

[12/08] AWS

## 1) 아마존 웹 서비스 회원가입 및 결제

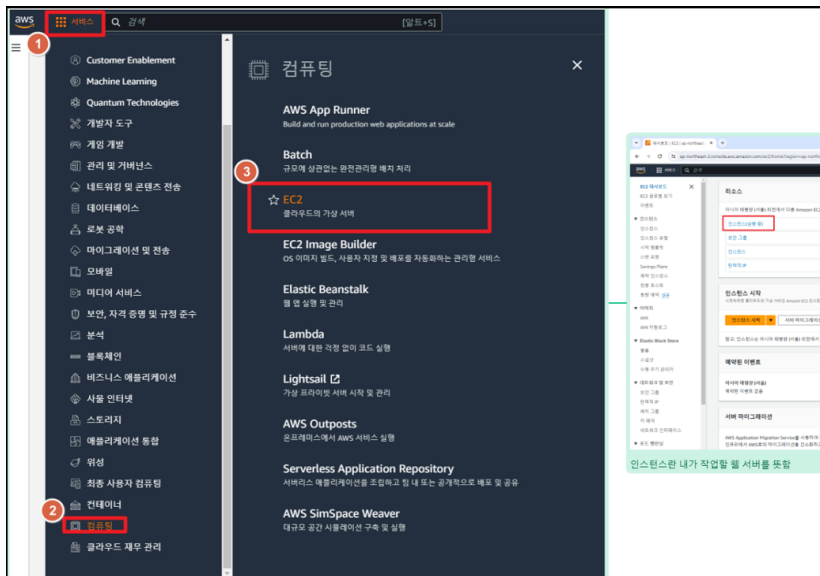
- 100원 결제 테스트됨

## 2) 인스턴스 생성 : 작업할 웹 서버

- 지역 : 서울로 변경



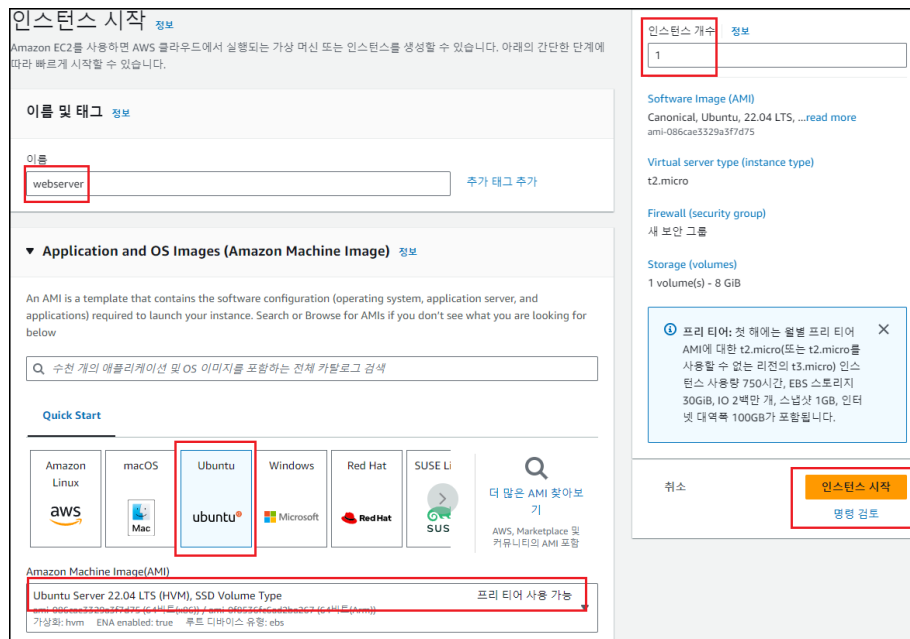
### 2-1) 인스턴스 시작



인스턴스 없음  
이 리전에는 인스턴스가 없음  
인스턴스 시작

### 2-2) 이름 및 태그 : 영문사용 권장

### 2-3) Application and OS Images (Amazon Machine Image) : ubuntu, 프리티어(무료)



- 갯수 1

2-4) 인스턴스 유형 : 다른 버전있으나 cpu 갯수와 메모리 크기 차이

2-5) 키페어 로그인 : 새 키 페어 생성

키 페어 생성

키 페어 이름  
키 페어를 사용하면 인스턴스에 안전하게 연결할 수 있습니다.  
webServer  
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

- 키 페어 이름 : webServer
- 생성시 : webServer.pem 파일이 자동 다운로드됨

2-6) 네트워크 설정

☒ Allow SSH traffic from the Internet  
Helps you connect to your instance  
☒ 인터넷에서 HTTPS 트래픽 허용  
예를 들어 웹 서버를 생성할 때 엔드포인트  
☒ 인터넷에서 HTTP 트래픽 허용  
예를 들어 웹 서버를 생성할 때 엔드포인트

- 섯다 허용

2-7) 스토리지 구성

1x 25

- 마우스로 클릭할 것
- 이후 우측 주황색 버튼(인스턴스 시작 클릭)

2-8) 완료

aws 서비스 검색

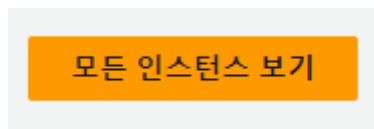
EC2 > 인스턴스 > 인스턴스 시작

☒ 성공  
인스턴스를 시작했습니다. (i-0b6c47b0c0557f01a)

▶ 로그 시작

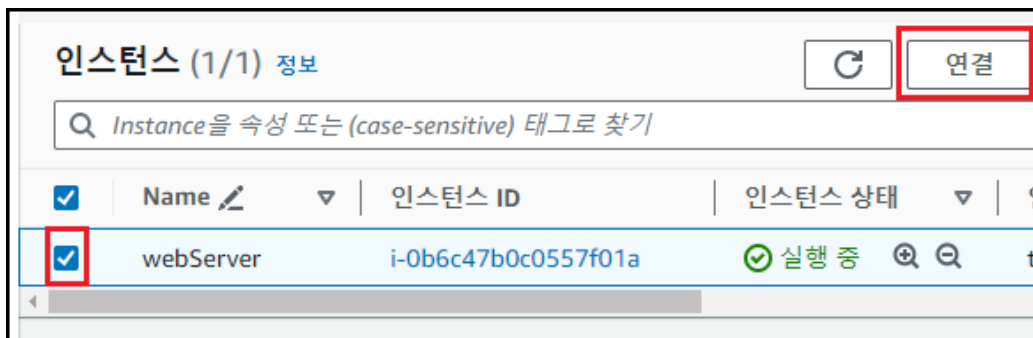
### 3) AWS 에서 ubuntu 사용

#### 3-1) 모든 인스턴스 보기



- 2-9 까지 완료한 화면에서 확인됨

#### 3-2) 연결



#### 3-3) ubuntu 의 화면

```
aws | 서비스 | 검색 | [알트+S]
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1012-aws x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Fri Dec  8 04:33:57 UTC 2023

System load:  0.0      Processes:      97
Usage of /:   6.4% of 24.05GB  Users logged in:  0
Memory usage: 21%      IP4 address for eth0: 172.31.40.243
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-40-243:~$
```

- GNU : 오픈 소스 공개하기 위한 규칙(그누 정책)
- Processes : ubuntu 적상 작동시 기본 처리기 갯수 97개

### 3-4) ubuntu 입력

```
ubuntu@ip-172-31-40-243:~$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages have been kept back:
  kpartx linux-aws linux-headers-aws linux-image-aws multipath-tools
The following packages will be upgraded:
  apparmor apt apt-utils bind9-dnsutils bind9-host bind9-libs cloud-init
  libapt-pkg6.0 libc-bin libc6 libcurl3-gnutls libcurl4 libgnutls30
  libpython3.10 libpython3.10-minimal libpython3.10-stdlib libsgutils2
  python3-cryptography python3-software-properties python3-urllib3
  vim-runtime vim-tiny xxd
66 upgraded, 0 newly installed, 0 to remove and 8 not upgraded.
Need to get 65.2 MB of archives.
After this operation, 4968 kB disk space will be freed.
Do you want to continue? [Y/n]
```

- sudo apt-get update
- sudo apt-get upgrade
  - dependency : 어떤 작업을 하기 위한 필수 파일, 라이브러리, 함수가 담긴 단어
- y
- date
  - 입력후 시간 확인, 현재 UTC 기준(전세계 표준 시간)

### 4) 인스턴스 종료

**인스턴스 (1) 정보**

Instance를 속성 또는 (case-sensitive) 태그로 찾기

Name	인스턴스 ID	인스턴스 상태
webServer	i-0b6c47b0c0557f01a	실행 중

**인스턴스 상태**

- 인스턴스 중지
- 인스턴스 시작
- 인스턴스 재부팅
- 인스턴스 최대 절전 모드
- 인스턴스 종료

**종료 인스턴스?**

⚠ EBS가 지원하는 인스턴스의 경우, 인스턴스가 종료될 때 기본적으로 루트 EBS 볼륨이 삭제됩니다. 로컬 드라이브의 스토리지는 모두 손실됩니다.

이 인스턴스를 종료하시겠습니까?

인스턴스 ID	종료 방지
i-01e58f9607756102e (webServer)	Disabled

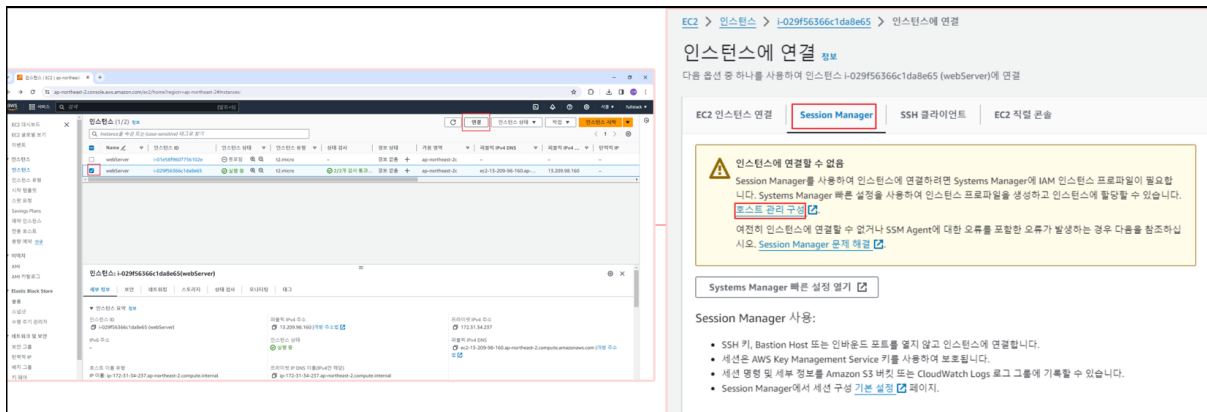
인스턴스 종료를 확인하려면 아래의 종료 버튼을 선택합니다. 종료 방지가 활성화된 인스턴스는 종료되지 않습니다. 인스턴스 종료는 실행 취소할 수 없습니다.

취소 **종료**

- 인스턴스가 구동중일 때 인스턴스에 연결된 모든 작업은 이루어지지 않음
  - 키 페어 삭제, 보안 그룹 삭제 등

## 5) 인스턴스 세부 사항

### 5-1) Session manager : AWS document 보는 법



The screenshot shows the AWS Management Console interface. On the left, there's a navigation pane with '인스턴스' (Instances) selected. The main area displays a list of EC2 instances. One instance, 'webServer', is highlighted. To the right, a detailed view of the '인스턴스에 연결' (Connect to Instance) page is shown. This page has tabs for 'EC2 인스턴스 연결', 'Session Manager', 'SSH 클라이언트', and 'EC2 직렬 콘솔'. The 'Session Manager' tab is active, showing a warning message about connecting to the instance using Session Manager. It states that Session Manager requires IAM instance profiles and provides links for '호스트 관리 구성' (Host Management Configuration) and 'Session Manager 문제 해결' (Session Manager Troubleshooting). Below the warning, there's a section for 'Systems Manager 빠른 설정 열기' (Open Systems Manager Quick Setup) and a 'Session Manager 사용' (Use Session Manager) section with a list of requirements: SSH key, Bastion Host or inbound port, AWS Key Management Service key, Amazon S3 bucket for CloudWatch Logs, and Session Manager configuration page.

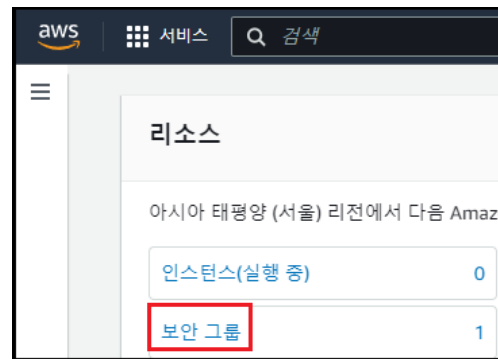
### 5-2) SSH 클라이언트 : 외부 접속과 관련, 포트번호 설정



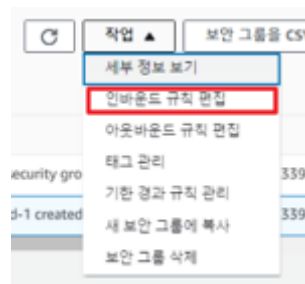
The screenshot shows the AWS Management Console interface, specifically the '인스턴스에 연결' (Connect to Instance) page for the 'webServer' instance. The 'SSH 클라이언트' (SSH Client) tab is active. The page provides instructions for connecting to the instance via SSH. It lists the instance ID 'i-029f56366c1da8e65 (webServer)' and provides a list of steps: 1. Open the SSH client. 2. Find the private key file. 3. Run the command 'chmod 400 webServer.pem' if needed. 4. Use the public DNS (ec2-13-209-98-160.ap-northeast-2.compute.amazonaws.com) to connect. An example command is provided: 'ssh -i "webServer.pem" ubuntu@ec2-13-209-98-160.ap-northeast-2.compute.amazonaws.com'. A note at the bottom states: '참고: 대부분의 경우 주어진 사용자 이름은 정확합니다. 하지만 AMI 사용 지침을 읽고 AMI 소유자가 기본 AMI 사용자 이름을 변경했는지 확인하십시오.'

### 5-3) EC2 직렬 콘솔 : 의미 없음

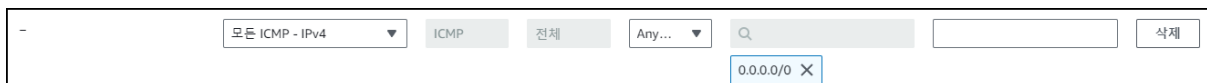
## 6) 보안 규칙



### 6-1) 인바운드 규칙 편집 : 내가 만든 ubuntu에 접속 허가를 해주는 것



보안그룹 이름 : launch-wizard-1



- 규칙추가 > 모든 ICMP -IPv4 / Anywhere-IPv4 > 규칙저장

### 6-2) 인스턴스 재부팅



- 서비스 > EC2 > 인스턴스 재부팅
- 보안 규칙 바뀔때마다 재부팅 권장

7) 탄력적 IP : 설정시 모든 사람이 들어올 수 있어 요금 부과됨, 유동 IP 를 고정 IP로 설정 가능

#### 7-1) 탄력적 IP 주소 할당

리소스

아시아 태평양 (서울) 리전에서 다음 Amazon

인스턴스(실행 중)	1
보안 그룹	2
인스턴스	1
탄력적 IP	0

탄력적 IP 주소 할당

✓

탄력적 IP 주소가 할당되었습니다.  
탄력적 IP 주소 3.39.169.41

#### 7-2) 탄력적 IP 주소 연결

탄력적 IP 주소: 3.39.169.41

리소스 유형

탄력적 IP 주소를 연결할 리소스의 유형을 선택합니다.

☒ 인스턴스
 ☐ 네트워크 인터페이스

⚠ 탄력적 IP 주소가 이미 연결된 인스턴스와 탄력적 IP 주소를 연결하면 이전에 연결된 탄력적 IP 주소가 연결 해제되지만 주소는 여전히 계정에 할당됩니다. 자세히 알아보기

프라이빗 IP 주소를 지정하지 않으면 탄력적 IP 주소가 기본 프라이빗 IP 주소와 연결됩니다.

인스턴스

재연결

이 리소스에 연결되어 있는 탄력적 IP 주소를 다른 리소스에 재연결할 수 있는지 여부를 지정합니다.

☐ 이 탄력적 IP 주소를 재연결하도록 허용

취소

연결

↻

작업 ▲

탄력적 IP 주소

세부 정보 보기

탄력적 IP 주소 릴리스

탄력적 IP 주소 연결

탄력적 IP 주소 연결 해제

역방향 DNS 업데이트

전송 활성화

전송 비활성화

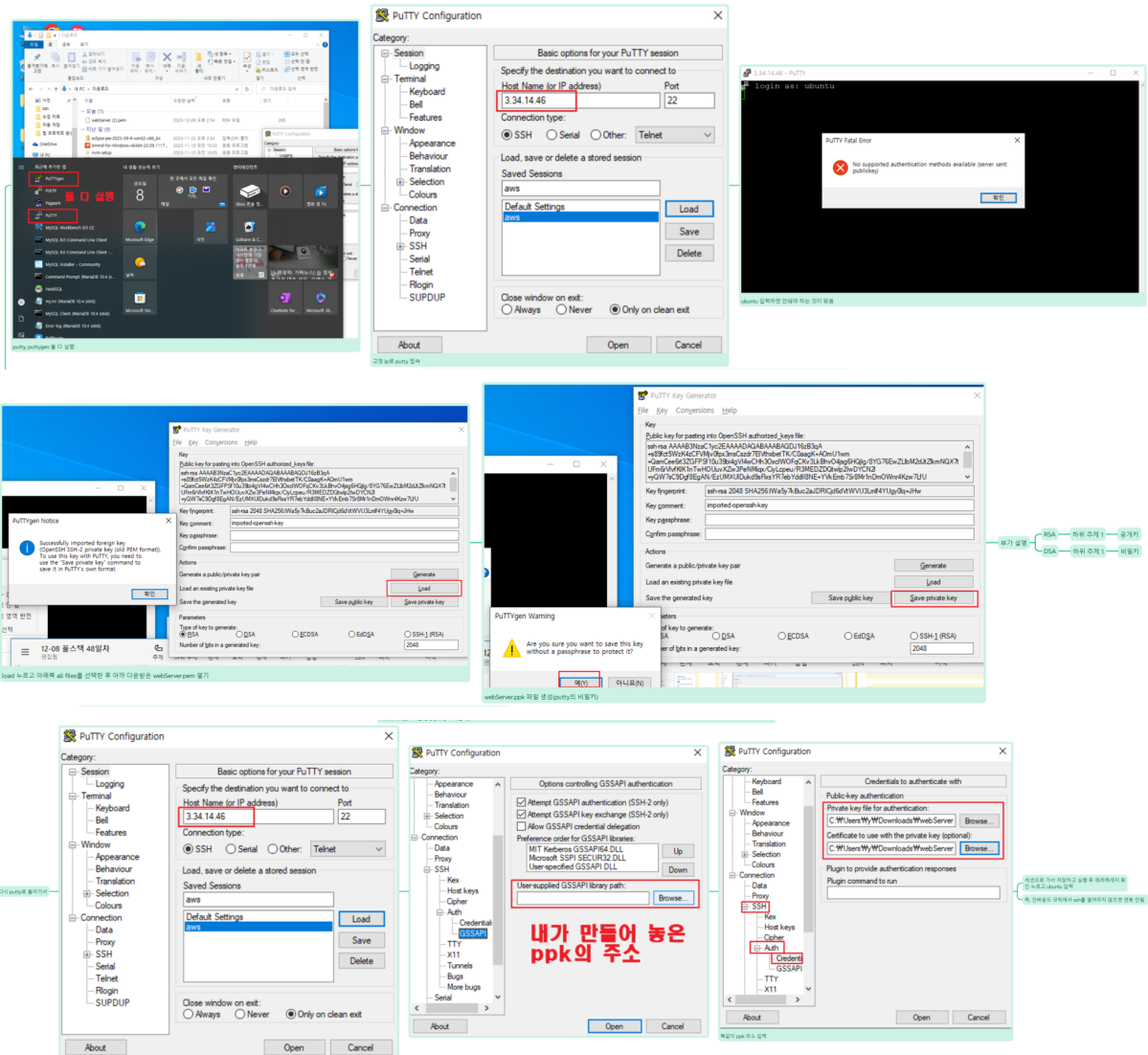
전송 수락

역방향

-

- 인스턴스 : 새로고침 버튼 누르면 자동으로 목록이 뜸
- 프라이빗 IP 주소 : 자동으로 목록 뜸
- 재연결 : 클릭 X
- 이후 인스턴스 재시작시 고정 IP로 변동되어 있음

## 8) Putty



## 9) 정리





## 1) AWS 사용을 위한 준비 작업

- root 로그인 상태에서의 진행 : gkmh0925@gmail.com

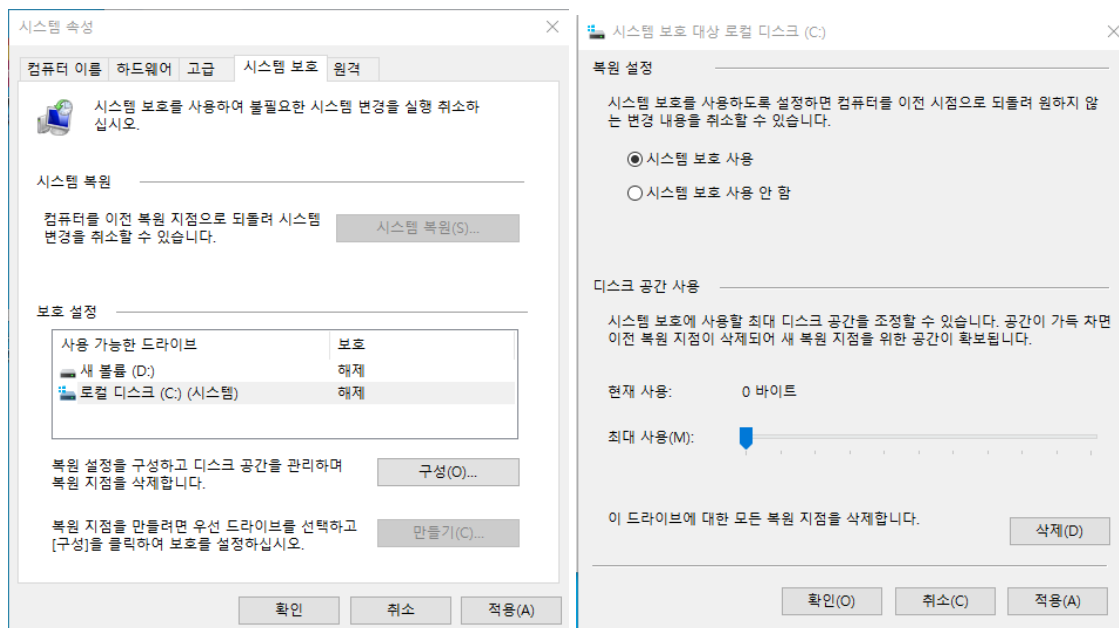
### 1-1) 암호화 방식

- RSA 암호화 방식을 AWS 에서 채택해서 사용할 예정

종류	RSA	DSA
암호화 원리	소인수 분해의 어려움에 기반한다	Discrete Logarithm 문제의 어려움에 기반한다.
Key 생성	RSA는 Key 생성이 느리다.	ElGamal 비슷하게 사용하므로, RSA보다 Key 생성이 빠르다
Encryption (Verifying)	Encryption 이 DSA보다 빠르다	Encryption이 RSA보다 느리다
Decryption (Signing)	Decryption은 DSA보다 느리다	Decryption은 RSA보다 빠르다
비고	Encryption (검증)에 최적화	Decryption (서명)에 최적화

- RSA(Rivest-Shamir-Adleman) : 비대칭키, 암호키와 복호키(해독)가 다름
- DAS(Digital Signature Algorithm) : 대칭키, 암호를 여는 열쇠가 하나만, 공인인증서

### 1-2) 복원 지점 만들기 : 설정 > 복원 지점 만들기 검색



- 구성 : 이미지파일화해서 저장
- 최대 사용 : 4~6% 설정(권장)
- 만들기 > 이름 설정 (기억할 시점) > 시스템 복원
  - 윈도우 업데이트 완료이후 사용 권장

## 2) AWS 에서의 사용 설정

리소스			EC2 Global view		
아시아 태평양 (서울) 리전에서 다음 Amazon EC2 리소스를 사용하고 있음:					
인스턴스(실행 중)	0	로드 밸런서	0	배치 그룹	0
보안 그룹	2	볼륨	2	스냅샷	0
인스턴스	2	전용 호스트	0	키 페어	1
탄력적 IP	1	Auto Scaling 그룹	0		

- 3) 키페어 / 4)보안그룹 / 5,6)인스턴스 / 7)탄력적 IP

### 3) 키페어

### 키 페어

프라이빗 키와 퍼블릭 키로 구성되는 키 페어는 인스턴스에 연결할 때 자격 증명을 증명하는 데 사용하는 보안 자격 증명 세트입니다.

이름

이름에는 최대 255개의 ASCII 문자가 포함됩니다. 앞 또는 뒤에 공백을 포함할 수 없습니다.

키 페어 유형 **정보**

☒ RSA
 ☐ ED25519

프라이빗 키 파일 형식

☒ .pem  
OpenSSH와 함께 사용
 ☐ .ppk  
PuTTY와 함께 사용

태그 - 선택 사항

리소스에 연결된 태그가 없습니다.

[새로운 태그 추가](#)

최대 50개의 태그를 더 추가할 수 있습니다.

- 순서 : 키페어 > 키페어 생성 > 키페어 생성
  - (1) 이름 : ex) 12-11kmh
  - (2) 키페어 유형 : RSA
  - (3) 프라이빗 키 파일 형식 : .pem
- 설명
  - pem : 리눅스와 기타 설정 / ppk : 윈도우만, 타인 전송 어려움
  - 프라이빗 키를 pem 방식으로 저장하여 향후 putty 와 연결

#### 4) 보안그룹

- 순서 : 보안그룹 > **보안그룹 생성** > **보안그룹 생성**

##### 4-1) 기본세부정보

- (1) 보안그룹 이름 : 12-11kmh
- (2) 설명 : exm

##### 4-2) 인바운드 규칙 : 들어올 때

- (3) 규칙 추가

- 사용자 지정 TCP / 8080 / 사용자 지정 / tomcat
- HTTP / 사용자 지정
- HTTPS / 사용자 지정
- SSH / 사용자 지정
- MySQL/Aurora / 사용자 지정
- 모든 ICMP - IPv4 / Anywhere IPv4
  - Anywhere IP4 : 0.0.0.0/0 - 전세계 어디에서든 접근 가능
  - 인바운드 규칙의 유형 : 구분을 위한 이름일 뿐 포트범위가 중요
  - ICMP(Internet Control Message Protocol, ICMP) : 핑 주고 받음
- 내 IP 로 할 경우 외부에서 접속 불가
  - 모뎀(강의장)에 들어오는 IP : 112.221.230.98 / 32

```
C:\Users\merimi>ping 183.109.99.20
Ping 183.109.99.20 32바이트 데이터 사용:
183.109.99.20의 응답: 바이트=32 시간<1ms TTL=64
183.109.99.20의 응답: 바이트=32 시간<1ms TTL=64
183.109.99.20의 응답: 바이트=32 시간<1ms TTL=64
183.109.99.20의 응답: 바이트=32 시간<1ms TTL=64

183.109.99.20에 대한 Ping 통계:
패킷: 보냄 = 4, 받음 = 4, 손실 = 0 (0% 손실).
왕복 시간(밀리초):
최소 = 0ms, 최대 = 0ms, 평균 = 0ms
```

##### 4-3) (4) 아웃바운드 규칙 : 내보낼 때

- 포트범위 전체 / 프로토콜 전체 / 대상 0.0.0.0/0

## 5) 인스턴스

- 순서 : 인스턴스 > **인스턴스 시작** > **인스턴스 시작**

### 5-1) 이름 및 태그 : (1) 아무거나, 변경가능 ex)12-11kmh

### 5-2) Application and OS Images (Amazon Machine Image)

- (2) Ubuntu
- (3) Ubuntu server 20.04 LTS 선택
  - 서버 하나기준 750시간 무료(1/n)
  - Ubuntu server 22.04 LTS 선택시 추후 DB 미설정 가능

### 5-3) (4) 키페어 : 앞서 설정한 12-11kmh 로 설정

### 5-4) 네트워크 설정

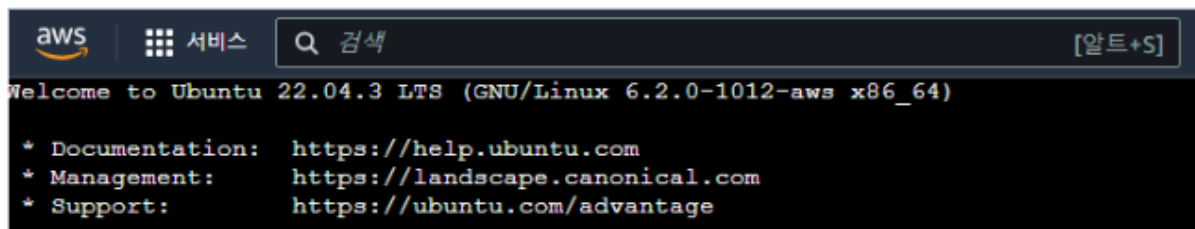
- (5) 기존 보안그룹 선택
- (6) 보안그룹생성 : launch-wizard-1

### 5-5) 스토리지 구성

- (7) 25 GiB

## 6) 인스턴스 연결

- 순서 : 해당하는 인스턴스 체크후 > 인스턴스 상태 > 인스턴스 시작 > F5 > 연결 > 연결 : ubuntu 서버 열림
  - 연결시 사용자 이름 : 현재의 root 는 ubuntu / AWS의 경우 ec2-user
- 인스턴스 중지 : 퍼블릭 IP 에 주소가 없어짐, 프라이빗은 존재함
- 인스턴스 종료 : 인스턴스 삭제
- 설명
  - 사설 : 외부 인터넷과 접속이 불가, 모두에게 공인 IP 부여 불가능, 개인도 5~6개 사용
    - 단, 사설을 공인처럼 바꾸는게 공유기(Router)
  - 공인 : 외부 인터넷과 접속이 가능



Public IPs 와 Private IPS가 확인됨 / ICMP 열었기 때문에 ping 주고 받기 가능해짐

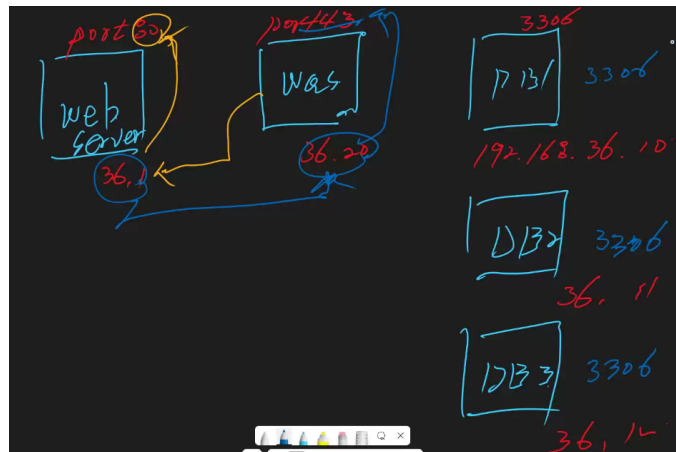
- (1) Public IPs 를 복사해둬, 추후 8-2)에서 사용
- 부가설명
  - AWS - EC2 를 생성시, 외부와 접속가능한 IP 하나는 필요 이것이 퍼블릭 IP
    - 퍼블릭 IP 주소라고 확인되지만 사설이다.
    - 퍼블릭 IP : 외부와 통신용, 공인, AWS가 배정해줌
- 퍼블릭 IP를 탄력적 IP 주소 할당시 : 고정 IP 가 됨, 여전히 공인

## 7) 탄력적 IP<sup>1</sup>

- 순서 : 탄력적 IP > **탄력적 IP 주소 할당** > **할당** > 작업 > 탄력적 IP 주소 연결 > **연결**
- (1) 인스턴스 : 기존 설정한 것 선택
- (2) 프라이빗 아이피 주소 : 자동으로 선택됨
  - 인스턴스 재부팅 권장
  - 프라이빗 아이피는 AWS내에서 다른 IP들과 구분하기 위함
  - 주의 : 미설정시 과금!!
- 종료 : 작업 > 탄력적 IP 주소 릴리즈 > 전체 삭제됨
- 설명
  - 탄력적 IP 할당을 통해 변동적인 공인 IP 주소를 고정적인 공인 IP 주소로 변경
  - 이후 인스턴스 연결에서 퍼블릭 IP 주소가 바뀐것이 확인됨
    - 인스턴스 연결시 사용자이름 ubuntu 가 쓰는 것이 루트명(사용자명)과 동일
    - AWS에서는 로그인 하지않고 우분투 사용가능

### 7-1) 부가 설명

- DB 와 스토리지는 보안규칙에서 적용
- 서버가 4개면 4개가 필요
- 여러 DB의 포트 번호는 3306를 중복 사용 가능, IP 주소는 다르게 할당해야 함

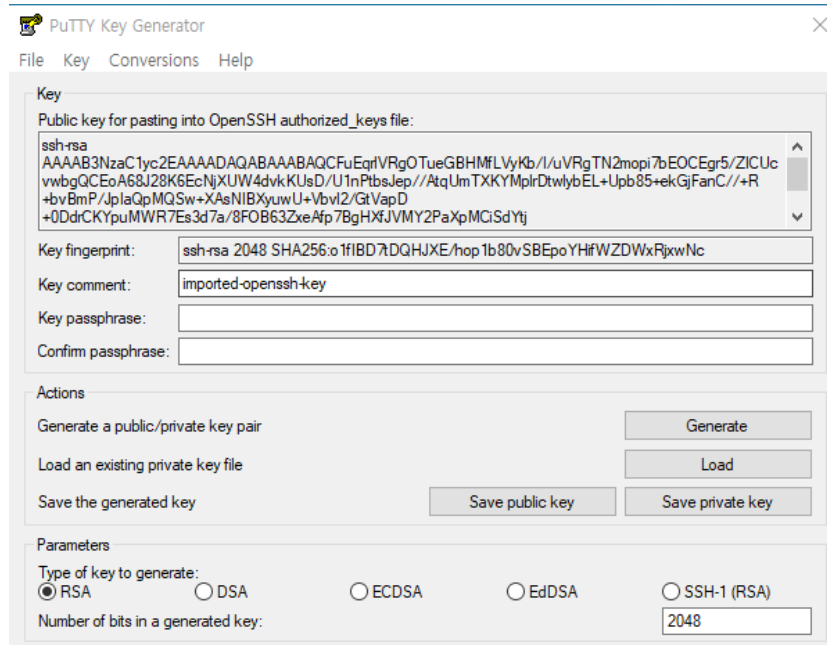


- 포트는 컴퓨터상에서 통신하기 위함, 다른 컴퓨터도 같은 포트 번호로 통신가능(동번호, 210동)
- IP는 컴퓨터간의 주소(호수번호, 1504호)
- **[AWS] EIP(탄력적 IP) 개념 & 사용 세팅 정리**
  - EIP(Elastic IP Address)는 인터넷을 통해 접속할 수 있는 고정적인 공인 IP 주소를 할당할 수 있고, 인스턴스에 연결할 수 있는 서비스다.
  - 모든 인스턴스 또는 네트워크 인터페이스에 탄력적 IP 주소를 연결 가능하다.
  - EIP를 할당받고 삭제하기 전까지 IP 주소를 쭉 유지할 수 있다.
  - ENI(Elastic Network Interface)에 퍼블릭 IP를 달게 되면 외부에서 접속
  - 하지만, 인스턴스의 **Public IP**는 고정된 IP 주소가 아니라 유동적인 IP 주소 이다.
  - AWS에서는 바로 한 번 할당 받으면 절대 바뀌지 않은 **Elastic IP**를 제공한다.

<sup>1</sup> <https://www.youtube.com/watch?v=pON9FLO49SY>

## 8) Putty

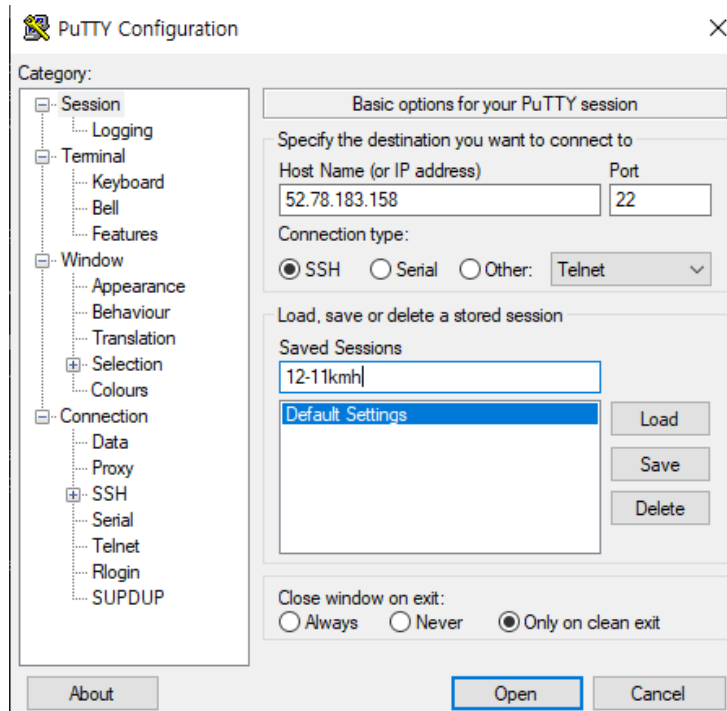
### 8-1) puttygen :: 키페어 생성



- generate : 윈도우상에서 키페어 생성, 생성시 전체 설정 동일화 필요
- (1) load : all.files 로 찾기 > 12-11kmh.pem > successfully
- (2) save private key(.ppk = 푸티) > 12-11kmh.pem 을 12-11kmh.ppk 로 새로생성 및 저장

### 8-2) putty

- ubuntu 는 복불 불가, 불편, 외부 변경 불허용이므로 putty를 사용함
- (1) PublicIPs: 52.78.183.158 을 Host Name 에 복불
  - 인스턴스 연결 > 연결 > 우분투(검은화면)의 pulicIPs 확인



- (2) appearance : 폰트 변경

- SSH > Auth
  - (3) Credential > Browser > 12-11kmh.ppk (2개다 설정)
  - (4) GSSAPI > Browser > 12-11kmh.ppk
    - 주의 : 바로 오픈 하지 말고, Session 으로 가서 저장 부터, 이후 (5) open
    - 에러 메시지 무시, 초기 한번만 나옴
- accept > ubuntu : 이제 두 화면은 동일

```

AWS
서비스 Q 검색 [일트+S]
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1012-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Mon Dec 11 05:45:21 UTC 2023

System load: 0.080078125      Processes: 97
Usage of /: 6.5% of 24.05GB   Users logged in: 1
Memory usage: 21%            IPv4 address for eth0: 172.31.8.123
Swap usage: 0%

0 updates can be applied immediately.
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Mon Dec 11 05:45:22 2023 from 13.209.1.59
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-8-123:~$
  
```

- 우분투 서버에서는 접속되는 것과 동일

## 9) mariadb 설치 및 설정 : ubuntu 화면상에서

### 9-1) 설치전 업데이트 및 업그레이드

- (1) `sudo apt-get update -y`

```
ubuntu@ip-172-31-8-123: ~  
kB]  
Get:33 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [11.4 kB]  
Get:34 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1179 kB]  
Get:35 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [191 kB]  
Get:36 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 c-n-f Metadata [520 B]  
Get:37 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [815 kB]  
Get:38 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [152 kB]  
Get:39 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [16.8 kB]  
Get:40 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [36.5 kB]  
Get:41 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [7060 B]  
Get:42 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [260 B]  
Fetched 28.7 MB in 5s (5536 kB/s)  
Reading package lists... Done  
ubuntu@ip-172-31-8-123:~$
```

- -y : all yes

- (2) `sudo apt-get upgrade > y`

### 9-2) mariaDB 설치

- (3) `sudo apt-get install mariadb-server > y`
  - EC2 에서의 설치된 상태를 의미
- (4) `cd /` : 최상위 루트 상태

```
ubuntu@ip-172-31-8-123:~$ cd /  
ubuntu@ip-172-31-8-123:/$
```

- (5) `sudo mysql_secure_installation > (enter) > y> 1234, 1234 > y > y > y > y`

### 9-3) mariaDB 접속

- (6) `sudo mariadb` : 접속됨
  - `mysql -u root -p > 1234` : 접속 불가



#### 9-4) 중간 확인

- date

```
ubuntu@ip-172-31-8-123:~$ date
Mon Dec 11 06:07:37 UTC 2023
ubuntu@ip-172-31-8-123:~$
```

- show databases;

```
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
+-----+
```

- select version();

```
MariaDB [(none)]> select version();
+-----+
| version() |
+-----+
| 10.6.12-MariaDB-0ubuntu0.22.04.1 |
+-----+
1 row in set (0.000 sec)
```

#### 9-5) 접속 환경 설정

- (7)(10) select host, user, plugin from mysql.user;

```
+-----+-----+-----+
| host      | user  | plugin |
+-----+-----+-----+
| localhost | root  | unix_socket |
+-----+-----+-----+
```

- unix\_socket : 으로 인해 mysql -u root -p 가 통용되지 않음
  - 외부 접속 불가를 원할 경우 사용해도 무방함
- (8) update mysql.user set plugin = 'mysql\_native\_password' where user='root';
- (9) flush privileges;
  - mysql 전체에 소켓-> mysql\_native\_password 강제로 입력
- select host, user, plugin from mysql.user; plugin -> mysql\_native\_password 변경 확인됨

#### 9-6) 시간 설정

- 확인 : `select @@global.time_zone, @@session.time_zone;`

```
MariaDB [(none)]> select @@global.time_zone, @@session.time_zone;
+-----+-----+
| @@global.time_zone | @@session.time_zone |
+-----+-----+
| SYSTEM            | SYSTEM              |
+-----+-----+
1 row in set (0.000 sec)
```

- 우분투의 시간 설정과 동일
- `set global time_zone='Asia/Seoul';`  
`set time_zone='Asia/Seoul'`
  - 실패, mariadb 자체에서 시각을 사용하지 않으므로
- (11) mariadb에서 exit 한후 실행
  - (12) `sudo timedatectl set-timezone Asia/Seoul && sudo timedatectl set-ntp true;`
  - (13) `date` 로 변경된 것 확인

```
ubuntu@ip-172-31-8-123:~$ sudo timedatectl set-timezone Asia/Seoul && sudo timedatectl set-ntp true;
ubuntu@ip-172-31-8-123:~$ date
Mon Dec 11 16:09:05 KST 2023
```

#### 9-7) 설치된 db 확인

- `sudo dpkg -l | grep mariadb`
  - dpkg : 데비안 패키지

#### 9-8) db 삭제

- `sudo apt-get purge mariadb-server`
  - db 삭제 중 server에 해당하는 파일만 지움
  - 리눅스는 하나씩
  - -get : 메모리 상주나 해당하는 내용 전체 로드

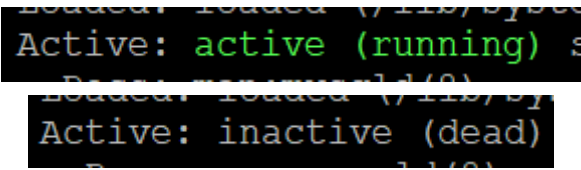
```
ubuntu@ip-172-31-8-123:~$ sudo dpkg -l | grep mariadb
ii  libmariadb3:amd64          1:10.6.12-0ubuntu0.22.04.1
64      MariaDB database client library
ii  mariadb-client-10.6        1:10.6.12-0ubuntu0.22.04.1
64      MariaDB database client binaries
ii  mariadb-client-core-10.6   1:10.6.12-0ubuntu0.22.04.1
64      MariaDB database core client binaries
ii  mariadb-common             1:10.6.12-0ubuntu0.22.04.1
64      MariaDB common configuration files
ii  mariadb-server-10.6        1:10.6.12-0ubuntu0.22.04.1
64      MariaDB database server binaries
ii  mariadb-server-core-10.6   1:10.6.12-0ubuntu0.22.04.1
64      MariaDB database core server files
ubuntu@ip-172-31-8-123:~$ sudo apt-get purge mariadb-server^C
ubuntu@ip-172-31-8-123:~$ sudo apt-get purge
```

`sudo apt-get purge` 상태에서 지우고자 하는 디렉토리를 마우스 우클릭하면 복붙

- 완전 삭제시 `sudo dpkg -l | grep mariadb` 하면 아무 리스트가 안뜸

#### 10) mariadb status

- sudo systemctl start mariadb
- sudo systemctl stop mariadb
- sudo systemctl status mariadb
  - 확인시



```
Loaded: loaded (/lib/systemd/systemd)
Active: active (running) since 2023-11-10 10:10:10; 1min 10s ago
Main PID: 1234
Status: "MariaDB server"
CGroup: /systemd/system/mariadb.service
└─ 1234
```

```
Loaded: loaded (/lib/systemd/systemd)
Active: inactive (dead) since 2023-11-10 10:10:10; 1min 10s ago
Main PID: 1234
Status: "MariaDB server"
CGroup: /systemd/system/mariadb.service
└─ 1234
```

- ctrl+c + (enter) 로 exit

#### 11) AWS 로 접속시

- root 는 ec2-user
- Linux 명령어체계가 다르므로 검색 필요

## [12/12] EC2 우분투(Ubuntu) MariaDB 외부접속 허용

### 1) 프로그램 설치

#### 1-2) 설치

- MySQL Workbench만 설치
  - Workbench download 검색 > [Windows \(x86, 64-bit\), MSI Installer](#)
- HeidiSQL 설치
  - HeidiSQL 검색 > 다운로드 > [Installer, 32/64 bit combined](#)

#### 1-3) 주의사항

- Workbench 와 HeidiSQL 는 둘중 하나만 켜서 사용할 것
- Workbench + putty or HeidiSQL + putty 는 가능
- 인스턴스 에러 발생, 충돌 우려
- CLI 수정되나 어려움이 큰 편

### 2) 진행 상황 및 보충 설명

- 12월 11일 : AWS-EC2에서 mariadb가 설치된 상황
- 내PC에서 AWS 접속시 보안규칙을 통과해야 함
- 리눅스에는 방화벽 설정이 존재하며, db or webserver가 외부 접속시 방화벽이 자동 활성화됨
  - 해당 설정을 배제해줘야 방화벽을 뚫고 진입 가능
  - putty 에서는 키패어를 통해 방화벽을 (암호화 기법으로) 우회/무효화 한 것

### 3) mariadb 의 외부 접속을 허용하는 방법

- 4) EC2 보안그룹 규칙 수정
- 5) MariaDB 설정 변경
- 6) User & Host 권한 변경
  - 이후 방화벽 설정 변경을 통해 접속 허용

### 4) EC2 보안그룹 규칙 수정

- AWS 에서 mariadb의 포트 3306설정
- 타 포트 사용해도 되나 mariadb가 안정적으로 구동되기 위한 기본 포트

인바운드 규칙 (5)						
<input type="text" value="Search"/>						
<input type="checkbox"/>	Name	보안 그룹 규칙 ID	IP 버전	유형	프로토콜	포트 범위
<input type="checkbox"/>	-	sgr-03da979b7dc1185...	IPv4	MYSQL/Aurora	TCP	3306

- Heidi의 경우도 3306포트를 통해 진입

## 5) MariaDB 설정 변경

- VI(vim) 편집기를 통해 127.0.0.1 을 0.0.0.0 으로 변경한 바 있음 (GITAS(1) 27p)
  - `sudo vim /etc/mysql/mariadb.conf.d/50-server.cnf` (root 단계에서 실행)
  - **bind-address = 0.0.0.0** : 외부(모든 ip) 접속 허용

```
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 127.0.0.1
```

3.35.213.179 , 127.0.0.1 두개 이상도 지정 가능

- vi 설명
  - 커서 없음, 키보드 화살표로 이동
  - : 명령어 모드
  - **ctrl+c** : insert 해제(붉은색 경고창시)
  - 뭔가 잘못 누르면 **ESC** , **:q!** 로 탈출
  - **expire\_log\_days** : 로그 기록 최대 기한 10일
- 순서 : 0에서 i 키 누름 > 0.0.0.0 작성, delete 키로 지움 > ESC > :wq
  - **i** : insert 모드 시작
  - **ESC** : insert 모드 해제
  - **:wq** : 기록 저장 종료
  - **:wq!** : 강제 저장 종료
- 이후 리부트할 것 : `sudo systemctl restart mysql`

## 6) User & Host 권한 변경

- mariadb 접속 : putty 실행 > ubuntu > mysql -u root -p > 1234

### 6-1) 사용자 생성 : (1) create user 'student'@'%' identified by 1234;

- user 는 테이블, mysql에 user라는 테이블이 존재하며 모든 아이디의 정보가 포함됨
- identified by : 비밀번호를 부여, identified by 를 붙이지 않아도 생성은 됨
- db 의 인가는 id,pw,ip 세개가 필요하므로 사용자 권한 설정에서 ip 를 부여해 외부 접속을 허용
- % : 모든 ip, all를 의미
  - 특정 ip 를 작성시 해당 아이디는 해당 아이피만 접속 가능
- 만약 IP 부분을 localhost 로 작성시 create user 'student'@'localhost' identified by 1234;
  - localhost : IP Address, localhost인 경우 외부 접속 불가

### 6-2) 사용자 삭제 : drop user 'student'@'localhost';

- 삭제는 ID 와 IP 만 지정해도 가능
- Password 가 있더라도 ID 와 IP 만 작성시 삭제됨
- 단, 삭제된 회원 정보는 별도의 db에 보관하는 것이 권장

### 6-3) user 관련 테이블 확인 : (2) select user, host, password from mysql.user;

```
MariaDB [(none)]> select user,host from mysql.user;
+-----+-----+
| user | host      |
+-----+-----+
| root | localhost |
+-----+-----+
```

### 6-4) 비밀번호 변경 : update user set password=password('1234') where user='root'

- 해당 구문을 사용하는 것을 권장, 모든 비밀번호 변경 가능
  - mysqladmin -u root password '1234'
  - set password for root=password('1234') 도 가능하나 첫 구문이 유효한편
- flush privileges;
  - DB 전체에 전파

### 6-5) 권한 설정 : (3) grant select on \*.\* to 'student00'@'%' identified by '1234' with grant option;

- CRUD 모든 권한 부여 : grant all privilege on ~ 이지만 사용할 일 없음
- 일반적으로 읽기 권한인 select 와 삽입 권한인 insert 정도만 주는 편
- 권한 회수의 경우 revoke

## 7) 방화벽 설정

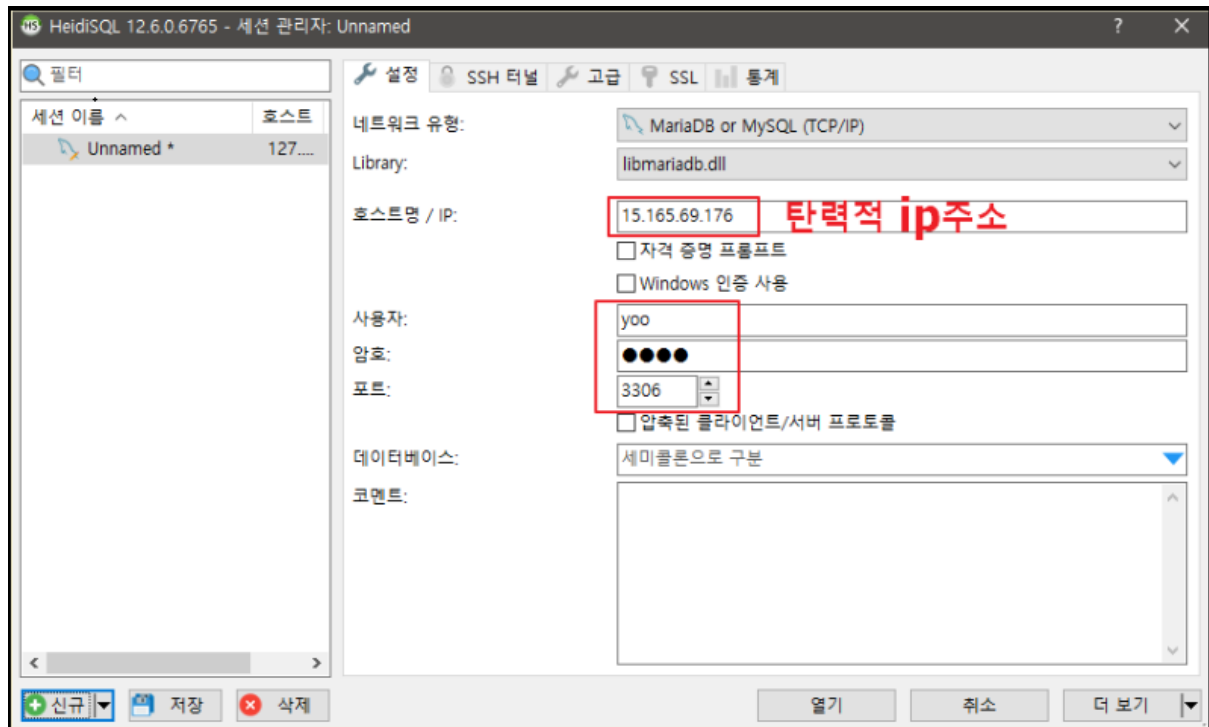
- AWS에서는 방화벽보다 인바운드 규칙이 우선됨

### 7-1) 권장 방화벽 설정

- sudo ufw enable : 상시 오픈
  - ufw : ubuntu firewall
- sudo ufw allow from 192.168.0.3 to any port 22 proto tcp
  - SSH로 접속할 수 있는 특정 ip를 지정하는 결과 도출
  - proto : 프로토콜 종류
- sudo ufw allow 123/udp : NTP(network time protocol) 서버 시간 동기화
- sudo ufw allow 80/tcp : HTTP용
- (1) sudo ufw allow 3306/tcp : DB용
- (2) sudo ufw status : 방화벽 상태 확인

## 8) HeidiSQL 를 통한 접속 및 확인

- 신규 > 호스트명 > 사용자 > 암호 > 확인 > DB 확인됨



## 9) Workbench를 통한 접속 및 확인