

AWS

Index

1. AWS

- 서비스 종류
- 회원가입
- 보안 설정
- Console

2. EC2

- 생성
- 접속

3. cron

- 스케줄링

AWS

Amazon Web Services

아마존 웹 서비스 (Amazon Web Services)

- 아마존닷컴의 클라우드 컴퓨팅 서비스

서비스 종류

- 컴퓨팅 (EC2)
- 서버리스 (Lambda)
- 스토리지 (S3)
- 데이터베이스 (RDS, Redis, DocumentDB)
- 라우팅 (Route 53)
- 머신러닝, 딥러닝



AWS 회원가입

- <https://aws.amazon.com/>
- 회원가입 (해외결제가 가능한 신용카드 필요)
- 루트 사용자 이메일 주소 : 로그인할 때 사용하는 ID
- 계정 이름 : AWS Console 접속 시 보이는 이름

- 이메일 인증코드 확인
- 관련 정보 입력 후 회원가입



새로운 AWS 계정으로 프리 티어 제품을 살펴보세요.

자세히 [알아보려면](#) aws.amazon.com/free를 방문하세요.



AWS에 가입

루트 사용자 이메일 주소
계정 복구 및 일부 관리 기능에 사용

AWS 계정 이름
계정의 이름을 선택합니다. 이름은 가입 후 계정 설정에서 변경할 수 있습니다.

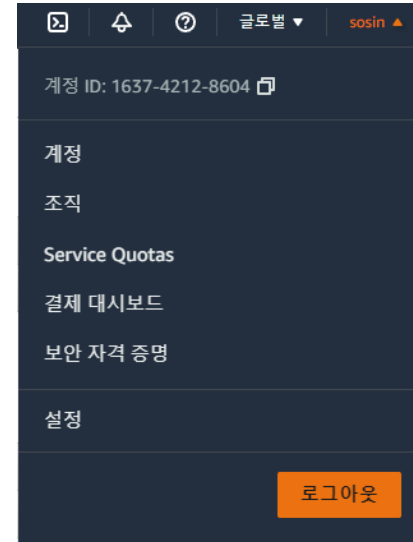
이메일 주소 확인

또는

기존 AWS 계정에 로그인

AWS 보안 설정

- 계정 보안을 위해 **MFA** 하는 것을 추천
- 우측 상단 닉네임 → 보안 자격 증명



- MFA 활성화를 통해 Google OTP 설치 후 설정

보안 자격 증명

이 페이지를 사용하여 AWS 계정의 자격 증명을 관리합니다. AWS IAM(Identity and Access Management) 사용자에게 대한 자격 증명을 관리하려면 [IAM 콘솔](#)을(를) 사용하십시오.

AWS 자격 증명 유형과 사용 방법에 대해 자세히 알아보려면 AWS 일반 참조의 [AWS 보안 자격 증명](#)을(를) 참조하십시오.

▲ 비밀번호

▼ 멀티 팩터 인증(MFA)

MFA를 사용하여 AWS 환경의 보안을 강화합니다. MFA 보호 계정에 로그인하려면 사용자 이름, 암호, MFA 디바이스에서 받은 인증 코드가 필요합니다.

[MFA 활성화](#)

▲ 액세스 키(엑세스 키 ID 및 비밀 액세스 키)

▲ CloudFront 키 페어

▲ X.509 인증서

▲ 계정 ID

AWS Management Console

- AWS 제품들을 한 곳에서 확인할 수 있는 콘솔
- 내가 접속한 서비스들을 한 눈에 확인할 수 있는 화면



EC2

Elastic Compute Cloud

Amazon Elastic Compute Cloud (EC2)

- 아마존의 컴퓨팅 자원을 빌려 사용하는 것
- 단순히 생각하면 컴퓨터를 하나 빌린다고 생각하면 됨
- 인스턴스: 가상 머신
- 실행 중인 서버에 대해 시간 당으로 비용을 지불하므로 Elastic (탄력적인)이라는 용어를 사용
- Region에 따라 VPC 구성을 다르게 해야 하기 때문에 여러 EC2를 활용하여 서버 구축할 시 Region을 고려해야 함



서울 ▼

sosin ▼

EC2 생성

- EC2 대시보드에서 인스턴스 화면으로 이동



- 인스턴스 시작 버튼을 눌러 인스턴스를 생성



인스턴스 설정

- 인스턴스 이름 설정

이름 및 태그 [정보](#)


이름

예: 내 웹 서버


추가 태그 추가

- AMI는 Ubuntu를 버전은 20.04 LTS, 18.04 LTS를 사용


Amazon Linux




Ubuntu




Windows



Red Hat



SUSE Linux



Ubuntu Server 20.04 LTS (HVM), SSD Volume Type

프리 티어 사용 가능 ▼

ami-0ea5eb4b05645aa8a (64비트(x86)) / ami-0678638ec320e38b6 (64비트(Arm))

가상화: hvm ENA 활성화됨: true 루트 디바이스 유형: ebs

- 인스턴스 유형은 프리티어 t2.micro (1 vCPU, 1 GiB 메모리)

t2.micro

프리 티어 사용 가능

패밀리: t2 1 vCPU 1 GiB 메모리 온디맨드 Linux 요금: 0.0144 USD 시간당

온디맨드 Windows 요금: 0.019 USD 시간당

키 페어 생성

- 인스턴스에 접속하기 위한 키페어를 생성

- 키페어는 1회만 다운로드 가능하므로 잘 관리해야 함.
해당 키 페어로 생성한 인스턴스는 키페어 분실 시 다시 접속하지 못하는 상황이 발생

- 우선은 VSCode의 OpenSSH를 통해 접속할 것이기 때문에 설정은 그대로, 이름만 입력 후 생성

키 페어 생성

×

키 페어를 사용하면 인스턴스에 안전하게 연결할 수 있습니다.

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

키 페어 이름

키 페어 이름 입력

이름은 최대 255개의 ASCII 문자를 포함할 수 있습니다. 선행 또는 후행 공백은 포함할 수 없습니다.

키 페어 유형

☒ RSA

RSA 암호화된 프라이빗 및 퍼블릭 키 페어

☐ ED25519

ED25519 암호화된 프라이빗 및 퍼블릭 키 페어(Windows 인스턴스에는 지원되지 않음)

프라이빗 키 파일 형식

☒ .pem

OpenSSH와 함께 사용

☐ .ppk

PuTTY와 함께 사용

취소

키 페어 생성

EC2 생성

- 현재 설정된 보안그룹은 없기 때문에 기본값으로 설정
- 스토리지도 기본값 (8 Gib)으로 설정
- 인스턴스 시작을 클릭하면 인스턴스가 생성됨

인스턴스 개수 [정보](#)

1

[소프트웨어 이미지\(AMI\)](#)

Canonical, Ubuntu, 20.04 LTS, ... [더 보기](#)
ami-0ea5eb4b05645aa8a

[가상 서버 유형\(인스턴스 유형\)](#)



t2.micro

[방화벽\(보안 그룹\)](#)

새 보안 그룹

[스토리지\(블륨\)](#)

1개의 볼륨 - 8GiB

 프리 티어: 첫 해에는 월별 프리 티어 AMI에 대한 t2.micro(또는 t2.micro를 사용할 수 없는 리전의 t3.micro) 인스턴스 사용량 750시간, EBS 스토리지 30GiB, IO 2백만 개, 스냅샷 1GB, 인터넷 대역폭 100GB가 포함됩니다. 

취소

인스턴스 시작

EC2 접속

- EC2 대시보드에서 인스턴스 클릭 시 인스턴스가 생성된 것을 확인할 수 있음
- 기본적으로 생성 시 키페어를 통해 어디서든 접속할 수 있도록 설정했기 때문에, 별다른 추가 설정 없이 VSCode에서 EC2로 접속을 시도

- 단, 어떤 컴퓨터에 접속할 것인지는 알아야하기 때문에, 퍼블릭 IPv4 주소를 복사

인스턴스 (1/1) 정보

🔍 검색

| <input checked="" type="checkbox"/> | Name ▾ | 인스턴스 ID | 인스턴스 상태 ▾ |
|-------------------------------------|---------|---------------------|-----------|
| <input checked="" type="checkbox"/> | 소신 컴퓨터1 | i-067549e7d4b4bacc3 | 🟢 실행 중 🔍 |

i-067549e7d4b4bacc3에 대한 인스턴스 요약 정보

less than a minute 전에 업데이트됨

인스턴스 ID

📋 i-067549e7d4b4bacc3

IPv6 주소

-

퍼블릭 IPv4 주소

📋 3.39.11.41 | [개방 주소법](#) 🔗

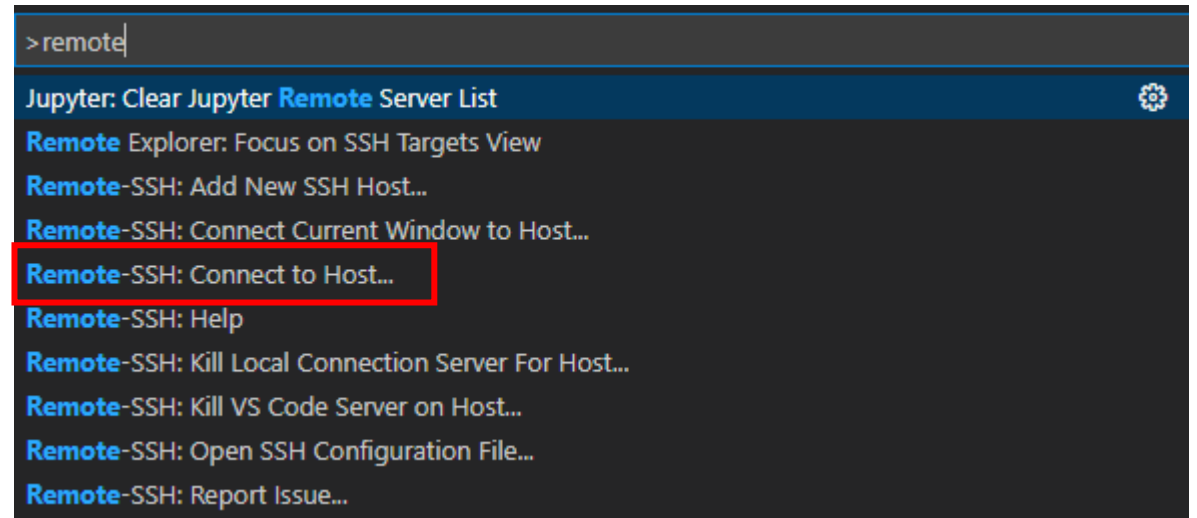
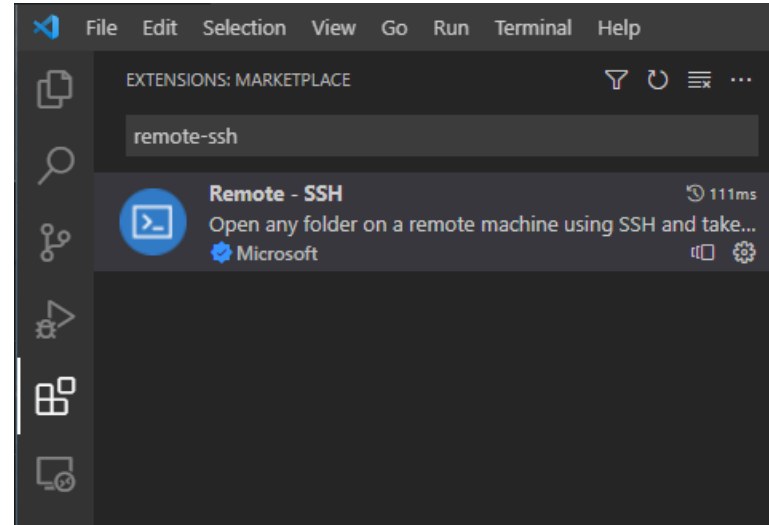
인스턴스 상태

🕒 대기 중

Remote-SSH 설치

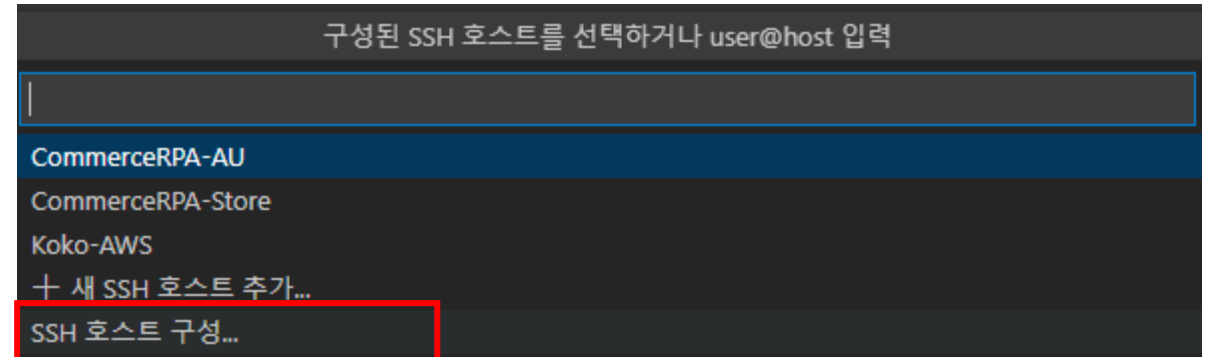
- SSH(Secure Shell) : 원격 호스트에 접속하기 위해 사용되는 보안 프로토콜
- VSCode Extensions에서 remote-ssh 검색 후 설치

- Command Palette 열기 (CTRL+SHIFT+P)
- remote-ssh: Connect to Host... 선택

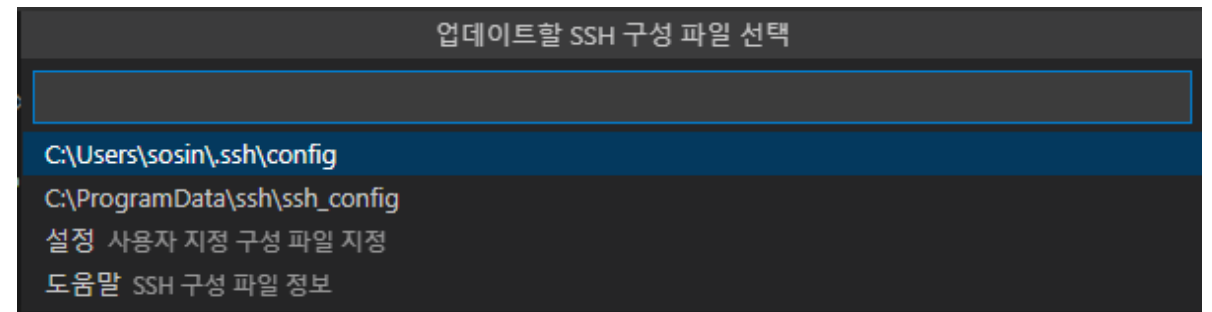


Remote-SSH 설정

- Configure SSH Hosts... 선택



- C:\users\{USER_NAME}\.ssh\config 선택

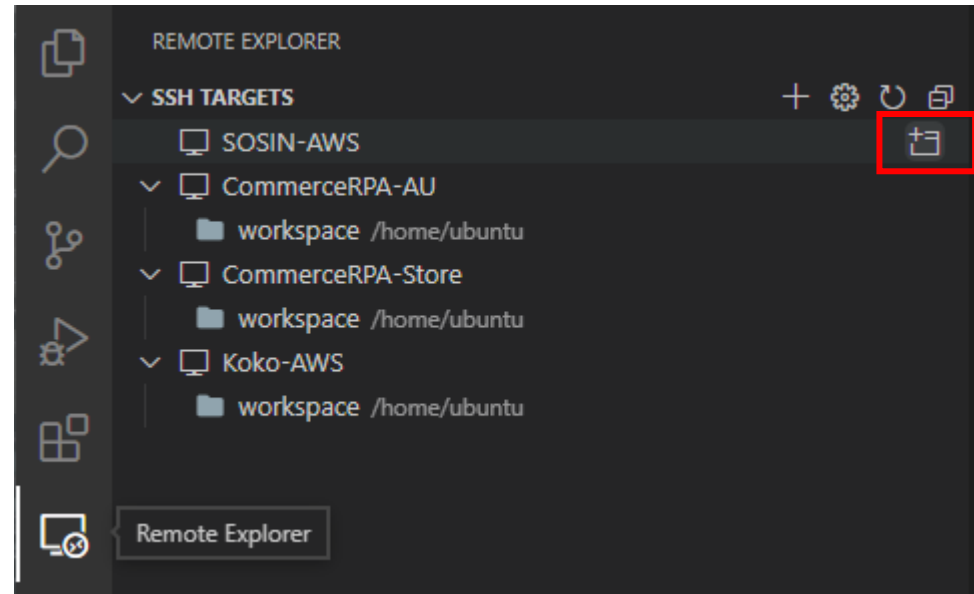


Host 정보 입력

- Host: 서버의 별칭 (스페이스바는 적용 X)
- HostName: 접속할 서버의 주소
- User: 접속하는 서버 계정이름
- IdentityFile: 접속할 서버의 키페어 파일 위치
- 변경내용 저장

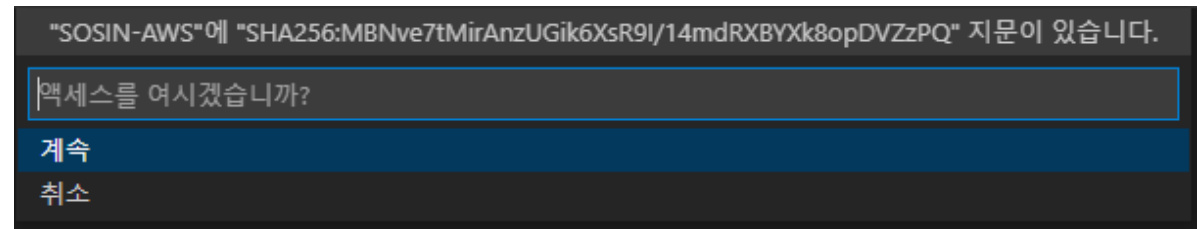
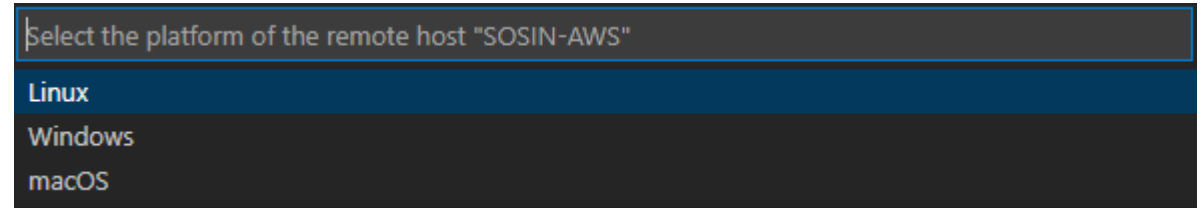
```
# Read more about SSH config files: https://linux.die.net/man/5/ssh\_config  
Host SOSIN-AWS  
    HostName 3.39.11.41  
    User ubuntu  
    IdentityFile ~/aws/sosin_aws.pem
```

- [Remote Explorer] 클릭
- 접속할 서버에 커서를 올리고 새 창 버튼을 클릭

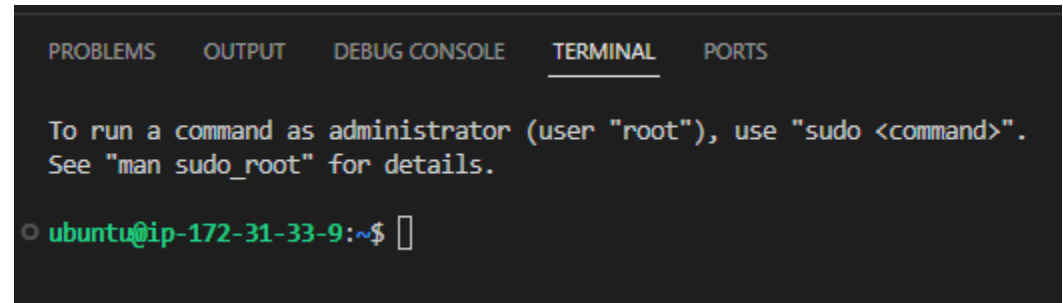


Host 정보 입력

- Linux
- Continue



- 서버 접속 성공



탄력적 IP

- 서버를 재부팅하거나 재시작해도 IP 주소를 변하지 않게 유지

cron

Job Scheduler on Unix-like OS

cron

- 유닉스 계열 컴퓨터 운영체제의 시간 기반 잡 스케줄러

- crontab -e 명령어로 설정파일에 접근
- 최초 파일 접근 시 editor 설정 (vim.tiny)

- select-editor 입력 시 editor 변경 가능

- 시간 설정 참고 사이트
- <https://crontab.guru/>

- 명령어 예시
- */10 * * * * /usr/bin/python
/home/ubuntu/workspace/main.py

🎯 A crontab file has five fields for specifying:

```
* * * * * command to be executed
- - - - -
| | | | |
| | | | +----- **DAY OF WEEK** (0-6) (Sunday=0)
| | | +----- **MONTH** (1-12)
| | +----- **DAY OF MONTH** (1-31)
| +----- **HOUR** (0-23)
+--- **MINUTE** (0-59)
```