

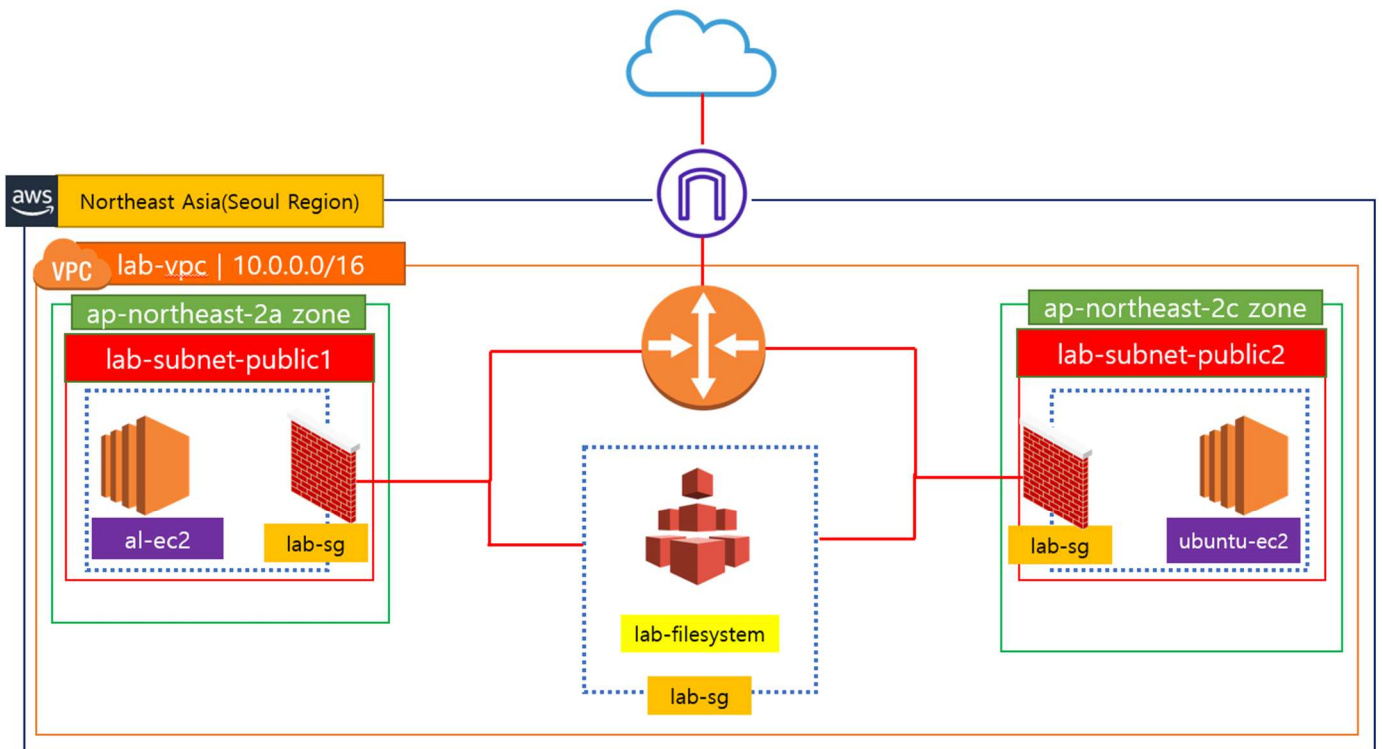
Lab3. Create Amazon EFS

목적

이번 실습에서는 Amazon EFS에 대해 다룬다. 파일 시스템을 설정하고 이 파일 시스템에 각각 다른 AZ에 있는 EC2 인스턴스가 NFS를 통해 같은 파일 시스템에 접근하고 파일을 생성하는 일련의 작업을 수행한다.

사전 준비물

AWS Free-Tier 계정



1. 다음과 같이 **lab-vpc**를 생성한다.

- A. [생성할 리소스] : VPC, 서브넷 등
- B. [이름 태그 자동 생성] : 자동 생성 / lab
- C. [IPv4 CIDR] : 10.0.0.0/16
- D. [테넌시] : 기본값

VPC 설정

생성할 리소스 정보

VPC 리소스만 생성 또는 VPC, 서브넷 등을 생성

☐ VPC만

☒ VPC, 서브넷 등

이름 태그 자동 생성 정보

이름 태그의 값을 입력합니다. 이 값은 VPC의 모든 리소스에 대한 이름 태그를 자동으로 생성하는 데 사용됩니다.

☒ 자동 생성

lab

IPv4 CIDR 블록 정보

CIDR 표기법을 사용하여 VPC의 시작 IP와 크기를 결정합니다.

10.0.0.0/1665,536 IPs

IPv6 CIDR 블록 정보

☒ IPv6 CIDR 블록 없음

☐ Amazon 제공 IPv6 CIDR 블록

테넌시 정보

기본값

- E. [가용 영역(AZ)] : 2
- F. [첫 번째 가용 영역] : ap-northeast-2a
- G. [두 번째 가용 영역] : ap-northeast-2c
- H. [퍼블릭 서브넷 수] : 2
- I. [프라이빗 서브넷 수] : 2
- J. [NAT 게이트웨이] : AZ당 1개

가용 영역(AZ) 정보

서브넷을 프로비저닝할 AZ 수를 선택합니다.고가용성을 위해서는 최소 2개 이상의 AZ를 사용하는 것이 좋습니다.

1

2

3

▼ AZ 사용자 지정

첫 번째 가용 영역

ap-northeast-2a ▼

두 번째 가용 영역

ap-northeast-2c ▼

퍼블릭 서브넷 수 정보

VPC에 추가할 퍼블릭 서브넷 수입니다. 인터넷을 통해 공개적으로 액세스할 수 있어야 하는 웹 애플리케이션에는 퍼블릭 서브넷을 사용합니다.

0

2

프라이빗 서브넷 수 정보

VPC에 추가할 프라이빗 서브넷 수입니다. 프라이빗 서브넷을 사용하여 퍼블릭 액세스가 필요 없는 백엔드 리소스를 보호합니다.

0

2

4

▶ 서브넷 CIDR 블록 사용자 지정

NAT 게이트웨이(\$) 정보

NAT 게이트웨이를 생성할 가용 영역(AZ) 수를 선택합니다. 각 NAT 게이트웨이에마다 요금이 부과됩니다.

없음

1개의 AZ에서

AZ당 1개

K. [DNS 옵션] : DNS 호스트 이름 활성화

NAT 게이트웨이(\$) 정보

NAT 게이트웨이를 생성할 가용 영역(AZ) 수를 선택합니다. 각 NAT 게이트웨이마다 요금이 부과됩니다.

없음

1개의 AZ에서

AZ당 1개

VPC 엔드포인트 정보

엔드포인트는 VPC에서 S3에 직접 액세스하여 NAT 게이트웨이 요금을 줄이고 보안을 강화할 수 있습니다. 기본적으로 모든 액세스 정책이 사용됩니다. 언제든지 이 정책을 사용자 지정할 수 있습니다.

없음

S3 게이트웨이

DNS 옵션 정보

☒ DNS 호스트 이름 활성화

☒ DNS 확인 활성화

취소

VPC 생성

2. 이제 EFS를 사용할 EC2 인스턴스를 생성한다. 인스턴스는 2개 생성하는데, 각각의 설정정보는 다음과 같다.

A. al-ec2

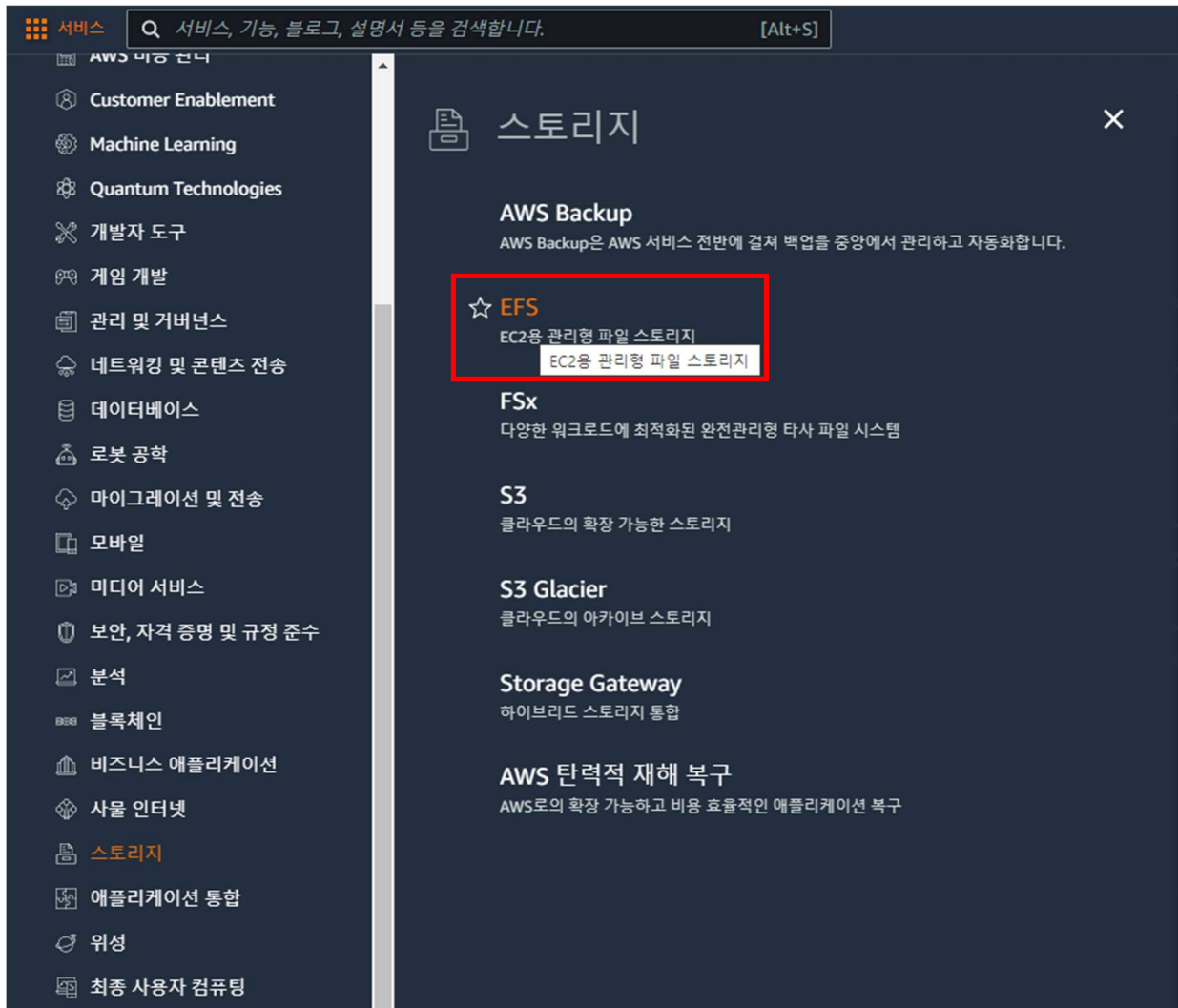
- ① [AMI 선택] : Amazon Linux 2 AMI(HVM)- Kernel 5.10, SSD Volume Type, 64비트(x86)
- ② [인스턴스 유형] : t2.micro
- ③ [네트워크] : lab-vpc, lab-subnet-public1-ap-northeast-2a, 퍼블릭 IP 자동 할당 활성화
- ④ [스토리지] : 범용 SSD(gp2), 30GiB
- ⑤ [태그] : Name/al-ec2
- ⑥ [보안 그룹] : 새 보안 그룹 생성, lab-sg, SSH, 22, 위치 무관, 0.0.0.0/0
- ⑦ [키 페어] : 새 키 페어 생성, al-ec2-key.pem

B. ubuntu-ec2

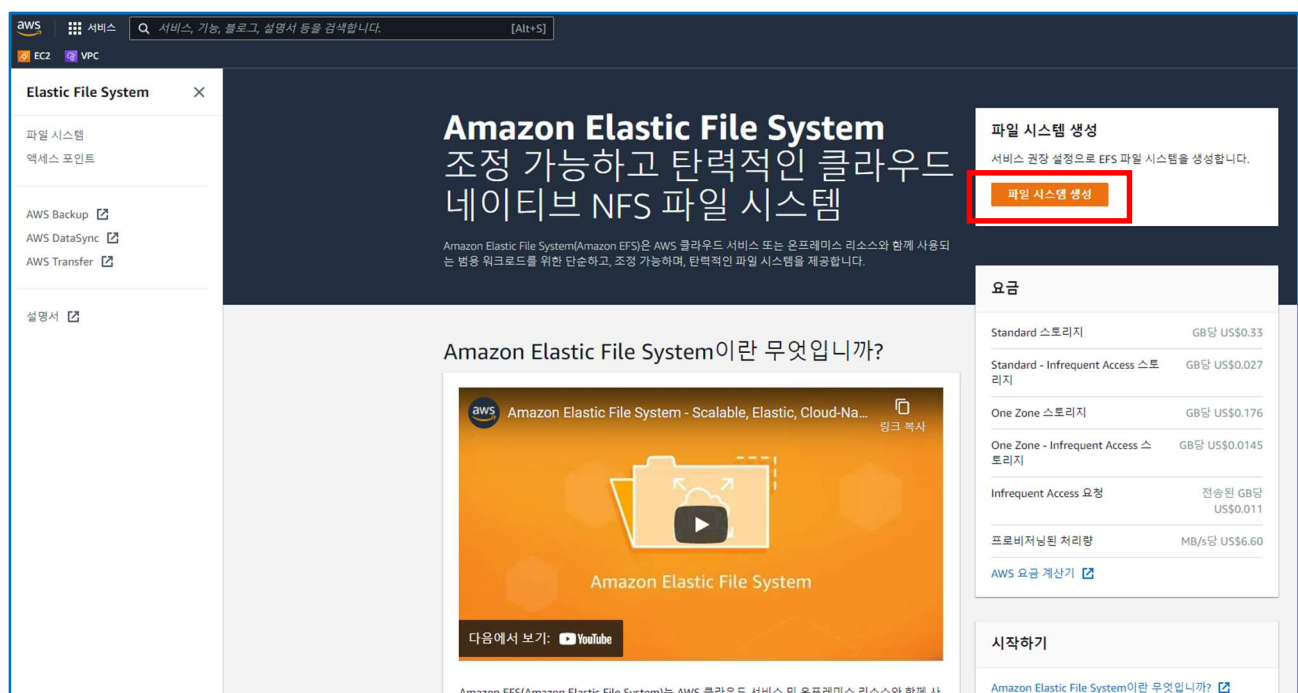
- ① [AMI 선택] : Ubuntu Server 20.04 LTS(HVM), SSD Volume Type, 64비트(x86)
- ② [인스턴스 유형] : t2.micro
- ③ [네트워크] : lab-vpc, lab-subnet-public2-ap-northeast-2c, 퍼블릭 IP 자동 할당 활성화
- ④ [스토리지] : 범용 SSD(gp2), 30GiB
- ⑤ [태그] : Name/ubuntu-ec2
- ⑥ [보안 그룹] : 기존 보안 그룹 선택, lab-sg, SSH, 22, 위치 무관, 0.0.0.0/0
- ⑦ [키 페어] : 새 키 페어 생성, ubuntu-ec2-key.pem

인스턴스 (2) 정보										
<div>🔍 검색</div>										
<div>< 1 > ⚙</div>										
<input type="checkbox"/>	Name	인스턴스 ID	인스턴스 상태	인스턴스 유형	상태 검사	경보 상태	가용 영역	퍼블릭 IPv4 DNS	퍼블릭 IPv4 ...	타
<input type="checkbox"/>	al-ec2	i-0cc08f356a149eb2a	🟢 실행 중 🔍	t2.micro	🟢 2/2개 검사 통과	경보 없음 +	ap-northeast-2a	ec2-15-165-18-160.ap-...	15.165.18.160	-
<input type="checkbox"/>	ubuntu-ec2	i-0187778e998cd0d1a	🟢 실행 중 🔍	t2.micro	🟢 2/2개 검사 통과	경보 없음 +	ap-northeast-2c	ec2-13-209-70-29.ap-n...	13.209.70.29	-

3. [서비스] > [스토리지] > [EFS]를 선택한다.



4. Amazon EFS 페이지이다. 우측 상단쪽 [파일 시스템 생성] 버튼을 클릭한다.



5. [파일 시스템 생성]페이지이다. 다음 각각의 값을 설정하고 보다 자세한 설정을 위해 [사용자 지정]을 클릭한다.

A. [이름] : lab-filessystem

B. [Virtual Private Cloud(VPC)] : lab-vpc

C. [가용성 및 내구성] : 리전

파일 시스템 생성

서비스 권장 설정으로 EFS 파일 시스템을 생성합니다. 자세히 알아보기

이름 - 선택 사항

파일 시스템 이름을 지정합니다.

lab-filessystem

이름은 256자를 넘을 수 없으며 여기에는 문자, 숫자 및 + = . _ : / 기호만 포함해야 합니다.

Virtual Private Cloud(VPC)

EC2 인스턴스를 파일 시스템에 연결하고자 하는 VPC를 선택합니다. 자세히 알아보기

vpc-0c0d36636b09bf825

lab-vpc

가용성 및 내구성

리전 스토리지 클래스를 사용하여 파일 시스템을 생성하려면 리전을 선택합니다(권장). One Zone 스토리지 클래스를 사용하여 파일 시스템을 생성하려면 One Zone을 선택합니다. 자세히 알아보기

☒ 리전

여러 AZ에 걸쳐 데이터를 중복 저장

☐ One Zone

단일 AZ 내에 데이터를 중복 저장

취소

사용자 지정

생성

6. 1단계 [파일 시스템 설정] 페이지이다. 다음의 각 값을 설정하고 [다음]을 클릭한다.

A. [이름] : lab-filesystem

B. [가용성 및 내구성] : 리전

C. [자동 백업] : 자동 백업 활성화

D. [수명 주기 관리] : IA로 전환 / 마지막 액세스 이후 30일 경과

E. [성능 모드] : 범용

Amazon EFS > 파일 시스템 > 생성

1단계
파일 시스템 설정

2단계
네트워크 액세스

3단계 - 선택 사항
파일 시스템 정책

4단계
검토 및 생성

파일 시스템 설정

일반

이름 - 선택 사항
파일 시스템 이름을 지정합니다.
lab-filesystem
이름은 256자를 넘을 수 없으며 여기에는 문자, 숫자 및 + - = _ / 기호만 포함해야 합니다.

가용성 및 내구성
리전 스토리지 클래스를 사용하여 파일 시스템을 생성하려면 리전을 선택합니다(권장). One Zone 스토리지 클래스를 사용하여 파일 시스템을 생성하려면 One Zone을 선택합니다. 자세히 알아보기

☒ 리전
여러 AZ에 걸쳐 데이터를 중복 저장

☐ One Zone
단일 AZ 내에 데이터를 중복 저장

자동 백업
권장 설정을 사용하여 AWS Backup으로 파일 시스템 데이터를 자동으로 백업합니다. 추가 요금이 적용됩니다. 자세히 알아보기

☒ 자동 백업 활성화

수명 주기 관리
EFS Intelligent-Tiering은 수명 주기를 사용하여 파일을 Standard와 Standard-Infrequent Access 스토리지 클래스 사이에서 이동함으로써 애플리케이션에 적합한 가격 및 성능 불연속을 자동으로 실현합니다. 자세히 알아보기

IA로 전환
파일을 Standard에서 Standard-Infrequent Access로 전환합니다.
마지막 액세스 이후 30일 경과

성능 모드
필요한 IOPS를 기반으로 파일 시스템의 성능 모드를 설정합니다. 자세히 알아보기

☒ 범용
웹 서비스 환경 및 콘텐츠 관리 시스템과 같은 지연 시간에 민감한 사용 사례에 적합

☐ 최대 I/O
더 높은 수준의 집계 처리량 및 조당 작업 수로 확장

태그 - 선택 사항
태그를 추가하여 키-값 페어를 리소스에 연결합니다. 자세히 알아보기

태그 키
Q Name X

태그 값 - 선택 사항
Q lab-filesystem X

태그 제거

태그 추가
태그를 49개 더 추가할 수 있음

취소 다음

F. [처리량 모드] : 버스트

G. [암호화] : 유휴 시 데이터 암호화 활성화

H. [태그] : Name / lab-filesystem

처리량 모드
파일 시스템의 처리량 제한이 결정되는 방식을 설정합니다. 자세히 알아보기

☒ 버스트
파일 시스템 크기에 따라 처리량 조정

☐ 프로비저닝됨
지정된 양으로 처리량 고정

암호화
파일 시스템의 유휴 데이터의 암호화를 활성화하려면 선택합니다. 기본적으로 AWS KMS 서비스 키(jaws/elasticfilesystem)를 사용합니다. 자세히 알아보기

☒ 유휴 시 데이터 암호화 활성화

▶ 암호화 설정 사용자 지정

태그 - 선택 사항
태그를 추가하여 키-값 페어를 리소스에 연결합니다. 자세히 알아보기

태그 키
Q Name X

태그 값 - 선택 사항
Q lab-filesystem X

태그 제거

태그 추가
태그를 49개 더 추가할 수 있음

취소 다음

7. 2단계 [네트워크 액세스] 페이지이다. 다음의 각 값을 설정하고 [다음]을 클릭한다.

A. Virtual Private Cloud(VPC) : lab-vpc

B. [탑재 대상]

① [가용 영역] : ap-northeast-2a, [서브넷 ID] : lab-subnet-public1-ap-northeast-2a, [보안 그룹] : lab-sg

② [가용 영역] : ap-northeast-2c, [서브넷 ID] : lab-subnet-public2-ap-northeast-2c, [보안 그룹] : lab-sg

네트워크 액세스

네트워크

Virtual Private Cloud (VPC)
EC2 인스턴스를 파일 시스템에 연결하고자 하는 VPC를 선택합니다. 자세히 알아보기

vpc-0b682c6e2dffbfddf
lab-vpc

탑재 대상
탑재 대상은 Amazon EFS 파일 시스템을 탑재할 수 있는 NFSv4 엔드포인트를 제공합니다. 가용 영역마다 탑재 대상을 한 개씩 생성하는 것이 좋습니다. 자세히 알아보기

가용 영역	서브넷 ID	IP 주소	보안 그룹	
ap-northeast-2a	subnet-07d51cc2eb5daba5c	자동	보안 그룹 선택 sg-03db0c341dfbeb965 lab-sg	제거
ap-northeast-2c	subnet-06b59395aa7fac2f6	자동	보안 그룹 선택 sg-03db0c341dfbeb965 lab-sg	제거

탑재 대상 추가

취소 이진 다음

8. 3단계 [파일 시스템 정책] 페이지이다. 기본값 그대로 놓고 [다음]을 클릭한다.

파일 시스템 정책 - 선택 사항

정책 옵션

이러한 공통 정책 옵션 중 하나 이상을 선택하거나 편집기를 사용하여 사용자 지정 정책을 하나 더 생성합니다. 자세히 알아보기

- ☐ 기본적으로 루트 액세스 금지*
- ☐ 기본적으로 읽기 전용 액세스 적용*
- ☐ 익명 액세스 금지
- ☐ 모든 클라이언트에 대해 전송 중 암호화 적용

* 자격 증명 기반 정책은 이러한 기본 권한을 대체할 수 있습니다.

▶ 추가 권한 부여

정책 편집기 (JSON)

지우기

수동으로 변경하면 편집기를 비울 때까지 왼쪽의 정책 옵션을 사용할 수 없습니다.

취소 이진 다음

9. 4단계 [검토 및 생성] 페이지이다. 실습을 위해 기본값 그대로 놓고 페이지를 스크롤다운하여 [생성] 버튼을 클릭한다.

검토 및 생성

1단계: 파일 시스템 설정 편집

필드	값	편집 가능합니까?
이름	lab-file-system	예
성능 모드	범용	아니오
처리량 모드	버스트	예
암호화됨	예	아니오
KMS 키 ID	-	아니오
수명 주기 관리	IA로 전환: 마지막 액세스 이후 30일 경과 IA 외부로 전환: 처음 액세스할 때	예
자동 백업	예	예
VPC ID	vpc-0b682c6e2dffbfddf (lab-vpc)	예
가용 영역	리전	아니오

3단계: 파일 시스템 정책 편집

▼ 파일 시스템 정책

취소 이전 생성

10. 잠시 기다리면 다음 그림과 같이 EFS가 성공적으로 생성된 것을 확인할 수 있다.

🟢 성공! 파일 시스템(fs-0836bc1abb7a14a8a)이 사용 가능함 파일 시스템 보기

Amazon EFS > 파일 시스템

📢 EFS 복제 소개

선택한 리전 또는 가용 영역에 파일 시스템의 최신 사본을 보관하세요.

[새로운 소식](#) | [설명서](#) | [AWS 스토리지 블로그](#)

파일 시스템 (1) 새부 정보 보기 삭제 **파일 시스템 생성**

	이름	파일 시스템 ID	암호화됨	총 크기	Standard/On e Zone의 크기	Standard- IA/One Zone- IA의 크기	프로비저닝된 처리량(MiB/s)	파일 시스템 상태	생성 시간	가용 영역
<input type="radio"/>	lab-file-system	fs-0836bc1abb7a14a8a	🟢 암호화됨	6.00 KiB	6.00 KiB	0바이트	-	🟢 사용 가능	Wed, 11 May 2022 08:14:54 GMT	리전

11. 이제 EFS에 접근할 EC2 인스턴스 페이지로 이동한다. 먼저 **al-ec2**를 선택하고 **[연결]** 버튼을 클릭한다. **[SSH 클라이언트]** 탭으로 이동하여 해당 **[퍼블릭 DNS]**의 주소를 이용하여 **SSH** 클라이언트 툴에서 연결한다.

EC2 > 인스턴스 > i-0cc08f356a149eb2a > 인스턴스에 연결

인스턴스에 연결 정보

다음 옵션 중 하나를 사용하여 인스턴스 i-0cc08f356a149eb2a (al-ec2)에 연결

EC2 인스턴스 연결 | Session Manager | **SSH 클라이언트** | EC2 직렬 콘솔

인스턴스 ID
i-0cc08f356a149eb2a (al-ec2)

- SSH 클라이언트를 엽니다.
- 프라이빗 키 파일을 찾습니다. 이 인스턴스를 시작하는 데 사용되는 키는 al-ec2-key.pem입니다.
- 필요한 경우 이 명령을 실행하여 키를 공개적으로 볼 수 없도록 합니다.
chmod 400 al-ec2-key.pem
- 퍼블릭 DNS을(를) 사용하여 인스턴스에 연결:
ec2-15-165-18-160.ap-northeast-2.compute.amazonaws.com

예:

```
ssh -i "al-ec2-key.pem" ec2-user@ec2-15-165-18-160.ap-northeast-2.compute.amazonaws.com
```

참고: 대부분의 경우 추정된 사용자 이름은 정확합니다. 하지만 AMI 사용 지침을 읽고 AMI 소유자가 기본 AMI 사용자 이름을 변경했는지 확인하십시오.

```
1 al-ec2 * +
Xshell 7 (Build 0108)
Copyright (c) 2020 NetSarang Computer, Inc. All rights reserved.

Type 'help' to learn how to use Xshell prompt.
[C:\~]$

Host 'ec2-15-165-18-160.ap-northeast-2.compute.amazonaws.com' resolved to 15.165.18.160.
Connecting to 15.165.18.160:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+J'.

WARNING! The remote SSH server rejected X11 forwarding request.

  _ | _ | _ )
  _ | ( _ /   Amazon Linux 2 AMI
  _ | \ _ | _ |

https://aws.amazon.com/amazon-linux-2/
2 package(s) needed for security, out of 2 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-10-0-7-49 ~]$
```

12. 두번째 인스턴스 **ubuntu-ec2**도 연결한다.

EC2 > 인스턴스 > i-0187778e998cd0d1a > 인스턴스에 연결

인스턴스에 연결 정보

다음 옵션 중 하나를 사용하여 인스턴스 i-0187778e998cd0d1a (ubuntu-ec2)에 연결

EC2 인스턴스 연결	Session Manager	SSH 클라이언트	EC2 직렬 콘솔
-------------	-----------------	------------------	-----------

인스턴스 ID
i-0187778e998cd0d1a (ubuntu-ec2)

1. SSH 클라이언트를 엽니다.
2. 프라이빗 키 파일을 찾습니다. 이 인스턴스를 시작하는 데 사용되는 키는 ubuntu-ec2-key.pem입니다.
3. 필요한 경우 이 명령을 실행하여 키를 공개적으로 볼 수 없도록 합니다.
chmod 400 ubuntu-ec2-key.pem
4. 퍼블릭 DNS을(를) 사용하여 인스턴스에 연결:
ec2-13-209-70-29.ap-northeast-2.compute.amazonaws.com

예:

```
ssh -i "ubuntu-ec2-key.pem" ubuntu@ec2-13-209-70-29.ap-northeast-2.compute.amazonaws.com
```

참고: 대부분의 경우 추정된 사용자 이름은 정확합니다. 하지만 AMI 사용 지침을 읽고 AMI 소유자가 기본 AMI 사용자 이름을 변경했는지 확인하십시오.

```
1 al-ec2 2 ubuntu-ec2 +
Memory usage: 20% IPv4 address for eth0: 10.0.24.144
Swap usage: 0%

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

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.

/usr/bin/xauth: file /home/ubuntu/.Xauthority does not exist
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-24-144:~$
```

13. **EFS** 시스템을 사용하기 위해 **보안그룹의 인바운드 규칙**을 추가해야 한다. 아래와 같이 **[규칙 추가]**를 클릭하여 설정하고 **[규칙 저장]**을 클릭하여 새 **인바운드 규칙**을 추가한다.

A. **[유형]** : NFS

B. **[프로토콜]** : TCP

C. **[포트 범위]** : 2049

D. **[소스]** : Anywhere-IPv4

VPC > 보안 그룹 > sg-03db0c341dfbeb965 - lab-sg > 인바운드 규칙 편집

인바운드 규칙 편집 정보

인바운드 규칙은 인스턴스에 도달하도록 허용된 수신 트래픽을 제어합니다.

인바운드 규칙 정보

보안 그룹 규칙 ID	유형 정보	프로토콜 정보	포트 범위 정보	소스 정보	설명 - 선택 사항 정보	삭제
sg-044ef6c82649a8531	SSH	TCP	22	사용자 지정	0.0.0.0/0	삭제
sg-0a95dfe6360447d26	SSH	TCP	22	사용자 지정	0.0.0.0/0	삭제
-	NFS	TCP	2049	Anywhere-I...	0.0.0.0/0	삭제

규칙 추가

취소 변경 사항 미리 보기 **규칙 저장**

14. 위에서 생성한 **EFS** 페이지로 이동한다. 우측 상단의 **[연결]**을 클릭한다.

Amazon EFS > 파일 시스템 > fs-0836bc1abb7a14a8a

lab-filesystem (fs-0836bc1abb7a14a8a)

삭제 **연결**

편집

일반

성능 모드
범용

처리량 모드
버스트

수명 주기 관리
IA로 전환: 마지막 액세스 이후 30일 경과
IA 외부로 전환: 처음 액세스할 때

가용 영역
리전

자동 백업
활성화됨

암호화됨
d79c25d2-7774-4736-9657-b3a16fa15cad (aws/elasticfilesystem)

파일 시스템 상태
사용 가능

DNS 이름
fs-0836bc1abb7a14a8a.efs.ap-northeast-2.amazonaws.com

측정 크기

모니터링 | 태그 | 파일 시스템 정책 | 액세스 포인트 | 네트워크 | 복제

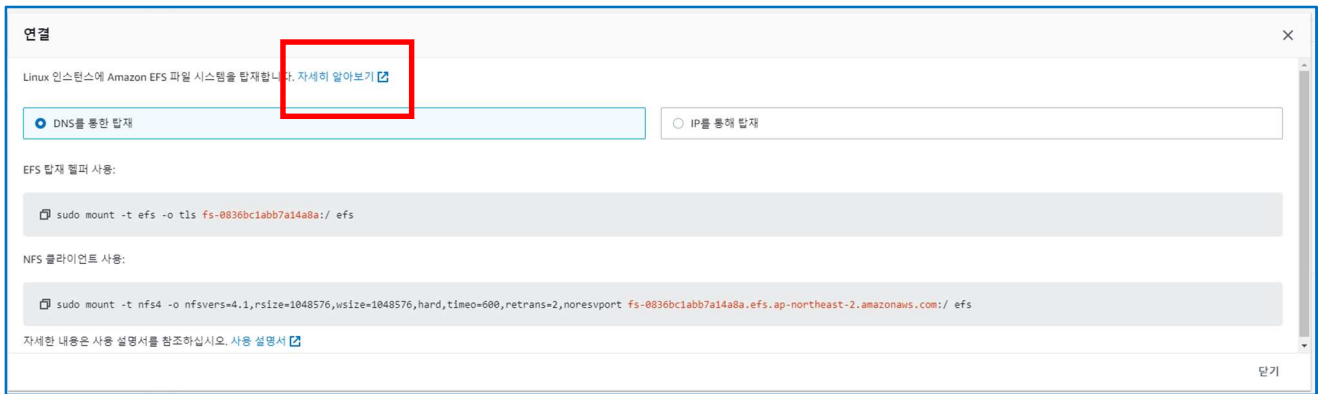
측정 크기

총 크기
6.00 KiB

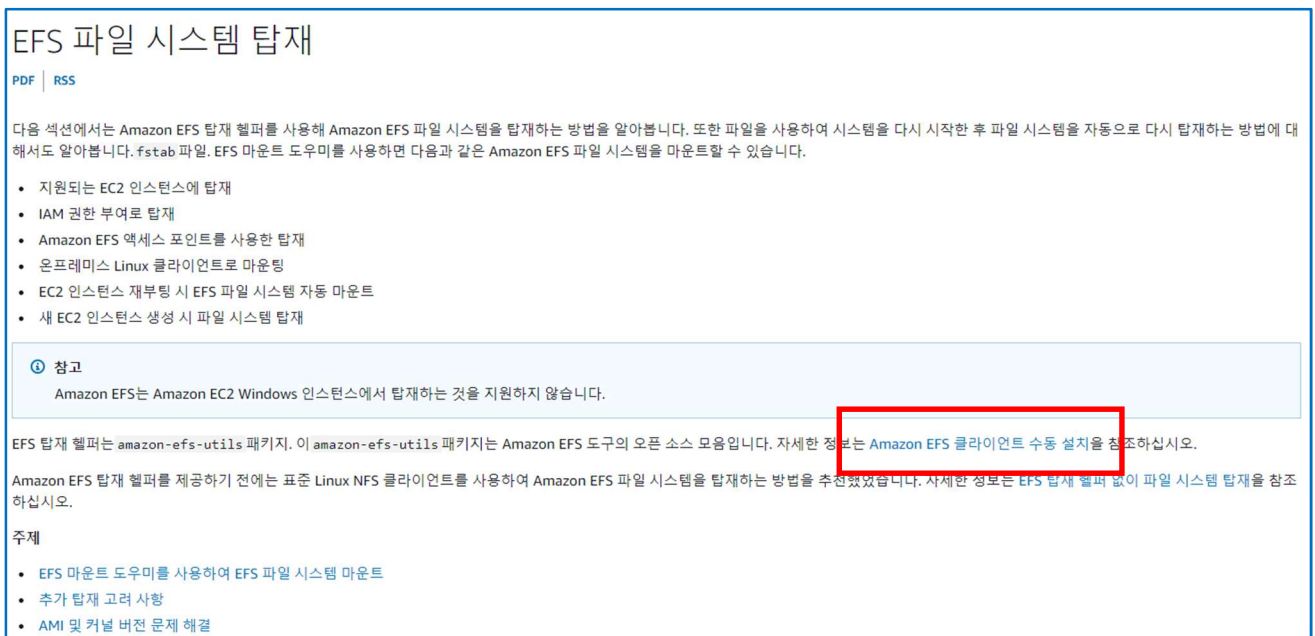
Standard/One Zone의 크기
6.00 KiB (100%)

Standard-IA/One Zone-IA의 크기

15. [연결] 페이지이다. [자세히 알아보기] 링크를 클릭해보자.



16. [EFS 파일 시스템 탑재] 페이지(https://docs.aws.amazon.com/ko_kr/efs/latest/ug/mounting-fs.html)로 이동한다. EFS 탑재 헬퍼를 설치하기 위해 [Amazon EFS 클라이언트 수동 설치] 링크를 클릭한다.



17. [Amazon EFS 클라이언트 수동 설치] 페이지(https://docs.aws.amazon.com/ko_kr/efs/latest/ug/installing-amazon-efs-utils.html)로 이동했다. 만일 EC2 인스턴스의 OS가 Amazon Linux이면 [Amazon Linux 및 Amazon Linux 2에 Amazon EFS 클라이언트 설치] 섹션에서 다음의 명령을 복사한다.

Amazon EFS 클라이언트 수동 설치

PDF | RSS

Amazon Linux와 Amazon Linux 2를 실행하는 Amazon EC2 리눅스 인스턴스와 macOS Big Sur를 실행하는 EC2 Mac 인스턴스 및 지원되는 기타 Linux 배포에 Amazon EFS 클라이언트를 수동으로 설치할 수 있습니다. 설치 절차에 대해서는 다음 섹션에서 설명합니다.

주제

- Amazon Linux 및 Amazon Linux 2에 Amazon EFS 클라이언트 설치
- 다른 Linux 배포판에서 Amazon EFS 클라이언트 설치
- macOS Big Sur를 실행하는 EC2 Mac 인스턴스에 Amazon EFS 클라이언트 설치

Amazon Linux 및 Amazon Linux 2에 Amazon EFS 클라이언트 설치

이 `amazon-efs-utils` 패키지는 Amazon Linux 및 Amazon Linux AMI 패키지 리포지토리에서 사용할 수 있으며, 이러한 AMI를 실행하는 EC2 인스턴스에 설치할 수 있습니다. 다른 Linux 배포에 Amazon EFS 클라이언트를 설치하려면 단원을 참조하십시오. [다른 Linux 배포판에서 Amazon EFS 클라이언트 설치](#).

Amazon Linux 및 Amazon Linux 2에 Amazon EFS 클라이언트 설치

이 `amazon-efs-utils` 패키지는 Amazon Linux 및 Amazon Linux AMI 패키지 리포지토리에서 사용할 수 있으며, 이러한 AMI를 실행하는 EC2 인스턴스에 설치할 수 있습니다. 다른 Linux 배포에 Amazon EFS 클라이언트를 설치하려면 단원을 참조하십시오. [다른 Linux 배포판에서 Amazon EFS 클라이언트 설치](#).

참고

AWS Direct Connect를 사용하고 있다면 [연습: 다음을 사용하여 온프레미스에서 파일 시스템 생성 및 마운트](#) AWS Direct Connect 및 VPN에서 설치 지침을 찾을 수 있습니다.

를 설치하려면 `amazon-efs-utils` Amazon Linux 2 및 Amazon Linux의 패키지

1. Amazon Linux 또는 Amazon Linux 2 EC2 인스턴스가 생성되어 있어야 합니다. 이렇게 하는 방법에 대한 정보는 단원을 참조하십시오. [단계 1: 인스턴스 시작의 Linux 인스턴스용 Amazon EC2 사용 설명서](#)
2. SSH를 통해 인스턴스 터미널에 액세스하고, 적절한 사용자 이름으로 로그인합니다. 이렇게 하는 방법에 대한 자세한 내용은 단원을 참조하십시오. [SSH를 사용하여 Linux 인스턴스에 연결의 Linux 인스턴스용 Amazon EC2 사용 설명서](#)
3. 다음 명령을 실행하여 `amazon-efs-utils` 패키지.

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



sudo yum install -y amazon-efs-utils

18. SSH 클라이언트 툴에서 **al-ec2** 창에서 위에서 복사한 명령어를 붙여넣기 해서 실행한다.

```
1 al-ec2 x 2 ubuntu-ec2 x +
[ec2-user@ip-10-0-7-49 ~]$
[ec2-user@ip-10-0-7-49 ~]$ sudo yum install -y amazon-efs-utils

Installed size: 500 k
Downloading packages:
(1/2): amazon-efs-utils-1.32.1-2.amzn2.noarch.rpm | 50 kB 00:00:00
(2/2): stunnel-4.56-6.amzn2.0.3.x86_64.rpm | 149 kB 00:00:00
-----
Total | 1.1 MB/s | 199 kB 00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : stunnel-4.56-6.amzn2.0.3.x86_64 1/2
  Installing : amazon-efs-utils-1.32.1-2.amzn2.noarch 2/2
  Verifying : amazon-efs-utils-1.32.1-2.amzn2.noarch 1/2
  Verifying : stunnel-4.56-6.amzn2.0.3.x86_64 2/2

Installed:
  amazon-efs-utils.noarch 0:1.32.1-2.amzn2

Dependency Installed:
  stunnel.x86_64 0:4.56-6.amzn2.0.3

Complete!
[ec2-user@ip-10-0-7-49 ~]$
```

19. 다시 [Amazon EFS 클라이언트 수동 설치] 페이지(https://docs.aws.amazon.com/ko_kr/efs/latest/ug/installing-amazon-efs-utils.html)로 가서, 만일 EC2 인스턴스의 OS가 Linux의 다른 배포판이라면, 특별히 이번 실습에서는 Ubuntu 배포판을 ubuntu-ec2에 설치했기 때문에 페이지를 스크롤다운하여 [다른 Linux 배포판에서 Amazon EFS 클라이언트 설치] 섹션으로 이동하여 목록 중에서 [빌드 및 설치 amazon-efs-utils 우분투와 데비안을 위한 데비안 패키지]로 이동한다.

다른 Linux 배포판에서 Amazon EFS 클라이언트 설치

을 (를) 열고 싶지 않다면 amazon-efs-utils 아마존 리눅스 또는 아마존 리눅스 2 AMI의 패키지 amazon-efs-utils 패키지는 GitHub에서도 사용할 수 있습니다.

패키지를 복제한 후 빌드 및 설치할 수 있습니다. amazon-efs-utils Linux 배포판이 지원하는 패키지 유형에 따라 다음 방법 중 하나를 사용합니다.

- **RPM**— Amazon Linux Red Hat Linux CentOS 및 유사한 시스템이 지원하는 패키지 유형입니다.
- **DEB**— Ubuntu, Debian 및 유사한 시스템이 지원하는 패키지 유형입니다.

▶ 빌드 및 설치 amazon-efs-utils RPM 패키지 (OpenSUSE 또는 SLES 이외의 배포판용)

▶ 빌드 및 설치 amazon-efs-utils OpenSUSE 및 SLES를 위한 RPM 패키지로

▼ 빌드 및 설치 amazon-efs-utils 우분투와 데비안을 위한 데비안 패키지로

20. 해당 페이지의 순서대로 차례로 실행한다.

```
$ sudo apt-get update
```

```
$ sudo apt-get -y install git binutils
```

```
$ git clone https://github.com/aws/efs-utils
```

```
$ cd efs-utils
```

```
$ ./build-deb.sh
```

```
$ sudo apt-get -y install ./build/amazon-efs-utils*deb
```

```
Setting up libnfsidmap2:amd64 (0.25-5.1ubuntu1) ...
Setting up libtirpc3:amd64 (1.2.5-1) ...
Setting up rpcbind (1.2.5-8) ...
Created symlink /etc/systemd/system/multi-user.target.wants/rpcbind.service → /lib/systemd/system/rpcbind.service.
Created symlink /etc/systemd/system/sockets.target.wants/rpcbind.socket → /lib/systemd/system/rpcbind.socket.
Setting up nfs-common (1:1.3.4-2.5ubuntu3.4) ...

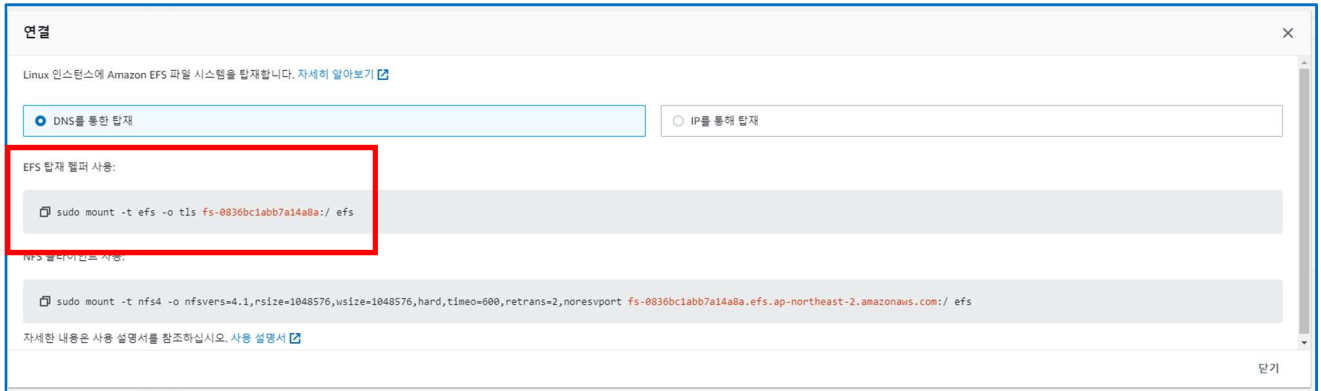
Creating config file /etc/idmapd.conf with new version
Adding system user `statd' (UID 115) ...
Adding new user `statd' (UID 115) with group `nogroup' ...
Not creating home directory `/var/lib/nfs'.
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-client.target → /lib/systemd/system/nfs-client.target.
Created symlink /etc/systemd/system/remote-fs.target.wants/nfs-client.target → /lib/systemd/system/nfs-client.target.
nfs-utils.service is a disabled or a static unit, not starting it.
Setting up amazon-efs-utils (1.32.2) ...
Processing triggers for systemd (245.4-4ubuntu3.16) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.7) ...
ubuntu@ip-10-0-24-144:~/efs-utils$
```

21. al-ec2 명령창에서 다음의 명령으로 **efs** 디렉토리를 생성한다.

```
$ mkdir efs
```

```
[ec2-user@ip-10-0-7-49 ~]$
[ec2-user@ip-10-0-7-49 ~]$ mkdir efs
[ec2-user@ip-10-0-7-49 ~]$ ls
efs
[ec2-user@ip-10-0-7-49 ~]$
```

22. 위 15번의 [연결] 페이지에서 [EFS 탑재 헬퍼 사용]의 부분을 복사한 다음, al-ec2 명령창에서 붙여넣기 하여 실행한다.



```
$ sudo mount -t efs -o tls fs-0836bc1abb7a14a8a:/ efs
```

```
[ec2-user@ip-10-0-7-49 ~]$ sudo mount -t efs -o tls fs-0836bc1abb7a14a8a:/ efs
[ec2-user@ip-10-0-7-49 ~]$
```

23. 다음의 명령으로 확인할 수 있다.

```
$ sudo df -h
```

```
[ec2-user@ip-10-0-7-49 ~]$
[ec2-user@ip-10-0-7-49 ~]$ sudo df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        474M   0    474M   0% /dev
tmpfs           483M   0    483M   0% /dev/shm
tmpfs           483M 520K   482M   1% /run
tmpfs           483M   0    483M   0% /sys/fs/cgroup
/dev/xvda1      30G   1.6G   29G    6% /
tmpfs           97M   0     97M   0% /run/user/0
tmpfs           97M   0     97M   0% /run/user/1000
127.0.0.1:/      8.0E   0    8.0E   0% /home/ec2-user/efs
[ec2-user@ip-10-0-7-49 ~]$
```

24. 또 다른 EC2 인스턴스인 **ubuntu-ec2**에서도 동일한 작업을 수행한다.

```
$ cd ~
```

```
$ mkdir efs
```

```
$ ls
```

```
$ sudo mount -t efs -o tls fs-0836bc1abb7a14a8a:/ efs
```

```
ubuntu@ip-10-0-24-144:~/efs-utils$  
ubuntu@ip-10-0-24-144:~/efs-utils$ cd ~  
ubuntu@ip-10-0-24-144:~$  
ubuntu@ip-10-0-24-144:~$  
ubuntu@ip-10-0-24-144:~$ mkdir efs  
ubuntu@ip-10-0-24-144:~$ ls  
efs  efs-utils  
ubuntu@ip-10-0-24-144:~$ sudo mount -t efs -o tls fs-0836bc1abb7a14a8a:/ efs  
ubuntu@ip-10-0-24-144:~$
```

25. 역시 **ubuntu-ec2**에서도 마운트 정보를 확인하자.

```
$ sudo df -h
```

```
ubuntu@ip-10-0-24-144:~$  
ubuntu@ip-10-0-24-144:~$ sudo df -h  
Filesystem      Size  Used Avail Use% Mounted on  
/dev/root        30G   1.7G   28G    6% /  
devtmpfs         478M    0   478M    0% /dev  
tmpfs            484M    0   484M    0% /dev/shm  
tmpfs            97M   888K    96M    1% /run  
tmpfs            5.0M    0    5.0M    0% /run/lock  
tmpfs            484M    0   484M    0% /sys/fs/cgroup  
/dev/loop2       62M    62M    0 100% /snap/core20/1405  
/dev/loop0       27M    27M    0 100% /snap/amazon-ssm-agent/5163  
/dev/loop3       68M    68M    0 100% /snap/lxd/22753  
/dev/loop1       56M    56M    0 100% /snap/core18/2344  
/dev/loop4       44M    44M    0 100% /snap/snapd/15177  
tmpfs            97M    0    97M    0% /run/user/1000  
127.0.0.1:/      8.0E    0    8.0E    0% /home/ubuntu/efs  
ubuntu@ip-10-0-24-144:~$
```

26. EFS 테스트를 위해 **ubuntu-ec2** 머신에서 다음과 같이 간단한 파일 한 개를 생성하고 간단한 메시지를 입력했다.

Hello, Amazon EFS^^

```
1 ubuntu-ec2 × +
ubuntu@ip-10-0-24-144:~$ cd efs
ubuntu@ip-10-0-24-144:~/efs$ sudo nano efstest.txt
ubuntu@ip-10-0-24-144:~/efs$ cat efstest.txt
Hello, Amazon EFS^^
ubuntu@ip-10-0-24-144:~/efs$
```

27. **al-ec2**에서 확인해 보자.

```
[ec2-user@ip-10-0-7-49 ~]$
[ec2-user@ip-10-0-7-49 ~]$ cd efs
[ec2-user@ip-10-0-7-49 efs]$ ls
efstest.txt
[ec2-user@ip-10-0-7-49 efs]$ cat efstest.txt
Hello, Amazon EFS^^
[ec2-user@ip-10-0-7-49 efs]$
```

28. 결국 2개의 EC2 인스턴스가 NFS를 통해 같은 EFS를 사용한다는 것을 확인할 수 있다.