AWS Storage Services 1:

EBS, EFS

Hyunchan, Park

http://oslab.jbnu.ac.kr

Division of Computer Science and Engineering

Jeonbuk National University

학습 내용

- Storage system 의 구조 및 각 계층의 역할

- EBS, EFS 소개 및 실습

(시작 전에!)

- 항상 불필요한 Instance 가 있다면 Terminate!
  - 앞서 과제를 다 수행했다면, Windows instance 는 “종료” 하자.

개인 과제 #4

- 제출: 하나의 ppt에 각 캡처 파일을 넣은 후, PDF 파일로 변환해 LMS “과제 4” 제출
  - 파일 이름: 학번.pdf
  - Page #1: 제목, 학번, 이름
  - Page #2~#3: 아래 캡처 화면 하나씩
- EBS, EFS 실습 화면 캡처 (리눅스에서만 수행)
  - EBS: 서로 다른 인스턴스에서 볼륨을 mount 하고 같은 파일을 확인하는 화면
    - 각 인스턴스에서 mount 수행 후, 캡처 (#2)
  - EFS: 서로 다른 인스턴스에서 파일 시스템을 mount 하고 같은 파일을 확인하는 화면
    - 각 인스턴스에서 mount 수행 후, 캡처 (#3)
- 기한: 10/19 (월) 23:59
  - 지각 감점: 5%p / 12H
  - 1주 이후 제출 차단

AWS Storage Services

요구 사항:	추천 제품:
Amazon EC2, 관계형 및 NoSQL 데이터베이스, 데이터 웨어하우징, 엔터프라이즈 애플리케이션, 빅 데이터 처리 또는 백업 및 복구를 위한 영구 로컬 스토리지	Amazon Elastic Block Store (Amazon EBS)
Linux 기반의 워크로드를 AWS 클라우드 서비스와 온프레미스 리소스에서 사용할 수 있도록 지원하는 간단하고 확장 가능하며 탄력적인 파일 시스템입니다. 이 제품은 애플리케이션을 중단하지 않고 온디맨드 방식으로 페타바이트 규모까지 확장하도록 구축되어 파일을 추가하고 제거할 때 자동으로 확장되고 축소되므로, 애플리케이션은 필요할 때 필요한 만큼 스토리지를 사용할 수 있습니다.	Amazon Elastic File System (Amazon EFS)
사용자 생성 콘텐츠, 활성 아카이브, 서버리스 컴퓨팅, 빅 데이터 스토리지 또는 백업 및 복구를 위해 인터넷 위치에서 데이터에 액세스할 수 있도록 지원하는 확장 가능하고 안정적인 플랫폼입니다.	Amazon Simple Storage Service (Amazon S3)
아카이브 및 규제 준수를 위해 테이프를 대체할 수 있는 매우 저렴한 장기 스토리지 클래스	Amazon Glacier 및 Amazon S3 Glacier Deep Archive

https://aws.amazon.com/ko/products/storage/

AWS Storage Services

요구 사항:	추천 제품:
고성능 컴퓨팅, 기계 학습 및 미디어 데이터 처리 워크플로우와 같이 컴퓨팅 집약적 워크로드에 최적화된 완전관리형 파일 시스템으로, Amazon S3에 완벽하게 통합되어 있습니다.	Amazon FSx for Lustre
Windows Server를 기반으로 구축된 완전관리형 네이티브 Microsoft Windows 파일 시스템으로, 이 제품을 사용하면 SMB 프로토콜 및 Windows NTFS, AD(Active Directory) 통합, DFS(분산 파일 시스템)에 대한 전체 지원을 비롯하여, 파일 스토리지가 필요한 Windows 기반 애플리케이션을 AWS로 쉽게 이전할 수 있습니다.	Amazon FSx for Windows File Server
버스팅, 계층화 또는 마이그레이션을 위해 Amazon 클라우드 스토리지로 온프레미스 환경을 보강해주는 하이브리드 스토리지 클라우드입니다.	AWS Storage Gateway
유형 및 크기에 상관없이 모든 데이터를 AWS 클라우드로, 혹은 AWS 클라우드에서 이동하는 작업을 간소화 및 가속화할 수 있도록 지원하는 서비스 포트폴리오입니다.	클라우드 데이터 마이그레이션 서비스
클라우드뿐 아니라 온프레미스에서도 AWS Storage Gateway를 사용해 AWS 서비스 전체에서 데이터 백업을 손쉽게 중앙화하고 자동화할 수 있는 완전관리형 백업 서비스입니다.	AWS Backup

https://aws.amazon.com/ko/products/storage/

AWS 스토리지 선택 옵션

Amazon S3

모든 타입에 대한 내구성 높은 개체 스토리지 서비스

경제적 활용

사용한 만큼만 지불하고, 미리 선타자가 필요 없으며 용량 계산 필요 없음

Amazon Glacier

자주 접근하지 않는 데이터에 대한 백업 서비스

백업 편이성

손쉽게 혼자 관리 가능

데이터 생명 주기에 따라 관리 가능

Amazon EBS

Amazon EC2에 사용할 수 있는 블록 스토리지

활용 용이성

내구성 및 보안성이 높으며, 가상 서버에서 데이터 처리 가능

Amazon EFS

Amazon EC2에 대한 네트워크 스토리지

손쉬운 확장성

블록 스토어 관리 비용 절감 및 공유 스토리지 관리 불필요

(현재 Preview로 제공)

AWS Storage Services

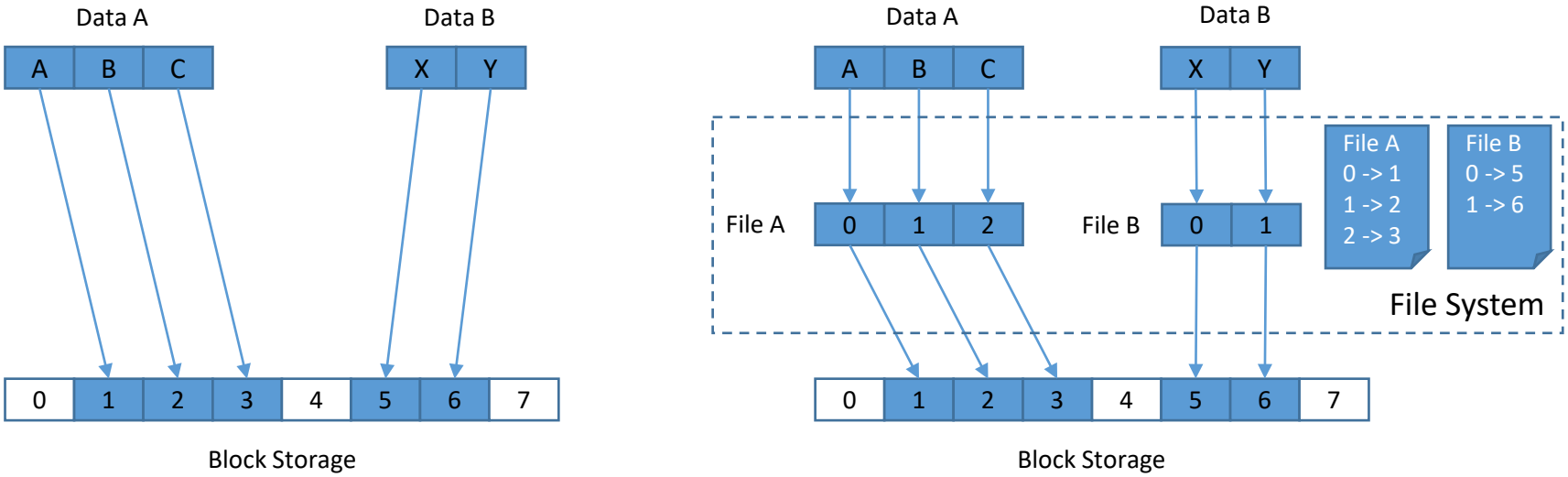
- EBS (Elastic Block Storage)
  - Amazon EC2 인스턴스에 연결된 영구 볼륨에 블록 데이터를 저장하고 이를 처리
  - 일반 HDD, SSD와 같은 블록 기반의 입출력을 지원하는 스토리지
  - EC2 VM의 OS, 데이터를 저장하기 위한 저장장치로 사용됨
  - 연간 고장률 AFR: 0.1%~0.2% (일반 HDD: 4%)

- EFS (Elastic File System)
  - 간편하고 확장 가능한 파일 시스템에 데이터를 저장 및 공유
  - Ext4, NTFS와 같이 파일 시스템 access semantics 에 따른 FS 인터페이스를 제공하는 서비스
  - 자체 고가용성, 내구성을 제공하도록 설계
  - 용량 자동 확장 및 축소

Block storage and File system

- Block storage
  - 512B 단위로 분할된, (sector)
  - 0부터 시작하는 선형 주소 공간 (linear address space)에,
  - 데이터를 읽거나 쓸 수 있는 스토리지 장치
- File system
  - 하나의 파일 시스템 내에서 각기 unique name 을 갖는,
  - File 이라는 독립적인 개체에,
  - 1B 단위로 분할된,
  - 0부터 시작하는 선형 주소 공간 (linear address space)에,
  - 데이터를 읽거나 쓸 수 있는 스토리지 시스템
  - 일반적으로 블록 스토리지 장치의 상단에 위치함
    - 실제 데이터는 블록 스토리지에 저장됨

Block storage and File system



- Block storage (left)
  - 데이터를 기록하기 위해 블록 스토리지 상의 주소를 직접 사용해야함
- File system (right)
  - 파일 시스템이 제공해주는 파일 이라는 추상적 개체를 이용
  - 모든 데이터는 0부터 시작하는 독립된 주소 공간에 기록할 수 있음
  - 실제 데이터가 기록되는 공간은 블록 스토리지이며,
  - 파일의 주소 공간과 블록 스토리지 주소 공간 사이의 mapping 을 파일 시스템이 관리
    - 이때 이 매핑 정보 또한 블록 스토리지에 저장되어야 함

Block storage and File system 장단점

- Block storage 장단점
  - (장점) 중간 계층을 거치지 않으므로 액세스가 빠르다.
  - (단점) 같은 주소 공간을 다수가 공유하므로, 데이터 관리가 어려움
    - 예) A와 B가 동시에 서로 다른 데이터를 0번 주소에 기록하려하면?
- File system 장단점
  - (장점) 파일마다 서로 주소 공간이 분리되어 있기 때문에,
    서로 분리되어야 할 데이터를 쉽게 관리할 수 있음
    - 예) 위의 예에서, A와 B는 서로 다른 파일의 주소 공간에 각각 접근하므로 문제가 없음
  - (단점) 성능 저하
    - 파일의 데이터가 실제 블록 스토리지 어느 주소에 위치하는지 변환이 필요 (mapping)
    - 그러나 mapping 정보와 데이터의 메모리 캐싱을 통해 많은 부분 해소됨

AWS Storage Services

스토리지 및 콘텐츠 전송

Amazon EFS

5GB

스토리지

간편성과 확장성을 갖춘 Amazon EC2 인스턴스용 공유 파일 스토리지 서비스

Amazon EFS에 대해 자세히 알아보기 »

세부 정보 보기

스토리지 및 콘텐츠 전송

Amazon Elastic Block Storage

30GB

범용(SSD) 또는 마그네틱을 원하는 대로 조합

안정적이고 지연 시간이 짧은 EC2 인스턴스용 영구 블록 수준 스토리지 볼륨

Amazon Elastic Block Storage에 대해 자세히 알아보기 »

세부 정보 보기

AWS Storage Services

- S3 (Simple Storage Service)
  - AWS의 가장 기본적인 Object based storage
  - 일반적인 클라우드 스토리지와 동일함 (예. 네이버/다음 클라우드)
  - 내구성: 99.999 999 999% (11-9)

- Glacier (뜻: 빙하)
  - 데이터 보관 및 백업을 위한 안전하고 내구성있는 스토리지
  - 싸고 성능이 느림
  - 연평균 99.999 999 999%의 내구성

AWS Storage Services

(1년 free tier)

스토리지 및 콘텐츠 전송

Amazon S3

5GB

표준 스토리지

보안성, 안정성 및 확장성을 갖춘 객체 스토리지 인프라

Amazon S3에 대해 자세히 알아보기 »

세부 정보 보기

(평생 무료)

스토리지 및 콘텐츠 전송

Amazon Glacier

10GB

스토리지 검색

안전하고 안정적인 장기 객체 스토리지

Amazon Glacier에 대해 자세히 알아보기 »

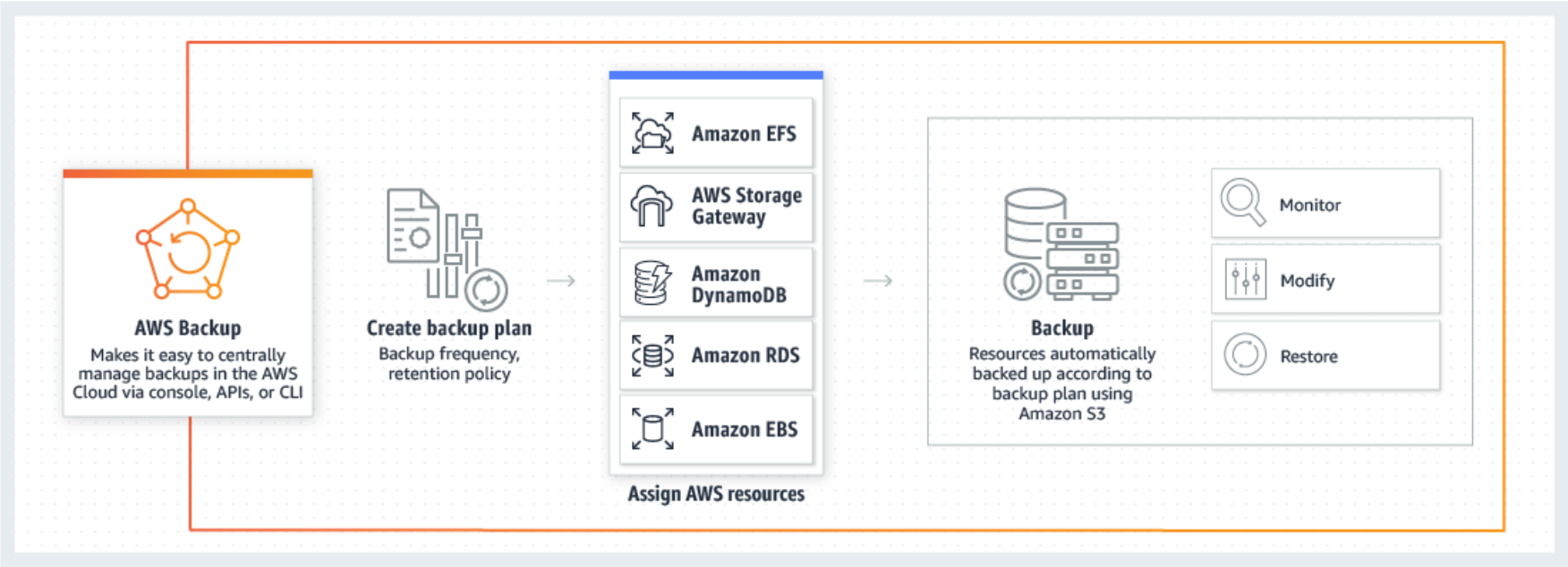
세부 정보 보기

AWS Storage Services

- AWS Storage Gateway
  - 사용자가 기존에 사용하던 스토리지 인프라 및 데이터를 포함한, 모든 스토리지 서비스를 AWS 클라우드와 통합할 수 있도록 지원
  - 파일, 볼륨, 테이프 인터페이스 지원
  - NFS, iSCSI 인터페이스 지원

AWS Backup

- 완전관리형 백업 서비스
- 중앙 관리 및 자동화
  - AWS Storage Gateway를 사용해 AWS 서비스 전체에서 데이터 백업을 손쉽게 중앙집중화하고 자동화
  - AWS Backup 콘솔에서 클릭 몇 번이면 백업 일정과 보존 관리를 자동화하는 백업 정책을 생성할 수 있음



EBS

EBS Pricing

- Amazon EBS 범용 SSD(gp2) 볼륨
  - $0.114 프로비저닝된 스토리지의 월별 GB당
  - 예) 30GB 한달 쓰면? $3.342 = 약 4천원
- Amazon EBS 프로비저닝된 IOPS SSD(io1) 볼륨
  - $0.128 프로비저닝된 스토리지의 월별 GB당
  - $0.067 프로비저닝된 월별 IOPS당
- Amazon EBS 처리량 최적화 HDD(st1) 볼륨
  - $0.051 프로비저닝된 스토리지의 월별 GB당
- Amazon EBS 콜드 HDD(sc1) 볼륨
  - $0.029 프로비저닝된 스토리지의 월별 GB당
- Amazon S3에 대한 Amazon EBS 스냅샷
  - $0.05 저장된 데이터의 월별 GB당

	SSD(Solid State Drive)		하드 디스크 드라이브(HDD)	
볼륨 유형	EBS 프로비저닝된 IOPS SSD(io1)	EBS 범용 SSD(gp2)*	처리량 최적화 HDD(st1)	콜드 HDD(sc1)
간략한 설명	고성능 SSD 볼륨은 지연 시간에 민감한 트랜잭션 워크로드를 위해 설계됨	범용 SSD 볼륨은 다양한 트랜잭션 워크로드를 위해 가격과 성능의 균형을 맞춤	저비용 HDD 볼륨은 자주 액세스하고 처리량 집약적인 워크로드를 위해 설계됨	최저비용 HDD 볼륨은 액세스 빈도가 낮은 워크로드를 위해 설계됨
사용 사례	I/O 집약적 NoSQL 및 관계형 데이터베이스	부트 볼륨, 짧은 지연 시간의 대화형 앱, 개발 및 테스트	빅 데이터, 데이터 웨어하우스, 로그 처리	일별 스캔 횟수가 적은 콜드 데이터
API 이름	io1	gp2	st1	sc1
볼륨 크기	4GB – 16TB	1GB – 16TB	500GB – 16TB	500GB – 16TB
볼륨당 최대 IOPS**	20,000	10,000	500	250
볼륨당 최대 처리량	320MB/초	160MB/초	500MB/초	250MB/초
인스턴스당 최대 IOPS	75,000	75,000	75,000	75,000
인스턴스당 최대 처리량	1,750MB/초	1,750MB/초	1,750MB/초	1,750MB/초
요금	월별 GB당 0.125 USD 프로비저닝된 IOPS당 0.065 USD	월별 GB당 0.10 USD	월별 GB당 0.045 USD	월별 GB당 0.025 USD
주요 성능 특성	IOPS	IOPS	MB/초	MB/초

실습 진행 내용

- 리눅스 인스턴스 2대를 생성 (Bitnami wordpress)
- EBS에서 새로운 볼륨 할당
- 한 인스턴스에 새볼륨 연결
  - 파일 시스템 생성 및 마운트 후, 예제 파일 1개 생성
- 볼륨 분리
- 다른 인스턴스에 볼륨 연결
  - 마운트 후, 예제 파일 내용 확인
- EBS 삭제

EC2 대시보드: 인스턴스 2개 생성

* t2micro, 인스턴스 2대, 서브넷에서 2a 혹은 2c 선택

EC2 대시보드: 생성 확인 및 이름 설정

Name	인스턴스 ID	인스턴스 유형	가용 영역	인스턴스 상태	상태 검사	IPv4 퍼블릭 IP
vm1	i-06744c1d4a8f733c5	t2.micro	ap-northeast-2a	running	2/2개 검사 ...	13.125.198.156
vm2	i-077cb663099cd8039	t2.micro	ap-northeast-2a	running	2/2개 검사 ...	13.209.18.253

인스턴스: i-06744c1d4a8f733c5 퍼블릭 DNS: ec2-13-125-198-156.ap-northeast-2.compute.amazonaws.com

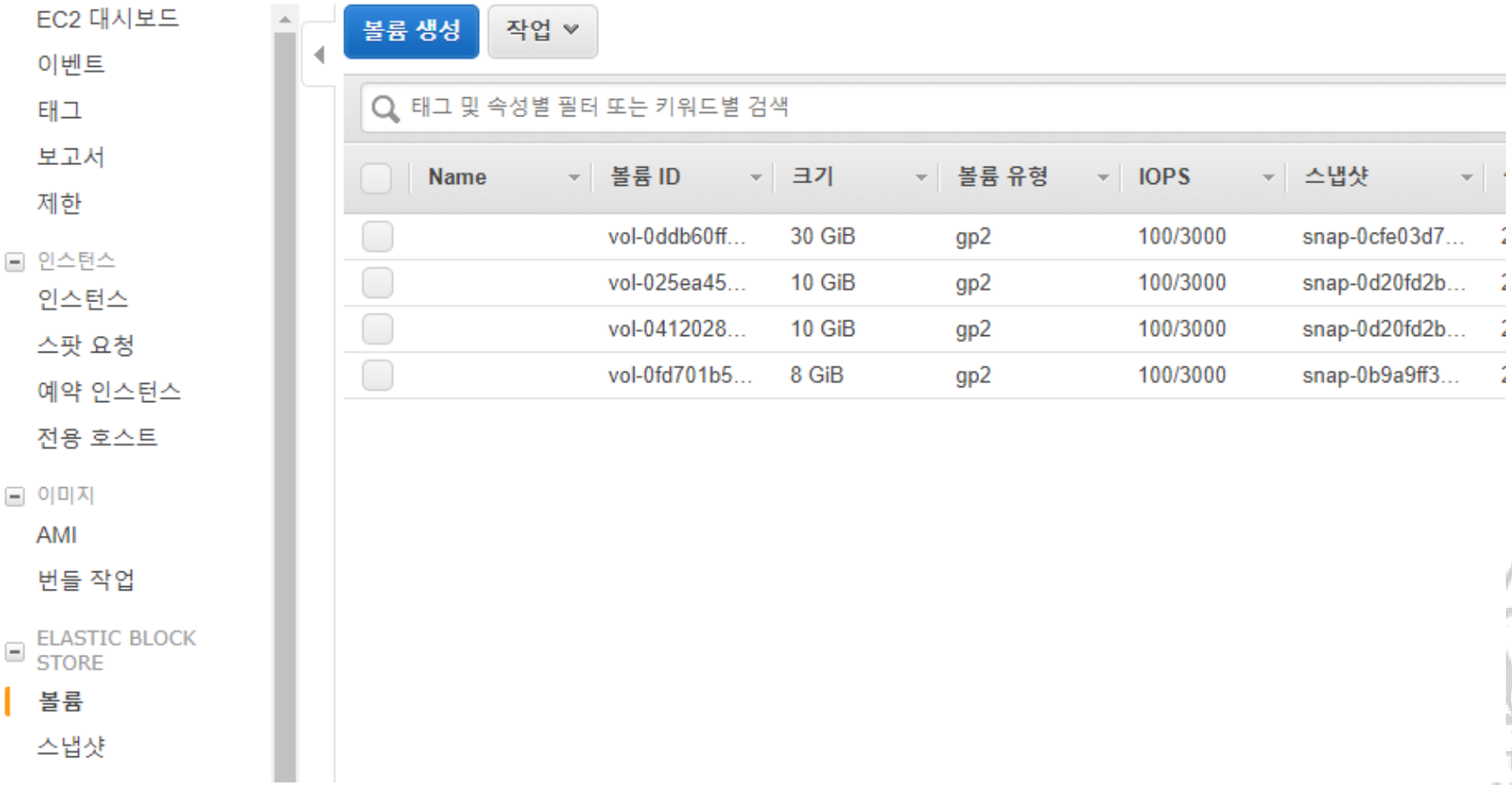
설명 | 상태 검사 | 모니터링 | 태그 | 사용량 제한

인스턴스 ID	i-06744c1d4a8f733c5
인스턴스 상태	running
인스턴스 유형	t2.micro
탄력적 IP	
가용 영역	ap-northeast-2a
보안 그룹	WordPress Certified by Bitnami and Automattic-5-2-3-0 on Ubuntu 16-04-AutogenByAWSMP-5. 인바운드 규칙 보기. 아웃바운드 규칙 보기

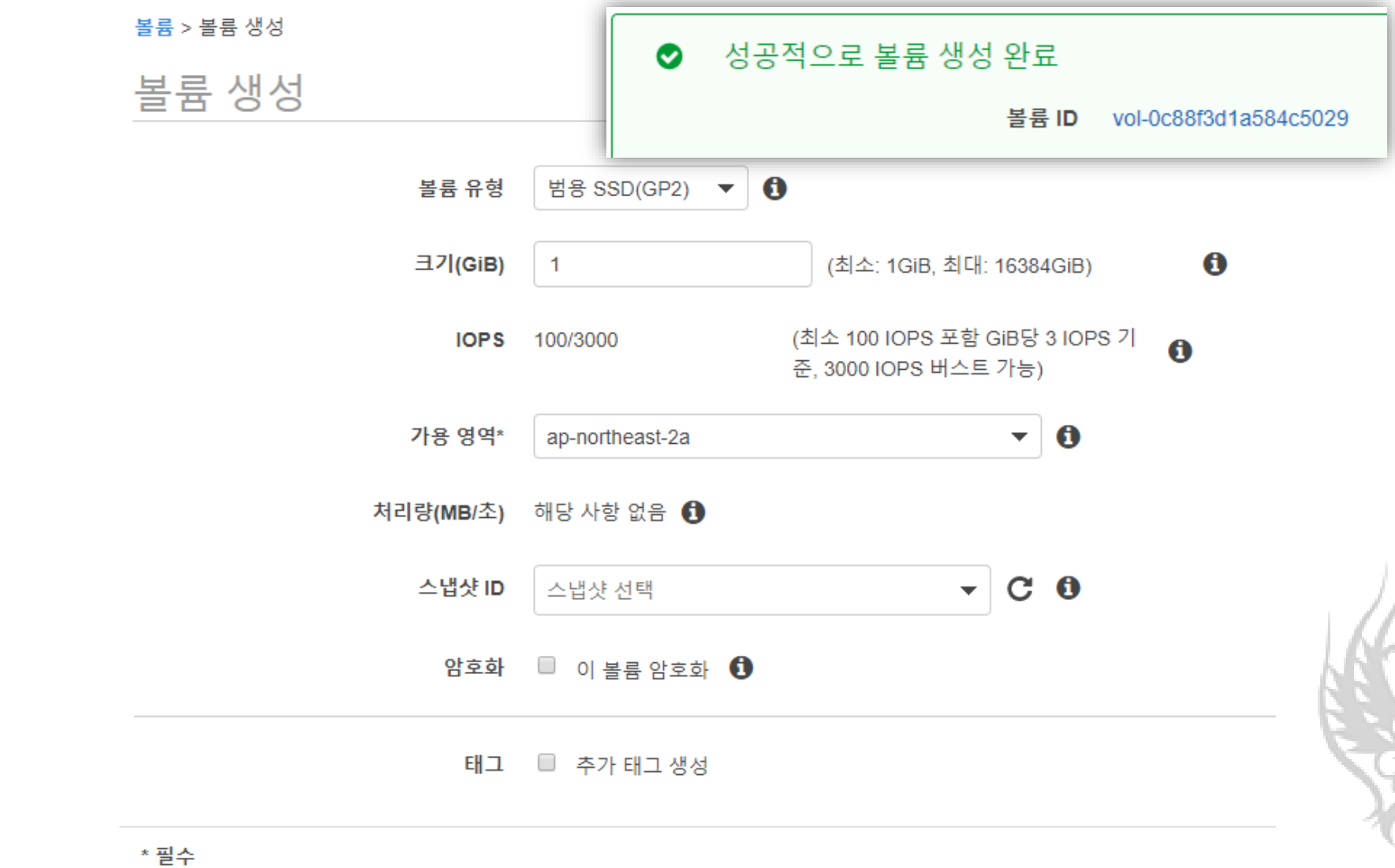
AMI location

979382823631/bitnami-wordpress-4.9.7-0-linux-redhat-7.4-x86_64-hvm-ebs

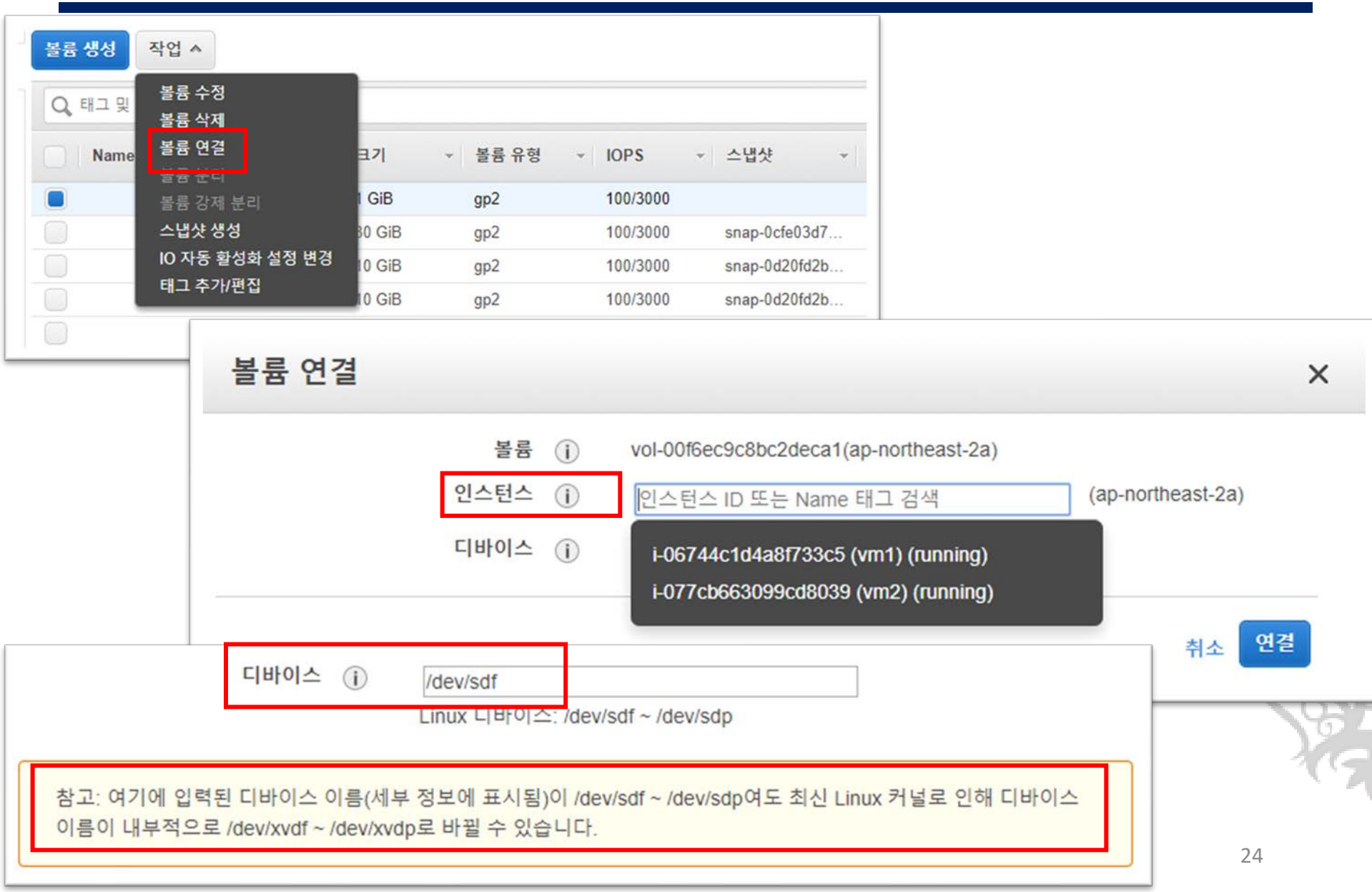
EC2 대시보드: EBS 볼륨



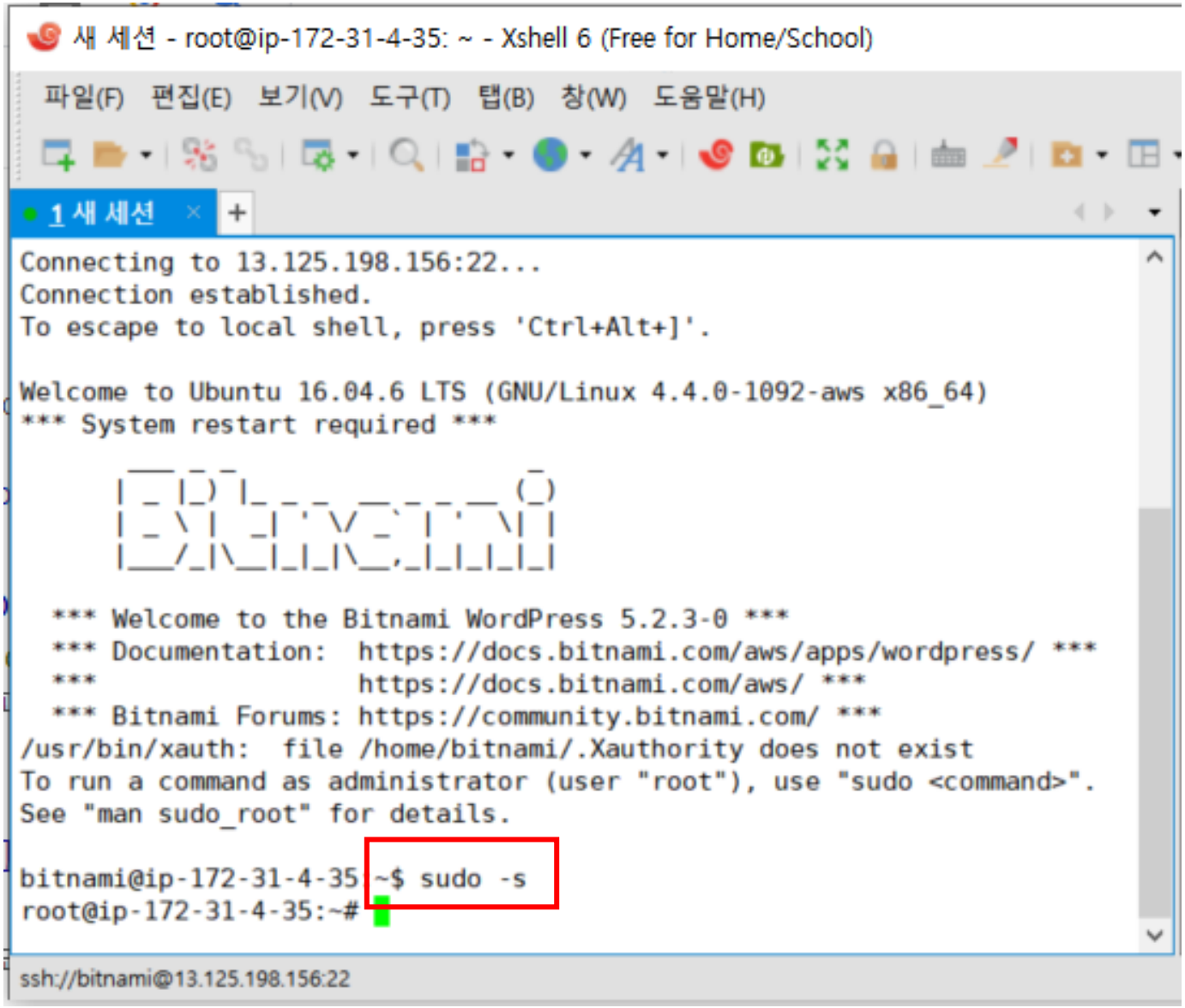
새로운 EBS 볼륨 생성


볼륨 > 볼륨 생성
볼륨 생성
성공적으로 볼륨 생성 완료
볼륨 ID vol-0c88f3d1a584c5029
볼륨 유형
범용 SSD(GP2)
크기(GiB)
1
(최소: 1GiB, 최대: 16384GiB)
IOPS
100/3000
(최소 100 IOPS 포함 GiB당 3 IOPS 기준, 3000 IOPS 버스트 가능)
가용 영역*
ap-northeast-2a
처리량(MB/초)
해당 사항 없음
스냅샷 ID
스냅샷 선택
암호화
이 볼륨 암호화
태그
추가 태그 생성
* 필수

새로운 EBS 볼륨을 1번 인스턴스에 연결



1번 인스턴스 SSH 연결 및 관리자 모드



장치 이름 확인

- lsblk (시스템 메시지 확인)
- 1G 용량의 disk 확인: xvdf -> /dev/xvdf 로 접근 가능

```
root@ip-172-31-4-35:~# lsblk
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda    202:0    0   10G  0 disk
└─xvda1 202:1    0   10G  0 part /
xvdf    202:80   0    1G  0 disk
loop0     7:0    0 12.6M  1 loop /snap/amazon-ssm-agent/295
loop1     7:1    0 86.9M  1 loop /snap/core/4917
loop2     7:2    0 89.1M  1 loop /snap/core/7917
loop3     7:3    0   18M  1 loop /snap/amazon-ssm-agent/1480
root@ip-172-31-4-35:~#
```

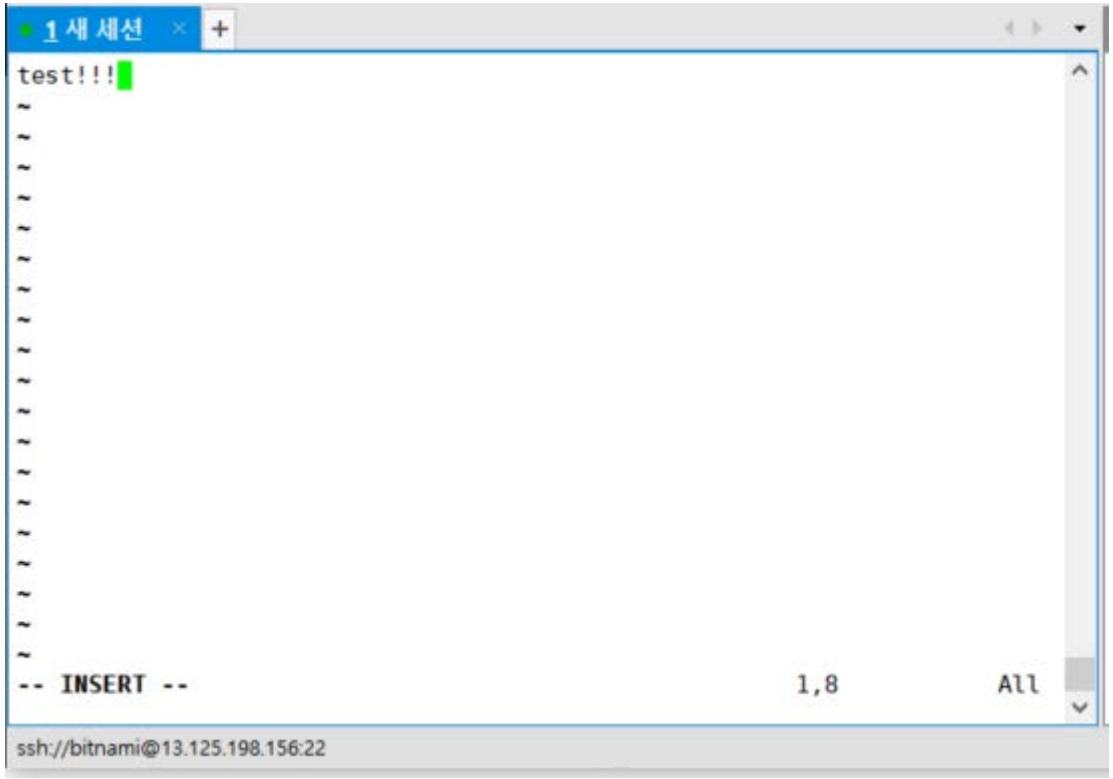
파일 시스템 생성 및 마운트

```
root@ip-172-31-4-35:~# mkfs -t ext4 /dev/xvdf
mke2fs 1.42.13 (17-May-2015)
Creating filesystem with 262144 4k blocks and 65536 inodes
Filesystem UUID: 46ca1b8b-bb85-45fb-b4eb-989068724fb8
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

root@ip-172-31-4-35:~# cd ~
root@ip-172-31-4-35:~# mkdir temp
root@ip-172-31-4-35:~# mount /dev/xvdf temp/
root@ip-172-31-4-35:~# ls -al temp/
total 24
drwxr-xr-x 3 root    root     4096 Oct 14 13:17 .
drwxr-xr-x 5 bitnami bitnami  4096 Oct 14 13:17 ..
drwx------ 2 root    root    16384 Oct 14 13:17 lost+found
root@ip-172-31-4-35:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            488M     0  488M   0% /dev
tmpfs           100M  3.3M   96M   4% /run
/dev/xvda1      9.7G  3.3G  6.4G  34% /
tmpfs           496M     0  496M   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           496M     0  496M   0% /sys/fs/cgroup
/dev/loop0       13M   13M     0 100% /snap/amazon-ssm-agent/295
/dev/loop1       87M   87M     0 100% /snap/core/4917
/dev/loop2       90M   90M     0 100% /snap/core/7917
/dev/loop3       18M   18M     0 100% /snap/amazon-ssm-agent/1480
tmpfs           100M     0  100M   0% /run/user/1000
/dev/xvdf       976M  1.3M  908M   1% /home/bitnami/temp
```

- mkfs: 파일 시스템 생성
- cd ~: 사용자 홈 디렉토리로 이동
- mount: 파일 시스템을 지정한 디렉토리를 통해 접근할 수 있도록 등록함
- df: 등록된 파일시스템 정보 확인

새로운 파일 생성

```
root@ip-172-31-4-35:~# vi temp/test.txt
root@ip-172-31-4-35:~# ls -al temp/
total 28
drwxr-xr-x 3 root    root     4096 Oct 14 13:20 .
drwxr-xr-x 5 bitnami bitnami  4096 Oct 14 13:20 ..
drwx------ 2 root    root    16384 Oct 14 13:17 lost+found
-rw-r--r-- 1 root    root        8 Oct 14 13:20 test.txt
root@ip-172-31-4-35:~#
```

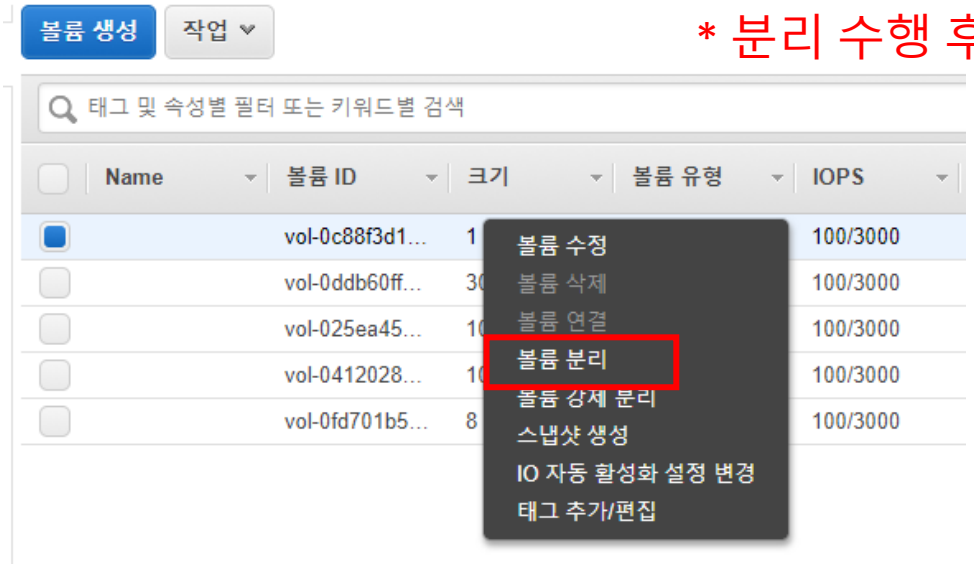


언마운트로 파일 시스템 등록 해제

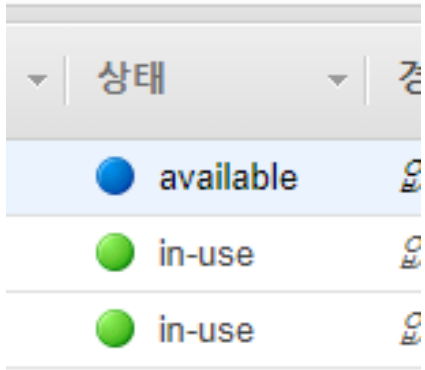
- umount: mount의 반대로 파일 시스템의 등록을 해제함
  - 해당 디렉토리의 파일이 오픈되어있거나, 디렉토리에 접근하고 있는 세션이 있다면 에러 발생

```
root@ip-172-31-4-35:~# umount temp
root@ip-172-31-4-35:~# df -h | grep xvdf
root@ip-172-31-4-35:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            488M     0  488M   0% /dev
tmpfs           100M  3.3M   96M   4% /run
/dev/xvda1      9.7G  3.3G  6.4G  34% /
tmpfs           496M     0  496M   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           496M     0  496M   0% /sys/fs/cgroup
/dev/loop0       13M   13M     0 100% /snap/amazon-ssm-agent/295
/dev/loop1       87M   87M     0 100% /snap/core/4917
/dev/loop2       90M   90M     0 100% /snap/core/7917
/dev/loop3       18M   18M     0 100% /snap/amazon-ssm-agent/1480
tmpfs           100M     0  100M   0% /run/user/1000
root@ip-172-31-4-35:~#
```

볼륨 분리 후, 리눅스 instance에 재연결



* 분리 수행 후, 상태가 available 로 변경되면 다시 연결



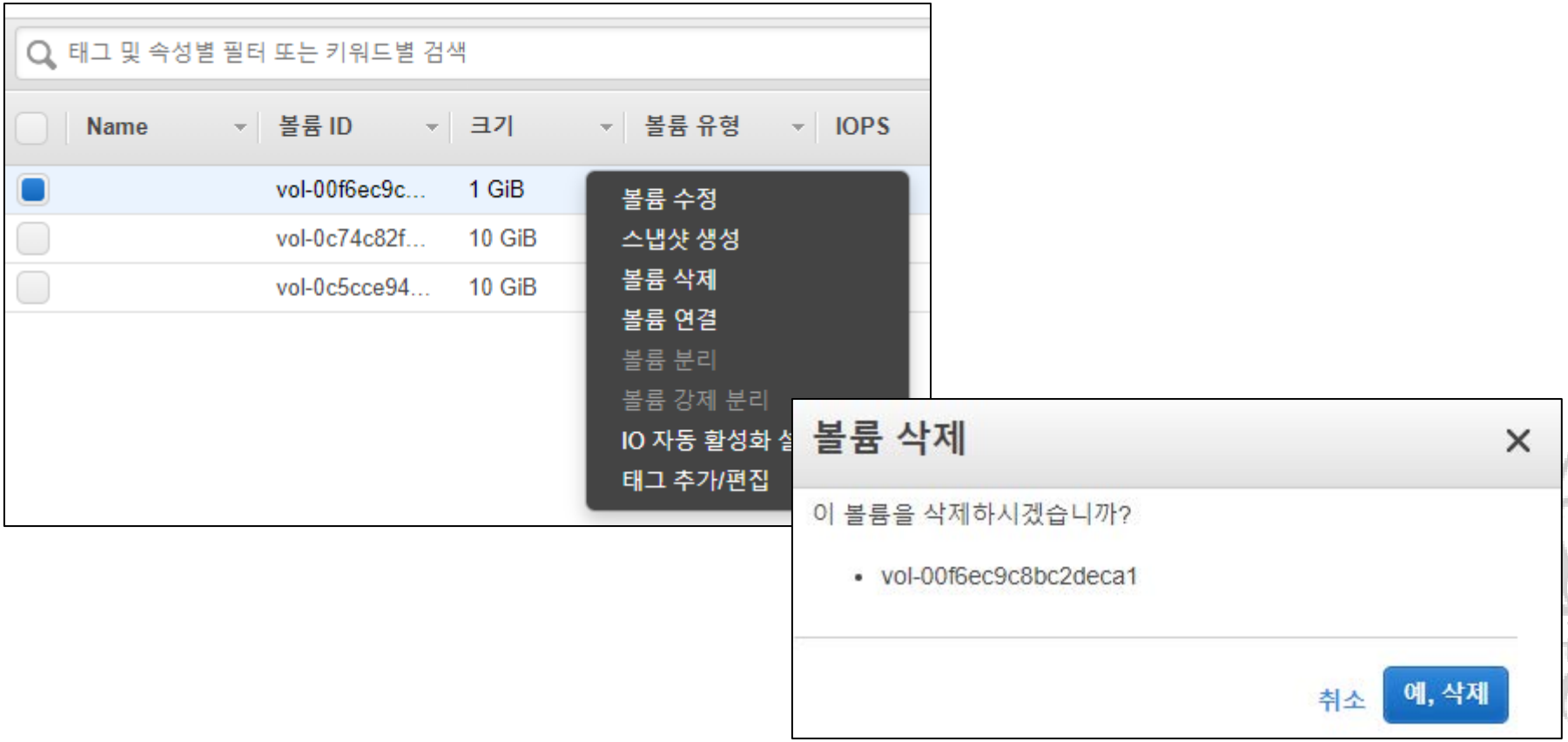
2번 인스턴스 접속, mount 수행하고 파일 확인

* 파일 시스템은 이미 만들었으므로, mkfs는 안 해도 됨

```
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-1092-aws x86_64)
*** System restart required ***

      ___ _ _                   _
     | _ |_) |_ _ _  __ _ _ __ (_)
     | _ \ |  _| ' \/ _` | '  \| |
     |___/_|\__|_|_|\__,_|_|_|_|_|

  *** Welcome to the Bitnami WordPress 5.2.3-0 ***
  *** Documentation:  https://docs.bitnami.com/aws/apps/wordpress/ ***
  ***                 https://docs.bitnami.com/aws/ ***
  *** Bitnami Forums: https://community.bitnami.com/ ***
/usr/bin/xauth:  file /home/bitnami/.Xauthority does not exist
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

bitnami@ip-172-31-7-103:~$ sudo -s
root@ip-172-31-7-103:~# lsblk | grep xvd
xvda    202:0    0   10G  0 disk
└─xvda1 202:1    0   10G  0 part /
xvdf    202:80   0    1G  0 disk
root@ip-172-31-7-103:~# cd ~
root@ip-172-31-7-103:~# mkdir temp
root@ip-172-31-7-103:~# mount /dev/xvdf temp/
root@ip-172-31-7-103:~# ls temp/
lost+found  test.txt
root@ip-172-31-7-103:~# cat temp/test.txt
test!!!
root@ip-172-31-7-103:~#
```

Umount 및 볼륨 분리, 삭제

```
root@ip-172-31-7-103:~# umount temp
root@ip-172-31-7-103:~#
```



EFS

EFS Pricing

리전: 아시아 태평양(서울)

https://aws.amazon.com/ko/efs/pricing/

표준 스토리지(GB-월)	0.33 USD
Infrequent Access 스토리지(GB-월)	0.0272 USD
Infrequent Access 요청(전송량(GB) 기준)	0.011 USD
프로비저닝된 처리량 유형(MB/s-월)	6.60 USD

AWS를 사용하는 첫 12개월 동안 매월 최대 5GB까지 EFS Standard 스토리지 클래스를 무료로 사용

- 예) 30GB 한달: $9.9 (약 12,000원)
- EBS보다 약 3배 가량 비쌈.
  - EFS는 보다 고급의 기능을 제공하기 때문
  - IA는 약 1/12 수준 (다만 계층 간 전송에 대해 요금 부과)

EFS Pricing

요금 예제

한 달에 청구되는 스토리지 양은 한 달 내내 사용하는 스토리지 공간의 평균을 기준으로 합니다. 스토리지 사용량은 "월별 GB" 단위로 측정되며, 이는 월말에 합산되어 월별 요금을 산출합니다. 이러한 요금 예제에서는 업계에서 인정한 예측에 따라 데이터 중 20%가 EFS Standard 스토리지 클래스에 저장되고 80%는 EFS IA 스토리지 클래스에 저장된다고 가정합니다.

월 단위로 비용이 청구되는 처리량은 해당 월에 EFS Standard 스토리지에서 허용하는 크기를 초과하여 프로비저닝된 평균 처리량에 기반합니다. 처리량은 "월별 MB/초" 단위로 측정되며, 이는 월말에 합산되어 월별 요금을 산출합니다.

예제 1:

파일 시스템이 미국 동부(버지니아 북부) 지역에 있고, 한달(31일) 내내 EFS Standard 스토리지 클래스에서 100GB의 스토리지를 사용하고, EFS Infrequent Access 스토리지 클래스에서 400GB의 스토리지를 사용한다고 가정합니다. 월말 시간당 GB 사용량은 다음과 같습니다.

총 EFS Standard 사용량(시간당 GB): 100GB x 31일 x (24시간/일) = 시간당 74,400GB
총 EFS IA 스토리지 사용량(시간당 GB): 400GB x 31일 x (24시간/일) = 시간당 297,600GB

시간당 GB를 합산한 후 월별 GB로 변환하여 월별 요금을 계산합니다.

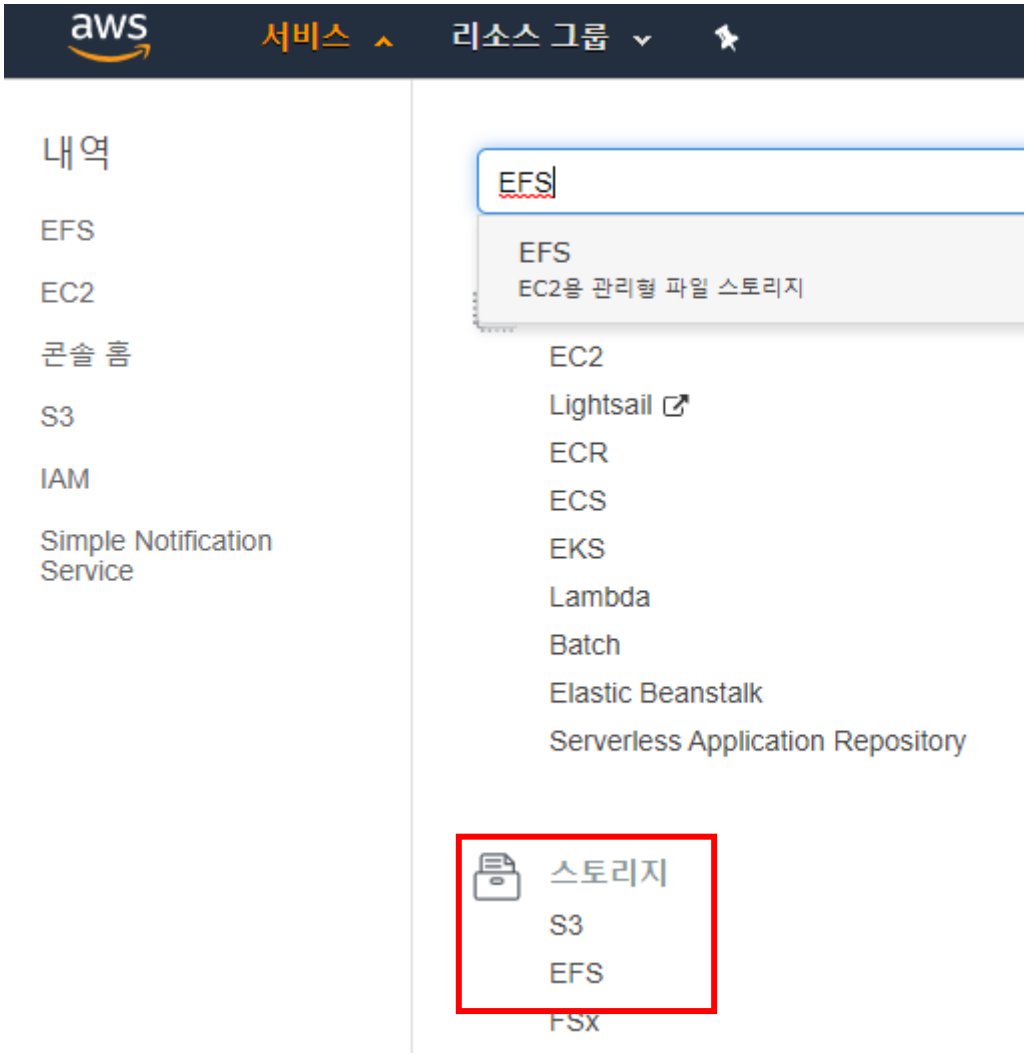
총 EFS Standard 스토리지 요금: 시간당 74,400GB x (1개월/744시간) x 월별 GB당 0.30 USD = 30.00 USD
총 EFS IA 스토리지 요금: 시간당 297,600GB x (1개월/744시간) x 월별 GB당 0.025 USD = 10.00 USD

총 월별 스토리지 요금: 30.00 USD + 10.00 USD = 40.00 USD 또는 월별 GB당 0.08 USD

실습 내용

- EBS에서 이용했던 2개의 인스턴스 이용
- 새로운 EFS 파일 시스템 생성
  - EFS는 한 리전의 모든 가용 영역에서 접근 가능하게 설정할 수 있음
- 1,2번 인스턴스에서 모두 연결 (mount)
- 1번 인스턴스에서 파일 생성
- 2번 인스턴스에서 파일 확인 후, 삭제
- 1번 인스턴스에서 확인
- 모든 인스턴스에서 umount
- EFS 삭제

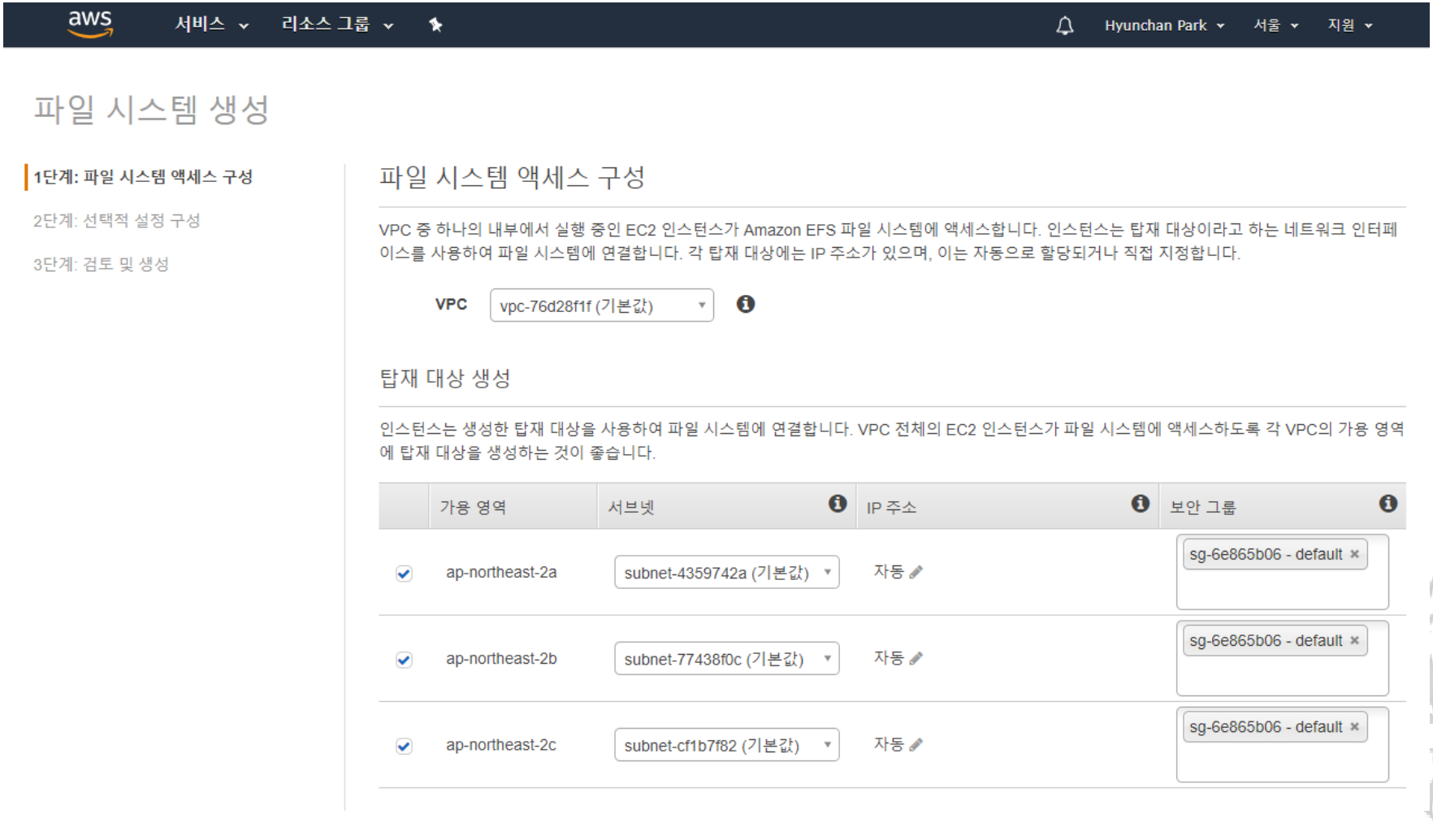
서비스에서 EFS 콘솔 진입



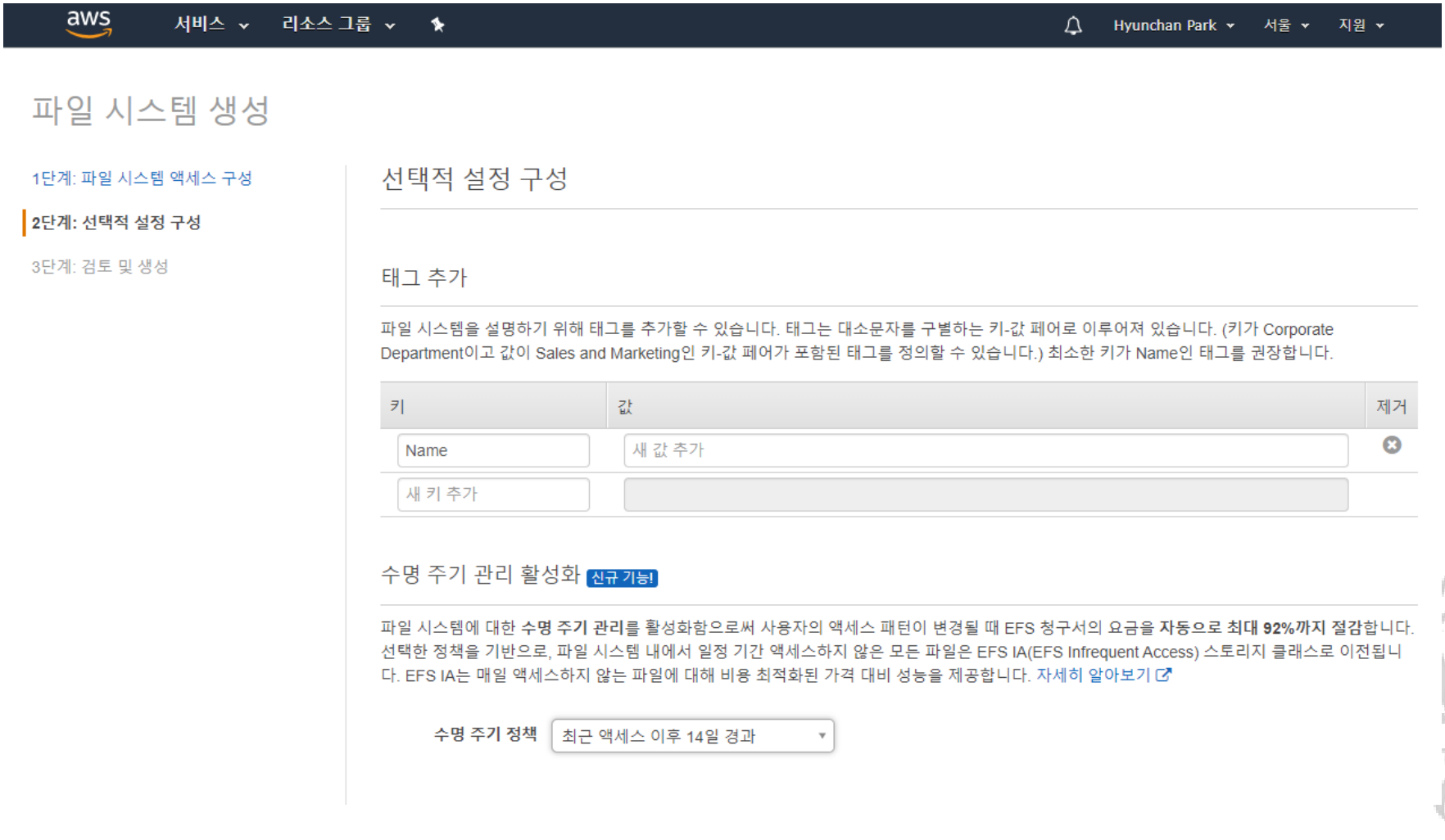
새로운 EFS 생성


aws
서비스
리소스 그룹
Hyunchan Park
서울
지원
파일 시스템
AWS DataSync
AWS Backup
EFS 수명 주기 관리를 통해 EFS 요금을 GB당 월 $0.08*로 줄이십시오. 자세히 알아보기
* 미국 동부(버지니아 북부) 지역의 요금은 EFS IA에 있는 스토리지의 80%로 추정됩니다.
새로운 소식 | 설명서 | AWS 스토리지 블로그
파일 시스템
파일 시스템 생성
작업
이름
파일 시스템 ID
측정 크기
탑재 대상 수
생성 날짜
파일 시스템 없음

액세스 구성 (기본값 사용)


aws
서비스
리소스 그룹
Hyunchan Park
서울
지원
파일 시스템 생성
1단계: 파일 시스템 액세스 구성
2단계: 선택적 설정 구성
3단계: 검토 및 생성
파일 시스템 액세스 구성
VPC 중 하나의 내부에서 실행 중인 EC2 인스턴스가 Amazon EFS 파일 시스템에 액세스합니다. 인스턴스는 탑재 대상이라고 하는 네트워크 인터페이스를 사용하여 파일 시스템에 연결합니다. 각 탑재 대상에는 IP 주소가 있으며, 이는 자동으로 할당되거나 직접 지정합니다.
VPC
vpc-76d28f1f (기본값)
탑재 대상 생성
인스턴스는 생성한 탑재 대상을 사용하여 파일 시스템에 연결합니다. VPC 전체의 EC2 인스턴스가 파일 시스템에 액세스하도록 각 VPC의 가용 영역에 탑재 대상을 생성하는 것이 좋습니다.
가용 영역
서브넷
IP 주소
보안 그룹
ap-northeast-2a
subnet-4359742a (기본값)
자동
sg-6e865b06 - default
ap-northeast-2b
subnet-77438f0c (기본값)
자동
sg-6e865b06 - default
ap-northeast-2c
subnet-cf1b7f82 (기본값)
자동
sg-6e865b06 - default

선택적 설정 구성 (기본값)


aws
서비스
리소스 그룹
Hyunchan Park
서울
지원
파일 시스템 생성
1단계: 파일 시스템 액세스 구성
2단계: 선택적 설정 구성
3단계: 검토 및 생성
선택적 설정 구성
태그 추가
파일 시스템을 설명하기 위해 태그를 추가할 수 있습니다. 태그는 대소문자를 구별하는 키-값 페어로 이루어져 있습니다. (키가 Corporate Department이고 값이 Sales and Marketing인 키-값 페어가 포함된 태그를 정의할 수 있습니다.) 최소한 키가 Name인 태그를 권장합니다.
키
값
제거
Name
새 값 추가
새 키 추가
수명 주기 관리 활성화
신규 기능!
파일 시스템에 대한 수명 주기 관리를 활성화함으로써 사용자의 액세스 패턴이 변경될 때 EFS 청구서의 요금을 자동으로 최대 92%까지 절감합니다. 선택한 정책을 기반으로, 파일 시스템 내에서 일정 기간 액세스하지 않은 모든 파일은 EFS IA(EFS Infrequent Access) 스토리지 클래스로 이전됩니다. EFS IA는 매일 액세스하지 않는 파일에 대해 비용 최적화된 가격 대비 성능을 제공합니다. 자세히 알아보기
수명 주기 정책
최근 액세스 이후 14일 경과

검토 및 생성

파일 시스템 생성

1단계: 파일 시스템 액세스 구성

2단계: 선택적 설정 구성

3단계: 검토 및 생성

검토 및 생성

파일 시스템 생성 전에 아래 구성을 검토합니다.

파일 시스템 액세스

VPC	가용 영역	서브넷	IP 주소	보안 그룹
vpc-76d28f1f (기본값)	ap-northeast-2a	subnet-4359742a (기본값)	자동	sg-6e865b06 - default
	ap-northeast-2b	subnet-77438f0c (기본값)	자동	sg-6e865b06 - default
	ap-northeast-2c	subnet-cf1b7f82 (기본값)	자동	sg-6e865b06 - default

선택적 설정

태그 추가된 태그 없음
성능 모드 General Purpose
처리량 모드 Bursting
암호화 됨 아니요
수명 주기 정책 최근 액세스 이후 14일 경과

취소 이전 파일 시스템 생성

파일 시스템

성공!

파일 시스템을 생성했습니다. 설치된 NFSv4.1 클라이언트를 사용하여 EC2 인스턴스로부터 파일 시스템을 탑재할 수 있습니다. 또한 AWS Direct Connect 또는 AWS VPN 연결을 통해 온프레미스 서버로부터 파일 시스템을 탑재할 수 있습니다. 여기를 클릭하여 EC2 탑재 지침을, 여기를 클릭하여 온프레미스 탑재 지침을 참조하십시오.

파일 시스템 생성 | 작업

	이름	파일 시스템 ID	측정 크기	탑재 대상 수	생성 날짜
◉	▼	fs-8e2450ef	6.0 KiB	3	2019. 10. 14. 13시 37분 54초 UTC

기타 세부 정보

소유자 ID 458235646241
파일 시스템 상태 사용 가능
성능 모드 General Purpose
처리량 모드 Bursting
암호화 됨 아니요
수명 주기 정책 최근 액세스 이후 14일 경과

태그

태그 관리

추가된 태그 없음

파일 시스템 액세스

파일 시스템 액세스 관리

DNS 이름 fs-8e2450ef.efs.ap-northeast-2.amazonaws.com

Amazon EC2 탑재 지침(로컬 VPC에서)
Amazon EC2 탑재 지침(VPC 피어링 연결 간)
온프레미스 탑재 지침

Amazon EC2 탑재 지침 1

Amazon EC2 탑재 지침(로컬 VPC에서)

EC2 인스턴스를 다음과 같이 설정하려면

- Amazon EC2 콘솔을 사용하여 탑재 대상에 대한 액세스를 활성화하는 VPC 보안 그룹과 EC2 인스턴스를 연결합니다. 예를 들어 "기본" 보안 그룹을 탑재 대상에 할당한 경우 "기본" 보안 그룹을 EC2 인스턴스에 할당해야 합니다. 자세히 알아보기
- SSH 클라이언트를 열고 EC2 인스턴스에 연결합니다. (연결 방법을 알아보십시오.)
- Amazon Linux EC2 인스턴스를 사용 중인 경우 다음 명령을 사용하여 EFS 탑재 헬퍼를 설치합니다.

  ```
  sudo yum install -y amazon-efs-utils
  ```

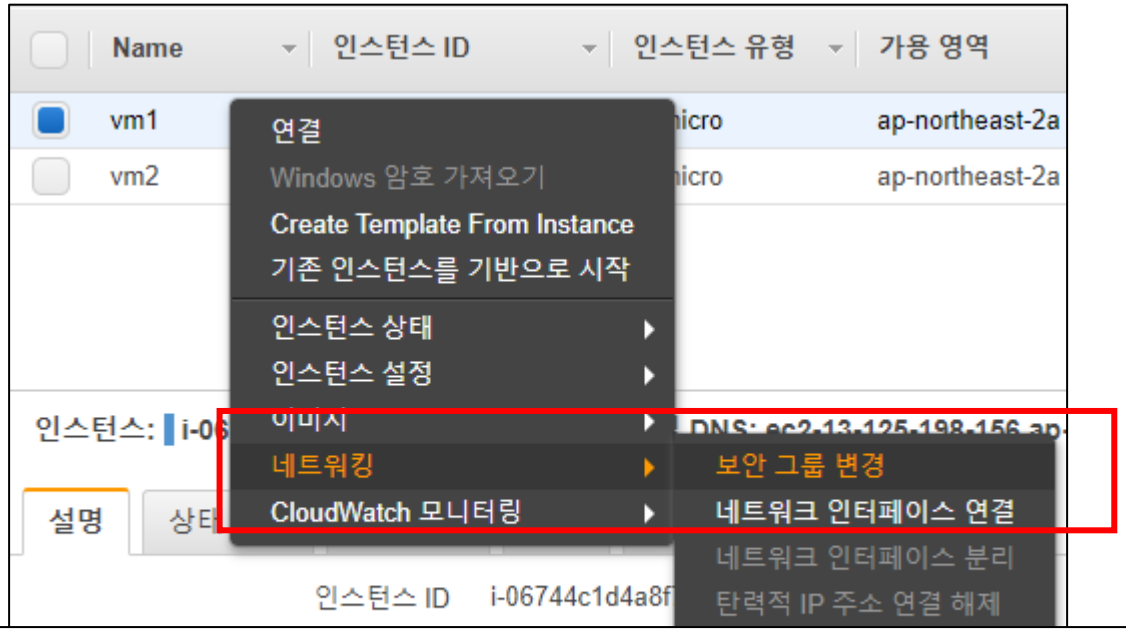
  Amazon Linux 인스턴스를 사용 중이지 않은 경우에도 EFS 탑재 헬퍼를 사용할 수 있습니다. 자세히 알아보기

  EFS 탑재 헬퍼를 사용 중이지 않은 경우 EC2 인스턴스에 NFS 클라이언트를 설치하십시오.
  - Red Hat Enterprise Linux 또는 SUSE Linux 인스턴스의 경우 이 명령을 사용합니다.

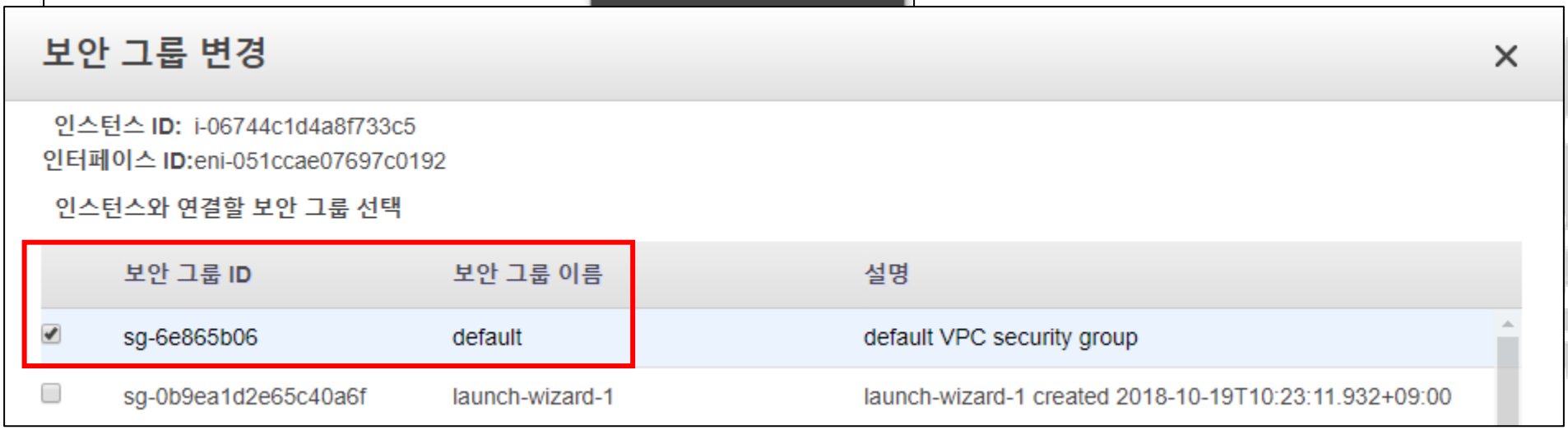
    ```
    sudo yum install -y nfs-utils
    ```
  - Ubuntu 인스턴스의 경우 이 명령을 사용합니다.

    ```
    sudo apt-get install nfs-common
    ```

인스턴스들 보안그룹을 EFS와 동일하게



- VPC의 기본 보안 그룹을 사용함 (ID는 다를 수 있음)
- 기존 보안그룹은 그대로 두고, 추가로 선택하면 됨



SSH 접속 후, nfs-utils 패키지 설치

- Sudo yum install nfs-utils
  - 관련 내용 동영상 참조

Amazon EC2 탑재 지침 2

Amazon EC2 탑재 지침(로컬 VPC에서)

파일 시스템 탑재

1. SSH 클라이언트를 열고 EC2 인스턴스에 연결합니다. (연결 방법을 알아보십시오).
2. EC2 인스턴스에 "efs"와 같은 새 디렉터리를 생성합니다.
   - `sudo mkdir efs`
3. 다음 메서드를 사용하여 파일 시스템을 탑재합니다. 전송 중 데이터 암호화가 필요한 경우 EFS 탑재 헬퍼와 TLS 탑재 옵션을 사용합니다.
   탑재 고려 사항
   - EFS 탑재 헬퍼 사용:
     `sudo mount -t efs fs-8e2450ef:/ efs`
   - EFS 탑재 헬퍼와 TLS 탑재 옵션 사용:
     `sudo mount -t efs -o tls fs-8e2450ef:/ efs`
   - NFS 클라이언트 사용:
     `sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-8e2450ef.efs.ap-northeast-2.amazonaws.com:/ efs`

연결할 수 없는 경우 문제 해결 설명서를 참조하십시오.

* 해당 EFS의 id, 주소가 자동 기입되어 있으므로, 복사->붙여넣기로 바로 이용 가능

```
bitnami@ip-172-31-4-35:~$ cd ~
bitnami@ip-172-31-4-35:~$ pwd
/home/bitnami
bitnami@ip-172-31-4-35:~$ mkdir efs
bitnami@ip-172-31-4-35:~$ sudo mount -t nfs4 -o nfsvers=4.1,rsize=104857
6,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-8e2450ef.efs.ap-n
ortheast-2.amazonaws.com:/ efs
bitnami@ip-172-31-4-35:~$ sudo chown bitnami:bitnami efs
bitnami@ip-172-31-4-35:~$ ls -al efs/
total 8
drwxr-xr-x 2 bitnami bitnami 6144 Oct 14 13:37 .
drwxr-xr-x 6 bitnami bitnami 4096 Oct 14 13:52 ..
bitnami@ip-172-31-4-35:~$ vi efs/test.txt
bitnami@ip-172-31-4-35:~$ cat efs/test.txt
test!!!
bitnami@ip-172-31-4-35:~$ ls efs/
test.txt
bitnami@ip-172-31-4-35:~$
```

* pwd: 현재 작업 중인 디렉토리 확인

* chown: 해당 파일의 소유자:그룹 변경

2번 인스턴스에서도 mount 후, 내용 확인

1 13.209.18.253

```
bitnami@ip-172-31-7-103:~$ cd ~
bitnami@ip-172-31-7-103:~$ pwd
/home/bitnami
bitnami@ip-172-31-7-103:~$ mkdir efs
bitnami@ip-172-31-7-103:~$ sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048
e2450ef.efs.ap-northeast-2.amazonaws.com:/ efs
bitnami@ip-172-31-7-103:~$ ls -al efs/
total 12
drwxr-xr-x 2 bitnami bitnami 6144 Oct 14 13:52 .
drwxr-xr-x 6 bitnami bitnami 4096 Oct 14 13:57 ..
-rw-rw-r-- 1 bitnami bitnami    8 Oct 14 13:52 test.txt
bitnami@ip-172-31-7-103:~$ cat efs/test.txt
test!!!
bitnami@ip-172-31-7-103:~$ rm efs/test.txt
bitnami@ip-172-31-7-103:~$ sudo umount efs/
bitnami@ip-172-31-7-103:~$
```

* 해당 파일 삭제하고,

umount 까지 수행

1번 인스턴스에서 다시 확인

- EFS 파일 시스템이 두 인스턴스에 공유된 상태였음
- 1번 인스턴스는 umount를 수행하지 않은 상태에서,
  2번 인스턴스에서 수행한 파일 삭제가 이루어진 것을 즉각 확인할 수 있음

```
bitnami@ip-172-31-4-35:~$ ls -al efs/
total 8
drwxr-xr-x 2 bitnami bitnami 6144 Oct 14 13:57 .
drwxr-xr-x 6 bitnami bitnami 4096 Oct 14 13:52 ..
bitnami@ip-172-31-4-35:~$ sudo umount efs
bitnami@ip-172-31-4-35:~$
```

EFS 삭제

- 양쪽 인스턴스에서 모두 umonut 한 후,
- EFS 콘솔에서 삭제 진행
  - EFS의 삭제는 불가역적이므로, ID를 수동 입력해 확인하는 절차가 있음
- 더 이상 사용하지 않는다면, 2개 인스턴스도 삭제