

김현수_AWS 클라우드 PuTTY 접속 방법 & 사용 설명 문서화

<사용프로그램>

- Putty

AWS가 뭘까? 왜쓰는 걸까?

우리가 실질적으로 만든 프로젝트는 톰캣(서버), 오라클(DB)로 스프링으로 프로젝트를 만든다해도

나만 로컬로 접속 가능하며 모든 사람들이 볼수 없다. 하지만 AWS 를 사용하여 인터넷 컴퓨터를 빌려쓰면

전세계 누구나 접속 가능하고 24시간 서버를 켜놔야하는 단점들도 AWS에서 커버가 가능하다.

트래픽이 많으면 느려지며, 고장나면 서비스 중단되는 부분들에 대한 개선점이라고 보면된다.

보안 또한 알아서 관리해주지만 비용적인 부분이 발생하는 부분이 있다.

localhost:8080 (나만 볼수있음.)

myproject.com (전세계가 볼수 있다.)

1. 먼저 AWS 사이트에 들어가서 회원가입 후 > EC2 검색 > 접속 > AWS EC2 인스턴스 생성 하자

The screenshot shows the AWS EC2 Instances page with the 'Launch Instance' wizard. The steps are as follows:

- Step 1: Set Instance Details**: Shows the instance name 'acehs91' and a note about choosing an AMI.
- Step 2: Choose AMI**: Shows the 'Amazon Machine Image(AMI)' section with various options like Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and Debian.
- Step 3: Configure Instance Details**: Shows the 'Quick Start' section with various operating system icons.
- Step 4: Add Storage**: Shows the storage configuration with 'Amazon EBS Volume' selected.
- Step 5: Set Security Group**: Shows the security group selection step.
- Step 6: Review and Launch**: Shows the review step with the instance type 't3.micro' and storage '1개의 볼륨 - 8GB'.
- Step 7: Launch Instance**: Shows the final launch button.

키 페어 생성

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

acehs91

이름에는 최대 255개의 ASCII 문자가 포함됩니다. 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

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

취소 **키 페어 생성**

2. .PPK 파일 다운로드 및 로컬 안전한 곳에 보관 후 - 인증서 .ppk 저장 → C:\aws

오른쪽 하단 > 인스턴스 시작으로 내가 아마존서버의 내 컴퓨터 하나 산것을 볼수 있다.

3. 보안 그룹에서 필수 포트(22, 80, 8080) 개방

한글로 설명쓰면 인바운드 만들지못함 - ERROR

각 포트 의미

포트	용도	설명
22	SSH	PuTTY로 서버 접속
80	HTTP	웹사이트 접속 (http://)
8080	Tomcat	스프링 프로젝트 접속

포트	용도	설명
1521	Oracle DB	오라클 DB 접속 (나중에 필요)
3305	MySQL?	혹시 MySQL 쓸 경우

The screenshot shows the AWS Management Console with the EC2 service selected. A specific security group, 'sg-03aad27d90c119311 - launch-wizard-1', is being configured. The 'Inbound Rules' section lists the following rules:

Name	보안 그룹 규칙 ID	IP 범위	유형	프로토콜	포트 범위	소스	설명
-	sgr-00fa0bf05000b2921	IPv4	HTTP	TCP	80	0.0.0.0/0	Web Access (http://)
-	sgr-03f4699d9keab23318	IPv4	Oracle-RDS	TCP	1521	0.0.0.0/0	Oracle DB
-	sgr-0ebdc355068297da8	IPv4	사용자 지정 TCP	TCP	3305	0.0.0.0/0	MySQL
-	sgr-031ff5f3ad36205d87	IPv4	사용자 지정 TCP	TCP	8080	0.0.0.0/0	Tomcat
-	sgr-08aa92b4cb382b6be	IPv4	SSH	TCP	22	0.0.0.0/0	SSH - PuTTY

⬇️ PuTTY 다운로드 & 설치

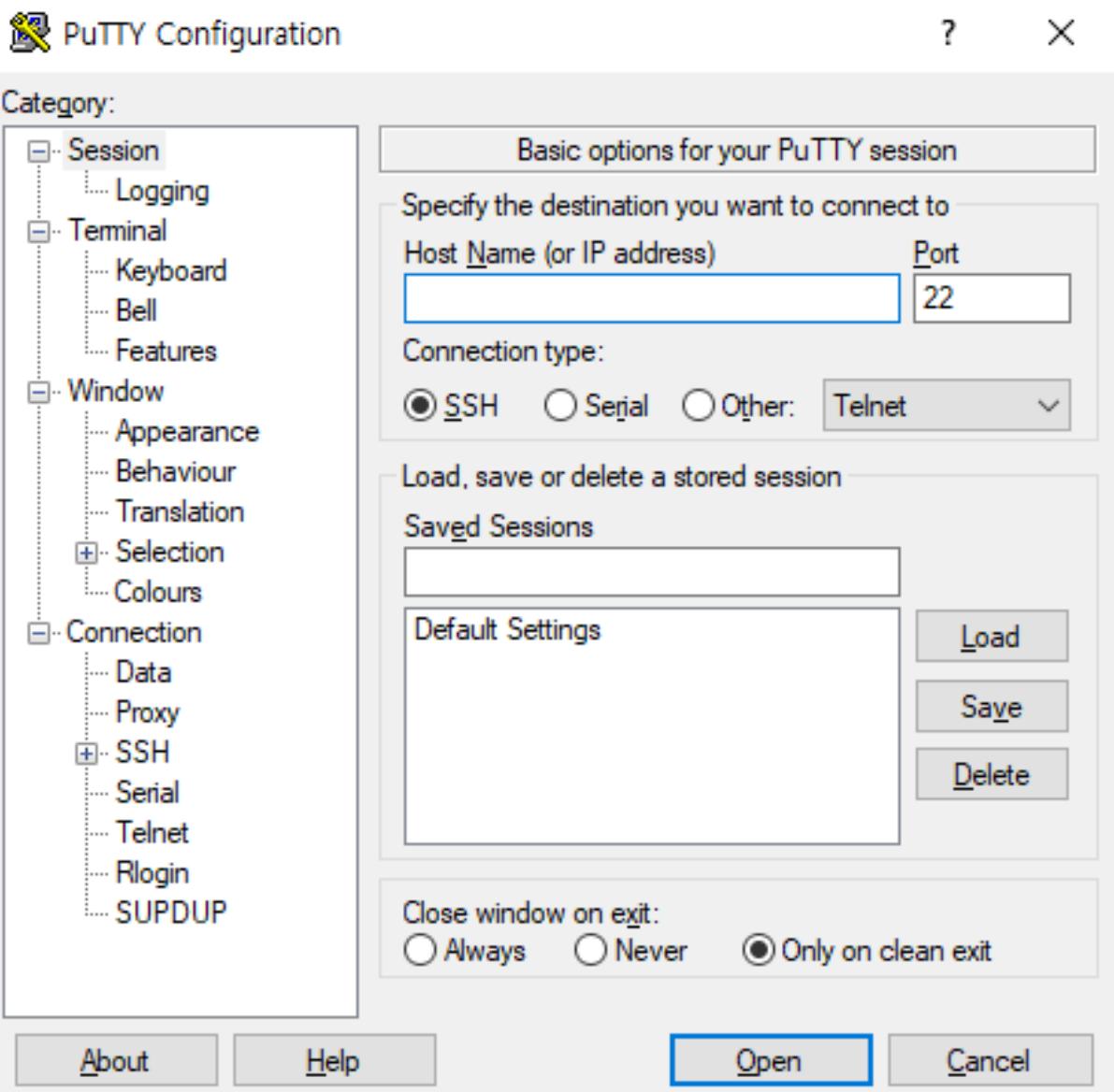
1 다운로드 사이트 접속

<https://www.putty.org/>

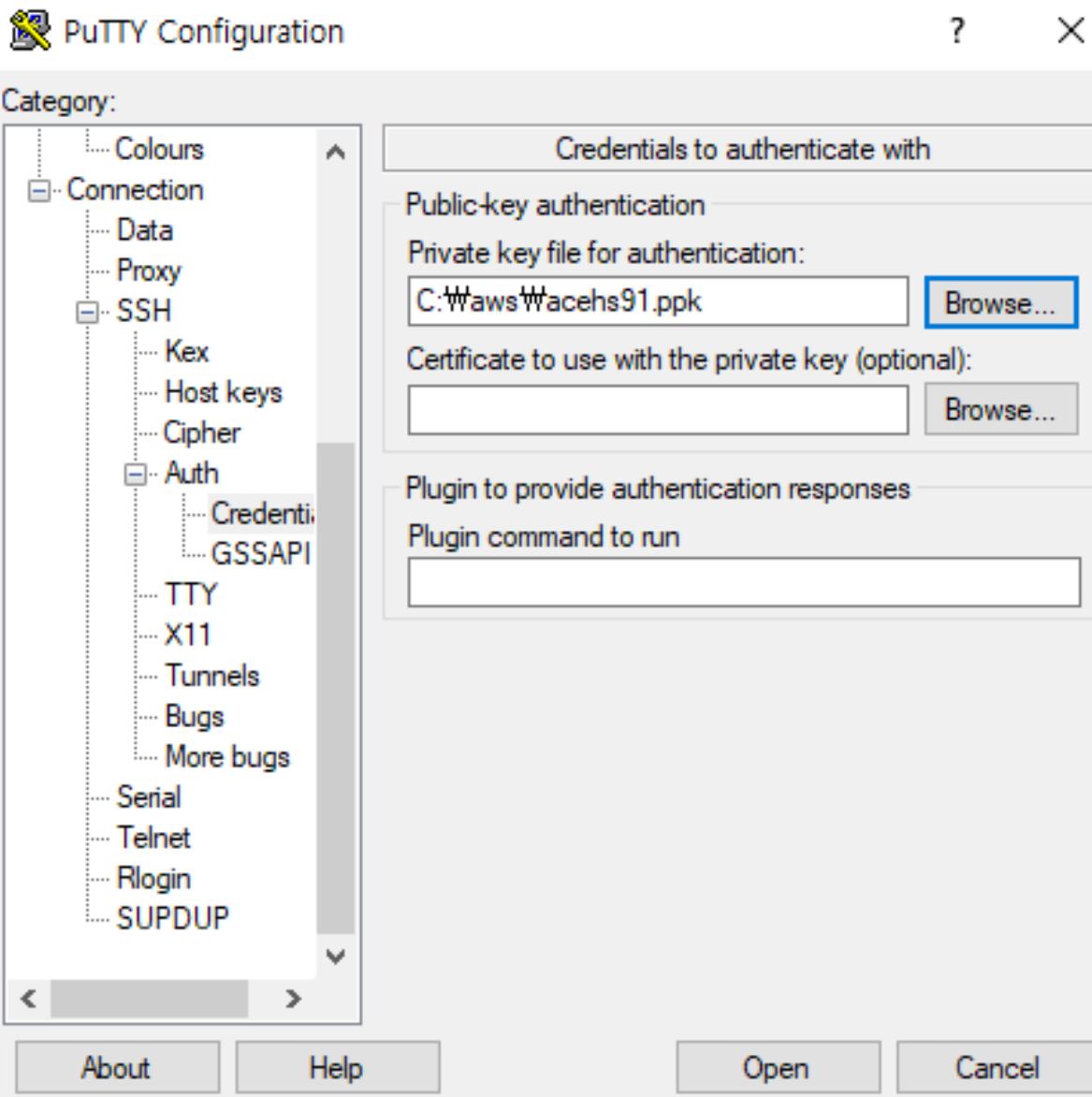
2 다운로드 파일

putty-64bit-0.81-installer.msi ← 이것을 다운받으면 된다.!

설치 후
시작 → PuTTY 검색 → 실행



초기화면



ppk 경로 설정

PuTTY Security Alert

X



The host key is not cached for this server:

18.232.127.18 (port 22)

You have no guarantee that the server is the computer you think it is.

The server's ssh-ed25519 key fingerprint is:

ssh-ed25519 255 SHA256:x4GsGXrDmjqaDFDEAa3AbypKpE8ni2UABZHCeKiUB4

If you trust this host, press "Accept" to add the key to PuTTY's cache and carry on connecting.

If you want to carry on connecting just once, without adding the key to the cache, press "Connect Once".

If you do not trust this host, press "Cancel" to abandon the connection.

Help

More info...

Accept

Connect Once

Cancel

Accept



PuTTY 실행 : ec2-user (캡쳐 사진보면 aws의 linux 2023 이 보인다.)

```
ec2-user@ip-172-31-75-43:~$ login as: ec2-user
Authenticating with public key "acehs91"
,
~\_ #####
~~ \##### Amazon Linux 2023
~~ \###|
~~ \#/ https://aws.amazon.com/linux/amazon-linux-2023
~~ V~' '-->
~~ /
~~ .-
~/ /-
/m/ '
[ec2-user@ip-172-31-75-43 ~]$
```

PuTTY 명령어 - 한줄씩 복사 사용 (밑에 설명 첨부)

```
Verifying : mod_lua-2.4.65-1.amzn2023.0.2.x86_64 13/13

Installed:
apr-1.7.5-1.amzn2023.0.4.x86_64
apr-util-1.6.3-1.amzn2023.0.2.x86_64
apr-util-lmdb-1.6.3-1.amzn2023.0.2.x86_64
apr-util-openssl-1.6.3-1.amzn2023.0.2.x86_64
generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
httpd-2.4.65-1.amzn2023.0.2.x86_64
httpd-core-2.4.65-1.amzn2023.0.2.x86_64
httpd-filesystem-2.4.65-1.amzn2023.0.2.noarch
httpd-tools-2.4.65-1.amzn2023.0.2.x86_64
libbrotli-1.0.9-4.amzn2023.0.2.x86_64
mailcap-2.1.49-3.amzn2023.0.3.noarch
mod_http2-2.0.27-1.amzn2023.0.3.x86_64
mod_lua-2.4.65-1.amzn2023.0.2.x86_64

Complete!
[ec2-user@ip-172-31-75-43 ~]$ sudo systemctl start httpd
[ec2-user@ip-172-31-75-43 ~]$ 
[ec2-user@ip-172-31-75-43 ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr
/lib/systemd/system/httpd.service.
[ec2-user@ip-172-31-75-43 ~]$
```

```
# 1. ec2-user로 로그인  
ec2-user  
# → Amazon Linux의 기본 사용자명  
# → PuTTY 접속 시 "login as:"에 입력하는 거
```

```
# 2. 서버 업데이트  
sudo yum update -y  
# sudo = 관리자 권한으로 실행  
# yum = Amazon Linux 패키지 관리자 (Ubuntu는 apt)  
# update = 설치된 모든 패키지 업데이트  
# -y = 질문에 자동으로 "yes" 답변 (사람 개입 없이 자동 진행)
```

```
# 3. Apache 웹서버 설치  
sudo yum install -y httpd  
# install = 새로운 프로그램 설치  
# httpd = Apache HTTP Server (웹서버)  
# → 웹사이트를 인터넷에 공개할 수 있게 해주는 프로그램  
# → http://18.232.127.18 으로 접속 가능하게 만들어줌
```

```
# 4. Apache 웹서버 시작 (켜기)  
sudo systemctl start httpd  
# systemctl = 리눅스 서비스 관리 명령어  
# start = 서비스 시작 (프로그램 실행)  
# httpd = Apache 웹서버  
# → 지금 당장 웹서버 켜기!
```

```
# 5. 부팅 시 자동 시작 설정  
sudo systemctl enable httpd  
# enable = 자동 시작 활성화  
# → 서버가 재부팅되어도 Apache가 자동으로 켜짐  
# → 안 하면 재부팅할 때마다 수동으로 start 해야 함
```

```
# 6. 웹사이트 파일 저장 폴더로 이동
```

```
cd /var/www/html  
# cd = Change Directory (폴더 이동)  
# /var/www/html = Apache 웹서버의 기본 웹 폴더  
# → 여기에 HTML 파일 넣으면 웹사이트로 보임  
# → index.html 파일이 메인 페이지
```

```
# 7. 현재 위치 확인  
pwd  
# pwd = Print Working Directory  
# → 지금 내가 어느 폴더에 있는지 경로 출력  
# → /var/www/html 나오면 정상
```

```
# 8. 웹페이지 파일 생성  
echo '<h1>Hello, AWS Web Server!</h1>' | sudo tee /var/www/html/index.html  
# echo '내용' = 텍스트 출력  
# | = 파이프 (앞의 출력을 뒤로 전달)  
# sudo tee = 관리자 권한으로 파일에 쓰기  
# /var/www/html/index.html = 메인 페이지 파일  
# → 웹사이트 메인 페이지 만들기!  
# → http://18.232.127.18 접속하면 "Hello, AWS Web Server!" 보임  
...
```

전체 흐름 요약

- 1. 서버 업데이트 (최신 상태로)
↓
- 2. Apache 웹서버 설치
↓
- 3. 웹서버 시작 (켜기)
↓
- 4. 자동 시작 설정 (재부팅해도 켜지게)
↓
- 5. 웹 폴더로 이동 (/var/www/html)
↓
- 6. index.html 파일 만들기 (메인 페이지)

↓

7. 완성! <http://18.232.127.18> 접속하면 보임!

...

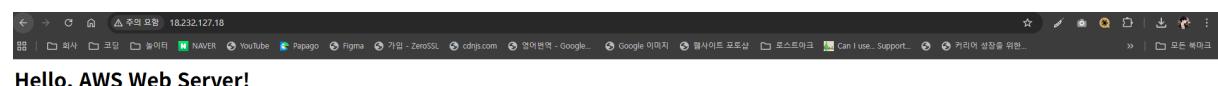
🔥 확인 방법

브라우저 주소창에: <http://18.232.127.18>

→ "Hello, AWS Web Server!" 보이면 성공!

→ 2025-1202 - 제출 김현수

<http://18.232.127.18/>



Hello, AWS Web Server!

노션 원본 링크 : https://www.notion.so/_AWS-PuTTY-2bde7b538c0480a1b129e5fb16308b72?source=copy_link