

AWSKRUG BEGINNER GROUP

# VPC, 서브넷, EC2 한 번에 이해하기:

AWS를 이용한 서버 배포 완벽 가이드

2024.09.27 김수진



# 발표자 소개



김수진

AUSG 8기, ACC 2기

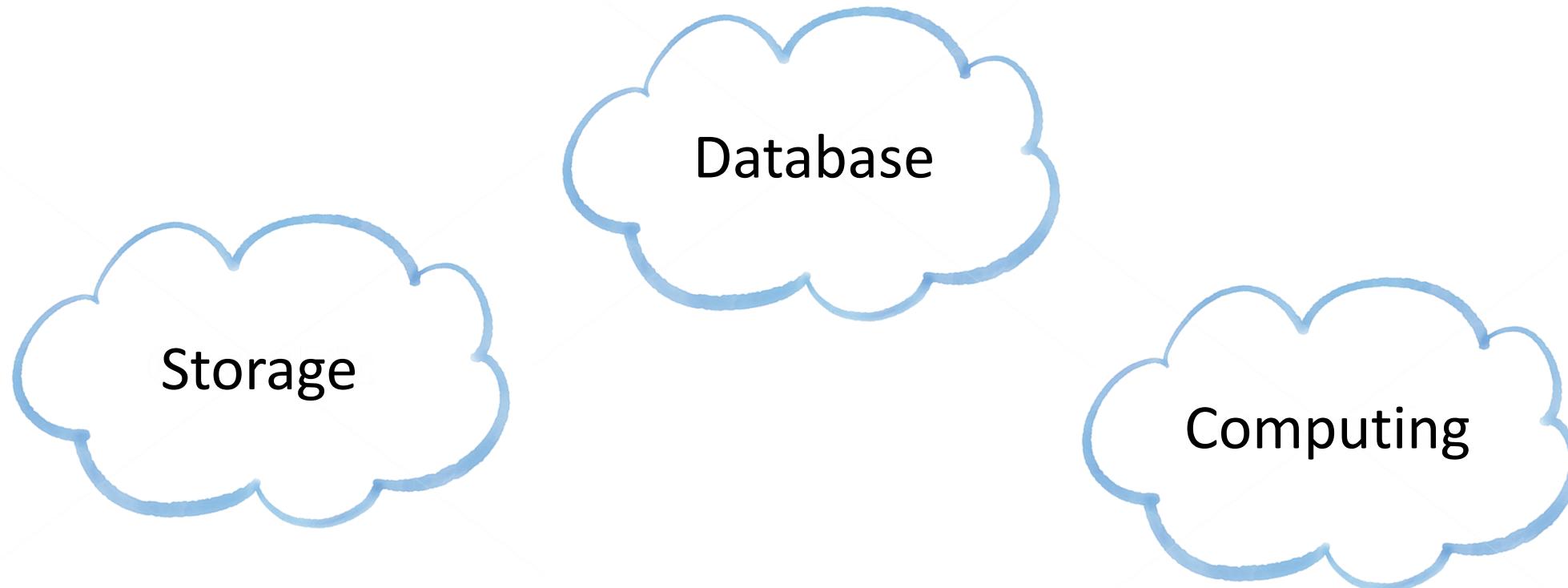
경희대 컴퓨터공학과 3학년 재학 중

# Contents

- 01       Cloud Computing
- 02       AWS
- 03       VPC, Subnet
- 04       EC2
- 05       서버 배포 가이드

# Cloud Computing

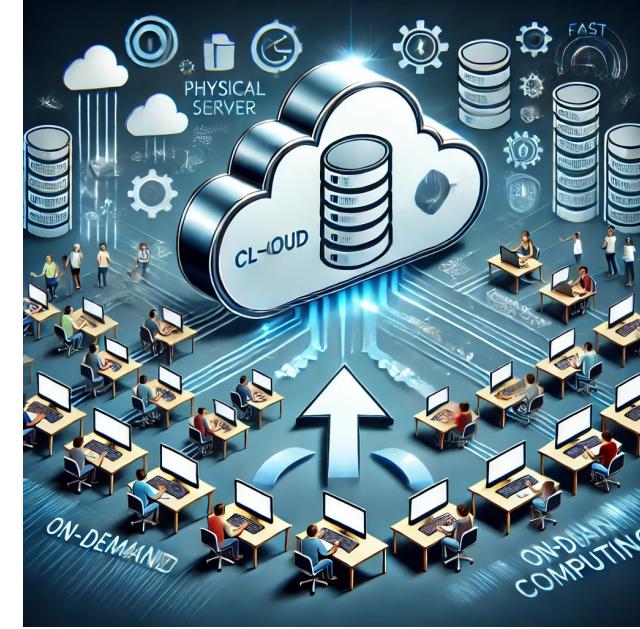
On-demand로 내가 자원을 쓴 만큼 비용을 지불하는 서비스



# on-premises VS on-demand



직접 물리적인 서버와 인프라를 구축해 사용하는 방식



AWS와 같은 클라우드 제공업체가 인프라를 구축해놓고,  
사용자는 원하는 만큼 쓰고 쓴 만큼만 비용을 지불하는 방식

# 클라우드 컴퓨팅을 쓰는 이유

→ On-Demand!

- 처음부터 과하게 인프라를 구성하지 않아도 됨
- 서비스 규모가 커지면 확장하면 된다!
- 다양한 서비스들을 빠르게 구축할 수 있음

# AWS

## 클라우드 컴퓨팅 제공 업체



32 Regions  
108 Availability Zones

→ 높은 가용성, 안정성 보장!

AWS는 자체적으로 구축해놓은 데이터센터를 운영하고,  
사용자는 편하게 빌려 쓸 수 있다!

필요한 만큼만 쓰고, 쓴 만큼 비용을 지불하면 됨

서비스 확장, 축소 시 편하게 스케일링 가능

# Seoul Region



## Region

: 가용영역이 3개 이상 구성된 지리적인 영역



## Availability Zone(AZ)

: 한 개 이상의 데이터센터로 이뤄진 논리적인 영역

### 가용 영역 정보

서브넷이 상주할 영역을 선택합니다. 선택하지 않으면 Amazon이 자동으로 선택합니다.

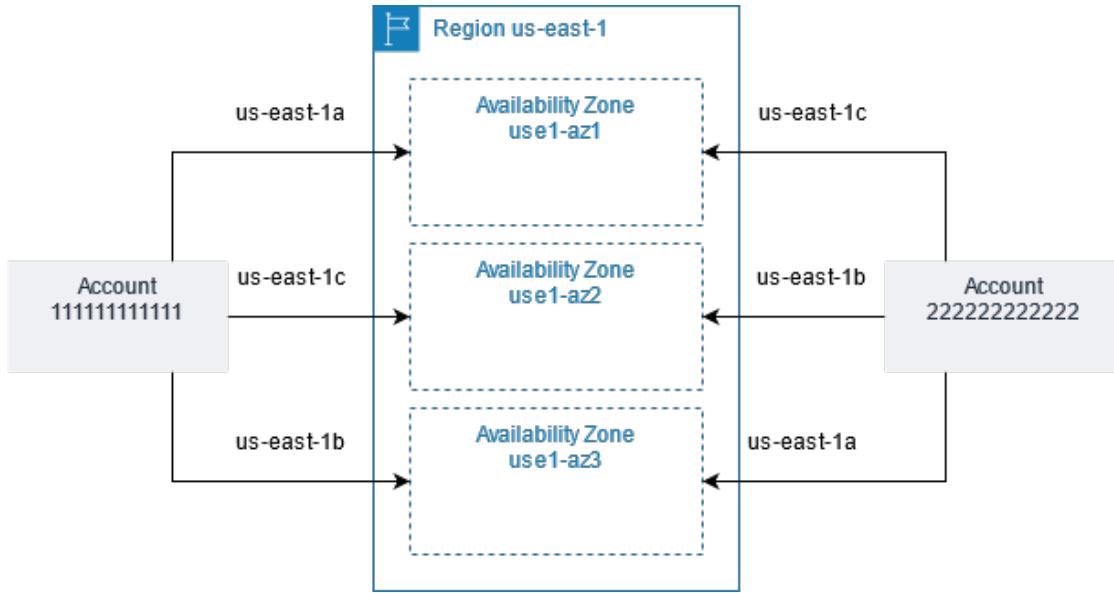
|  |                |
|--|----------------|
| 기본 설정 없음   | ▲              |
| <input type="text"/>   |                |
| 기본 설정 없음   |                |
| 아시아 태평양 (서울) / ap-northeast-2a<br>ID: apne2-az1 네트워크 경계 그룹: ap-northeast-2 | ap-northeast-2 |
| 아시아 태평양 (서울) / ap-northeast-2b<br>ID: apne2-az2 네트워크 경계 그룹: ap-northeast-2 | ap-northeast-2 |
| 아시아 태평양 (서울) / ap-northeast-2c<br>ID: apne2-az3 네트워크 경계 그룹: ap-northeast-2 | ap-northeast-2 |
| 아시아 태평양 (서울) / ap-northeast-2d<br>ID: apne2-az4 네트워크 경계 그룹: ap-northeast-2 | ap-northeast-2 |

# Seoul Region



그럼 가용영역 a에 너무 몰리는 거 아니야..?

# AZ는 랜덤 매핑!



## 가용 영역 정보

서브넷이 상주할 영역을 선택합니다. 선택하지 않으면 Amazon

아시아 태평양 (서울) / ap-northeast-2a



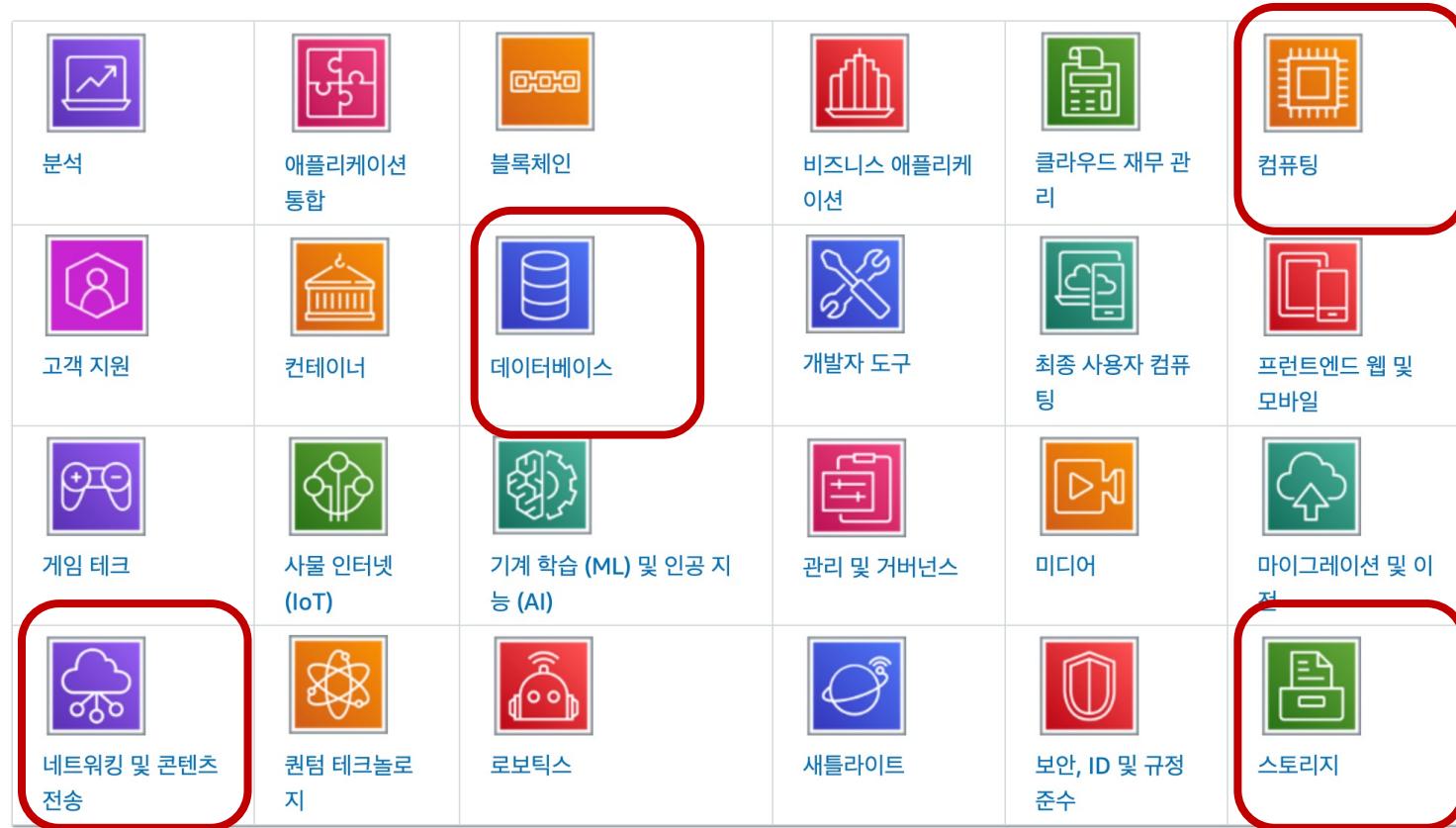
용인 데이터센터



목동 데이터센터

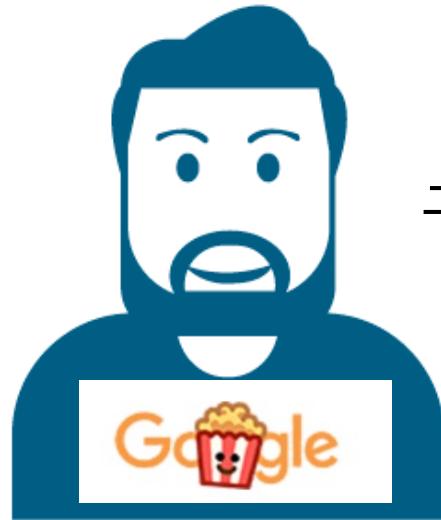
똑같은 가용영역 a라도, 내가 선택한 가용영역 a의 데이터센터는 다른 사용자의 가용영역 a 데이터센터와 다를 수 있다!

# AWS 서비스



왜 EC2, VPC, RDS, S3, Route53.. 들이 유명한 거지..?

# 브라우저에서의 사용자 요청 처리 과정



구글!



# 사용자 요청 처리 과정



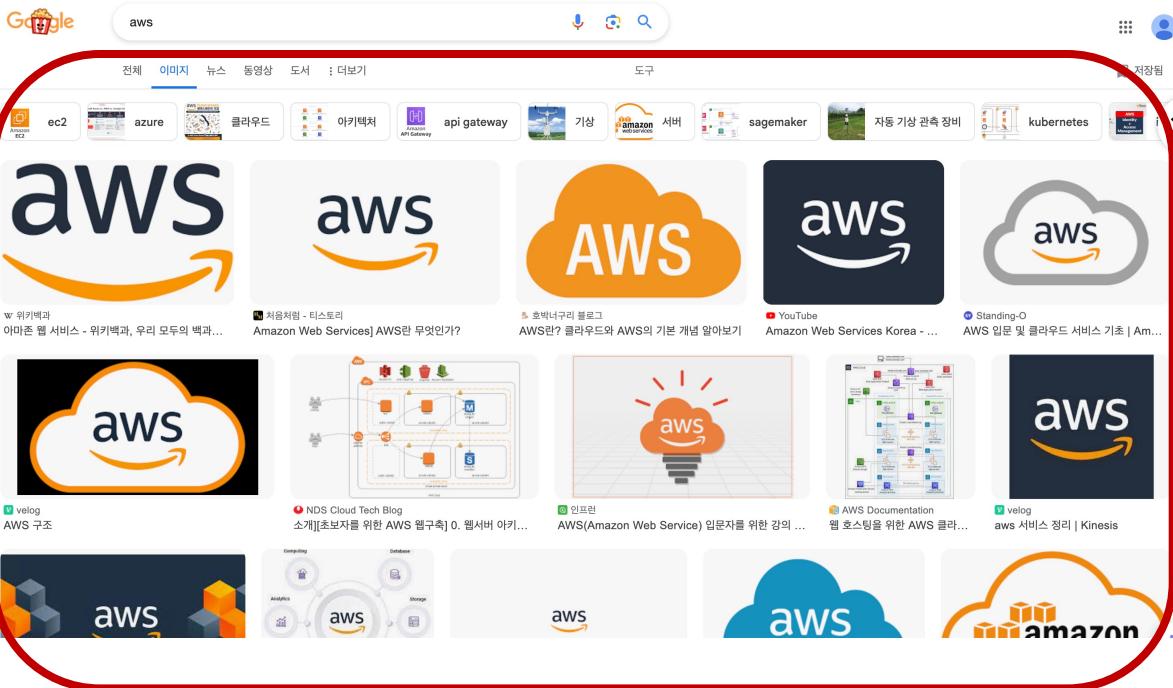
# 사용자 요청 처리 과정

A screenshot of a Google search results page. The search bar at the top contains the query "aws". Below the search bar, there are navigation links: 전체, 이미지, 뉴스, 동영상, 쇼핑, 도서, 지도, 더보기, and 도구. The main content area shows search results for AWS. The first result is a link to the "Amazon Web Services" website, titled "클라우드 서비스 | 클라우드 컴퓨팅 솔루션 - AWS". The snippet below the title reads: "Amazon Web Services는 안정성이고 확장 가능하며 저렴한 클라우드 컴퓨팅 서비스를 제공합니다. 무료로 가입하여 사용량에 따라 요금을 지불하세요." Below this, there are several other search results: "AWS Management Console", "Amazon EC2", "AWS 클라우드", "AWS 요금", and "새로운 소식". A red oval highlights the first search result, and a black double-headed arrow points from this result to a blue cylinder icon representing a database.



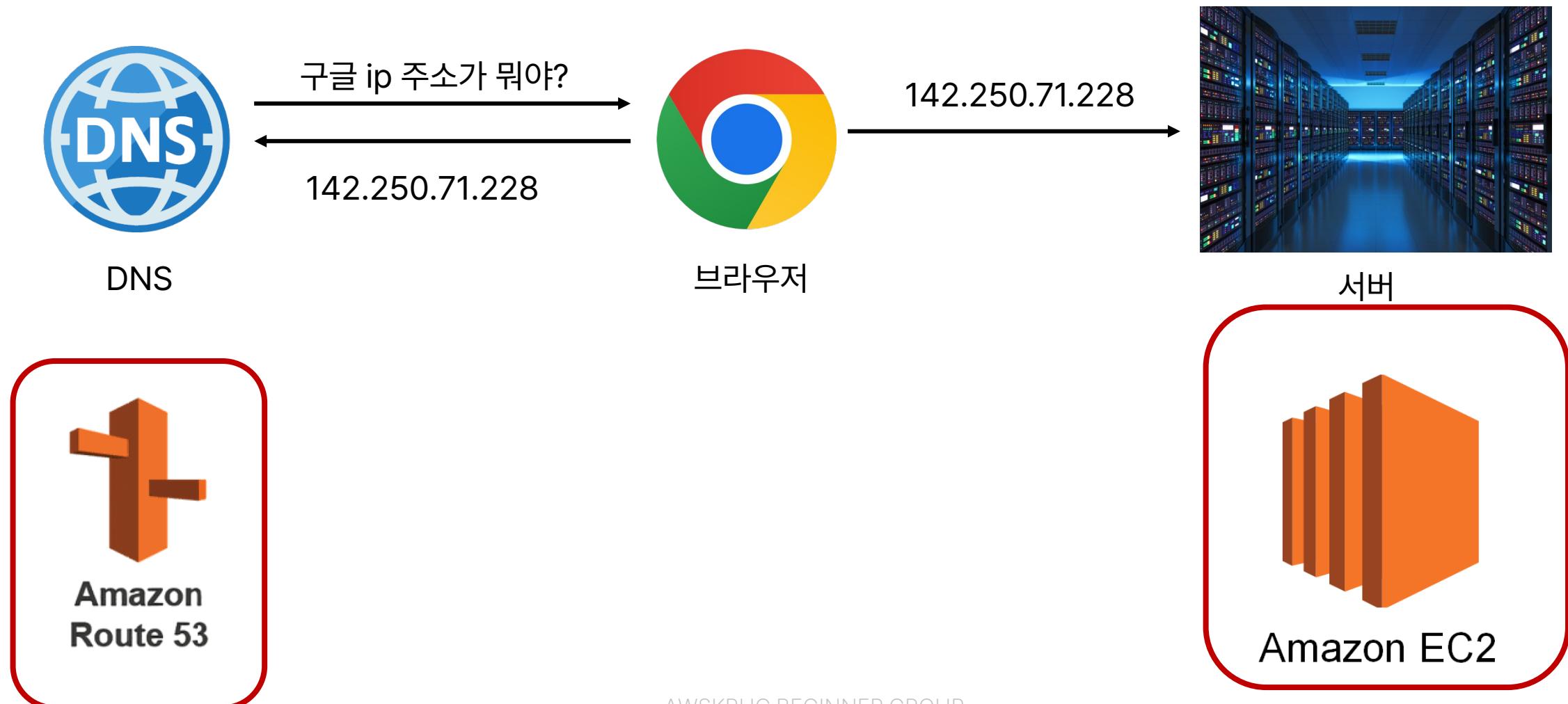
데이터베이스

# 사용자 요청 처리 과정

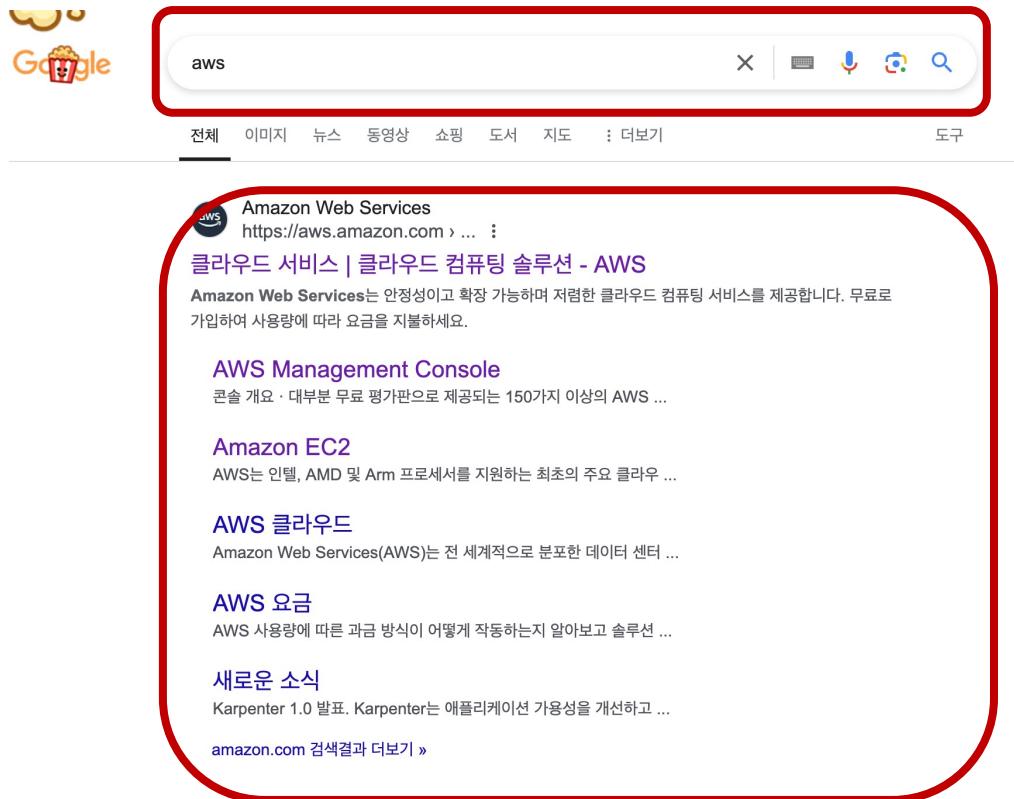


스토리지

# 사용자 요청 처리 과정



# 사용자 요청 처리 과정

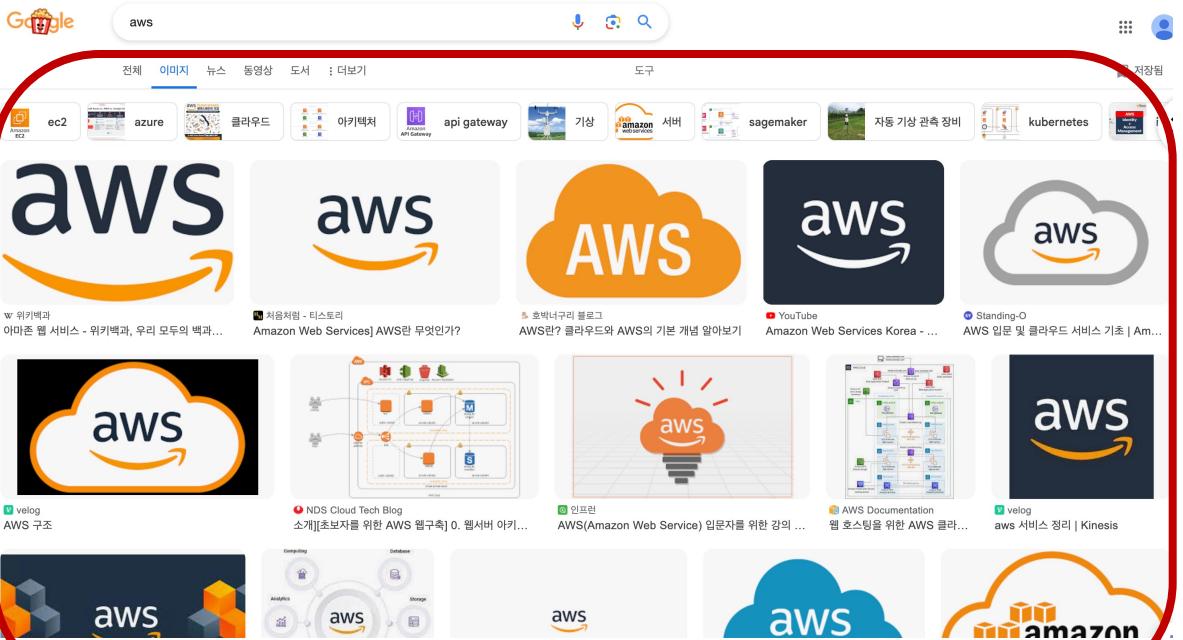


데이터베이스



Amazon RDS

# 사용자 요청 처리 과정



스토리지



# EC2

Elastic Compute Cloud



가상의 서버를 운영할 수 있는 서비스

나의 요구사항에 맞게 커스터마이징 가능!

운영체제

- 리눅스
- 윈도우
- macOS

CPU

- Arm
- x86

# EC2 Instance

: 가상의 컴퓨팅 환경

페이지 콘텐츠

## 범용

범용 인스턴스는 균형 있는 컴퓨팅, 메모리 및 네트워킹 리소스를 제공하며, 다양한 여러 워크로드에 사용할 수 있습니다. 이 인스턴스는 웹 서버 및 코드 리포지토리와 같이 이러한 리소스를 동등한 비율로 사용하는 애플리케이션에 적합합니다.

|     |     |          |     |     |     |     |      |     |    |     |      |     |
|-----|-----|----------|-----|-----|-----|-----|------|-----|----|-----|------|-----|
| M7g | M7i | M7i-Flex | M7a | Mac | M6g | M6i | M6in | M6a | M5 | M5n | M5zn | M5a |
| M4  | T4g | T3       | T3a | T2  |     |     |      |     |    |     |      |     |

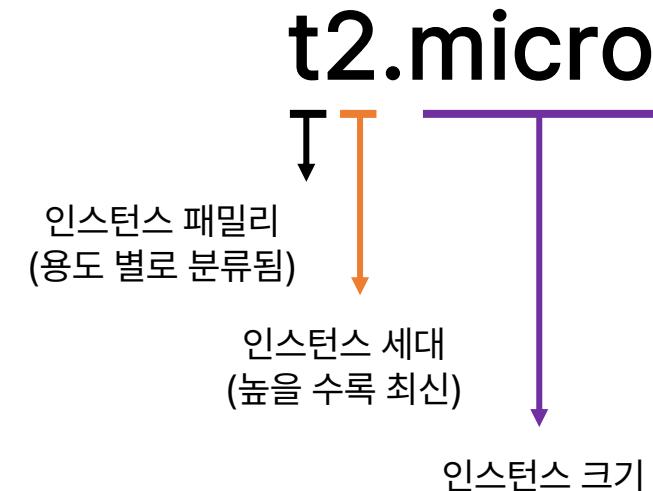
[Amazon EC2 T2 인스턴스](#)는 기본 수준의 CPU 성능과 더불어 기본 수준을 넘어 버스트 할 수 있는 기능을 제공하는 버스트 가능 성능 인스턴스입니다.

T2 무제한 인스턴스는 워크로드가 필요로 하는 한 높은 CPU 성능을 유지할 수 있습니다. 대부분의 범용 워크로드에서 T2 무제한 인스턴스는 추가 비용 없이 충분한 성능을 제공합니다. 인스턴스 실행에 장기간 높은 CPU 사용률이 필요한 경우, vCPU-시간당 5센트의 추가 고정 요금으로 인스턴스를 실행할 수 있습니다.

기본 성능과 순간 성능 기능은 CPU 크레딧에 의해 좌우됩니다. T2 인스턴스는 인스턴스 크기에 따라 정해진 비율로 CPU 크레딧을 계속 받게 되며, 유휴 상태일 때 CPU 크레딧을 축적하고 활성 상태일 때 CPU 크레딧을 사용하게 됩니다. T2 인스턴스는 마이크로 서비스, 자연 시간이 짧은 대화식 애플리케이션, 중소형 데이터베이스, 가상 데스크톱, 개발, 빌드 및 스테이징 환경, 코드 리포지토리 및 제품 프로토타입을 비롯하여 다양한 범용 워크로드에 적합합니다. 자세한 내용은 [버스트 가능 성능 인스턴스](#)를 참조하십시오.

기능:

- 최대 3.3GHz의 인텔 제온 스케일러블 프로세서(하스웰 E5-2676 v3 또는 브로드웰 E5-2686 v4)



# EC2 Instance 상태

인스턴스 (1) 정보

인스턴스를 속성 또는 (case-sensitive) 태그로 찾기

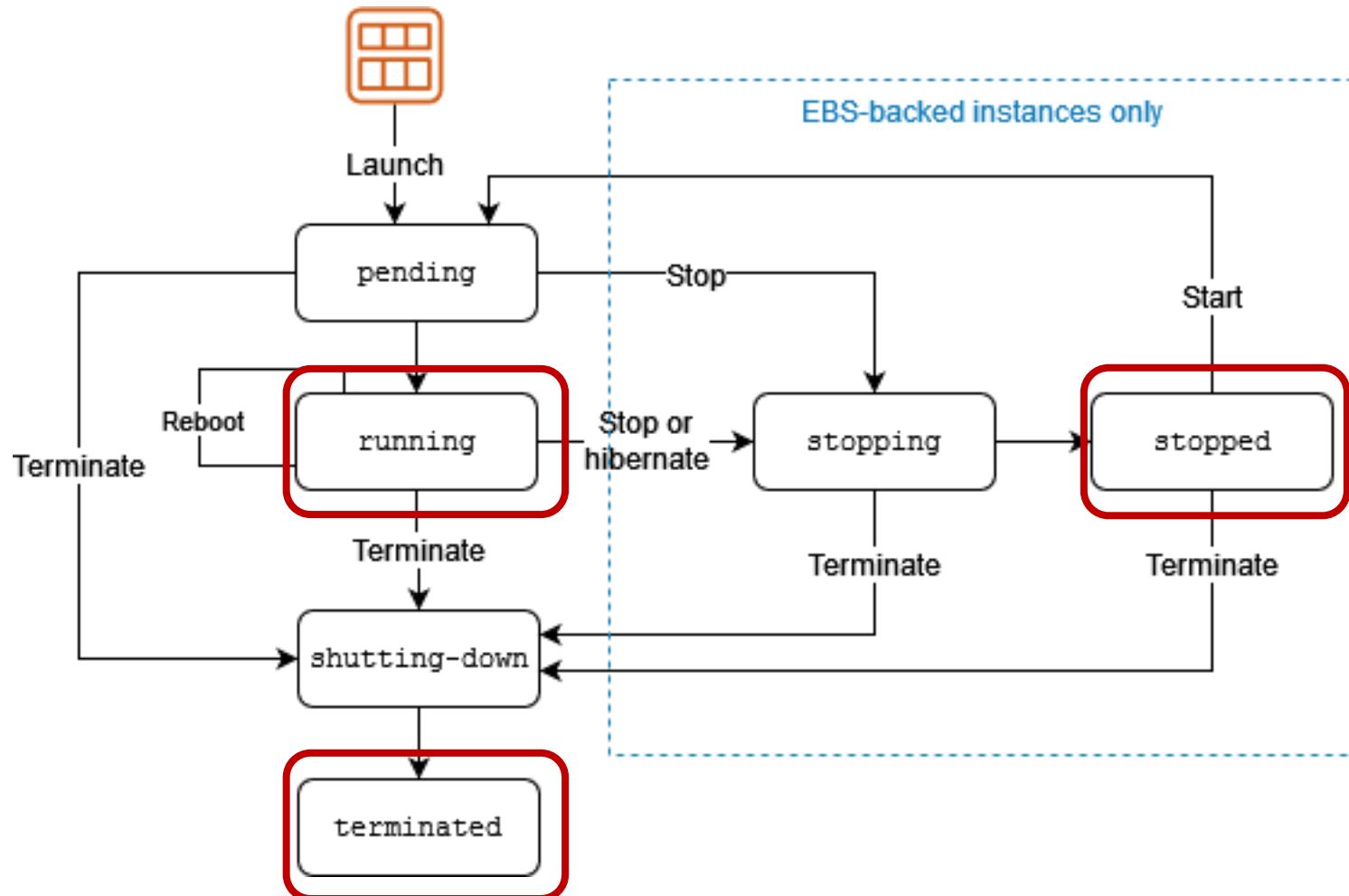
| <input type="checkbox"/> | Name   | ▼ | 인스턴스 ID                             | 인스턴스 상태     | ▼ |
|--------------------------|--------|---|-------------------------------------|-------------|---|
| <input type="checkbox"/> | simter |   | <a href="#">i-09c3172d56b709c23</a> | _minus_ 중지됨 |   |

인스턴스 상태 ▲

작업 ▾

- 인스턴스 중지
- 인스턴스 시작
- 인스턴스 재부팅
- 인스턴스 최대 절전 모드
- 인스턴스 종료(삭제)

# EC2 Instance 상태



# AMI(Amazon Machine Image)

: EC2의 가상 서버 템플릿

▼ 애플리케이션 및 OS 이미지(Amazon Machine Image) 정보

AMI는 인스턴스를 시작하는 데 필요한 소프트웨어 구성(운영 체제, 애플리케이션 서버 및 애플리케이션)이 포함된 템플릿입니다. 아래에서 찾고 있는 항목이 보이지 않으면 AMI를 검색하거나 찾아보세요.

수천 개의 애플리케이션 및 OS 이미지를 포함하는 전체 카탈로그 검색

최근 사용 Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat SUSE

AWS Marketplace 및 커뮤니티의 AMI 포함

더 많은 AMI 찾아보기

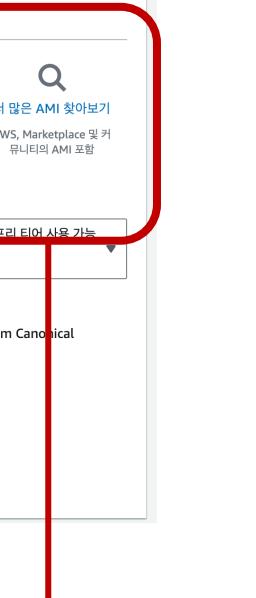
Amazon Machine Image(AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type  
 ami-05d2438ca66594916 (64비트(x86)) / ami-0b48860f51bc4313e (64비트(Arm))  
 가상화: hvm ENA 활성화됨: true 루트 디바이스 유형: ebs

설명  
Ubuntu Server 24.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

아키텍처 AMI ID 사용자 이름 확인된 공급업체

64비트(x86) ami-05d2438ca66594916 4916 ubuntu



▼ 애플리케이션 및 OS 이미지(Amazon Machine Image) 정보

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type  
 ami-05d2438ca66594916 (64비트(x86)) / ami-0b48860f51bc4313e (64비트(Arm))  
 가상화: hvm ENA 활성화됨: true 루트 디바이스 유형: ebs

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type

ami-056a29f2eddc40520 (64비트(x86)) / ami-03af4c483d5709cc8 (64비트(Arm))  
 가상화: hvm ENA 활성화됨: true 루트 디바이스 유형: ebs

Ubuntu Server 20.04 LTS (HVM) with SQL Server 2022 Standard

ami-08b2c3a9f2695e351 (64비트(x86))  
 가상화: hvm ENA 활성화됨: true 루트 디바이스 유형: ebs

Ubuntu Pro - Ubuntu Server Pro 24.04 LTS (HVM), SSD Volume Type

ami-02b38caf3c75cae8 (64비트(x86)) / ami-04076fa0dccbfba680 (64비트(Arm))  
 가상화: hvm ENA 활성화됨: true 루트 디바이스 유형: ebs

Deep Learning Base OSS Nvidia Driver GPU AMI (Ubuntu 22.04)

ami-08cbd56a1a23a1c6 (64비트(x86)) / ami-08f37a4e51f600100 (64비트(Arm))  
 가상화: hvm ENA 활성화됨: true 루트 디바이스 유형: ebs

Deep Learning OSS Nvidia Driver AMI GPU PyTorch 2.3 (Ubuntu 20.04)

ami-0edd08d4aa9666b9f (64비트(x86))

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type

ami-05d2438ca66594916 (64비트(x86)) / ami-0b48860f51bc4313e (64비트(Arm))  
 가상화: hvm ENA 활성화됨: true 루트 디바이스 유형: ebs

AWSKRUG BEGINNER GROUP

▼ 애플리케이션 및 OS 이미지(Amazon Machine Image) 정보

프리 티어 사용 가능 ✓

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type  
 ami-05d2438ca66594916 (64비트(x86)) / ami-0b48860f51bc4313e (64비트(Arm))  
 가상화: hvm ENA 활성화됨: true 루트 디바이스 유형: ebs

프리 티어 사용 가능

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type  
 ami-056a29f2eddc40520 (64비트(x86)) / ami-03af4c483d5709cc8 (64비트(Arm))  
 가상화: hvm ENA 활성화됨: true 루트 디바이스 유형: ebs

Ubuntu Server 20.04 LTS (HVM) with SQL Server 2022 Standard  
 ami-08b2c3a9f2695e351 (64비트(x86))  
 가상화: hvm ENA 활성화됨: true 루트 디바이스 유형: ebs

Ubuntu Pro - Ubuntu Server Pro 24.04 LTS (HVM), SSD Volume Type  
 ami-02b38caf3c75cae8 (64비트(x86)) / ami-04076fa0dccbfba680 (64비트(Arm))  
 가상화: hvm ENA 활성화됨: true 루트 디바이스 유형: ebs

Deep Learning Base OSS Nvidia Driver GPU AMI (Ubuntu 22.04)  
 ami-08cbd56a1a23a1c6 (64비트(x86)) / ami-08f37a4e51f600100 (64비트(Arm))  
 가상화: hvm ENA 활성화됨: true 루트 디바이스 유형: ebs

Deep Learning OSS Nvidia Driver AMI GPU PyTorch 2.3 (Ubuntu 20.04)  
 ami-0edd08d4aa9666b9f (64비트(x86))

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type  
 ami-05d2438ca66594916 (64비트(x86)) / ami-0b48860f51bc4313e (64비트(Arm))  
 가상화: hvm ENA 활성화됨: true 루트 디바이스 유형: ebs

Amazon Machine Image(AMI) 선택

AMI는 인스턴스를 시작하는 데 필요한 소프트웨어 구성(운영 체제, 애플리케이션 서버 및 애플리케이션)이 포함된 템플릿입니다. AWS, 사용자 커뮤니티 또는 AWS Marketplace에서 제공하는 AMI를 선택하거나 자체 AMI 중 하나를 선택할 수 있습니다.

선택한 AMI: ami-0847e1e59553197ba (미리 선택)

선택

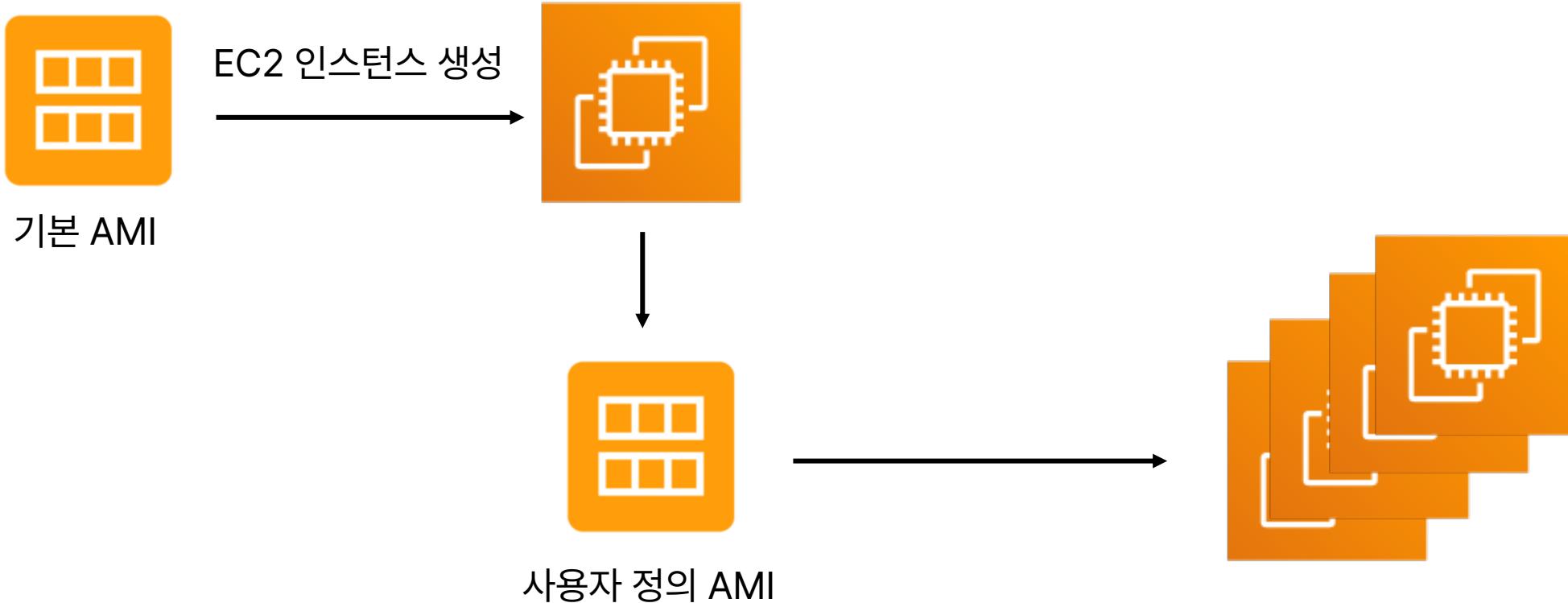
Amazon Linux 2023 AMI  
 ami-06f73fc34ddfd65c2 (64비트(x86), uefi-preferred) / ami-091ax52c31df1:84 (64비트(Arm), uefi)  
 Amazon Linux 2023은 5년간 장기 지원을 제공하는 최신 베이스 Linux 기반 OS입니다. AWS에 최적화되어 있으며 클라우드에서 호스팅되는 커뮤니티 버전입니다.

선택

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type  
 ami-06087749a704b8168 (64비트(x86)) / ami-01f22bbab1f75e10 (64비트(Arm))  
 Amazon Linux 2는 5년간 지원을 제공합니다. Amazon EC2에 성능 최적화된 Linux kernel 5.10과 systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, 최신 소프트웨어 패키지를 추가적으로 제공합니다.

선택

# 사용자 정의 AMI



# EBS(Elastic Block Store)

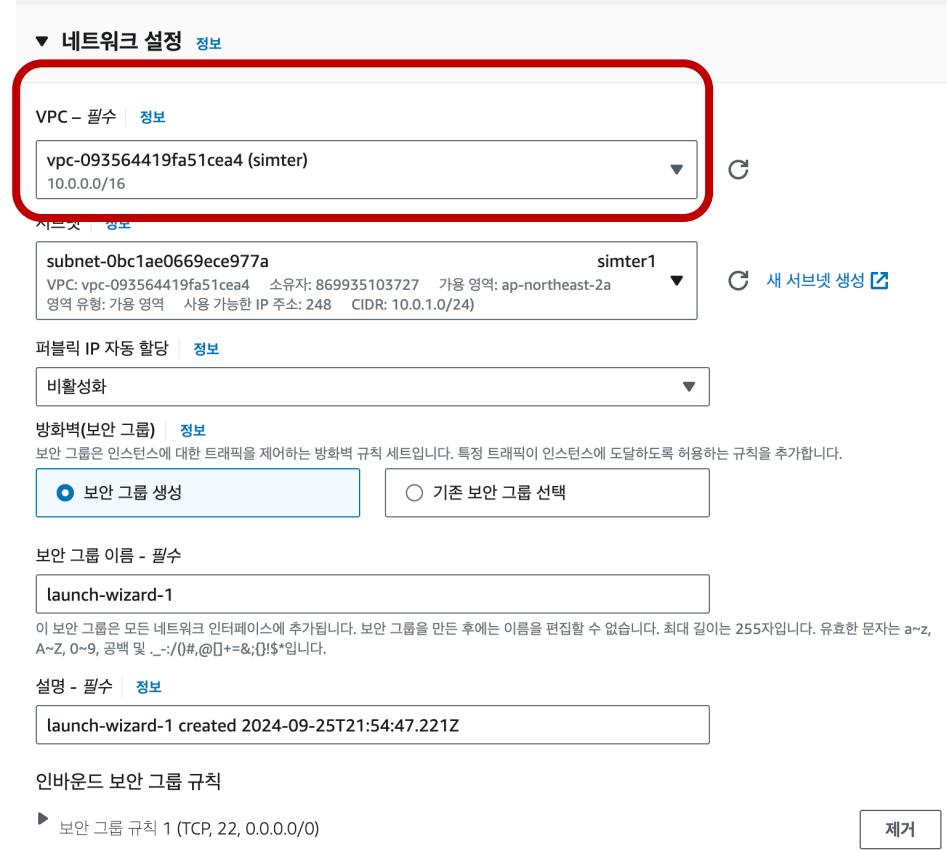
: EC2 인스턴스 블록 스토리지



인스턴스가 종료되어도 데이터는 그대로 유지됨!

EBS 연결을 해제한 다음 다른 인스턴스에 연결할 수도 있다.

# EC2 네트워크 설정



# VPC(Virtual Private Cloud)

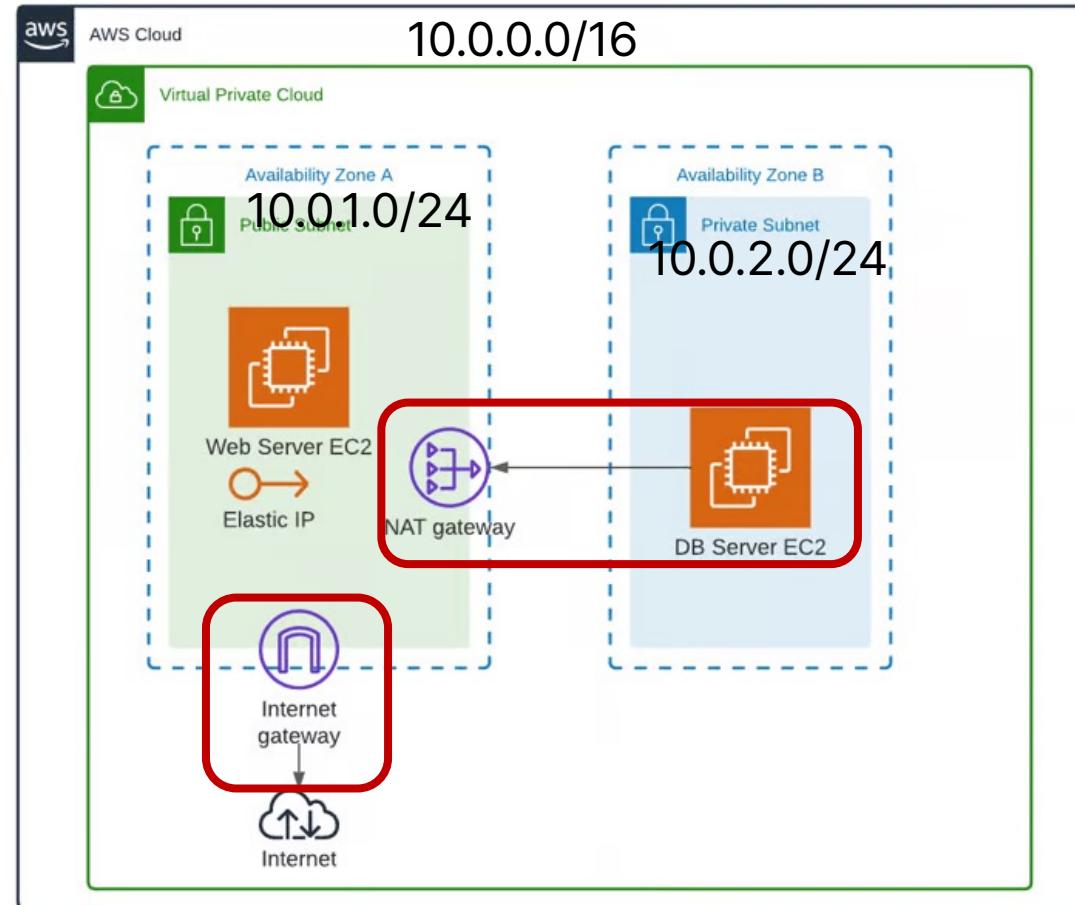
: 가상의 Private 클라우드 네트워크



- 리전마다 독립적으로 생성
- 독립된 클라우드 네트워크
- 하나의 VPC를 여러 개의 서브넷으로 나눌 수 있음

# Subnet

: VPC를 더 작게 나눈 네트워크

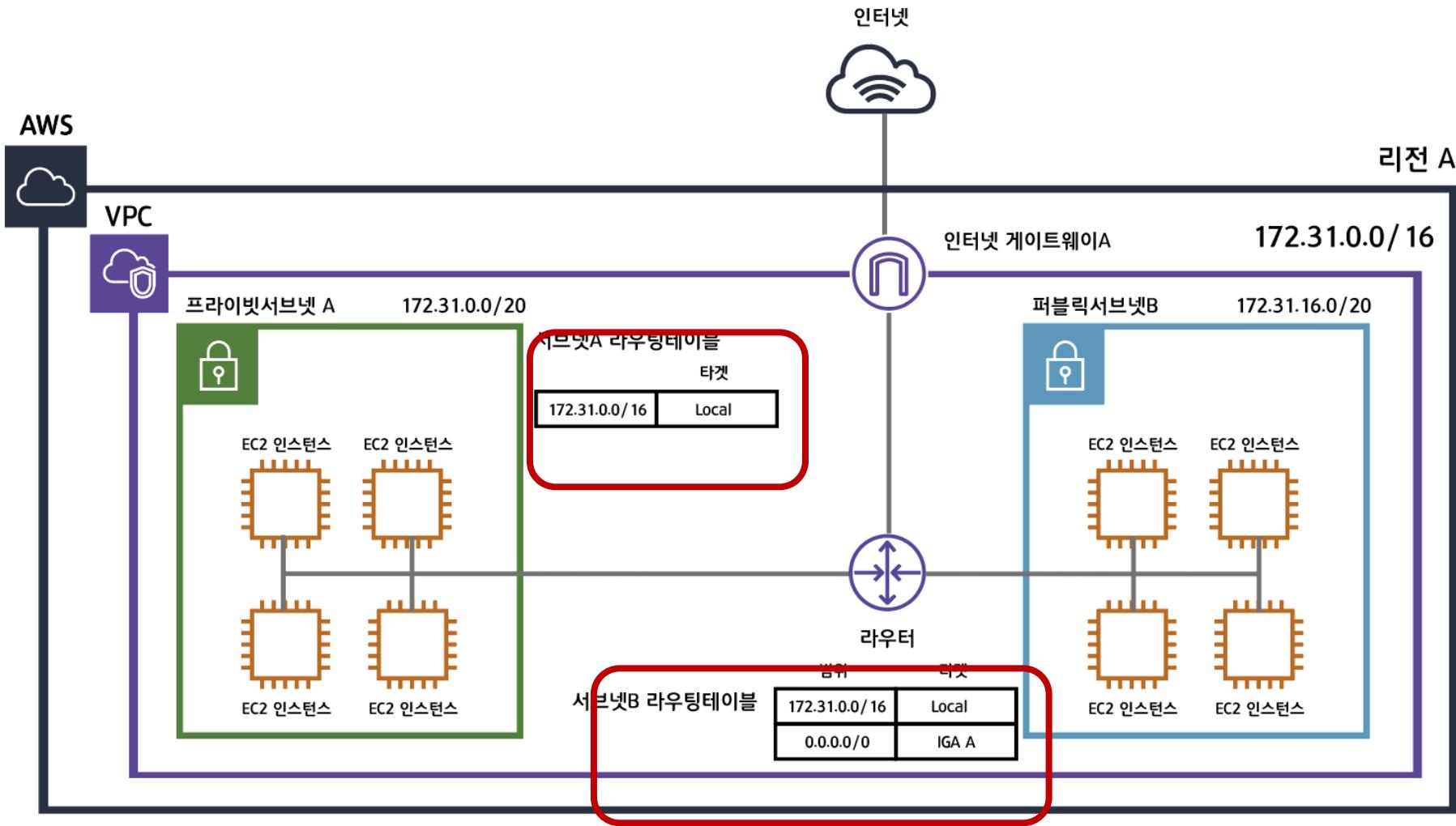


# Routing Table

: 네트워크에서 데이터가 어디로 전송되어야 할지 결정하는 표

- VPC와 서브넷 간의 트래픽 흐름을 제어하기 위해 사용
- 하나의 서브넷은 하나의 라우팅 테이블에 연결됨
- **VPC 내부 통신을** 위해서 기본적으로 local 라우팅이 설정되어 있음

# Routing Table





그럼 대체 Private Subnet은 왜 쓰는 거지?

# Private Subnet

: 외부 인터넷과 직접적으로 연결되지 않은 네트워크

→ 외부에서 접근하지 못 하므로 보안 ↑

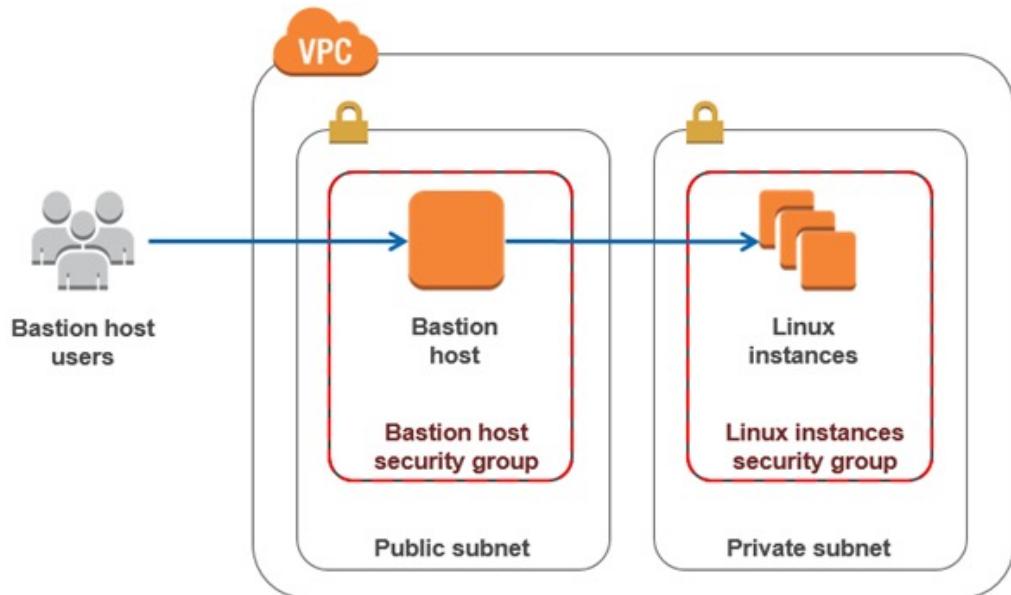
```
+-----+  
| Tables_in_README_TO_RECOVER_TNA |  
+-----+  
| README |  
+-----+  
1 row in set (0.01 sec)  
  
mysql> SELECT * FROM README;  
+-----+  
+-----+  
| id | Message  
+-----+  
| Bitcoin_Address | Email  
+-----+  
| 1 | To recover your lost databases pay 0.006 BTC (Bitcoin) to this address: 1NbDyxBH2vMQteEbc2Rcq6LLXZTzw71nR8. After your payment contact us at recovery10956@proton.me with your server IP (52.79.128.208) and transaction ID. We will reply with your backup within few minutes. | recovery10956@proton.me | 1NbDyxBH2vMQteEbc2Rcq6LLXZTzw71nR8 |  
+-----+  
+-----+  
1 row in set (0.01 sec)  
  
mysql>
```

Public에 뒀다가 탈취당하고 비트코인 요구받음ㅠㅠ

# Private Subnet

: 외부 인터넷과 직접적으로 연결되지 않은 네트워크

## 접근 방법 1: Bastion Host

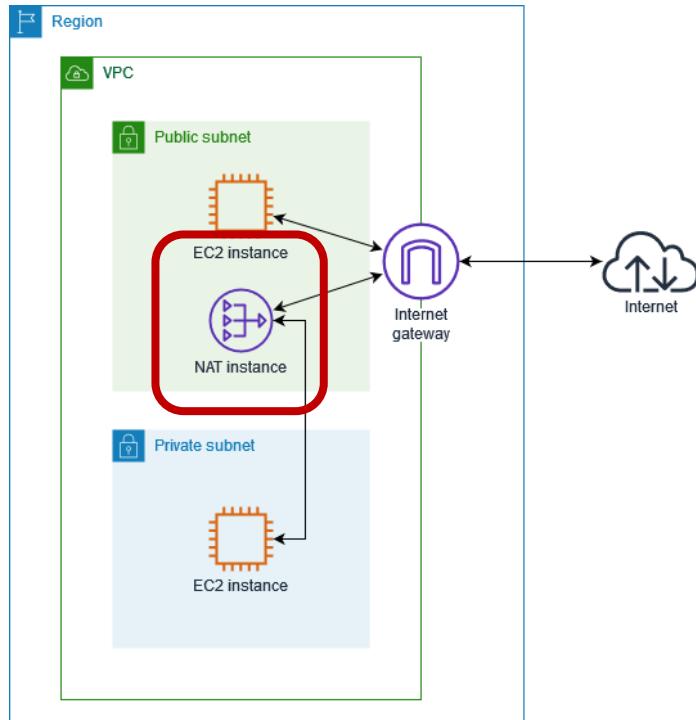


퍼블릭 서브넷에 위치한 서버를 접속한 후  
해당 서버를 거쳐 Private Subnet 내의 리소스에 접속

# Private Subnet

: 외부 인터넷과 직접적으로 연결되지 않은 네트워크

## 접근 방법 2 : NAT Gateway



1. Public Subnet에 **NAT Gateway** 생성
2. Private Subnet의 라우팅 테이블에 **NAT Gateway**를 타고 외부로 나가는 경로 ( $0.0.0.0/0$ ) 추가

# Firewall(방화벽)

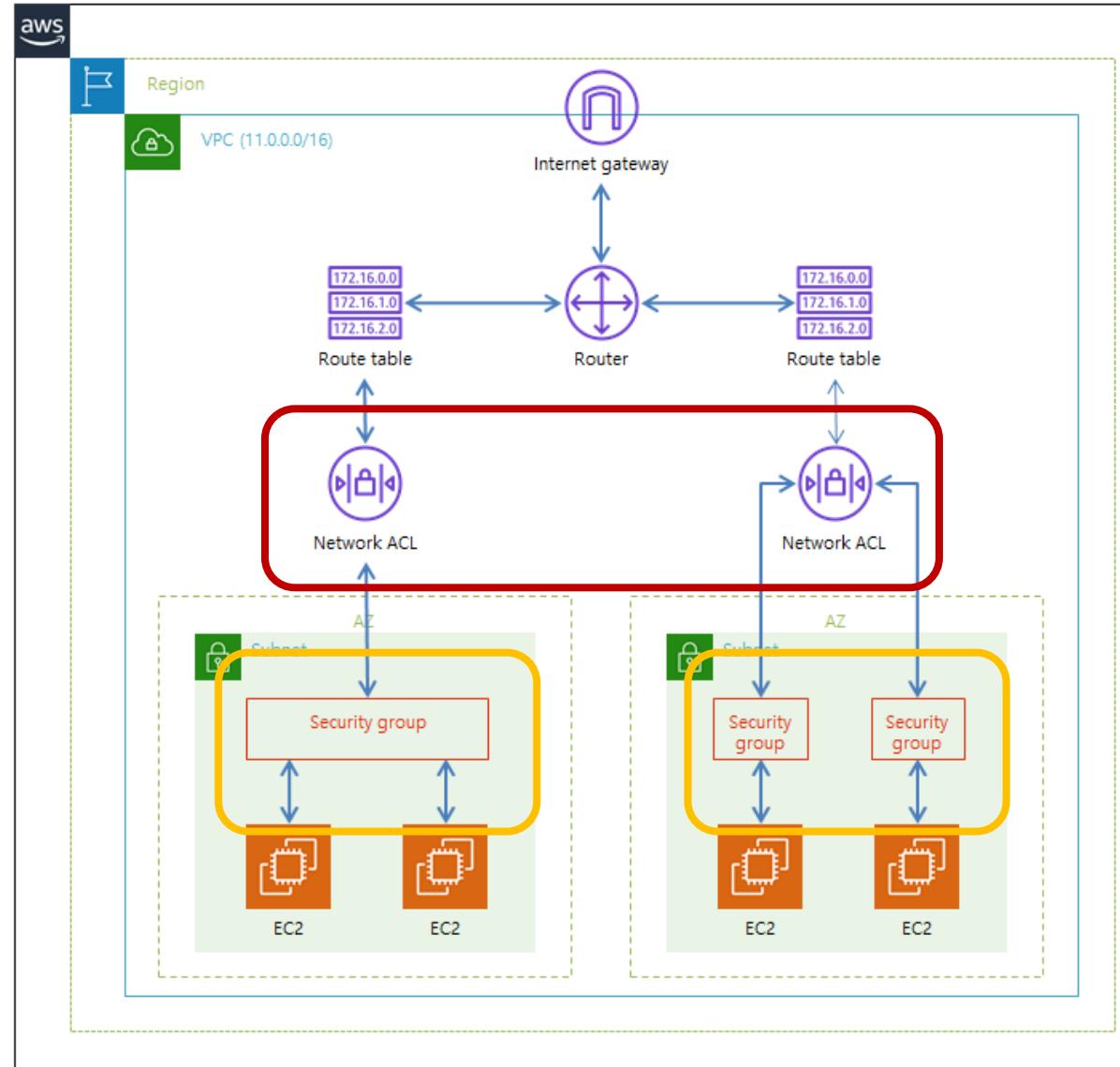
## Security Group(보안그룹)

- 인스턴스 제어
- Stateful 접근 통제(이전 상태 정보 기억)
- 인바운드 규칙에서 트래픽을 허용했으면  
아웃바운드 규칙에 상관없이 자동으로 접근 허용
- 허용 규칙만 정함

## Network ACL

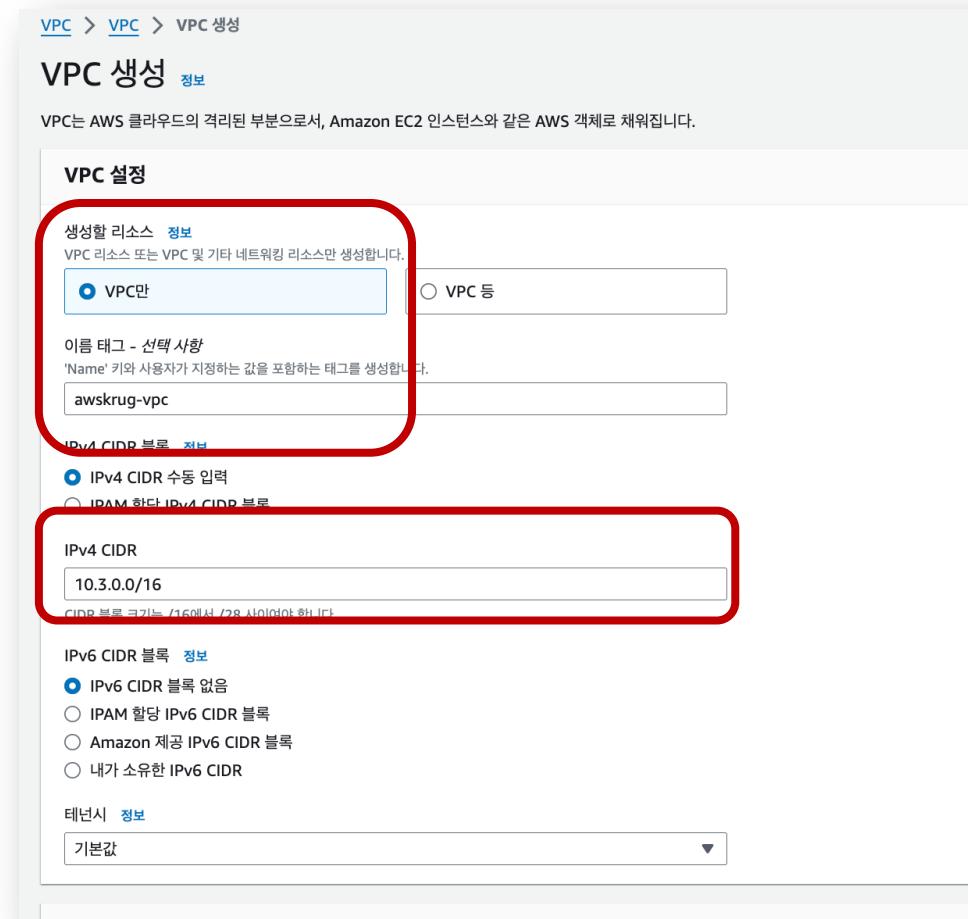
- 서브넷 제어
- Stateless(이전 상태 정보 기억 x)
- 인바운드 트래픽 허용했어도 아웃바운드 규칙  
으로 허용 여부 판단
- 허용, 거부 규칙 정함

# Firewall



# VPC, 서브넷 구성

## 1. VPC 생성



# VPC, 서브넷 구성

## 1. VPC 생성

The screenshot shows the AWS VPC console with the following details:

| 세부 정보                           |                                      | 정보                                  |                                      |
|---------------------------------|--------------------------------------|-------------------------------------|--------------------------------------|
| VPC ID<br>vpc-0bb2da70c7d981a4f | 상태<br>Available                      | DNS 호스트 이름<br>비활성화됨                 | DNS 확인<br>활성화됨                       |
| 테넌시<br>Default                  | DHCP 옵션 세트<br>dopt-005aba26b00fdad09 | 기본 라우팅 테이블<br>rtb-09d162f1db37d3eaf | 기본 네트워크 ACL<br>acl-093cb5b83c56a6627 |
| 기본 VPC<br>아니요                   | IPv4 CIDR<br>10.3.0.0/16             | IPv6 풀                              | IPv6 CIDR(네트워크 경계 그룹)                |
| 네트워크 주소 사용 지표<br>비활성화됨          | Route 53 Resolver DNS 방화벽 규칙 그룹<br>- | 소유자 ID<br>869935103727              | -                                    |

라우팅 테이블, 네트워크 ACL 자동 생성됨

# VPC, 서브넷 구성

## 1. VPC 생성

The screenshot shows the AWS Network ACL configuration interface. At the top, there is a search bar and a button labeled "네트워크 ACL 생성". Below it, a table lists one Network ACL:

| Name                  | 네트워크 ACL ID | 연결 대상 | 기본값                                 | VPC ID    | 인바운드 규칙 수  | 아웃바운드 규칙 수 | 소유 |
|-----------------------|-------------|-------|-------------------------------------|-----------|------------|------------|----|
| acl-093cb5b83c56a6627 | -           | 예     | vpc-0bb2da70c7d981a4f / awskrug-vpc | 2 인바운드 규칙 | 2 아웃바운드 규칙 | 869        |    |

Below this, there are three tabs: "세부 정보", "인바운드 규칙", and "아웃바운드 규칙". The "인바운드 규칙" tab is selected, showing two inbound rules:

| 규칙 번호 | 유형     | 프로토콜 | 포트 범위 | 소스        | 허용/거부 |
|-------|--------|------|-------|-----------|-------|
| 100   | 모든 트래픽 | 모두   | 모두    | 0.0.0.0/0 | Allow |
| *     | 모든 트래픽 | 모두   | 모두    | 0.0.0.0/0 | Deny  |

The "아웃바운드 규칙" tab is also visible, showing two outbound rules:

| 규칙 번호 | 유형     | 프로토콜 | 포트 범위 | 대상        | 허용/거부 |
|-------|--------|------|-------|-----------|-------|
| 100   | 모든 트래픽 | 모두   | 모두    | 0.0.0.0/0 | Allow |
| *     | 모든 트래픽 | 모두   | 모두    | 0.0.0.0/0 | Deny  |

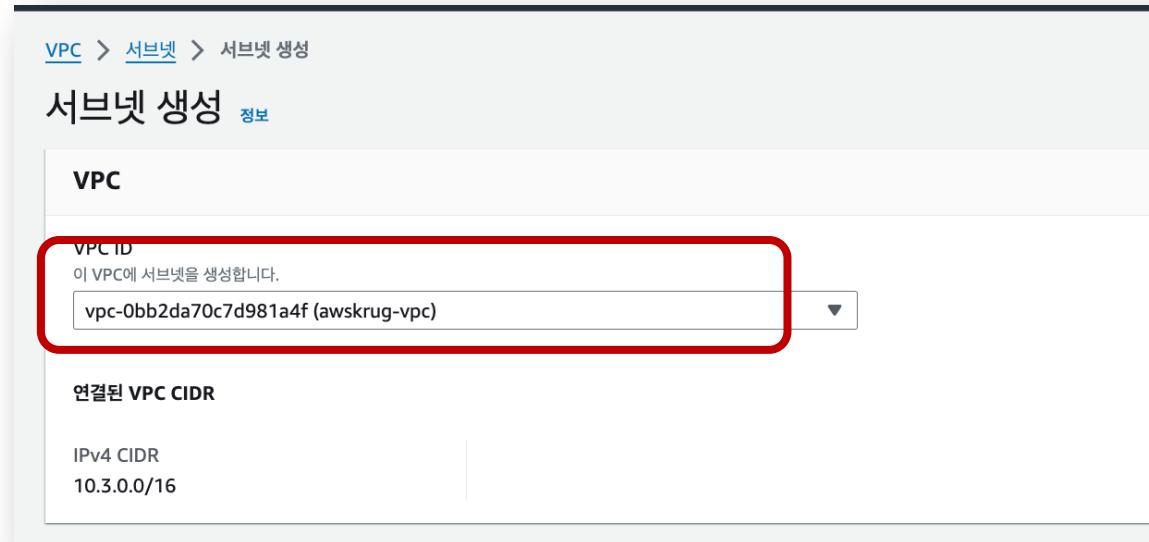
A red box highlights the second outbound rule (rule number \*).

네트워크 ACL : 기본적으로 모든 트래픽과 IP 대역 허용

# VPC, 서브넷 구성

## 2. Subnet 생성

생성한 VPC 선택



# VPC, 서브넷 구성

## 2. Subnet 생성

**서브넷 설정**  
서브넷의 CIDR 블록 및 가용 영역을 지정합니다.

**1/1개 서브넷**

서브넷 이름  
'Name' 키와 사용자가 지정하는 값을 포함하는 태그를 생성합니다.

이름은 최대 256자까지 입력할 수 있습니다.

가용 영역 **정보**  
서브넷이 상주할 영역을 선택합니다. 선택하지 않으면 Amazon이 자동으로 선택합니다.

IPv4 VPC CIDR 블록 **정보**  
서브넷에 대해 VPC의 IPv4 CIDR 블록을 선택합니다. 서브넷의 IPv4 CIDR이 이 블록 내에 있어야 합니다.

IPv4 서브넷 CIDR 블록  
 56 IPs

**1/1개 서브넷**

서브넷 이름  
'Name' 키와 사용자가 지정하는 값을 포함하는 태그를 생성합니다.

이름은 최대 256자까지 입력할 수 있습니다.

가용 영역 **정보**  
서브넷이 상주할 영역을 선택합니다. 선택하지 않으면 Amazon이 자동으로 선택합니다.

IPv4 VPC CIDR 블록 **정보**  
서브넷에 대해 VPC의 IPv4 CIDR 블록을 선택합니다. 서브넷의 IPv4 CIDR이 이 블록 내에 있어야 합니다.

IPv4 서브넷 CIDR 블록  
 256 IPs

# VPC, 서브넷 구성

## 3. 라우팅 테이블 생성

VPC > 라우팅 테이블 > 라우팅 테이블 생성

### 라우팅 테이블 생성 정보

라우팅 테이블은 VPC, 인터넷 및 VPN 연결 내 서브넷 간에 패킷이 전달되는 방법을 지정합니다.

#### 라우팅 테이블 설정

이름 - 선택 사항  
'Name' 키와 사용자가 지정하는 값을 포함하는 태그를 생성합니다.

awskrug-routing-table

VPC  
이 라우팅 테이블에 대해 사용할 VPC입니다.

vpc-0bb2da70c7d981a4f (awskrug-vpc)

태그

태그는 AWS 리소스에 할당하는 레이블입니다. 각 태그는 키와 선택적 값으로 구성됩니다. 태그를 사용하여 리소스를 검색 및 필터링하거나 AWS 비용을 추적할 수 있습니다.

| 키    | 값 - 선택 사항             |
|------|-----------------------|
| Name | awskrug-routing-table |

새 태그 추가

49을(를) 태그가 더 추가할 수 있습니다.

취소 라우팅 테이블 생성

# VPC, 서브넷 구성

## 3. 라우팅 테이블 생성

라우팅 테이블: [rtb-0015de8387b76cbe7 / awskrug-routing-table](#)

라우팅 (2)

검색: 라우팅 필터링

| 대상          | действие |
|-------------|----------|
| 10.3.0.0/16 | local    |

# 실습 – VPC, 서브넷 구성

## 4. 라우팅 테이블 생성

The screenshot shows two overlapping AWS management console windows. The left window is titled 'rtb-0015de8387b76cbe7 / awskrug-routing-table' and displays basic information about the routing table, including its ID, VPC association, and a list of associated subnets. The right window is titled '서브넷 연결 편집' (Edit Subnet Association) and lists available and selected subnets for association.

**Left Window (Routing Table Details):**

- 라우팅 테이블 ID: rtb-0015de8387b76cbe7
- VPC: vpc-0bb2da70c7d981a4f | awskrug-vpc
- 기본: 아니요
- 소유자 ID: 869935103727

**Right Window (Subnet Association Editor):**

VPC > 라우팅 테이블 > rtb-0015de8387b76cbe7 > 서브넷 연결 편집

### 서브넷 연결 편집

이 라우팅 테이블과 연결된 서브넷을 변경합니다.

#### 이용 가능한 서브넷 (2/2)

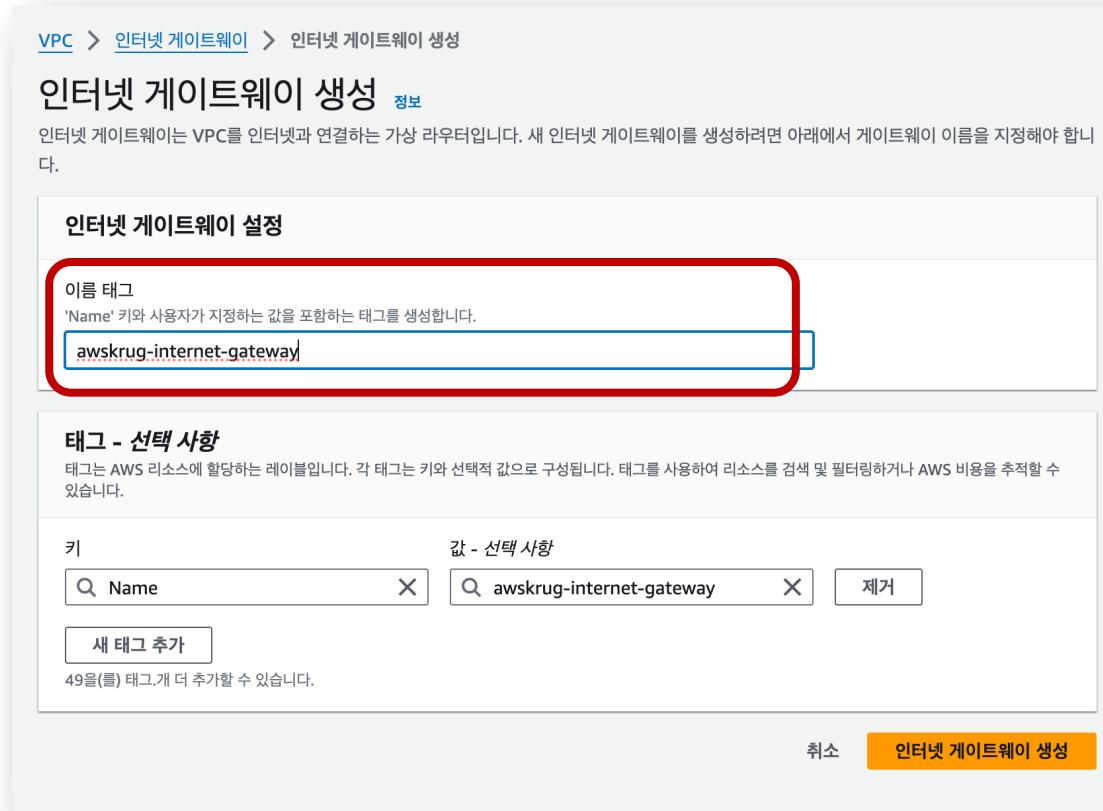
| 선택                                  | 이름                     | 서브넷 ID                   | IPv4 CIDR   |
|-------------------------------------|------------------------|--------------------------|-------------|
| <input checked="" type="checkbox"/> | awskrug-public-subnet  | subnet-0f94be6cddcc6cfbc | 10.3.1.0/24 |
| <input checked="" type="checkbox"/> | awskrug-private-subnet | subnet-02a7227ec48ee0d9e | 10.3.2.0/24 |

**선택한 서브넷:**

- subnet-0f94be6cddcc6cfbc / awskrug-public-subnet
- subnet-02a7227ec48ee0d9e / awskrug-private-subnet

# VPC, 서브넷 구성

## 4. 인터넷 게이트웨이 생성



# VPC, 서브넷 구성

## 4. 인터넷 게이트웨이 – VPC 연결

The screenshot shows two overlapping AWS CloudFormation stack pages.

**Left Stack:** igw-0f853771a324df4c9 / awskrug-internet-gateway

- General Information:** Status: Detached, VPC ID: -
- Tags:** Name: awskrug-internet-gateway

**Right Stack:** VPC에 연결(igw-0f853771a324df4c9)

- VPC Selection:** A search bar contains the VPC ID: vpc-0bb2da70c7d981a4f.
- Actions:** A button labeled "인터넷 게이트웨이 연결" (Internet Gateway Connection) is visible.

# VPC, 서브넷 구성

## 4. 라우팅 테이블 라우팅 편집 → 외부와 통신 가능!

The screenshot shows the AWS VPC Routing Table configuration interface. On the left, the VPC Dashboard sidebar lists various VPC-related services. The main pane displays the 'Routing Tables' section with a table of routing tables. One row is selected, showing the 'awskrug-routing-table' with ID 'rtb-0015de8387b76cbe7'. A detailed view for this table is shown below, with the 'Routing' tab selected. In the bottom right corner, a modal window titled 'Routing Rule' is open, showing a list of destination ranges. Two entries are highlighted with a red box: '0.0.0.0/0' (targeting 'Internet Gateway') and 'igw-0f853771a324df4c9' (targeting 'local').

| Name  | Routing Table ID      | Associated Subnet        | Target | VPC                 |
|---|-----------------------|--------------------------|--------|---------------------|
| -   | rtb-0234df639d93733b0 | 2 서브넷                    | -      | vpc-093564419fa51ce |
| -   | rtb-09d162f1db37d3eaf | -                        | -      | vpc-0bb2da70c7d981a |
| <input checked="" type="checkbox"/> awsjson-routing-table | rtb-0015de8387b76cbe7 | subnet-0f94be6cdcc6cf... | -      | vpc-0bb2da70c7d981a |

**awsjson-routing-table / awsjson-routing-table**

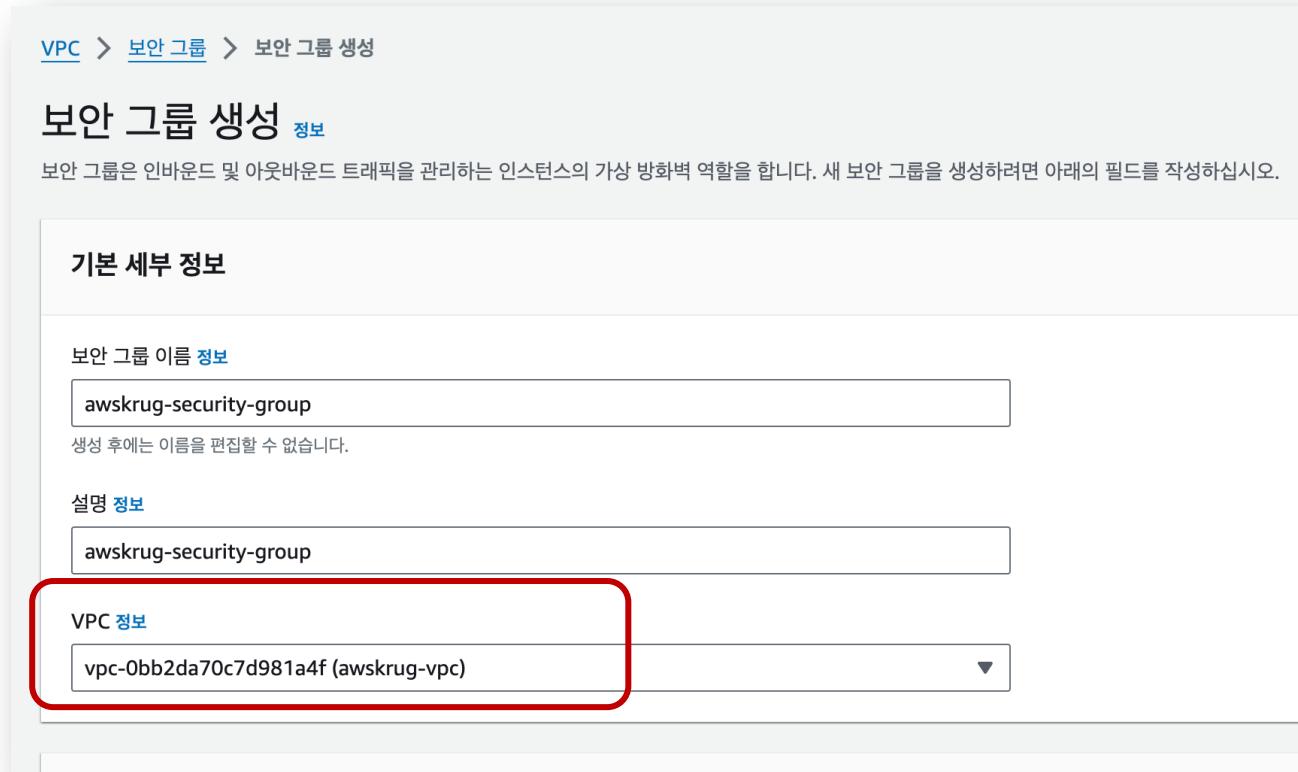
대상 대상  
10.3.0.0/16 local

0.0.0.0/0 (highlighted)  
인터넷 게이트웨이  
igw-0f853771a324df4c9 (highlighted)

# Public Subnet에 EC2 생성

## 1. 보안그룹 설정

따로 지정하지 않으면 신규 보안 그룹이 자동으로 생성됨



# Public Subnet에 EC2 생성

## 1. 보안그룹 설정

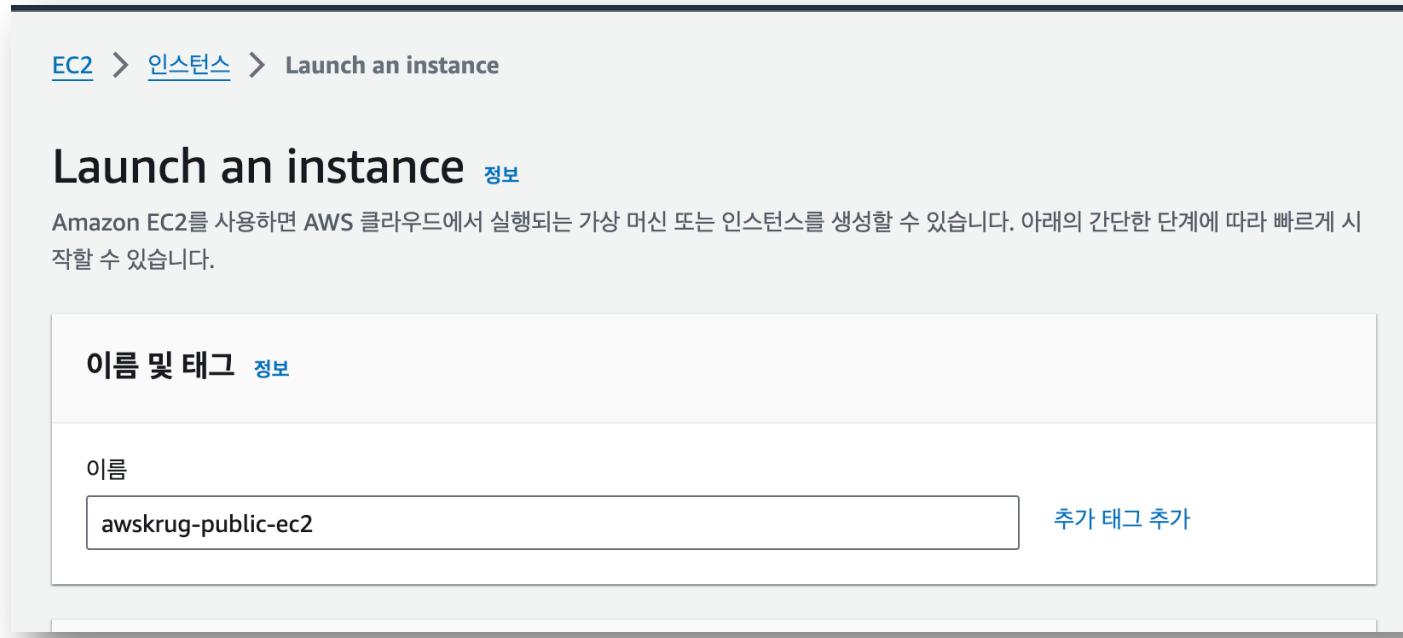
인바운드 규칙 정보

| 유형 정보 | 프로토콜 정보 | 포트 범위 정보 | 소스 정보                      |
|-------|---------|----------|----------------------------|
| HTTP  | TCP     | 80       | 내 IP<br>221.153.106.111/32 |
| SSH   | TCP     | 22       | 내 IP<br>221.153.106.111/32 |

규칙 추가

# Public Subnet에 EC2 생성

## 2. EC2 생성



# Public Subnet에 EC2 생성

## 2. EC2 생성

▼ 애플리케이션 및 OS 이미지(Amazon Machine Image) 정보

AMI는 인스턴스를 시작하는 데 필요한 소프트웨어 구성(운영 체제, 애플리케이션 서버 및 애플리케이션)이 포함된 템플릿입니다. 아래에서 찾고 있는 항목이 보이지 않으면 AMI를 검색하거나 찾아보세요.

수천 개의 애플리케이션 및 OS 이미지를 포함하는 전체 카탈로그 검색

최근 사용Quick Start



Amazon  
Linux



macOS



Ubuntu



Windows



Red Hat



SUSE

더 많은 AMI 찾아보기

AWS, Marketplace 및 커뮤니티의 AMI 포함

Amazon Machine Image(AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type

ami-05d2438ca66594916 (64비트(x86)) / ami-0b48860f51bc4313e (64비트(Arm))

기상화: hvm ENA 활성화됨: true 루트 디바이스 유형: ebs

프리 티어 사용 가능 ▾

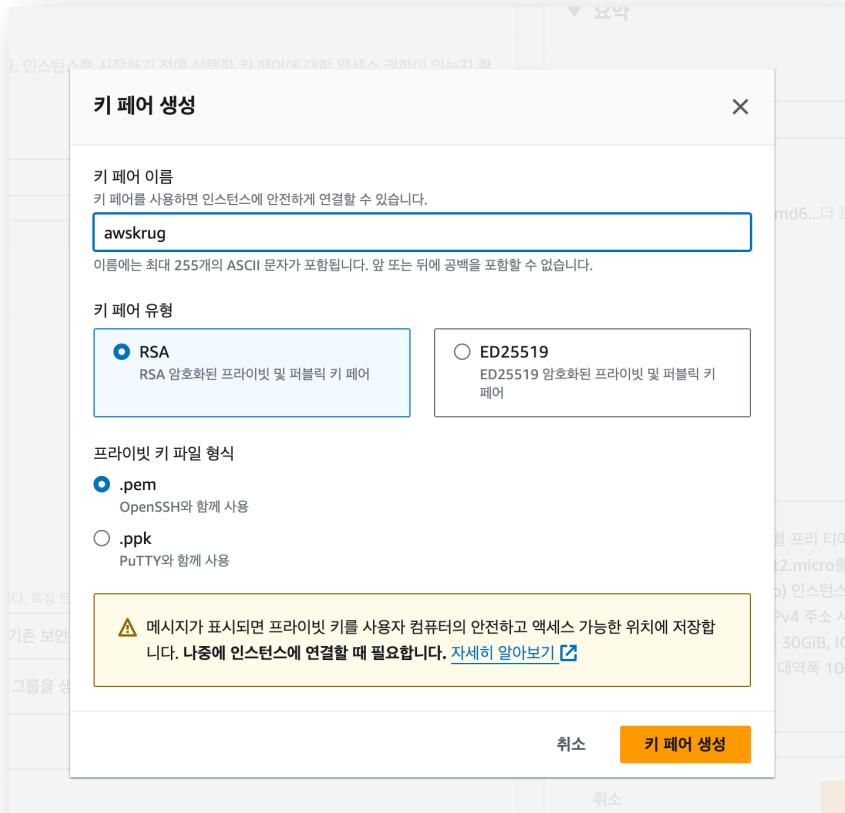
설명

Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

|           |                       |        |          |
|-----------|-----------------------|--------|----------|
| 아키텍처      | AMI ID                | 사용자 이름 | (i)      |
| 64비트(x86) | ami-05d2438ca66594916 | ubuntu | 확인된 공급업체 |
|           | 4916                  |        |          |

# Public Subnet에 EC2 생성

## 2. EC2 생성 – 키 페어 생성

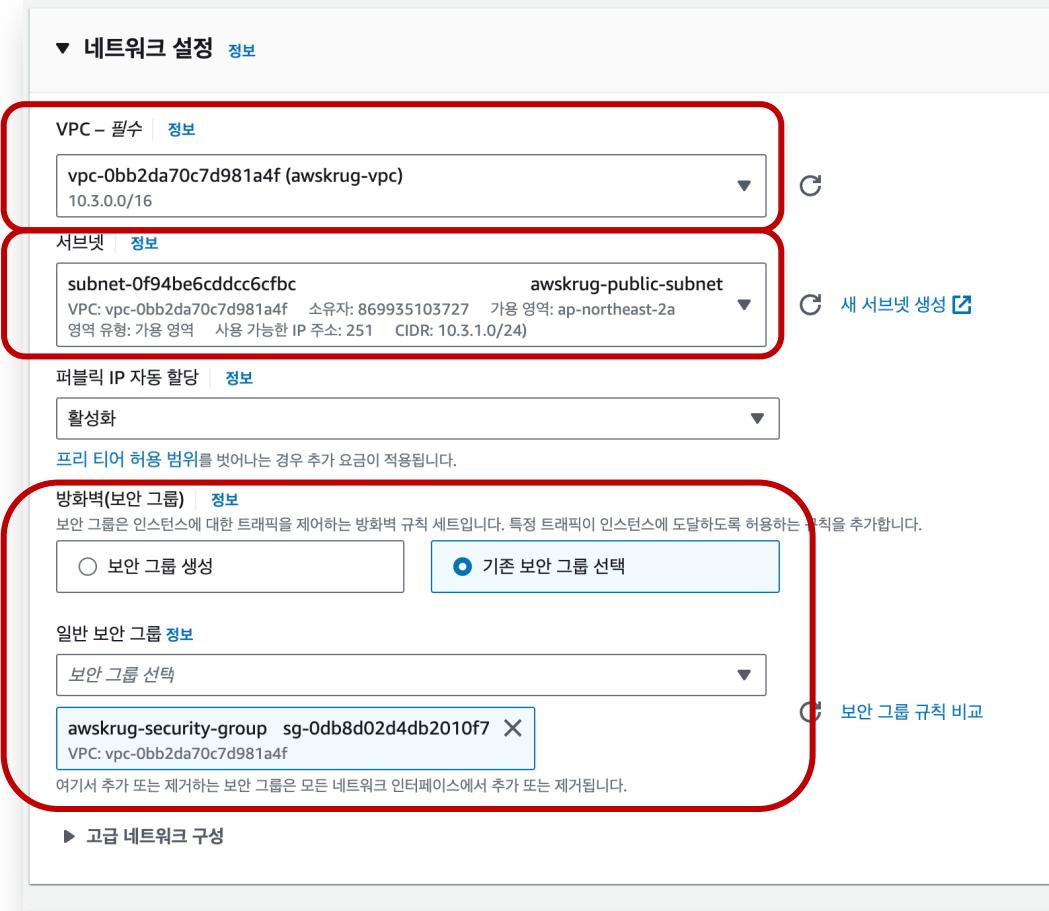


## SSH로 EC2 인스턴스에 접속할 때 사용!

인스턴스 생성 시 한 번만 다운로드 가능하고  
분실 시 복구할 수 없으므로 안전하게 보관해야한다.

# Public Subnet에 EC2 생성

## 2. EC2 생성



# Public Subnet에 EC2 생성

## 2. EC2 생성

The screenshot shows the AWS EC2 instance configuration screen for an instance named 'i-04e7530e503d950f6 (awskrug-public-ec2)'. The 'Networking' tab is selected. The interface is in Korean.

**네트워킹 세부 정보**

| 설정                      | 값   |
|-------------------------|---|
| 퍼블릭 IPv4 주소             | 13.125.209.92   <a href="#">개방 주소법</a>                          |
| 퍼블릭 IPv4 DNS            | -   |
| 서브넷 ID                  | <a href="#">subnet-0f94be6cdcc6cfbc (awskrug-public-subnet)</a> |
| 가용 영역                   | <a href="#">ap-northeast-2a</a>                                 |
| RBN을 게스트 OS 호스트 이름으로 사용 | 비활성   |

**프라이빗 IPv4 주소**

| 설정                       | 값   |
|--------------------------|---|
| 프라이빗 IP DNS 이름(IPv4만 해당) | ip-10-3-1-232.ap-northeast-2.compute.internal |

**VPC ID**

| 설정     | 값   |
|--------|---|
| VPC ID | <a href="#">vpc-0bb2da70c7d981a4f (awskrug-vpc)</a> |

**보조 프라이빗 IPv4 주소**

| 설정              | 값 |
|-----------------|---|
| 보조 프라이빗 IPv4 주소 | - |

**Outpost ID**

| 설정         | 값 |
|------------|---|
| Outpost ID | - |

**네트워크 인터페이스 (1) 정보**

# Public Subnet에 EC2 생성

## 탄력적 IP

EC2 인스턴스는 기본적으로 퍼블릭 IP 주소를 할당받을 수 있지만,  
인스턴스가 중지되거나 재부팅되면 IP 주소가 변경됨!

탄력적 IP를 할당하면 인스턴스가 중지되거나 재부팅되어도 IP주소가 바뀌지 않는다.

한 인스턴스에서 다른 인스턴스로 재할당이 가능함!

# Public Subnet에 EC2 생성

## 탄력적 IP

EC2 > 탄력적 IP 주소 > 탄력적 IP 주소 할당 정보

### 탄력적 IP 주소 설정 정보

피블릭 IPv4 주소 풀

Amazon의 IPv4 주소 풀

BYOIP 방식으로 AWS 계정에 가져오는 피블릭 IPv4 주소입니다. (풀을 찾을 수 없으므로 옵션이 비활성화됨) [자세히 알아보기](#)

Outpost에서 사용하기 위해 온프레미스 네트워크에서 생성한 고객 소유의 IPv4 주소 풀입니다. (고객 소유 풀을 찾을 수 없기 때문에 옵션이 비활성화됨) [자세히 알아보기](#)

Allocate using an IPv4 IPAM pool (option disabled because no public IPv4 IPAM pools with AWS service as EC2 were found)

네트워크 경계 그룹 정보

ap-northeast-2

글로벌 정적 IP 주소

AWS Global Accelerator는 AWS 엣지 로케이션의 애니캐스트를 사용하여 전 세계에 발표된 글로벌 정적 IP 주소를 제공할 수 있습니다.  
이를 통해 Amazon 글로벌 네트워크를 사용하여 사용자 트래픽의 가용성과 지연 시간을 개선할 수 있습니다. [자세히 알아보기](#)

액셀러레이터 생성

### 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.

탄력적 IP 주소가 할당되었습니다.  
탄력적 IP 주소 43.202.14.56

이 탄력적 IP 주소 연결

### 탄력적 IP 주소 (1/1)

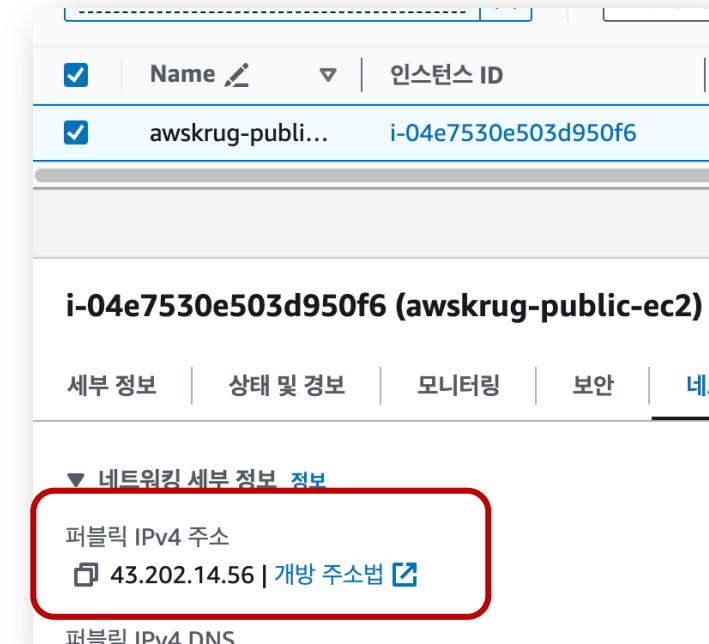
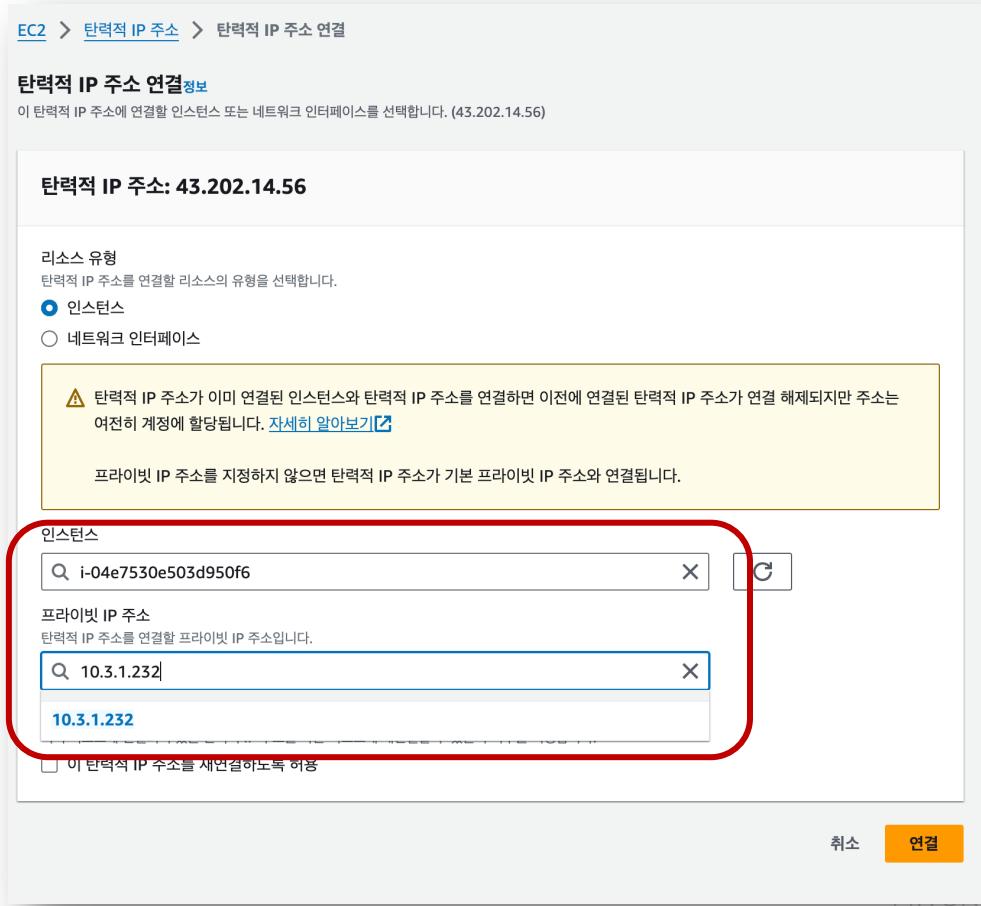
Find resources by attribute or tag

피블릭 IPv4 주소 : 43.202.14.56 X Clear filters

| Name | 할당된 IPv4 주소  | 유형     | 할당 ID                      | 역방향 DNS 레코드 |
|------|--------------|--------|----------------------------|-------------|
| -    | 43.202.14.56 | 퍼블릭 IP | eipalloc-01c3bd45e6a3eebaa | -           |

# Public Subnet에 EC2 생성

## 탄력적 IP

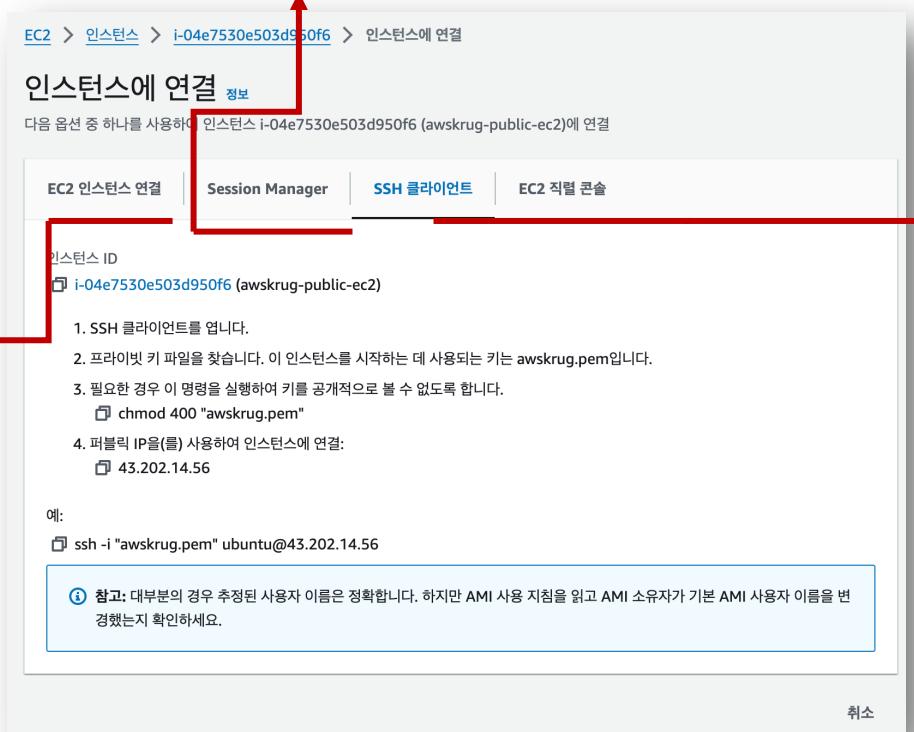


# Public Subnet에 EC2 생성

## EC2 접속 방법

22번 포트를 열지 않고 EC2 인스턴스에 연결할 수 있는 방법

브라우저에서 직접  
EC2 인스턴스에 SSH로 접속



리눅스/유닉스 기반의 EC2 인스턴스에  
원격으로 접속

# Public Subnet에 EC2 생성

## 3. SSH로 EC2 접속

The screenshot shows the AWS EC2 Instance Connect interface for connecting to an instance. The top navigation bar includes 'EC2' > '인스턴스' > 'i-04e7530e503d950f6' > '인스턴스에 연결'. The main title is '인스턴스에 연결' with a '정보' link. Below it, a note says '다음 옵션 중 하나를 사용하여 인스턴스 i-04e7530e503d950f6 (awskrug-public-ec2)에 연결'. The tabs at the top are 'EC2 인스턴스 연결', 'Session Manager', 'SSH 클라이언트' (which is selected), and 'EC2 직렬 콘솔'. The 'SSH 클라이언트' tab has sub-options: '인스턴스 ID', 'SSH 키 파일 찾기', '명령어 실행', and '터미널'. The '인스턴스 ID' section shows 'i-04e7530e503d950f6 (awskrug-public-ec2)'. The '명령어 실행' section contains the following steps:

1. SSH 클라이언트를 엽니다.
2. 프라이빗 키 파일을 찾습니다. 이 인스턴스를 시작하는 데 사용되는 키는 awskrug.pem입니다.
3. 필요한 경우 이 명령을 실행하여 키를 공개적으로 볼 수 없도록 합니다.  
     chmod 400 "awskrug.pem"
4. 퍼블릭 IP을(를) 사용하여 인스턴스에 연결:  
     43.202.14.56

Below these steps, there's an example command: '예:  
 ssh -i "awskrug.pem" ubuntu@43.202.14.56'. A note at the bottom left says: '참고: 대부분의 경우 추정된 사용자 이름은 정확합니다. 하지만 AMI 사용 지침을 읽고 AMI 소유자가 기본 AMI 사용자 이름을 변경했는지 확인하세요.' A '취소' button is at the bottom right.

chmod 400 "awskrug.pem"

소유자만 읽기 권한을 가지도록 설정

ssh -i "awskrug.pem" ubuntu@43.202.14.56

EC2 인스턴스 접속

# Public Subnet에 EC2 생성

## 3. SSH로 EC2 접속

```
sujin@sujinui-MacBookPro Downloads % ssh -i "awskrug.pem" ubuntu@43.202.14.56
The authenticity of host '43.202.14.56 (43.202.14.56)' can't be established.
ED25519 key fingerprint is SHA256:3zuiGXGU741dv3FS3Dh8HsemB/OcRXBpUU2vctvus2c.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:29: 13.125.209.92
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '43.202.14.56' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

System information as of Thu Sep 26 08:06:24 UTC 2024

  System load:  0.0          Processes:      105
  Usage of `/:  22.9% of 6.71GB  Users logged in:   1
  Memory usage: 20%
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or
proxy settings

Last login: Thu Sep 26 07:56:21 2024 from 121.137.48.239
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-3-1-232:~$
```

```
ubuntu@ip-10-3-1-232:~$ ls -al
total 32
drwxr-x--- 4 ubuntu ubuntu 4096 Sep 26 07:56 .
drwxr-xr-x 3 root   root   4096 Sep 26 01:50 ..
-rw----- 1 ubuntu ubuntu   37 Sep 26 07:56 .bash_history
-rw-r--r-- 1 ubuntu ubuntu  220 Mar 31 08:41 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu 3771 Mar 31 08:41 .bashrc
drwx----- 2 ubuntu ubuntu 4096 Sep 26 07:47 .cache
-rw-r--r-- 1 ubuntu ubuntu  807 Mar 31 08:41 .profile
drwx----- 2 ubuntu ubuntu 4096 Sep 26 01:50 .ssh
ubuntu@ip-10-3-1-232:~$ pwd
/home/ubuntu
ubuntu@ip-10-3-1-232:~$
```

# Public Subnet에 EC2 생성

## 4. Springboot 배포

방법은 정말정말 많다..!

1. scp 파일명 username@remote\_host:/원격/경로

2. 인스턴스 접속

```
-rw----- 1 ubuntu ubuntu    11178 Sep 25 17:03 nonup.out
-rw-rw-r-- 1 ubuntu ubuntu    42562 Sep 26 04:51 output.log
drwxrwxr-x 9 ubuntu ubuntu     4096 Sep 17 16:44 server
-rw-r--r-- 1 ubuntu ubuntu 82286609 Sep 26 05:05 simter-0.0.1-SNAPSHOT.jar
ubuntu@ip-10-0-1-94:~$ █
```

3. nohup java -jar <jar 파일 경로> log.txt 2>&1 &

# Private Subnet에 RDS 설정

## RDS

클라우드에서 관계형 데이터베이스를 만들고 관리할 수 있는 서비스

다중 AZ 배포로 가용성 ↑

Auto Scaling 기능으로 자동 확장이 가능!

The screenshot shows the AWS VPC Subnet Configuration interface. It consists of three vertically stacked panels, each titled '서브넷 설정' (Subnet Settings).

- Top Panel:** Shows the first step of creating a new subnet. It has a single input field for '서브넷 이름' (Subnet Name) containing 'awskrug-private-subnet2'. Below it is a note: '이름은 최대 256자까지 입력할 수 있습니다.' (Name can be up to 256 characters long). A 'Next Step' button is visible at the bottom right.
- Middle Panel:** Shows the second step of defining the IPv4 CIDR block. It has a dropdown menu set to '아시아 태평양 (서울) / ap-northeast-2a'. Below it is a note: '선택하지 않으면 Amazon이 자동으로 선택합니다.' (If not selected, Amazon will automatically select it). A 'Next Step' button is visible at the bottom right.
- Bottom Panel:** Shows the third step of defining the IPv4 subnet CIDR block. It has a dropdown menu set to '10.3.0.0/16'. Below it is a note: '선택하지 않으면 Amazon이 자동으로 선택합니다.' (If not selected, Amazon will automatically select it). A 'Next Step' button is visible at the bottom right.

# Private Subnet에 RDS 설정

## 1. 서브넷 그룹 생성

The screenshot shows the 'Amazon RDS' console with the 'Subnet Groups' section selected. The main page displays a list of existing subnet groups like 'awskrug-subnet-group'. On the right, a modal window titled 'DB 서브넷 그룹 생성' (Create DB Subnet Group) is open.

**Subnet Group Details:**

- 이름:** awsrukug-subnet-group
- 설명:** awsrukug-subnet-group
- VPC:** awsrukug-vpc (vpc-0bb2da70c7d981a4f)

**Add Subnet:** This section is highlighted with a red box. It shows three subnets selected under 'Available Regions':

- ap-northeast-2b
- ap-northeast-2a
- ap-northeast-2c

Below this, three specific subnets are listed:

- subnet-0010ff09890606615 (10.3.3.0/24)
- subnet-02a7227ec48ee0d9e (10.3.2.0/24)
- subnet-057864bab74e3e179 (10.3.4.0/24)

A note at the bottom of this section states: '① 다중 AZ DB 클러스터의 경우 3개의 서로 다른 가용 영역에서 3개의 서브넷을 선택해야 합니다.' (For a multi-AZ DB cluster, you must select 3 subnets from 3 different availability zones).

**Selected Subnets:**

| 가용 영역           | 서브넷 ID                   | CIDR 블록     |
|-----------------|--------------------------|-------------|
| ap-northeast-2a | subnet-0010ff09890606615 | 10.3.3.0/24 |
| ap-northeast-2c | subnet-02a7227ec48ee0d9e | 10.3.2.0/24 |
| ap-northeast-2b | subnet-057864bab74e3e179 | 10.3.4.0/24 |

# Private Subnet에 RDS 설정

## 2. RDS 생성

RDS > 데이터베이스 생성

### 데이터베이스 생성 정보

데이터베이스 생성 방식 선택

표준 생성  
가용성, 보안, 백업 및 유지 관리에 대한 옵션을 포함하여 모든 구성 옵션을 설정합니다.

순수운 생성  
원장 모범 사례 구성을 사용합니다. 일부 구성 옵션은 데이터베이스를 생성한 후 변경할 수 있습니다.

### 엔진 옵션

엔진 유형 정보

Aurora (MySQL Compatible)

Aurora (PostgreSQL Compatible)

MySQL

MariaDB

PostgreSQL

Oracle

### 설정

DB 인스턴스 식별자 정보  
DB 인스턴스 이름을 입력하세요. 이름은 현재 AWS 리전에서 AWS 계정이 소유하는 모든 DB 인스턴스에 대해 고유해야 합니다.

DB 인스턴스 식별자는 대소문자를 구분하지 않지만 'mydbinstance'와 같이 모두 소문자로 저장됩니다. 제약: 1~60자의 영숫자 또는 하이픈으로 구성되어야 합니다. 첫 번째 문자는 글자여야 합니다. 하이픈 2개가 연속될 수 없습니다. 하이픈으로 끝날 수 없습니다.

▼ 자격 증명 설정

미스터 사용자 이름 정보  
DB 인스턴스의 미스터 사용자에 로그인 ID를 입력하세요.

1~16자의 영숫자. 첫 번째 문자는 글자여야 합니다.

자격 증명 관리  
AWS Secrets Manager를 사용하거나 미스터 사용자 자격 증명을 관리할 수 있습니다.

AWS Secrets Manager에서 관리 - 가장 뛰어난 안정성  
RDS는 자동으로 암호를 생성하고 AWS Secrets Manager를 사용하여 전체 수명 주기 동안 암호를 관리합니다.

자체 관리  
사용자가 암호를 생성하거나 RDS에서 암호를 생성하고 사용자가 관리할 수 있습니다.

암호 자동 생성  
Amazon RDS에서 자동으로 암호를 생성하거나 사용자가 직접 암호를 지정할 수 있습니다.

마스터 암호 정보

Password strength Strong

최소 제약 조건: 8자 이상의 인쇄 가능한 ASCII 문자를 사용합니다. / " @ 기호는 포함할 수 없습니다.

마스터 암호 확인 정보

# Private Subnet에 RDS 설정

## 2. RDS 생성

**인스턴스 구성**  
아래의 DB 인스턴스 구성 옵션은 위에서 선택한 엔진에서 지원하는 옵션으로 제한됩니다.

**DB 인스턴스 클래스 정보**

▼ 필터 숨기기

Amazon RDS 최적화된 쓰기를 지원하는 인스턴스 클래스 표시 [정보](#)  
Amazon RDS 최적화된 쓰기는 추가 비용 없이 쓰기 처리량(throughput)을 최대 2배 늘립니다.

이전 세대 클래스 포함

스텠다드 클래스(m 클래스 포함)  
 메모리 최적화 클래스(r 및 x 클래스 포함)  
 버스터블 클래스(t 클래스 포함)

**db.t4g.micro**  
2 vCPUs 1 GiB RAM 네트워크: 최대 2,085Mbps

**스토리지**

**스토리지 유형** [정보](#)  
이제 프로비저닝된 IOPS SSD(io2) 스토리지 블룸을 사용할 수 있습니다.  
**범용 SSD(gp3)**  
스토리지와 독립적으로 성능 조정

**활당된 스토리지** [정보](#)  
20 GiB  
최소: 20GiB, 최대: 6,144GiB

DB 인스턴스의 스토리지를 수정하면 DB 인스턴스의 상태가 스토리지 최적화 상태가 됩니다. 스토리지 최적화 작업이 완료되어도 인스턴스는 계속 사용할 수 있습니다. [자세히 알아보기](#)

▶ 고급 설정  
400GiB 미만의 활당된 스토리지에는 3,000 IOPS의 기준 IOPS와 125MiBps의 스토리지 처리량이 포함됩니다.

▼ 스토리지 자동 조정

**스토리지 자동 조정** [정보](#)  
애플리케이션의 필요에 따라 데이터베이스 스토리지의 동적 조정 지원을 제공합니다.  
 스토리지 자동 조정 활성화  
이 기능을 활성화하면 지정한 임계값 초과 후 스토리지를 늘릴 수 있습니다.

**최대 스토리지 임계값** [정보](#)  
데이터베이스를 지정된 임계값으로 자동 조정하면 오금이 부과됩니다.  
1000 GiB  
활당된 스토리지 값은 22GiB~6,144GiB여야 합니다

# Private Subnet에 RDS 설정

## 2. RDS 생성

The screenshot shows the AWS RDS creation wizard, specifically the step for configuring a database in a private subnet.

**연결 정보**

**컴퓨팅 리소스**

- EC2 컴퓨팅 리소스에 연결 안 함
- EC2 컴퓨팅 리소스에 연결

이 데이터베이스의 컴퓨팅 리소스에 대한 연결을 설정하지 않습니다. 연결을 설정하면 컴퓨팅 리소스가 이 데이터베이스에 연결할 수 있도록 연결 설정이 자동으로 변경됩니다.

**EC2 인스턴스 정보**

이 데이터베이스의 컴퓨팅 리소스로 추가할 EC2 인스턴스를 선택합니다. VPC 보안 그룹이 이 EC2 인스턴스에 추가됩니다. VPC 보안 그룹은 EC2 인스턴스가 데이터베이스에 액세스하도록 허용하는 인바운드 규칙과 함께 데이터베이스에 추가됩니다.

i-04e7530e503d950f6  
awskrug-public-ec2

**컴퓨팅 리소스를 추가할 때 일부 VPC 설정을 변경할 수 없음**

EC2 컴퓨팅 리소스를 추가하면 이 데이터베이스에 대한 VPC, DB 서브넷 그룹 및 퍼블릭 액세스 설정이 자동으로 선택됩니다. EC2 인스턴스가 데이터베이스에 액세스할 수 있도록 하기 위해 VPC 보안 그룹 rds-ec2-X가 데이터베이스에 추가되고 ec2-rds-X라는 다른 그룹이 EC2 인스턴스에 추가됩니다. 컴퓨팅 리소스를 제거해야만 데이터베이스의 새 보안 그룹을 제거 할 수 있습니다.

**Virtual Private Cloud(VPC) 정보**

VPC를 선택합니다. VPC는 이 DB 인스턴스의 가장 네트워킹 환경을 정의합니다.

awskrug-vpc (vpc-0bb2da70c7d981a4f)  
4 서브넷, 3 가용 영역

해당 DB 서브넷 그룹이 있는 VPC만 나열됩니다.

**데이터베이스 awskrug-database 생성 중**

데이터베이스를 시작하는 데 몇 분 정도 걸릴 수 있습니다. DB 생성이 완료되는 동안 awskrug-database의 설정을 사용하여 제안된 데이터베이스 추가 기능의 구성을 간소화할 수 있습니다.

**DB 서브넷 그룹 정보**

DB 서브넷 그룹을 선택합니다. DB 서브넷 그룹은 선택한 VPC에서 DB 인스턴스가 어떤 서브넷과 IP 범위를 사용할 수 있는지를 정의합니다.

- 기존 항목 선택
- 자동 설정

기존 DB 서브넷 그룹  
awskrug-subnet-group  
3 서브넷, 3 가용 영역

**퍼블릭 액세스 정보**

- 예
- 아니요

RDS는 데이터베이스에 퍼블릭 IP 주소를 할당합니다. VPC 외부의 Amazon EC2 인스턴스 및 다른 리소스가 데이터베이스에 연결할 수 있습니다. 데이터베이스에 연결할 수 있는 리소스를 지정하는 VPC 보안 그룹을 하나 이상 선택합니다.

**VPC 보안 그룹(방화벽) 정보**

데이터베이스에 대한 액세스를 허용할 VPC 보안 그룹을 하나 이상 선택합니다. 보안 그룹 규칙이 적절한 수신 트래픽을 허용하는지 확인합니다.

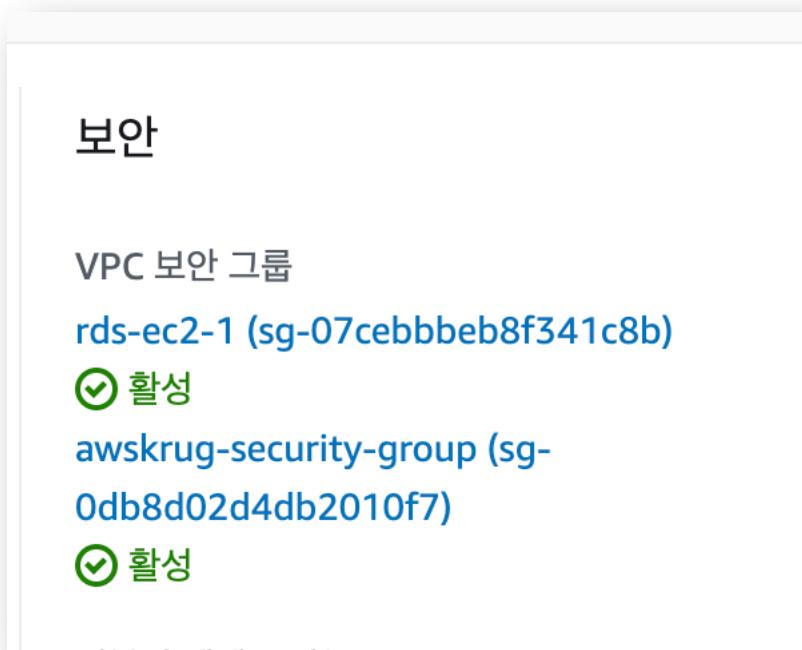
- 기존 항목 선택
- 새로 생성

기존 VPC 보안 그룹  
awskrug-security-group X

자격 증명 세부 정보 보기

# Private Subnet에 RDS 설정

## 2. RDS 생성



| 보안 그룹 (2) 정보  |      |                      |           |
|---|------|----------------------|-----------|
| <input type="text"/> Find resources by attribute or tag |      |                      |           |
| <input type="checkbox"/>                                | Name | 보안 그룹 ID             | 보안 그룹 이름  |
| <input type="checkbox"/>                                | -    | sg-07cebbbeb8f341c8b | rds-ec2-1 |
| <input type="checkbox"/>                                | -    | sg-0caa6fc6d8fdf3460 | ec2-rds-1 |

# Private Subnet에 RDS 설정

## 2. RDS 생성

rds-ec2-1

→ rds에 ec2가 접근할 수 있음



# Private Subnet에 RDS 설정

## 2. RDS 생성

ec2-rds-1

→ ec2가 rds로 나갈 수 있음

The screenshot shows the AWS CloudFormation console with a security group configuration. The 'Outbound Rules' tab is selected. A single rule is listed:

| 전 | 유형           | 프로토콜 | 포트 범위 | 대상                                     | 설명          |
|---|--------------|------|-------|--|-------------|
|   | MySQL/Aurora | TCP  | 3306  | <a href="#">sg-07cebbbeb8f341c8...</a> | Rule to RDS |

# Private Subnet에 RDS 설정

## 3. EC2 인스턴스를 통해 RDS 접속

### 1. Ec2 인스턴스 접속

```
ssh -i "awskrug.pem" ubuntu@43.202.14.56
```

### 2. 우분투에 mysql 설치

```
sudo apt update
```

```
sudo apt install mysql-client -y
```

```
ubuntu@ip-10-3-1-232:~$ sudo apt install mysql-client -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  mysql-client-8.0 mysql-client-core-8.0 mysql-common
The following NEW packages will be installed:
  mysql-client mysql-client-8.0 mysql-client-core-8.0 mysql-common
0 upgraded, 4 newly installed, 0 to remove and 0 not upgraded.
Need to get 2832 kB of archives.
After this operation, 61.8 MB of additional disk space will be used.
Ign:1 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client-core-8.0 amd64 8.0.39-0ubuntu0.24.04.1
Ign:2 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 mysql-common all 5.8+1.1.0build1
Ign:3 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client-8.0 amd64 8.0.39-0ubuntu0.24.04.1
Ign:4 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client all 8.0.39-0ubuntu0.24.04.1
Ign:1 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client-core-8.0 amd64 8.0.39-0ubuntu0.24.04.1
Ign:2 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 mysql-common all 5.8+1.1.0build1
Ign:3 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client-8.0 amd64 8.0.39-0ubuntu0.24.04.1
Ign:4 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client all 8.0.39-0ubuntu0.24.04.1
Ign:1 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client-core-8.0 amd64 8.0.39-0ubuntu0.24.04.1
Ign:2 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 mysql-common all 5.8+1.1.0build1
Ign:3 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client-8.0 amd64 8.0.39-0ubuntu0.24.04.1
Ign:4 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client all 8.0.39-0ubuntu0.24.04.1
Ign:1 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client-core-8.0 amd64 8.0.39-0ubuntu0.24.04.1
Err:2 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 mysql-common all 5.8+1.1.0build1
```

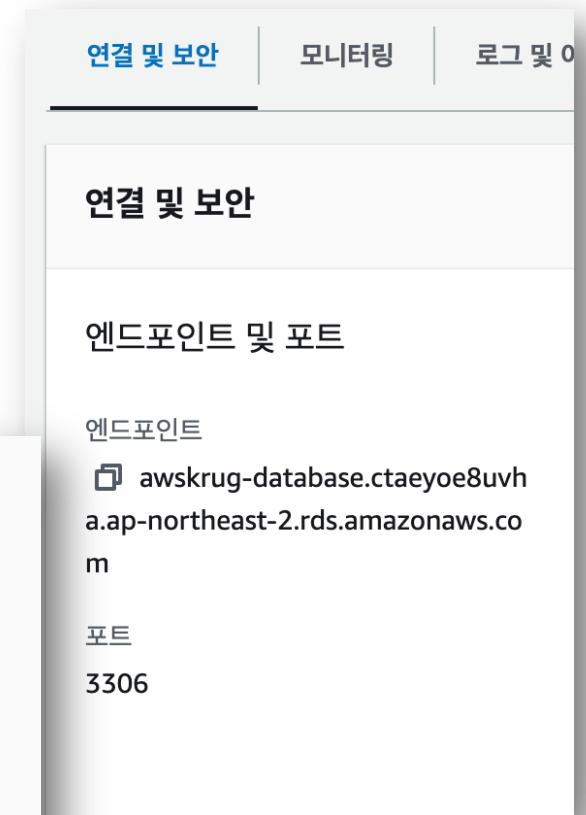
# Private Subnet에 RDS 설정

## 3. EC2 인스턴스를 통해 RDS 접속

### 3. RDS 접속

```
mysql -h <RDS 엔드포인트> -u <RDS 사용자 이름> -p
```

```
mysql> show databases;
+-----+
| Database |
+-----+
| awsjson_database |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.01 sec)
```



# Private Subnet에 RDS 설정

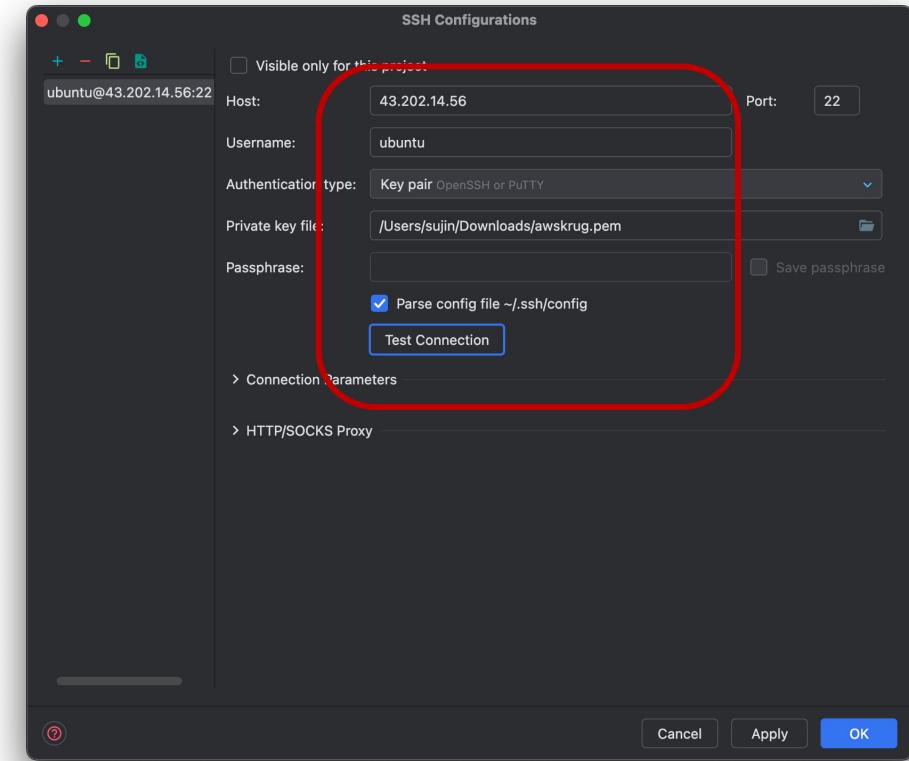
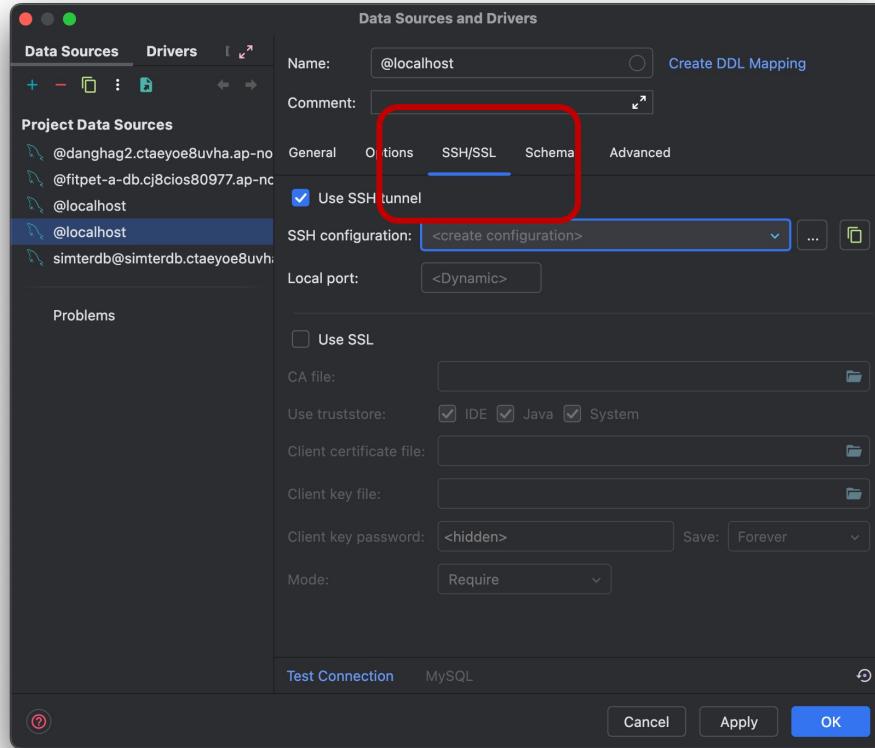
Datagrip에서 RDS ssh로 접속



# Private Subnet에 RDS 설정

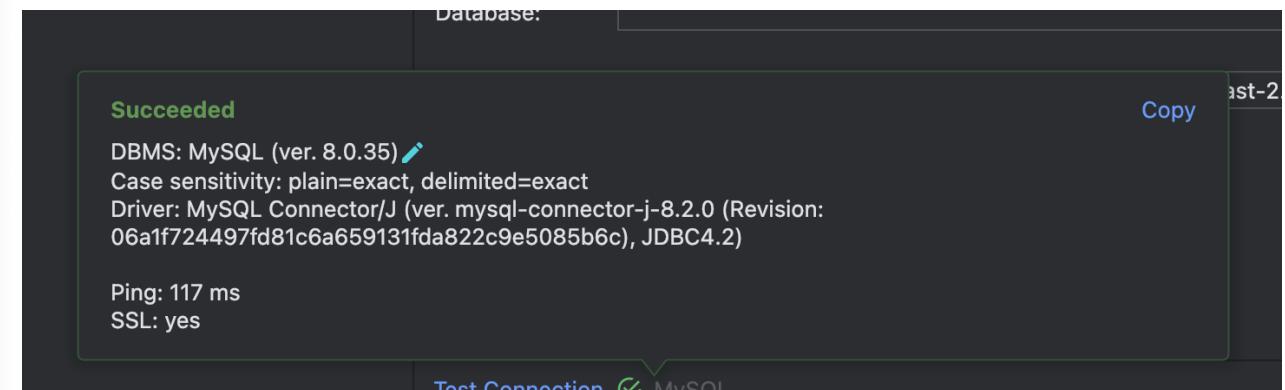
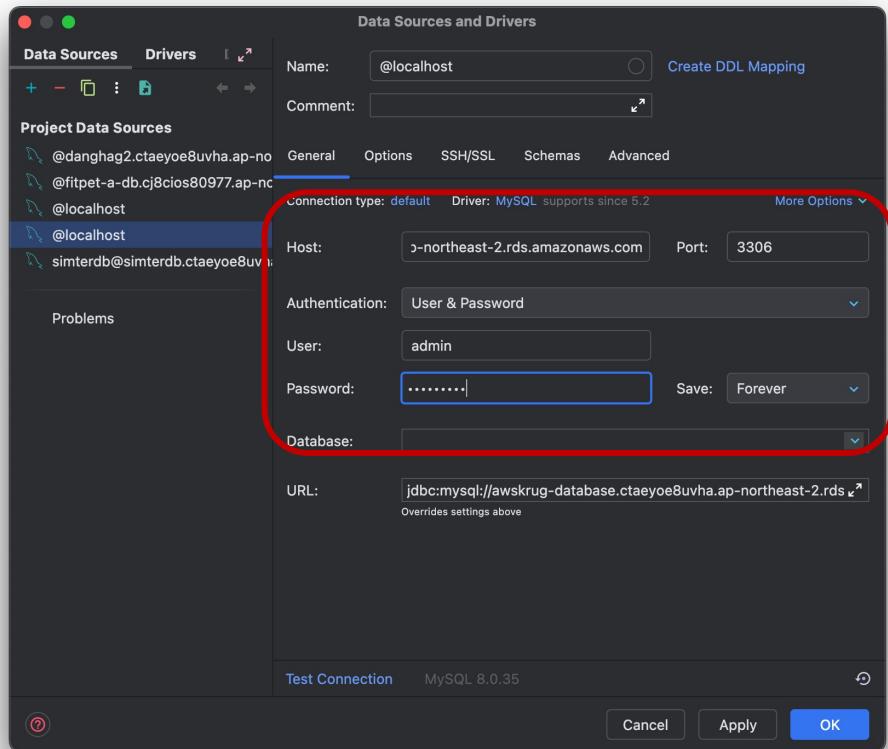
Datagrip에서 RDS ssh로 접속

EC2 public IP주소



# Private Subnet에 RDS 설정

Datagrip에서 RDS ssh로 접속



감사합니다

