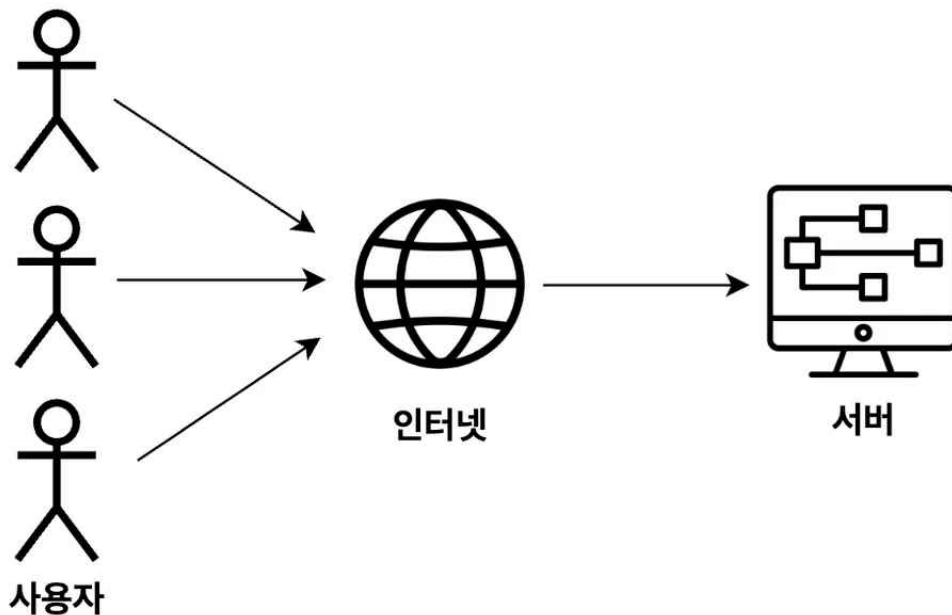


제 목	01장 AWS EC2 서버 세팅	
상세내용	백엔드 CI/CD에 필요한 AWS 지식	

## 1. 배포(Deployment)란?

### ✓ 배포(Deployment)란?

개발자들은 "이제 기능 구현도 끝났고 테스트도 끝났으니 배포하자!"라는 말을 자주 한다. 여기서 **배포**란 무슨 뜻일까? **배포(Deployment)**란 **다른 사용자들이 인터넷을 통해서 사용할 수 있게 만드는 걸 의미한다.** 쉽게 얘기해서 우리가 만든 웹 페이지나 서버를 다른 사람들이 사용하려면 인터넷 상에 배포가 되어있어야 한다.



자신의 컴퓨터에서 개발을 할 때는 localhost라는 주소로 테스트도 하고 개발을 한다. 하지만 이 localhost는 다른 컴퓨터에서는 접근이 불가능한 주소이다. 배포를 하게 되면 IP(ex. 124.16.2.1)나 도메인(ex. www.naver.com)과 같이 고유의 주소를 부여받게 되고, 다른 컴퓨터에서 그 주소로 접속할 수 있게 된다. 이게 바로 배포(Deployment)다.

따라서 어떤 서비스를 완성했다면, 그 다음 단계로 해야 하는 게 **배포(Deployment)**이다. 코딩을 배울 때도 기본적인 백엔드 서버를 만들 수 있는 역량이 갖춰졌다면, 그 다음에 배워야 하는 게 **배포(Deployment)**이다.

## 2. EC2란? / EC2를 왜 배울까? / 현업에서 EC2는 주로 언제 쓸까?

### ✓ EC2(Elastic Compute Cloud)란?

▶ 한 줄 요약 : 컴퓨터를 빌려서 원격으로 접속해 사용하는 서비스이다.

EC2를 쉽게 얘기하면 하나의 컴퓨터를 의미한다.

### ✓ EC2(Elastic Compute Cloud)를 왜 배울까

서버를 배포하기 위해서는 컴퓨터가 필요하다. 내가 가진 컴퓨터에서 서버를 배포해 다른 사용자들이 인터넷을 통해 접근할 수 있게 만들 수도 있다. 하지만 내 컴퓨터로 서버를 배포하면 24시간 동안 컴퓨터를 켜놔야 한다. 그리고 인터넷을 통해 내 컴퓨터에 접근할 수 있게 만들다보니 보안적으로도 위험할 수도 있다.

이러한 불편함 때문에 내가 가지고 있는 컴퓨터를 사용하지 않고, AWS EC2라는 컴퓨터를 빌려서 사용하는 것이다. 이 외에도 AWS EC2는 여러 부가기능들(로깅, 오토스케일링, 로드밸런싱 등)을 많이 가지고 있다

### ✓ 현업에서는 ?

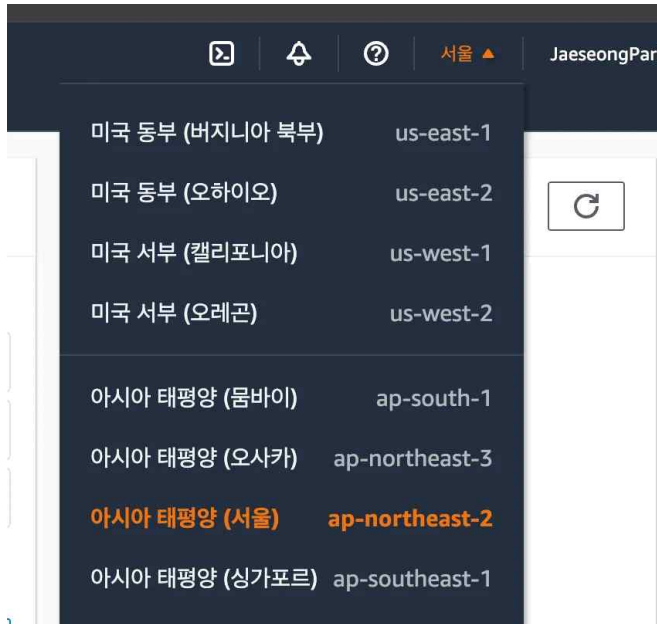
현업에서도 실제 서버를 배포할 때 AWS EC2를 아주 많이 사용한다. 백엔드 서버를 배포해야 할 때면 EC2에 서버를 배포해서 사용한다.

▶ “그러면 프론트엔드 웹 페이지를 배포할 때는 AWS EC2를 사용하지 않는걸까?”

프론트엔드 웹 페이지를 배포할 때 AWS EC2를 사용할 수도 있다. 하지만 AWS EC2보다 vercel, netlify 또는 AWS S3를 사용해서 주로 배포한다. 이게 어떤 것들인지 모를 경우 **\*\*“AWS EC2는 백엔드 서버를 배포할 때 주로 사용하는구나”**\*\*라고만 기억해도 괜찮다.

## [실습] 1. 리전(Region) 선택하기

✓ AWS EC2 서비스로 들어가서 리전(Region) 선택하기



AWS EC2를 시작하기 위해서는 우선적으로 리전(Region)을 먼저 선택해야 한다.

✓ 리전(Region)이란?

리전(Region)이란 인프라를 지리적으로 나누어 배포한 각각의 데이터 센터를 의미한다.



말이 어렵다. 조금 더 쉽게 풀어서 EC2에 대입해서 생각해보자. 우린 EC2가 **컴퓨터를 빌려서 원격으로 접속해 사용하는 서비스**라는 걸 알고 있다. 여기서 EC2를 통해 빌려서 쓸 수 있는 컴퓨터들이 전 세계적으로 다양하게 분포해있다. 이렇게 컴퓨터들이 위치한 위치를 보고 AWS에서는 **리전(Region)**이라고 한다.

## ✓ 리전(Region)의 특징

1. AWS는 전 세계적으로 다양한 Region을 보유하고 있다.



2. 각 Region은 고유의 이름을 가지고 있다.

- ex) us-east-1, eu-west-3

<div> <span>서울 ▲</span> </div>	
미국 동부 (버지니아 북부)	us-east-1
미국 동부 (오하이오)	us-east-2
미국 서부 (캘리포니아)	us-west-1
미국 서부 (오레곤)	us-west-2
아시아 태평양 (뭄바이)	ap-south-1
아시아 태평양 (오사카)	ap-northeast-3
<b>아시아 태평양 (서울)</b>	<b>ap-northeast-2</b>
아시아 태평양 (싱가포르)	ap-southeast-1
아시아 태평양 (시드니)	ap-southeast-2
아시아 태평양 (도쿄)	ap-northeast-1

## ✓ 리전(Region)은 어떤 기준으로 선택할까?

사람들이 애플리케이션을 사용할 때는 네트워크를 통해 통신하게 된다. 이 때, 사용자의 위치와 애플리케이션을 실행시키고 있는 컴퓨터와 위치가 멀면 멀수록 속도가 느려진다. 따라서 **\*\*애플리케이션의 주된 사용자들의 위치와 지리적으로 가까운 리전(Region)을 선택하는 것이 유리\*\***하다.

예를 들어, 한국 유저들이 주로 사용하는 서비스를 만들거라면 리전(Region)을 **\*\*아시아 태평양(서울)\*\***로 선택하면 된다.

## ✓ 많이 하는 실수

**\*\*아시아 태평양(서울)\*\*** 리전에서 EC2를 생성해놓고, 실수로 **\*\*미국 동부(버지니아 북부) 리전\*\***에 들어가서 생성한 EC2가 없어졌다고 당황하는 경우가 있다.

리전(Region)마다 EC2가 따로따로 관리가 되고 있으니 이 점 유의하자.

## [실습] 2. EC2 셋팅하기 - 기본 설정

### ✓ 1. 이름 및 태그

이름 및 태그 정보

이름

예: 내 웹 서버

추가 태그 추가

EC2의 이름을 설정하는 곳이다. 이름을 지을 때는 이 컴퓨터가 어떤 역할을 하는지 알아볼 수 있게 작성한다.

ex) instagram-server

## ✓ 2. Application and OS Images (Amazon Machine Image)

### Ubuntu 22.04 LTS 선택

Application and OS Images (Amazon Machine Image)
정보

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

수천 개의 애플리케이션 및 OS 이미지를 포함하는 전체 카탈로그 검색

Recents

Quick Start

Amazon Linux

aws

macOS

Mac

Ubuntu

ubuntu

Windows

Microsoft

Red Hat

Red Hat

SUSE Linux

SUSE

더 많은 AMI 찾아보기

AWS, Marketplace 및 커뮤니티의 AMI 포함

Amazon Machine Image(AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type

ami-086cae3329a3f7d75 (64-bit (x86)) / ami-0f8536fc6ad2ba267 (64-bit (Arm))

가상화: hvm ENA enabled: true 루트 디바이스 유형: ebs

프리 티어 사용 가능

설명

Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2023-09-19

아키텍처

AMI ID

64-bit (x86)

ami-086cae3329a3f7d75

확인된 공급 업체

OS를 선택하는 단계이다. OS(운영체제)란 Mac, Windows 7, Windows 10, Windows 11 같은 것들이 OS이다. 하지만 Windows나 Mac OS는 생각보다 용량도 많이 차지하고 성능도 많이 잡아먹는다. 그래서 서버를 배포할 컴퓨터의 OS는 훨씬 가벼운 **Ubuntu**를 많이 사용한다.

## ✓ 3. 인스턴스 유형

우선 **인스턴스**라는 뜻부터 정리하고 가자. **인스턴스**란, **AWS EC2에서 빌리는 컴퓨터 1대를 의미**한다.

그럼 **\*\*인스턴스 유형\*\***은 무슨 뜻일까? **\*\*컴퓨터 사양을 의미\*\***한다. 컴퓨터 사양이 좋으면 좋을수록 많은 수의 요청을 처리할 수 있고, 무거운 서버나 프로그램을 돌릴 수 있다.

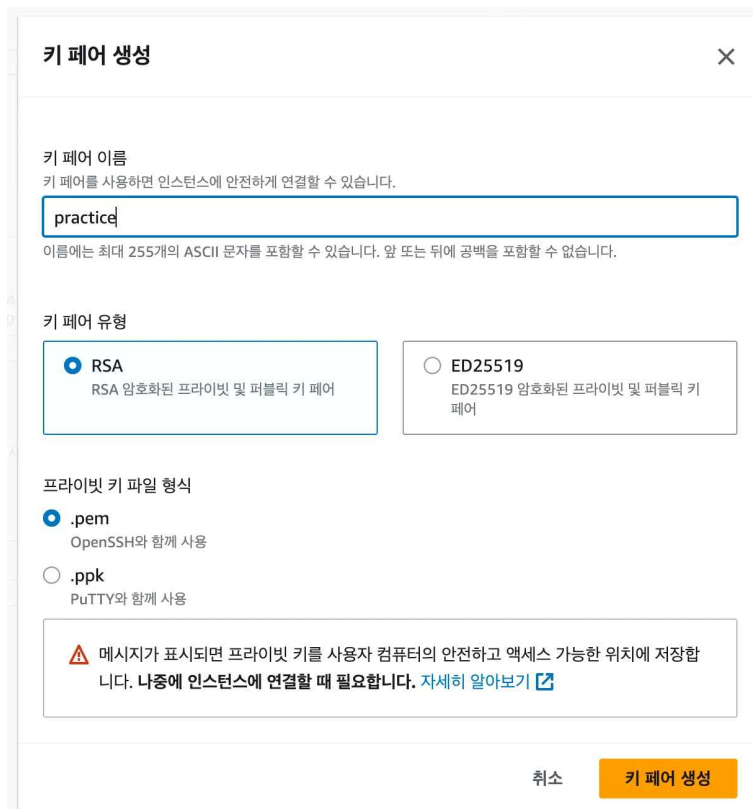
- 6 -

프리 티어에 해당하는 **t2.micro**를 사용할 것이다

여기서 많이들 오해하는 게 **t2.micro**는 학습할 때나 테스트할 때만 쓰는 안 좋은 사양의 컴퓨터라고 생각한다. 하지만 실제 서비스에서 활용해도 될 정도로 나름 괜찮은 사양이다. 하루 방문자 수가 2,000명 정도였던 서비스를 운영했었는데 문제 없이 잘 돌아갔다. 성능에 문제가 직접적으로 생기기 전까지는 너무 걱정하지 말자.

#### ✓ 4. 키 페어(로그인)

키 페어(Key Pair)는 무슨 뜻일까? EC2 컴퓨터에 접근할 때 사용하는 비밀번호라고 생각하면 된다. 말 그대로 열쇠(Key, 키)의 역할을 한다.



- **\*\*키 페어 이름\*\***은 어떤 EC2에 접근하기 위한 키 페어였는 지 알아볼 수 있게 지정하면 좋다.
- **`RSA`**와 **`.pem`**을 선택한 후에 키 페어를 생성하면 된다. **`.ED25519`**가 뭔지, **`.ppk`**가 뭔지는 몰라도 된다. 중요하지 않다.
- 키 페어를 생성하면 파일이 하나 다운받아질텐데, 그 파일은 잃어버리면 안 되니 잘 보관해놔야 한다.
- **\*\*참고\*\*** 실습에서는 `키 페어`를 활용해서 EC2에 접근하지 않고, 더 편한 방법으로 접근할 예정이다.

## [실습] 3. EC2 셋팅하기 - 보안그룹 설정

### ✓ 네트워크 설정

▼ 네트워크 설정 정보

편집

네트워크 정보

vpc-d0897abb

서브넷 정보

기본 설정 없음(가용 영역의 기본 서브넷)

퍼블릭 IP 자동 할당 정보

활성화

Firewall (security groups) 정보

보안 그룹은 인스턴스에 대한 트래픽을 제어하는 방화벽 규칙 세트입니다. 특정 트래픽이 인스턴스에 도달하도록 허용하는 규칙을 추가합니다.

Create security group

Select existing security group

We'll create a new security group called 'launch-wizard-7' with the following rules:

☒ Allow SSH traffic from
 

위치 무관  
0.0.0.0/0

Helps you connect to your instance

☐ 인터넷에서 HTTPS 트래픽 허용
 

예를 들어 웹 서버를 생성할 때 엔드포인트를 설정하려면

☐ 인터넷에서 HTTP 트래픽 허용
 

예를 들어 웹 서버를 생성할 때 엔드포인트를 설정하려면

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

×



▼ 네트워크 설정 정보

VPC - 필수 정보

vpc-d0897abb

(기본값) ▼

↻

172.31.0.0/16

서브넷 정보

기본 설정 없음

▼

↻

새 서브넷 생성 ↗

퍼블릭 IP 자동 할당 정보

활성화

▼

Firewall (security groups) 정보

보안 그룹은 인스턴스에 대한 트래픽을 제어하는 방화벽 규칙 세트입니다. 특정 트래픽이 인스턴스에 도달하도록 허용하는 규칙을 추가합니다.

☒ Create security group

☐ Select existing security group

보안 그룹 이름 - 필수

launch-wizard-7

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and .\_-:/()#,@[]+=&{}!\$\*

설명 - 필수 정보

launch-wizard-7 created 2023-11-10T09:23:06.868Z

인바운드 보안 그룹 규칙

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

제거

유형 정보

ssh ▼

프로토콜 정보

TCP

포트 범위 정보

22

소스 유형 정보

위치 무관 ▼

원본 정보

Q CIDR, 접두사 목록 또는 보안 그룹

0.0.0.0/0 X

설명 - optional 정보

e.g. SSH for admin desktop

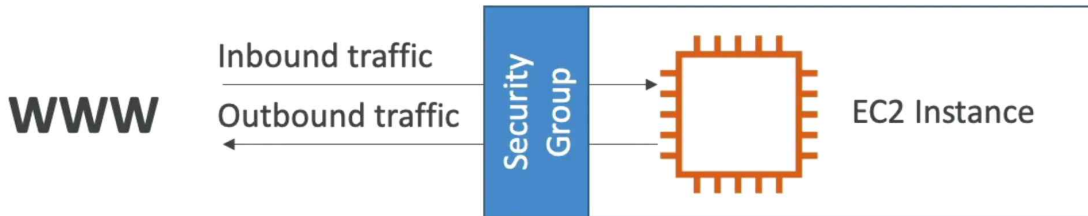
네트워크 설정 칸을 보면 **\*\*VPC\*\***와 **\*\*Security Groups(보안 그룹)\*\***가 보인다. 여기서 VPC라는 개념은 AWS를 입문하는 입장에서는 크게 중요하지 않으니 넘어가자. 나중에 AWS에 어느 정도 익숙해졌을 때 VPC를 학습하도록 하자. VPC를 몰라도 서버를 배포하는 데 아무 문제가 없다.

하지만 **\*\*Security Groups(보안 그룹)\*\***은 서버를 배포할 때 중요한 개념이므로 자세히 알아보자.

- 9 -

## ✓ 보안 그룹(Security Group)이란?

보안 그룹(Security Group)이란 AWS 클라우드에서의 네트워크 보안을 의미한다.



**EC2 인스턴스**를 **집**이라고 생각한다면, **보안 그룹**은 **집 바깥 쪽에 쳐져있는 울타리와 대문**이라고 생각하면 된다. 집에 접근할 때 울타리의 대문에서 접근해도 되는 요청인지 보안 요원이 검사를 하는 것과 비슷하다.

인터넷에서 일부 사용자가 EC2 인스턴스에 접근(액세스)하려고 한다고 가정해보자. 위 그림과 같이 EC2 인스턴스 주위에 방화벽 역할을 할 **보안 그룹(Security Group)**을 만들고 보안 그룹에 규칙을 지정한다. 이 보안 규칙에는 **인바운드 트래픽(즉, 외부에서 EC2 인스턴스로 보내는 트래픽)**에서 어떤 트래픽만 허용할 지 설정할 수 있고, **아웃바운드 트래픽(즉, EC2 인스턴스에서 외부로 나가는 트래픽)**에서 어떤 트래픽만 허용할 지 설정할 수 있다.

보안 그룹을 설정할 때는 허용할 **IP 범위**와 **포트(port)**를 설정할 수 있다.

## ✓ 보안그룹 설정

외부에서 EC2로 접근할 포트는 **22번 포트**와 **80번 포트**라고 생각해서 이 2가지에 대해 인바운드 보안 그룹 규칙을 추가했다. 왜냐하면 22번 포트는 우리가 EC2에 원격 접속할 때 사용하는 포트이고, 80번 포트에는 백엔드 서버를 띄울 예정이기 때문이다. 그리고 어떤 IP에서든 전부 접근할 수 있게 만들기 위해 소스 유형은 **원치 무관**으로 설정했다.

### 인바운드 보안 그룹 규칙

▼ 보안 그룹 규칙 1 (TCP, 22, 0.0.0.0/0)

제거

#### 유형 정보

ssh ▼

#### 프로토콜 정보

TCP

#### 포트 범위 정보

22

#### 소스 유형 정보

위치 무관 ▼

#### 원본 정보

Q CIDR, 접두사 목록 또는 보안 그룹

0.0.0.0/0 X

#### 설명 - optional 정보

예: 관리자 데스크톱용 SSH

▼ 보안 그룹 규칙 2 (TCP, 80, 0.0.0.0/0)

제거

#### 유형 정보

HTTP ▼

#### 프로토콜 정보

TCP

#### 포트 범위 정보

80

#### 소스 유형 정보

위치 무관 ▼

#### 원본 정보

Q CIDR, 접두사 목록 또는 보안 그룹

0.0.0.0/0 X

#### 설명 - optional 정보

예: 관리자 데스크톱용 SSH

## [실습] 4. EC2 셋팅하기 - 스토리지 구성

### ✓ 스토리지 구성

▼ 스토리지 구성 정보

Advanced

1x 8 GiB gp3

루트 볼륨 (암호화되지 않음)

새 볼륨 추가

0 x 파일 시스템

편집

▼ Storage (volumes)
정보
Simple

EBS 볼륨

세부 정보 숨기기

▶ 볼륨 1 (AMI 루트) (8 GiB, EBS, General purpose SSD (gp3))

새 볼륨 추가

파일 시스템

세부 정보 표시

우리가 쓰고 있는 노트북이나 데스크톱 컴퓨터는 전부 하드디스크를 가지고 있다. 하드디스크는 컴퓨터에서 파일을 저장하는 공간이다. EC2도 하나의 컴퓨터이다보니 여러 파일들을 저장할 저장 공간이 필요하다. 이 저장 공간을 보고 **EBS(Elastic Block Storage)**라고 부른다. 즉, EBS란 EC2 안에 부착되어 있는 일종의 하드디스크라고 생각하면 된다. EBS와 같은 저장 공간을 조금 더 포괄적인 용어로 **스토리지(Storage), 볼륨(Volume)**이라고 부른다.

## ✓ 셋팅

▼ 스토리지 구성
정보
Advanced

1x

30

GiB

gp3

▼

루트 볼륨 (암호화되지 않음)

새 볼륨 추가

0 x 파일 시스템

편집

스토리지의 종류를 보면 gp3 이외에도 여러가지 종류의 스토리지가 있다. 하지만 가성비가 좋은 **gp3**를 선택해주자. 용량을 30GiB를 설정한 이유는 프리 티어에서 **30GiB**까지 무료로 제공해주기 때문이다. 이 스토리지의 크기는 추후에 늘릴 수도 있으므로 처음 설정할 때 너무 큰 고민을 할 필요는 없다.

## [실습] 5. EC2 접속하기

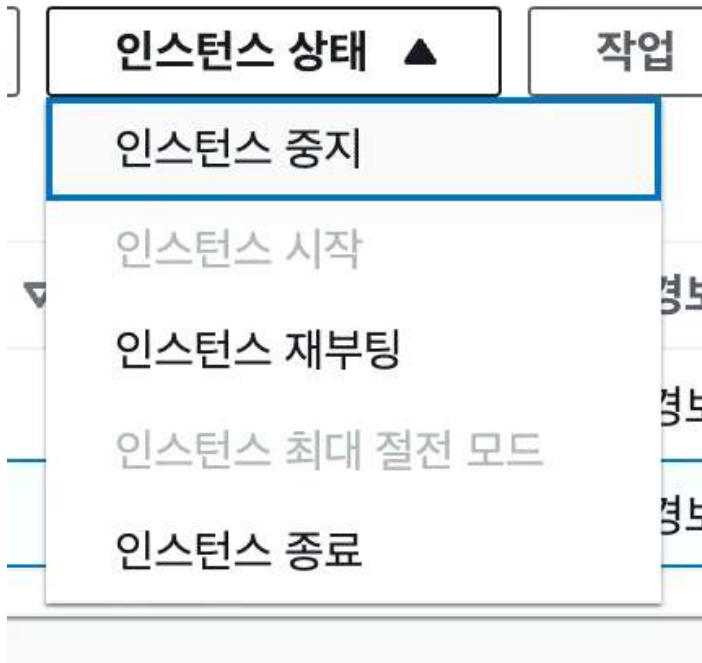
### ✓ 생성된 인스턴스 정보 해석하기

#### 1. 세부 정보

The screenshot displays the AWS Management Console for an EC2 instance. The instance is named 'i-08ac6537770093abe(testtest)'. The 'Instance state' is 'Running' (실행 중), and the 'Public IPv4 address' is '15.165.203.130'. Other details include the instance type 't2.micro', VPC ID 'vpc-d0897abb', and subnet 'subnet-3a9dcc76'. The console also shows the instance's IP address, DNS name, and other configuration details.

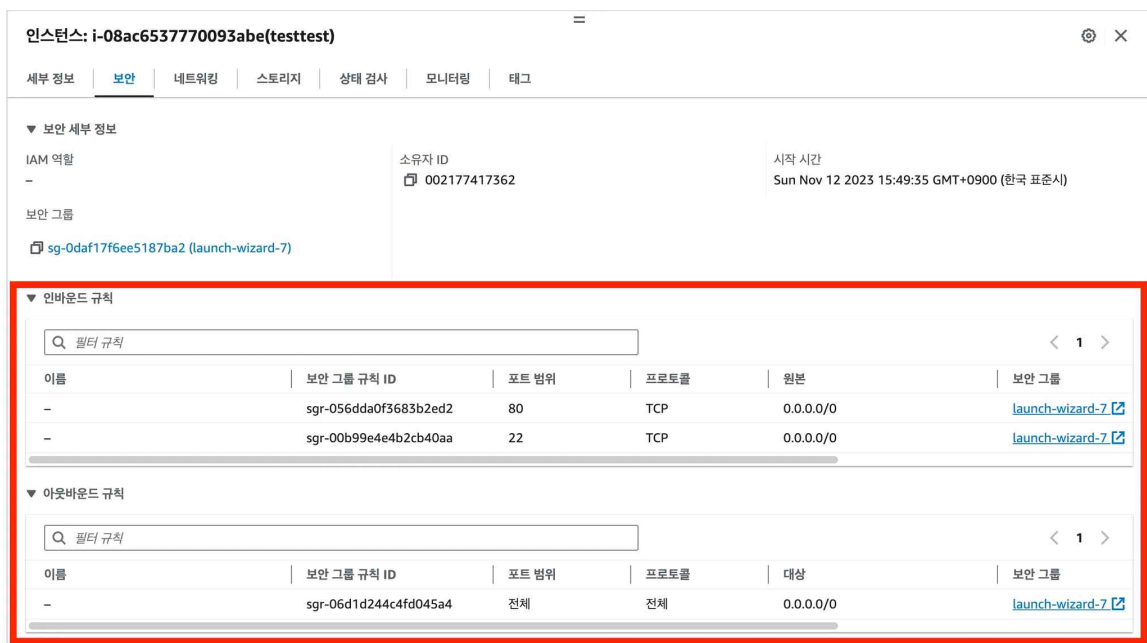
세부 정보에서 눈여겨 봐야 할 부분은 2가지 밖에 없다. **\*\*퍼블릭 IPv4 주소\*\*와 \*\*인스턴스 상태\*\***이다.

- **\*\*퍼블릭 IPv4 주소\*\***는 EC2 인스턴스가 생성되면서 부여받은 IP 주소이다. EC2 인스턴스에 접근하려면 이 IP 주소로 접근하면 된다.
- **\*\*인스턴스 상태\*\***는 말그대로 EC2 인스턴스가 **\*\*실행 중\*\***이라는 뜻은 컴퓨터가 켜져있다는 뜻이다.



EC2 인스턴스를 **중지**, **재부팅**, **종료**도 할 수 있다. 우리가 쓰는 컴퓨터와 아주 유사하다. **재부팅**은 말그대로 컴퓨터를 재시작시키는 걸 의미하고, **중지**는 컴퓨터를 잠시 꺼놓는 걸 의미한다. **종료**는 컴퓨터를 아예 삭제시킨다는 걸 의미한다. **EC2 인스턴스를 한 번 종료하면 도중에 취소할 수 없으니 조심해야 한다.**

## 2. 보안 (보안 그룹)



인스턴스 생성 시 설정한 보안 그룹에 대한 정보가 나온다.

### 3. 네트워크

인스턴스: i-08ac6537770093abe(testtest)

네트워킹 세부 정보

퍼블릭 IPv4 주소: 15.165.203.130 [개방 주소법]

퍼블릭 IPv4 DNS: ec2-15-165-203-130.ap-northeast-2.compute.amazonaws.com [개방 주소법]

서브넷 ID: subnet-3a9dccc76

가용 영역: ap-northeast-2c

RBN을 게스트 OS 호스트 이름으로 사용: 비활성

프라이빗 IPv4 주소: 172.31.31.144

프라이빗 IP DNS 이름(IPv4만 해당): ip-172-31-31-144.ap-northeast-2.compute.internal

IPv6 주소: -

통신 사업자 IP 주소(임시): -

RBN DNS 호스트 이름 IPv4 응답: 활성화

VPC ID: vpc-d0897abb

보조 프라이빗 IPv4 주소: -

Outpost ID: -

네트워킹 인터페이스 (1) 정보

인터페이스 ID	설명	IPv4 접두사	IPv6 접두사	퍼블릭 IPv4 주소	프라이빗 IPv4 주소	프라이빗 IPv4 DNS	IPv6 주소
eni-07e9533b15e4e3d17	-	-	-	15.165.203.130	172.31.31.144	ip-172-31-31-144.ap-...	-

탄력적 IP 주소 (0) 정보

이름	할당된 IPv4 주소	유형	주소 풀	할당 ID
이 인스턴스와 연결된 탄력적 IP 주소가 없음				

퍼블릭 IPv4 주소는 생성한 EC2 인스턴스의 IP 주소를 뜻한다.

### 4. 스토리지

인스턴스: i-08ac6537770093abe(testtest)

루트 디바이스 세부 정보

루트 디바이스 이름: /dev/sda1

루트 디바이스 유형: EBS

EBS 최적화: 비활성

블록 디바이스

블록 ID	디바이스 이름	볼륨 크기(GiB)	연결 상태	연결 시간	암호화됨	KMS 키 ID	종료 시 삭제
vol-0df4f58148431b596	/dev/sda1	30	연결됨	2023/11/12 15:49 GMT+9	아니요	-	예

최근 루트 볼륨 대체 작업

작업 ID	작업 상태	시작 시간	완료 시간	태그
최근 루트 볼륨 대체 작업 없음				

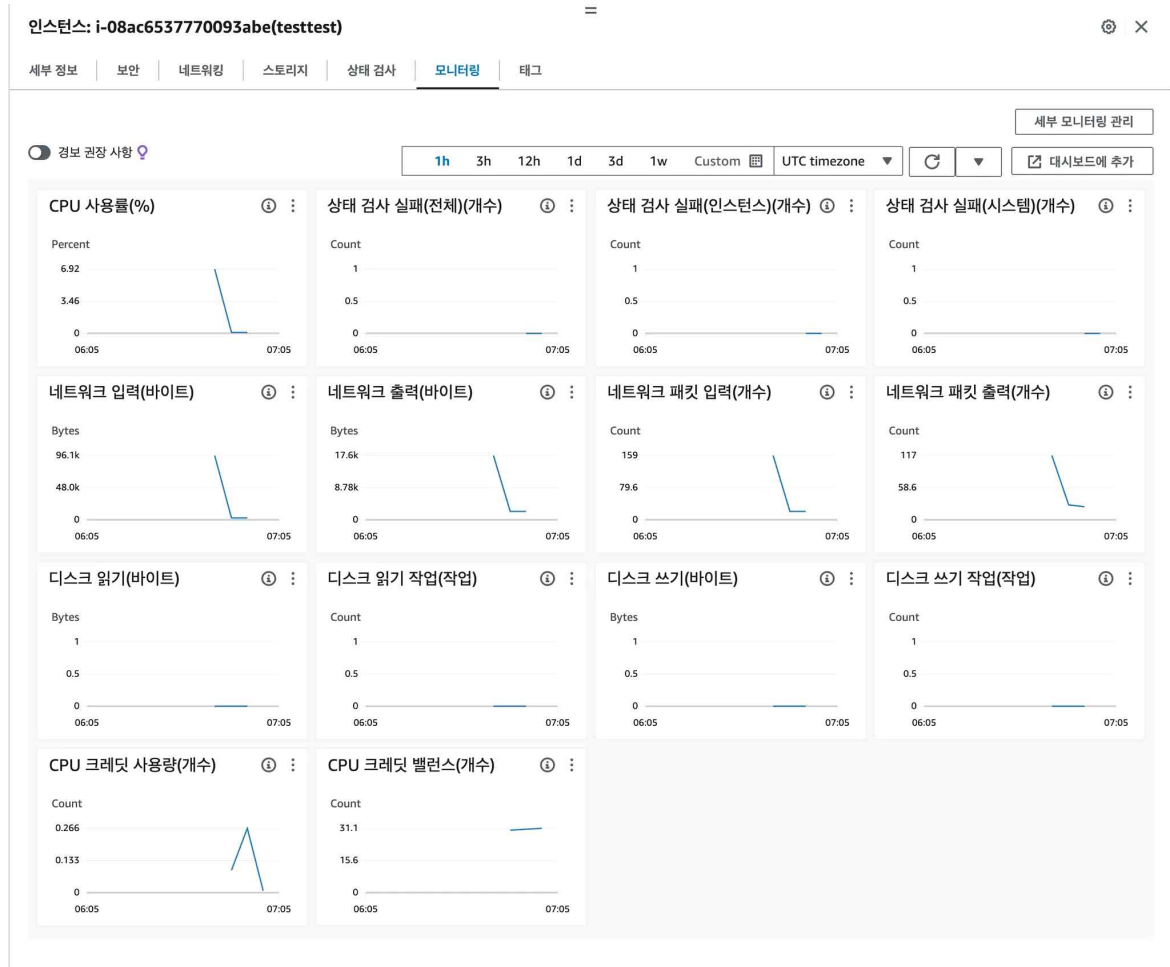
인스턴스 생성 시 설정한 스토리지에 대한 정보가 나온다.

### 5. 상태 검사

(잘 안 보는 창이다.)

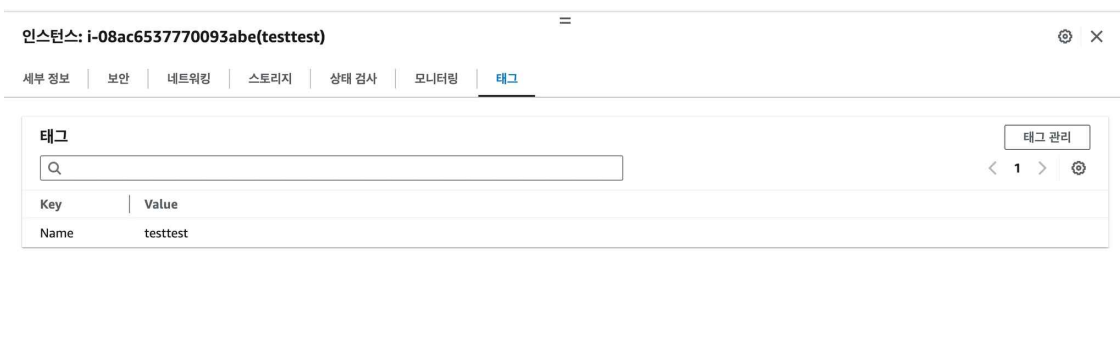
## 6. 모니터링

EC2 인스턴스에 관련한 여러가지 정보를 볼 수 있는 창이다. AWS를 처음 입문할 때는 자주 볼 일이 없는 창이지만, 나중에 **실제 서버를 운영할 때는 자주 보게되는 창**이다. EC2 인스턴스가 정상적으로 작동하고 있는 지, EC2 인스턴스의 성능을 향상시켜주어야 하는 건 아닌 지 아래 지표를 통해 파악할 수 있다.



## 7. 태그

(잘 안 보는 창이다.)





## ✓ EC2에 접속하기

EC2 > 인스턴스 > i-03ef50530634ac976 > 인스턴스에 연결

### 인스턴스에 연결 정보

다음 옵션 중 하나를 사용하여 인스턴스 i-03ef50530634ac976 (practice)에 연결

EC2 인스턴스 연결

Session Manager

SSH 클라이언트

EC2 직렬 콘솔

인스턴스 ID  
i-03ef50530634ac976 (practice)

연결 유형

☒ EC2 Instance Connect를 사용하여 연결  
퍼블릭 IPv4 주소가 있는 EC2 인스턴스 연결 브라우저 기반 클라이언트를 사용하여 연결합니다.

☐ EC2 인스턴스 연결 엔드포인트를 사용하여 연결  
프라이빗 IPv4 주소 및 VPC 엔드포인트가 있는 EC2 인스턴스 연결 브라우저 기반 클라이언트를 사용하여 연결합니다.

퍼블릭 IP 주소  
15.164.61.142

사용자 이름  
인스턴스를 시작하는 데 사용되는 AMI에 정의된 사용자 이름을 입력합니다. 사용자 지정 사용자 이름을 정의하지 않은 경우 기본 사용자 이름인 ubuntu를(를) 사용합니다.  
ubuntu

**참고:** 대부분의 경우 기본 사용자 이름 ubuntu은(는) 정확합니다. 하지만 AMI 사용 지침을 읽고 AMI 소유자가 기본 AMI 사용자 이름을 변경했는지 확인하십시오.

취소 **연결**

## [실습] 6. 탄력적 IP 연결하기

### ✓ 탄력적 IP가 왜 필요할까?

EC2 인스턴스를 생성하면 IP를 할당받는다. 하지만 이렇게 할당받은 IP는 임시적인 IP이다. EC2 인스턴스를 잠깐 중지시켰다가 다시 실행시켜보면 IP가 바뀌어있다. EC2 인스턴스를 중지시켰다가 다시 실행시킬 때마다 IP가 바뀌면 굉장히 불편하다. 그래서 중지시켰다가 다시 실행시켜도 바뀌지 않는 고정 IP를 할당받아야 한다.

그게 바로 **탄력적 IP**이다

현업에서도 EC2 인스턴스를 생성하면 탄력적 IP를 대부분 필수적으로 설정한다

▼ 이미지

AMI

AMI 카탈로그

▼ Elastic Block Store

볼륨

스냅샷

수명 주기 관리자

▼ 네트워크 및 보안

보안 그룹

**탄력적 IP**

배치 그룹

탄력적 IP 주소 (1/1)

Q 탄력적 IP 주소 필터링



Name



jscode

## [실습] 7. Express 서버를 EC2에 배포하기

### ✓ 1. Ubuntu 환경에서 Node.js 설치하는 법

Ubuntu 환경에서 Express 서버를 실행시키려면 Node.js가 설치되어 있어야 한다.  
그래서 Ubuntu에 우선 Node.js를 설치해보자

깃: <https://github.com/nodesource/distributions>

```
$ sudo su
$ apt-get update && /
apt-get install -y ca-certificates curl gnupg && /
mkdir -p /etc/apt/keyrings && /
curl -fsSL https://deb.nodesource.com/gpgkey/nodesource-repo.gpg.key | sudo
o gpg --dearmor -o /etc/apt/keyrings/nodesource.gpg && /
NODE_MAJOR=20 && /
echo "deb [signed-by=/etc/apt/keyrings/nodesource.gpg] https://deb.nodesour
ce.com/node_${NODE_MAJOR}.x nodistro main" | sudo tee /etc/apt/sources.list.
d/nodesource.list && /
apt-get update && /
apt-get install nodejs -y
```

## ✓ 2. 잘 설치됐는 지 확인하기

```
$ node -v
```

```
root@ip-172-31-31-144:/home/ubuntu# node -v
v20.9.0
```

## ✓ 3. Github으로부터 Express 프로젝트 clone하기

Github : <https://github.com/JSCODE-EDU/ec2-express-sample>

```
$ git clone https://github.com/JSCODE-EDU/ec2-express-sample
$ cd ec2-express-sample
$ npm i
```

## ✓ 4. .env 파일 직접 만들기

.env와 같은 민감한 파일은 Git으로 버전 관리를 하지 않는게 일반적이다. 따라서 .env 파일은 별도로 EC2 인스턴스에 올려주어야 한다. 하지만 .env 파일을 EC2 인스턴스에 올리는 작업보다는, .env

.env

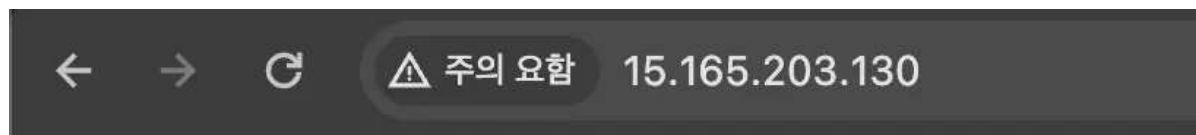
```
DATABASE_NAME=my_database
```

## ✓ 5. pm2 설치해서 서버 실행시키기

Node 기반의 서버는 pm2를 활용해서 많이 실행시킨다. 서비스를 운영하는 데 있어서 유용한 기능들을 pm2가 많이 가지고 있기 때문이다.

```
$ sudo npm i -g pm2
$ sudo pm2 start app.js
```

## ✓ 6. 잘 작동하는 지 확인하기



.env 테스트 : my\_database

## [실습] 7-2. Spring Boot 서버를 EC2에 배포하기

### ✓ 1. Ubuntu 환경에서 JDK 설치하는 법

Spring Boot는 3.x.x 버전을 사용할 예정이고, JDK는 17버전을 사용할 예정이다. 그에 맞게 환경을 설치해보자.

```
$ sudo apt update && /
sudo apt install openjdk-17-jdk -y
```

## ✓ 2. 잘 설치됐는 지 확인하기

```
$ java -version
```

```
ubuntu@ip-172-31-40-248:~$ java -version
openjdk version "17.0.8.1" 2023-08-24
OpenJDK Runtime Environment (build 17.0.8.1+1-Ubuntu-0ubuntu122.04)
OpenJDK 64-Bit Server VM (build 17.0.8.1+1-Ubuntu-0ubuntu122.04, mixed mode, sharing)
```

## ✓ 3. Github으로부터 Spring Boot 프로젝트 clone하기

Github : <https://github.com/JSCODE-EDU/ec2-spring-boot-sample>

```
$ git clone https://github.com/JSCODE-EDU/ec2-spring-boot-sample.git
$ cd ec2-spring-boot-sample
```

## ✓ 4. **application.yml** 파일 직접 만들기

**application.yml**와 같은 민감한 정보가 포함된 파일은 Git으로 버전 관리를 하지 않는게 일반적이다. 따라서 **application.yml** 파일은 별도로 EC2 인스턴스에 올려주어야 한다. 하지만 **application.yml** 파일을 EC2 인스턴스에 올리는 작업보다는, **application.yml** 파일을 직접 만드는 게 훨씬 간단하다.

**src/main/resources/application.yml**

```
server:
  port: 80
```

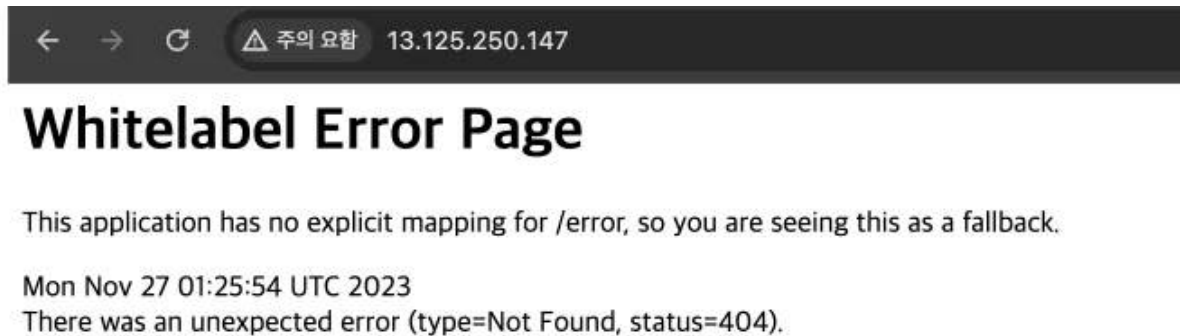
## ✓ 5. 서버 실행시키기

```
$ ./gradlew clean build # 기존 빌드된 파일을 삭제하고 새롭게 JAR로 빌드
$ cd ~/ec2-spring-boot-sample/build/libs
$ sudo java -jar ec2-spring-boot-sample-0.0.1-SNAPSHOT.jar
```

참고) 백그라운드에서 Spring Boot 실행시키기

```
$ sudo nohup java -jar ec2-spring-boot-sample-0.0.1-SNAPSHOT.jar &
```

## ✓ 6. 잘 작동하는 지 확인하기



## [실습] 8. 비용 나가지 않게 EC2 깔끔하게 종료하기

### ✓ 1. EC2 인스턴스 종료하기



## ✓ 2. 탄력적 IP 릴리스하기

