



# AWS / Client-Server 환경 만들기

# 시간계획

- 오늘도 파이팅 입니다.^^

시간	학습내용
09:00~10:00	몽고DB 실습
10:20~11:20	AWS 서버 구축
11:40~12:40	몽고DB 설치, 클라이언트 서버 환경 구축
12:40~14:00	즐거운 점심 시간
14:00~15:00	팀 프로젝트
15:20~16:20	
16:40~17:50	

# AWS 클라우드 컴퓨팅

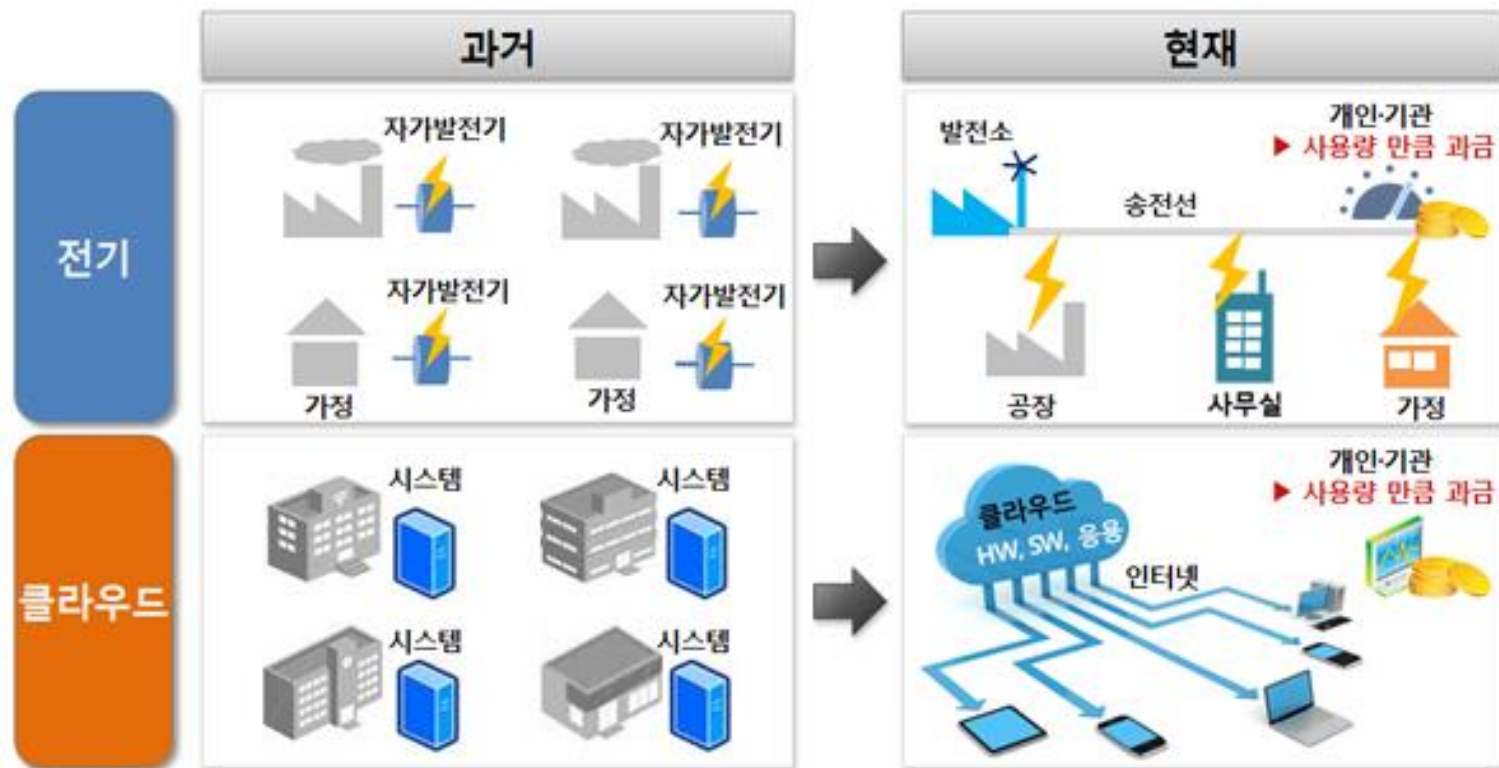
# 학습 목표

---

- 클라우드 컴퓨팅에 대해 이해한다.
- 클라우드 컴퓨팅의 다양한 서비스 형태를 이해한다.
- AWS 클라우드의 EC2 서비스를 사용한다.
- EC2 서버에 몽고DB를 설치하여 인터넷 기반 클라이언트 서버 환경을 구축한다.
- VS code로 EC2와 연결한 개발 환경을 구축한다.

# 클라우드 컴퓨팅 이해

- 클라우드 컴퓨팅의 발전



# 클라우드 컴퓨팅

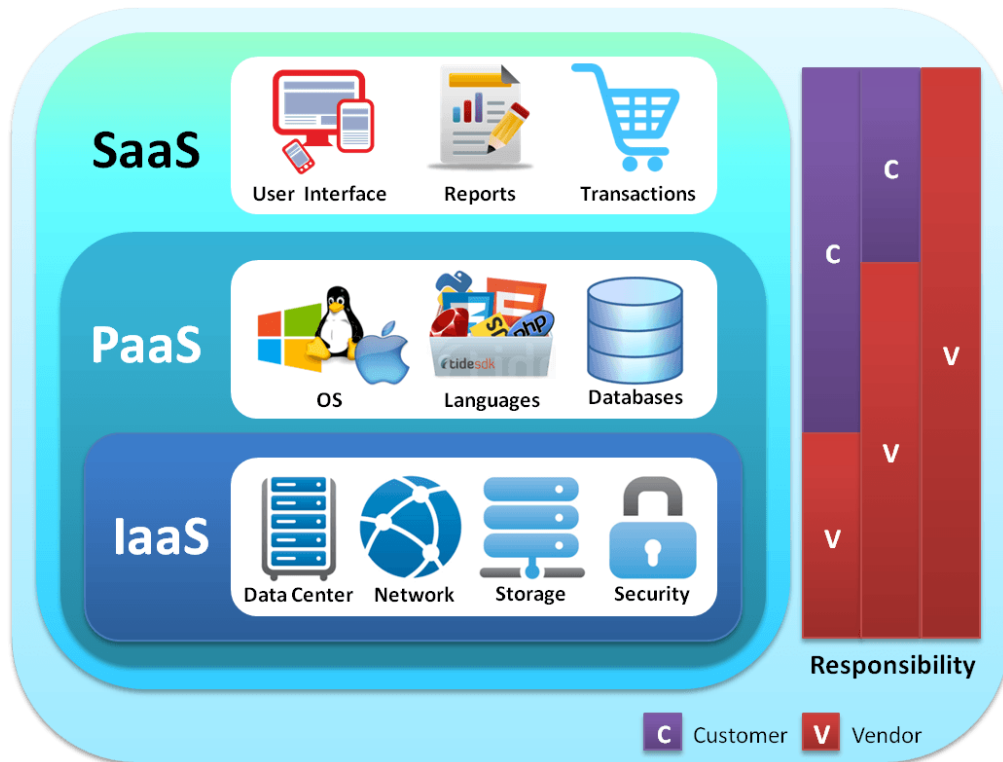
- IT 자원을 서비스로 제공하는 방법
- IT 자원인 CPU, RAM, 네트워크, 스토리지 등을 고객에게 임대해주고 가금을 함



# 클라우드 컴퓨팅

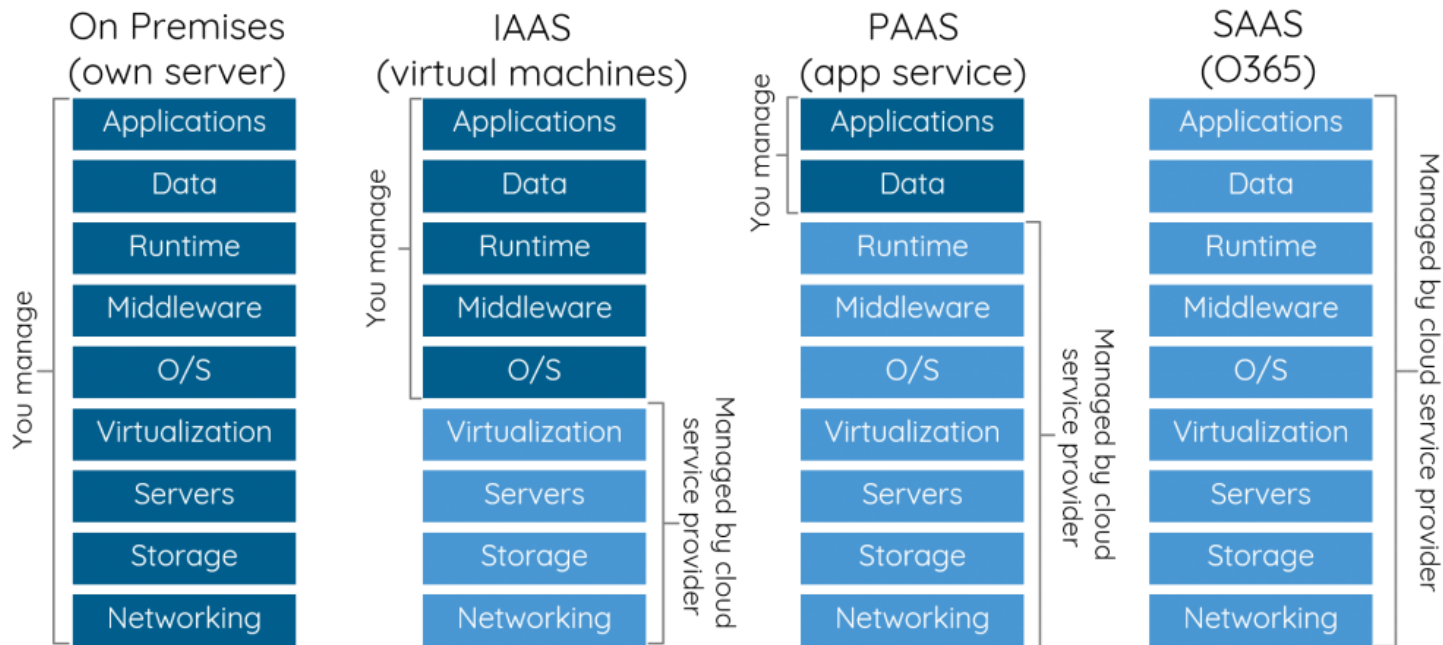
- 서비스 모델에 따른 분류

- IaaS; Platform as a Service  
(아이아스, 이아스)
- PaaS; Software as a Service  
(파스)
- SaaS; Software as a Service  
(사스)



# 클라우드 컴퓨팅

- IAAS(Infrastructure-as-a-Service), PaaS : Platform as a Service
- SaaS(Software as a Service), On-Promise(기존방식)





# 클라우드 컴퓨팅의 장점, 단점

## 장점

- 비용절감(서버 구축비, 유지관리비, 인건비)
- 쉬운 사용성
- QoS(Quality of Service): 밴더와 계약한 만큼 사용
- 신뢰성(네트워크 규모, 로드밸런싱, 페일오버 등)

## 단점

- 공급자가 제공하는 SW를 사용해야함
- 클라우드 컴퓨팅안에서 보안에 대한 것을 스스로 관리해야함

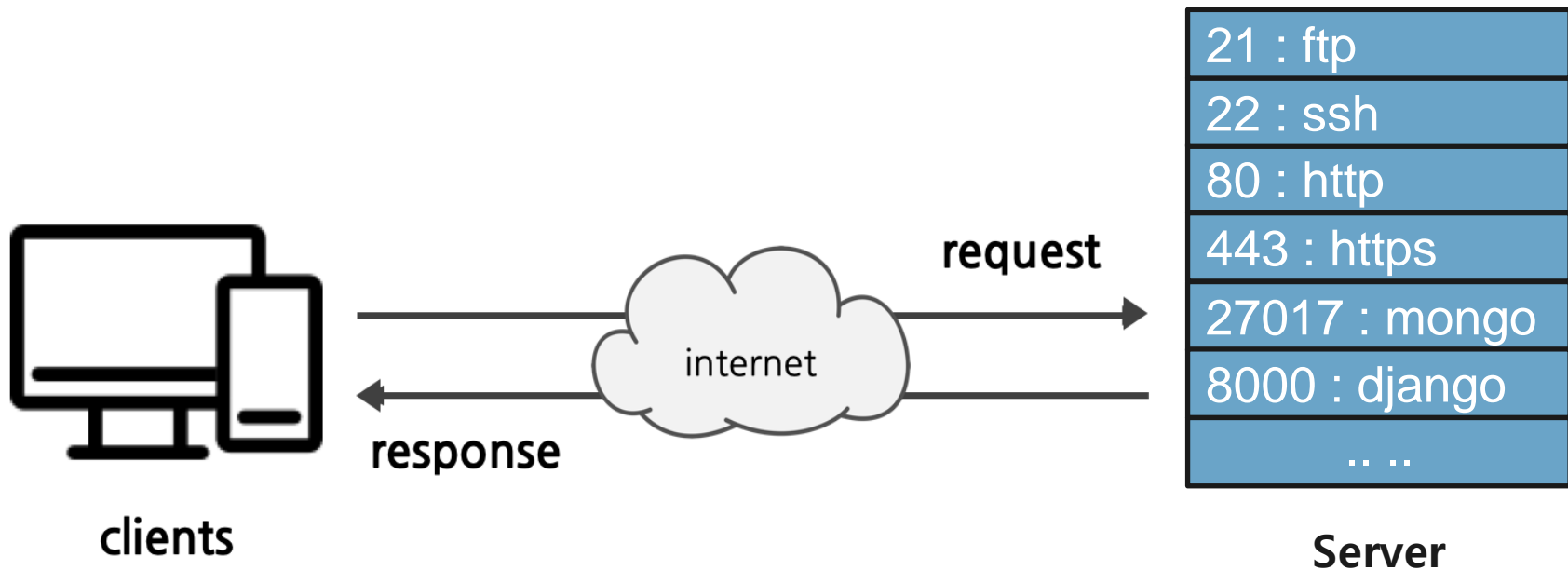
# 클라우드 컴퓨팅 서비스 업체



글로벌 3대 업체



# 클라이언트 서버 환경 이해



서비스를 요청하는 쪽

서비스를 제공하는 쪽

# EC2(Amazon Elastic Compute Cloud)



- 인스턴스 : 가상 컴퓨팅 환경
- AMI(Amazon Machine Image) : 서버에 필요한 운영체제와 여러 소프트웨어들이 적절히 구성된 상태로 제공된 템플릿
- 인스턴스 유형 : CPU, 메모리, 스토리지, 네트워킹 용량의 여러가지 유형
- 탄력적 IP 주소 : 동적 클라우드 컴퓨팅을 위한 고정 IPv4 주소
- 보안그룹 : 인스턴스에 연결할 수 있는 프로토콜, 포트, 소스 IP 범위를 지정하는 방화벽 기능

# AWS EC2 서버 구축하기

# AWS EC2 서버 구축하기

---

- 회원가입
  - <https://aws.amazon.com/ko/free/>
  - 준비물 : 해외 결제가 가능한 신용카드  
email 주소

# AWS EC2 서버 구축하기

- EC2 또는 인스턴스(서버)생성
  - 기본 : 동적 IP
- Elastic IP(탄력적IP) 생성
  - 고정 IP 생성
  - 가금이 됨
- 자기 PC에서 EC2 접속
- 몽고 DB 설치
- VS code로 EC2 접속

EC2 공식 문서

[https://docs.aws.amazon.com/ko\\_kr/AWSEC2/latest/UserGuide/concepts.html](https://docs.aws.amazon.com/ko_kr/AWSEC2/latest/UserGuide/concepts.html)

EC2 시작 방법

[https://docs.aws.amazon.com/ko\\_kr/AWSEC2/latest/UserGuide/concepts.html#how-to-get-started](https://docs.aws.amazon.com/ko_kr/AWSEC2/latest/UserGuide/concepts.html#how-to-get-started)

# AWS 몽고DB 설치하기



# ASW EC2 리눅스 서버에 몽고DB설치하고 접속하기

---

- 첨부 파일 확인하기 : [aws\\_몽고유\\_설치\\_접속.pdf](#)

# 리눅스 터미널에서 AWS 접속하기

# AWS 서버 접속하기

리눅스 CLI 터미널에서 다음을 입력한다.

```
$ chmod 400 aaaaaa.pem (자신의 aws 인증키)
```

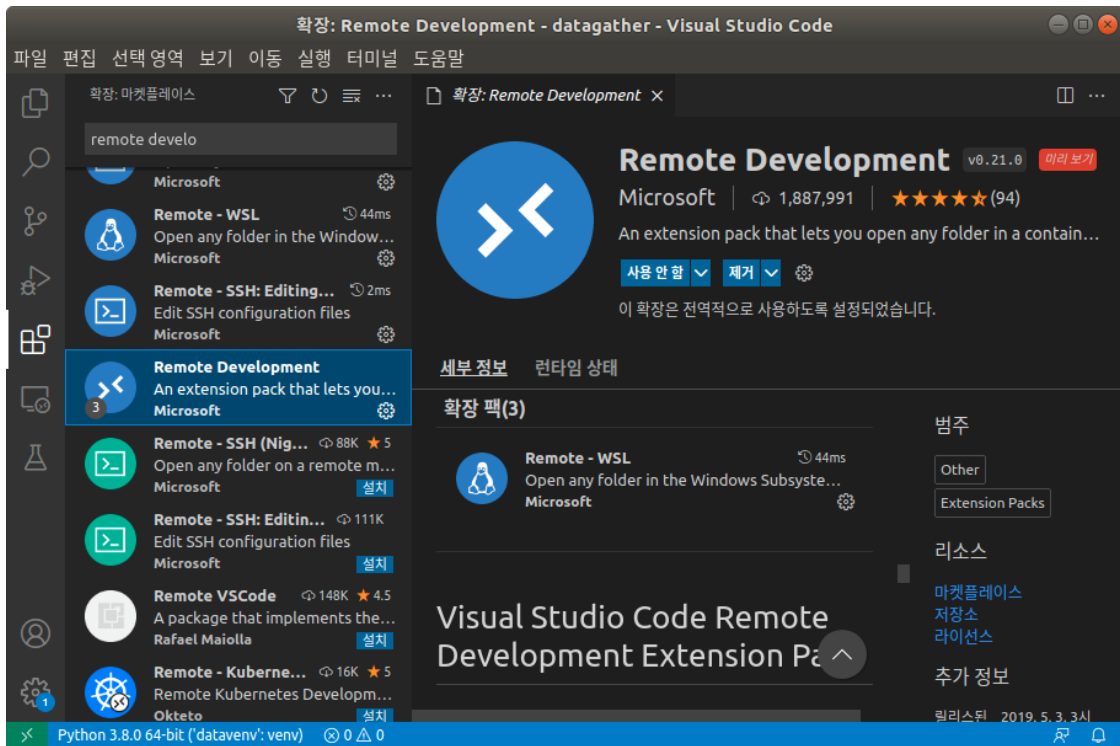
```
$ ssh -i aaaaaa.pem ubuntu@자신의 공인IP
```

인스턴스 (2) 정보							
인스턴스 필터링							
인스턴스 유형 ▼	상태 검사	경보 상태	가용 영역 ▼	퍼블릭 IPv4 DNS ▼	퍼블릭 IPv4 ... ▼	탄력적 IP ▼	탄력적 IP ▼
2.micro	✔ 2/2개 검사 통과...	경보 없음 +	ap-northeast-2a	ec2-52-78-231-126.ap-...	공인IP	-	-
2.micro	-	경보 없음 +	ap-northeast-2c	-	-	-	-

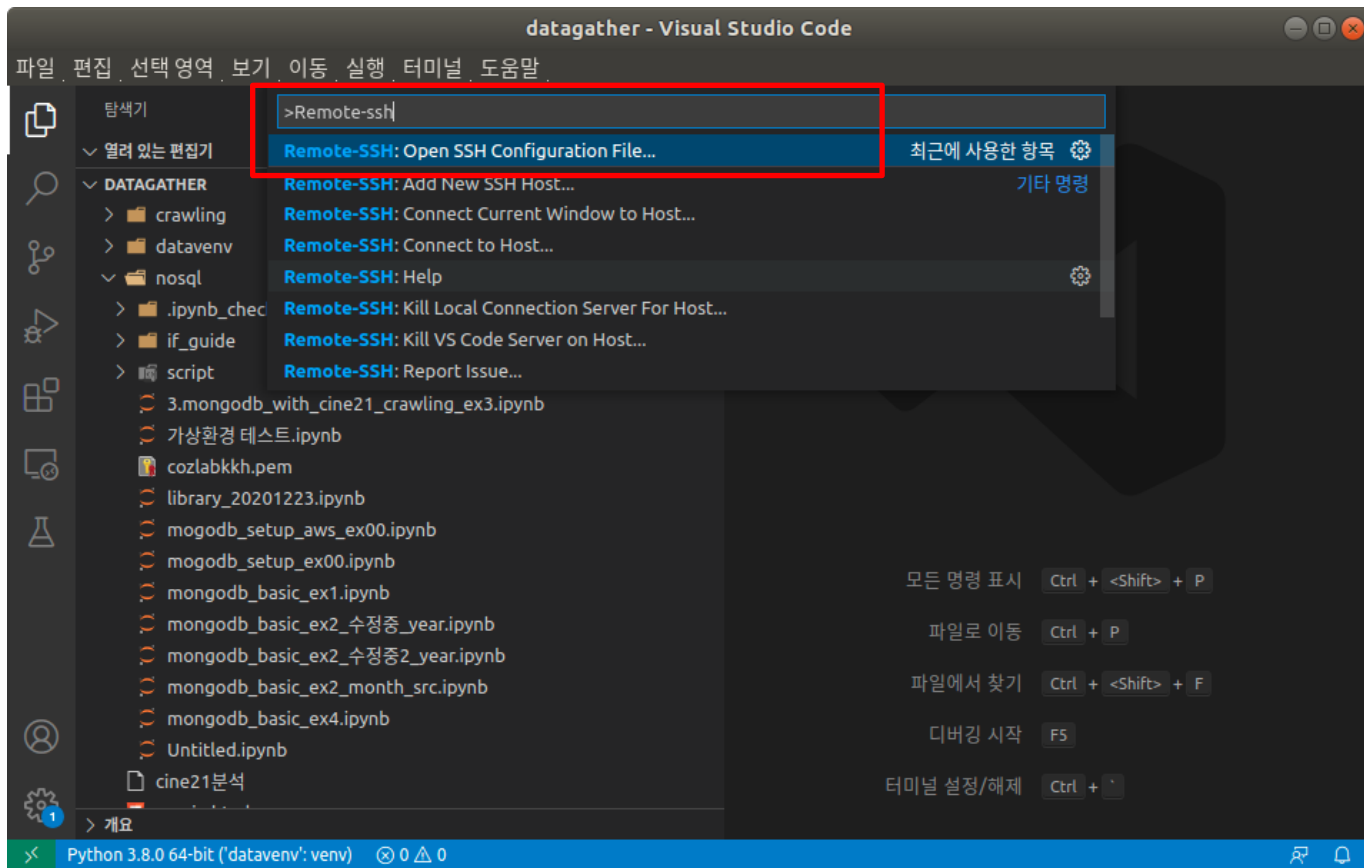
# Vs 코드로 AWS 접속하기

# VS-code AWS 접속하기

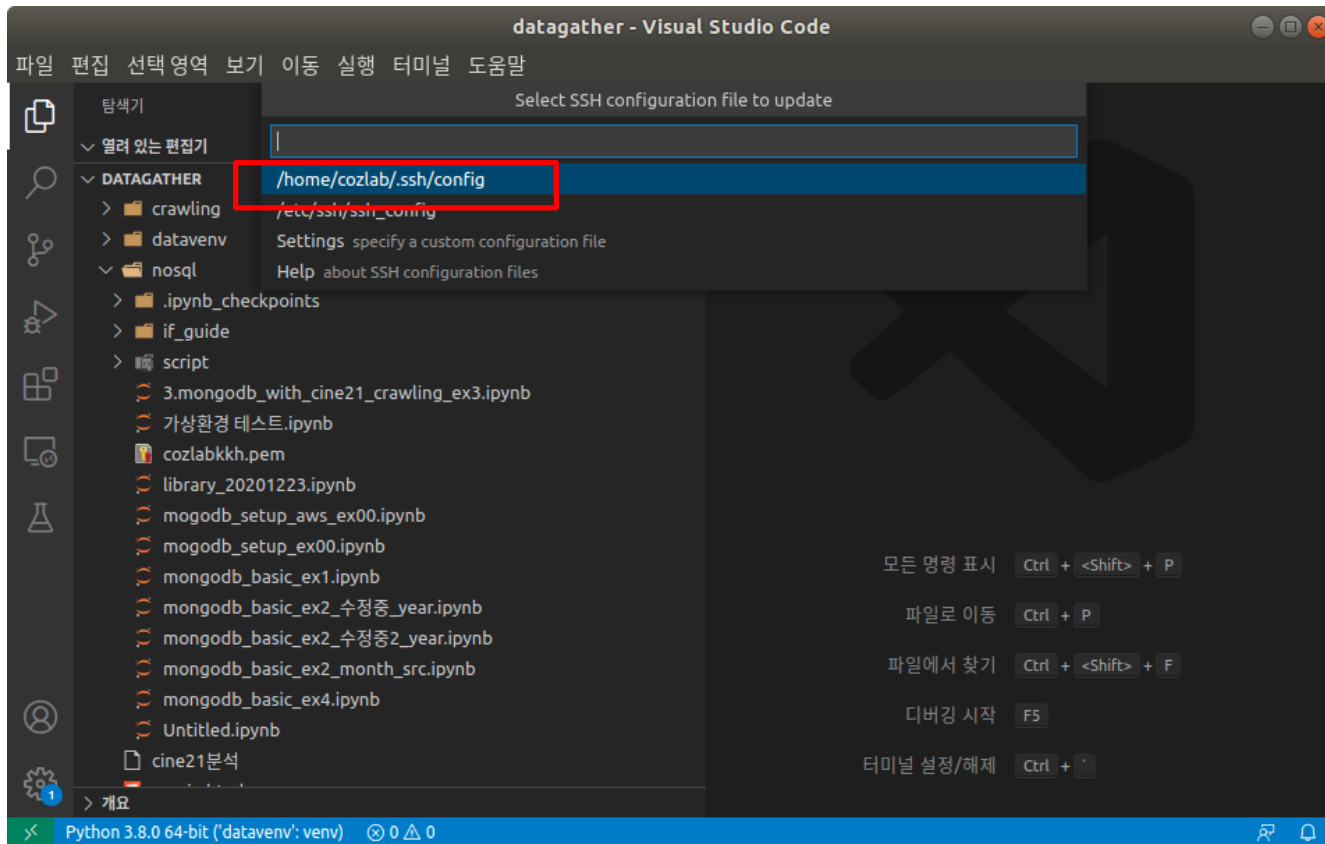
- 확장 프로그램 설치 : remote Development



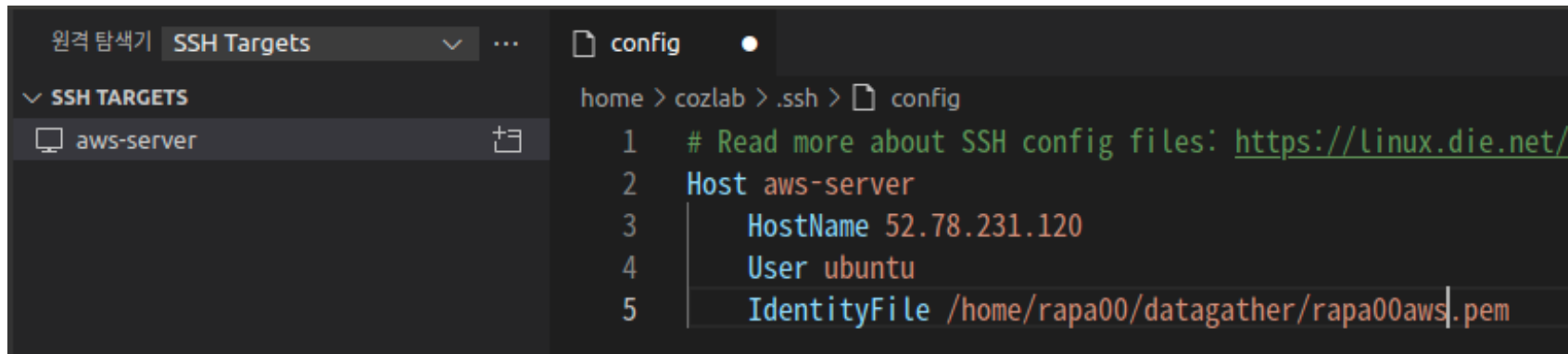
# VS-code AWS 접속하기



# VS-code AWS 접속하기



# VS-code AWS 접속하기



The screenshot shows the VS Code interface. On the left, the 'SSH Targets' sidebar is open, showing a target named 'aws-server'. The main editor displays the 'config' file in the '.ssh' directory. The file content is as follows:

```
home > cozlab > .ssh > config
1  # Read more about SSH config files: https://linux.die.net/
2  Host aws-server
3      HostName 52.78.231.120
4      User ubuntu
5      IdentityFile /home/rapa00/datagather/rapa00aws.pem
```

설정 값 예 :

Host aws-server

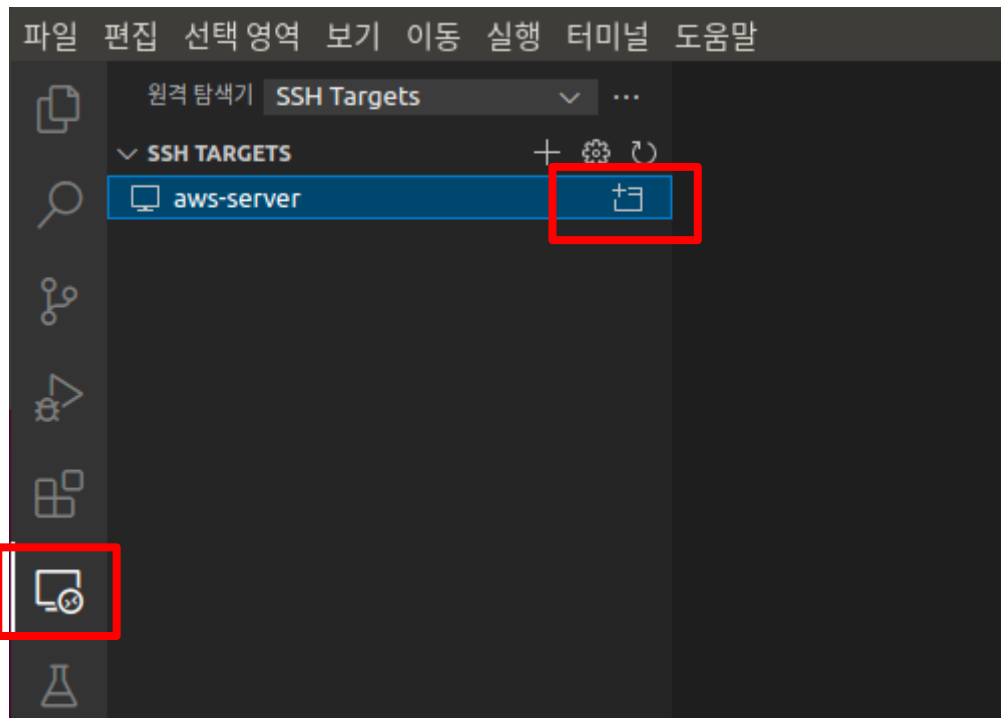
HostName 52.78.231.120 # 인스턴스 공인 IP

User ubuntu # 따로 지정하지 않았다면

IdentityFile /home/rapa00/datagather/rapa00aws.pem

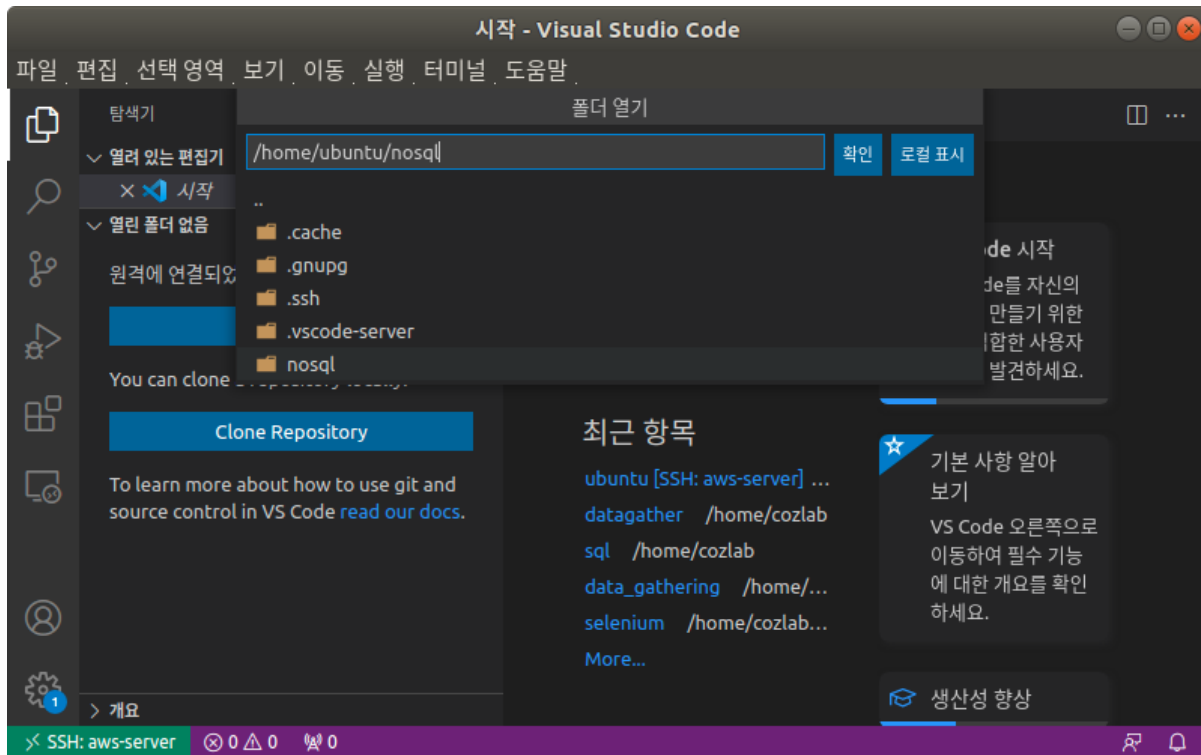


# VS-code AWS 접속하기



# VS-code AWS 접속하기

- open 할 폴더 선택하기



# VS-code AWS 접속하기

- aws nosql 폴더로 오픈 된 상태

