

AWS EC2 : 서버 구축하기 매뉴얼 1

[AWS EC2 : Oracle DB 서버 구축하기]

AWS 에 Oracle 18c XE 를 사용하여 데이터베이스 서버를 구축하는 과정을 단계별로 설명하겠습니다.

1. AWS 계정 생성 및 로그인

1. AWS 계정 생성

- [AWS 홈페이지](https://aws.amazon.com/ko/)에 접속합니다. : <https://aws.amazon.com/ko/>
- 우측 상단의 "AWS 계정 생성" 버튼을 클릭하고 이메일 주소, 비밀번호, 계정이름 등을 입력하여 계정을 만듭니다. : 발급된 **12 자리 숫자 계정**은 별도로 기억해 둡니다.
- 결제 정보를 등록해야 합니다. "**프리 티어(free tier)**"를 사용하면 무료로 이용할 수 있습니다만, 생성할 서버 환경의 용량이 많이 요구되므로 비용 발생은 부득이하게 발생한다는 것을 고려해야 합니다.

[참고]

AWS EC2 의 **프리 티어(Free Tier)**를 이용할 경우, 다음과 같은 제약 조건이 있습니다. **프리 티어**는 신규 가입한 AWS 사용자에게 제공되며, 첫 12 개월 동안 무료로 사용할 수 있는 자격을 의미합니다. 하지만 사용량과 기능에는 제한이 있으므로 주의가 필요합니다.

1. 사용량 제한

- 인스턴스 유형 제한
 - 무료로 사용할 수 있는 EC2 인스턴스 유형은 **t2.micro** 또는 **t4g.micro** 입니다.
 - t2.micro: x86 기반의 인스턴스
 - t4g.micro: ARM 기반의 인스턴스
 - vCPU: 1 개
 - 메모리: 1GB
- 월간 사용 시간
 - 한 달에 **750 시간**의 사용 시간이 무료입니다.
 - 750 시간은 한 달(30 일) 동안 **1 개의 EC2 인스턴스를 24 시간 계속 실행할 수 있는 시간**입니다.
 - 2 개의 인스턴스를 동시에 실행하면 시간 제한을 초과할 수 있습니다.

2. 스토리지 제한

- EBS 볼륨 제한

- 최대 30GB의 **EBS(Elastic Block Store)** 스토리지를 무료로 사용할 수 있습니다.
- EBS 스토리지 유형은 **General Purpose SSD(gp2)** 또는 **Magnetic** 중에서 선택 가능합니다.
- 스냅샷이나 추가 스토리지 사용 시 비용이 부과됩니다.
- **AWS Elastic File System(EFS)**
 - 5GB의 표준 스토리지를 무료로 사용할 수 있습니다.

3. 데이터 전송 제한

- **인바운드 데이터 전송**
 - 모든 인바운드 트래픽은 무료입니다.
- **아웃바운드 데이터 전송**
 - 매달 **1GB의 아웃바운드 데이터 전송**이 무료입니다.
 - 초과분에 대해 지역별로 비용이 부과됩니다.

4. 리소스 및 서비스 제한

- **퍼블릭 IP 및 Elastic IP**
 - 1개의 인스턴스에서 퍼블릭 IP는 무료로 제공됩니다.
 - 하지만 Elastic IP를 할당한 후 인스턴스를 중지하면 Elastic IP에 요금이 부과됩니다.
- **Load Balancer**
 - 프리 티어에는 로드 밸런서 사용이 포함되지 않습니다. Application Load Balancer(ALB) 등을 사용하면 요금이 부과됩니다.
- **OS 선택**
 - Amazon Linux 2, Ubuntu, RHEL, SUSE 등의 기본 운영 체제 이미지는 무료로 제공됩니다.
 - 일부 서드파티 운영 체제나 AMI(Amazon Machine Image)는 추가 비용이 발생할 수 있습니다.

5. 추가 서비스 제한

- **EC2 외 서비스**
 - 프리 티어는 EC2 뿐 아니라 RDS, S3, Lambda 등의 다른 AWS 서비스에도 적용되지만, 각각의 서비스에 별도의 제한 조건이 있습니다.
 - 예: S3는 5GB의 스토리지를 무료로 제공하지만, 초과 사용 시 비용이 발생합니다.

6. 초과 사용 비용

- 프리 티어 사용량을 초과하면 자동으로 유료 요금이 부과됩니다. 따라서 초과 사용을 방지하려면 다음 사항에 유의하세요:
 - 인스턴스를 사용하지 않을 때는 **중지(Stop)**하거나 **삭제(Terminate)**합니다.
 - 월별 사용량과 비용을 AWS 콘솔의 **Billing and Cost Management**에서 모니터링합니다.

7. 프리 티어 적용 기간

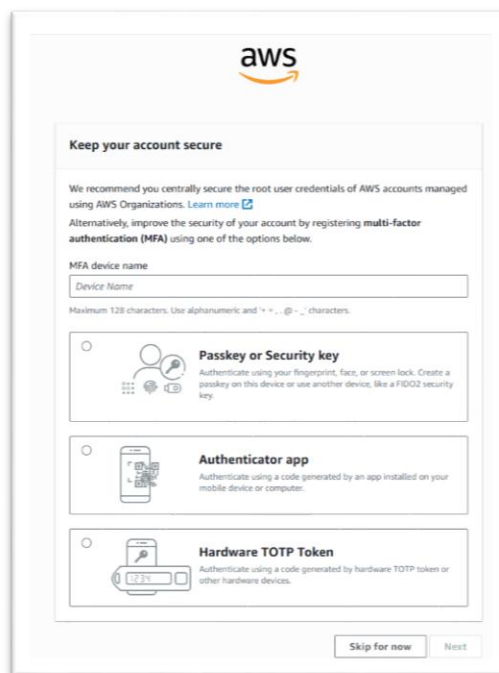
- 신규 AWS 계정에서 가입 후 12 개월 동안 프리 티어 혜택이 적용됩니다.
- 12 개월이 지나면 모든 사용량에 대해 표준 요금이 부과됩니다.

주의 사항

1. 리소스 중지과 삭제:
 - EC2 인스턴스를 중지하지 않으면 750 시간의 제한을 초과할 가능성이 있으니 필요 시 중지하거나 삭제합니다.
2. 비용 초과 방지:
 - AWS Billing 설정에서 **비용 경고 알림**을 활성화하여 사용량을 모니터링합니다.
3. 서비스 간 요금 확인:
 - EC2 외에 사용하는 AWS 서비스(S3, RDS 등)에서도 초과 요금이 발생할 수 있습니다.

2. AWS 콘솔 로그인

- 계정 생성 후 “내 계정 ∨” 을 클릭하여 확장 메뉴에서 “[AWS Management](#) Console”을 클릭하여 로그인합니다. : root email 을 이용하여 로그인합니다.



이 화면은 AWS 계정 보안을 강화하기 위한 다중 인증(MFA, Multi-Factor Authentication) 설정 화면입니다. 사용자는 AWS 루트 계정 보안을 강화하기 위해 MFA 장치를 등록할 수 있습니다.

[참고]

화면 주요 내용

1. MFA 장치 이름 입력

- "MFA device name"이라는 필드에 장치 이름을 입력할 수 있습니다.
- 최대 128 자까지 가능하며, 알파벳, 숫자, 그리고 특수 문자('+ = . , @ - _')를 사용할 수 있습니다.

2. MFA 설정 옵션 아래의 3 가지 인증 옵션 중 하나를 선택하여 MFA 를 활성화할 수 있습니다:

- **Passkey or Security key**
 - 지문, 얼굴 인식 또는 화면 잠금과 같은 인증 방식을 사용하는 보안키(FIDO2 기반) 또는 패스키를 설정합니다.
- **Authenticator app**
 - 스마트폰 또는 컴퓨터에 설치된 **인증 앱**(예: Google Authenticator, Microsoft Authenticator 등)을 사용하여 생성된 코드를 입력하는 방식입니다.
- **Hardware TOTP Token**
 - 하드웨어 TOTP(Time-Based One-Time Password) 토큰 장치에서 생성된 코드를 입력하는 방식입니다.

3. Skip for now

- 화면 하단의 **"Skip for now"** 버튼을 클릭하면 MFA 설정을 건너뛸 수 있습니다. 다만, 계정 보안을 위해 설정하는 것이 필수입니다.

활용 및 권장 사항

- AWS 계정 보안을 위해 **Authenticator app** 옵션을 사용하는 것이 가장 일반적이며 설정이 간편합니다. : 모바일 Play Store 에서 인증앱을 선택해서 설치하고 진행합니다.
- MFA 를 설정하면 AWS 계정 접근 시 추가 인증이 필요하므로 보안 수준이 크게 향상됩니다.
- 장치 이름은 사용자가 기억하기 쉽도록 지정합니다(예: "MyPhone-MFA").

-
- 처음 로그인 이후에는 **AWS 콘솔로 바로 접속해서**
<https://aws.amazon.com/ko/console/>
우측 상단의 **"콘솔에 로그인"** 버튼을 클릭하여 로그인합니다. : **Account ID(12 자리 숫자 계정)**이 기억나지 않으면 아래쪽의 **"Sign in using root user email"** 을 이용하여 로그인합니다.

2. EC2 인스턴스 생성

1. EC2 서비스 선택

- AWS 콘솔에서 검색창에 "EC2"를 입력하고 클릭합니다.

2. 인스턴스 시작

- 우측 상단의 "인스턴스 시작" 버튼을 클릭합니다. >> 이름 입력 : 만들 서버의 이름

3. 운영 체제 선택

- Amazon Linux 2 또는 Ubuntu 20.04 를 선택합니다.
- 주의: Oracle Database 는 리눅스 기반 운영 체제에서 설치됩니다.

4. 인스턴스 유형 선택 및 스토리지 용량 지정

- 프리 티어를 사용하려면 t2.micro 를 선택합니다.
- 프리 티어 최대 용량인 30 기가로 지정합니다.
- Oracle DB Server 실행하려면 t2.medium 이상을 선택합니다.
- EBS : 50G 이상 지정

5. 키 페어 생성

- 새 키 페어를 생성하고 다운로드합니다. 이 파일은 외부에서 EC2 에 접속하는 데 필요합니다. : OpenSSH (pem) 타입으로 생성함

6. 보안 그룹 설정

- SSH(포트 22)와 Oracle XE 가 사용하는 포트(기본적으로 1521)를 허용합니다.
- 소스는 My IP 로 설정하여 본인의 IP 만 접속 가능하게 설정합니다.

7. 인스턴스 시작

- 설정을 확인한 후 인스턴스를 시작합니다.

AWS EC2 에서 인스턴스를 생성한 후, 보안 그룹에서 인바운드 규칙을 추가하는 방법과 순서

개념 요약

- 보안 그룹(Security Group): AWS EC2 인스턴스에 적용되는 가상 방화벽
- 인바운드 규칙(Inbound Rule): 외부에서 EC2 로 들어오는 트래픽 허용 조건을 정의
- 포트 번호나 프로토콜에 따라 외부 접근을 제한하거나 허용할 수 있음

순서대로 따라하기: 인바운드 규칙 추가

1단계: AWS 콘솔 접속

- <https://console.aws.amazon.com> 에 로그인

2단계: EC2 대시보드로 이동

- 상단 검색창에서 "EC2" 입력 → EC2 서비스 클릭
- 왼쪽 사이드바에서 인스턴스 클릭
- 인바운드 규칙을 추가할 인스턴스를 선택

3단계: 보안 그룹 확인 및 이동

- 선택한 인스턴스 하단에 있는 **설명 탭**에서 “**보안**” 섹션 확인
- **보안 그룹** 이름(예: launch-wizard-1)을 클릭 → 보안 그룹 세부 화면으로 이동

4단계: 인바운드 규칙 탭에서 편집

- 상단 탭 중 **인바운드 규칙** 클릭
- 오른쪽 상단의 **편집(Edit inbound rules)** 버튼 클릭

5단계: 규칙 추가

필요한 포트에 따라 규칙을 추가합니다..

유형(Type)	프로토콜	포트 범위	소스	설명
SSH	TCP	22	내 IP	리눅스 터미널 접속용
HTTP	TCP	80	0.0.0.0/0	웹 서버 기본 포트
HTTPS	TCP	443	0.0.0.0/0	보안 웹 접속용
Oracle DB	TCP	1521	0.0.0.0/0 (또는 내 IP)	Oracle 접속용
Oracle EM	TCP	5500	0.0.0.0/0 (또는 내 IP)	Oracle 웹 관리자 접속용

- Source 는 가급적 “**내 IP**” 또는 신뢰된 IP 만 지정하는 것이 보안에 좋습니다.
 - “내 IP”를 선택하면 자동으로 내 공인 IP 가 입력됩니다.
 - “0.0.0.0/0”은 모든 IP 에 개방되므로 테스트용으로만 사용하거나 꼭 필요한 경우에만 씁니다.

6단계: 저장

- 모든 규칙을 입력했으면, 하단의 **규칙 저장(Save rules)** 버튼 클릭

결과 확인

이제 외부에서 해당 포트로 EC2 인스턴스에 접근이 가능해졌습니다. 예를 들어:

- SSH: `ssh -i key.pem ubuntu@<EC2 IP>`
- 웹: `http://<EC2 IP>`
- Oracle SQL Developer: 1521 포트 연결
- Oracle EM: `https://<EC2 IP>:5500/em`

팁: 보안 주의사항

- **0.0.0.0/0** 은 위험할 수 있으므로 꼭 필요한 경우에만 사용하세요.
- 테스트 후에는 포트를 닫거나 IP 범위를 제한하세요.
- 고정 IP 가 있는 경우에는 해당 IP 만 허용하는 것이 좋습니다.

3. EC2 인스턴스에 접속

AWS EC2 인스턴스에 접속하는 방법은 여러 가지가 있습니다.

아래는 AWS 콘솔에서 연결, SSH 클라이언트 사용, PuTTY와 PuTTYgen 사용에 대한 설명입니다.

1. AWS 콘솔에서 연결

1-1. AWS 관리 콘솔에 로그인

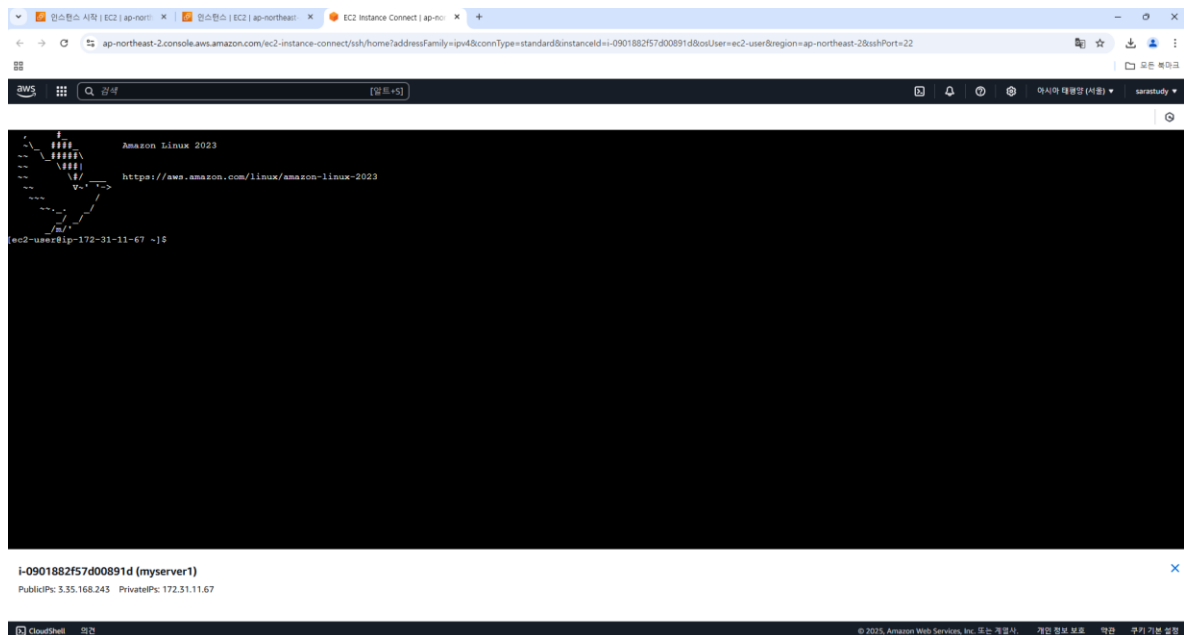
1. AWS 계정으로 [AWS 관리 콘솔](#)에 로그인합니다.
2. EC2 대시보드로 이동합니다:
 - "서비스" 메뉴에서 "EC2"를 클릭합니다.
 - "인스턴스"를 선택합니다.

1-2. 인스턴스 선택 및 연결

1. EC2 인스턴스 목록에서 연결하려는 인스턴스를 선택합니다.
2. 상단의 "연결" 버튼을 클릭합니다.
3. "브라우저 기반 SSH 클라이언트"를 선택한 후 "연결" 버튼을 클릭합니다.

1-3. 브라우저 터미널

- AWS에서 제공하는 웹 기반 SSH 클라이언트가 실행됩니다.
- 추가 소프트웨어 없이 AWS 콘솔 내에서 인스턴스에 연결됩니다.



2. SSH 클라이언트 사용

2-1. 사전 준비

1. 로컬 컴퓨터에 SSH 클라이언트가 설치되어 있어야 합니다.
 - 대부분의 Linux 및 macOS에는 기본적으로 SSH 클라이언트가 포함되어 있습니다.
 - Windows 사용자는 **Windows 터미널** 또는 **PowerShell**에서 SSH를 사용할 수 있습니다.
2. AWS에서 EC2 생성 시 다운로드한 .pem 키 파일을 준비합니다.

2-2. 키 파일 권한 설정

1. 다운로드한 .pem 파일의 권한을 제한합니다:

```
chmod 400 your-key-file.pem
```

2-3. SSH 명령으로 접속

1. 터미널(또는 PowerShell)을 열고 아래 명령을 실행합니다:

```
ssh -i "your-key-file.pem" ec2-user@<인스턴스-퍼블릭-IP>
```

- your-key-file.pem: 키 파일의 경로 (" "는 파일명에 공백이 있으면 반드시 사용함, 권장)
- <인스턴스-퍼블릭-IP>: EC2 인스턴스의 퍼블릭 IP 주소

2. 연결에 성공하면 SSH를 통해 EC2 인스턴스에 접속됩니다.

3. PuTTY와 PuTTYgen 사용

3-1. PuTTY와 PuTTYgen 설치

1. PuTTY와 PuTTYgen을 다운로드하여 설치합니다.
 - [PuTTY 공식 사이트](https://www.putty.org/)에서 다운로드 가능합니다.:
 - "Download PuTTY" 클릭 >> 3개의 파일을 다운받음

We also publish the latest PuTTY installers for all Windows architectures as

MSI ("Windows Installer")

64-bit x86: [putty-64bit-0.82-installer.msi](#) [\(signature\)](#)

64-bit Arm: [putty-arm64-0.82-installer.msi](#) [\(signature\)](#)

32-bit x86: [putty-0.82-installer.msi](#) [\(signature\)](#)

Unix source archive

putty.exe (the SSH and Telnet client itself)

64-bit x86: [putty.exe](#) [\(signature\)](#)

puttygen.exe (a RSA and DSA key generation utility)

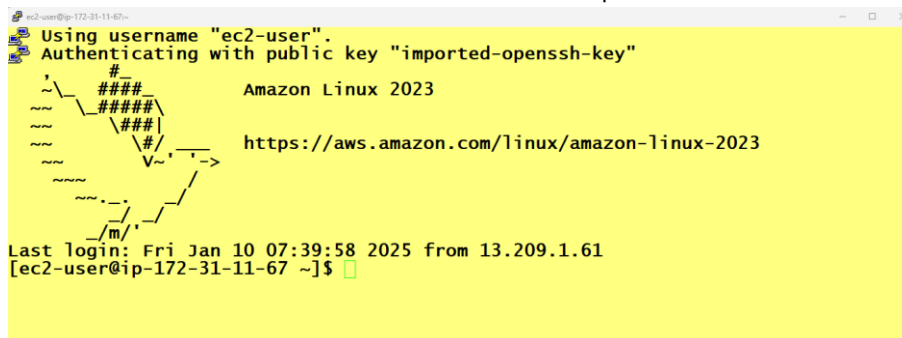
64-bit x86: [puttygen.exe](#) [\(signature\)](#)

3-2. PuTTYgen으로 .pem 파일 변환

1. PuTTYgen 을 실행합니다.
2. "Load" 버튼을 클릭하고 .pem 키 파일을 선택합니다.
3. 변환 후 "Save private key" 버튼을 클릭하여 .ppk 형식으로 키 파일을 저장합니다.

3-3. PuTTY로 접속

1. PuTTY 를 실행합니다.
2. 세션 설정:
 - "Host Name (or IP address)"에 EC2 인스턴스의 퍼블릭 IP 를 입력합니다.
예 : ec2-user@3.35.168.243
 - "Connection type"으로 "SSH"를 선택합니다.
3. 키 파일 등록:
 - 왼쪽 메뉴에서 "Connection > SSH > Auth > Credential"로 이동합니다.
 - "Private key file for authentication > Browse..."에서 .ppk 키 파일을 선택합니다.
4. "Open" 버튼을 클릭하여 인스턴스에 연결합니다.
5. 첫 연결 시, 보안 경고 메시지가 나타나면 "Accept"를 클릭합니다.



비교 및 선택 :

- **AWS 콘솔에서 연결:** 브라우저만으로 접속 가능하며 편리하지만 기능이 제한적입니다.
- **SSH 클라이언트 사용:** 터미널 환경에서 직접 작업할 수 있어 효율적이며, 대부분의 개발자에게 적합합니다.
- **PuTTY와 PuTTYgen 사용:** Windows 사용자가 주로 사용하며 GUI 기반 설정을 제공합니다.

4. Oracle 18c XE 다운로드 및 설치

AWS Linux, Ubuntu 에서는 oracle preinstall 이 지원되지 않아 직접 의존성을 설치해도 오라클이 정상 설치되지 않는 것으로 확인되었습니다. 그러므로 두번째 도커를 사용한 설치 방법으로 진행합니다.

1. 직접 다운로드 및 설치

1-1. 사전 준비

1. EC2 인스턴스 생성:

- Amazon Linux 2, Ubuntu, CentOS 등 지원하는 리눅스 배포판 중 하나를 선택하여 EC2 인스턴스를 생성합니다.
- 스토리지는 최소 20GB 이상 할당하세요.

2. 시스템 요구사항:

- 메모리: 최소 1GB (권장 2GB 이상)
- 스토리지: 최소 16GB 이상 (권장 20GB 이상)
- 프로세서: x86-64 아키텍처

3. 보안 그룹 설정:

- TCP 1521 포트(Oracle Listener 포트)를 열어 외부 접속을 허용합니다.
- 22 번 포트(SSH)를 열어 EC2 에 접근합니다.

1-2. AWS Linux에서 직접 다운로드 및 설치

1. Oracle 18c XE RPM 다운로드

1. EC2 인스턴스에 SSH 로 접속:

```
ssh -i "your-key-file.pem" ec2-user@<EC2_PUBLIC_IP>
```

2. Oracle XE 설치 파일 다운로드:

Oracle Database 18c Express Edition(XE)의 RPM 패키지를 다운로드하려면, Oracle 공식 웹사이트에서 제공하는 다운로드 페이지를 통해 진행하셔야 합니다.

Oracle 은 직접적인 파일 다운로드 링크를 제공하지 않으며, 다운로드를 위해 Oracle 계정으로 로그인해야 합니다.

다운로드 절차:

1. Oracle 계정 로그인:

- [Oracle Database Express Edition \(XE\) 18c 다운로드 페이지](#)에 접속합니다.
- Oracle 계정으로 로그인합니다. 계정이 없으신 경우, 무료로 생성할 수 있습니다.

2. 라이선스 동의 및 파일 다운로드:

- 로그인 후, 라이선스 동의란에 체크한 후, 운영 체제에 맞는 설치 파일을 선택합니다.
- Linux x64 용 oracle-database-xe-18c-1.0-1.x86_64.rpm 파일을 선택하여 다운로드합니다.

3. 파일 전송:

- 다운로드한 파일을 scp 명령어 등을 사용하여 AWS EC2 인스턴스로 전송합니다.

4. EC2 인스턴스에서 설치:

- EC2 인스턴스에 접속하여, 전송한 RPM 파일이 있는 디렉터리로 이동합니다.
- 필요한 의존성 패키지를 설치한 후, 다음 명령어로 Oracle Database XE 를 설치합니다:

```
sudo yum -y localinstall oracle-database-xe-18c-1.0-1.x86_64.rpm
```

설치가 완료되면, 데이터베이스를 구성합니다:

```
sudo /etc/init.d/oracle-xe-18c configure
```

- 구성 과정에서 관리자 비밀번호 등을 설정하게 됩니다.

2. Oracle Preinstallation 패키지 설치 (오라클 설치 전 수행)

1. Amazon Linux 2 의 경우 **preinstall** 제한 조건:

- Amazon Linux 2 는 Oracle 의 공식 Preinstallation 패키지를 지원하지 않습니다. 대신 CentOS 기반 패키지를 수동으로 설치하거나, 수동으로 필요한 의존성을 설정해야 합니다.
- CentOS/RedHat 환경에서는 오라클 설치 전에 다음 명령 먼저 설치:

```
sudo yum install -y oracle-database-preinstall-18c
```

2. Amazon Linux 에서 preinstall 이 제공되지 않으므로 직접 필요한 의존성 설치:

```
sudo yum install -y binutils compat-libcap1 gcc gcc-c++ glibc glibc-devel \
ksh libaio libaio-devel libX11 libXau libXi libXtst make sysstat
```

3. 데이터베이스 시작 및 확인

1. Oracle 데이터베이스 서비스 시작:

```
sudo systemctl start oracle-xe-18c
```

2. 상태 확인:

```
sudo systemctl status oracle-xe-18c
```

2. 도커를 이용한 설치

2-1. 사전 준비

1. EC2 인스턴스에 Docker 설치: (aws Linux 일때)

```
sudo yum update -y
sudo yum install docker -y
sudo service docker start
sudo usermod -a -G docker ec2-user
```

2. Docker Compose 설치(선택):

```
sudo curl -L "https://github.com/docker/compose/releases/download/2.0.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose
```

우분투 일때는 다음과 같이 실행합니다:

```
sudo apt update -y
sudo apt install docker.io -y
sudo systemctl start docker
sudo usermod -aG docker $USER
```

설명:

1. **sudo apt update -y**: Ubuntu 에서 패키지 목록을 업데이트하는 명령어입니다.
2. **sudo apt install docker.io -y**: Docker 를 설치합니다. Ubuntu 에서는 docker.io 패키지를 사용합니다.
3. **sudo systemctl start docker**: Docker 서비스를 시작합니다.
4. **sudo usermod -aG docker \$USER**: 현재 사용자를 Docker 그룹에 추가합니다. \$USER 를 사용하면 현재 로그인한 사용자의 이름이 자동으로 사용됩니다.

주의: 이 명령어를 실행한 후에는 변경 사항을 적용하려면 로그아웃 후 다시 로그인해야 합니다.

2-2. 도커 이미지로 Oracle XE 설치

1. Docker Hub에서 Oracle XE 이미지 다운로드

1. Oracle XE Docker 이미지 다운로드:

```
newgrp docker // 현재 세션을 로그아웃한 뒤 다시 로그인하거나, 다음 명령으로 쉘을 재시작합니다
docker pull gvenzl/oracle-xe:18.4.0-full
```

2. Docker 컨테이너 실행

1. Oracle 컨테이너 실행:

```
docker run -d --name oracle-xe18c ₩
-p 1521:1521 ₩
-p 5500:5500 ₩
-e ORACLE_PASSWORD=YourPasswordHere ₩
```

```
gvenzl/oracle-xe:18.4.0-full
```

- -p 1521:1521: 호스트와 컨테이너 간 포트 매핑
- -e ORACLE_PWD=your_password: 데이터베이스 관리자 비밀번호 설정

3. 컨테이너 상태 확인

1. 실행 중인 컨테이너 확인:

```
docker ps
```

2. 컨테이너 내부 접속:

```
docker exec -it oracle-xe18c bash
```

3. 데이터베이스 상태 확인:

```
sqlplus sys/YourPasswordHere@localhost/XEPDB1 as sysdba
```

[참고]

AWS 에 구축된 Oracle 데이터베이스 서버에 클라이언트 PC 에서 SQL Developer 툴로 접속하는 방법을 단계별로 설명합니다.

[AWS EC2 에서 Oracle 21c XE (Express Edition)를 설치하는 방법]

Oracle 21c XE 는 현재 Oracle 공식 홈페이지에서 리눅스용 RPM 패키지로 제공되며, 주로 Oracle Linux 또는 Red Hat 기반의 OS 에서 원활하게 설치됩니다.

1. EC2 인스턴스 생성

- **OS 선택:** Oracle Linux 8.x, 또는 RHEL 8.x (CentOS 8 도 가능하지만 권장되지 않음)
- **인스턴스 타입:** 최소 t3.medium 이상 권장 (2 vCPU / 4 GB RAM 이상)
- **스토리지:** 최소 20GB 이상 권장
- **보안 그룹 설정:** 아래 포트 오픈
 - SSH: 22
 - Oracle Listener: 1521
 - Oracle EM Express: 5500 (선택)

2. Oracle 21c XE 설치 파일 다운로드

Oracle 공식 홈페이지에서 다운로드해야 하며, wget 등으로 직접 다운로드 는 불가능합니다.

1. [Oracle XE 21c 다운로드 링크](#) 에서 .rpm 파일 다운로드
2. 로컬에서 EC2 로 업로드 (scp 명령 사용 또는 S3 업로드 후 EC2 에서 wget)

```
scp oracle-database-xe-21c-1.0-1.ol8.x86_64.rpm ec2-user@<EC2_PUBLIC_IP>:/home/ec2-user/
```

3. 의존 패키지 설치 및 Oracle XE 설치

1) 필수 패키지 설치

```
sudo dnf install -y oracle-database-preinstall-21c
```

Oracle Linux 가 아닌 경우 EPEL 추가나 다른 의존성 설정이 필요할 수 있음.

2) Oracle XE 설치

```
sudo dnf install -y oracle-database-xe-21c-1.0-1.ol8.x86_64.rpm
```

설치가 완료되면 초기 설정을 진행합니다.

4. Oracle XE 초기화 (설정)

```
sudo /etc/init.d/oracle-xe-21c configure
```

프롬프트에 따라 SYS/SYSTEM 패스워드 등을 입력합니다.

5. 서비스 확인 및 실행

```
sudo systemctl enable oracle-xe-21c  
sudo systemctl start oracle-xe-21c
```

상태 확인:

```
sudo systemctl status oracle-xe-21c
```

6. 접속 테스트

```
sqlplus system/<비밀번호>@localhost/XEPDB1
```

또는 SQL Developer 등 외부 도구로 접속 시:

- 호스트: EC2 퍼블릭 IP
- 포트: 1521
- 서비스 이름: XEPDB1
- 사용자: system 또는 sys
- 비밀번호: 설정한 비밀번호

7. Oracle EM Express 접속 (웹 UI)

- 접속 주소: https://<EC2_IP>:5500/em
- 브라우저에서 접속 시 보안 예외 허용 필요

참고사항

- t2.micro 와 같은 프리티어에서는 메모리 부족으로 설치 실패할 수 있습니다.

- 방화벽(firewalld 등) 실행 중이라면 포트 허용 필요:

```
sudo firewall-cmd --permanent --add-port=1521/tcp  
sudo firewall-cmd --permanent --add-port=5500/tcp  
sudo firewall-cmd --reload
```


[AWS EC2 의 Ubuntu OS 에 Oracle 21c XE(Express Edition)를 Docker 컨테이너로 설치하는 과정]

전제 조건

- AWS EC2 인스턴스는 **Ubuntu 20.04 또는 22.04** 기준입니다.
- Docker 와 Docker Compose 는 미리 설치되어 있지 않다고 가정합니다.
- Oracle 21c XE 는 Oracle 이 공식 Docker 이미지를 **공개하지 않기 때문에**, Dockerfile 을 이용해 직접 빌드하거나 GitHub 에서 제공되는 Dockerfile 을 사용하는 방법을 이용합니다.

전체 순서 요약

1. EC2 인스턴스 생성 및 접속
2. Docker & Docker Compose 설치
3. Oracle XE Docker 이미지 준비
4. Oracle XE 컨테이너 실행
5. 테스트 및 접속 확인

1. EC2 인스턴스 생성 및 접속

1. AWS 콘솔 → EC2 → 인스턴스 시작
2. **AMI**: Ubuntu Server 20.04 또는 22.04 LTS 선택
3. **인스턴스 유형**: t3.medium 이상 권장 (메모리 최소 4GB)
4. **보안 그룹**: 다음 포트 열기
 - SSH: 22
 - Oracle Listener: 1521
 - Oracle EM Express: 5500 (웹 UI)
5. 인스턴스 시작 후, SSH 로 접속:

```
ssh -i "your-key.pem" ubuntu@<your-ec2-public-ip>
```

2. Docker 및 Docker Compose 설치

```
# 시스템 업데이트
sudo apt update && sudo apt upgrade -y

# Docker 설치
sudo apt install -y docker.io
```

```
# Docker 자동 시작 설정
sudo systemctl enable docker
sudo systemctl start docker

# 현재 사용자도 docker 그룹에 추가 (로그아웃 후 재접속 필요)
sudo usermod -aG docker $USER

# Docker Compose 설치 (버전 최신 확인 가능)
sudo curl -L "https://github.com/docker/compose/releases/download/v2.24.6/docker-compose-$(uname
-s)-$(uname -m)" -o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose
```

3. Oracle 21c XE Docker 이미지 준비

방법 1: GitHub Dockerfile로 직접 빌드 (추천)

```
# git 설치
sudo apt install -y git

# Oracle XE 21c Dockerfile 클론
git clone https://github.com/oracle/docker-images.git
cd docker-images/OracleDatabase/SingleInstance/dockerfiles

# Oracle XE 21c 압축파일 다운로드 (수동 필요)
# https://www.oracle.com/database/technologies/oracle-database-software-downloads.html
# OracleXE21c_Linux.x64.zip 파일을 다운로드 후 SCP 로 EC2 로 업로드

# 예: 로컬에서 EC2 로 전송
scp -i your-key.pem OracleXE21c_Linux.x64.zip ubuntu@<your-ec2-public-ip>:~

# 압축파일을 적절한 위치로 이동
mv ~/OracleXE21c_Linux.x64.zip ./21.3.0/

# Docker 이미지 빌드
cd 21.3.0
./buildContainerImage.sh -v 21.3.0 -x
```

빌드가 완료되면 oracle/database:21.3.0-xe 이미지가 생성됩니다.

4. Oracle XE Docker 컨테이너 실행

```
docker run -d --name oracle-xe \
-p 1521:1521 -p 5500:5500 \
-e ORACLE_PWD=Oracle123 \
oracle/database:21.3.0-xe
```

- ORACLE_PWD=Oracle123: SYS, SYSTEM 계정의 초기 비밀번호
- 컨테이너가 초기화되고 DB가 생성되기까지 몇 분 소요될 수 있습니다.

5. Oracle XE 접속 확인

SQL*Plus 접속 (컨테이너 내부에서)

```
docker exec -it oracle-xe bash
sqlplus sys/Oracle123@localhost:1521/XEPDB1 as sysdba
```

SQL Developer 등 외부 클라이언트 접속 정보

- 호스트: EC2 퍼블릭 IP
- 포트: 1521
- SID or 서비스 이름: XEPDB1
- 사용자: system 또는 sys
- 비밀번호: Oracle123

(선택) Oracle EM Express 웹 접속

브라우저에서 다음 주소로 접속:

<https://<EC2 퍼블릭 IP>:5500/em>

첫 접속 시 보안 예외 허용 필요 (Self-signed SSL)

보안 팁

- 실제 운영에 사용할 경우, 방화벽 설정을 통해 접속 IP 제한
- 비밀번호는 강력하게 설정 및 주기적 변경 권장

참고 사항

- Oracle Docker 이미지는 Oracle에서 공식적으로 Docker Hub에 올려놓지 않기 때문에 직접 빌드하는 방식이 표준입니다.
- EC2에 swap 메모리를 설정하면 메모리 부족 방지에 도움될 수 있습니다.

1. SQL Developer 설치

1. SQL Developer 다운로드

- Oracle 공식 웹사이트에서 SQL Developer 를 다운로드합니다.

[SQL Developer 다운로드 링크](#)

2. SQL Developer 설치

- 다운로드한 파일을 압축 해제하고 실행합니다. 설치 과정은 간단하며, 실행 파일을 더블 클릭하면 실행됩니다.

2. AWS Oracle 데이터베이스 서버 정보 확인

1. EC2 퍼블릭 IP 확인

- AWS 관리 콘솔에서 EC2 인스턴스로 이동하여 퍼블릭 IP 주소를 확인합니다.

2. Oracle XE 포트 확인

- Oracle Database 는 기본적으로 **1521 번 포트**를 사용합니다.
- AWS EC2 보안 그룹에서 **포트 1521** 이 열려 있는지 확인하세요.
 - EC2 > 보안 그룹 > 인바운드 규칙 편집 > **포트 1521** 추가

3. SQL Developer에서 접속 설정

1. SQL Developer 실행

- SQL Developer 를 실행하고 시작 화면에서 **새 연결(New Connection)** 버튼을 클릭합니다.

2. 새 연결 설정

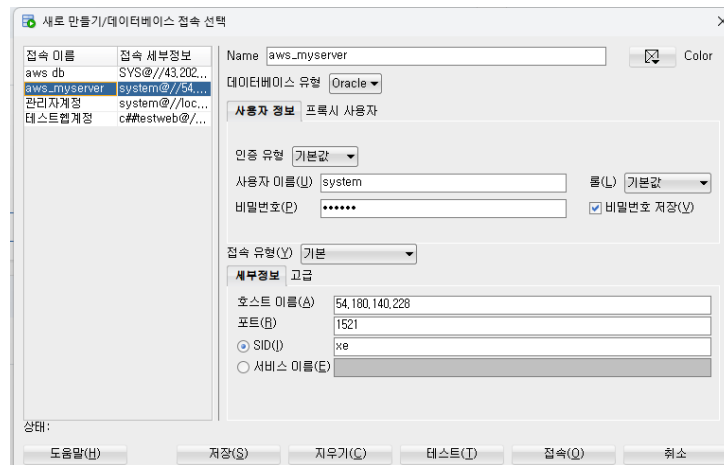
- 연결 이름: 원하는 이름을 입력 (예: AWS_Oracle_XE)
- 사용자 이름: 데이터베이스 계정 이름 (예: test_user)
- 비밀번호: 해당 계정의 비밀번호 (예: test_password)
- 호스트 이름: EC2 인스턴스의 퍼블릭 IP (예: 12.34.56.78)
- 포트: 1521
- SID: XE (Oracle XE 의 기본 SID)

3. 연결 테스트

- "테스트(Test)" 버튼을 클릭하여 연결 상태를 확인합니다.
- "성공(Success)" 메시지가 표시되면 연결에 성공한 것입니다.

4. 연결 저장 및 접속

- "저장(Save)"을 클릭한 후 "연결(Connect)" 버튼을 눌러 데이터베이스에 접속합니다.



[참고]

Spring Boot 웹 애플리케이션에서 AWS EC2 에 배포된 Oracle DB 서버와 연동하기 위한 URL 을 구성하려면 다음과 같이 설정하면 됩니다.

1. Oracle DB URL 구성

Spring Boot 에서 Oracle DB 서버에 접속하기 위한 URL 은 일반적으로 다음 형식으로 지정됩니다:

```
jdbc:oracle:thin:@<HOST>:<PORT>:<SID>
```

- jdbc:oracle:thin: JDBC 드라이버 및 프로토콜.
- <HOST>: AWS EC2 Oracle DB 서버의 퍼블릭 IP 또는 DNS 이름.
- <PORT>: Oracle Listener 가 사용하는 포트 (기본값은 1521).
- <SID>: 데이터베이스의 SID (예: XE).

만약 AWS EC2 의 Oracle DB 서버에 퍼블릭 IP 가 54.123.45.67 이고, 포트가 기본값 1521 이며, SID 가 XE 인 경우, URL 은 다음과 같이 설정합니다:

```
jdbc:oracle:thin:@54.123.45.67:1521:XE
```

2. Spring Boot application.properties 설정

Spring Boot 의 application.properties 파일에 다음과 같이 설정합니다:

```
spring.datasource.url=jdbc:oracle:thin:@54.123.45.67:1521:XE
spring.datasource.username=your_username # 데이터베이스 사용자 이름
spring.datasource.password=your_password # 데이터베이스 비밀번호
```

```
spring.datasource.driver-class-name=oracle.jdbc.OracleDriver
spring.jpa.database-platform=org.hibernate.dialect.OracleDialect
```

또는 Spring Boot application.yml 설정

application.yml 파일을 사용하는 경우, 아래와 같이 설정합니다:

```
spring:
  datasource:
    url: jdbc:oracle:thin:@54.123.45.67:1521:XE
    username: your_username
    password: your_password
    driver-class-name: oracle.jdbc.OracleDriver
  jpa:
    database-platform: org.hibernate.dialect.OracleDialect
```