

스프링부트로 RestFulAPI 구현하기

# 13장 AWS EC2 배포

- AWS EC2

박명회

## 13장 AWS란

**AWS**는 "**Amazon Web Services**"의 약자로, 아마존(**Amazon**)이 제공하는 클라우드 컴퓨팅 플랫폼입니다. 이 서비스는 개발자와 기업에게 컴퓨팅 파워, 데이터 저장, 컴퓨팅 리소스, 데이터베이스, 네트워킹 등의 다양한 **IT** 서비스를 제공합니다.

**AWS**는 인프라스트럭처를 관리하고 유연한 확장성을 제공하여 기업이 비용을 절감하고 애플리케이션을 더욱 효율적으로 운영할 수 있도록 돕는 클라우드 서비스 플랫폼입니다.

이 서비스는 기업이 인터넷을 통해 컴퓨팅 파워 및 데이터 스토리지를 확보할 수 있도록 합니다. **AWS**는 서버 호스팅, 데이터베이스 저장소, 컴퓨팅 파워, 콘텐츠 전송 등의 다양한 서비스를 제공하여 기업의 **IT** 인프라를 클라우드로 이전하고 비용을 절감하고자 하는 기업에게 인기가 있습니다.

## 13장 로그인


- 루트 사용자는 **AWS** 계정의 최상위 권한을 가지고 있으며 보안상의 이유로 최소한으로만 사용해야 합니다.
- **IAM** 사용자는 개별적으로 관리되는 사용자로 필요에 따라 권한을 할당하여 **AWS** 리소스를 안전하게 관리합니다.

❗

새 로그인 UI 사용해보기

공식 출시 전에 새롭게 개선된 Amazon Web Services 로그인 환경을 확인해 보세요.

새 로그인 활성화



로그인

☒ 루트 사용자  
무제한 액세스 권한이 필요한 작업을 수행하는 계  
정 소유자입니다. 자세히 알아보기

☐ IAM 사용자  
일일 작업을 수행하는 계정 내 사용자입니다. 자세  
히 알아보기

루트 사용자 이메일 주소

다음

계속 진행하는 경우 AWS 고객 계약 또는 AWS 서비스에  
대한 기타 계약 및 개인 정보 보호 정책에 동의하게 됩니다.  
이 사이트는 필수 쿠키를 사용합니다. 자세한 내용은 쿠키  
고지를 참조하세요.

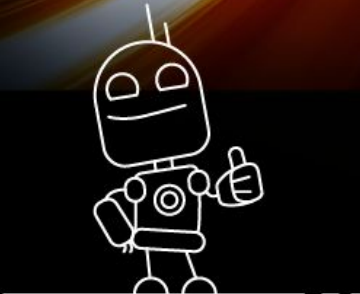
AWS를 처음 사용하십니까?

AWS 계정 새로 만들기

Amazon Lightsail

Lightsail은 AWS를 가장  
쉽게 시작할 수 있는 방법

자세히 알아보기 »



## 13장 EC2생성 및 접속

### - EC2 생성합니다.

EC2 > 인스턴스 > Launch an instance

Launch an instance 정보

Amazon EC2를 사용하면 AWS 클라우드에서 실행되는 가상 머신 또는 인스턴스를 생성할 수 있습니다. 아래의 간단한 단계에 따라 빠르게 시작할 수 있습니다.

이름 및 태그 정보

이름

예: 내 웹 서버

추가 태그 추가

▼ 애플리케이션 및 OS 이미지(Amazon Machine Image) 정보

AMI는 인스턴스를 시작하는 데 필요한 소프트웨어 구성(운영 체제, 애플리케이션 서버 및 애플리케이션)이 포함된 템플릿입니다. 아래에서 찾고 있는 항목이 보이지 않으면 AMI를 검색하거나 찾아보세요.

Q 수천 개의 애플리케이션 및 OS 이미지를 포함하는 전체 카탈로그 검색

Quick Start

Amazon Linux

aws

macOS

Mac

Ubuntu

ubuntu®

Windows

Microsoft

Red Hat

Red Hat

SUSE Linux

SUS

더 많은 AMI 찾아보기

AWS, Marketplace 및 커뮤니티의 AMI 포함

Amazon Machine Image(AMI)

▼ 요약

인스턴스 개수 정보

1

소프트웨어 이미지(AMI)

Amazon Linux 2023 AMI 2023.5.2...더 보기

ami-07f15eb4844514508

가상 서버 유형(인스턴스 유형)

t2.micro

방화벽(보안 그룹)

새 보안 그룹

스토리지(블록)

1개의 블록 – 8GiB

❗ 프리 티어: 첫 해에는 월별 프리 티어

AMI에 대한 t2.micro(또는 t2.micro를 사용할 수 없는 리전의 t3.micro) 인스턴스 사용량 750시간, 월별 퍼블릭 IPv4 주소 사용량 750시간, EBS 스토리지 30GiB, IO 2백만 개, 스냅샷 1GB, 인터넷 대역폭 100GB가 포함됩니다.

취소

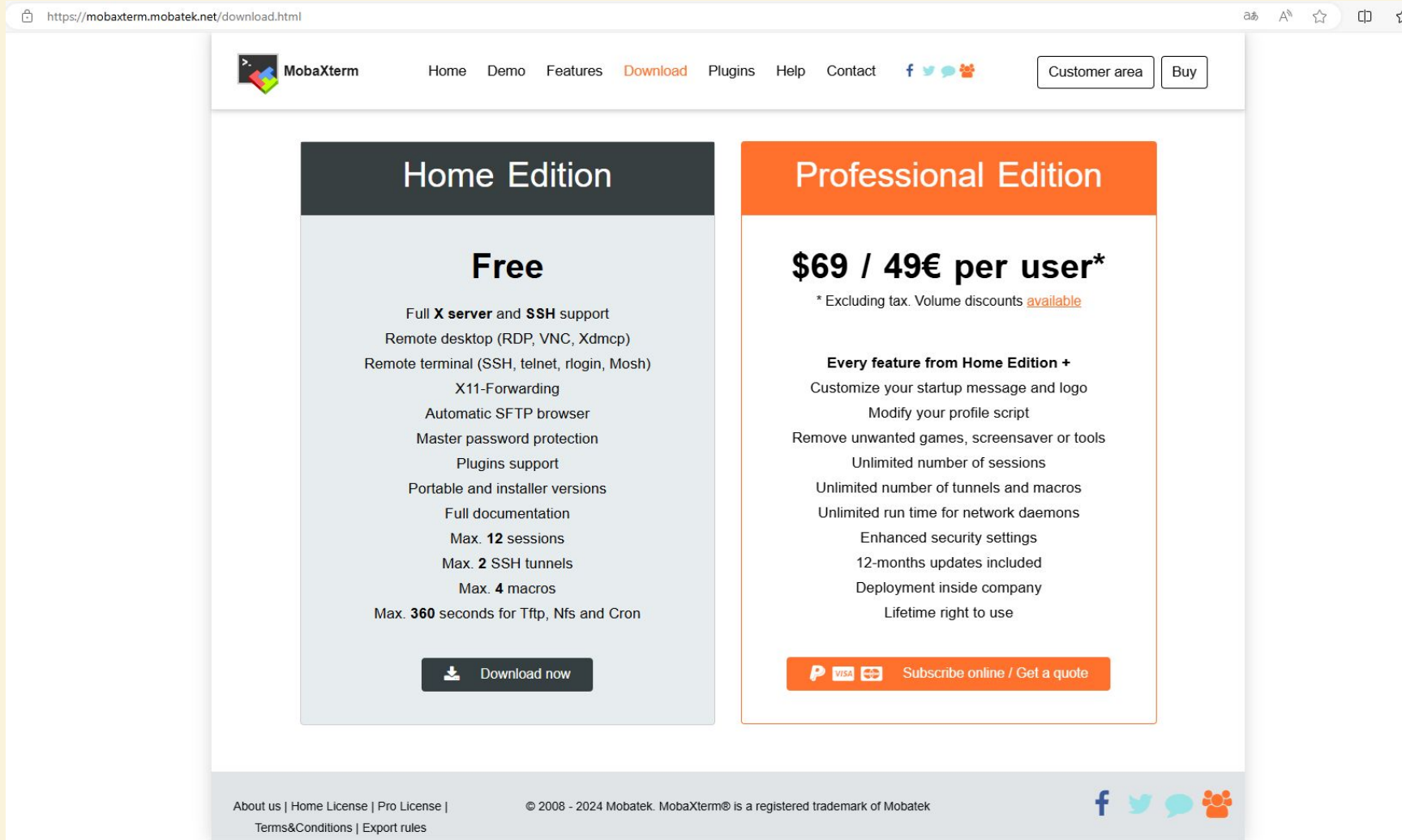
인스턴스 시작

명령 검토

## 13장 EC2생성 및 접속

[MobaXterm Xserver with SSH, telnet, RDP, VNC and X11 - Home Edition \(mobatek.net\)](https://mobaxterm.mobatek.net/download.html)

### - 접속 프로그램 mobaXterm 다운로드 및 설치



The screenshot shows the MobaXterm website's download page. The page is divided into two main sections: Home Edition and Professional Edition. The Home Edition is free and includes features like X server and SSH support, remote desktop, and terminal. The Professional Edition costs \$69 / 49€ per user\* and includes all Home Edition features plus additional capabilities like custom startup messages, unlimited sessions, and enhanced security settings. Both editions have a 'Download now' button. The Professional Edition also has a 'Subscribe online / Get a quote' button. The footer contains links for 'About us', 'Home License', 'Pro License', 'Terms&Conditions', and 'Export rules', along with copyright information for 2008-2024 Mobatek.

https://mobaxterm.mobatek.net/download.html

MobaXterm

Home Demo Features **Download** Plugins Help Contact

Customer area Buy

### Home Edition

#### Free

- Full **X server** and **SSH** support
- Remote desktop (RDP, VNC, Xdmcp)
- Remote terminal (SSH, telnet, rlogin, Mosh)
- X11-Forwarding
- Automatic SFTP browser
- Master password protection
- Plugins support
- Portable and installer versions
- Full documentation
- Max. **12** sessions
- Max. **2** SSH tunnels
- Max. **4** macros
- Max. **360** seconds for Tftp, Nfs and Cron

Download now

### Professional Edition

#### \$69 / 49€ per user\*

\* Excluding tax. Volume discounts [available](#)

**Every feature from Home Edition +**

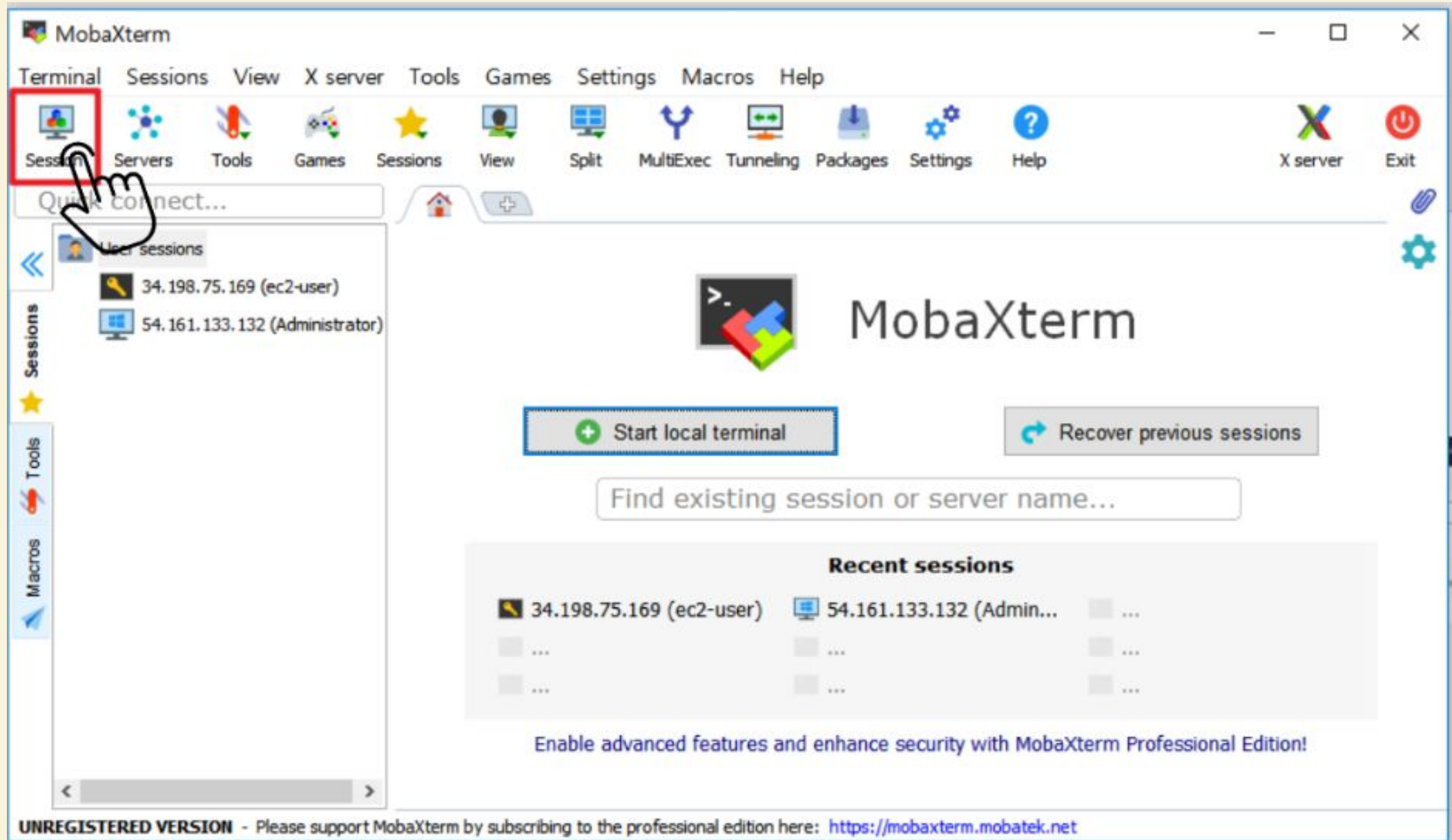
- Customize your startup message and logo
- Modify your profile script
- Remove unwanted games, screensaver or tools
- Unlimited number of sessions
- Unlimited number of tunnels and macros
- Unlimited run time for network daemons
- Enhanced security settings
- 12-months updates included
- Deployment inside company
- Lifetime right to use

Subscribe online / Get a quote

About us | Home License | Pro License | Terms&Conditions | Export rules

© 2008 - 2024 Mobatek. MobaXterm® is a registered trademark of Mobatek

## 13장 EC2생성 및 접속





## 13장 EC2생성 및 접속

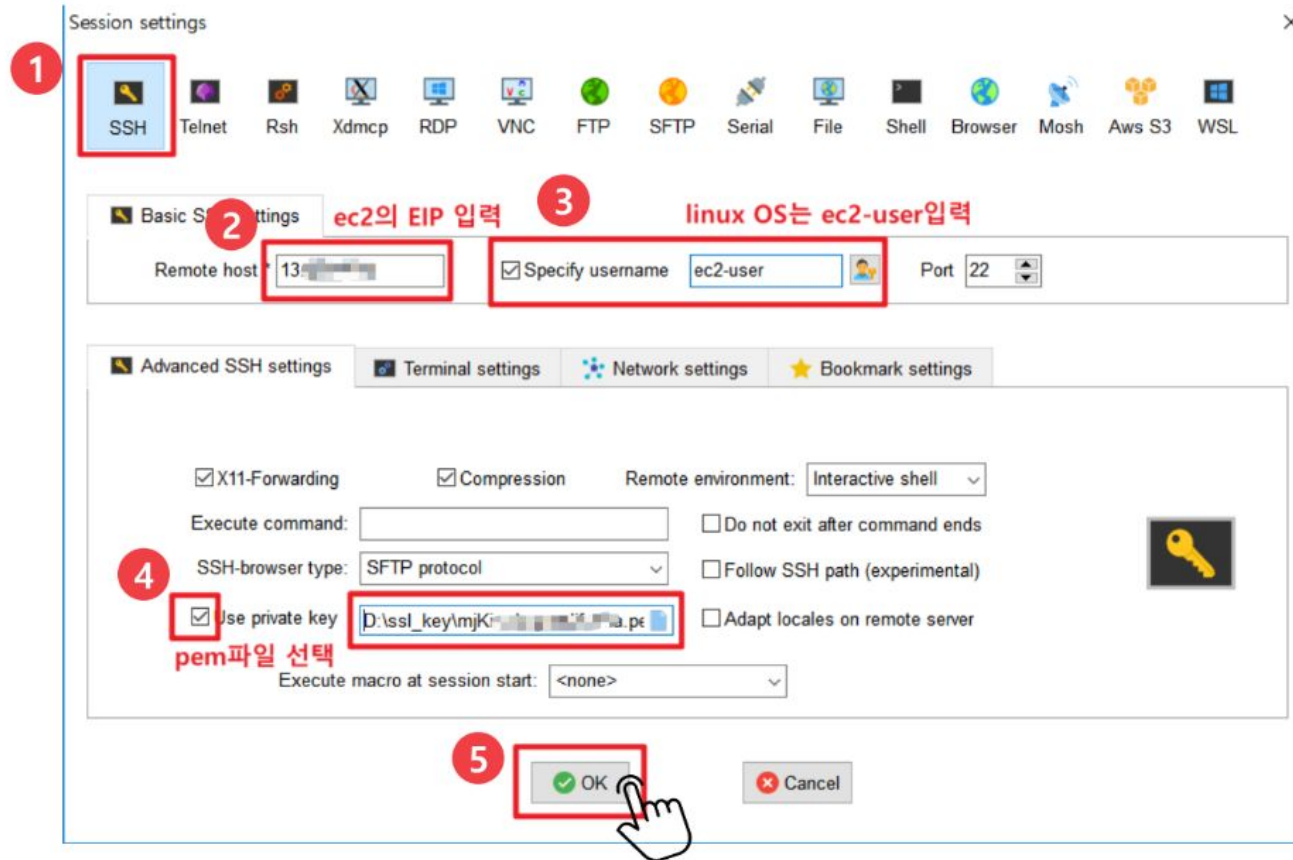
1번 - SSH 클릭해주세요.

2번 - 여러분이 접속하고자 하는 인스턴스의 EIP를 입력해주세요.

3번 - Linux OS의 username은 'ec2-user'입니다. 저와 똑같이 입력해주셔야 합니다.

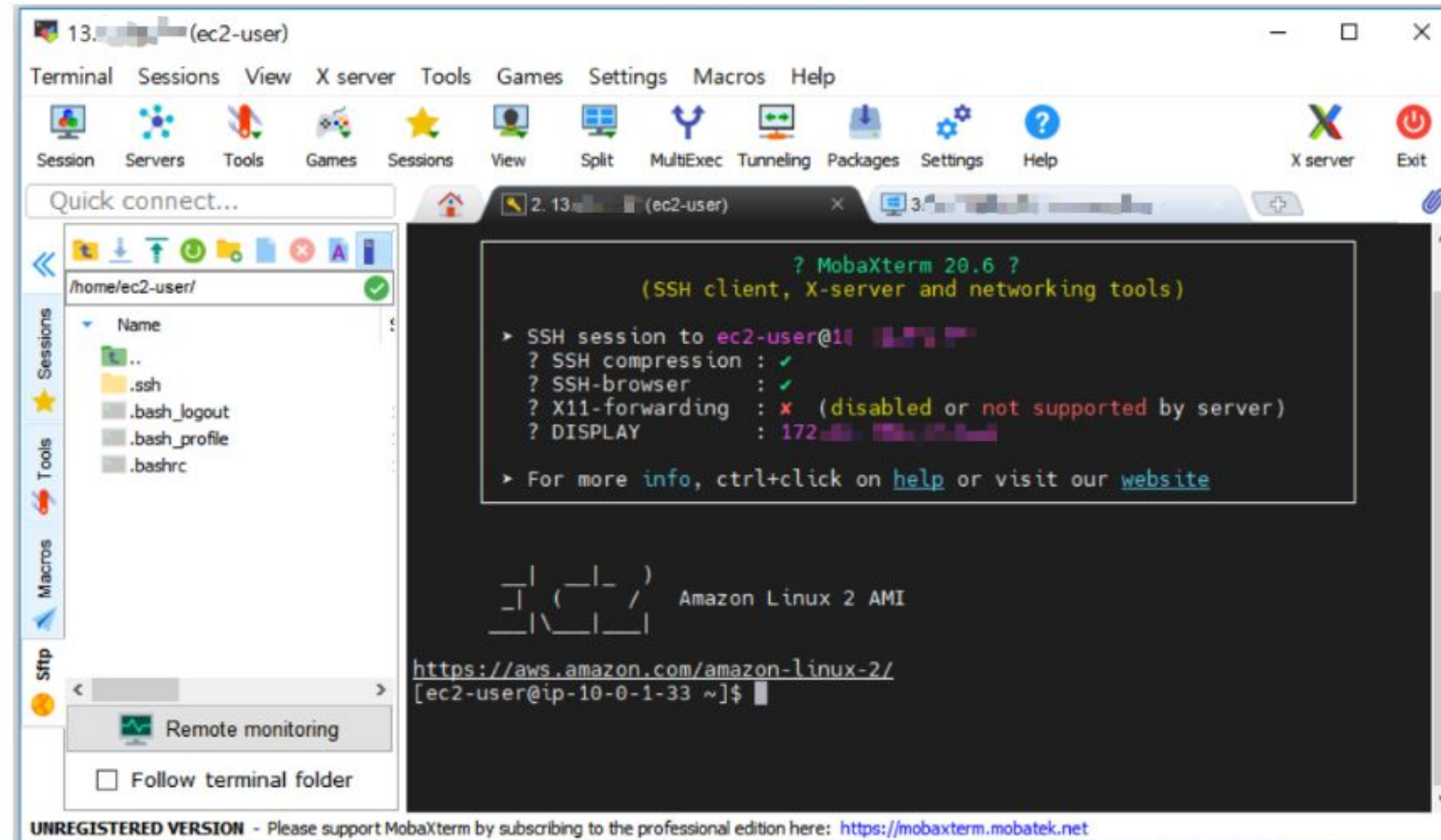
4번 - 인스턴스에 접속하기 위해서는 PEM파일이 필요하다고 했습니다. 우선 왼쪽에 체크를 해주시고, 인스턴스를 만들 때 다운로드 받았던 PEM파일 위치를 선택해주세요!

5번 - 1-4까지 완료 되었다면, OK버튼을 클릭해주세요.



## 13장 EC2생성 및 접속

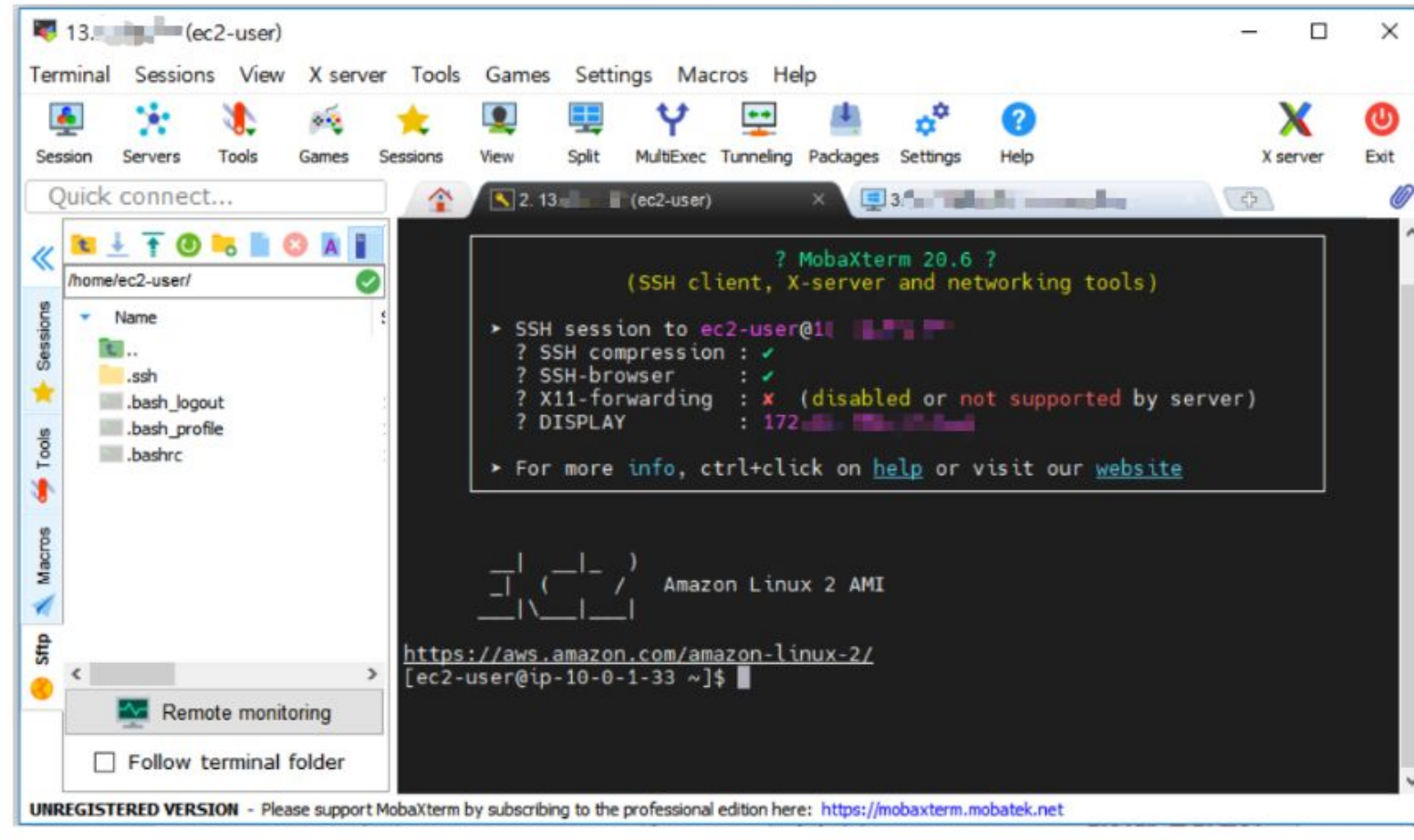
### 접속 완료 화면





## 13장 EC2생성 및 접속

### 접속 완료 화면



# 13장 EC2서버 용량 늘리기

<https://forested-buffet-8b9.notion.site/aws-bf46262d0e19429a83740254edbc0841>

## aws 메모리부족

소유자	인증	태그
비어 있음	비어 있음	비어 있음

AWS EC2 인스턴스 프리티어 t2.micro 를 사용한다면, 부족한 메모리용량 때문에 서비스가 중단되는 경우가 있다.  
물론 서버에 올려둔 프로젝트의 무게 자체를 줄이는 방법으로 해결할 수도 있지만, EC2에서 가용할 수 있는 메모리 용량을 증가시켜서 해결할 수도 있다.  
이번 포스팅에서는 AWS EC2가 메모리용량 부족으로 다운될 때, 메모리 용량을 증설 시킬 방법에 대해서 포스팅해본다.

### 1. 인스턴스 유형 변화

간단한 방법으로 인스턴스의 유형 자체를 변경하는 것이다.  
t2.micro가 메모리 용량이 모자라다면, 다른 인스턴스를 가져다쓰면 되는것이다.  
현재 사용하는 인스턴스에서 메모리만 바꿀수 없다. 인스턴스는 이미 구조가 완성된 고정된 서버컴퓨터이다.  
따라서 우리는 대여하는 인스턴스를 하나의 PC방 컴퓨터라고 생각하면 쉽다.  
PC방 컴퓨터 또한 자리마다 스펙이 다른 이미 완성된 컴퓨터가 있고, 우리는 그 컴퓨터의 스펙을 보고 자리를 대여한다. PC방 컴퓨터도 스펙이 낮은 자리에 앉아서 부품을 바꿔달라고 할 수 없는가?

## 13장 docker 엔진 설치

<https://docs.docker.com/engine/install/ubuntu/>

The screenshot shows the Docker documentation website. The browser address bar displays `docs.docker.com/engine/install/ubuntu/`. The page has a blue header with the Docker logo and navigation links: `Guides`, `Manuals` (active), and `Reference`. A search icon and a sun icon are on the right. The left sidebar contains a table of contents with links to `Overview`, `Get Docker`, `Docker Desktop`, `Docker Extensions`, `Docker Scout`, `Docker Engine` (expanded), `Overview`, `Install` (expanded), `Overview`, `CentOS`, `Debian`, `Fedora`, `RHEL`, `SLES`, `Ubuntu` (highlighted), `Raspberry Pi OS (32-bit)`, `Binaries`, and `Post installation steps`. The main content area has a breadcrumb trail: `Manuals / Docker Engine / Install / Ubuntu`. The title is `Install Docker Engine on Ubuntu`. Below the title, it says: "To get started with Docker Engine on Ubuntu, make sure you [meet the prerequisites](#), and then follow the [installation steps](#)." The section `Prerequisites` is expanded, showing `Firewall limitations`. A warning box states: "Warning: Before you install Docker, make sure you consider the following security implications and firewall incompatibilities." Below the warning, there are two bullet points: "If you use `ufw` or `firewalld` to manage firewall settings, be aware that when you expose container ports using Docker, these ports bypass your firewall rules. For more information, refer to [Docker and ufw](#)." and "Docker is only compatible with `iptables-nft` and `iptables-legacy`. Firewall rules created with `nft` are not supported on a system with Docker installed. Make sure that any firewall rulesets you use are created with `iptables` or `ip6tables`, and that you add them to the `DOCKER-USER` chain, see [Packet filtering and firewalls](#)." On the right, there are links to `Edit this page` and `Request changes`, followed by a `Table of contents` section with links to `Prerequisites`, `Firewall limitations`, `OS requirements`, `Uninstall old versions`, `Installation methods`, `Install using the apt repository`, `Install from a package`, `Uninstall Docker Engine`, and `Next steps`. An `Ask AI` button is in the bottom right corner.

docs.docker.com/engine/install/ubuntu/

docker.docs Guides Manuals Reference

Overview

Get Docker

Docker Desktop

Docker Extensions

Docker Scout

Docker Engine

Overview

Install

Overview

CentOS

Debian

Fedora

RHEL

SLES

Ubuntu

Raspberry Pi OS (32-bit)

Binaries

Post installation steps

Manuals / Docker Engine / Install / Ubuntu

## Install Docker Engine on Ubuntu

To get started with Docker Engine on Ubuntu, make sure you [meet the prerequisites](#), and then follow the [installation steps](#).

### Prerequisites

#### Firewall limitations

**Warning**

Before you install Docker, make sure you consider the following security implications and firewall incompatibilities.

- If you use `ufw` or `firewalld` to manage firewall settings, be aware that when you expose container ports using Docker, these ports bypass your firewall rules. For more information, refer to [Docker and ufw](#).
- Docker is only compatible with `iptables-nft` and `iptables-legacy`. Firewall rules created with `nft` are not supported on a system with Docker installed. Make sure that any firewall rulesets you use are created with `iptables` or `ip6tables`, and that you add them to the `DOCKER-USER` chain, see [Packet filtering and firewalls](#).

[Edit this page](#)

[Request changes](#)

#### Table of contents

- Prerequisites
  - Firewall limitations
  - OS requirements
  - Uninstall old versions
- Installation methods
  - Install using the apt repository
  - Install from a package
- Uninstall Docker Engine
- Next steps

Ask AI

## 13장 도커 명령어

리눅스에서는 docker 명령어 앞에 항상 sudo를 붙여야 하는 불편함이 있었는데, 현재 계정을 docker 그룹에 추가하면 권한 문제가 해결된다. 방법은 아래와 같다.

# 도커 명령어 그룹추가

```
sudo usermod -aG docker $USER
```

# 도커 네트워크 생성

```
docker network create javamysql
```

# 도커 네트워크 확인

```
docker network ls
```

## 13장 도커 MYSQL 컨테이너

# 도커 MYSQL 컨테이너 생성

```
docker run -d --name mysql-container --network javamysql -e MYSQL_ROOT_PASSWORD=1234 -p 3306:3306 mysql
```

# 도커 MYSQL 컨테이너 확인

```
docker ps
```

# 도커 MYSQL 컨테이너 삭제

```
docker rmi mysql-container
```

## 13장 도커 JAVA 컨테이너

### dockerfile maven

```
FROM openjdk:17-jdk-slim AS build
```

```
COPY pom.xml mvnw ./
```

```
COPY .mvn .mvn
```

```
RUN ./mvnw dependency:resolve
```

```
COPY src src
```

```
RUN ./mvnw package
```

```
FROM openjdk:17-jdk-slim
```

```
WORKDIR demo
```

```
COPY --from=build target/*.jar demo.jar
```

```
ENTRYPOINT ["java", "-jar", "demo.jar"]
```



## 13장 도커 JAVA 컨테이너

dockerfile

<https://github.com/keeganwitt/docker-gradle/blob/502641f3165bdd0e1381ffb6e77f567d6f6d8893/jdk17/Dockerfile>

```
FROM gradle:jdk17-alpine as builder
WORKDIR /build
```

```
# 그래들 파일이 변경되었을 때만 새롭게 의존패키지 다운로드 받게함.
COPY build.gradle settings.gradle /build/
RUN gradle build -x test --parallel --continue > /dev/null 2>&1 || true
```

```
# 빌더 이미지에서 애플리케이션 빌드
COPY . /build
RUN gradle build -x test --parallel
```

```
# APP
FROM openjdk:17.0-slim
WORKDIR /app
```

```
# 빌더 이미지에서 jar 파일만 복사
COPY --from=builder /build/build/libs/*-SNAPSHOT.jar ./app.jar
```

```
EXPOSE 8080
```

```
# root 대신 nobody 권한으로 실행
USER nobody
ENTRYPOINT ["java", "-jar", "-Djava.security.egd=file:/dev/./urandom", "-Dsun.net.inetaddr.ttl=0", "app.jar"]
```

## 13장 도커 JAVA 컨테이너

# 이미지 restful 생성

```
docker build -t restful .
```

# 이미지 restful 생성

```
docker tag restful parkmyoung Choi/restful
```

# 컨테이너 java-container 생성

```
docker run -p 8080:8080 -e SPRING_DATASOURCE_URL=jdbc:mysql://mysql-container:3306/pmh -e  
SPRING_DATASOURCE_PASSWORD=1234 -d --network javamysql --name java-container  
parkmyoung Choi/restful
```

감사합니다