

# AWS Ubuntu

명령어로 openjdk설치

인스턴스 생성/연결  
Ubuntu2018버전

# 인스턴스 생성 - 우분투 18.04 LTS 버전

**SUSE Linux**  
프리 티어 사용 가능

**SUSE Linux Enterprise Server 12 SP5 (HVM), SSD Volume Type - ami-04aa88aebb9fef83**  
SUSE Linux Enterprise Server 12 Service Pack 5 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.  
루트 디바이스 유형: ebs    가상화 유형: hvm    ENA 활성화: 예

선택

64비트(x86)

**Ubuntu**  
프리 티어 사용 가능

**Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0b9064170e32bde34 (64비트 x86) / ami-026141f3d5c6d2d0c (64비트 Arm)**  
Ubuntu Server 18.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).  
루트 디바이스 유형: ebs    가상화 유형: hvm    ENA 활성화: 예

선택

☒ 64비트(x86)  
☐ 64비트(Arm)

**Amazon Linux**  
프리 티어 사용 가능

**Amazon Linux 2 with .Net Core, PowerShell, Mono, and MATE Desktop Environment - ami-026ff5d827271b86**  
.NET Core 5.0, Mono 6.12, PowerShell 7.1, and MATE DE pre-installed to run your .NET applications on Amazon Linux 2 with Long Term Support (LTS).  
루트 디바이스 유형: ebs    가상화 유형: hvm    ENA 활성화: 예

선택

64비트(x86)

**Ubuntu**

**Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-0d563aeddd4be7fff (64비트 x86) / ami-0bff25b43a4479334 (64비트 Arm)**  
Ubuntu Server 16.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).  
AMI ID: ami-0bff25b43a4479334

선택

## 단계 2: 인스턴스 유형 선택

Amazon EC2는 각 사용 사례에 맞게 최적화된 다양한 인스턴스 유형을 제공합니다. 인스턴스는 애플리케이션을 실행할 수 있는 가상 서버입니다. 이러한 인스턴스에는 CPU, 메모리, 스토리지 및 네트워킹 용량의 다양한 조합이 있으며, 애플리케이션에 사용할 적절한 리소스 조합을 유연하게 선택할 수 있습니다. 인스턴스 유형과 이 인스턴스 유형이 컴퓨팅 요건을 충족하는 방식에 대해 자세히 알아보기.

필터링 기준: 모든 인스턴스 패밀리 현재 세대 열 표시/숨기기

	그룹	유형	vCPUs	메모리 (GiB)	인스턴스 스토리지 (GB)	EBS 최적화 사용 가능	네트워크 성능	IPv6 지원
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS 전용	-	낮음에서 중간	예
<input checked="" type="checkbox"/>	t2	t2.micro 프리 티어 사용 가능	1	1	EBS 전용	-	낮음에서 중간	예
<input type="checkbox"/>	t2	t2.small	1	2	EBS 전용	-	낮음에서 중간	예
<input type="checkbox"/>	t2	t2.medium	2	4	EBS 전용	-	낮음에서 중간	예
<input type="checkbox"/>	t2	t2.large	2	8	EBS 전용	-	낮음에서 중간	예
<input type="checkbox"/>	t2	t2.xlarge	4	16	EBS 전용	-	보통	예
<input type="checkbox"/>	t2	t2.2xlarge	8	32	EBS 전용	-	보통	예
<input type="checkbox"/>	t3	t3.nano	2	0.5	EBS 전용	예	최대 5기가비트	예
<input type="checkbox"/>	t3	t3.micro	2	1	EBS 전용	예	최대 5기가비트	예
<input type="checkbox"/>	t3	t3.small	2	2	EBS 전용	예	최대 5기가비트	예
<input type="checkbox"/>	t3	t3.medium	2	4	EBS 전용	예	최대 5기가비트	예
<input type="checkbox"/>	t3	t3.large	2	8	EBS 전용	예	최대 5기가비트	예

취소


이전

검토 및 시작

다음: 인스턴스 세부 정보 구성

## 단계 7: 인스턴스 시작 검토

인스턴스 시작 세부 정보를 검토하십시오. 이전으로 돌아가서 각 섹션에 대한 변경 내용을 편집할 수 있습니다. 키 페어를 인스턴스에 할당하고 시작 프로세스를 완료하려면 [시작]을 클릭합니다.

 **인스턴스 보안을 개선하십시오. 보안 그룹 launch-wizard-7이(가) 세계에 개방되어 있습니다.**  
인스턴스를 모든 IP 주소에서 액세스할 수 있습니다. 보안 그룹 규칙을 업데이트하여 알려진 IP 주소에서만 액세스를 허용하는 것이 좋습니다.  
실행 중인 애플리케이션이나 서비스에 쉽게 액세스할 수 있도록 보안 그룹에서 추가 포트를 열 수도 있습니다. 예를 들어, 웹 서버용으로 HTTP(80)를 엽니다. 보안 그룹 편집

**AMI 세부 정보**

**Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0b9064170e32bde34**  
Ubuntu Server 18.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).  
루트 디바이스 유형: ebs    가상화 유형: hvm

AMI 편집

**인스턴스 유형**

인스턴스 유형	ECU	vCPUs	메모리 (GiB)	인스턴스 스토리지 (GB)	EBS 최적화 사용 가능	네트워크 성능
t2.micro	-	1	1	EBS 전용	-	Low to Moderate

인스턴스 유형 편집

**보안 그룹**

보안 그룹 이름: launch-wizard-7

설명: launch-wizard-7 created 2021-05-18T11:12:20.910+09:00

유형	프로토콜	포트 범위	소스	설명
SSH	TCP	22	0.0.0.0/0	

보안 그룹 편집

**인스턴스 세부 정보**

인스턴스 유형	프로토콜	포트 범위	소스	설명
SSH	TCP	22	0.0.0.0/0	

인스턴스 세부 정보 편집

취소

이전

시작하기

## 인스턴스 (1/3) 정보



연결

인스턴스 상태 ▼

작업 ▼

인스턴스 사

Q 인스턴스 필터링

	Name ▼	인스턴스 ID	인스턴스 상태 ▼	인스턴스 유형 ▼	상태 검사	경보 상태	가용 영역 ▼	퍼블릭 IPv4 DNS ▼	퍼블릭 IPv4 ... ▼	탄력적 I
<input type="checkbox"/>	ubuntu18	i-0fea70ba27d24e36a	⊖ 종료됨	t2.micro	-	경보 없음	+	us-east-2b	-	52.14.73.212
<input type="checkbox"/>	ubuntu18	i-0dba3fa097c46e635	⊖ 중지됨	t2.micro	-	경보 없음	+	us-east-2c	-	18.220.255.205
<input checked="" type="checkbox"/>	ubuntu18-2	i-09738b94b16cb7c95	✔ 실행 중	t2.micro	-	경보 없음	+	us-east-2c	ec2-18-189-43-235.us-...	18.189.43.235

```
ubuntu@ip-172-31-43-148: ~
Usage of /: 14.7% of 7.69GB  Users logged in: 0
Memory usage: 19%          IP address for eth0: 172.31.43.148
Swap usage: 0%

0 packages can be updated.
0 of these updates are security updates.

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-172-31-43-148:~$
```

파워셸 연결

Open jdk설치

## Apt-get 업데이트

```
ubuntu@ip-172-31-43-148:~$ sudo apt-get update
```

```
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic InRelease  
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease [48.5 kB]  
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease [51.6 kB]  
Get:4 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]  
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [8570 kB]  
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic/universe Translation [48.5 kB]
```

자바 버전확인. Jre만 깔려있음

```
ubuntu@ip-172-31-43-148:~$ java -version
```

```
Command 'java' not found, but can be installed with:
```

```
sudo apt install default-jre  
sudo apt install openjdk-11-jre-headless  
sudo apt install openjdk-8-jre-headless
```

Openjdk설치

```
ubuntu@ip-172-31-43-148:~$ sudo apt install openjdk-8-jdk
```

```
Reading package lists... Done  
Building dependency tree  
Reading state information... Done
```

# Jdk 환경변수설정

```
ubuntu@ip-172-31-43-148:~$ cd /usr/lib/jvm/java-8-openjdk-amd64
ubuntu@ip-172-31-43-148:/usr/lib/jvm/java-8-openjdk-amd64$ sudo vi /etc/environment
```

1. Jdk파일위치로 이동
2. 환경변수 수정

```
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin"

~
1,1 All
```

Shift + a

커서를 맨뒤로 옮긴 후  
insert모드로 변경

```
JAVA_HOME="/usr/lib/jvm/java-8-openjdk-amd64"
```

변수 추가



```
ubuntu@ip-172-31-43-148: /usr/lib/jvm/java-8-openjdk-amd64
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin"
JAVA_HOME="/usr/lib/jvm/java-8-openjdk-amd64"

:w
```

Esc 누른 뒤 명령어 실행

:w  
= 저장하기

:p  
= 끝내기(저장 안할 시 알림창)

:p!  
= 강제종료(저장 안해도 꺼짐)

```
ubuntu@ip-172-31-43-148: /usr/lib/jvm/java-8-openjdk-amd64$ source /etc/environment
ubuntu@ip-172-31-43-148: /usr/lib/jvm/java-8-openjdk-amd64$ javac
Usage: javac <options> <source files>
where possible options include:
  -g                  Generate all debugging info
  -g:none             Generate no debugging info
  -g:{lines,vars,source} Generate only some debugging info
  -nowarn             Generate no warnings
  -verbose            Output messages about what the compiler is doing
  -deprecation        Output source locations where deprecated APIs are used
  -classpath <path>  Specify where to find user class files and annotation processors
  -cp <path>         Specify where to find user class files and annotation processors
  -sourcepath <path> Specify where to find input source files
  -bootclasspath <path> Override location of bootstrap class files
  -extdirs <dirs>    Override location of installed extensions
  -endorseddirs <dirs> Override location of endorsed standards path
  -proc:{none,only}  Control whether annotation processing and/or compilation is done.
  -processor <class1>[,<class2>,<class3>...] Names of the annotation processors to run; bypasses
default discovery process
  -processorpath <path> Specify where to find annotation processors
  -parameters        Generate metadata for reflection on method parameters
```

Source  
: 변수설정  
적용

Javac  
: 설정확인