

Lab. EC2를 이용해서 Linux Instance 서버 만들기

1. 목적

Amazon EC2(Elastic Compute Cloud)를 사용하여 Linux 인스턴스를 생성하고 접속하는 방법을 학습한다. 또한 생성된 Linux 서버의 시작, 중지 및 EC2 인스턴스에 대한 삭제 방법을 다뤄본다. 이 학습은 AWS Free-Tier를 활용하여 진행한다.

2. 사전 준비물

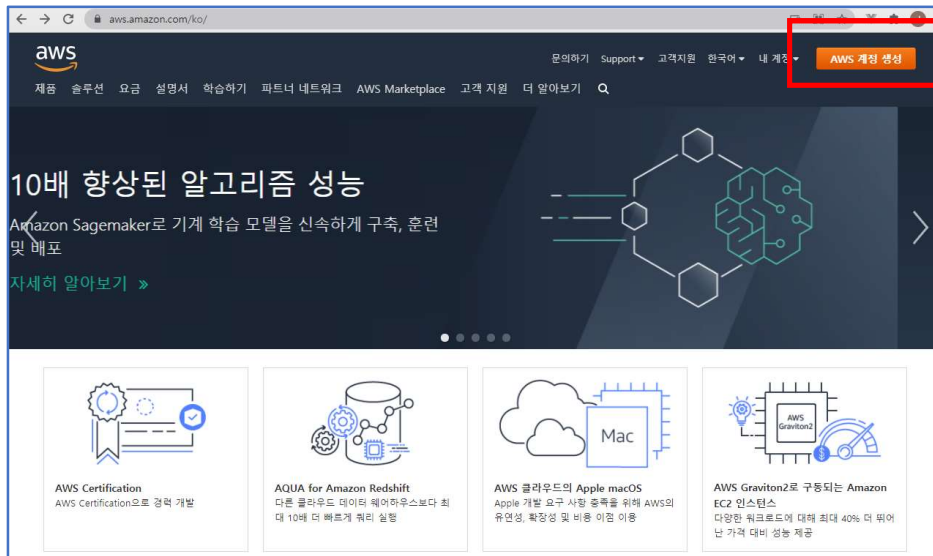
- AWS Free-Tier 계정
- Google Chrome or Mozilla Firefox

3. Tasks

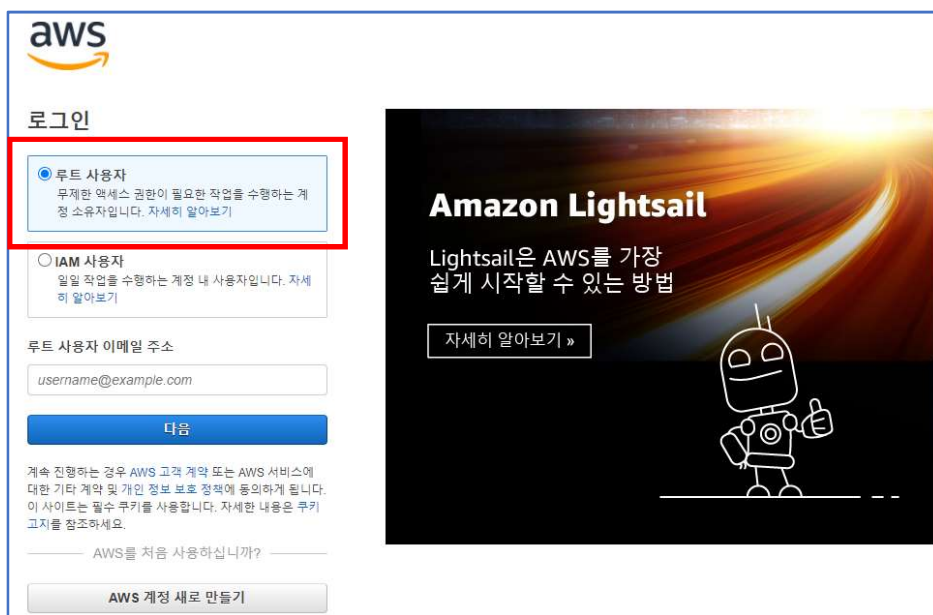
- Task1. AWS Login
- Task2. VPC Network 구성하기
- Task3. 보안 그룹 생성하기
- Task4. EC2 Instance 생성하기
- Task5. Ubuntu Linux 인스턴스 접속하기
- Task6. Linux 서버 시작, 중지하기
- Task7. Linux Server 인스턴스 영구 삭제하기

Task1. AWS Login

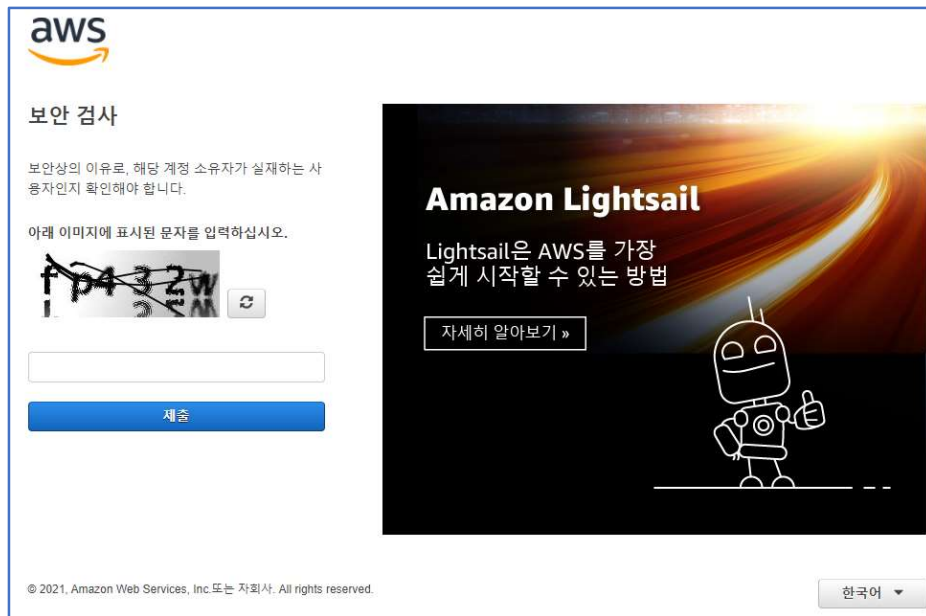
1. 웹 브라우저를 열고 <https://aws.amazon.com/ko/> 에 접속한다. 우상단에 [콘솔에 로그인] 버튼이 보이면 클릭하고, 아래의 그림처럼 [AWS 계정 생성]이라는 버튼이 보여도 오렌지색 버튼을 클릭한다.



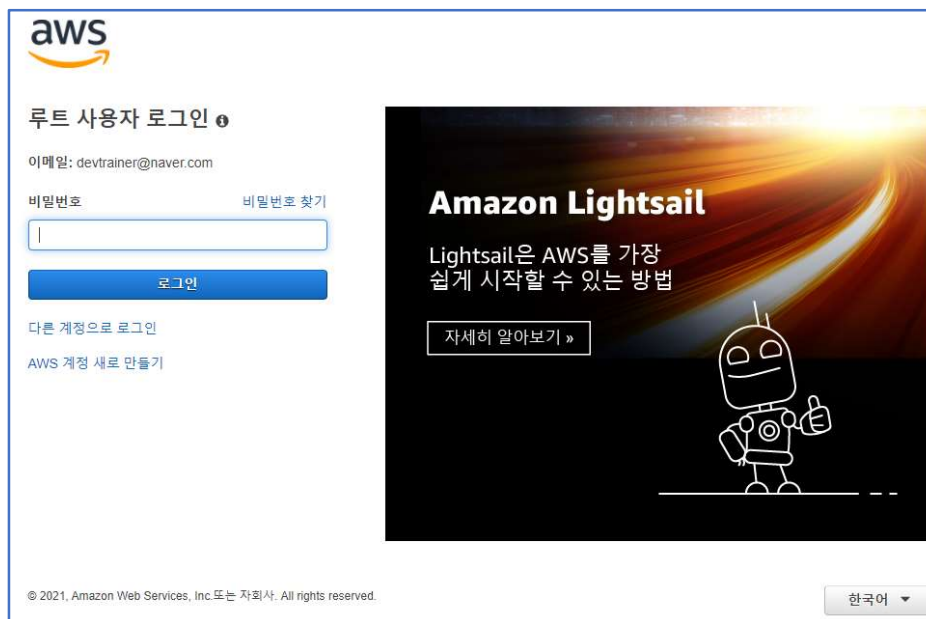
2. 이전에 생성한 AWS 계정 정보를 이용해서 로그인을 진행한다. [루트 사용자]를 선택하고, [루트 사용자 이메일 주소]를 넣고 [다음] 버튼을 클릭한다.



3. 기계를 이용한 자동 로그인을 방지하기 위해 AWS에서는 아래와 같이 보안검사를 시행하고 있다. 그림에 보이는 대로 입력하고 [제출] 버튼을 클릭하자.

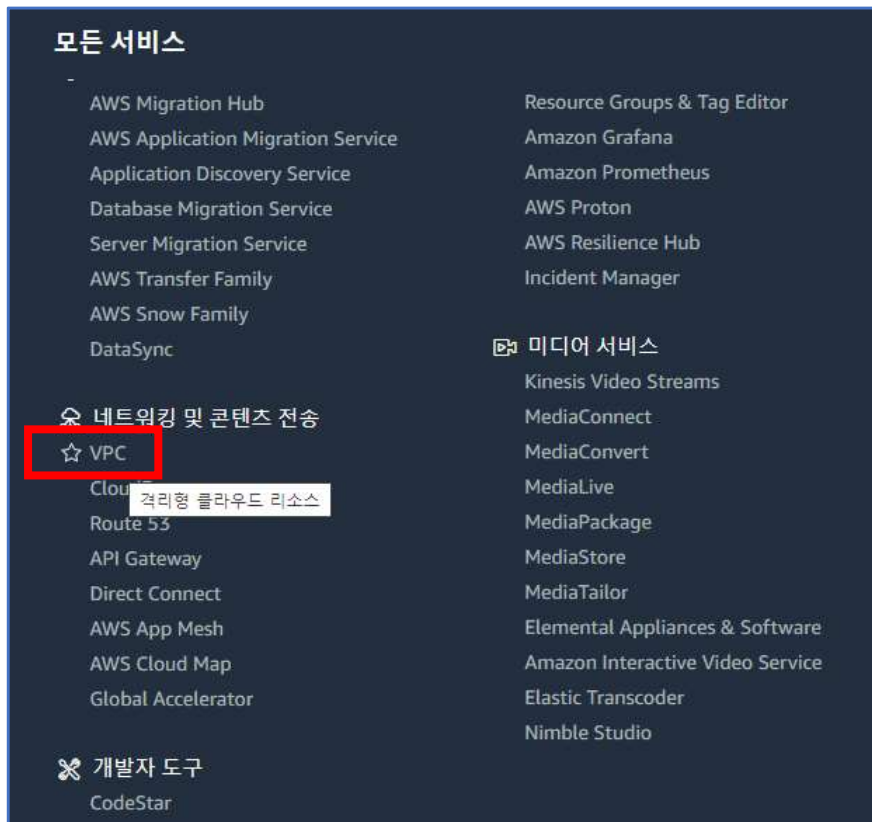


- 이전에 생성했던 계정의 비밀번호를 입력하고 **[로그인]** 버튼을 클릭한다. 한번 더 보안 검사를 요구할 수도 있다.

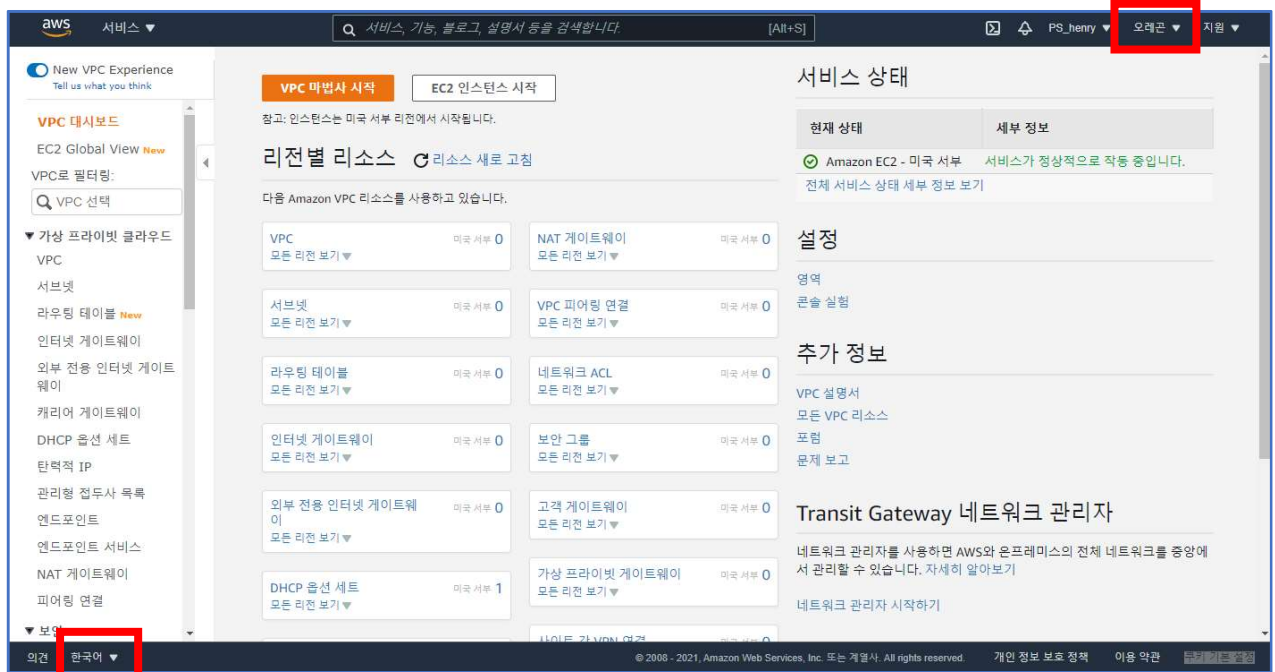


Task2. VPC Network 구성하기

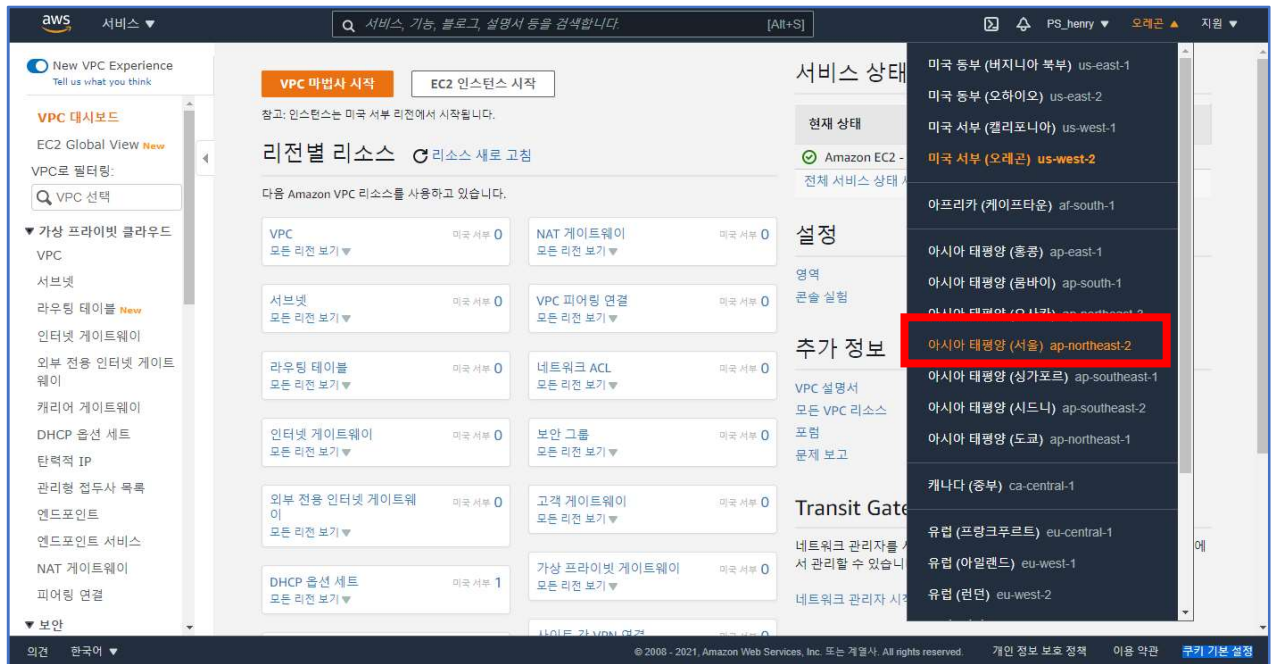
1. AWS Console에서 [서비스] > [네트워킹 및 콘텐츠 전송] > [VPC]를 클릭하여 들어간다.



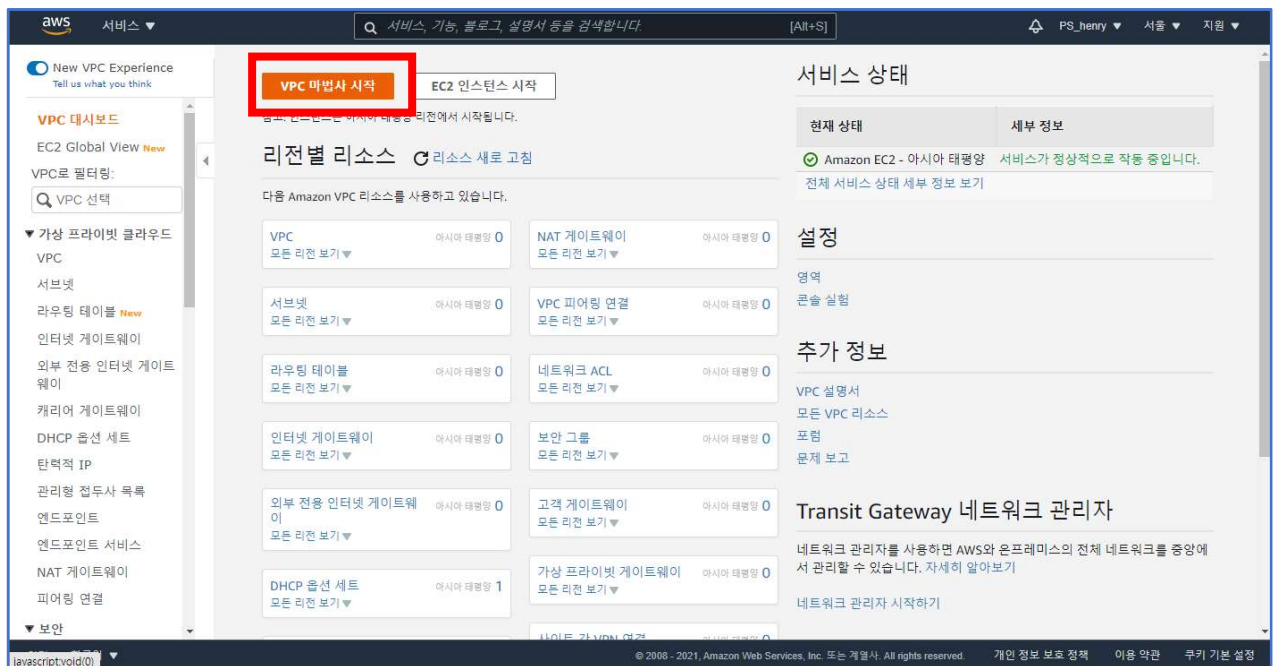
2. VPC 페이지로 들어왔다. 먼저 확인할 것은 화면 좌측 하단의 언어에서 [한국어]로 설정되어 있는지 확인한다. 또한 화면 우측 상단의 Region이 서울인지 확인한다.



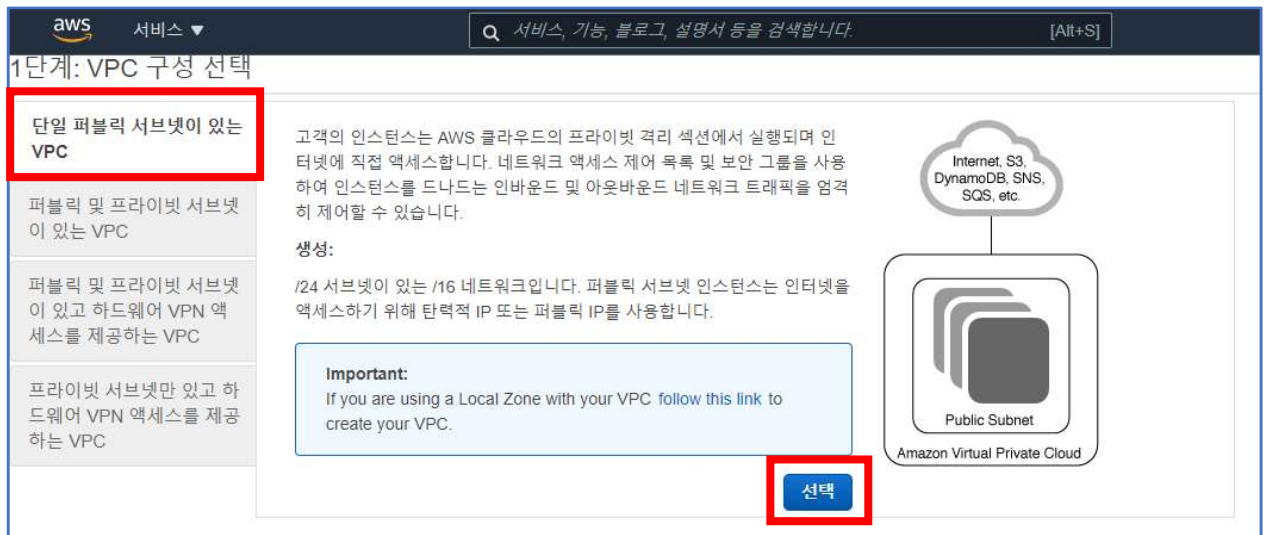
3. 만일 Region이 서울이 아니라면 다음 그림과 같이 설정하여 [아시아 태평양(서울)]로 맞춘다.



4. 페이지 위쪽의 [VPC 마법사 시작]을 클릭하여 VPC 설정을 시작하도록 한다.



5. [1단계:VPC 구성 선택]단계 4가지 종류에서 제일 위에 있는 [단일 퍼블릭 서브넷이 있는 VPC]를 선택하고 [선택] 버튼을 클릭한다.

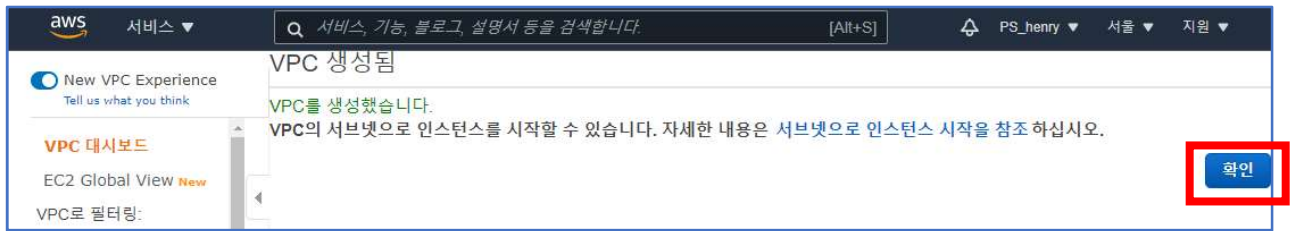


6. [2단계: 단일 퍼블릭 서브넷이 있는 VPC] 단계이다. 다음과 같이 설정하고 나머지 값은 기본값을 그대로 사용한다. 모든 설정이 마치면 [VPC 생성] 파란색 버튼을 클릭한다.

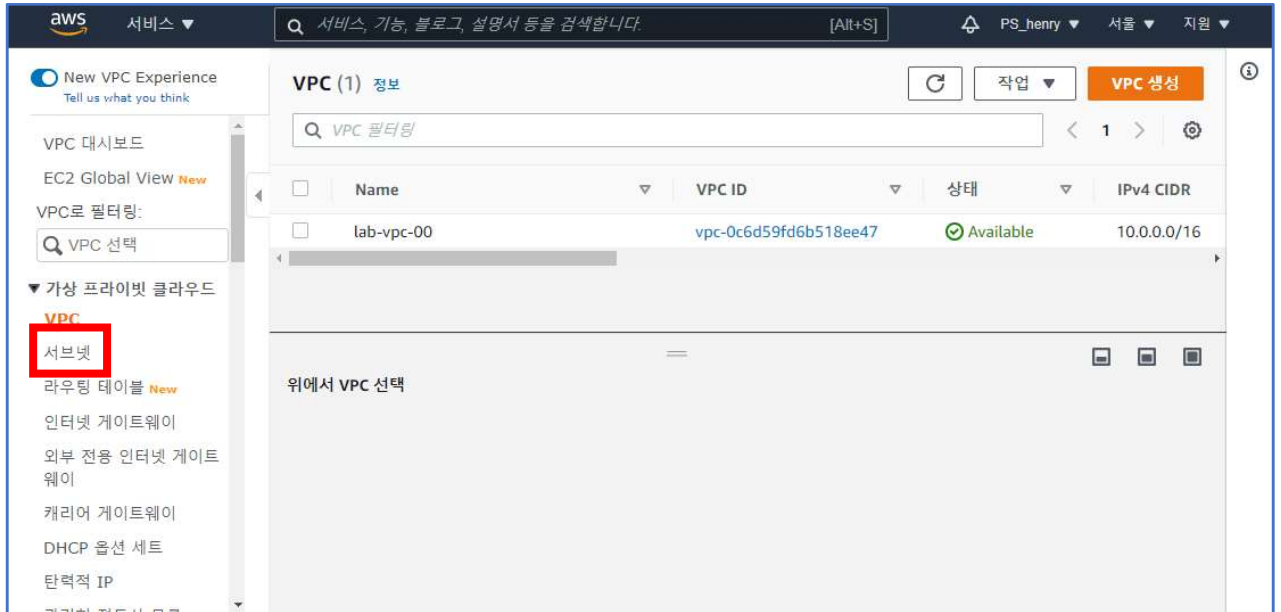
- A. [IPv4 CIDR 블록] : 10.0.0.0/16
- B. [VPC 이름] : lab-vpc-xx
- C. [퍼블릭 서브넷의 IPv4 CIDR] : 10.0.10.0/24
- D. [가용 영역] : ap-northeast-2a
- E. [서브넷 이름] : lab-vpc-public-subnet-xx



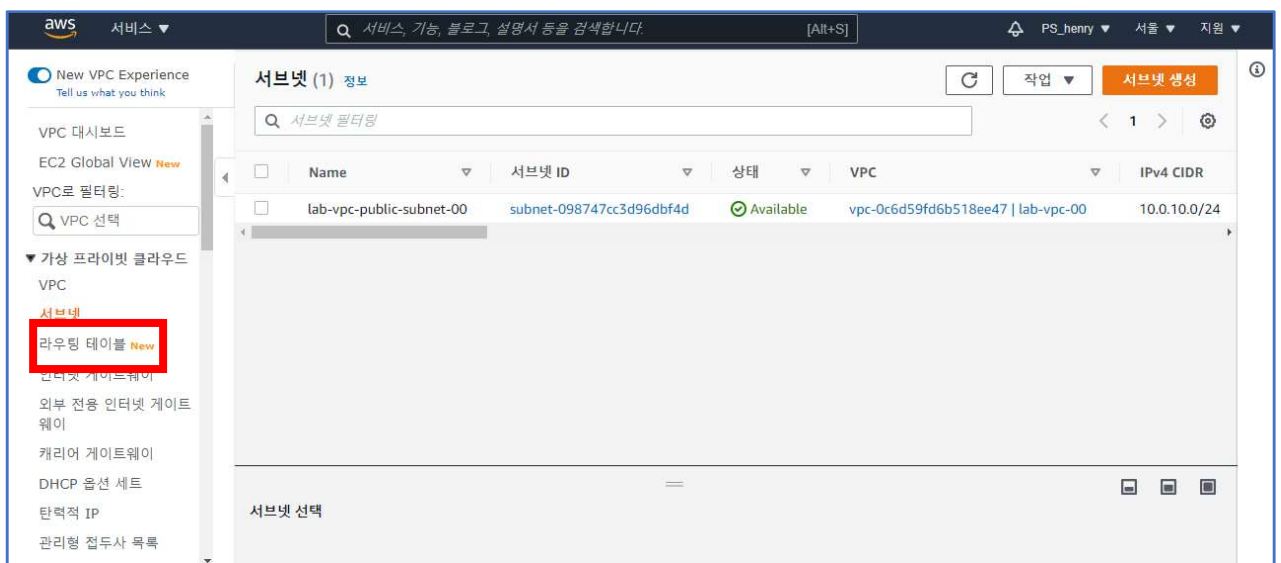
7. VPC가 성공적으로 생성되었음을 확인한다. **[확인]** 버튼을 클릭한다.



8. 방금 생성한 VPC 페이지로 이동된다. 좌측 메뉴 중 **[가상 프라이빗 클라우드] > [서브넷]**을 클릭한다.



9. VPC를 생성할 때 같이 생성했던 서브넷을 확인할 수 있다. 이번에는 **[서브넷]** 메뉴 밑에 있는 **[라우팅 테이블]** 메뉴를 클릭한다.



10. 현재 라우팅 테이블의 이름을 지정하지 않아서 **[Name]**이 빠져있지만 **[명시적 서브넷 연결]**을 보면 방금 생성된 라우팅 테이블을 확인할 수 있다. 해당 라우팅 테이블을 선택하면 화면 하단에 보다 자세한 정보를 확인할 수 있다. **[세부 정보]** 오른쪽 탭인 **[라우팅]** 탭을 클릭해보자.

The screenshot shows the AWS Management Console interface for a VPC. The left sidebar contains navigation links for VPC resources. The main content area displays the '라우팅 테이블 (1/2) 정보' page. A table lists routing tables, with 'rtb-0f56bdf0613ff3e5f' selected. Below the table, the '라우팅' tab is active, showing the '세부 정보' (Detailed Information) section. This section displays the following details:

라우팅 테이블 ID	기본	명시적 서브넷 연결	옛지 연결
rtb-0f56bdf0613ff3e5f	예	-	-

Additional details shown include the VPC ID (vpc-0c6d59fd6b518ee47) and the user ID (540643697040).

11. 라우팅 정보를 보면 10.0.0.0/16을 사용하는 IP는 VPC내(local)에서 처리하며, 그 외의 IP는 대상이 인터넷 게이트웨이로 되어 있음을 확인할 수 있다.

The screenshot shows the AWS Management Console interface for a VPC. The left sidebar contains navigation links for VPC resources. The main content area displays the '라우팅 테이블 (1/2) 정보' page. The '라우팅' tab is active, showing the '라우팅 (2)' section. This section displays a list of routes with the following details:

대상	대상	상태	전파됨
10.0.0.0/16	local	활성	아니요
0.0.0.0/0	igw-016bb7dceefff62a3	활성	아니요

12. 이번에는 [라우팅] 탭 오른쪽의 [서브넷 연결] 탭을 클릭하여 이동한다. 퍼블릭 서브넷에서 생성된 라우팅은 이 라우팅 테이블에서 처리함을 확인할 수 있다.

The screenshot displays the AWS Management Console interface for the 'New VPC Experience'. The left sidebar contains navigation links for VPC services, including '라우팅 테이블' (Routing Tables). The main content area is titled '라우팅 테이블 (1/2) 정보' (Routing Table (1/2) Info). A table lists routing tables, with the second one selected. Below the table, the '서브넷 연결' (Subnet Connections) tab is active and highlighted with a red box. This tab shows '명시적 서브넷 연결 (1)' (Explicit Subnet Connections (1)). A table lists the explicit connections, showing a public subnet connected to the routing table. The '서브넷 연결' tab is highlighted with a red box.

Name	라우팅 테이블 ID	명시적 서브넷 연결	엣지 연결	기본	VPC
-	rtb-0f56bdf0613ff3e5f	-	-	예	vpc-0c6d5...
<input checked="" type="checkbox"/>	rtb-0ef1d035cd6047d12	subnet-098747cc3d96d...	-	아니요	vpc-0c6d5...

rtb-0ef1d035cd6047d12

세부 정보 | 라우팅 | **서브넷 연결** | 엣지 연결 | 라우팅 전파 | 태그

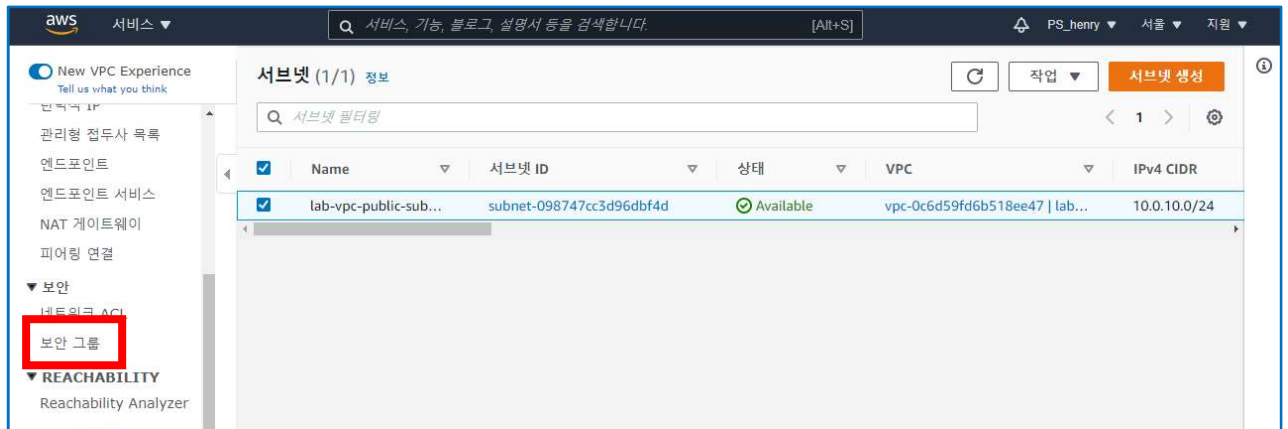
명시적 서브넷 연결 (1)

서브넷 ID	IPv4 CIDR	IPv6 CIDR
subnet-098747cc3d96dbf4d / lab-vpc-public-subnet-00	10.0.10.0/24	-

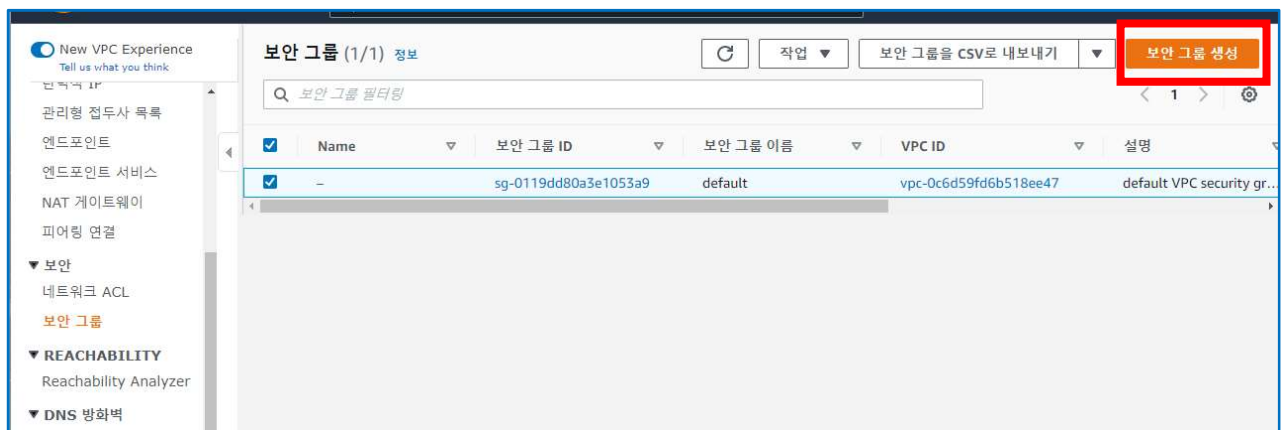
명시적 연결이 없는 서브넷 (0)

Task3. 보안 그룹 생성하기

1. 네트워크 ACL이 서브넷 단위의 방화벽 역할을 한다면, 보안 그룹은 인스턴스에 대한 Inbound 및 Outbound 트래픽을 제어하는 가상 방화벽 역할을 한다.
2. 페이지 좌측 메뉴 중 [보안] > [보안 그룹]을 클릭한다.



3. 보안 그룹 페이지로 들어왔다. 새 보안 그룹을 생성하기 위해 [보안 그룹 생성] 버튼을 클릭한다.



4. 다음과 같이 설정한다.
 - A. [보안 그룹 이름] : docker-ubuntu-sg
 - B. [설명] : Security group for docker-ubuntu instance
 - C. [VPC] : lab-vpc-00

기본 세부 정보

보안 그룹 이름 [정보](#)

docker-ubuntu-sg

생성 후에는 이름을 편집할 수 없습니다.

설명 [정보](#)

Security group for docker-ubuntu instance

VPC [정보](#)

Q |

vpc-0c6d59fd6b518ee47 (lab-vpc-00)

10.0.0.0/16

5. 동일한 페이지를 스크롤다운하여 **[인바운드 규칙]** 섹션으로 이동한다. 새 규칙을 추가하기 위해 **[규칙 추가]** 버튼을 클릭한다.

인바운드 규칙 [정보](#)

이 보안 그룹에는 인바운드 규칙이 없습니다.

규칙 추가

6. 다음 그림과 같이 2개의 규칙을 추가한다.

A. **[유형]** : 모든 ICMP – IPv4, **[프로토콜]** : ICMP, **[포트 범위]** : 전체, **[소스]** : Anywhere-IPv4

B. **[유형]** : SSH, **[프로토콜]** : TCP, **[포트 범위]** : 22, **[소스]** : Anywhere-IPv4

인바운드 규칙 [정보](#)

유형 정보	프로토콜 정보	포트 범위 정보	소스 정보	설명 - 선택 사항 정보	
모든 ICMP - IPv4 ▼	ICMP	전체	Anywh... ▼ Q		삭제
			0.0.0.0/0 X		
SSH ▼	TCP	22	Anywh... ▼ Q		삭제
			0.0.0.0/0 X		

규칙 추가

7. [아웃바운드 규칙]은 기본값 그대로 사용한다. 모든 설정을 마치면 페이지를 계속 스크롤다운하여 페이지 제일 하단의 [보안 그룹 생성] 버튼을 클릭한다.

아웃바운드 규칙 정보

유형 정보

모든 트래픽 ▼

프로토콜 정보

전체

포트 범위 정보

전체

대상 정보

사용자 ... ▼

0.0.0.0/0 ✕

설명 - 선택 사항 정보

삭제

규칙 추가

태그 선택 사항

태그는 사용자가 AWS 리소스에 할당하는 레이블입니다. 각 태그는 키와 값(선택 사항)으로 구성됩니다. 태그를 사용하여 리소스를 검색 및 필터링하거나 AWS 비용을 추적할 수 있습니다.

리소스와 연결된 태그가 없습니다.

새로운 태그 추가

최대 50개의 태그를 더 추가할 수 있습니다.

취소

보안 그룹 생성

8. 방금 생성한 보안 그룹을 확인할 수 있다.

New VPC Experience

Tell us what you think

관리형 접두사 목록

엔드포인트

엔드포인트 서비스

NAT 게이트웨이

피어링 연결

▼ 보안

네트워크 ACL

보안 그룹

▼ REACHABILITY

Reachability Analyzer

▼ DNS 방화벽

규칙 그룹 New

도메인 목록 New

▼ 네트워크 방화벽

방화벽

방화벽 정책

네트워크 방화벽 규칙 그룹

▼ VPN(가상 프라이빗 네트워크)

고객 게이트웨이

가상 프라이빗 게이트웨이

사이트 간 VPN 연결

보안 그룹 sg-0b9d65892e2a4326b | docker-ubuntu-sg)이 생성되었습니다.

세부 정보

VPC > 보안 그룹 > sg-0b9d65892e2a4326b - docker-ubuntu-sg

작업 ▼

sg-0b9d65892e2a4326b - docker-ubuntu-sg

세부 정보

보안 그룹 이름

docker-ubuntu-sg

보안 그룹 ID

sg-0b9d65892e2a4326b

설명

Security group for docker-ubuntu instance

VPC ID

vpc-0c6d59fd6b518ee47

소유자

540643697040

인바운드 규칙 수

2 권한 항목

아웃바운드 규칙 수

1 권한 항목

인바운드 규칙

아웃바운드 규칙

태그

인바운드 규칙 (2)

태그 관리

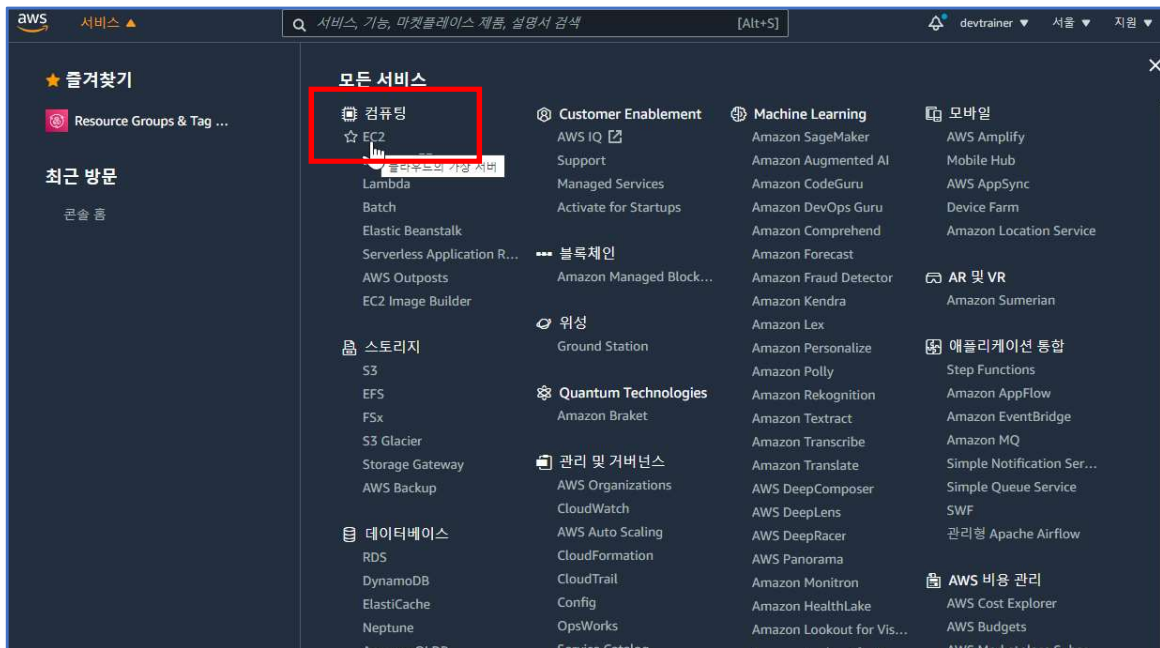
인바운드 규칙 편집

보안 그룹 규칙 필터

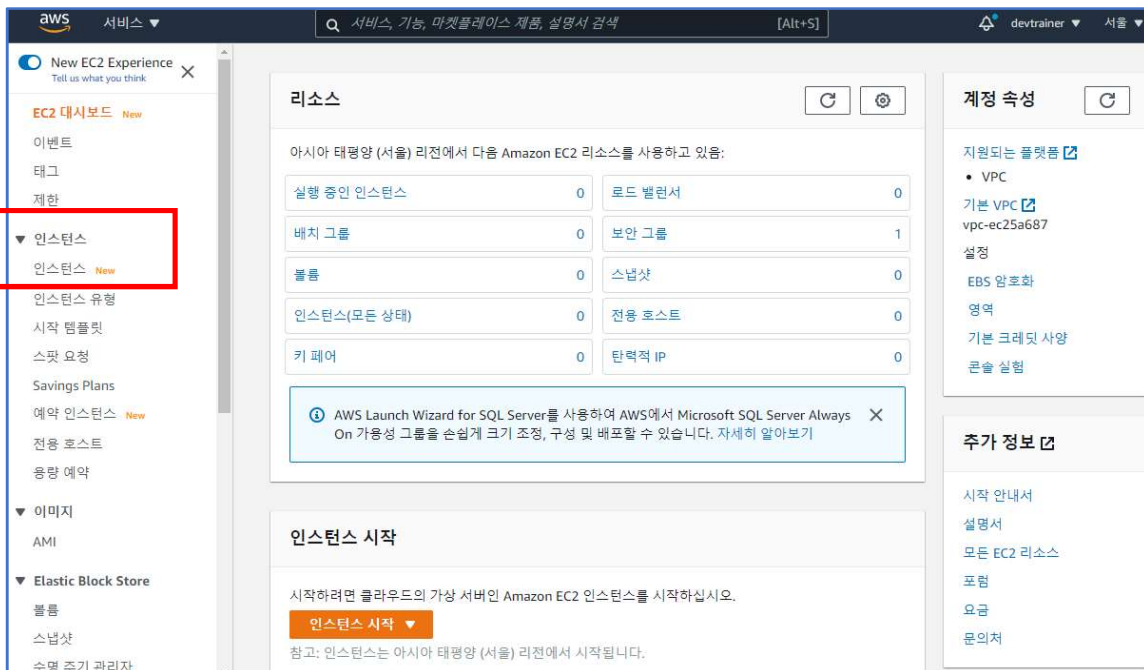
	Name	보안 그룹 규칙 ID	IP 버전	유형	프로토콜
<input type="checkbox"/>	-	sgr-0b4cbce1c7f5a2005	IPv4	SSH	TCP
<input type="checkbox"/>	-	sgr-099fd246b8a7e4f50	IPv4	모든 ICMP - IPv4	ICMP

Task4. EC2 Instance 생성하기

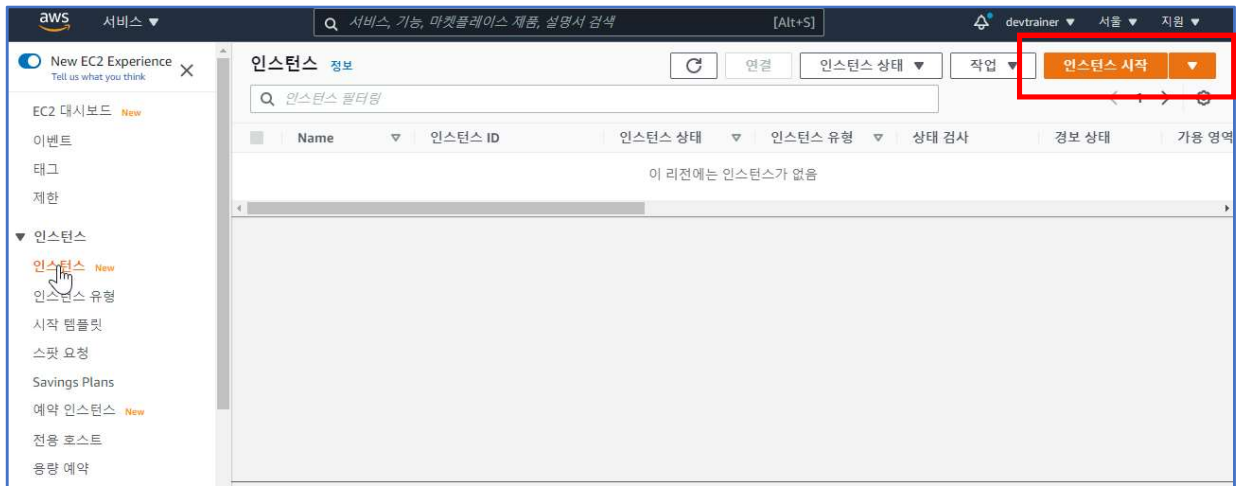
1. 좌측 상단의 [서비스] > [컴퓨팅] > [EC2]를 클릭하여 해당 페이지로 이동한다.



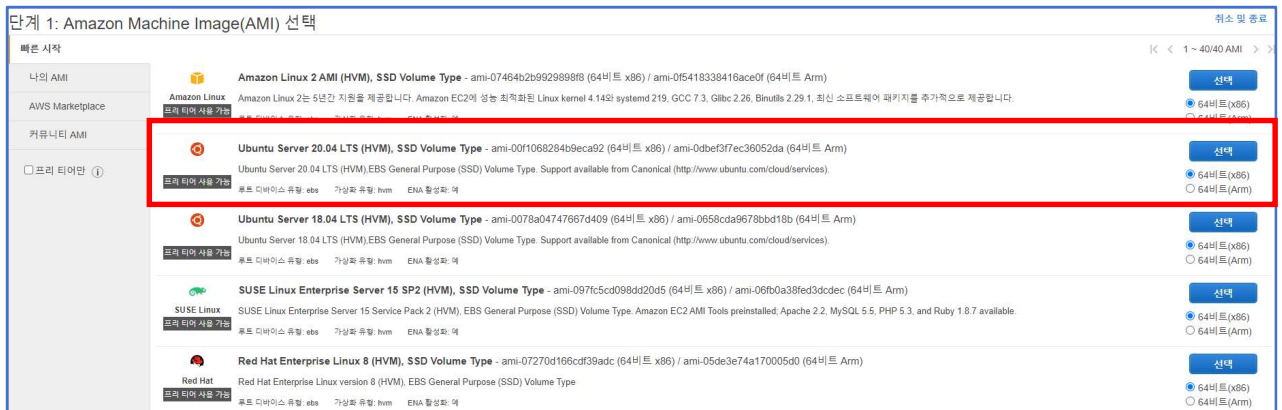
2. 왼쪽 항목에서 [인스턴스]를 선택하여 해당 페이지로 이동한다.



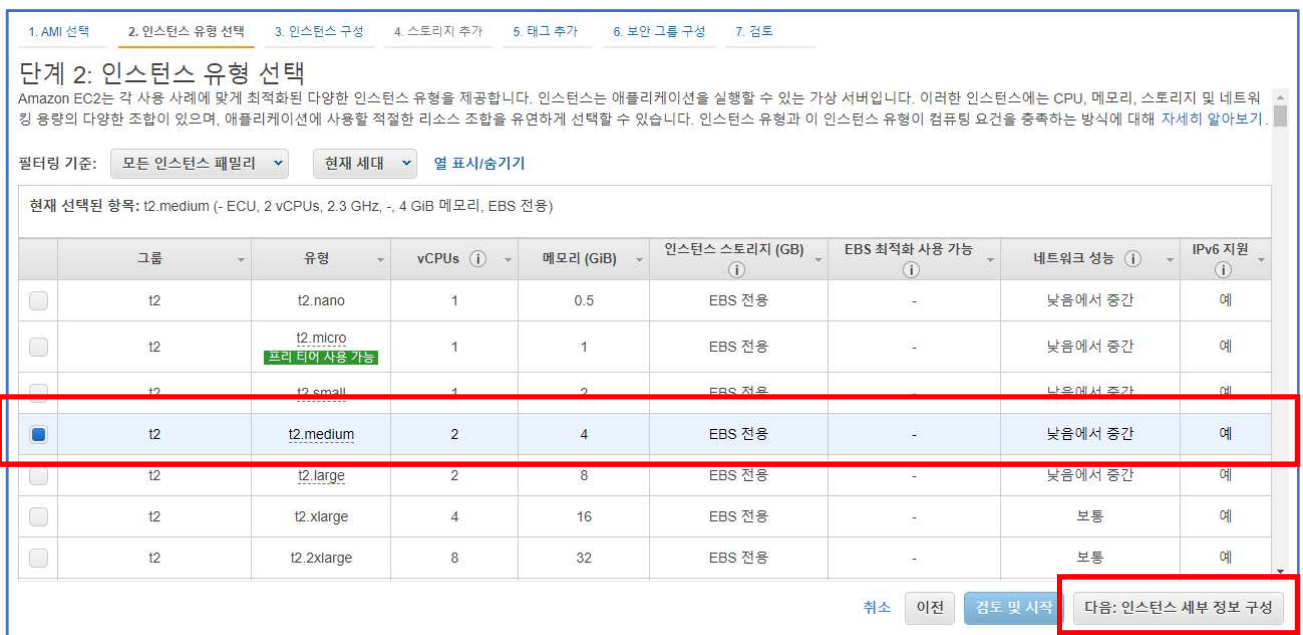
3. 우측 상단의 [인스턴스 시작] 오렌지 색 버튼을 클릭한다.



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.medium]를 선택 후, [다음: 인스턴스 세부 정보 구성] 버튼을 클릭한다.



6. [단계 3:인스턴스 세부 정보 구성] 페이지에서 다음의 각 값을 입력하고 나머지 값은 기본값 그대로 사용한다. 그리고 [다음:스토리지 추가] 버튼을 클릭한다.

A. [인스턴스 개수] : 1

B. [네트워크] : lab-vpc-00

C. [서브넷] : lab-vpc-public-subnet-00

D. [퍼블릭 IP 자동 할당] : 활성화

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

단계 3: 인스턴스 세부 정보 구성

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

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

인스턴스 개수 Auto Scaling 그룹 시작

구매 옵션 ☐ 스팟 인스턴스 요청

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

서브넷 새 서브넷 생성
251개 IP 주소 사용 가능

퍼블릭 IP 자동 할당

배치 그룹 ☐ 배치 그룹에 인스턴스 추가

용량 예약

도메인 조인 디렉터리 새 디렉터리 생성

IAM 역할 새 IAM 역할 생성

취소 이전 검토 및 시작 **다음: 스토리지 추가**

7. [단계 4:스토리지 추가] 페이지에서, Linux Server는 스토리지 크기가 8GiB로 맞춰져 있는데, Free-Tier 자격으로 최대 사용할 수 있는 스토리지 크기는 30GB이지만 수업을 위해 Linux Server 인스턴스 스토리지 크기를 50GiB로 설정한다. [다음:태그 추가] 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 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 시작한 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-038eea3a9d19f498e	<input type="text" value="50"/>	범용 SSD(gp2)	150/3000	해당 사항 없음	<input checked="" type="checkbox"/>	암호화5

새 볼륨 추가

취소 이전 검토 및 시작 **다음: 태그 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.올(름) 선택합니다. IAM 정책에 태그를 생성할 수 있는 권한이 포함되어 있는지 확인합니다.

태그 추가 (최대 50개 태그)

9. [키]에 "Name"를, [값]에 "Ubuntu Docker Server"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 | Ubuntu Docker Server | [x] | [x] | [x] | [x]

다른 태그 추가 (최대 50개 태그)

취소 | 이전 | 검토 및 시작 | **다음: 보안 그룹 구성**

10. [단계 6:보안 그룹 구성] 페이지에서, [보안 그룹 할당]을 [기본 보안 그룹 선택]을 선택한다. 이미 앞 Task에서 설정한 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 ID	이름	설명	작업
sg-0119dd80a3e1053a9	default	default VPC security group	새로 복사
sg-0b9d65892e2a4326b	docker-ubuntu-sg	Security group for docker-ubuntu instance	새로 복사

sg-0b9d65892e2a4326b에 대한 인바운드 규칙 (선택한 보안 그룹: sg-0b9d65892e2a4326b)

유형 ⓘ	프로토콜 ⓘ	포트 범위 ⓘ	소스 ⓘ	설명 ⓘ
SSH	TCP	22	0.0.0.0/0	
모든 ICMP - IPv4	모두	해당 사항 없음	0.0.0.0/0	

취소 | 이전 | **검토 및 시작**

의견 한국어 ▼ © 2008 - 2021, Amazon Web Services, Inc. 또는 계열사. All rights reserved. 개인 정보 보호 정책 이용 약관 쿠키 기본 설정

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

1. AMI 선택2. 인스턴스 유형 선택3. 인스턴스 구성4. 스토리지 추가5. 태그 추가6. 보안 그룹 구성7. 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.

⚠

해당 인스턴스 구성은 프리 티어에 사용할 수 없습니다.
프리 티어에 사용할 수 있는 인스턴스를 시작하려면 AMI 선택, 인스턴스 유형, 구성 옵션 또는 스토리지 디바이스를 확인하십시오. 프리 티어 자격 및 사용량 제한에 대해 자세히 알아보십시오.

이 메시지를 다시 표시 하

⚠

인스턴스 보안을 개선하십시오. 보안 그룹 docker-ubuntu-sg이(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)을 엽니다. 보안 그룹 편집

AMI 세부 정보

AMI 편집

Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-0f8b8babb98cc66d0

프리 티어
사용 가능

루트 디바이스 유형: ebs 가상화 유형: hvm

Ubuntu Server 20.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

인스턴스 유형

인스턴스 유형 편집

인스턴스 유형	ECU	vCPUs	메모리 (GiB)	인스턴스 스토리지 (GiB)	EBS 최적화 사용 가능	네트워크 성능
t2.medium	-	2	4	EBS 전용	-	Low to Moderate

보안 그룹

보안 그룹 편집

취소

이전

시작하기

12. [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.

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

기존 키 페어 선택

가 페어를 선택하십시오

키 페어 없음

⚠

키 페어 없음

키 페어가 없습니다. 계속하려면 위에서 [새 키 페어 생성] 옵션을 선택하여 새 키 페어를 작성하십시오.

취소

인스턴스 시작

13. [기존 키 페어 선택] 드롭다운을 클릭하면 보이는 3개의 항목 중에 “새 키 페어 생성”을 선택하고, [키 페어 이름]에 “**Docker-Ubuntu-RSAKey**”를 입력 후 [키 페어 다운로드] 버튼을 클릭하여 “**Docker-Ubuntu-RSAKey.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
Docker-Ubuntu-RSAKey

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

다른 이름으로 저장

내 PC > 로컬 디스크 (C:) > Temp

Temp 검색

구성 새 폴더

즐거찾기

OneDrive - Personal

내 PC

3D 개체

Desktop

Downloads

동영상

문서

사진

음악

로컬 디스크 (C:)

Google Drive (G:)

네트워크

2020

AUtempR

전자정부 표준프레임워크 교육자료

파일 이름(N): Docker-Ubuntu-RSAKey.pem

파일 형식(T): PEM 파일 (*.pem)

폴더 숨기기

저장(S)

취소

14. 키 페어 다운로드 완료 후 **[인스턴스 시작]** 버튼을 클릭한다. 인스턴스는 보통 5 ~ 10분 정도 시간이 걸린다.

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

Docker-Ubuntu-RSAKey

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. **[시작 상태]** 페이지가 나타난다. 현재 방금 생성한 인스턴스가 시작 중임을 알 수 있다. 페이지 하단의 **[인스턴스 보기]** 버튼을 클릭한다.

시작 상태

지금 인스턴스를 시작 중입니다.
다음 인스턴스 시작 ID: i-0e13a5f8c1f14dc8c 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 누적됩니다. 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.

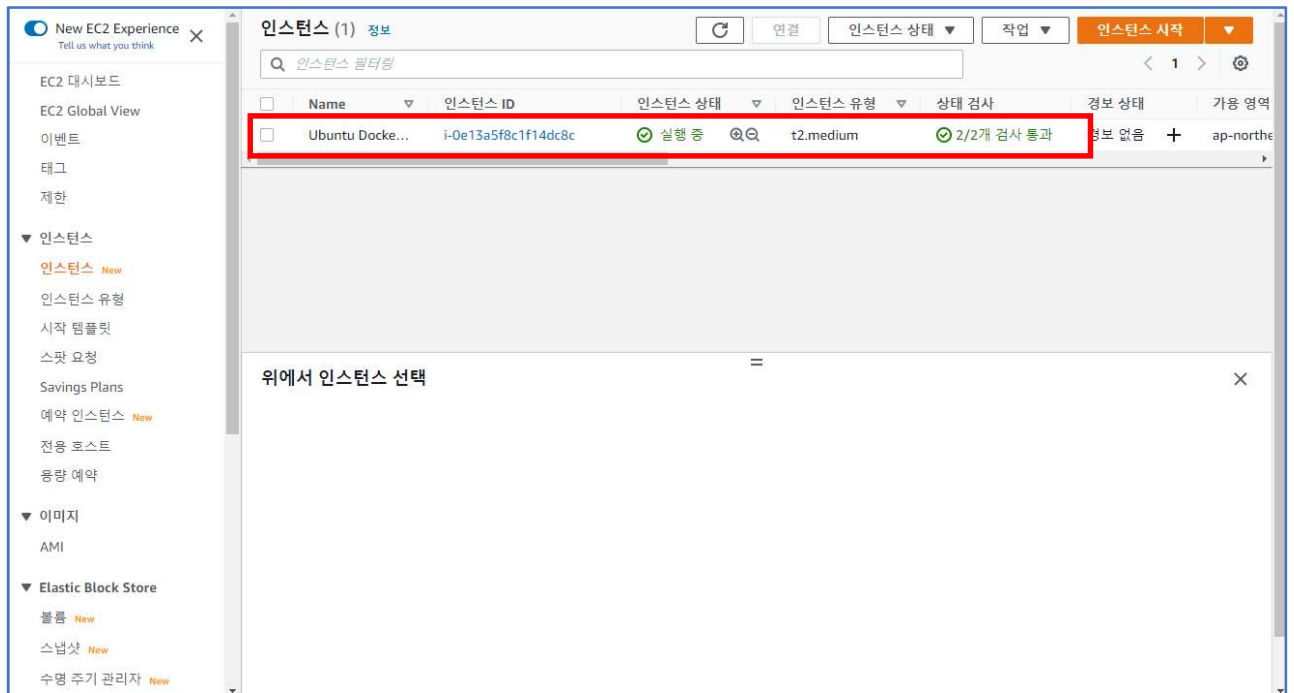
상태 검사 경보 생성 해당 인스턴스가 상태 검사를 통과하지 못하는 경우 알림을 받습니다. (추가 요금이 적용될 수 있음)

추가 EBS 볼륨 생성 및 연결 (추가 요금이 적용될 수 있음)

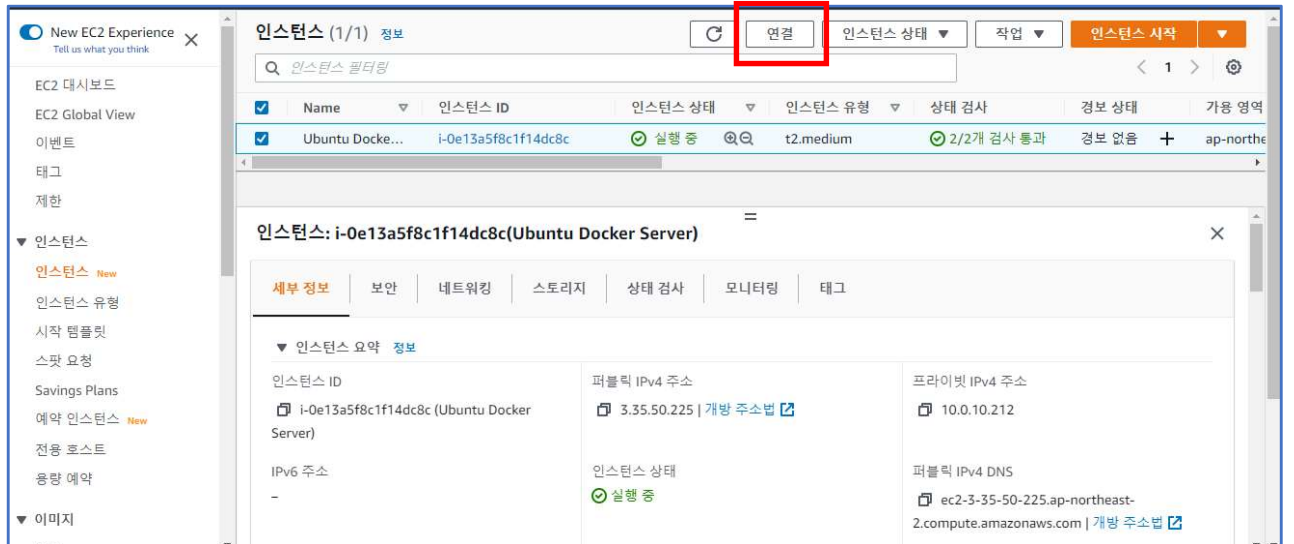
보안 그룹 관리

인스턴스 보기

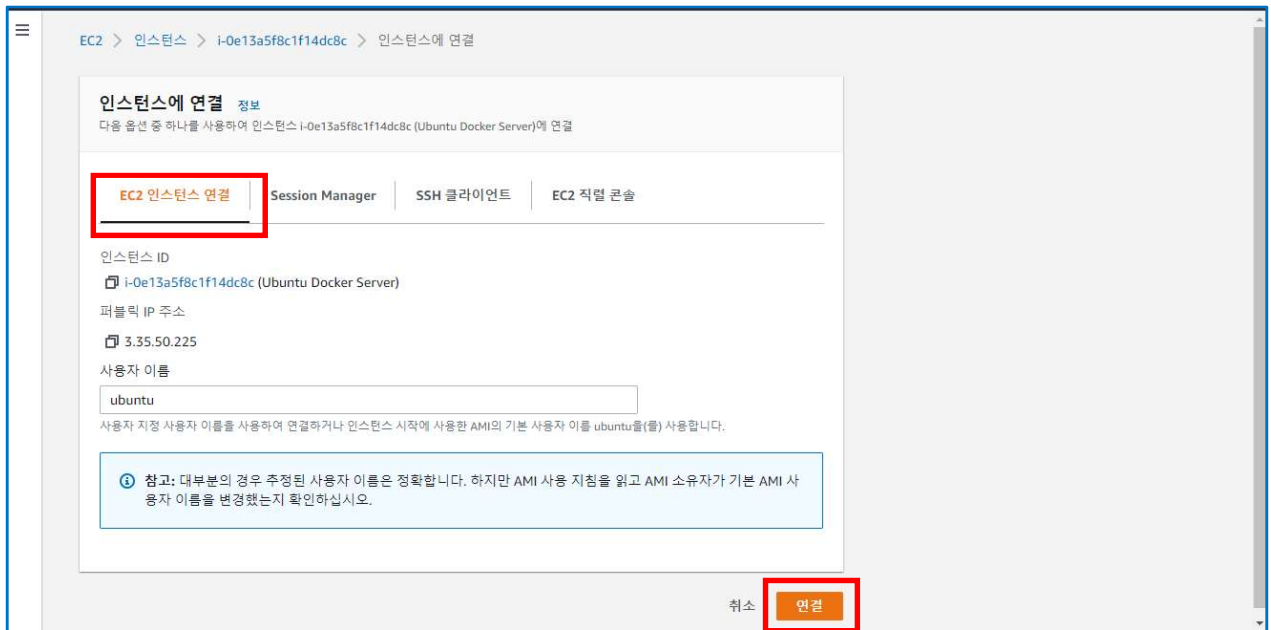
16. 인스턴스가 생성되면 시스템 상태 검사와 인스턴스 상태 검사 2가지를 수행한다. [상태 검사]가 [2/2개 검사 통과]라고 상태 검사가 모두 마칠 때까지 기다린다. 상태 검사가 모두 마치면 이제 인스턴스와 연결할 수 있다.



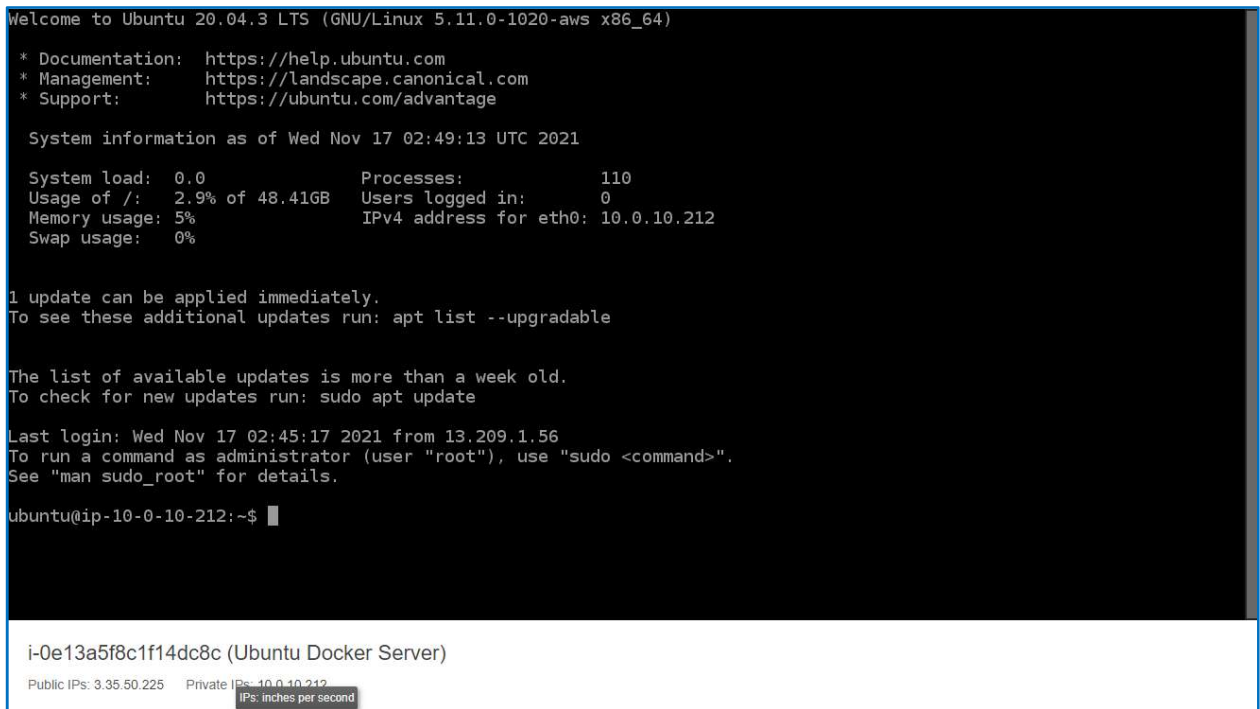
17. 해당 인스턴스를 선택하고 페이지 상단의 [연결]을 클릭한다.



18. [인스턴스에 연결] 페이지에서 [EC2 인스턴스 연결] 탭을 선택한다. 그리고 [연결] 버튼을 클릭한다.

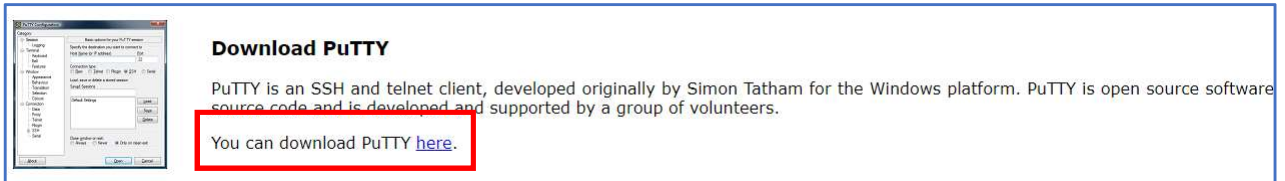


19. 생성한 인스턴스에 잘 연결되는 것을 확인할 수 있다.

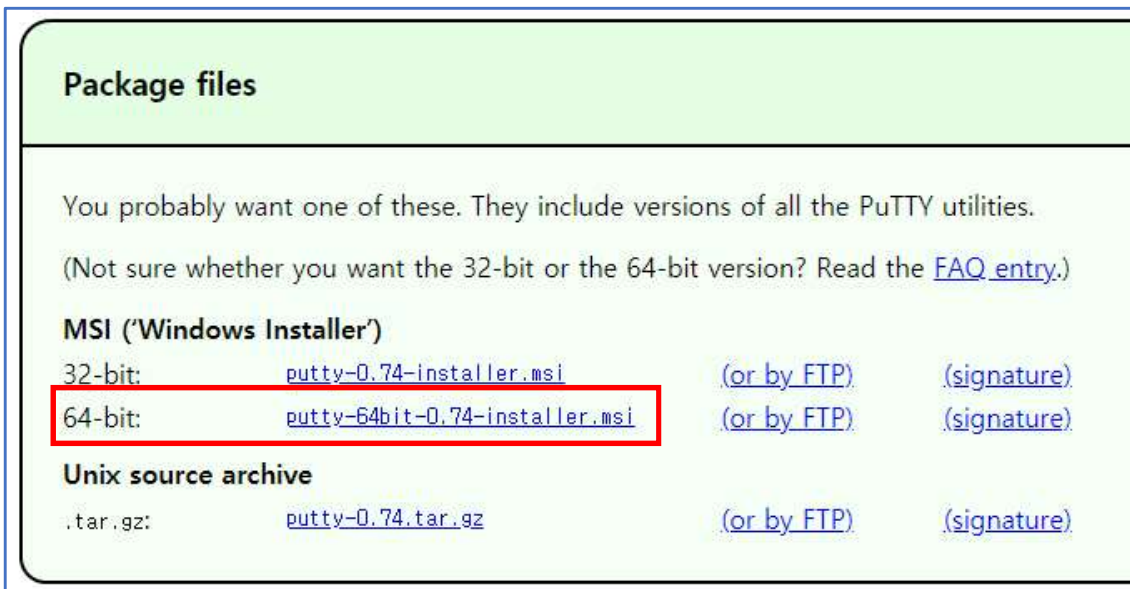


Task5. Ubuntu Linux 인스턴스 접속하기

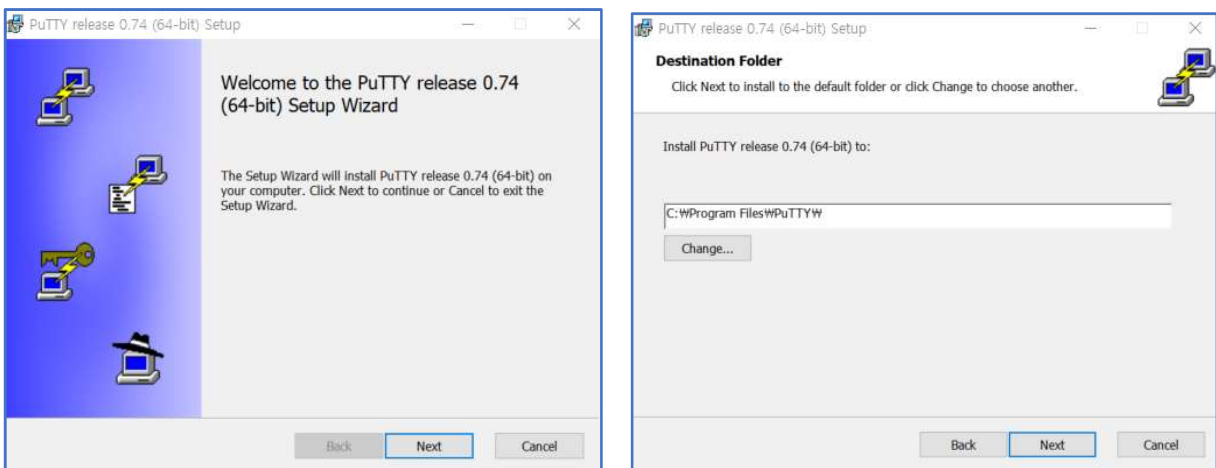
1. Linux 인스턴스 접속을 위해서는 일반적으로 SSH 접속용 프로그램이 필요하다. 가장 일반적으로 사용하는 SSH 툴은 **PuTTY**이다. <https://www.putty.org/> 에 접속한 후, **[Download PuTTY]** 섹션의 "You can download PuTTY **here**"의 **here** 링크를 클릭한다.

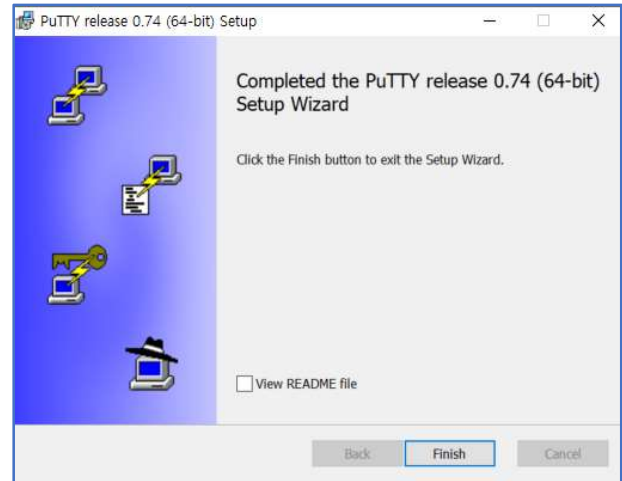
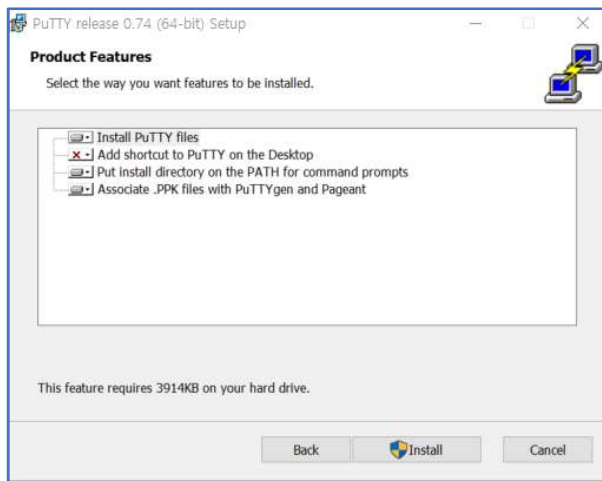


2. **[Download PuTTY:latest release(0.74)]**페이지에서 본인 PC 혹은 Notebook의 운영체제 버전(**Windows** or Unix)과 CPU Architecture(32-bit or **64-bit**)를 확인하여 다운로드 받을 수 있도록 링크를 클릭한다. 여기서는 일반적으로 Windows(MSI)의 64-bit를 다운로드받기 위해 해당 링크(**putty-64bit-0.74-install.msi**)를 클릭하도록 하겠다.

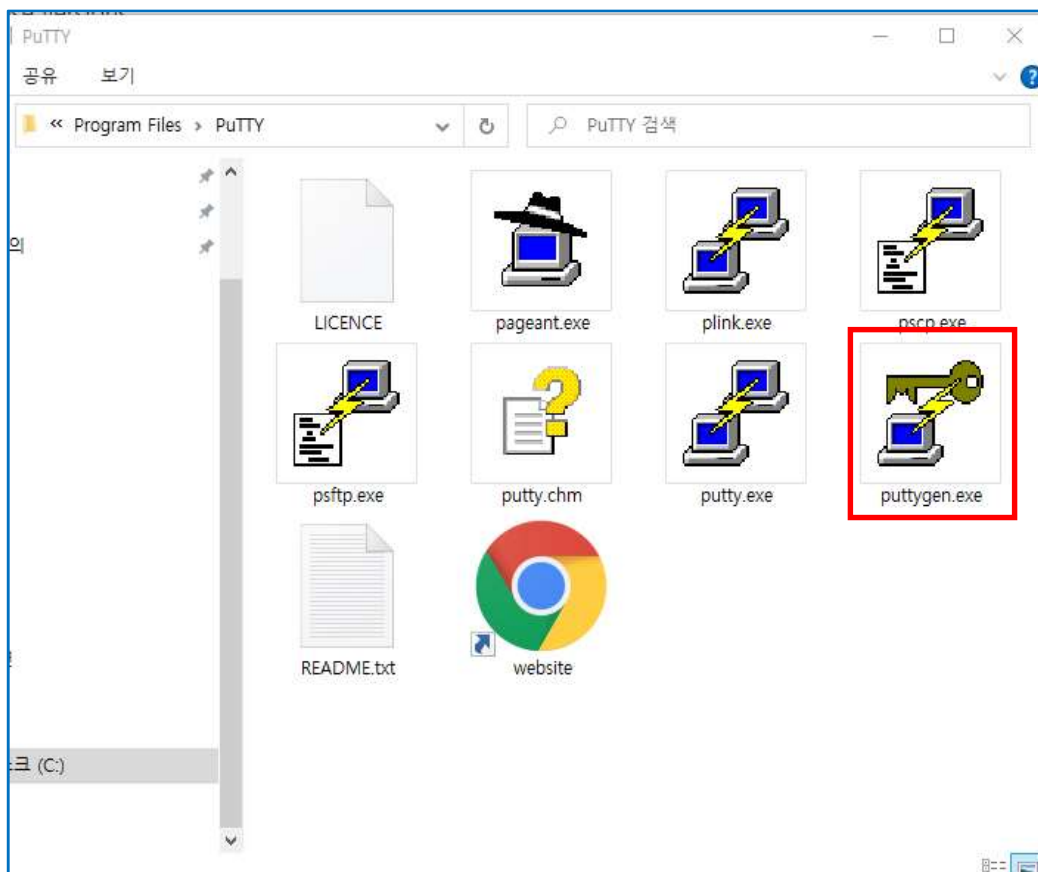


3. 해당 파일이 다운로드가 끝나면 바로 탐색기에서 더블클릭하여 프로그램을 설치한다. 설치할 때에는 해당 화면에서 기본값을 사용하도록 계속 **[Next]** 그리고 **[Install]** 버튼을 클릭한다.

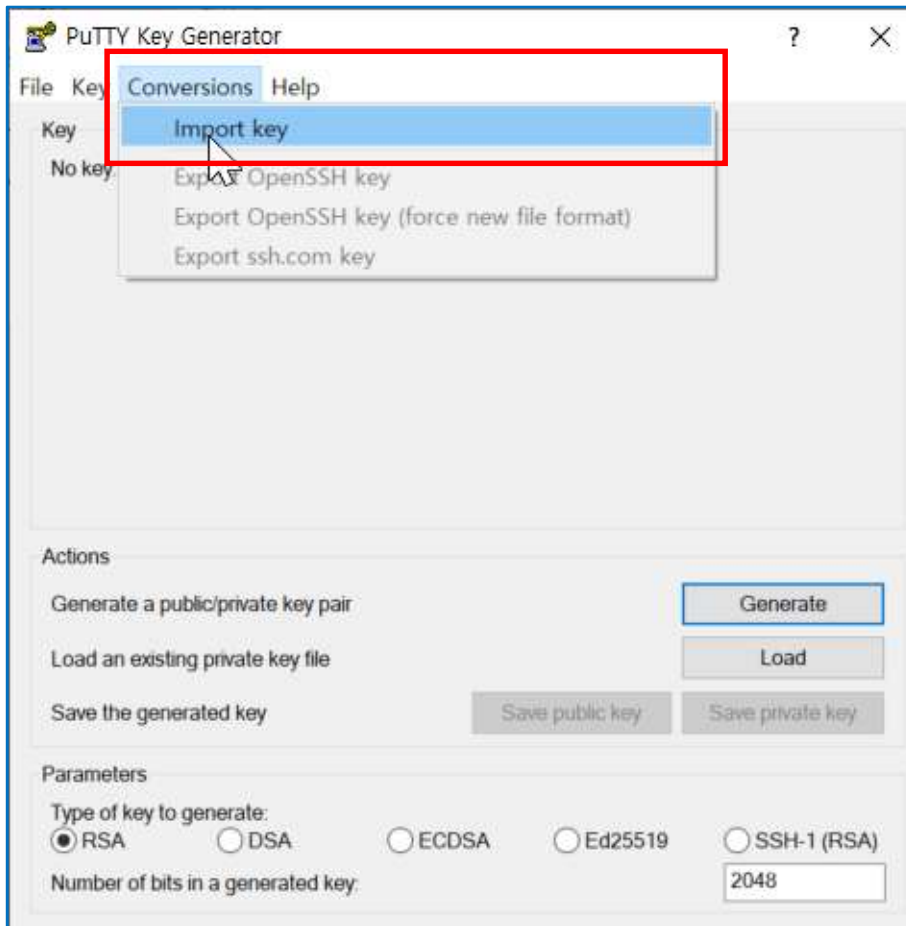




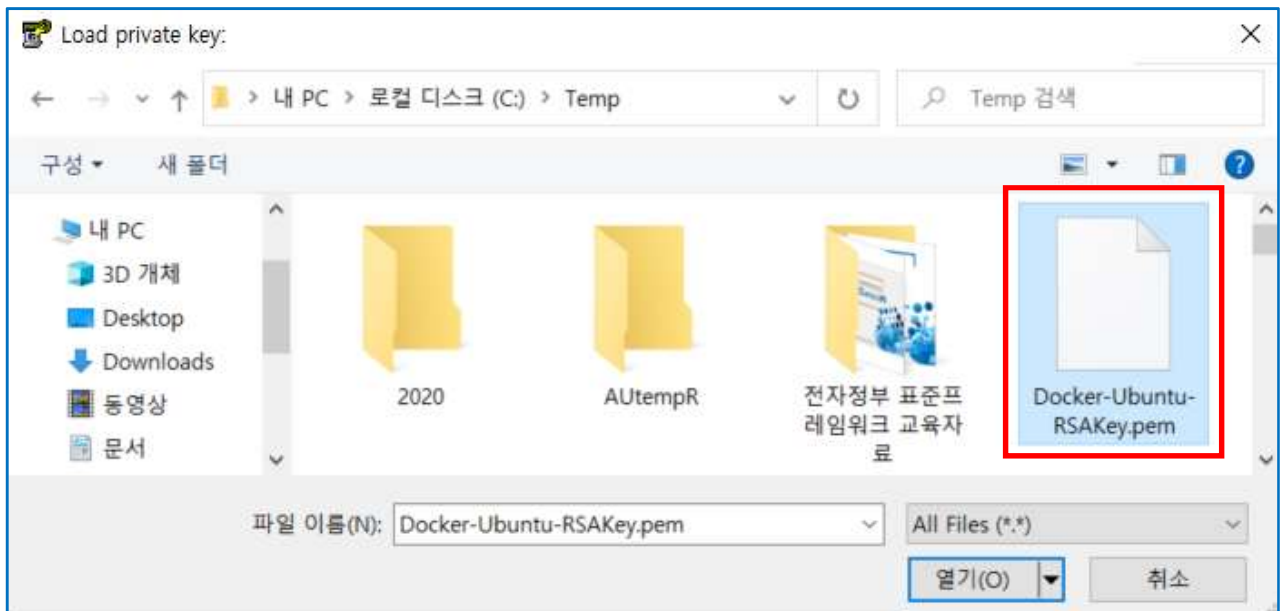
4. 위에서 이미 다운로드 받은 “키 페어 파일”을 PuTTY 프로그램과 연결하기 위해 PuTTY 프로그램이 설치된 경로(C:\Program Files\PuTTY)로 이동한다. 그 폴더에 가면 “puttygen.exe”파일이 있는데, 더블클릭하여 실행한다.



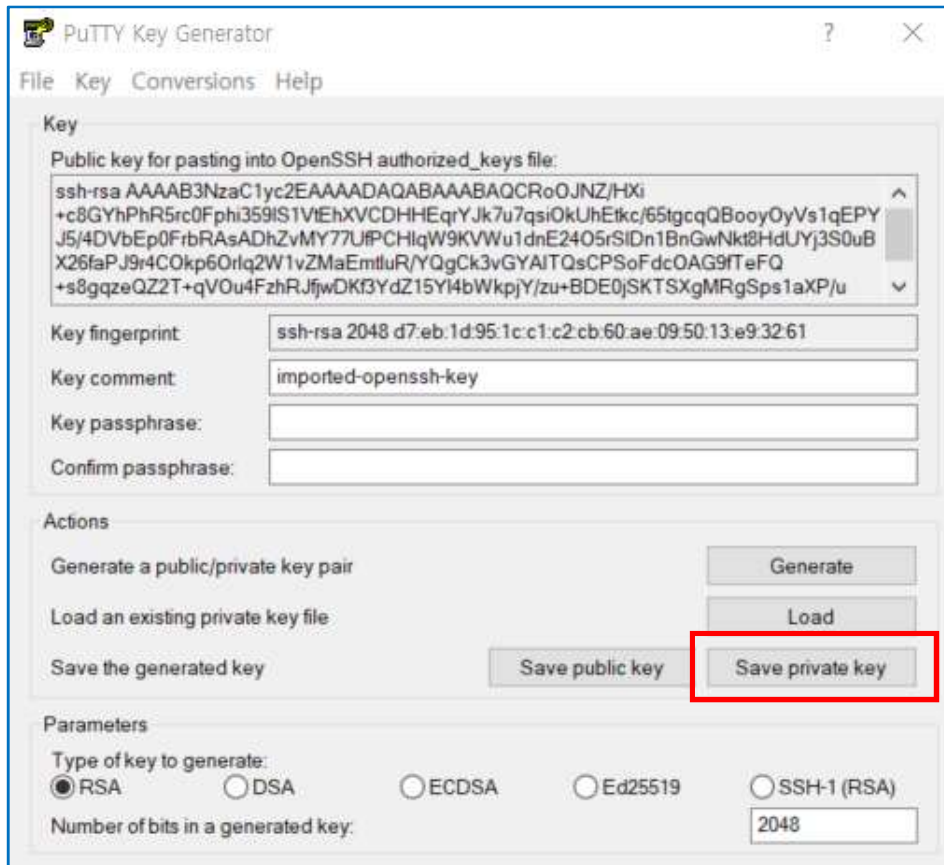
5. [PuTTY Key Generator]창에서 [Conversions] > [Import Key] 메뉴를 선택한다.



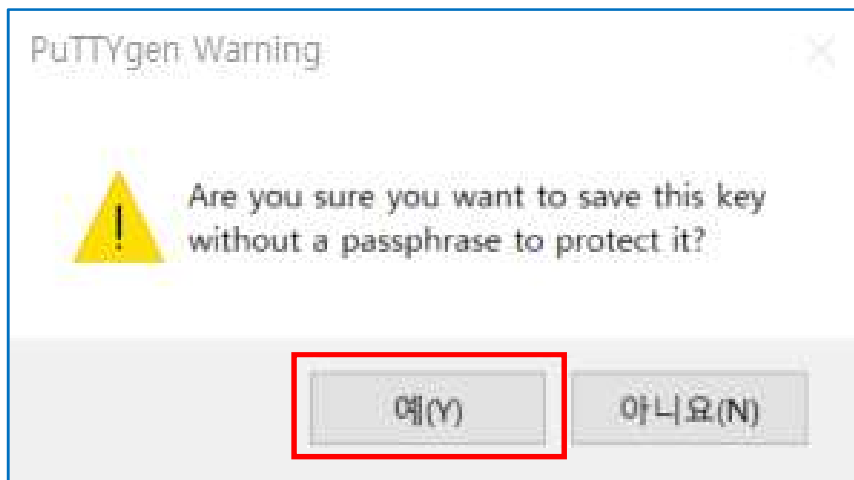
6. 이미 다운로드 받은 키 페어 파일(Docker-Ubuntu-RSAKey.pem)을 선택하고 [열기]를 클릭한다.



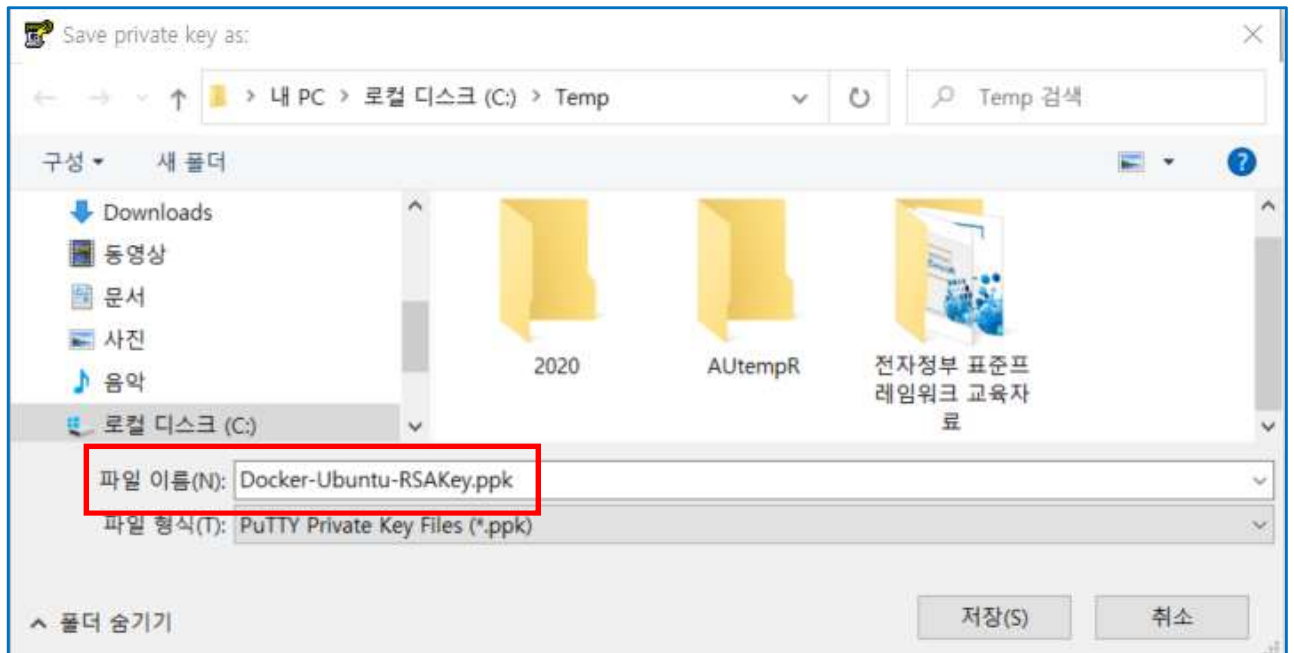
7. PuTTY로 Import할 Private Key의 생성을 위해 **[Save private key]** 버튼을 클릭한다.



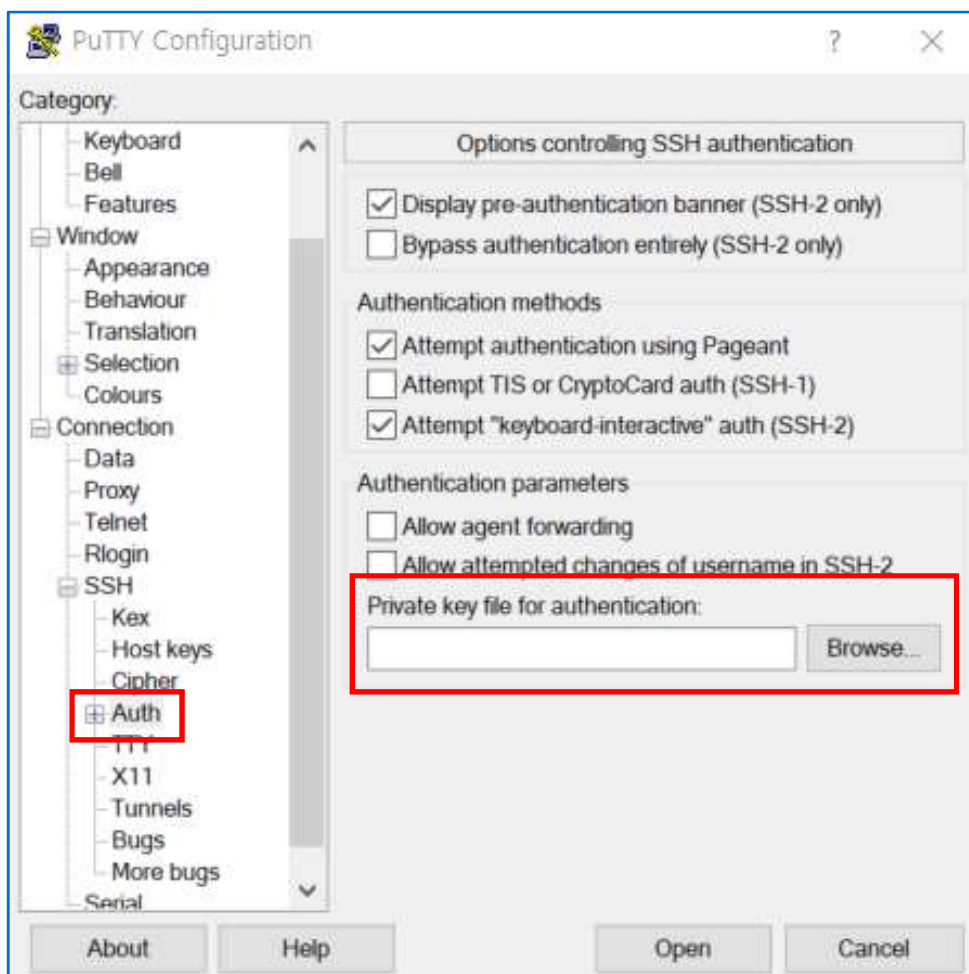
8. **[PuTTYgen Warning]** 창에서 **[예]**를 클릭한다.



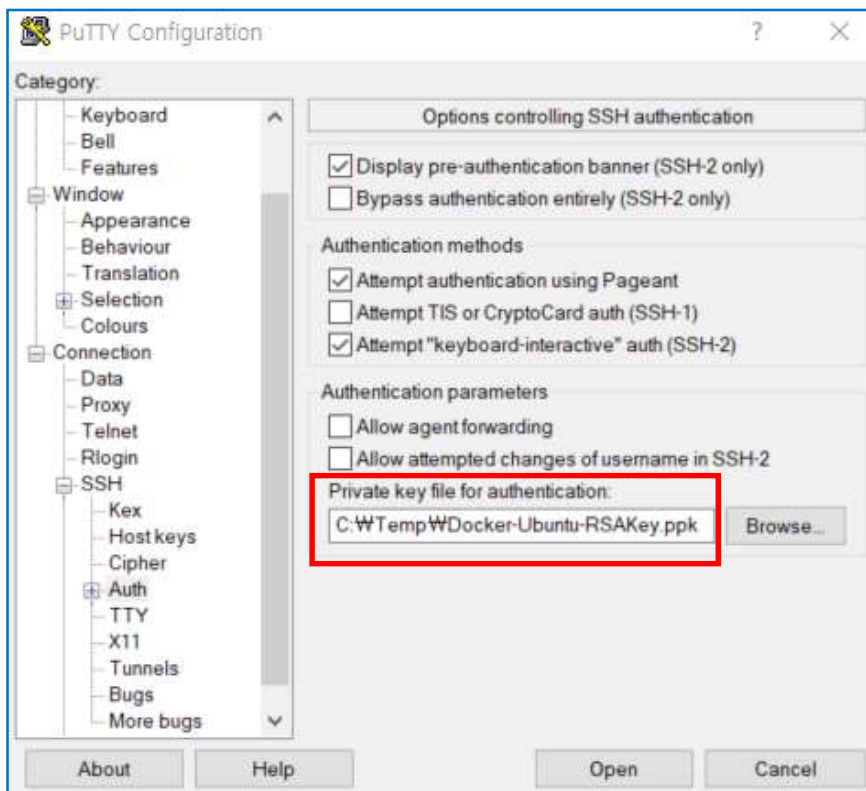
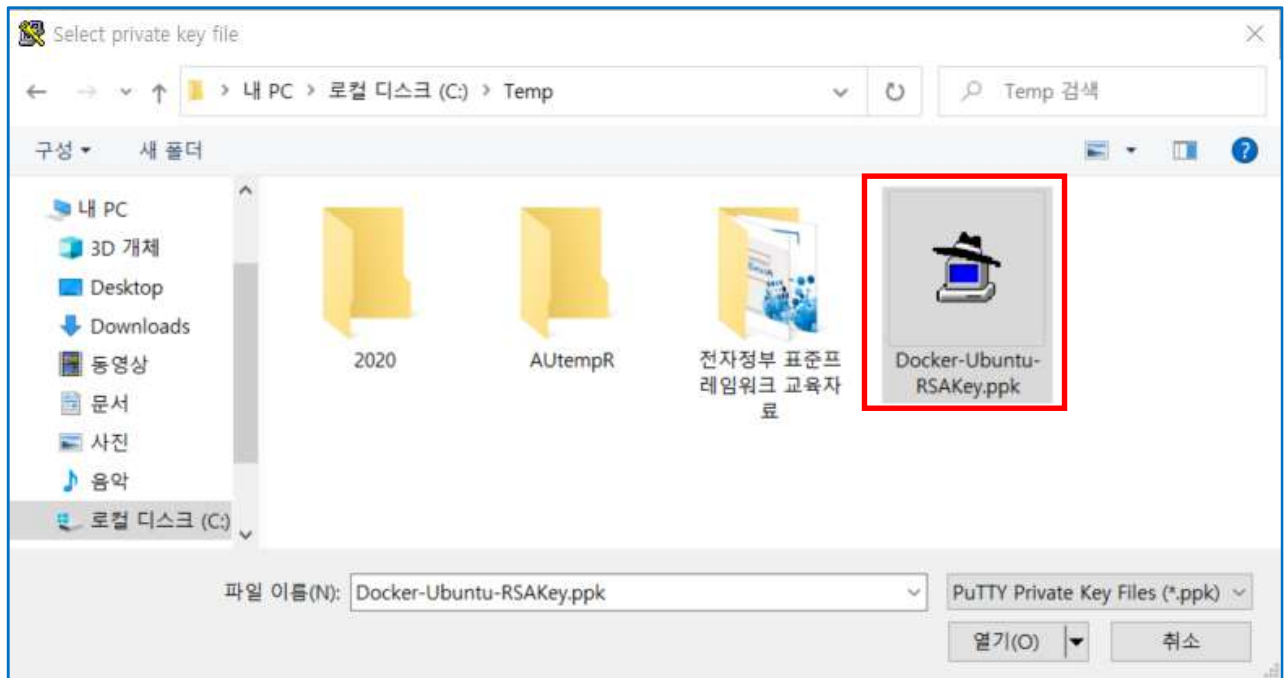
9. 이전에 pem 파일을 다운로드 받았던 동일한 폴더에 “**Docker-Ubuntu-RSAKey.ppk**” 파일을 저장하기 위해 **[저장]** 버튼을 클릭한다. 저장한 후, **[PuTTY Key Generator]**창은 닫는다.



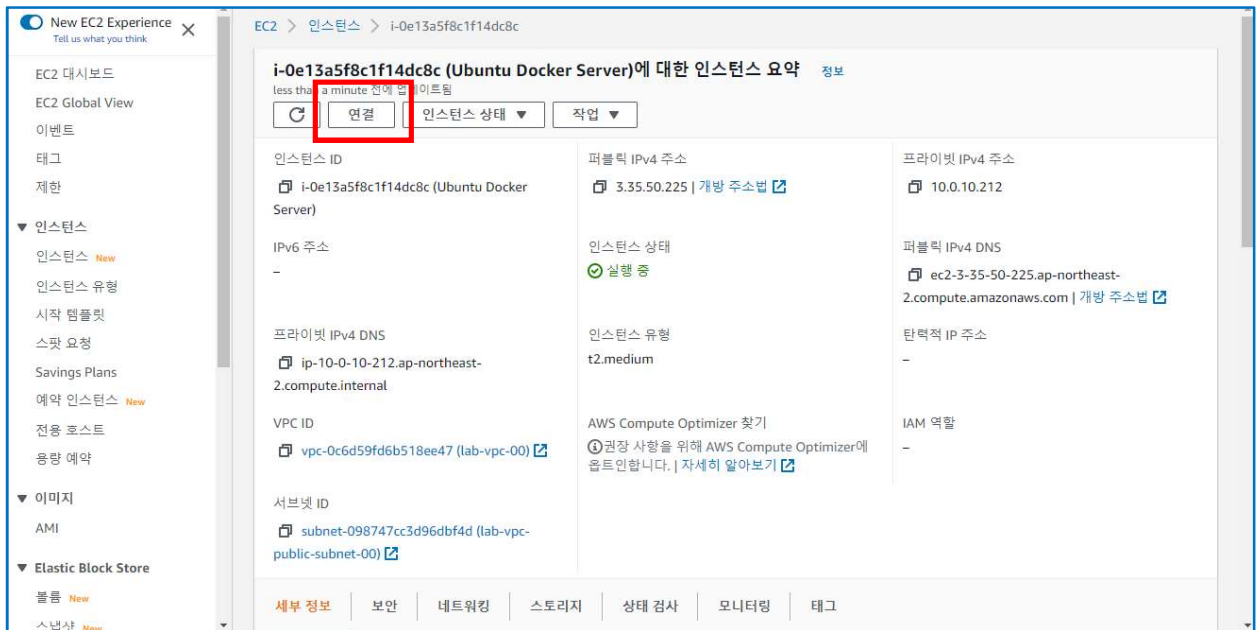
10. 이미 설치한 PuTTY 프로그램을 실행한 다음, **[Connection] > [SSH] > [Auth]** 메뉴의 “**Private key file for authentication:**”의 **[Browse...]** 버튼을 클릭한다.



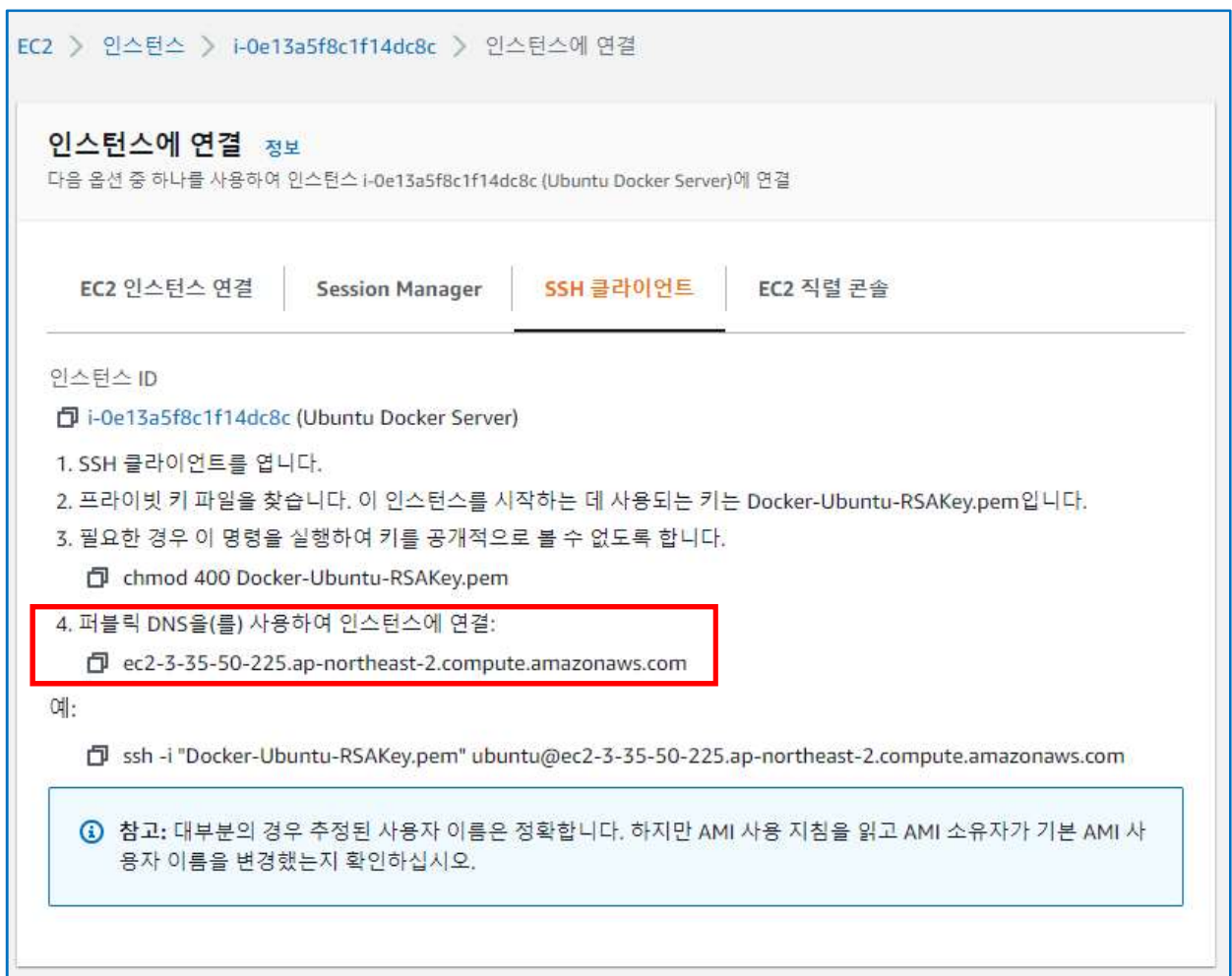
11. 위에서 이미 저장한 Private Key의 저장위치에서 “Dokcer-Ubuntu-RSAKey.ppk” 파일을 선택하고 [열기] 버튼을 클릭한다.



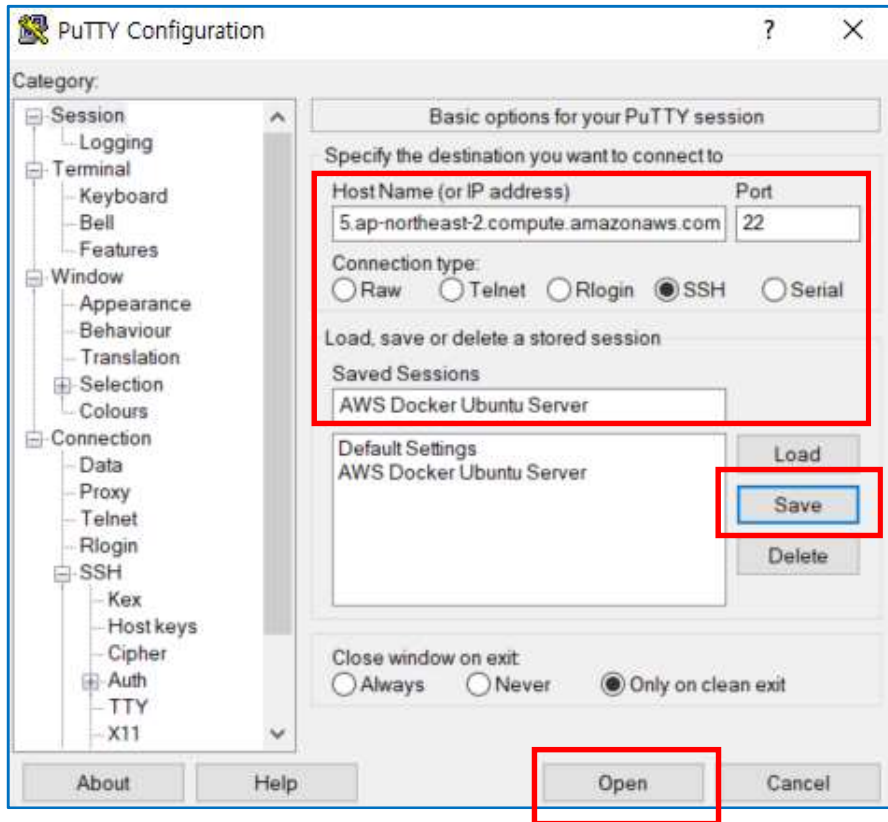
12. 다시 AWS 인스턴스 페이지로 돌아가서 이미 여러분이 생성한 Linux 인스턴스의 **[인스턴스 ID]**를 클릭하여 해당 인스턴스 요약페이지로 이동한다. 접속할 Linux 인스턴스 요약페이지에서 **[연결]** 버튼을 클릭한다.



13. **[SSH 클라이언트]** 탭을 클릭한다. 순서의 4번에 보면 “퍼블릭 DNS을(를) 사용하여 인스턴스에 연결” 아래에 있는 주소를 복사한다.



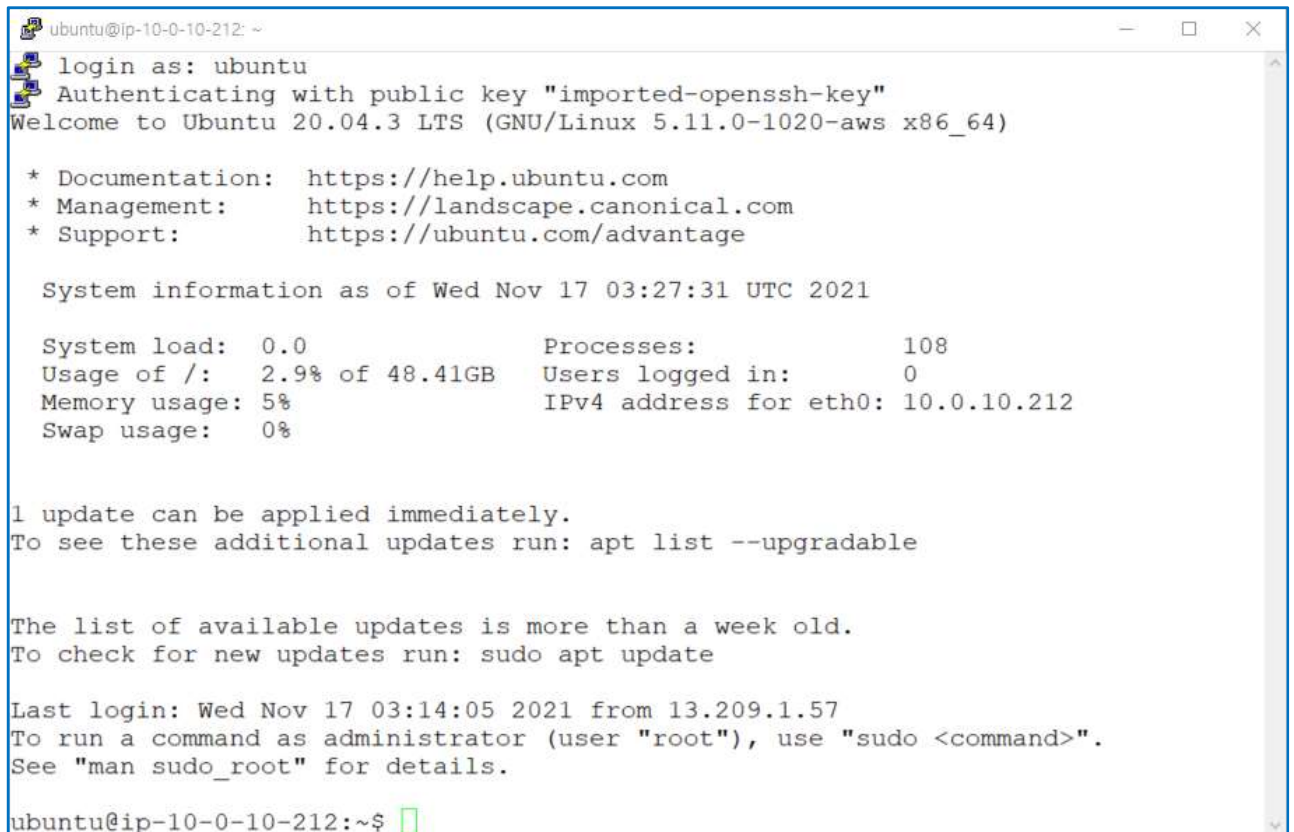
14. 다시 PuTTY 프로그램으로 돌아와서, [Session] 메뉴의 "Host Name(or IP address)"의 텍스트박스에 방금 복사한 주소를 붙여넣기 한다. 그리고 "Port"는 22번, "Connection type"은 SSH가 선택되어 있음을 확인한 다음, "Saved Sessions"의 항목에 "AWS Docker Ubuntu Server"라고 입력하고, [Save] 버튼을 클릭한다. 그리고 나서 마지막으로 [Open] 버튼을 클릭하여 Linux 인스턴스와 연결한다.



15. [PuTTY Security Alert]창에서 [예(Y)]를 선택한다.



16. AWS에 생성한 Linux 인스턴스와 원격으로 연결하는 창이 나타난다. [login as:] 에 "ubuntu"라고 입력하고 Enter key를 누른다.

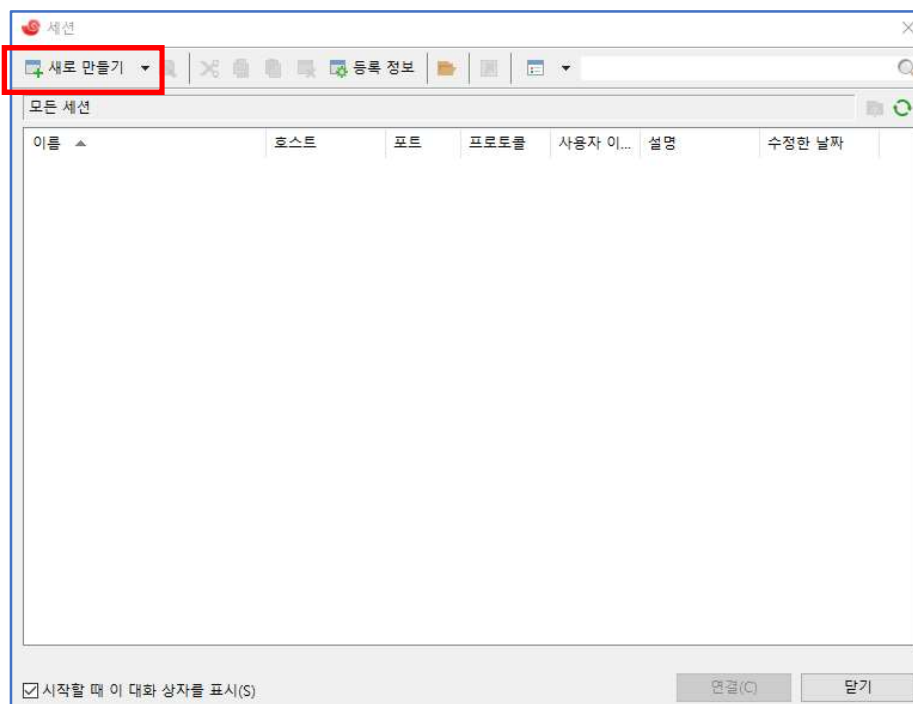


```
ubuntu@ip-10-0-10-212: ~  
login as: ubuntu  
Authenticating with public key "imported-openssh-key"  
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1020-aws x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
System information as of Wed Nov 17 03:27:31 UTC 2021  
  
System load:  0.0          Processes:            108  
Usage of /:   2.9% of 48.41GB Users logged in:      0  
Memory usage: 5%          IPv4 address for eth0: 10.0.10.212  
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: Wed Nov 17 03:14:05 2021 from 13.209.1.57  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-10-0-10-212:~$
```

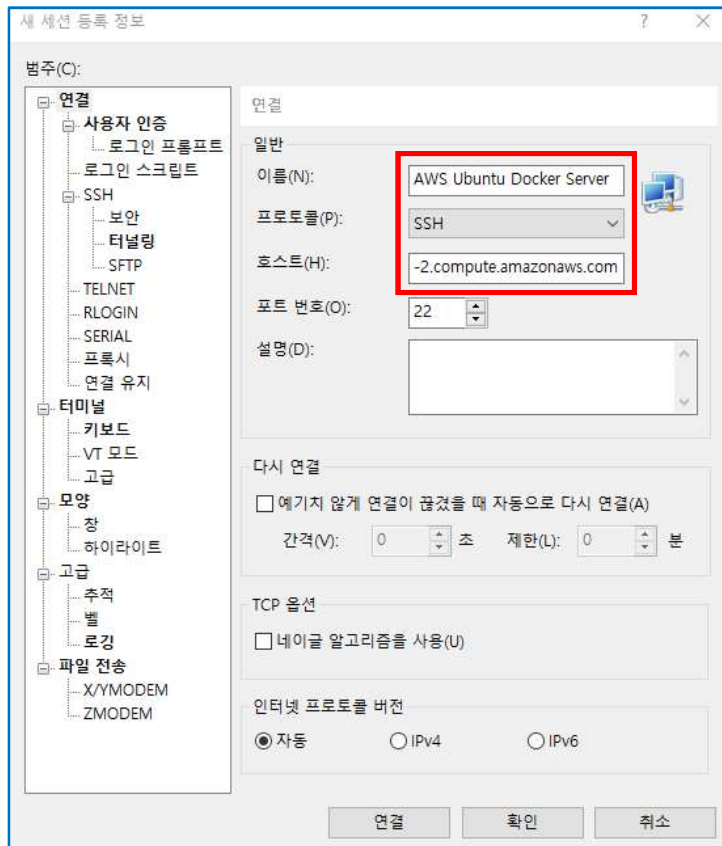
17. Linux 인스턴스 접속을 완료했다.

18. 참고로, Xshell로 접속할 때는 다음과 같다.

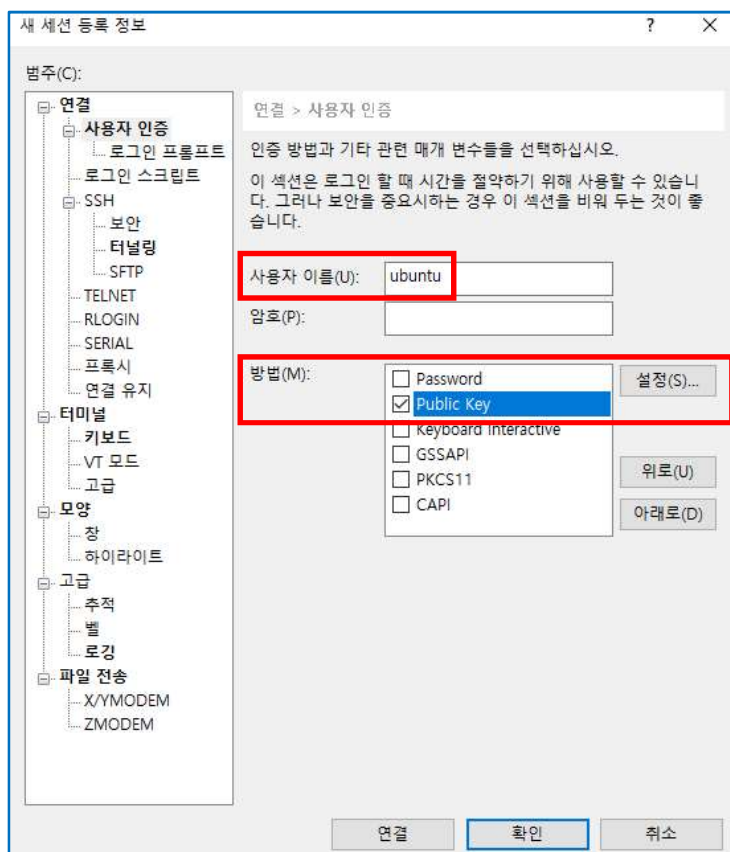
- A. 설치한 Xshell 프로그램을 실행하면 다음과 같이 새로운 세션 설정을 위한 [세션] 창이 나타난다. [새로 만들기] 버튼을 클릭한다.



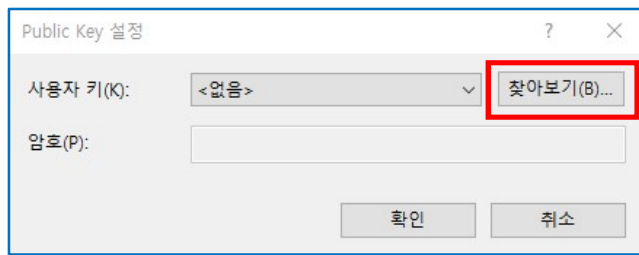
- B. 먼저 [연결]에서, [일반] > [이름]을 "AWS Ubuntu Docker Server"라고 입력하고, [호스트]는 방금 생성한 EC2 인스턴스의 [퍼블릭 IPv4 DNS] 값을 넣는다.



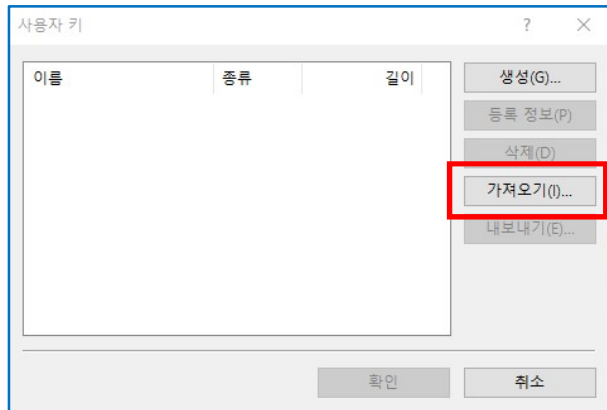
- C. 다음은, [연결] > [사용자 인증]을 선택한다. [사용자 이름]은 ubuntu이고, [방법]에 [Public Key]을 선택하고 [설정] 버튼을 클릭한다.



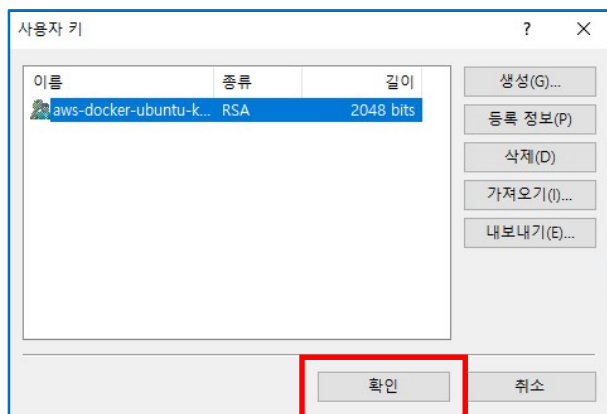
- D. **[Public Key 설정]** 창이 나타난다. **[찾아보기]**를 클릭한다.



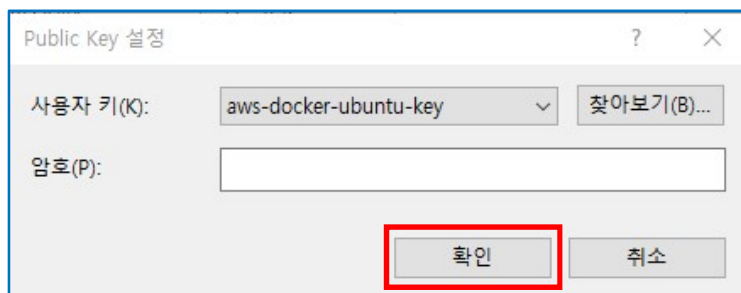
- E. **[사용자 키]**창에서 **[가져오기]**를 선택한다.



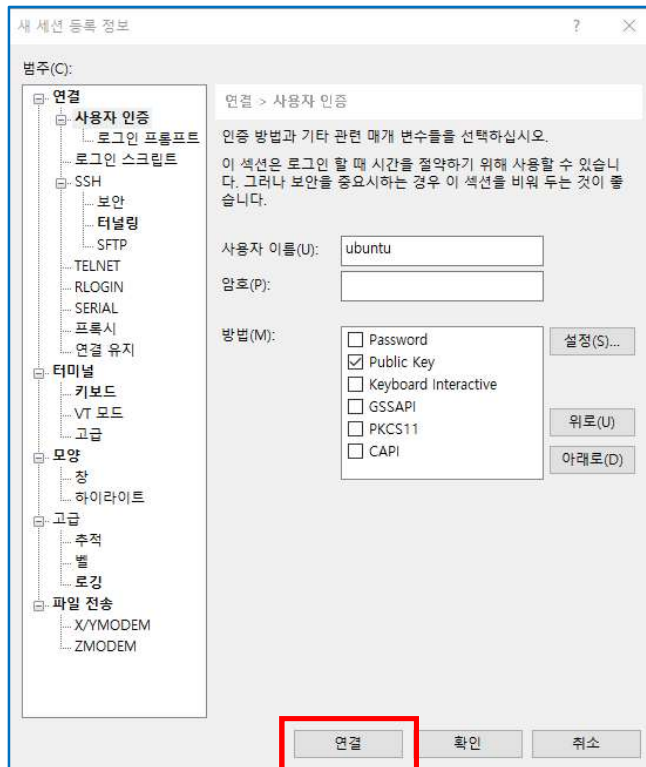
- F. 이미 다운로드 받은 pem 파일을 선택하면 다음 그림과 같다. 키를 선택하고 **[확인]** 버튼을 클릭한다.



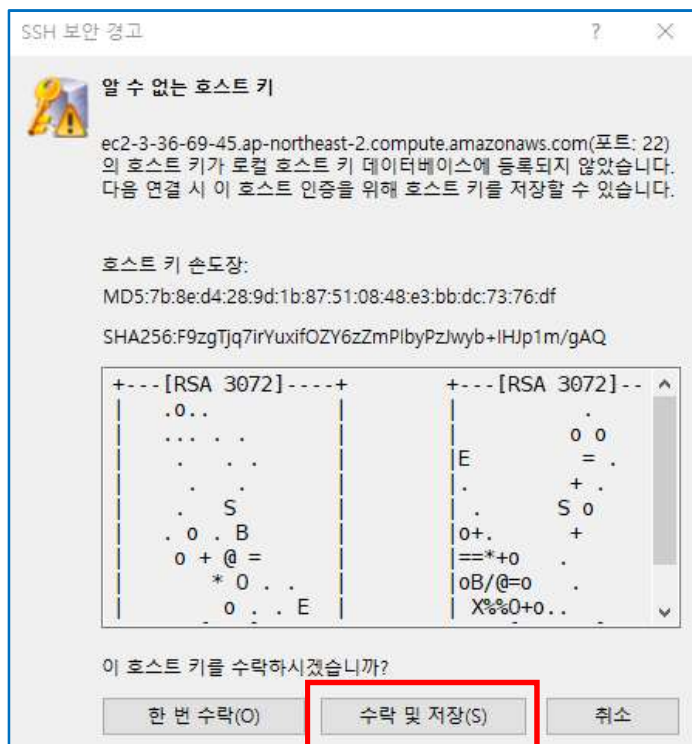
- G. 다시 **[Public Key 설정]**창으로 돌아왔다. **[암호]**는 넣지 않아도 된다. **[확인]** 버튼을 클릭한다.



H. 다시 [새 세션 등록 정보] 창으로 돌아왔다. [연결] 버튼을 클릭한다.



I. [SSH 보안 경고] 창이 나타난다. [수락 및 저장] 버튼을 클릭한다.



J. 연결에 성공하였다.

The screenshot shows an Xshell terminal window titled "AWS Ubuntu Docker Server - ubuntu@ip-10-0-10-250: ~ - Xshell 7 (Free for Home/School)". The terminal output includes the following text:

```
System information as of Wed Nov 17 06:21:49 UTC 2021

System load: 0.16          Processes:           116
Usage of /:  2.8% of 48.41GB Users logged in:       0
Memory usage: 5%          IPv4 address for eth0: 10.0.10.250
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-10-0-10-250:~$
```

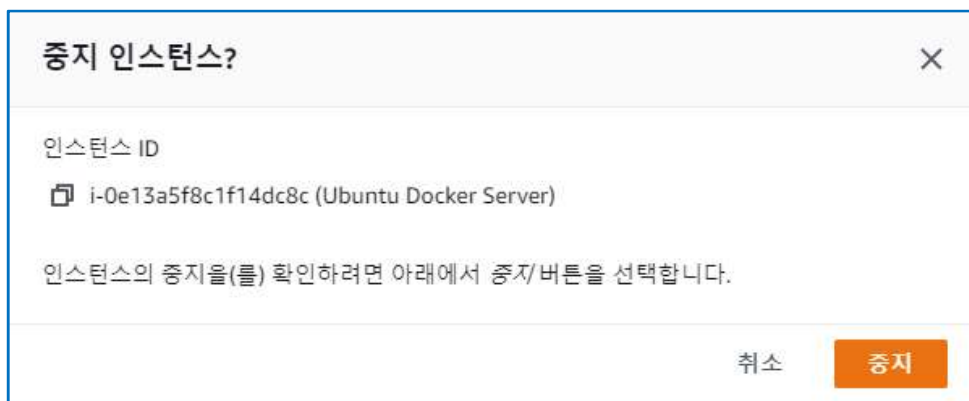
On the left side of the terminal, there is a sidebar titled "세션 관리" (Session Management) showing a list of sessions. Below it, a table provides details for the selected session:

이름	호스트	포트	프로토콜	사용자 이름	설명
AWS Ubuntu Doc...	ec2-3-36-69-45.a...	22	SSH	ubuntu	

The status bar at the bottom of the window indicates the connection is established via SSH2 on a terminal window, with a resolution of 98x27 and a refresh rate of 27.26. It also shows "1 세션" (1 session) and other system metrics.

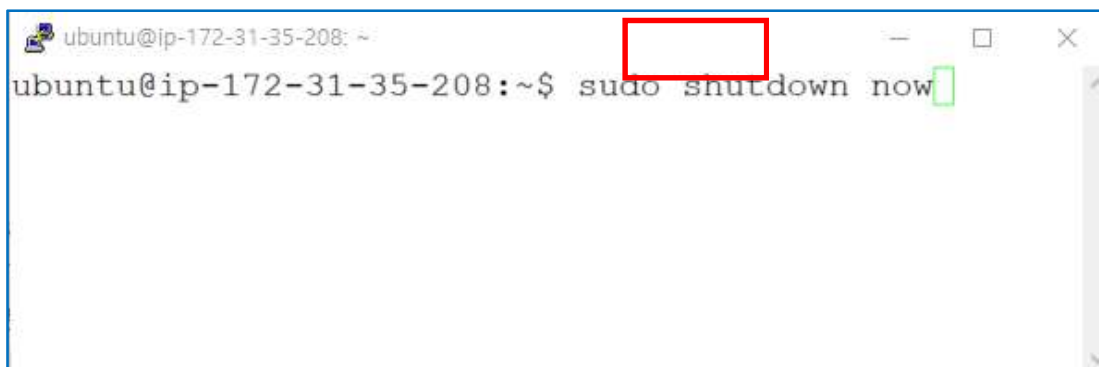
Task6. Linux 서버 시작, 중지하기

1. 방금 생성한 Linux Server 인스턴스를 중지시키기 위해서 해당 인스턴스 요약창에서 [인스턴스 상태] > [인스턴스 중지]를 선택한다. 그리고 [중지 인스턴스]창에서 [중지]를 선택한다.



2. 또는 PuTTY 창에서 다음의 명령어를 수행함으로써 서버를 중지시킬 수 있다.

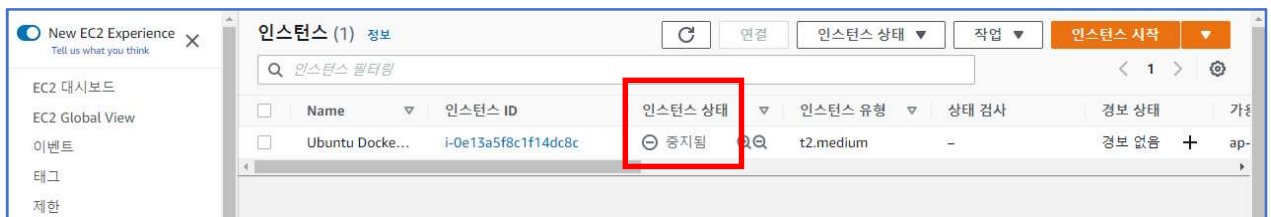
\$ sudo shutdown now



3. Linux server 인스턴스와 연결이 종료되었다.



4. 잠시 후 [인스턴스] 페이지에서 해당 Linux Server 인스턴스가 "중지됨"을 확인할 수 있다.



5. 다시 해당 인스턴스를 시작하려면 [인스턴스 요약] 페이지에서 [인스턴스 시작]을 선택하면 된다.



6. 다시 연결하려면 해당 인스턴스의 [인스턴스 요약] 페이지에서 [인스턴스 유형]이 "실행 중"임을 확인한 후, 위의 과정을 다시 실행하면 된다. 다시 서버를 연결할 때에는 PuTTY 창의 [Session] 메뉴의 "Host Name(or IP address)"의 텍스트박스에 [퍼블릭 IPv4 DNS]의 값을 복사해서 붙여넣고 [Open] 버튼을 클릭하면 된다.

New EC2 Experience
Tell us what you think

EC2 대시보드

EC2 Global View

이벤트

태그

제한

인스턴스

인스턴스 New

인스턴스 유형

시작 템플릿

스팟 요청

Savings Plans

예약 인스턴스 New

전용 호스트

용량 예약

EC2 > 인스턴스 > i-0e13a5f8c1f14dc8c

i-0e13a5f8c1f14dc8c (Ubuntu Docker Server)에 대한 인스턴스 요약

정보

refresh

연결

인스턴스 상태 ▼

작업 ▼

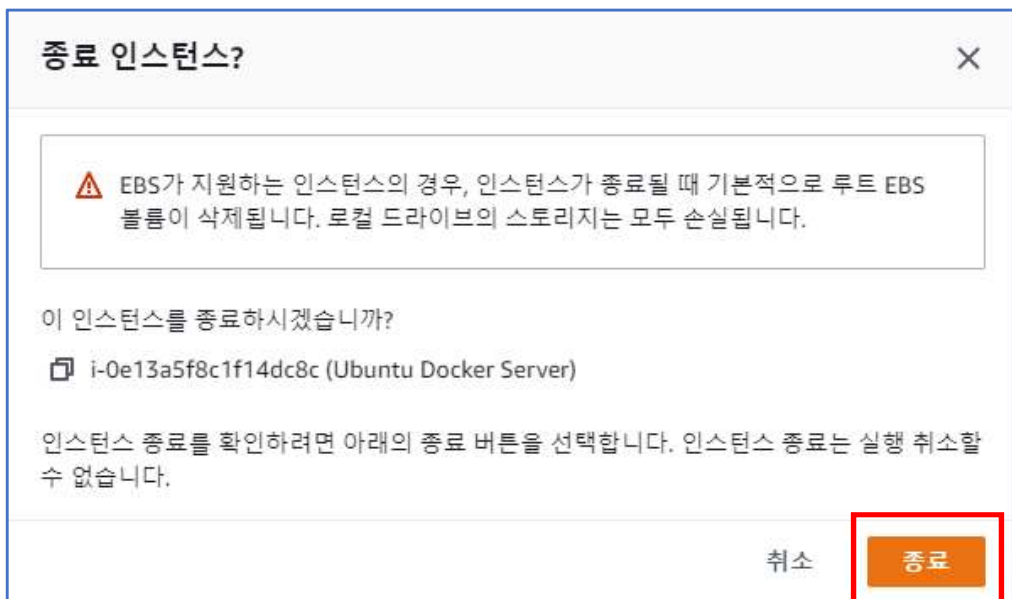
인스턴스 ID i-0e13a5f8c1f14dc8c (Ubuntu Docker Server)	퍼블릭 IPv4 주소 3.34.53.82 개방 주소법	프라이빗 IPv4 주소 10.0.10.212
IPv6 주소 -	인스턴스 상태 실행 중	퍼블릭 IPv4 DNS ec2-3-34-53-82.ap-northeast-2.compute.amazonaws.com 개방 주소법
프라이빗 IPv4 DNS ip-10-0-10-212.ap-northeast-2.compute.internal	인스턴스 유형 t2.medium	탄력적 IP 주소 -
VPC ID vpc-0c6d59fd6b518ee47 (lab-vpc-00) 링크	AWS Compute Optimizer 찾기 권장 사항을 위해 AWS Compute Optimizer에 업로드합니다. 자세히 알아보기	IAM 역할 -

Task7. Linux Server 인스턴스 영구 삭제하기

1. 해당 인스턴스의 [인스턴스 요약] 페이지에서 [인스턴스 유형]이 "중지됨"을 확인 한 다음, [인스턴스 상태]에서 [인스턴스 종료]를 선택한다.



2. [인스턴스 종료]를 선택하면 아래의 그림과 같이 [종료 인스턴스]창이 나타나고 여기서 [종료]를 클릭한다.



3. 잠시 뒤, [인스턴스] 페이지에서 확인해 보면 해당 인스턴스가 "종료됨" 상태임을 알 수 있다.

