AWS Ubuntu

명령어로 openjdk설치

인스턴스 생성/연결 Ubuntu2018버전

인스턴스 생성 - 우분투 18.04 LTS 버전

6

SUSE Linux Enterprise Server 12 SP5 (HVM), SSD Volume Type - ami-04aa88aebb9fefd83

선택

SUSE Linux

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.

64비트(x86)

프리 티어 사용 가능

루트 디바이스 유형: ebs 가상화 유형: hvm ENA 활성화: 예

(

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0b9064170e32bde34 (64비트 x86) / ami-026141f3d5c6d2d0c (64비트 Arm)

선택

Ubuntu Server 18.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services). 프리 티어 사용 가능

● 64비트(x86)

루트 디바이스 유형: ebs 가상화 유형: hvm ENA 활성화: 예

○ 64비트(Arm)

Ŷ

Amazon Linux 2 with .Net Core, PowerShell, Mono, and MATE Desktop Environment - ami-026fff5d827271b86

선택

Amazon Linux

.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).

64비트(x86)

프리 티어 사용 가능

루트 디바이스 유형: ebs 가상화 유형: hvm ENA 활성화: 예

AMI 편집

보안 그룹 편집

0

Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-0d563aeddd4be7fff (64비트 x86) / ami-0bff25b43a4479334 (64비트 Arm)

선택

Ubuntu Server 16.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

인스턴스 스토리지

■ CAHI = (voc)

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, 메 모리, 스토리지 및 네트워킹 용량의 다양한 조합이 있으며, 애플리케이션에 사용할 적절한 리소스 조합을 유연하게 선택할 수 있습니다. 인스턴스 유형과 이 인스턴스 유형이 컴 퓨팅 요건을 충족하는 방식에 대해 자세히 알아보기

필터링 기준: 모든 인스턴스 패밀리 🔻

현재 세대 🔻 열 표시/숨기기

현재 선택된 항목: t2.micro (- ECU, 1 vCPUs, 2.5 GHz, -, 1 GiB 메모리, EBS 전용)

그룹	유형	vCPUs (i) +	메모리 (GiB) 🔻	(GB) (i)	가능 (i)	네트워크 성능 ϳ 🔻	원 () *
t2	t2.nano	1	0.5	EBS 전용	-	낮음에서 중간	예
t2	t2.micro 프리 티어 사용 가능	1	1	EBS 전용	-	낮음에서 중간	예
t2	t2.small	1	2	EBS 전용	-	낮음에서 중간	예
t2	t2.medium	2	4	EBS 전용	-	낮음에서 중간	예
t2	t2.large	2	8	EBS 전용	-	낮음에서 중간	예
t2	t2.xlarge	4	16	EBS 전용	-	보통	예
t2	t2.2xlarge	8	32	EBS 전용	-	보통	예
t3	t3.nano	2	0.5	EBS 전용	예	최대 5기가비트	예
t3	t3.micro	2	1	EBS 전용	예	최대 5기가비트	예
t3	t3.small	2	2	EBS 전용	예	최대 5기가비트	예
t3	t3.medium	2	4	EBS 전용	ф	최대 5기가비트	예
t3	t3.large	2	8	EBS 전용	ф	최대 5기가비트	예

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 [**시** 작]을 클릭합니다.

인스턴스 보안을 개선하십시오. 보안 그룹 launch-wizard-7이(가) 세계에 개방되어 있습니다.

인스틴스를 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 주가 포트를 열 수도 있습니다. 예를 들어, 웹 서버용으로 HTTP(80)를 엽니다. 보 안 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 유현·ebs 가상화 유현·hym

▼ 인스턴스 유형

인스턴스 유형 편집

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

▼ 보안 그룹

launch-wizard-7

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

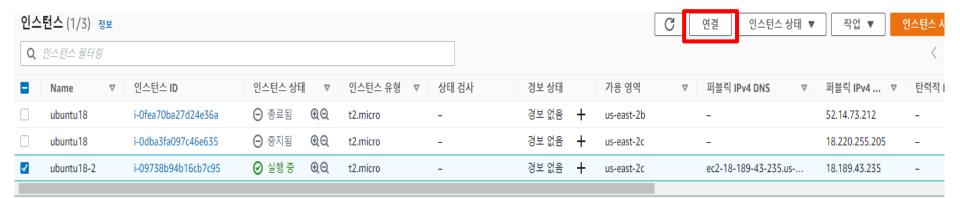
유형 (i) 프로토콜 (i) 포트 범위 () 소스 (i) 설명 (i) 0.0.0.0/0

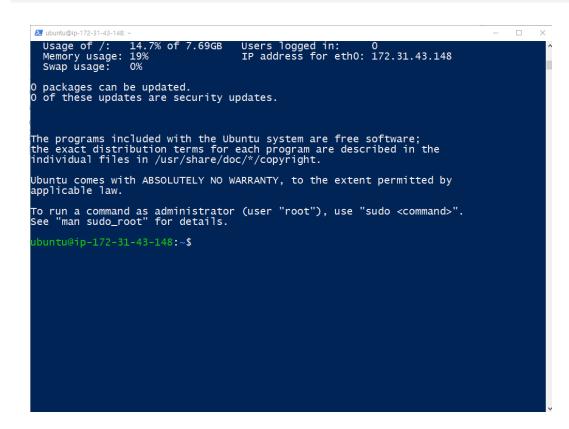
▶ 인스턴스 세부 정보

인스턴스 세부 정보 편집

취소 이전

다음: 인스턴스 세부 정보 구성





파워쉘 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 [
kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease
.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 Pack
[8570 kB]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic/universe Translation
```

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<mark>:</mark>/usr/lib/jvm/java-8-openjdk-amd64\$ sudo vi /etc/environment

- 1. Jdk파일위치로 이동
- 2. 환경변수 수정

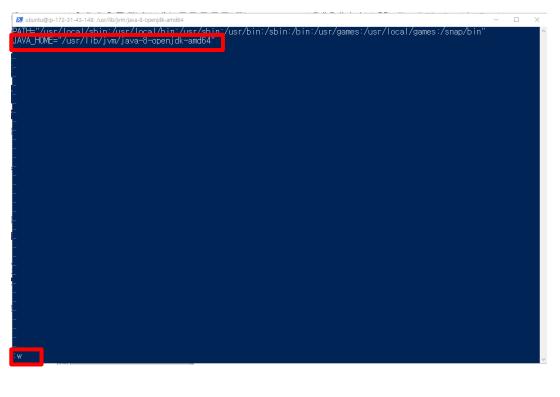


Shift + a

커서를 맨뒤로 옮긴 후 insert모드로 변경

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

변수 추가



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-amd649 javac
Usage: javac <options> <source files>
where possible options include:
                             Generate all debugging info
  -a
                             Generate no debugging info
  -q:none
                             Generate only some debugging info
  -g:{lines,vars,source}
                             Generate no warnings
  -nowarn
                             Output messages about what the compiler is doing
  -verbose
  -deprecation
                             Output source locations where deprecated APIs are used
  -classpath <path>
                             Specify where to find user class files and annotation processors
                             Specify where to find user class files and annotation processors
  -cp <path>
  -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
                             Specify where to find annotation processors
  -processorpath <path>
                             Generate metadata for reflection on method parameters
  -parameters
```

Source :변수설정 적용

Javac

: 설정확인