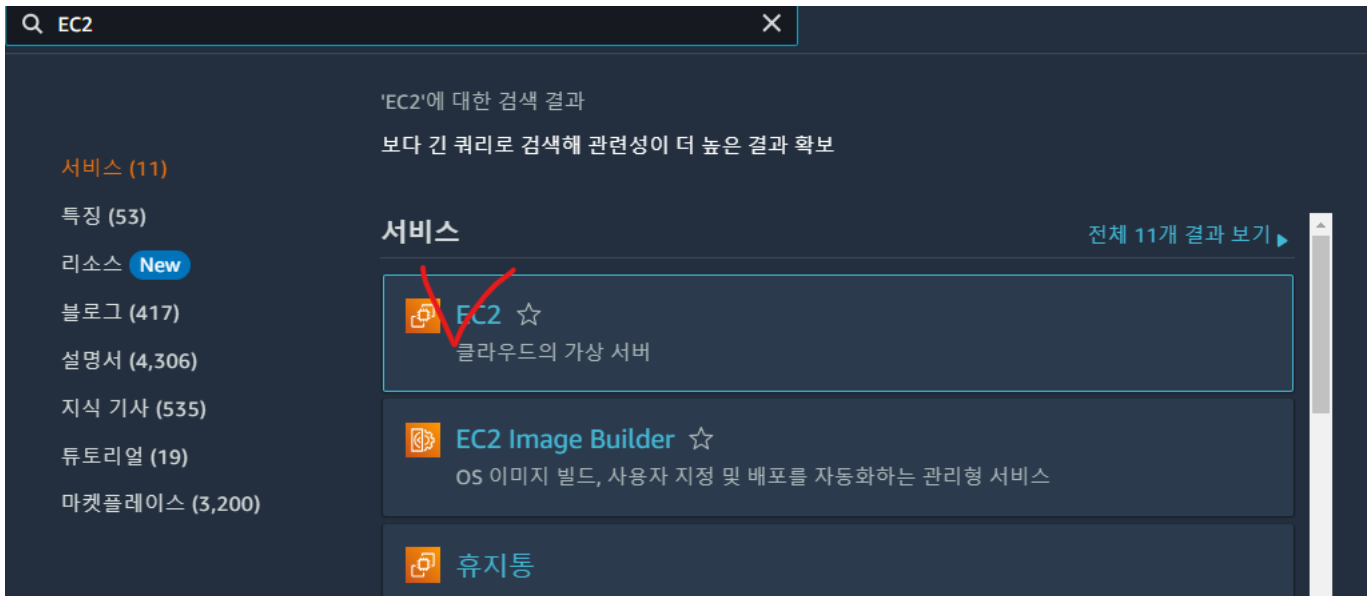


# AWS\_Cloud(2023-11-16 Commit)

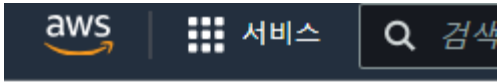
## ■ Flask 서비스 AWS로 배포하기 - AWS 서버 대여

### 1. AWS 가상서버 대여하기

- EC2 서비스로 들어가기



- 인스턴스 클릭



EC2 대시보드



EC2 글로벌 보기

이벤트

▼ 인스턴스

인스턴스

인스턴스 유형

시작 템플릿

스팟 요청

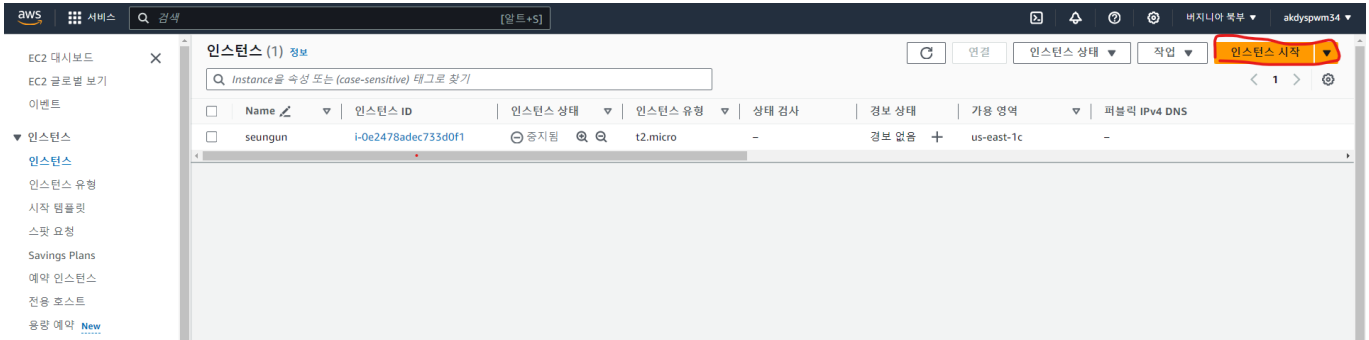
Savings Plans

예약 인스턴스

전용 호스트

용량 예약 New

- 인스턴스 시작 버튼 클릭



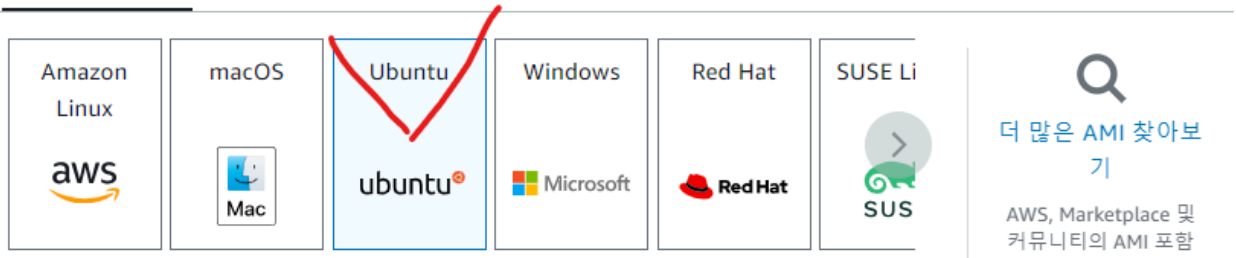
- 들어가서 Ubuntu Server 22.04 선택

### ▼ Application and OS Images (Amazon Machine Image) 정보

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

🔍 수천 개의 애플리케이션 및 OS 이미지를 포함하는 전체 카탈로그 검색

#### Quick Start



#### Amazon Machine Image(AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type  
ami-0fc5d935ebf8bc3bc (64-bit (x86)) / ami-016485166ec7fa705 (64-bit (Arm))  
가상화: hvm ENA enabled: true 루트 디바이스 유형: ebs

프리 티어 사용 가능 ▼

- 인스턴스 유형(프리티어) - t2.micro(1CPU, 1GB메모리)

### ▼ 인스턴스 유형 정보

#### 인스턴스 유형

##### t2.micro

프리 티어 사용 가능

패밀리: t2 1 vCPU 1 GiB 메모리 현재 세대: true  
온디맨드 Windows 기본 요금: 0.0162 USD 시간당  
온디맨드 SUSE 기본 요금: 0.0116 USD 시간당  
온디맨드 RHEL 기본 요금: 0.0716 USD 시간당  
온디맨드 Linux 기본 요금: 0.0116 USD 시간당

☐ 모든 세대

[인스턴스 유형 비교](#)

소프트웨어가 사전 설치된 AMI에는 추가 비용이 적용됩니다.

- 인스턴스 시작 버튼 클릭 -> 키페어생성

## 키 페어 생성



키 페어를 선택하지 않은 것으로 확인되었습니다. 인스턴스에 연결하려면 키 페어를 생성하는 것이 좋습니다.

☒ 새 키 페어 생성

☐ 키 페어 없이 계속 진행

## 키 페어 이름

키 페어를 사용하면 인스턴스에 안전하게 연결할 수 있습니다.

**이름을 입력**

이름에는 최대 255개의 ASCII 문자를 포함할 수 있습니다. 앞 또는 뒤에 공백을 포함할 수 없습니다.

## 키 페어 유형

☒ RSA  
RSA 암호화된 프라이빗 및 퍼블릭 키 페어

☐ ED25519  
ED25519 암호화된 프라이빗 및 퍼블릭 키 페어

## 프라이빗 키 파일 형식

☒ .pem  
OpenSSH와 함께 사용

☐ .ppk  
PuTTY와 함께 사용

메시지가 표시되면 프라이빗 키를 사용자 컴퓨터의 안전하고 액세스 가능한 위치에 저장합니다. 나중에 인스턴스에 연결할 때 필요합니다. [자세히 알아보기](#)

취소

키 페어 생성

- 키페어 생성되면 .pem 파일이 중요함(이러면 인스턴스 생성 완료) 지금은 중지 상태지만 바로 생성하면 시작 상태로 변경됨

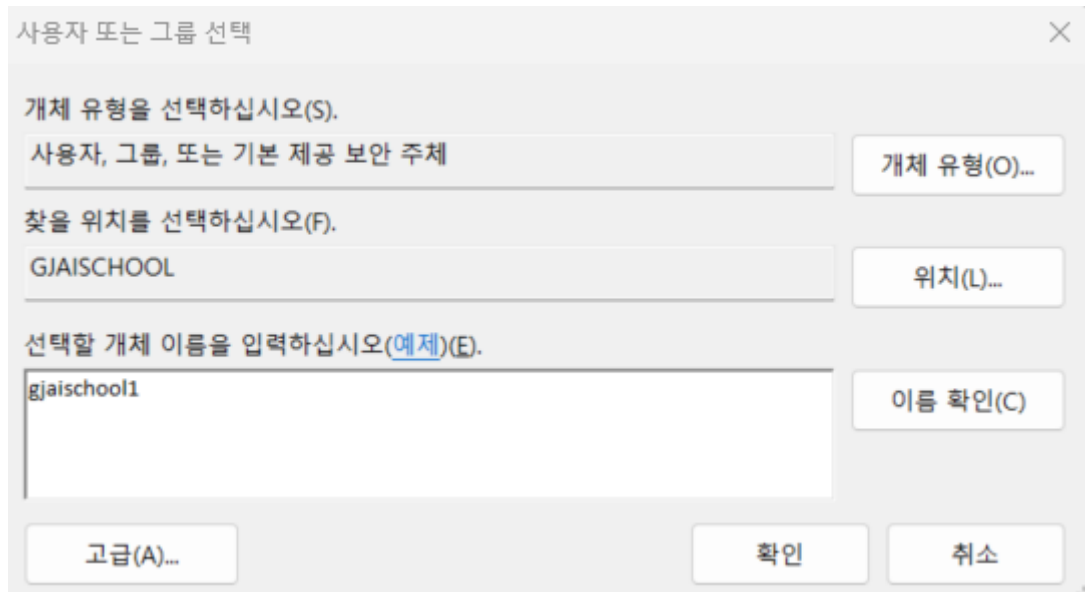
인스턴스 (1) 정보							
<input type="text" value="Instance를 속성 또는 (case-sensitive) 태그로 찾기"/>							
<input type="checkbox"/>	Name	인스턴스 ID	인스턴스 상태	인스턴스 유형	상태 검사	경보 상태	가용 영역
<input type="checkbox"/>	seungun	i-Oe2478adec733d0f1		t2.micro	-	경보 없음	us-east-1c

## 2. .pem 파일 경로 설정 (윈도우ver)

- 로컬디스크C -> 사용자 -> gjaischool1 에 .ssh폴더 만들기



- 권한 설정 (.ssh파일 우클릭 -> 속성 -> 보안 -> 고급 -> 상속 사용안함 -> 이개체에서 상속된 사용자 권한을 모두 제거 -> 추가 -> 보안주체선택)



- 확인 후 쓰기권한 체크후 적용해준다.
- 끝을 낸 뒤 .pem 파일을 해당 .ssh 폴더에 넣어준다.

## 3. ec2 서버에 접속하기

- \$ssh -i [키 페어 경로] [유저 이름]@[퍼블릭 DNS 주소] -> user이름은 보통 ubuntu로 설정됨

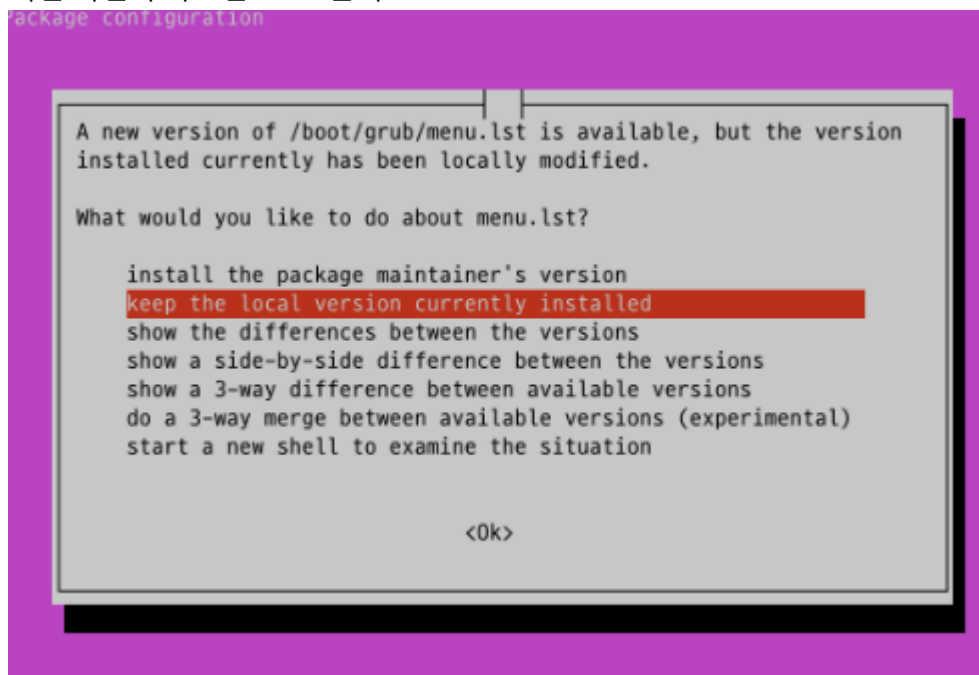
```
명령 프롬프트
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\gjaischoo1>ssh -i ~/.ssh/[redacted].pem ubuntu@[redacted].amazonaws.com|
```

- enter 후 -> yes 입력후 엔터 -> 우분투 환경에 잘 접속된것을 확인할 수 있음 ※ 만약에 yes가 안뜬다면 위과정을 잘 실행했는지 다시한번 확인

```
Last login: Wed Nov 15 08:34:54 2023 from [redacted]
ubuntu@ip-172-31-[redacted] ~$ |
```

- 패키지 정보를 업데이트 : sudo apt-get update
- 패키지 의존성 검사 및 업그레이드: sudo apt-get dist-upgrade
- 이런 화면이 나오면 enter 클릭



- python3 패키지 매니저(pip3) 설치 : sudo apt-get install python3-pip

## 4. Github에 프로젝트 업로드하기

- vscode 사용

```
if __name__ == '__main__':
    app.run(debug=False, host='0.0.0.0')
```

- debug는 상관없고 host는 0.0.0.0으로 설정해줘야함
- app.py 가있는곳으로 터미널 접속하기
- 종속성 만들기 : `pip3 freeze >> requirements.txt`
- 종속성 내용 확인하여 잘 들어갔는지 확인 : `cat requirements.txt`
- git 초기화 : `git init`
- git에 현재 폴더 전체를 담기 : `git add .`
- 담은 파일들을 업로드할 레포지토리 주소를 origin이라는 이름으로 추가 : `git remote add origin [레포지토리 주소]`
- commit 하기 : `git commit -m "하고싶은말"`
- 담은 파일들을 origin이라는 front\_end\_v1 이름의 브랜치에 레포지토리 주소로 업로드 ex): `git push origin front_end_v1`

### front\_end\_v1 브랜치를 default로 설정하기

- 내가 올린 원격저장소 branch 클릭

The screenshot shows the GitHub interface for the repository 'ai-speaker' by user 'leeseungun1'. The '3 branches' link is highlighted with a red circle. Below the repository name, there's a message: 'Your front\_end\_v1 branch isn't protected'. A table lists recent commits by 'leeseungun1 server' with files like 'static', 'templates', 'a.html', 'app.py', and 'requirements.txt'. The right sidebar shows repository statistics: 0 stars, 1 watching, 0 forks, and a language distribution chart (HTML 60.1%, CSS 16.9%, JavaScript 12.9%, SCSS 9.8%).

- switch 버튼을 통해 front\_end\_v1브랜치로 default 바꿔준다. 이러면 EC2 서버에 업로드할 준비가 끝남

The screenshot shows the GitHub repository interface. At the top, there's a search bar and tabs for Overview, Yours, Active, Stale, All branches, and a New branch button. Below this, the 'Default branch' section shows 'front\_end\_v1' as the default branch, updated 16 hours ago by leeseungun1. A red circle highlights a warning icon. Below the warning, it says 'Your front\_end\_v1 branch isn't protected' and provides a 'Protect this branch' button. The 'Your branches' section lists 'master' and 'main' branches, both updated yesterday by leeseungun1. The 'Active branches' section also lists 'master' and 'main' branches, both updated yesterday by leeseungun1.

## 5. EC2 서버에서 git clone 하기

- 이 프롬프트에서 실행

```
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Wed Nov 15 08:34:54 2023 from 121.66.158.212
ubuntu@ip-172-31-20-110:~$
```

- 프로젝트 파일은 모두 srv폴더에 다운을 받는다
- 현재 유저인 ubuntu로 소유권 바꾸기 : `sudo chown -R ubuntu:ubuntu /srv/`
- `cd /`
- `ls -al -> drwxr-xr-x 2 ubuntu ubuntu 4096 Jan 12 17:33 srv` 권한이 ubuntu로 바뀐걸 확인할 수 있음
- srv 폴더로 이동 : `cd /srv`
- `git clone [레포지토리 주소]`
- ls를 통해 레포지토리 이름과 동일한 폴더가 생기는지 확인

## 6. uWSGI 연결하기

- 가상환경을 만들기전에 `python3-venv`를 설치: `sudo apt-get install python3-venv`
- 가상환경의 위치는 현재사용하고 있는 유저인 ubuntu 홈폴더에 만들기
- `cd ~`
- `python3 -m venv myvenv`
- 가상환경 활성화: `source myvenv/bin/activate`
- `cd /srv/레포지토리이름/`



- 종속성 설치: `pip3 install -r requirements.txt` (설치가 안되면 일일이 설치해줘야함)
- `sudo python3 app.py` 하면 서버가 실행됨

## 7. 보안그룹 인바운드 규칙 주기 (위에걸로는 포트설정이 안되어 있어 실행이 안됨)

- 인스턴스 ID 클릭

Name	인스턴스 ID	인스턴스 상태	인스턴스 유형	상태 검사	경보 상태	가용 영역	퍼블릭 IPv4 DNS
seungun	i-0e2478adec733d0f1	실행 중	t2.micro	2/2개 검사 통과...	경보 없음	us-east-1c	ec2-34-227-74-220.compute-1.amazonaws.com

- 보안 클릭

i-0e2478adec733d0f1 (seungun)에 대한 인스턴스 요약 정보		
인스턴스 ID i-0e2478adec733d0f1 (seungun)	퍼블릭 IPv4 주소 34.227.74.220   <a href="#">개방 주소법</a>	프라이빗 IPv4 주소 172.31.20.110
IPv6 주소 -	인스턴스 상태 실행 중	퍼블릭 IPv4 DNS ec2-34-227-74-220.compute-1.amazonaws.com   <a href="#">개방 주소법</a>
호스트 이름 유형 IP 이름: ip-172-31-20-110.ec2.internal	프라이빗 IP DNS 이름 (IPv4만 해당) ip-172-31-20-110.ec2.internal	탄력적 IP 주소 -
프라이빗 리소스 DNS 이름 응답 IPv4(A)	인스턴스 유형 t2.micro	AWS Compute Optimizer 찾기 <a href="#">권장 사항을 위해 AWS Compute Optimizer에 업데이트합니다.</a>   <a href="#">자세히 알아보기</a>
자동 할당된 IP 주소 34.227.74.220 [퍼블릭 IP]	VPC ID vpc-043ed984317bac88b	Auto Scaling 그룹 이름 -
IAM 역할 -	서브넷 ID subnet-04daa2619c3894ad6	
IMDSv2 Required		

- 보안그룹 밑에 sg-... 클릭

보안 세부 정보		
IAM 역할 -	소유자 ID 325407071077	시작 시간 Thu Nov 16 2023 09:24:58 GMT+0900 (한국 표준시)
보안 그룹 sg-0f966311b88fe9276 (launch-wizard-2)		

- 인바운드 규칙 편집

Name	보안 그룹 규칙 ID	IP 버전	유형	프로토콜	포트 범위	소스
-	sgr-0ec0faa5749c72771	IPv4	HTTP	TCP	80	0.0.0.0/0
-	sgr-0f92b6b471dc240e2	IPv4	사용자 지정 TCP	TCP	5000	0.0.0.0/0
-	sgr-0e6238bfb94652e39	IPv4	SSH	TCP	22	0.0.0.0/0

- 규칙추가

인바운드 규칙 정보

보안 그룹 규칙 ID	유형 정보	프로토콜 정보	포트 범위 정보	소스 정보	설명 - 선택 사항 정보	삭제
sgr-0ec0faa5749c72771	HTTP	TCP	80	사용자 ... 0.0.0.0/0		삭제
sgr-0f92b6b471dc240e2	사용자 지정 TCP	TCP	5000	사용자 ... 0.0.0.0/0		삭제
sgr-0e6238fb94652e39	SSH	TCP	22	사용자 ... 0.0.0.0/0		삭제

**규칙 추가**

- 플라스크는 5000 포트 django는 80 포트를 사용함 위의 규칙대로 설정해주면 됨
- 다시 `sudo python3 app.py` 실행
- 이제 퍼블릭 IPv4 주소 + :port 번호 입력하면 접속가능함

인스턴스 (1/1) 정보

Instance를 속성 또는 (case-sensitive) 태그로 찾기

상태 검사	경보 상태	가용 영역	퍼블릭 IPv4 DNS	퍼블릭 IPv4 주소	탄력적 IP	IPv6 IP
2/2개 검사 통과...	경보 없음	us-east-1c	[redacted]	34.227.74	-	-