

Auto M-M

Meeting Minutes

(음성인식 기술을 활용하여 회의록을 자동으로 작성하는 서비스)

LMNOP

1. 주요 기능

1 음성 녹음 파일을 업로드 받는다.



2 음성 파일 스크립트를 텍스트 파일로 제시한다.



3 AI 요약을 제공한다.



2. UI/UX (Figma)

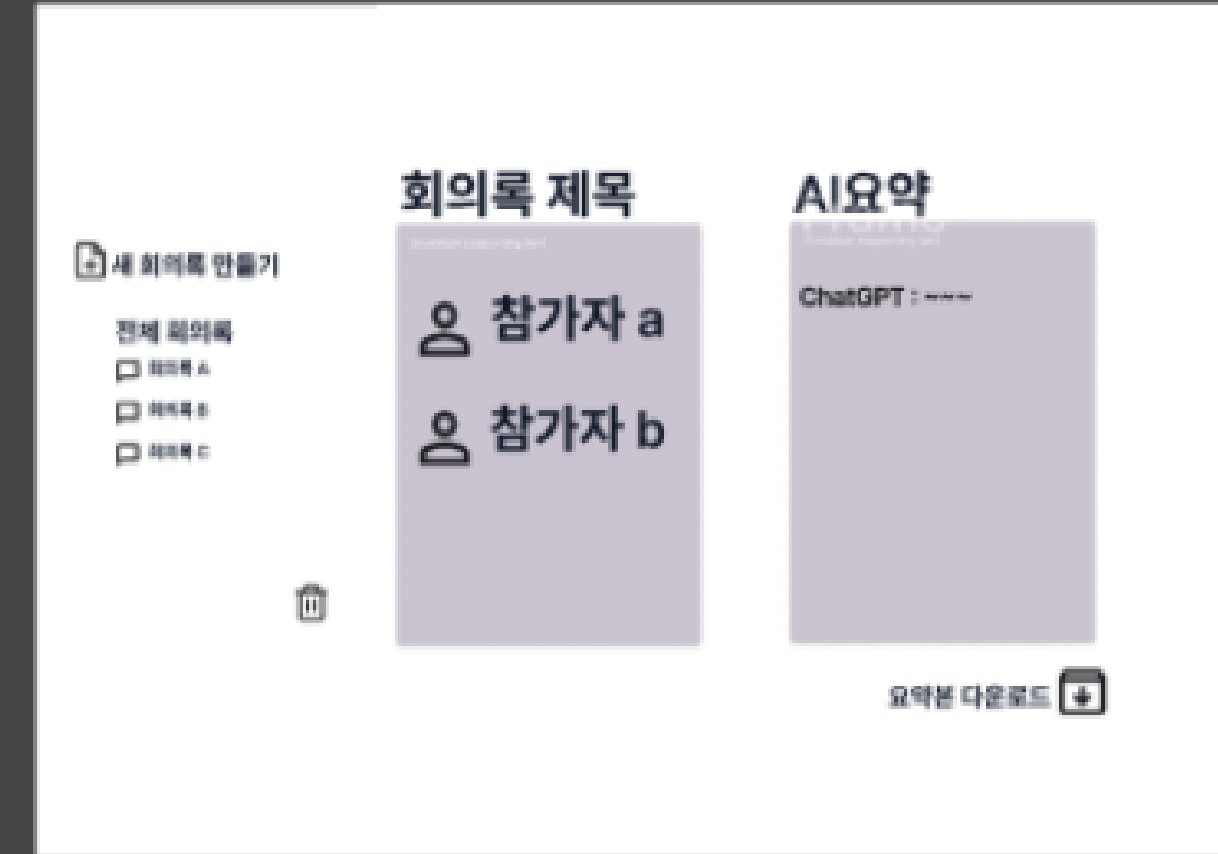
회의록파일 업로드 전



회의록 작성



AI 요약



3. 코드 리뷰

음성 파일 읽어와 Whisper 모델 사용해 텍스트로 변환

음성 인식 함수

```
✓ def transcribe_audio(audio_file_path):  
    audio_file = open(audio_file_path, "rb")  
    transcript = client.audio.transcriptions.create(  
        file=audio_file,  
        model="whisper",  
        language="ko",  
        response_format="text",  
        temperature=0.0,  
    )  
    print(transcript)  
    return transcript
```

3. 코드 리뷰

오디오 파일을 텍스트로 변환하고, 변환된 텍스트를 요약하여 JSON 형식으로 반환

```
# 요약 함수
def summarize_text(text):
    client = AzureOpenAI(
        api_key=str(api_key), # Azure Open AI Key
        api_version=str(api_version), # Azure OpenAI API model
        azure_endpoint=str(endpoint) # Azure Open AI end point(매직에 끝)
    )
    template = """
당신은 텍스트 요점 정리 함수이며, 반환값은 반드시 JSON 데이터여야 합니다.
STEP별로 작업을 수행하면서 그 결과를 아래의 출력 결과 JSON 포맷에 작성하세요.
STEP-1. 아래 세 개의 백틱으로 구분된 텍스트를 원문 그대로 읽어올 것
STEP-2. 텍스트를 개조식으로 요약하세요.
개조식 요약의 정의는 다음과 같다.:
1. 소제목1
    1-1. 내용
    1-2. 내용
    1-3. 내용
...
2. 소제목2
    2-1. 내용
    2-2. 내용
    2-3. 내용
...

다음의 말투로 번역할 것: ["지구의 나이는 45억 살이다.", "세종대왕은 조선의 위대한 국왕이다."]
```{text}```
"""
 return summary
```

출력 결과: [{"STEP-1": text의 첫 50글자, "STEP-2": <text를 요점 정리 결과>}]

```
template = template.format(text=text)

context = [{"role": "user", "content": template}]
response = client.chat.completions.create(
 model="gpt-4o",
 messages=context,
 temperature=0,
 top_p=0,
 seed=1234
)
summary = response.choices[0].message.content
print(summary)
llm = OpenAI(api_key=openai.api_key)
prompt = PromptTemplate(
input_variables=["text"],
template="Please summarize the following meeting transcript:\n\n{text}"
)
chain = LLMChain(llm=llm, prompt=prompt)
summary = chain.run(text=text)
return summary
```

### 3. 코드 리뷰

```
파일 업로드 받기
uploaded_file = st.file_uploader("Choose an audio file...", type=["wav"])

if uploaded_file is not None:
 with open("temp_audio.wav", "wb") as f:
 f.write(uploaded_file.getbuffer())

음성 인식 (음성 파일을 텍스트로 변환)
transcript = transcribe_audio("temp_audio.wav")
st.subheader("Meeting Transcript")
st.text_area("Transcript", transcript, height=300)

텍스트 요약
if st.button("Summarize Transcript"):
 summary = summarize_text(transcript)
 st.subheader("Summary")
 st.text_area("Summary", summary, height=150)
```

- 사용자가 .WAV 형식의 음성 파일을 업로드할 수 있도록 파일 업로더를 설정
- 업로드된 파일을 임시 파일(TEMP\_AUDIO.WAV)로 저장
- 업로드된 파일이 있을 경우, 음성 인식 함수를 호출하여 텍스트 트랜스크립트를 생성
- 생성된 트랜스크립트를 화면에 출력
- "SUMMARIZE TRANSCRIPT" 버튼을 클릭
  - 요약 함수를 호출하여 요약된 텍스트를 화면에 출력

## 4. UI 구성

# Automatic Meeting Transcript and Summarization

Upload an audio file to transcribe and summarize the meeting.

Choose an audio file...



Drag and drop file here

Limit 200MB per file • WAV

Browse files

## 5. 실행 화면



### Meeting Transcript

#### Transcript

그래서 처음 가시 설치하시는 것은 요놈을 제가 추천해드립니다. 자 우분투가 어려워요. 그러면 여러분들은 윈도우즈에서 JDK를 설치하셔도 됩니다. 그런데 그 환경을 수요일까지는 만들어 오셔야 돼요. 수요일까지, 수요일날 공장 실습을 할거예요. 그래서 그 방법 확인을 할거예요. 제가 드리고 싶은 마음은 다 드렸고 제가 초대해서 다시 올려놓을게요 라고 했었던 것은 준법보사법을 10월 25일인가 수정해서 다시 올려놓을거예요. 늦게 오신 분을 위해서 하나만 더 슬랩 초대 링크를 보냈으니까 그게 20일인가 그 안에 가입을 안하시면 등록을 안하시면 링크에 접속을 못해요. 그래서 그 안에 하셔야 되고 분명히 기말고사 보기전에 링크좀 보내주세요 이런 분들이 반드시 계세요. 모든 공지는 슬랩으로 합니다. 보충이 오늘 있습니다. 예를 들어서 뭐 있습니다. 뭐 하셔야 됩니다. 이런 것들을 공지할 때는 슬랩에다가 우선적으로 공지를 할겁니다. 그래서 슬랩을 잘 보시면 되고 이제 끝나기 1분전 1분 동안 뭘 얘기할거냐. 나나도 제가 얘기를 계속 반복해서 할거야. 했던 얘기를 또 하고 또 하고 듣기 싫은데 또 들어야 되고 그렇게 될거라고. 그런데 매일 처음에는 몰라요. 한번 얘기하면. 자 그러면 저 놈이 무슨 얘기를 하는지 모르지. 그런데 내가 이거 설명할 때 A설명할 때 B도 또 설명하고 C설명할 때 A도 또 설명하고 했던 얘기를 계속 하게 될텐데 그러면 저절로 알겠대요. 아 그때 저 놈이 이 얘기를 한거였나. 그러면 이제 조금씩 조금씩 학습하게 되는거죠. 그래서 초반에 할머니가 또 하고 했다고 해서 너무 싫어할 필요는 없다고.

Summarize Transcript

### Summary

#### Summary

STEP-2 . {

"1. 가시 설치 추천": {

"1-1. 우분투가 어려운 경우 윈도우즈에서 JDK 설치 가능",

"1-2. 수요일까지 환경 설정 필요",

"1-3. 수요일에 실습 예정"

},

"2. 슬랩 초대 링크": {



# 6. 배포 과정

aws

서비스

검색

[알트+S]

Cloud9

EC2

CloudFormation

WAF & Shield

S3

오하이오

haram

EC2 대시보드

EC2 글로벌 보기

이벤트

인스턴스

인스턴스

인스턴스 유형

시작 템플릿

스팟 요청

Savings Plans

예약 인스턴스

전용 호스트

용량 예약

이미지

AMI

AMI 카탈로그

Elastic Block Store

볼륨

스냅샷

수명 주기 관리자

EC2 > 인스턴스 > i-04c5845937c570111

i-04c5845937c570111 (academyManager\_BE)에 대한 인스턴스 요약 정보

less than a minute 전에 업데이트됨

인스턴스 ID

i-04c5845937c570111 (academyManager\_BE)

퍼블릭 IPv4 주소

18.223.237.251 | [개방 주소법](#)

인스턴스 상태

실행 중

퍼블릭 IPv4 주소

172.31.15.116

퍼블릭 IPv4 DNS

ec2-18-223-237-251.us-east-2.compute.amazonaws.com | [개방 주소법](#)

호스트 이름 유형

IP 이름: ip-172-31-15-116.us-east-2.compute.internal

프라이빗 리소스 DNS 이름 응답

IPv4(A)

프라이빗 IP DNS 이름(IPv4만 해당)

ip-172-31-15-116.us-east-2.compute.internal

인스턴스 유형

t2.micro

자동 할당된 IP 주소

18.223.237.251 [퍼블릭 IP]

VPC ID

vpc-0f22898490d5b28d8

IAM 역할

-

서브넷 ID

subnet-0a4b3043c08fb8dba

인스턴스 ARN

탄력적 IP 주소

-

AWS Compute Optimizer 찾기

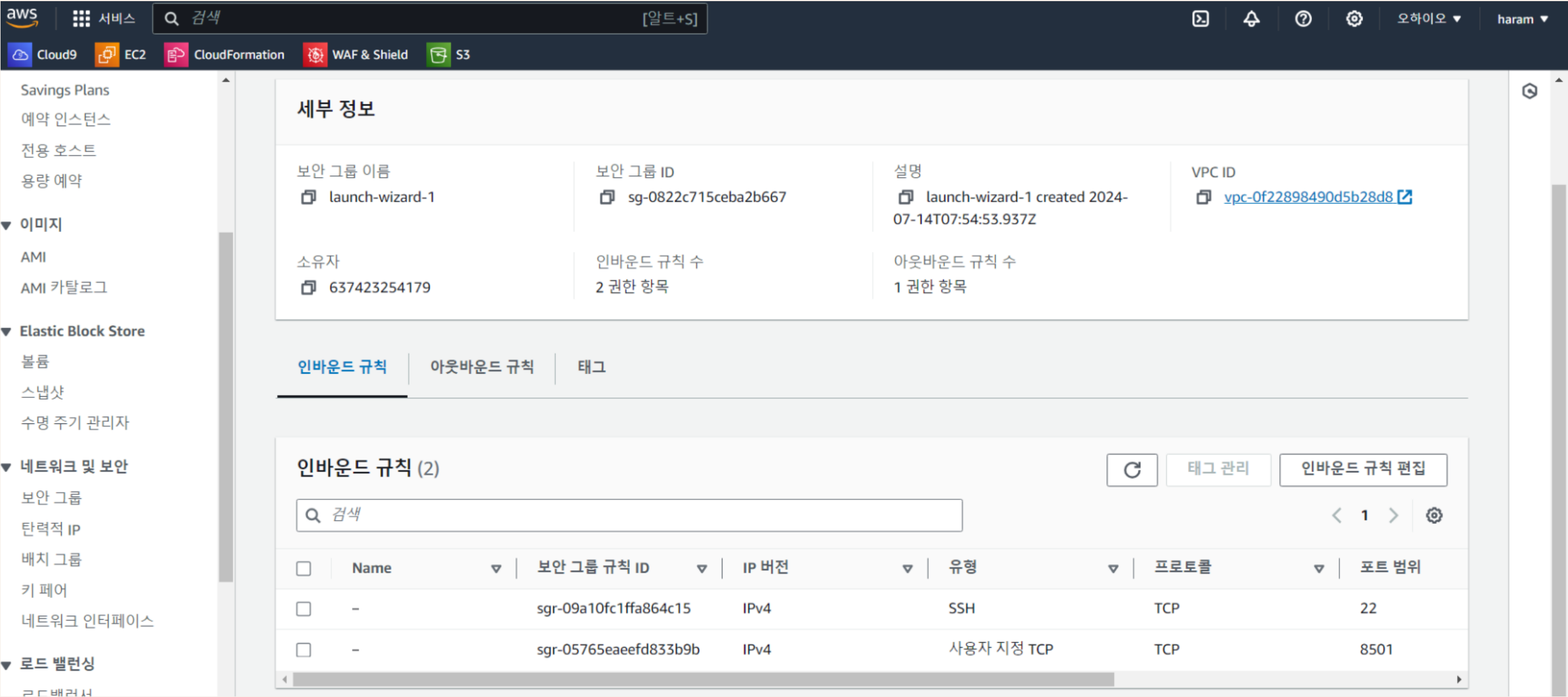
권장 사항을 위해 AWS Compute Optimizer에 옵트인합니다. | [자세히 알아보기](#)

Auto Scaling 그룹 이름

-

## AWS EC2 인스턴스 생성

# 6. 배포 과정



## AWS EC2 보안그룹 포트 설정