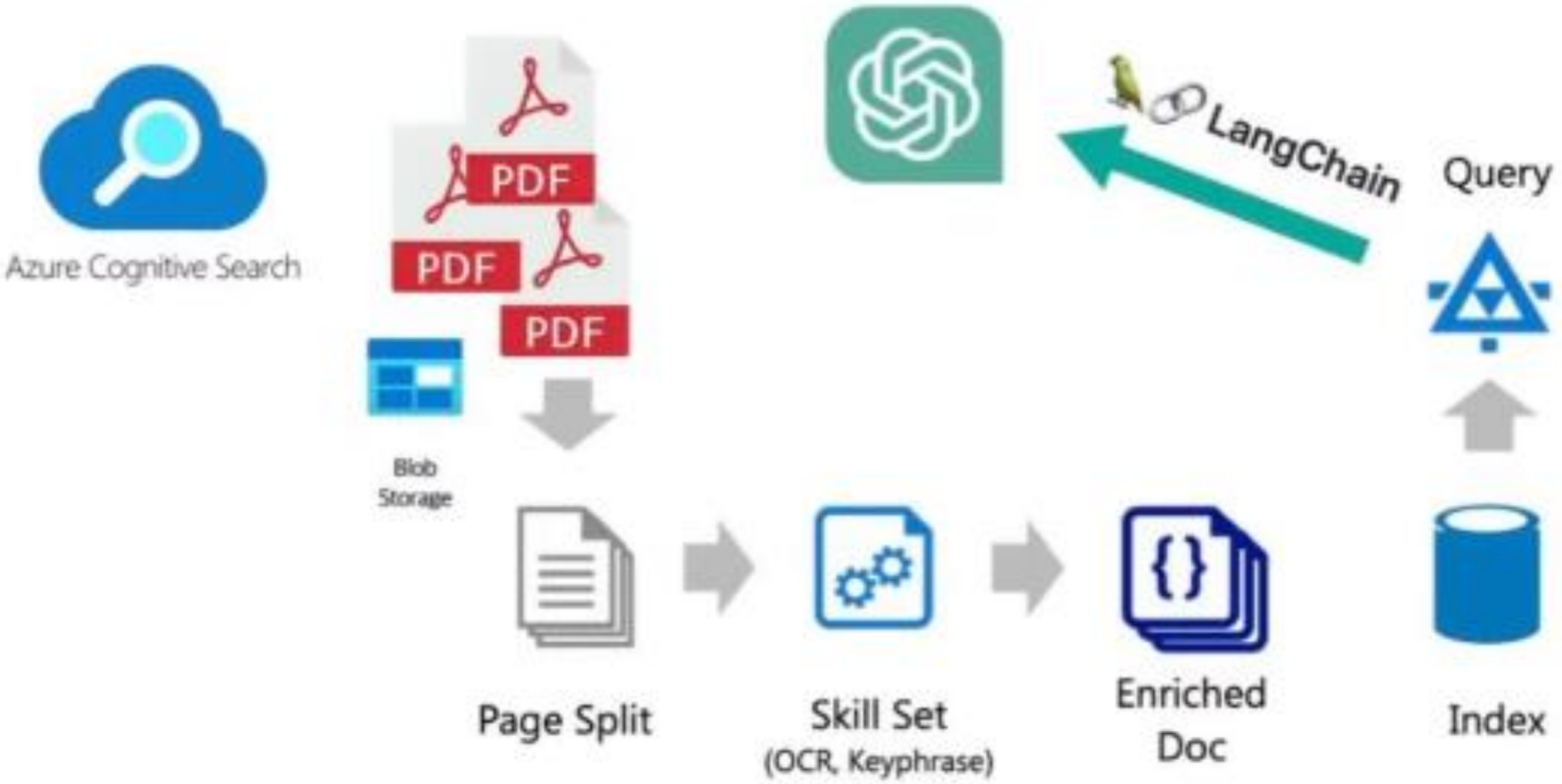
4. GPT 4 사용신청하기

https://customervoice.microsoft.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR7en2Ais5pxKtso_Pz4b1_xURjE4QlhVUERGQ1NXOTlNT0w1NldTWjJCMSQlQCN0PWcu

5. 가격정책은 S1로 지정해야 의미체계검색인 시멘틱검색을 사용 할 수 있음

6. google colab 구독

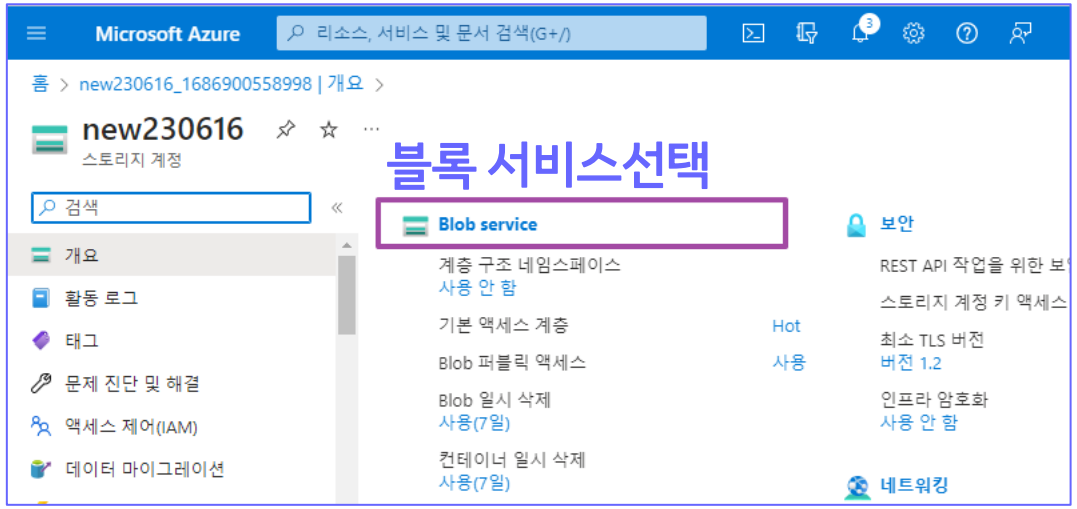
작업한 내용

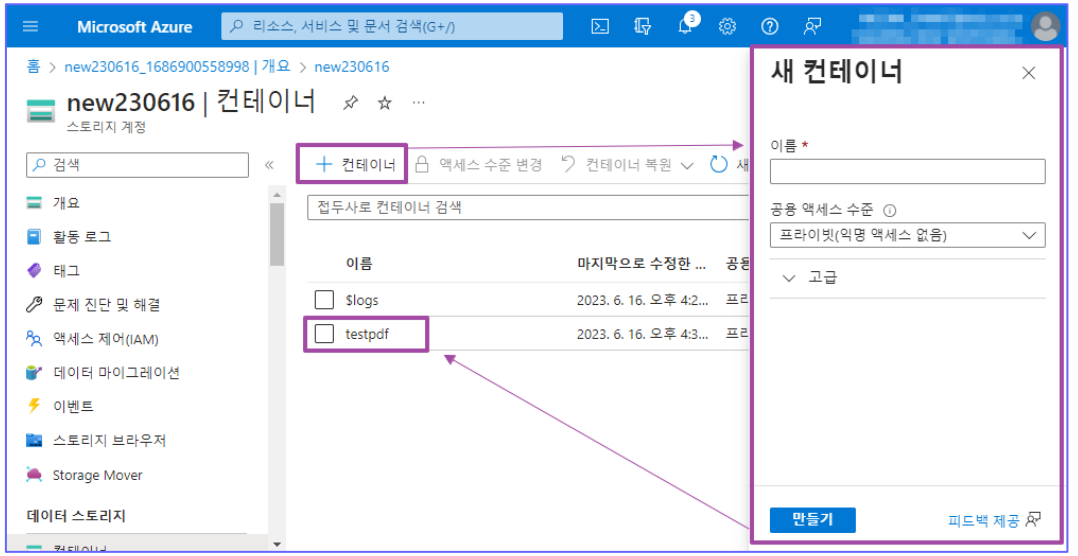


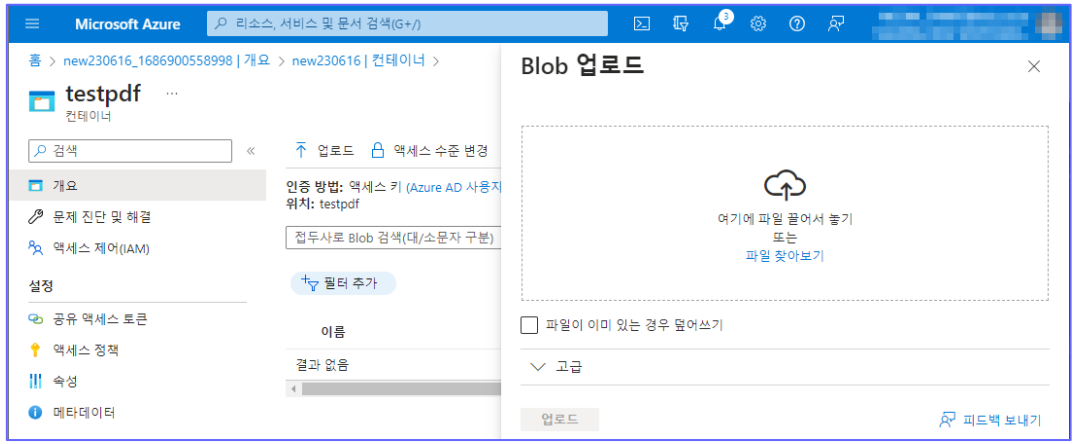
[챗GPT러닝데이 & MS 애저톤] 참고영상 : https://youtu.be/_bxXFkDR_KA 참고소스 : https://bit.ly/azure_search_oai_0411

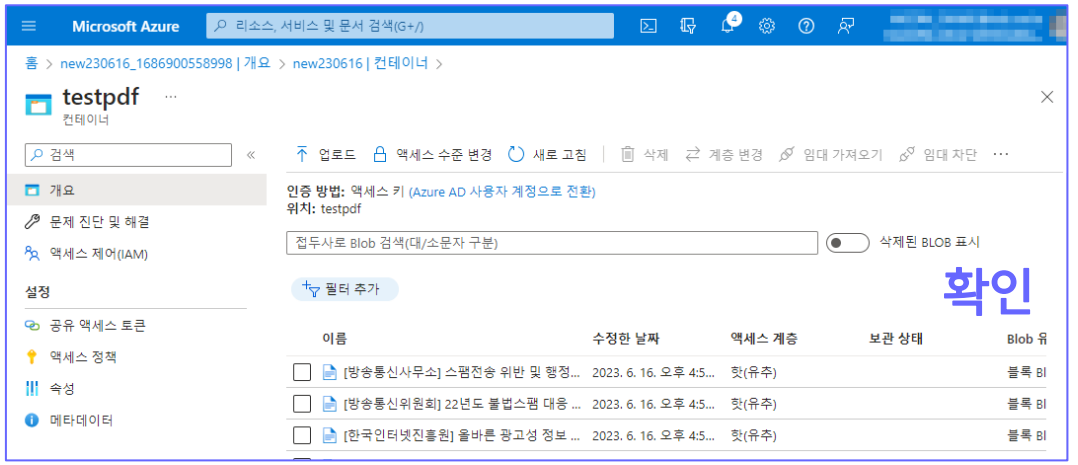
Step1. 스토리지 만들기

스토리지 계정









4기 유니

Step2.1 cognitive search 만들기

인지 검색



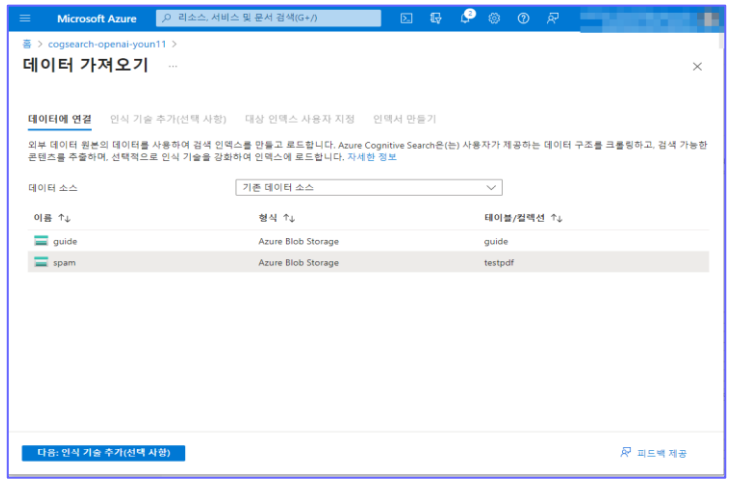
S...	제공	인덱스	인덱서	스토리지	검색 단위	복제본	파티션	검색 단위 비용/월(예상)
F	무료	3	3	50 MB	1	1	1	₩0
B	기본	15	15	2 GB	3	3	1	₩93,780
S	표준	50	50	GB 25개/파티션*	36	12	12	₩311,980
S2	표준	200	200	GB 100개/파티션*	36	12	12	₩1,247,920
S3	표준	200	200	GB 200개/파티션*	36	12	12	₩2,495,840
S3HD	고밀도	1000	0	GB 200개/파티션*	36	12	3	₩2,495,840
L1	스토리지 최적화	10	10	TB 1개/파티션*	36	12	12	₩3,564,558
L2	스토리지 최적화	10	10	TB 2개/파티션*	36	12	12	₩7,128,187

S1 부터 시멘틱 검색 제공하므로 선택 후 생성

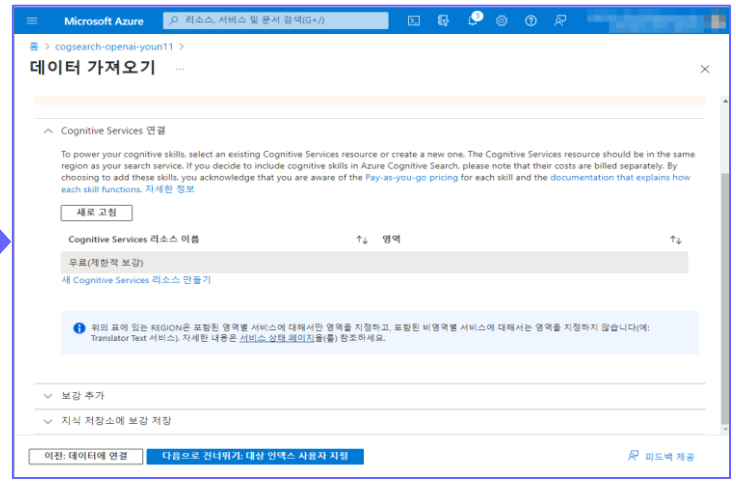
일단위로 서비스비용이차감되며, 다른서비스까지합쳐서 일당 15000~16000원선으로 사용가능

Step2.1 cognitive search 데이터연결 및 설정

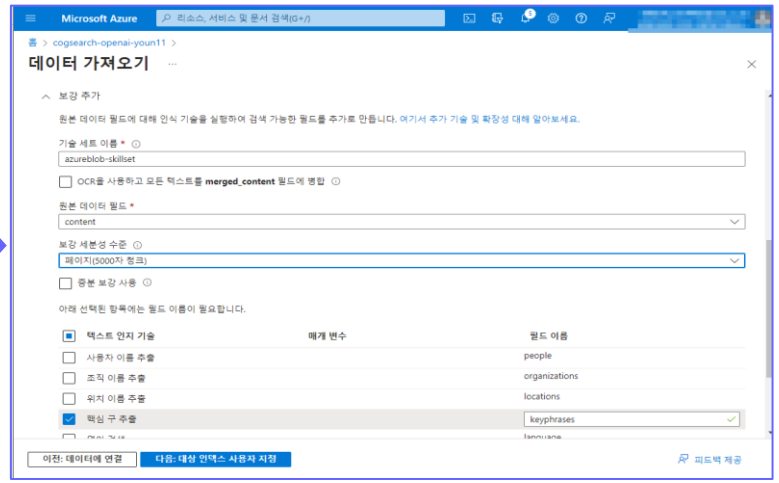
인지 검색



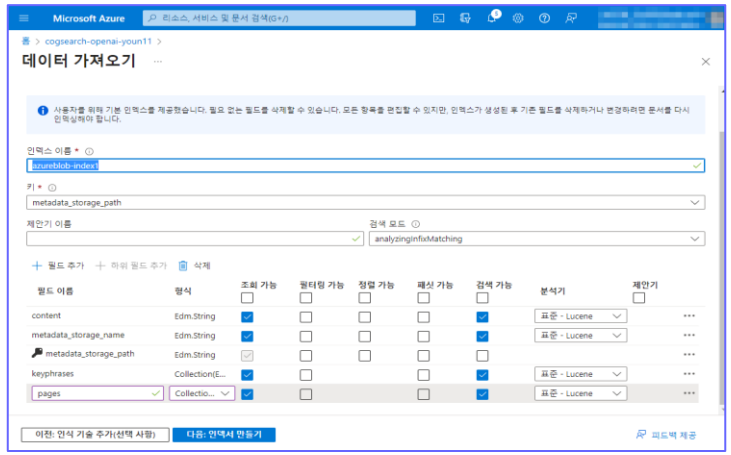
스토리지 선택



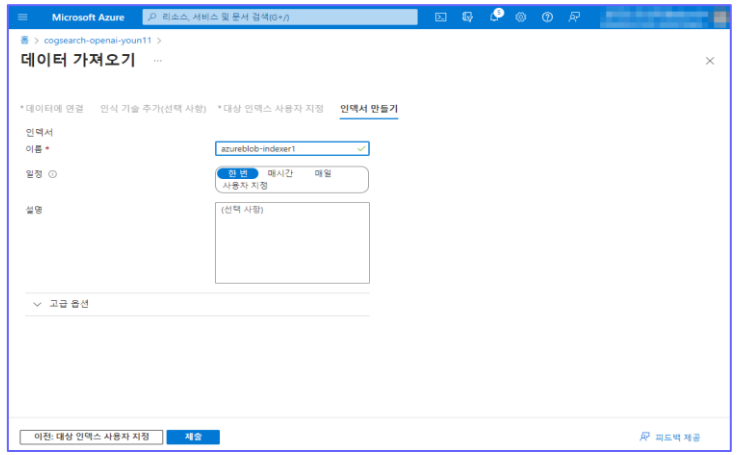
"무료"선택



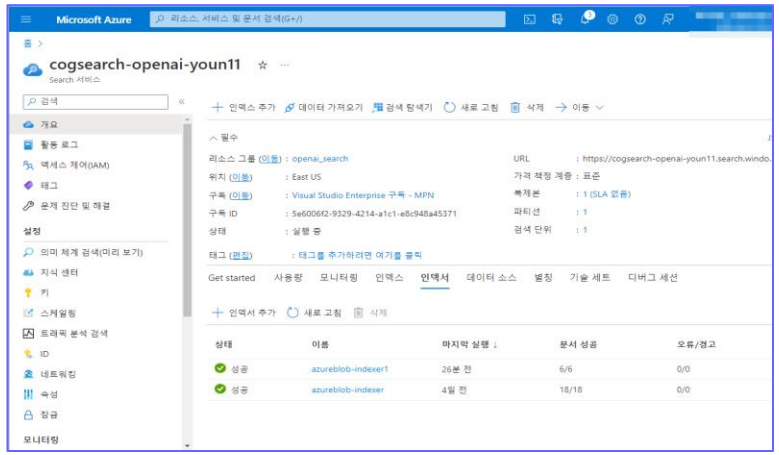
"페이지 5000자 청크, 핵심구 추출"



컨텐츠, 스토리지이름과 패스 , keyphrases 선택 후 pages 필드 추가



한번으로 선택



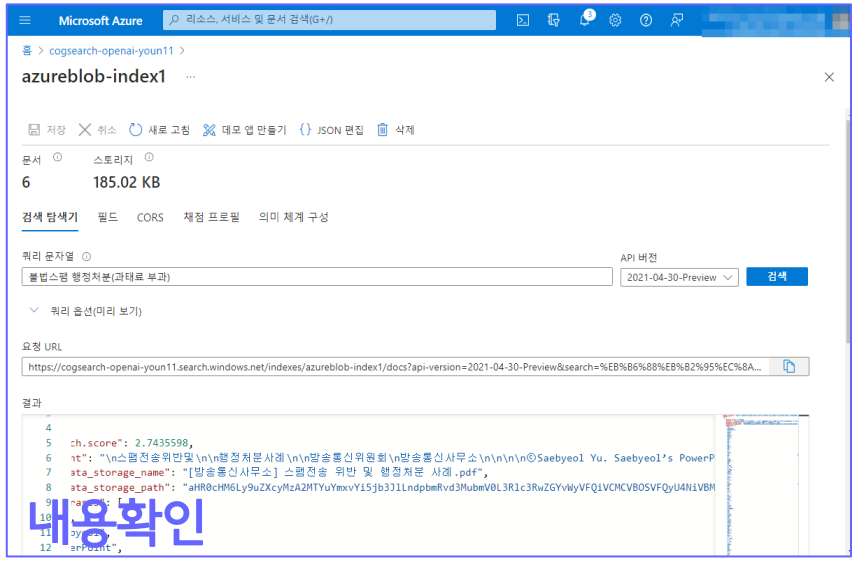
인덱서가 완료되면 연결끝!

4기 유니

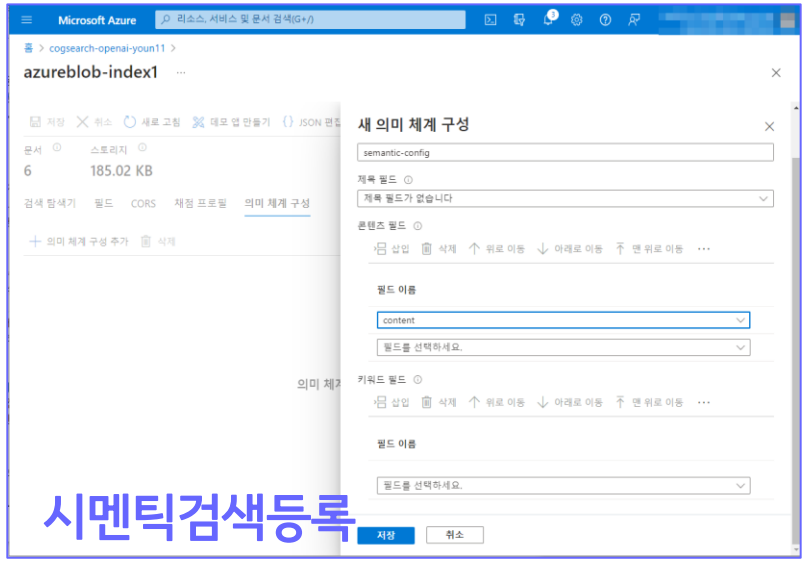
Step2.2 cognitive search 인덱서, 인덱스 설정

인지 검색

인덱스

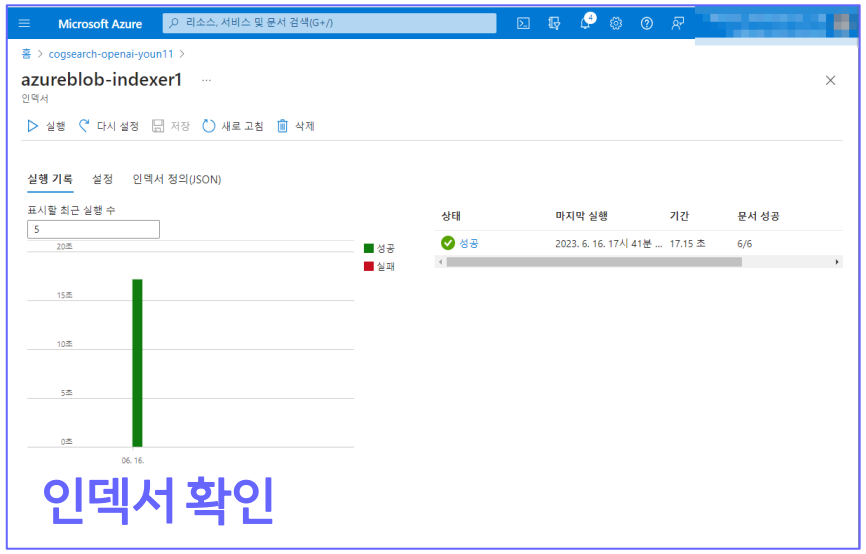


내용확인

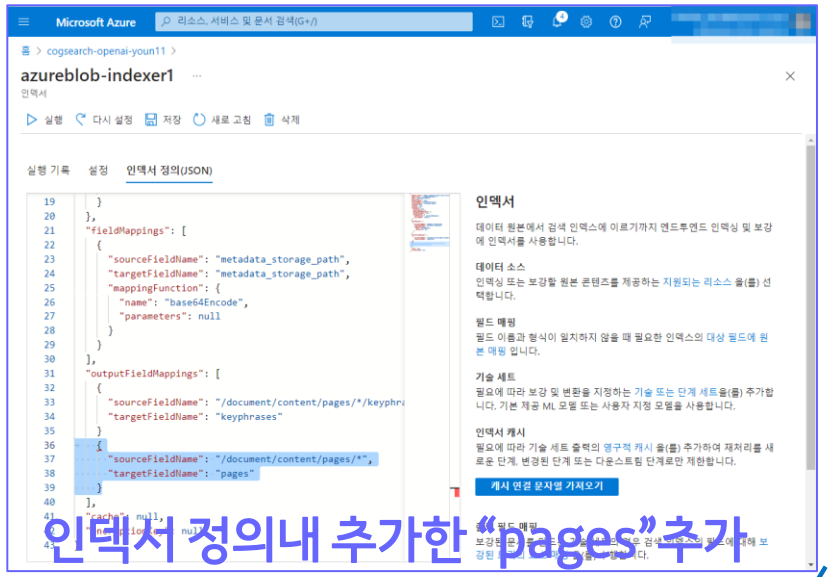


시멘틱검색등록

인덱서



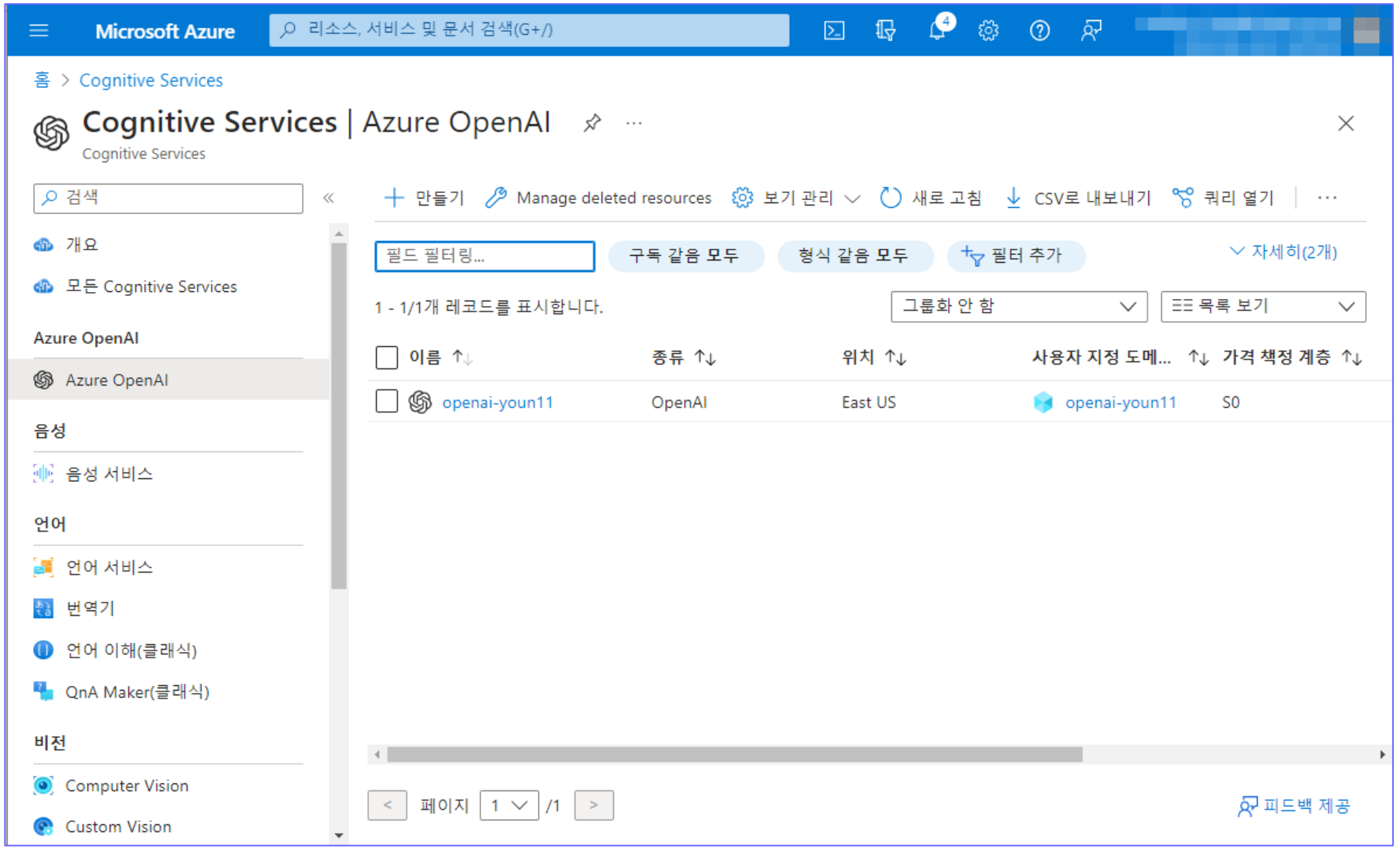
인덱서 확인



인덱서 정의내 추가한 “pages”추가

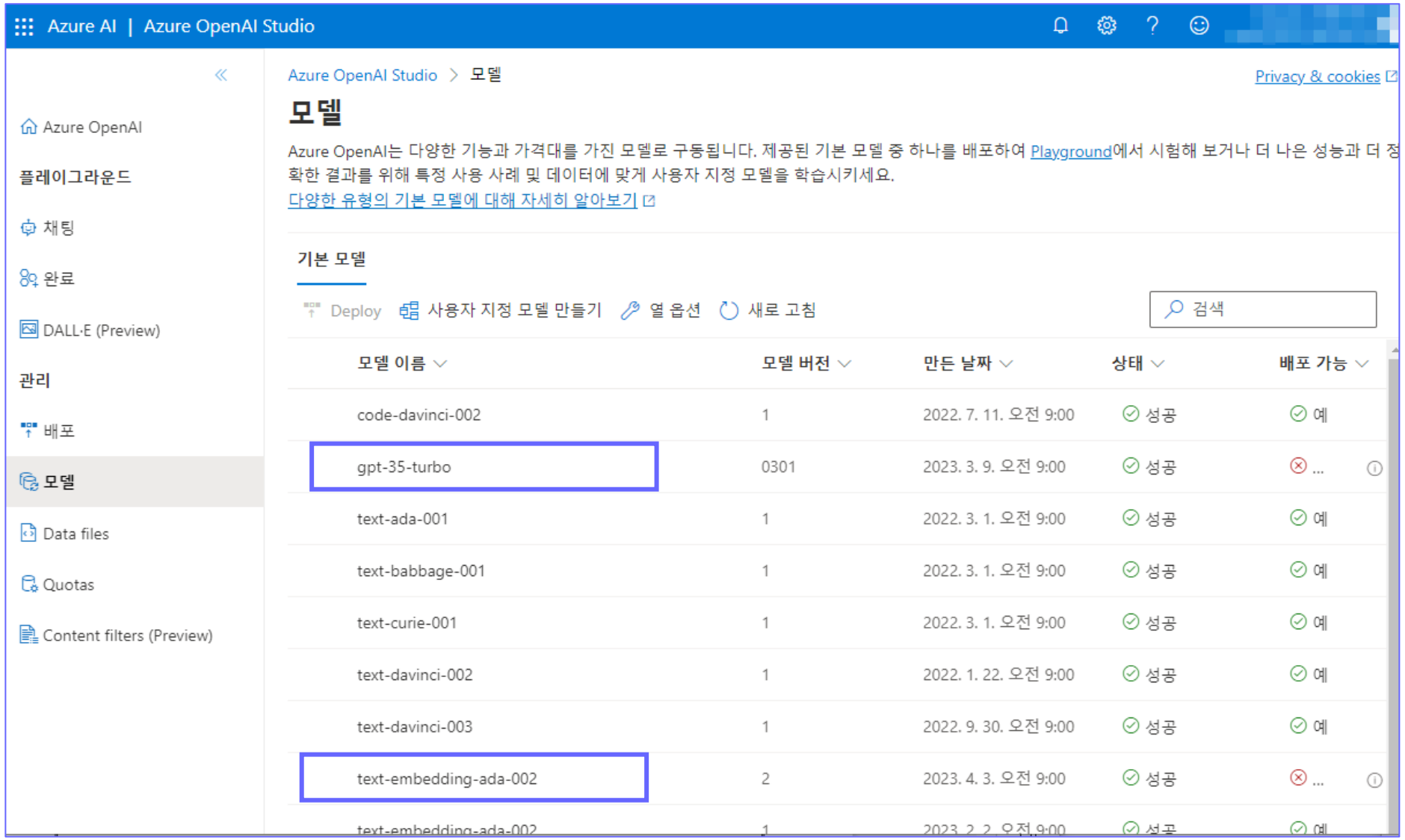
Step3.1 OpenAi 구성

Azure OpenAI

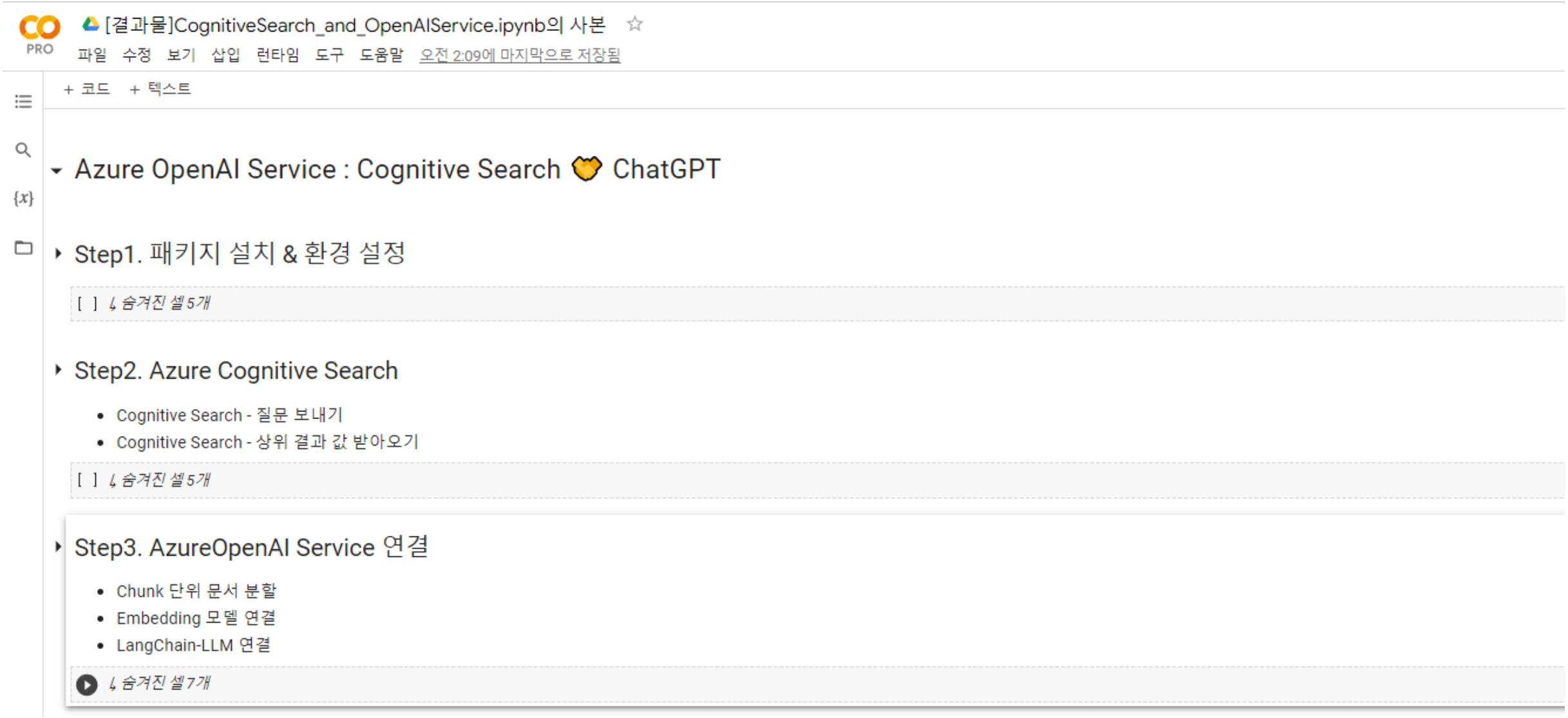


Step3.2 Azure OpenAi 사용할 모델 배포

Azure OpenAI



Step4. 코랩 구성


[결과물]CognitiveSearch_and_OpenAIService.ipynb의 사본
파일 수정 보기 삽입 런타임 도구 도움말 오전 2:09에 마지막으로 저장됨
+ 코드 + 텍스트
Azure OpenAI Service : Cognitive Search ChatGPT
Step1. 패키지 설치 & 환경 설정
[ ] 숨겨진 셀 5개
Step2. Azure Cognitive Search
Cognitive Search - 질문 보내기
Cognitive Search - 상위 결과 값 받아오기
[ ] 숨겨진 셀 5개
Step3. AzureOpenAI Service 연결
Chunk 단위 문서 분할
Embedding 모델 연결
LangChain-LLM 연결
숨겨진 셀 7개

Step4.1 패키지 설치 & 환경 설정

▾ Step1. 패키지 설치 & 환경 설정

```
# 필수 패키지 설치

!pip install OpenAI
!pip install Langchain
!pip install faiss-cpu
!pip install tiktoken
```

```
[ ] # 패키지 가져오기

    import os
    import urllib
    import requests
    import logging
    from collections import OrderedDict
    from IPython.display import display, HTML

    # LangCahin & OpenAI 패키지
    import langchain
    import openai
    from langchain.chat_models import AzureChatOpenAI
    from langchain.vectorstores import FAISS
    from langchain.docstore.document import Document
    from langchain.chains import RetrievalQAWithSourcesChain
    from langchain.embeddings import OpenAIEmbeddings
```

Step4.2 Azure Cognitive Search로 문서 파일 검색

Step2. Azure Cognitive Search

- Cognitive Search - 질문 보내기
- Cognitive Search - 상위 결과 값 받아오기

질문 예시: (2022년 이후 등장한 개념/연구 결과 위주)

1. What is GPT-4?
2. Can GPT-4 accept images as input data?
3. Can ChatGPT help employees be more productivity?
4. Compare GPT-4 and GPT-3
5. What is OTP?

```
[ ] # 질문 설정
    QUESTION = '다이렉트센드 가격을 알려줘'
```

```
[ ] # Azure Cognitive Search REST API 호출(get)

    index_name = 'azureblob-index'

    url = AZURE_SEARCH_ENDPOINT + '/indexes/'+ index_name + '/docs'
    url += '?api-version={}'.format(AZURE_SEARCH_API_VERSION)
    url += '&search={}'.format(QUESTION) # 질문
    url += '&select=*'
    url += '&$top=5'  # 문서 개수 제한
    url += '&queryLanguage=en-US'
    url += '&queryType=semantic' # 의미 체계 검색
    url += '&semanticConfiguration=semantic-config'
    url += '&$count=true'
    url += '&speller=lexicon' # 쿼리 맞춤법 검사
    url += '&answers=extractive|count-3'
    url += '&captions=extractive|highlight-false'

    resp = requests.get(url, headers=headers)
    search_results = resp.json() # 결과값

    #print(search_results)
    print('API 호출 결과 :', resp.status_code)
    print('검색 문서 수: {}, : 상위 문서 수: {}'.format(search_results['@odata.count'], len(search_results['value'])))
```

Step4.3 AzureOpenAI Service 연결

Step3. AzureOpenAI Service 연결

- Chunk 단위 문서 분할
- Embedding 모델 연결
- LangChain-LLM 연결

```
[ ] # 문서 분할

    docs = []
    for key,value in file_content.items():
        for page in value['chunks']:
            docs.append(Document(page_content=page,
                                 metadata={"source": value["file_name"]}))
    print("Number of chunks:",len(docs))
```

```
Number of chunks: 1
```

```
[ ] # Embedding 모델 생성
    embeddings = OpenAIEmbeddings(model="text-embedding-ada-002",
                                  deployment="test1",
                                  chunk_size=1) # Azure OpenAI embedding 사용시 주의
```

```
[ ] # Vector Store 생성
    %%time
    vector_store = FAISS.from_documents(docs, embeddings)
```

```
CPU times: user 613 ms, sys: 69.6 ms, total: 683 ms
Wall time: 2.98 s
```

```
[ ] # LangChain🦜 & Azure GPT😈 연결

    llm = AzureChatOpenAI(deployment_name='test',
                          temperature=0.0, max_tokens=500)

    qa = RetrievalQAWithSourcesChain.from_chain_type(llm=llm,
                                                     chain_type='stuff',
                                                     retriever=vector_store.as_retriever(),
                                                     return_source_documents=True)
```

Step4.4 결과확인

```
[ ]  result = qa({'question': QUESTION})

     print('🙋‍♀️질문 :', QUESTION)
     print('🤖답변 :', result['answer'])
     print('📄 참고 자료 :', result['sources'].replace(",", "₩n"))
```

```
🙋‍♀️질문 : [redacted]격을 알려줘
🤖답변 : [redacted]단일 가격제로 운영되며, 선충전 후 차감 방식이며, 이용요금은 메일 0.5원, 단문 9.5원, 장문 27원, 포토 85원, 알림톡 7.5원, 친구톡(텍스트) 13.5원, 친구톡(이미지) 20원이다. 최소 결제금액은 5,000원(VAT 별도)이며, 환불은 결제방식과 동일하게 처리된다.

📄 참고 자료 [redacted]payment_chatbot_FAQ.pdf
```

Step4.4 그라디오로 챗봇 UI구성

