

Lab2. Using AWS EBS

목적

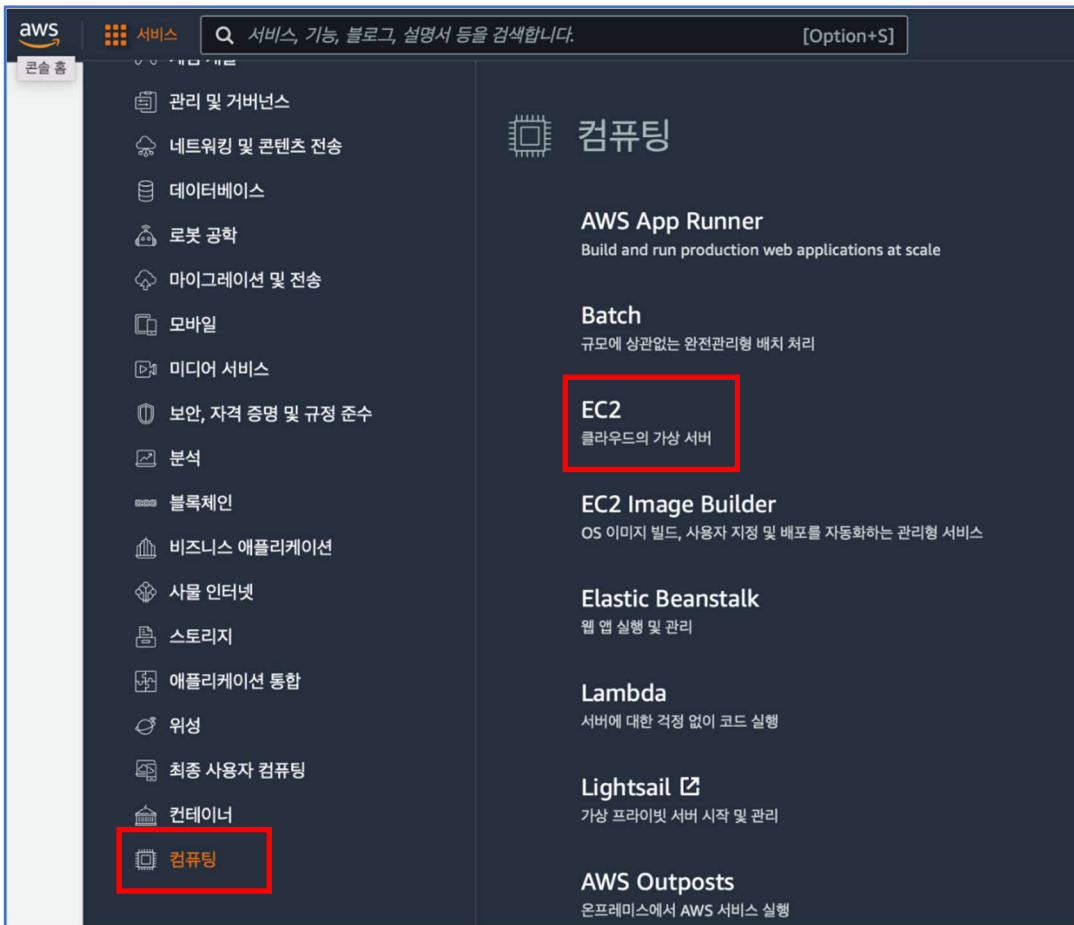
Amazon EC2에서 쉽고 고성능의 블록 스토리지 서비스인 EBS를 사용하는 방법에 대해 알아보자. 이 학습은 AWS Free-Tier를 활용하여 진행한다.

사전 준비물

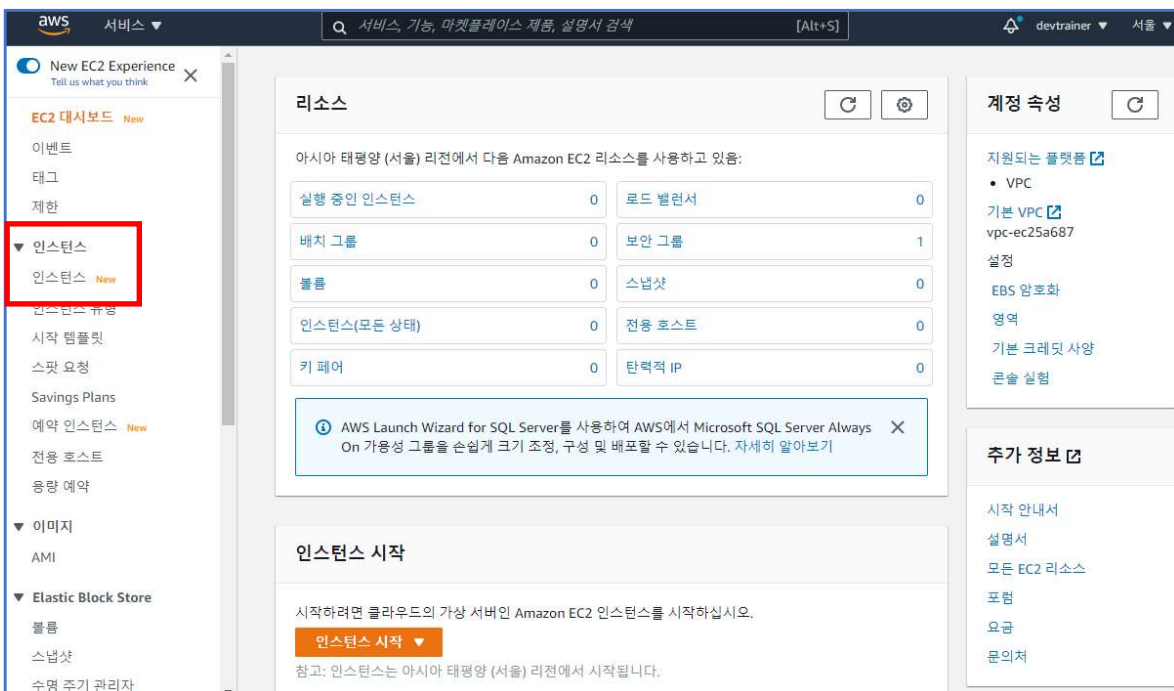
AWS Free-Tier 계정

Linux 서버 EC2 인스턴스 생성하기

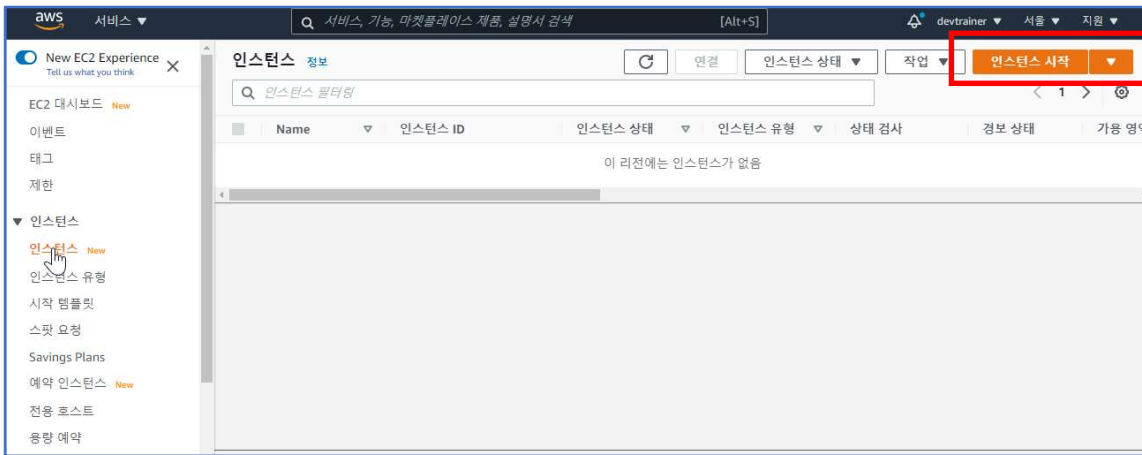
1. EC2 생성을 위해 좌측 상단의 [서비스] > [컴퓨팅] > [EC2]를 클릭하여 해당 페이지로 이동한다.



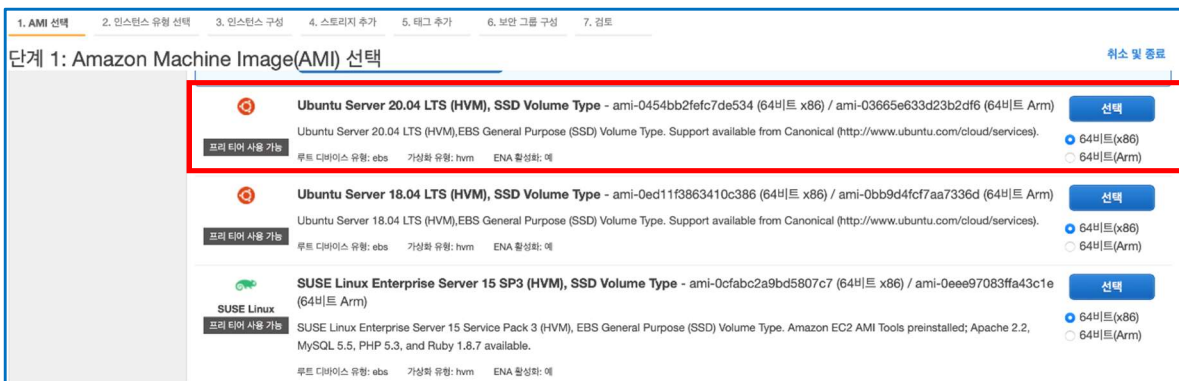
2. 왼쪽 항목에서 [인스턴스]를 선택하여 해당 페이지로 이동한다.



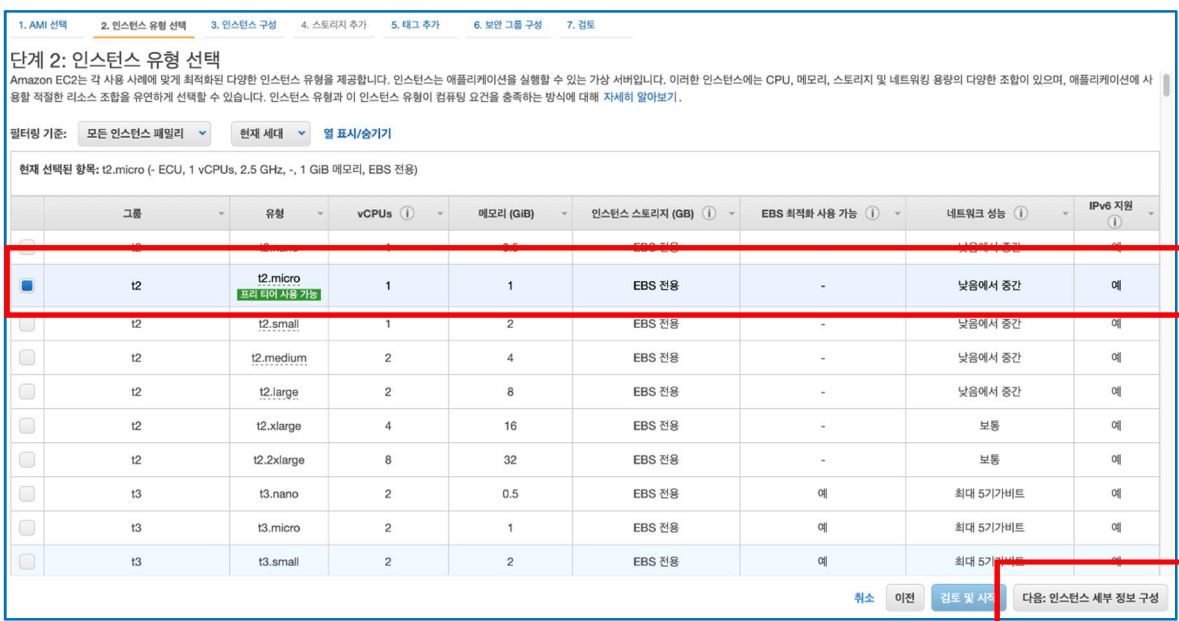
3. 우측 상단의 [인스턴스 시작] 오렌지 색 버튼을 클릭한다.



4. [단계 1: Amazon Machine Image(AMI) 선택] 페이지에서 [Ubuntu Server 20.04 LTS(HVM), SSD Volume Type] 서버를 찾은 후 [64비트(x86)]이 선택되어 있는 것을 확인한 후 [선택] 버튼을 클릭한다.



5. [단계 2:인스턴스 유형 선택] 페이지에서, [t2.micro]를 선택 후, [다음:인스턴스 세부 정보 구성] 버튼을 클릭한다.



6. [단계 3:인스턴스 세부 정보 구성] 페이지에서, [네트워크]는 이전 Lab에서 생성한 **lab-vpc**, [서브넷]은 **public subnet**, [퍼블릭 IP 자동 할당]은 **활성화**를 선택하고, [다음:스토리지 추가] 버튼을 클릭한다.

단계 3: 인스턴스 세부 정보 구성

기본 VPC 없음. 다른 VPC 또는 새 기본 VPC 생성을(를) 선택합니다.

요구 사항에 적합하게 인스턴스를 구성합니다. 동일한 AMI의 여러 인스턴스를 시작하고 스팟 인스턴스를 요청하여 보다 저렴한 요금을 활용하며 인스턴스에 액세스 관리를 담당하는 등 다양한 기능을 사용할 수 있습니다.

인스턴스 개수 1 Auto Scaling 그룹 시작

구매 옵션 ☐ 스팟 인스턴스 요청

네트워크 새 VPC 생성
기본 VPC가 없습니다. 새 기본 VPC 생성.

서브넷 새 서브넷 생성
4090개 IP 주소 사용 가능

퍼블릭 IP 자동 할당

호스트 이름 유형

DNS Hostname ☒ 리소스 기반 IPv4(A 레코드) DNS 요청 활성화
☐ 리소스 기반 IPv6(AAAA 레코드) DNS 요청 활성화

배치 그룹 ☐ 배치 그룹에 인스턴스 추가

용량 예약

취소 이전 검토 및 저장 다음: 스토리지 추가

7. [단계 4:스토리지 추가] 페이지에서, Linux Server는 스토리지 크기가 **8GiB**로 맞춰져 있다. [볼륨 유형]은 “**마크 네틱(standard)**”로, 설정한다. 그리고 [새 볼륨 추가] 버튼을 클릭하여 다음과 같이 설정한 후, [다음:태그 추가] 버튼을 클릭한다.

- A. [볼륨 유형] : EBS
- B. [디바이스] : /dev/sdb
- C. [크기] : 2GiB
- D. [볼륨 유형] : 범용 SSD(gp2)
- E. [종료 시 삭제] : 체크
- F. 나머지 값은 기본값 그대로

단계 4: 스토리지 추가

인스턴스가 다음 스토리지 디바이스 설정으로 시작됩니다. 추가 EBS 볼륨 및 인스턴스 스토어 볼륨을 인스턴스에 연결하거나 루트 볼륨의 설정을 편집할 수 있습니다. 인스턴스를 시작할 후 추가 EBS 볼륨을 연결할 수도 있지만, 인스턴스 스토어 볼륨은 연결할 수 없습니다. Amazon EC2의 스토리지 옵션에 대해 자세히 알아보십시오.

볼륨 유형	디바이스	스냅샷	크기(GiB)	볼륨 유형	IOPS	처리량(MB/초)	종료 시 삭제	암호화
루트	/dev/sda1	snap-0fdb1de3deef1ceb0	8	마크 네틱(standard)	해당 사항 없음	해당 사항 없음	<input checked="" type="checkbox"/>	암호화되지 않음
EBS	/dev/sdb	검색(대소문자 구분)	2	범용 SSD(gp2)	100/3000	해당 사항 없음	<input checked="" type="checkbox"/>	암호화되지 않음

새 볼륨 추가

8. [태그 추가] 버튼을 누른다.

1. AMI 선택 2. 인스턴스 유형 선택 3. 인스턴스 구성 4. 스토리지 추가 5. 태그 추가 6. 보안 그룹 구성 7. 검토

단계 5: 태그 추가

태그는 대소문자를 구별하는 키-값 페어로 이루어져 있습니다. 예를 들어 키가 Name이고 값이 Webserver인 태그를 정의할 수 있습니다.
태그 복사본은 볼륨, 인스턴스 또는 둘 다에 적용될 수 있습니다.
태그는 모든 인스턴스 및 볼륨에 적용됩니다. Amazon EC2 리소스 태그 지정에 대해 자세히 알아보기.

키 (최대 128자)	값 (최대 256자)	인스턴스 ⓘ	볼륨 ⓘ	네트워크 인터페이스 ⓘ
이 리소스에는 현재 태그가 없습니다.				
[태그 추가] 버튼 또는 Name 태그를 추가하려면 클릭합니다. 윗(↑) 선택합니다. IAM 정책에 태그를 생성할 수 있는 권한이 포함되어 있는지 확인합니다.				

태그 추가 (최대 50개 태그)

9. [키]에 "Name"을, [값]에 "lab2-ubuntu-ec2"를 입력한 다음, [다음:보안 그룹 구성] 버튼을 클릭한다. 태그는 해당 인스턴스를 표현하는 여러 이름으로 사용될 수 있다. EC2의 이름을 붙인다고 생각하고 넣으면 된다. 여러 인스턴스가 있을 경우 이를 태그별로 구분하면 검색이나 그룹 짓기 편하므로 여기서 본인 서비스의 인스턴스를 나타낼 수 있는 값으로 등록하면 된다.

1. AMI 선택 2. 인스턴스 유형 선택 3. 인스턴스 구성 4. 스토리지 추가 5. 태그 추가 6. 보안 그룹 구성 7. 검토

단계 5: 태그 추가

태그는 대소문자를 구별하는 키-값 페어로 이루어져 있습니다. 예를 들어 키가 Name이고 값이 Webserver인 태그를 정의할 수 있습니다.
태그 복사본은 볼륨, 인스턴스 또는 둘 다에 적용될 수 있습니다.
태그는 모든 인스턴스 및 볼륨에 적용됩니다. Amazon EC2 리소스 태그 지정에 대해 자세히 알아보기.

키 (최대 128자)	값 (최대 256자)	인스턴스 ⓘ	볼륨 ⓘ	네트워크 인터페이스 ⓘ
Name	lab2-ubuntu-ec2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

다른 태그 추가 (최대 50개 태그)

취소 이전 검토 및 시작 **다음: 보안 그룹 구성**

10. [단계 6:보안 그룹 구성] 페이지에서, 기본 보안 그룹 설정 정보 확인 후, [검토 및 시작] 버튼을 클릭한다.

1. AMI 선택 2. 인스턴스 유형 선택 3. 인스턴스 구성 4. 스토리지 추가 5. 태그 추가 6. 보안 그룹 구성 7. 검토

단계 6: 보안 그룹 구성

보안 그룹은 인스턴스에 대한 트래픽을 제어하는 방화벽 규칙 세트입니다. 이 페이지에서는 특정 트래픽을 인스턴스에 도달하도록 허용할 규칙을 추가할 수 있습니다. 예를 들면 웹 서버를 설정하여 인터넷 트래픽을 인스턴스에 도달하도록 허용하려는 경우 HTTP 및 HTTPS 트래픽에 대한 무제한 액세스를 허용하는 규칙을 추가합니다. 새 보안 그룹을 생성하거나 아래에 나와 있는 기존 보안 그룹 중에서 선택할 수 있습니다. Amazon EC2 보안 그룹에 대해 자세히 알아보기.

보안 그룹 할당: ☒ 새 보안 그룹 생성
☐ 기존 보안 그룹 선택

보안 그룹 이름:
설명:

유형	프로토콜	포트 범위	소스	설명
SSH	TCP	22	사용자 지정 0.0.0.0/0	예: SSH for Admin Desktop

규칙 추가

경고

소스가 0.0.0.0/0인 규칙은 모든 IP 주소에서 인스턴스에 액세스하도록 허용합니다. 알려진 IP 주소의 액세스만 허용하도록 보안 그룹을 설정하는 것이 좋습니다.

취소 이전 **검토 및 시작**

11. [범용(SSD)에서 부팅]창이 나타난다. Boot Device를 마그네틱으로 설정하면 나타나는 창이다. 목록에서 [마그네틱을 이 인스턴스의 부트 볼륨으로 계속 사용]을 선택하고 [다음] 버튼을 클릭한다.

범용(SSD)에서 부팅

범용(SSD) 볼륨은 대부분의 애플리케이션에 필요한 성능을 충족할 수 있도록 볼륨 크기에 상관없이 볼륨당 3000 IOPS의 순간 최대 성능과, GiB당 3 IOPS의 일관된 기본 성능을 제공합니다.

☐ 범용(SSD)을 콘솔에서 시작하는 모든 인스턴스의 기본 부트 볼륨으로 설정(권장)
☐ 범용(SSD)을 이 인스턴스에 대한 부트 볼륨으로 설정
☒ 마그네틱을 이 인스턴스의 부트 볼륨으로 계속 사용

프리 티어 사용 가능 고객은 최대 30GB의 범용(SSD) 스토리지를 사용할 수 있습니다.

☐ 다시 표시 안 함

다음

12. [단계 7:인스턴스 시작 검토] 페이지에서, 지금까지 구성된 정보를 확인 한 다음, 수정 및 변경사항이 없다면 [시작하기] 버튼을 클릭한다.

1. AMI 선택 2. 인스턴스 유형 선택 3. 인스턴스 구성 4. 스토리지 추가 5. 태그 추가 6. 보안 그룹 구성 7. 검토

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-1이(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 [AMI 편집](#)

Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-0454bb2fec7de534
프리 티어 사용 가능 Ubuntu Server 20.04 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

인스턴스 유형 [인스턴스 유형 편집](#)

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

보안 그룹 [보안 그룹 편집](#)

보안 그룹 이름 launch-wizard-1

[취소](#) [이전](#) [시작하기](#)

13. [기존 키 페어 선택 또는 새 키 페어 생성] 페이지가 나타난다.

기존 키 페어 선택 또는 새 키 페어 생성

키 페어는 AWS에 저장하는 퍼블릭 키와 사용자가 저장하는 프라이빗 키 파일로 구성됩니다. 이 둘을 모두 사용하여 SSH를 통해 인스턴스에 안전하게 접속할 수 있습니다. Windows AMI의 경우 인스턴스에 로그인하는 데 사용되는 암호를 얻으려면 프라이빗 키 파일이 필요합니다. Linux AMI의 경우, 프라이빗 키 파일을 사용하면 인스턴스에 안전하게 SSH로 연결할 수 있습니다. Amazon EC2는 ED25519 및 RSA 키 페어 유형을 지원합니다.

참고: 선택한 키 페어가 이 인스턴스에 대해 승인된 키 세트에 추가됩니다. [퍼블릭 AMI에서 기존 키 페어 제거](#)에 대해 자세히 알아보십시오.

☒ 기존 키 페어 선택

☐ 새 키 페어 생성

☐ 키 페어 없이 계속

☐ 해당 프라이빗 키 파일에 액세스할 수 있으며 이 파일이 없으면 내 인스턴스에 로그인할 수 없음을 확인합니다.

[취소](#) [인스턴스 시작](#)

14. [기존 키 페어 선택] 드롭다운 버튼을 클릭하면 보이는 3개의 항목 중에 “새 키 페어 생성”을 선택하고, [키 페어 이름]에 “lab2-ubuntu-ec2-key”를 입력 후 [키 페어 다운로드] 버튼을 클릭하여 “lab2-ubuntu-ec2-key.pem” 파일을 로컬 컴퓨터에 보관한다. 이 파일이 없으면 EC2에 접근할 수 없기 때문에 잘 보관해야 한다.

기존 키 페어 선택 또는 새 키 페어 생성

키 페어는 AWS에 저장하는 퍼블릭 키와 사용자가 저장하는 프라이빗 키 파일로 구성됩니다. 이 둘을 모두 사용하여 SSH를 통해 인스턴스에 안전하게 접속할 수 있습니다. Windows AMI의 경우 인스턴스에 로그인하는 데 사용되는 암호를 얻으려면 프라이빗 키 파일이 필요합니다. Linux AMI의 경우, 프라이빗 키 파일을 사용하면 인스턴스에 안전하게 SSH로 연결할 수 있습니다. Amazon EC2는 ED25519 및 RSA 키 페어 유형을 지원합니다.

참고: 선택한 키 페어가 이 인스턴스에 대해 승인된 키 세트에 추가됩니다. 퍼블릭 AMI에서 기존 키 페어 제거에 대해 자세히 알아보십시오.

새 키 페어 생성

키 페어 유형

☒ RSA ☐ ED25519

키 페어 이름

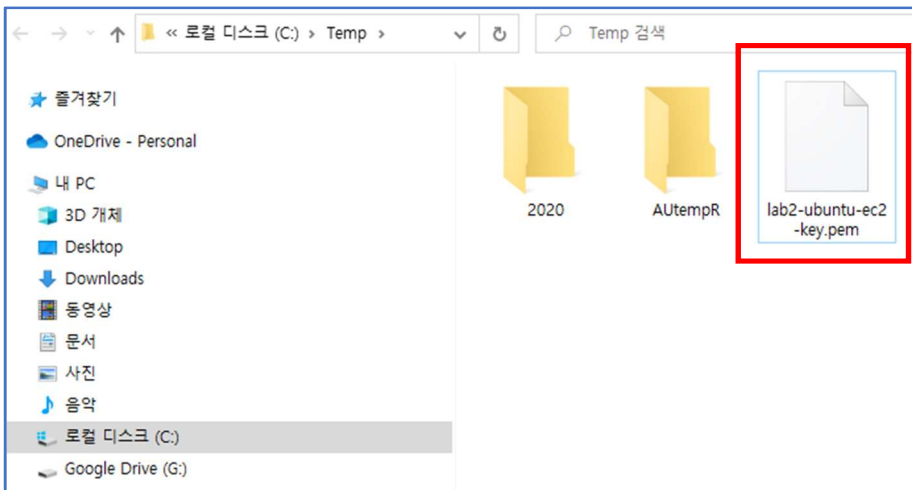
lab2-ubuntu-ec2-key

키 페어 다운로드

계속하려면 먼저 프라이빗 키 파일(*.pem 파일)을 다운로드해야 합니다. 액세스할 수 있는 안전한 위치에 저장합니다. 파일은 생성되고 나면 다시 다운로드할 수 없습니다.

취소

인스턴스 시작



15. 키 페어 다운로드 완료 후 [인스턴스 시작] 버튼을 클릭한다.

기존 키 페어 선택 또는 새 키 페어 생성

키 페어는 AWS에 저장하는 퍼블릭 키와 사용자가 저장하는 프라이빗 키 파일로 구성됩니다. 이 둘을 모두 사용하여 SSH를 통해 인스턴스에 안전하게 접속할 수 있습니다. Windows AMI의 경우 인스턴스에 로그인하는 데 사용되는 암호를 얻으려면 프라이빗 키 파일이 필요합니다. Linux AMI의 경우, 프라이빗 키 파일을 사용하면 인스턴스에 안전하게 SSH로 연결할 수 있습니다. Amazon EC2는 ED25519 및 RSA 키 페어 유형을 지원합니다.

참고: 선택한 키 페어가 이 인스턴스에 대해 승인된 키 세트에 추가됩니다. 퍼블릭 AMI에서 기존 키 페어 제거에 대해 자세히 알아보십시오.

새 키 페어 생성

키 페어 유형

☒ RSA
 ☐ ED25519

키 페어 이름

lab2-ubuntu-ec2-key

키 페어 다운로드

계속하려면 먼저 프라이빗 키 파일(*.pem 파일)을 다운로드해야 합니다. 액세스할 수 있는 안전한 위치에 저장합니다. 파일은 생성되고 나면 다시 다운로드할 수 없습니다.

취소

인스턴스 시작

16. 인스턴스는 보통 5 ~ 10분 정도 시간이 걸린다. 페이지 우측 하단의 [인스턴스 보기]를 클릭한다.

시작 상태

지금 인스턴스를 시작 중입니다.

다음 인스턴스 시작이 개시됨: i-049143f1a6dd01f1a

시작 로그 보기

예상 요금 알림 받기

결제 알림 생성 AWS 결제 예상 요금이 사용자가 정의한 금액을 초과하는 경우(예를 들면 프리 티어를 초과하는 경우) 이메일 알림을 받습니다.

인스턴스에 연결하는 방법

인스턴스를 시작 중이며, 사용할 준비가 되어 실행 중 상태가 될 때까지 몇 분이 걸릴 수도 있습니다. 새 인스턴스에서는 사용 시간이 즉시 시작되어 인스턴스를 중지 또는 종료할 때까지 계속 누적입니다.

인스턴스 보기를 클릭하여 인스턴스의 상태를 모니터링합니다. 인스턴스가 실행 중 상태가 되고 나면 [인스턴스] 화면에서 인스턴스에 연결할 수 있습니다. 인스턴스에 연결하는 방법 알아보기.

다음은 시작에 도움이 되는 유용한 리소스입니다.

- Linux 인스턴스에 연결하는 방법
- Amazon EC2: 사용 설명서
- AWS 프리 티어에 대해 알아보기
- Amazon EC2: 토론 포럼

인스턴스가 시작되는 동안 다음을 수행할 수도 있습니다.

- 상태 검사 결과 생성 해당 인스턴스가 상태 검사를 통과하지 못하는 경우 알림을 받습니다. (추가 요금이 적용될 수 있음)
- 추가 EBS 볼륨 생성 및 연결 (추가 요금이 적용될 수 있음)
- 보안 그룹 관리

인스턴스 보기

17. [인스턴스] 항목을 클릭하여 해당 페이지로 이동하면 방금 우리가 생성한 인스턴스가 생성되어 [인스턴스 상태]가 "실행 중"임을 확인할 수 있다.

인스턴스 (1) 정보

연결

인스턴스 상태

작업

인스턴스 시작

검색

i-049143f1a6dd01f1a

필터 지우기

Name	인스턴스 ID	인스턴스 상태	인스턴스 유형	상태 검사	경보 상태	가용 영역	퍼블릭 IPv4 DNS
lab2-ubuntu-ec2	i-049143f1a6dd01f1a	실행 중	t2.micro	2/2개 검사 통과	경보 없음	ap-northeast-2a	ec2-3-38-210-210.ap-n...

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

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

블록 ID	디바이스 이름	블록 크기(GiB)	연결 상태	연결 시간	암호화됨	KMS 키 ID
vol-0573125e5ccbbc033	/dev/sda1	8	연결됨	Mon Mar 28 2022 13:44:27 ...	아니요	-
vol-0daf58171ff0833c7	/dev/sdb	2	연결됨	Mon Mar 28 2022 13:44:27 ...	아니요	-

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

인스턴스

인스턴스 New

인스턴스 유형

시작 템플릿

스팟 요청

Savings Plans

예약 인스턴스 New

전용 호스트

용량 예약

이미지

AMI New

AMI 카탈로그

Elastic Block Store

블록 New

스냅샷 New

수명 주기 관리자 New

블록 (2)

Q

블록 필터링

<input type="checkbox"/>	Name	블록 ID	유형	크기	IOPS	처리량	스냅샷
<input type="checkbox"/>	lab2-ubuntu-ec2	vol-0daf58171ff0833c7	gp2	2 GiB	100	-	-
<input type="checkbox"/>	lab2-ubuntu-ec2	vol-0573125e5ccbbc033	standard	8 GiB	-	-	snap-0fdb1de...

20. 방금 생성한 인스턴스를 연결해보자. 인스턴스 요약 페이지에서 **[연결]**을 클릭한다.

EC2 > 인스턴스 > i-049143f1a6dd01f1a		
i-049143f1a6dd01f1a (lab2-ubuntu-ec2)에 대한 인스턴스 요약 정보		
<div> <div></div> <div>연결</div> <div>인스턴스 상태 ▼</div> <div>작업 ▼</div> </div>		
인스턴스 ID i-049143f1a6dd01f1a (lab2-ubuntu-ec2)	퍼블릭 IPv4 주소 3.38.210.210 개방 주소법	프라이빗 IPv4 주소 10.0.2.129
IPv6 주소 -	인스턴스 상태 실행 중	퍼블릭 IPv4 DNS ec2-3-38-210-210.ap-northeast-2.compute.amazonaws.com 개방 주소법
호스트 이름 유형 IP 이름: ip-10-0-2-129.ap-northeast-2.compute.internal	프라이빗 IP DNS 이름(IPv4만 해당) ip-10-0-2-129.ap-northeast-2.compute.internal	프라이빗 리소스 DNS 이름 응답 IPv4(A)
인스턴스 유형 t2.micro	탄력적 IP 주소 -	VPC ID vpc-00420ebbb30144045 (lab-vpc)
AWS Compute Optimizer 찾기 권장 사항을 위해 AWS Compute Optimizer에 옵트인합니다. 자세히 알아보기	IAM 역할 -	서브넷 ID subnet-0f8cdbc75eeba4bbd (lab-subnet-public1-ap-northeast-2a)

21. [인스턴스에 연결] 페이지에서 [EC2 인스턴스 연결] 탭의 [연결] 오렌지색 버튼을 클릭한다.

EC2 > 인스턴스 > i-049143f1a6dd01f1a > 인스턴스에 연결

인스턴스에 연결 정보

다음 옵션 중 하나를 사용하여 인스턴스 i-049143f1a6dd01f1a (lab2-ubuntu-ec2)에 연결

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

인스턴스 ID
i-049143f1a6dd01f1a (lab2-ubuntu-ec2)

퍼블릭 IP 주소
3.38.210.210

사용자 이름

사용자 지정 사용자 이름을 사용하여 연결하거나 인스턴스 시작에 사용한 AMI의 기본 사용자 이름 ubuntu를(를) 사용합니다.

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

취소 **연결**

22. 방금 생성한 인스턴스에 잘 연결이 된 것을 확인할 수 있다.

```
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1022-aws x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Mon Mar 28 05:34:59 UTC 2022

System load:  0.0          Processes:            107
Usage of /:   18.4% of 7.69GB Users logged in:       1
Memory usage: 21%          IPv4 address for eth0: 10.0.2.129
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

Last login: Mon Mar 28 05:08:27 2022 from 13.209.1.56
ubuntu@ip-10-0-2-129:~$
```

AWS EC2의 EBS 사용하기

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

```
ubuntu@ip-10-0-2-129:~$  
ubuntu@ip-10-0-2-129:~$ df -h  
Filesystem      Size  Used Avail Use% Mounted on  
/dev/root        7.7G  1.5G   6.3G  19% /  
devtmpfs         479M    0   479M   0% /dev  
tmpfs            485M    0   485M   0% /dev/shm  
tmpfs            97M   816K   97M    1% /run  
tmpfs            5.0M    0    5.0M   0% /run/lock  
tmpfs            485M    0   485M   0% /sys/fs/cgroup  
/dev/loop1       56M   56M     0 100% /snap/core18/2253  
/dev/loop4       43M   43M     0 100% /snap/snapd/14066  
/dev/loop0       25M   25M     0 100% /snap/amazon-ssm-agent/4046  
/dev/loop2       62M   62M     0 100% /snap/core20/1242  
/dev/loop3       68M   68M     0 100% /snap/lxd/21835  
tmpfs            97M    0   97M    0% /run/user/1000  
ubuntu@ip-10-0-2-129:~$
```

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

```
ubuntu@ip-10-0-2-129:~$ lsblk  
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT  
loop0        7:0    0    25M  1 loop /snap/amazon-ssm-agent/4046  
loop1        7:1    0  55.5M  1 loop /snap/core18/2253  
loop2        7:2    0  61.9M  1 loop /snap/core20/1242  
loop3        7:3    0  67.2M  1 loop /snap/lxd/21835  
loop4        7:4    0  42.2M  1 loop /snap/snapd/14066  
xvda        202:0    0     8G   0 disk  
└─xvda1     202:1    0     8G   0 part /  
xvdb        202:16   0     2G   0 disk  
ubuntu@ip-10-0-2-129:~$
```

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

```
ubuntu@ip-10-0-2-129:~$ sudo file -s /dev/xvdb  
/dev/xvdb: data
```

4. 다음의 명령어를 사용하여 기존의 파일시스템의 정보를 확인해보자. 현재 /dev/xvda1은 파일시스템이 ext4 파일시스템임을 알 수 있다.

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

```
ubuntu@ip-10-0-2-129:~$ sudo file -s /dev/xvda1
/dev/xvda1: Linux rev 1.0 ext4 filesystem data, UUID=436cf32d-5e3d-46ca-b557-f870c8a25794, volume name "cloudimg-rootfs" (needs journal recovery) (extents) (64bit) (large files) (huge files)
ubuntu@ip-10-0-2-129:~$
```

5. 다음의 명령어를 사용하여 /dev/xvdb의 파일시스템을 ext4로 설정한다.

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

```
ubuntu@ip-10-0-2-129:~$ sudo mkfs -t ext4 /dev/xvdb
mke2fs 1.45.5 (07-Jan-2020)
Creating filesystem with 524288 4k blocks and 131072 inodes
Filesystem UUID: 3060f1ad-8580-44c4-97b9-29f77a0414e1
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

ubuntu@ip-10-0-2-129:~$
```

6. 이제 data 디렉토리에 디렉토리를 마운트해야 한다. 먼저 루트 디렉토리 밑에 data 디렉토리를 생성한다. 그리고 이 data 디렉토리를 /dev/xvdb와 마운트한다.

```
ubuntu@ip-10-0-2-129:~$ sudo mkdir /data
ubuntu@ip-10-0-2-129:~$
ubuntu@ip-10-0-2-129:~$
ubuntu@ip-10-0-2-129:~$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0        7:0      0   25M  1 loop /snap/amazon-ssm-agent/4046
loop1        7:1      0  55.5M  1 loop /snap/core18/2253
loop2        7:2      0  61.9M  1 loop /snap/core20/1242
loop3        7:3      0  67.2M  1 loop /snap/lxd/21835
loop4        7:4      0  42.2M  1 loop /snap/snapd/14066
xvda        202:0     0    8G   0 disk
└─xvda1     202:1     0    8G   0 part /
xvdb        202:16    0    2G   0 disk
ubuntu@ip-10-0-2-129:~$
ubuntu@ip-10-0-2-129:~$ sudo mount /dev/xvdb /data
ubuntu@ip-10-0-2-129:~$
```


7. 마운트 후 lsblk를 실행하면 성공적으로 마운트됐음을 확인할 수 있다.

```
ubuntu@ip-10-0-2-129:~$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0        7:0    0   25M  1 loop /snap/amazon-ssm-agent/4046
loop1        7:1    0  55.5M  1 loop /snap/core18/2253
loop2        7:2    0  61.9M  1 loop /snap/core20/1242
loop3        7:3    0  67.2M  1 loop /snap/lxd/21835
loop4        7:4    0  42.2M  1 loop /snap/snapd/14066
xvda        202:0    0    8G   0 disk
└─xvda1     202:1    0    8G   0 part /
xvdb        202:16   0    2G   0 disk /data
ubuntu@ip-10-0-2-129:~$
```

8. data 디렉토리로 이동하여 Hello.txt 파일을 생성한다.

```
ubuntu@ip-10-0-2-129:~$ cd /data
ubuntu@ip-10-0-2-129:/data$
ubuntu@ip-10-0-2-129:/data$ sudo touch Hello.txt
ubuntu@ip-10-0-2-129:/data$ ls
Hello.txt  lost+found
```

9. 또한 Nano 에디터를 사용하여 Hello.txt에 다음과 같이 Hello, Amazon EBS!!!라고 텍스트를 입력한다.

```
ubuntu@ip-10-0-2-129:/data$
ubuntu@ip-10-0-2-129:/data$ cat Hello.txt
Hello, Amazon EBS!!!
ubuntu@ip-10-0-2-129:/data$
```

10. 이번에는 매번 새로 부팅할 때마다 EBS 볼륨에 마운트하기 위해 해당 디바이스를 /etc/fstab 파일에 등록해야 한다. 먼저 원본 파일을 fstab.bak로 백업한다.

```
ubuntu@ip-10-0-2-129:/data$ ls /etc/fstab
/etc/fstab
ubuntu@ip-10-0-2-129:/data$
ubuntu@ip-10-0-2-129:/data$ sudo cp /etc/fstab /etc/fstab.bak
ubuntu@ip-10-0-2-129:/data$
ubuntu@ip-10-0-2-129:/data$
ubuntu@ip-10-0-2-129:/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
```

```
GNU nano 4.8 /etc/fstab
LABEL=cloudimg-rootfs / ext4 defaults,discard 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-10-0-2-129:/data$ sudo file -s /dev/xvdb
/dev/xvdb: Linux rev 1.0 ext4 filesystem data, UUID=3060f1ad-8580-44c4-97b9-29f77a0414e1 (needs journal recovery) (extents) (64bit) (large files) (huge files)
ubuntu@ip-10-0-2-129:/data$
```

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

```
ubuntu@ip-10-0-2-129:/data$ cd ..
ubuntu@ip-10-0-2-129:/data$ sudo unmount /data
sudo: unmount: command not found
ubuntu@ip-10-0-2-129:/data$ sudo umount /data
ubuntu@ip-10-0-2-129:/data$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0        7:0    0   25M  1 loop /snap/amazon-ssm-agent/4046
loop1        7:1    0  55.5M  1 loop /snap/core18/2253
loop2        7:2    0  61.9M  1 loop /snap/core20/1242
loop3        7:3    0  67.2M  1 loop /snap/lxd/21835
loop4        7:4    0  42.2M  1 loop /snap/snapd/14066
xvda        202:0    0    8G   0 disk
└─xvda1     202:1    0    8G   0 part /
xvdb        202:16   0    2G   0 disk
ubuntu@ip-10-0-2-129:/data$
```


14. 다음의 명령으로 작동하는 /etc/fstab 파일을 테스트해 보자.

\$ sudo mount -a

```
ubuntu@ip-10-0-2-129:/$ sudo mount -a
ubuntu@ip-10-0-2-129:/$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0        7:0      0   25M  1 loop /snap/amazon-ssm-agent/4046
loop1        7:1      0  55.5M  1 loop /snap/core18/2253
loop2        7:2      0  61.9M  1 loop /snap/core20/1242
loop3        7:3      0  67.2M  1 loop /snap/lxd/21835
loop4        7:4      0  42.2M  1 loop /snap/snapd/14066
xvda        202:0     0    8G   0 disk
└─xvda1     202:1     0    8G   0 part /
xvdb        202:16    0    2G   0 disk /data
ubuntu@ip-10-0-2-129:/$ █
```