

## Lab. Using AWS EBS on Ubuntu 22.04 LTS

### 목적

Amazon EC2에서 쉽고 고성능의 블록 스토리지 서비스인 EBS를 사용하는 방법에 대해 알아보자. 이 학습은 AWS Free-Tier를 활용하여 진행한다.

### 사전 준비물

AWS Free-Tier 계정

# Ubuntu Linux 서버 EC2 인스턴스 생성하기

1. EC2 생성을 위해 좌측 상단의 [서비스] > [컴퓨팅] > [EC2]를 클릭하여 해당 페이지로 이동한다.
2. 왼쪽 항목에서 [인스턴스]를 선택하여 해당 페이지로 이동한다.
3. 우측 상단의 [인스턴스 시작] 오렌지 색 버튼을 클릭한다.
4. 다음과 같이 **EC2 Instance**를 생성한다.

- ① [이름] : lab-ubuntu-ec2
- ② [AMI] : Ubuntu Server 22.04 LTS (HVM), SSD Volume Type, 64비트(x86)
- ③ [인스턴스 유형] : t2.micro
- ④ [키 페어 이름] > [새 키 페어 생성] : lab-ubuntu-ec2-key
- ⑤ [네트워크 설정] > [편집]
  - i. [VPC] : Default-VPC
  - ii. [서브넷] : Public Subnet, ap-northeast-2a
  - iii. [퍼블릭 IP 자동 할당] : 활성화
  - iv. [방화벽(Security groups)] > [보안 그룹 생성] : lab-sg
  - v. [인바운드 보안 그룹 규칙] : ssh, 22

5. 특별히 [스토리지 구성]은 [어드밴스드] 링크를 클릭하여 다음과 같이 설정한다.

## ① [볼륨 1 (AMI 루트)(사용자 지정)]

- i. [스토리지 유형] : EBS
- ii. [디바이스 이름] : /dev/sda1
- iii. [크기] : 8GiB
- iv. [볼륨 유형] : 마그네틱(표준)
- v. [종료 시 삭제] : 예
- vi. [암호화됨] : 암호화되지 않음
- vii. [새 볼륨 추가] 버튼 클릭

## ② [볼륨 2 (사용자 지정)]

- i. [스토리지 유형] : EBS
- ii. [디바이스 이름] : /dev/sdb
- iii. [크기] : 2GiB

- iv. [볼륨 유형] : gp2
- v. [종료 시 삭제] : 예
- vi. [암호화됨] : 암호화되지 않음

▼ 스토리지(볼륨) 정보
단순

---

EBS 볼륨
세부 정보 숨기기

---

▼ 볼륨 1 (AMI 루트)(사용자 지정)

<div style="margin-bottom: 10px;"> <span>스토리지 유형 정보</span>  EBS </div> <div style="margin-bottom: 10px;"> <span>크기(GiB) 정보</span>  <input type="text" value="8"/> </div> <div> <span>종료 시 삭제 정보</span>  <input type="text" value="예"/> </div>	<div style="margin-bottom: 10px;"> <span>디바이스 이름 - required 정보</span>  /dev/sda1 </div> <div style="margin-bottom: 10px;"> <span>볼륨 유형 정보</span>  <input type="text" value="standard"/> </div> <div> <span>암호화됨 정보</span>  <input type="text" value="암호화되지 않음"/> </div>	<div style="margin-bottom: 10px;"> <span>스냅샷 정보</span>  snap-00644c3e5be79236b </div> <div> <span>IOPS 정보</span>  해당 사항 없음 </div>
---	---	---

KMS 키 정보  

선택
▼

이 볼륨에서 암호화가 설정된 경우에만 KMS 키를 적용할 수 있습니다.

▼ 볼륨 2 (사용자 지정)
제거

<div style="margin-bottom: 10px;"> <span>스토리지 유형 정보</span>  EBS </div> <div style="margin-bottom: 10px;"> <span>크기(GiB) 정보</span>  <input type="text" value="2"/> </div> <div> <span>종료 시 삭제 정보</span>  <input type="text" value="예"/> </div>	<div style="margin-bottom: 10px;"> <span>디바이스 이름 - required 정보</span>  <input type="text" value="/dev/sdb"/> </div> <div style="margin-bottom: 10px;"> <span>볼륨 유형 정보</span>  <input type="text" value="gp2"/> </div> <div> <span>암호화됨 정보</span>  <input type="text" value="암호화되지 않음"/> </div>	<div style="margin-bottom: 10px;"> <span>스냅샷 정보</span>  <div style="border: 1px solid #ccc; padding: 2px; display: flex; align-items: center;"> <span style="flex-grow: 1;">선택</span> <span>▼</span> </div> </div> <div> <span>IOPS 정보</span>  100 / 3000 </div>
---	--	--

KMS 키 정보  

선택
▼

이 볼륨에서 암호화가 설정된 경우에만 KMS 키를 적용할 수 있습니다.

6. [요약]을 한번 더 확인하고 [인스턴스 시작]을 클릭한다.

▼ 요약

인스턴스 개수 정보

1

소프트웨어 이미지(AMI)

Canonical, Ubuntu, 22.04 LTS, ...더 보기

ami-086cae3329a3f7d75

가상 서버 유형(인스턴스 유형)

t2.micro

방화벽(보안 그룹)

새 보안 그룹

스토리지(볼륨)

2개의 볼륨 - 10GiB

취소

인스턴스 시작

명령 검토

7. 방금 생성한 인스턴스를 클릭하여 인스턴스 요약 페이지로 들어간다. 항목 중 **[스토리지]** 탭을 클릭한다. 아래 그림과 같이 두개의 스토리지가 연결되어 있음을 확인할 수 있다.

인스턴스: i-0fe0d7dd624947c8c(lab-ubuntu-ec2)

세부 정보 | 보안 | 네트워킹 | **스토리지** | 상태 검사 | 모니터링 | 태그

▼ 루트 디바이스 세부 정보

루트 디바이스 이름

/dev/sda1

루트 디바이스 유형

EBS

EBS 최적화

비활성

▼ 블록 디바이스

Q 블록 디바이스 필터링

볼륨 ID	디바이스 이름	볼륨 크기(GiB)	연결 상태	연결 시간	암호화됨	KMS 키 ID	종료 시 삭제
<a href="#">vol-05b350b7e8f5a536a</a>	/dev/sda1	8	연결됨	2023/10/29 08:14 GMT+9	아니요	-	예
<a href="#">vol-0a562e2dab57b5d73</a>	/dev/sdb	2	연결됨	2023/10/29 08:14 GMT+9	아니요	-	예

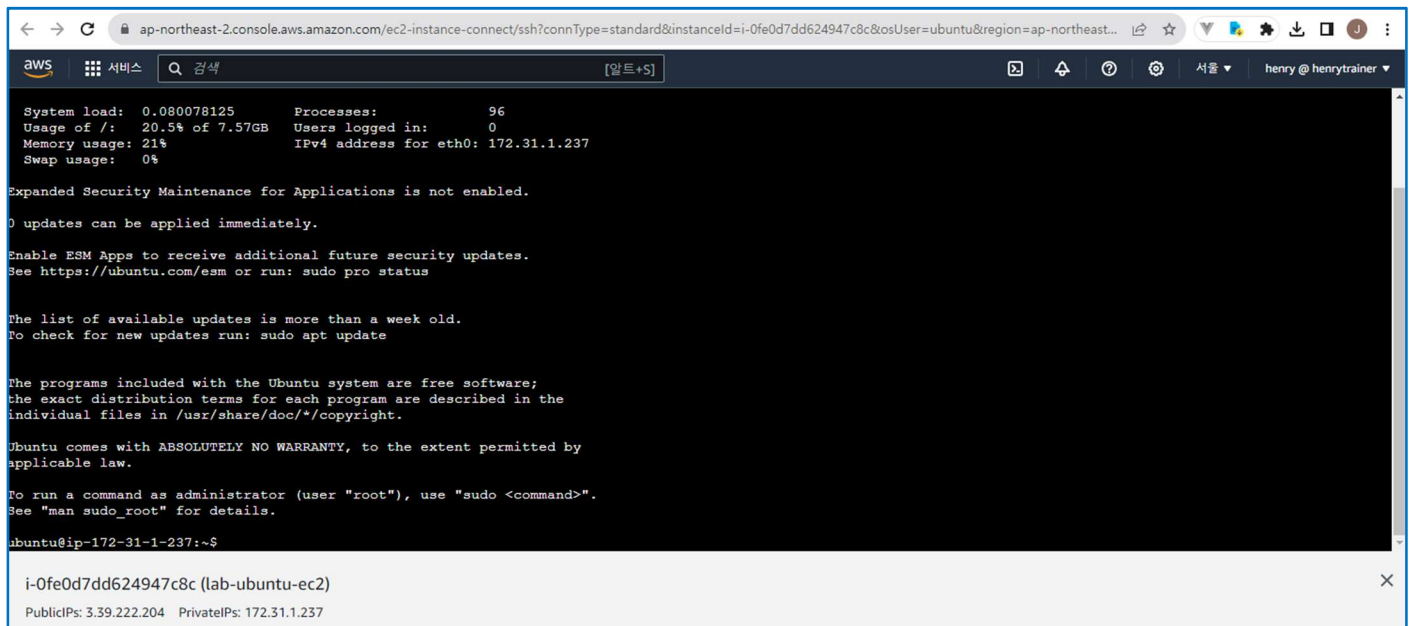
8. 이번에는 좌측 메뉴 중 **[Elastic Block Store] > [볼륨]**을 선택한다. 앞에서 인스턴스 생성시 함께 생성했던 스토리지를 확인할 수 있다.

볼륨 (2) 정보

Q 검색

<input type="checkbox"/>	Name ▼	볼륨 ID ▼	유형 ▼	크기 ▼	IOPS ▼	처리량 ▼	스냅샷 ▼	생성 완료 ▼	가용 영역 ▼	볼륨 상태
<input type="checkbox"/>	-	<a href="#">vol-0a562e2dab57b5d73</a>	gp2	2 GiB	100	-	-	2023/10/29 08:14 GMT+9	ap-northeast-2a	사용 중
<input type="checkbox"/>	-	<a href="#">vol-05b350b7e8f5a536a</a>	standard	8 GiB	-	-	snap-00644c3...	2023/10/29 08:14 GMT+9	ap-northeast-2a	사용 중

9. 방금 생성한 인스턴스를 연결해보자. 인스턴스 요약 페이지에서 **[연결]**을 클릭한다.
10. **[인스턴스에 연결]** 페이지에서 **[EC2 인스턴스 연결]** 탭의 **[연결]** 오렌지색 버튼을 클릭한다.
11. 방금 생성한 인스턴스에 잘 연결이 된 것을 확인할 수 있다.



The screenshot shows a web browser window with the AWS Management Console URL. The terminal window displays the following content:

```
System load:  0.080078125    Processes:           96
Usage of /:   20.5% of 7.57GB Users logged in:       0
Memory usage: 21%          IPv4 address for eth0: 172.31.1.237
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

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-172-31-1-237:~$
```

Below the terminal window, the instance details are shown:

```
i-0fe0d7dd624947c8c (lab-ubuntu-ec2)
PublicIPs: 3.39.222.204 PrivateIPs: 172.31.1.237
```

## AWS EC2의 EBS 사용하기

1. 인스턴스와 연결이 성공하면 `df -h` 명령을 사용하여 포맷되고 마운트된 볼륨을 확인한다.

```
ubuntu@ip-172-31-1-237:~$  
ubuntu@ip-172-31-1-237:~$ df -h  
Filesystem      Size  Used Avail Use% Mounted on  
/dev/root        7.6G  1.6G  6.0G   21% /  
tmpfs            475M    0  475M    0% /dev/shm  
tmpfs            190M  836K  190M    1% /run  
tmpfs            5.0M    0   5.0M    0% /run/lock  
/dev/xvda15      105M   6.1M   99M    6% /boot/efi  
tmpfs            95M   4.0K   95M    1% /run/user/1000  
ubuntu@ip-172-31-1-237:~$
```

2. 또한 `lsblk` 명령어를 사용하여 기본적으로 모든 연결된 드라이브를 확인한다.

```
ubuntu@ip-172-31-1-237:~$ lsblk  
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS  
loop0        7:0    0   24.6M  1 loop /snap/amazon-ssm-agent/7528  
loop1        7:1    0   55.7M  1 loop /snap/core18/2790  
loop2        7:2    0   63.5M  1 loop /snap/core20/2015  
loop3        7:3    0  111.9M  1 loop /snap/lxd/24322  
loop4        7:4    0   40.8M  1 loop /snap/snapd/20092  
xvda        202:0    0    8G   0 disk  
├─xvda1     202:1    0   7.9G   0 part /  
├─xvda14    202:14   0    4M   0 part  
└─xvda15    202:15   0  106M   0 part /boot/efi  
xvdb        202:16   0    2G   0 disk  
ubuntu@ip-172-31-1-237:~$
```

3. 위 그림에 보면 `xvdb` 디바이스는 Raw Block Device이다. 따라서 마운트하고 사용하기 전에 파일 시스템을 생성해야 한다. 다음 그림과 같이 단순히 디바이스에 대해 `data`라고 표시되면, 해당 디바이스에 어떤 파일 시스템도 없다는 것을 확인하는 것이다. 그래서 파일 시스템을 생성해야 한다.

\$ `sudo file -s /dev/xvdb`

```
ubuntu@ip-172-31-1-237:~$ sudo file -s /dev/xvdb  
/dev/xvdb: data  
ubuntu@ip-172-31-1-237:~$
```

4. 다음의 명령어를 사용하여 기존의 파일시스템의 정보를 확인해보자. 현재 **/dev/xvda1**은 파일시스템이 **ext4** 파일시스템임을 알 수 있다.

```
$ sudo file -s /dev/xvda1
```

```
ubuntu@ip-172-31-1-237:~$ sudo file -s /dev/xvda1
/dev/xvda1: Linux rev 1.0 ext4 filesystem data, UUID=4f575094-453e-450d-aeed-215d8cbcbf58, volume name "cloudimg-rootfs" (needs journal recovery) (extents) (64bit) (large files) (huge files)
ubuntu@ip-172-31-1-237:~$
```

5. 다음의 명령어를 사용하여 **/dev/xvdb**의 파일시스템을 **ext4**로 설정한다.

```
$ sudo mkfs -t ext4 /dev/xvdb
```

```
ubuntu@ip-172-31-1-237:~$ sudo mkfs -t ext4 /dev/xvdb
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 524288 4k blocks and 131072 inodes
Filesystem UUID: 6d66873f-ba2e-4956-9d4f-0bb588f72f34
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

6. 이제 **data** 디렉토리에 디렉토리를 마운트한다. 먼저 루트 디렉토리 밑에 **data** 디렉토리를 생성한다. 그리고 이 **data** 디렉토리를 **/dev/xvdb**와 마운트한다.

```
$ sudo mkdir /data
```

```
$ lsblk
```

```
$ sudo mount /dev/xvdb /data
```

```
ubuntu@ip-172-31-1-237:~$ sudo mkdir /data
ubuntu@ip-172-31-1-237:~$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0        7:0    0   24.6M  1 loop /snap/amazon-ssm-agent/7528
loop1        7:1    0   55.7M  1 loop /snap/core18/2790
loop2        7:2    0   63.5M  1 loop /snap/core20/2015
loop3        7:3    0  111.9M  1 loop /snap/lxd/24322
loop4        7:4    0   40.8M  1 loop /snap/snapd/20092
xvda        202:0    0    8G   0 disk
├─xvda1     202:1    0   7.9G  0 part /
├─xvda14    202:14   0    4M   0 part
└─xvda15    202:15   0  106M  0 part /boot/efi
xvdb        202:16   0    2G   0 disk
ubuntu@ip-172-31-1-237:~$ sudo mount /dev/xvdb /data
ubuntu@ip-172-31-1-237:~$
```



7. 마운트 후 **lsblk**를 실행하면 성공적으로 마운트되었음을 확인할 수 있다.

```
ubuntu@ip-172-31-1-237:~$ lsblk
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
loop0        7:0      0   24.6M  1 loop /snap/amazon-ssm-agent/7528
loop1        7:1      0   55.7M  1 loop /snap/core18/2790
loop2        7:2      0   63.5M  1 loop /snap/core20/2015
loop3        7:3      0  111.9M  1 loop /snap/lxd/24322
loop4        7:4      0   40.8M  1 loop /snap/snapd/20092
xvda         202:0     0     8G   0 disk
├─xvda1      202:1     0    7.9G  0 part /
├─xvda14     202:14    0     4M   0 part
└─xvda15     202:15    0   106M  0 part /boot/efi
xvdb         202:16    0     2G   0 disk /data
ubuntu@ip-172-31-1-237:~$
```

8. **data** 디렉토리로 이동하여 **Hello.txt** 파일을 생성한다.

```
$ cd /data
```

```
$ sudo touch Hello.txt
```

```
ubuntu@ip-172-31-1-237:~$ cd /data
ubuntu@ip-172-31-1-237:/data$ sudo touch Hello.txt
ubuntu@ip-172-31-1-237:/data$ ls
Hello.txt  lost+found
ubuntu@ip-172-31-1-237:/data$
```

9. 에디터를 사용하여 **Hello.txt**에 다음과 같이 **Hello, Amazon EBS!!!**라고 텍스트를 입력한다.

```
$ sudo nano Hello.txt
```

```
$ cat Hello.txt
```

```
ubuntu@ip-172-31-1-237:/data$ sudo nano Hello.txt
ubuntu@ip-172-31-1-237:/data$ cat Hello.txt
Hello, Amazon EBS!!!
ubuntu@ip-172-31-1-237:/data$
```



10. 매번 새로 부팅할 때마다 EBS 볼륨에 마운트하기 위해서는 해당 디바이스를 **/etc/fstab** 파일에 등록해야 한다. 먼저 원본 파일을 **fstab.bak**로 백업한다.

```
$ ls /etc/fstab
```

```
$ sudo cp /etc/fstab /etc/fstab.bak
```

```
$ sudo nano /etc/fstab
```

```
ubuntu@ip-172-31-1-237:/data$ ls /etc/fstab
/etc/fstab
ubuntu@ip-172-31-1-237:/data$ sudo cp /etc/fstab /etc/fstab.bak
ubuntu@ip-172-31-1-237:/data$ sudo nano /etc/fstab
```

11. 그리고 **/etc/fstab** 파일을 열어서 편집한다. 다음과 같이 새 엔트리를 추가한다. 순서대로 앞에서부터, 디바이스 이름, 마운트 포인트, 파일시스템, 마운트 옵션을 추가한다. 입력이 모두 마치면 파일을 저장한다.

```
/dev/xvdb /data ext4 defaults,nofail 0 2
```

```
LABEL=cloudimg-rootfs / ext4 discard,errors=remount-ro 0 1
LABEL=UEFI /boot/efi vfat umask=0077 0 1
/dev/xvdb /data ext4 defaults,nofail 0 2
```

12. 이제 다음의 명령으로 방금 생성한 볼륨의 정보를 확인할 수 있다.

```
$ sudo file -s /dev/xvdb
```

```
ubuntu@ip-172-31-1-237:/data$ sudo file -s /dev/xvdb
/dev/xvdb: Linux rev 1.0 ext4 filesystem data, UUID=6d66873f-ba2e-4956-9d4f-0bb588f72f34 (needs journal recovery) (extents) (64bit) (large files) (huge files)
ubuntu@ip-172-31-1-237:/data$
```

13. 이제 마운트 해제한다. 마운트해제 후 **lsblk**를 실행하면 **/dev/xvdb**가 마운트되어 있지 않음을 확인할 수 있다.

```
$ cd ..
```

```
$ sudo umount /data
```

```
$ lsblk
```

```

ubuntu@ip-172-31-1-237:/data$
ubuntu@ip-172-31-1-237:/data$ cd ..
ubuntu@ip-172-31-1-237:/$ sudo umount /data
ubuntu@ip-172-31-1-237:/$ lsblk
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
loop0         7:0    0  24.6M  1 loop /snap/amazon-ssm-agent/7528
loop1         7:1    0  55.7M  1 loop /snap/core18/2790
loop2         7:2    0  63.5M  1 loop /snap/core20/2015
loop3         7:3    0 111.9M  1 loop /snap/lxd/24322
loop4         7:4    0  40.8M  1 loop /snap/snapd/20092
xvda         202:0    0    8G    0 disk
├─xvda1       202:1    0   7.9G  0 part /
├─xvda14      202:14   0    4M    0 part
└─xvda15      202:15   0   106M  0 part /boot/efi
xvdb         202:16   0    2G    0 disk
ubuntu@ip-172-31-1-237:/$

```

14. 다음의 명령으로 작동하는 /etc/fstab 파일을 테스트해 보자. mount -a 옵션은 /etc/fstab에 지정한 모든 파일 시스템을 마운트하는 명령이다.

```
$ sudo mount -a
```

```
$ lsblk
```

```

ubuntu@ip-172-31-1-237:/$
ubuntu@ip-172-31-1-237:/$ sudo mount -a
ubuntu@ip-172-31-1-237:/$ lsblk
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
loop0         7:0    0  24.6M  1 loop /snap/amazon-ssm-agent/7528
loop1         7:1    0  55.7M  1 loop /snap/core18/2790
loop2         7:2    0  63.5M  1 loop /snap/core20/2015
loop3         7:3    0 111.9M  1 loop /snap/lxd/24322
loop4         7:4    0  40.8M  1 loop /snap/snapd/20092
xvda         202:0    0    8G    0 disk
├─xvda1       202:1    0   7.9G  0 part /
├─xvda14      202:14   0    4M    0 part
└─xvda15      202:15   0   106M  0 part /boot/efi
xvdb         202:16   0    2G    0 disk /data
ubuntu@ip-172-31-1-237:/$

```