



11_AWS 웹 호스팅을 이용한 프로젝트 배포

내용

? AWS(Amazon Web Services)

- 클라우드 컴퓨팅 및 관련 서비스를 제공하는 아마존 닷컴의 클라우드 컴퓨팅 플랫폼
 - 서버, 데이터베이스, 저장소, 네트워크, 인공지능, 머신 러닝, 개발 도구, 보안 등 다양한 IT 리소스와 서비스를 클라우드에서 제공
-

⚡ EC2(클라우드 가상 컴퓨팅) 이용 웹 호스팅

(가입 후 12개월 동안 1개의 인스턴스 Free Tier(무료) 이용)



잘못 이용하면 요금이 부과될 수 있습니다! (미사용 시 모든 인스턴스 삭제 요망)

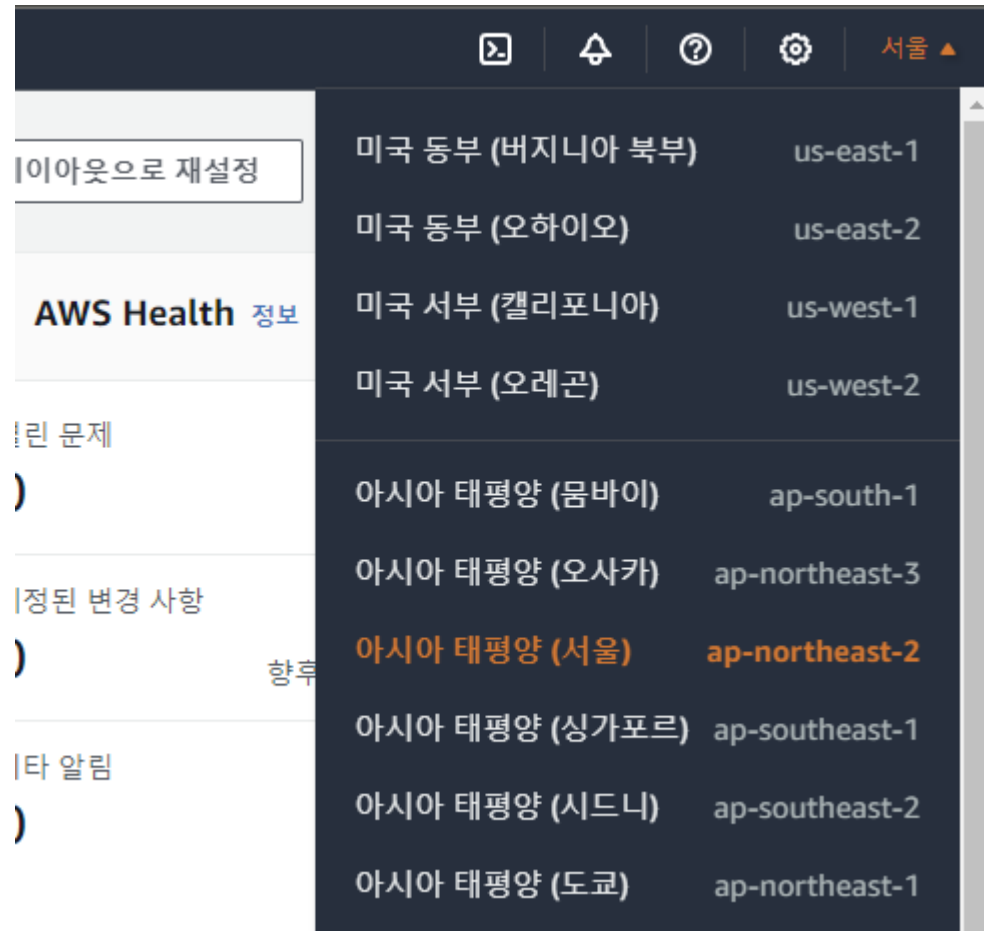
사용량 확인은 로그인 후 우측 상단 계정 → 결제 대시보드 → 프리티어 탭에서 확인

1. 회원 가입

- 개인 정보 영어로 작성(영어 이름, 영어 주소)
- 해외 결제 가능한 신용/체크 카드필요

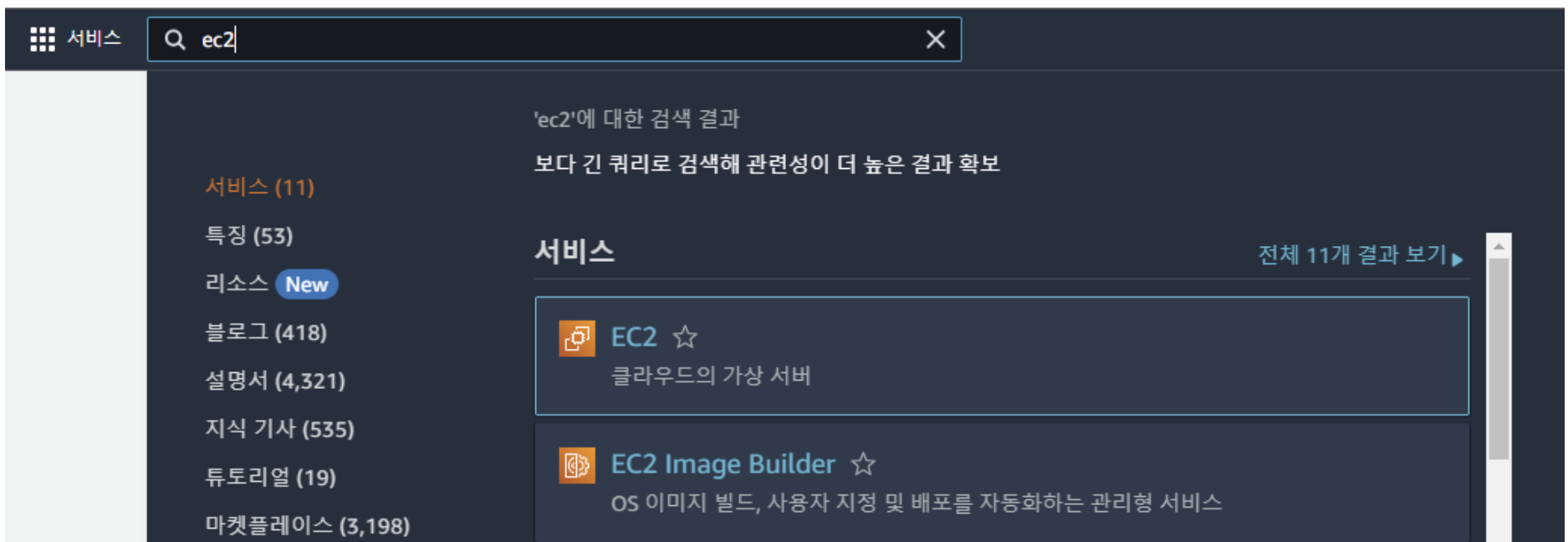
2. Region(지역) 설정

- 로그인 → 콘솔 홈 → 우측 상단



3. EC2 대시보드로 이동

- 상단 검색에 "ec2" 검색 후 이동



4. 인스턴스(가상 컴퓨터) 생성하기

EC2 대시보드

EC2 글로벌 보기

이벤트

▼ 인스턴스

인스턴스

인스턴스 유형

시작 템플릿

스팟 요청

Savings Plans

예약 인스턴스

전용 호스트

용량 예약 [New](#)

▼ 이미지

AMI

AMI 카탈로그

▼ Elastic Block Store

볼륨

스냅샷

수명 주기 관리자

▼ 네트워크 및 보안

보안 그룹

리소스

EC2 Global view

아시아 태평양 (서울) 리전에서 다음 Amazon EC2 리소스를 사용하고 있음:

인스턴스(실행 중)	0	로드 밸런서	0
배치 그룹	0	보안 그룹	1
볼륨	0	스냅샷	0
인스턴스	0	전용 호스트	0
키 페어	0	탄력적 IP	0
Auto Scaling 그룹	0		

인스턴스 시작

시작하려면 클라우드의 가장 서버인 Amazon EC2 인스턴스를 시작하십시오.

인스턴스 시작 ▼

서버 마이그레이션

참고: 인스턴스는 아시아 태평양 (서울) 리전에서 시작됩니다.

서비스 상태

AWS Health 대시보드

리전

아시아 태평양 (서울)

영역

영역 이름	영역 ID
-------	-------

이름 : 자유롭게 작성

AMI(Amazon Machine Image) : Amazon Linux(프리 티어)

아키텍처 : 64비트(x86)

인스턴스 유형 : t2.micro(프리 티어)

이름 및 태그 정보

이름

my server

추가 태그 추가

▼ 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

Quick Start

Amazon Linux

aws

macOS

Mac

Ubuntu

ubuntu

Windows

Microsoft

Red Hat

Red Hat

SUSE L

SUS

더 많은 AMI 찾아보기

AWS, Marketplace 및 커뮤니티의 AMI 포함

Amazon Machine Image(AMI)

Amazon Linux 2023 AMI

ami-01123b84e2a4fba05 (64비트(x86)) / ami-0c68ab5091e5f073a (64비트(Arm))

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

프리 티어 사용 가능

설명

Amazon Linux 2023 AMI 2023.2.20231113.0 x86_64 HVM kernel-6.1

아키텍처

64비트(x86)

AMI ID

ami-01123b84e2a4fba05

확인된 공급 업체

▼ 인스턴스 유형 정보

인스턴스 유형

t2.micro

프리 티어 사용 가능

온디맨드 RHEL 기본 요금: 0.0744 USD 시간당

온디맨드 Linux 기본 요금: 0.0144 USD 시간당

온디맨드 SUSE 기본 요금: 0.0144 USD 시간당

온디맨드 Windows 기본 요금: 0.019 USD 시간당

모든 세대

인스턴스 유형 비교

소프트웨어가 사전 설치된 AMI에는 추가 비용이 적용됩니다.

▼ 요약

인스턴스 개수 정보

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.2.2...read more
ami-01123b84e2a4fba05

Virtual server type (instance type)

t2.micro

Firewall (security group)

새 보안 그룹

Storage (volumes)

1 volume(s) - 8 GiB

프리 티어: 첫 해에는 월별 프리 티어 AMI에 대한 t2.micro(또는 t2.micro를 사용할 수 없는 리전의 t3.micro) 인스턴스 사용량 750시간, EBS 스토리지 30GiB, IO 2백만 개, 스냅샷 1GB, 인터넷 대역폭 100GB가 포함됩니다.

취소

인스턴스 시작

명령 검토

5. 키 페어 생성하기

- EC2 서버에 접속하기 위해 생성 해야되는 필수 파일 (접속 방법, 비밀번호가 저장된 파일 정도로 생각)
- 새 키페어 생성 클릭

11_AWS 웹 호스팅을 이용한 프로젝트 배포

4

▼ 키 페어(로그인) 정보

키 페어를 사용하여 인스턴스에 안전하게 연결할 수 있습니다. 인스턴스를 시작하기 전에 선택한 키 페어에 대한 액세스 권한이 있는지 확인하세요.

키 페어 이름 - 필수

선택

새 키 페어 생성

- 키 페어 생성

키 페어 이름은 자유롭게 작성 / RSA 선택 / .pem 선택

키 페어 생성



키 페어 이름

키 페어를 사용하면 인스턴스에 안전하게 연결할 수 있습니다.

my-server-key

이름에는 최대 255개의 ASCII 문자를 포함할 수 있습니다. 앞 또는 뒤에 공백을 포함할 수 없습니다.

키 페어 유형

☒ RSA

RSA 암호화된 프라이빗 및 퍼블릭 키 페어

☐ ED25519

ED25519 암호화된 프라이빗 및 퍼블릭 키 페어

프라이빗 키 파일 형식

☒ .pem

OpenSSH와 함께 사용

☐ .ppk

PuTTY와 함께 사용

메시지가 표시되면 프라이빗 키를 사용자 컴퓨터의 안전하고 액세스 가능한 위치에 저장합니다. 나중에 인스턴스에 연결할 때 필요합니다. [자세히 알아보기](#)

취소

키 페어 생성

💡 다운로드된 파일은 안전한 곳에 잃어 버리지 않게 보관!

한번 다운로드 후 재 다운로드 불가!!
(분실 시 기존 키 페어 폐기 후 새로 생성)

6. 네트워크 설정

- SSH 트래픽 접속 허용을 위한 IP 지정
 - 어디서든 생성된 인스턴스에 접속할 수 있게 하려면 : 위치무관(0.0.0.0/0)
 - 현재 사용중인 컴퓨터로만 생성된 인스턴스에 접속할 수 있게 하려면 : 내 IP
 - 다른 지정된 컴퓨터로만 생성된 인스턴스에 접속할 수 있게 하려면 하려면 : 사용자 지정
- SSH 트래픽 접속 설정 후 오른쪽 상단 **편집** 클릭

▼ 네트워크 설정 정보

편집

네트워크 정보

vpc-0cdb6fc94b79d2e5a

서브넷 정보

기본 설정 없음(가용 영역의 기본 서브넷)

퍼블릭 IP 자동 할당 정보

활성화

방화벽(보안 그룹) 정보

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

☒ 보안 그룹 생성

☐ 기존 보안 그룹 선택

Well create a new security group called launch-wizard-1 with the following rules:

☒ Allow SSH traffic from

Helps you connect to your instance

위치 무관
0.0.0.0/0

☐ 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.

×

7. 보안 그룹 편집

- 인바운드 규칙 설정
 - ssh / TCP / 22 / 위치 무관
 - 사용자 지정 TCP / TCP / 80 / 위치 무관

▼ 네트워크 설정 정보

VPC - 필수 정보

vpc-0cdb6fc94b79d2e5a (기본값) ▼

172.31.0.0/16

↻

서브넷 정보

기본 설정 없음 ▼

↻ 새 서브넷 생성 ↗

퍼블릭 IP 자동 할당 정보

활성화 ▼

방화벽(보안 그룹) 정보

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

☒ 보안 그룹 생성
 ☐ 기존 보안 그룹 선택

보안 그룹 이름 - 필수

launch-wizard-1

이 보안 그룹은 모든 네트워크 인터페이스에 추가됩니다. 보안 그룹을 만든 후에는 이름을 편집할 수 없습니다. 최대 길이는 255자입니다. 유효한 문자는 a~z, A~Z, 0~9, 공백 및 _-:/0#,@[]+=&{}!\$*입니다.

설명 - 필수 정보

launch-wizard-1 created 2023-11-20T02:54:55.612Z

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

🔍 CIDR, 접두사 목록 또는 보안 그룹

0.0.0.0/0 ✕

설명 - optional 정보

e.g. SSH for admin desktop

▼ Security group rule 2 (TCP, 80, 0.0.0.0/0)

제거

유형 정보

사용자 지정 TCP ▼

프로토콜 정보

TCP

포트 범위 정보

80

소스 유형 정보

위치 무관 ▼

원본 정보

🔍 CIDR, 접두사 목록 또는 보안 그룹

0.0.0.0/0 ✕

설명 - optional 정보

e.g. SSH for admin desktop

보안 그룹 규칙 추가

8. 스토리지 구성

- 기본 8Gib / gp3 사용

11_AWS 웹 호스팅을 이용한 프로젝트 배포

7

▼ 스토리지 구성

정보

Advanced

1x

8

GiB

gp3

▼

루트 볼륨 (암호화되지 않음)

📘

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

✕

새 볼륨 추가

🕒

백업 정보를 보려면 새로 고침 클릭

↻

The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x 파일 시스템

편집

9. 설정 완료 후 인스턴스 생성

취소

인스턴스 시작

명령 검토

10. 좌측 메뉴 탭 → 인스턴스 클릭 → 생성된 인스턴스 확인

인스턴스 (1/1) 정보

🔄

연결

인스턴스 상태 ▼

작업 ▼

인스턴스 시작 ▼

🔍 Instance를 속성 또는 (case-sensitive) 태그로 찾기

< 1 > ⚙️

<input checked="" type="checkbox"/>	Name ↗	인스턴스 ID	인스턴스 상태 ▼	인스턴스 유형 ▼	상태 검사	정보 상태	가용 영역 ▼	퍼블릭 IPv4 DNS ▼	퍼블릭 IPv4 ... ▼	탄력적 IP	IPv6 IP ▼	모니터링
<input checked="" type="checkbox"/>	my server	i-0ed346f323cc49d64	실행 중	t2.micro	초기화	경보 없음 +	ap-northeast-2c	ec2-13-125-159-211.ap...	13.125.159.211	-	-	disab

인스턴스: i-0ed346f323cc49d64(my server)

⚙️ ✕

세부 정보

보안

네트워킹

스토리지

상태 검사

모니터링

태그

▼ 인스턴스 요약 정보

인스턴스 ID

i-0ed346f323cc49d64 (my server)

IPv6 주소

-

호스트 이름 유형

IP 이름: ip-172-31-35-181.ap-northeast-2.compute.internal

프라이빗 리소스 DNS 이름 응답

IPv4(A)

자동 할당된 IP 주소

13.125.159.211 [퍼블릭 IP]

IAM 역할

-

IMDSv2

Required

▼ 인스턴스 세부 정보 정보

플랫폼

퍼블릭 IPv4 주소

13.125.159.211 |개방 주소법

인스턴스 상태

실행 중

프라이빗 IP DNS 이름(IPv4만 해당)

ip-172-31-35-181.ap-northeast-2.compute.internal

인스턴스 유형

t2.micro

VPC ID

vpc-0cdb6fc94b79d2e5a

서브넷 ID

subnet-052be317a2b8b76ee

프라이빗 IPv4 주소

172.31.35.181

퍼블릭 IPv4 DNS

ec2-13-125-159-211.ap-northeast-2.compute.amazonaws.com |개방 주소법

탄력적 IP 주소

-

AWS Compute Optimizer 찾기

권장 사항을 위해 AWS Compute Optimizer에 옵트인합니다. | 자세히 알아보기

Auto Scaling 그룹 이름

-

AMI ID

모니터링

★ 탄력적 IP 설정(Elastic IP)

- AWS EC2의 인스턴스는 인스턴스가 중지 → 실행 될 때 마다 외부 접근이 가능한 public IP가 변하는 특징이 있음(보안상의 이유)
- 탄력적 IP : 외부에서 접근 가능한 별도의 고정된 IP



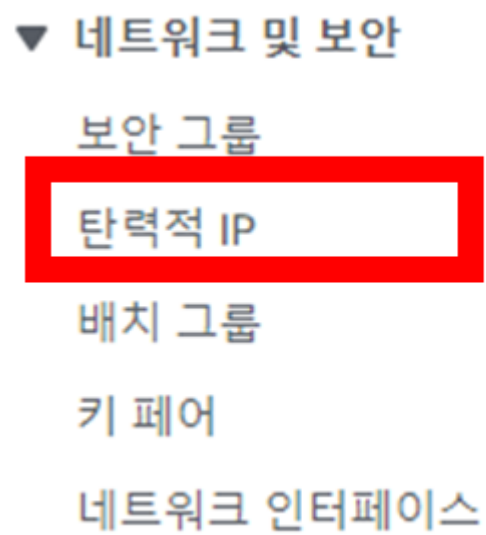
탄력적 IP는 생성 후 미 사용 시 요금이 부과되기 때문에
필요한 만큼만 생성한 후 무조건 사용!

**⚠ 탄력적 IP 요금 정책 변경으로 24년 2월 1일 부터
사용 중인 탄력적 IP도 시간당 0.005 USD 추가되도록
변경되었습니다**

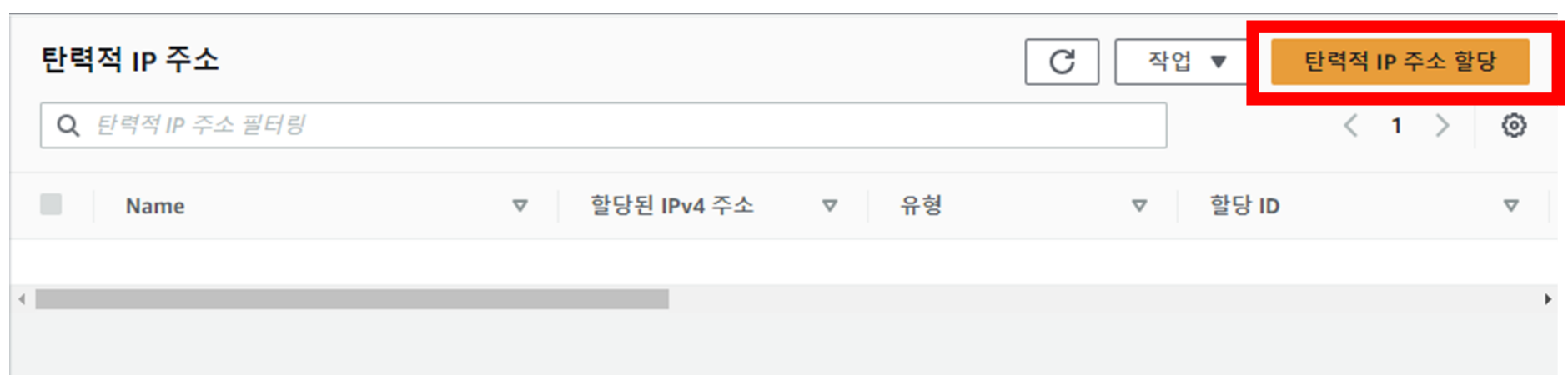
(30일 기준 3.6 USD, 한화 4,500~4,700 원 예상)

<https://aws.amazon.com/ko/blogs/korea/new-aws-public-ipv4-address-charge-public-ip-insights/>

1. EC2 대시보드 → 좌측 메뉴 탭 → 네트워크 및 보안 → 탄력적 IP



2. 탄력적 IP 주소 할당 클릭



3. 할당 클릭

탄력적 IP 주소 할당

정보

탄력적 IP 주소 설정

정보

네트워크 경계 그룹

정보

Q ap-northeast-2

X

퍼블릭 IPv4 주소 풀

Amazon의 IPv4 주소 풀

BYOIP 방식으로 AWS 계정에 가져오는 퍼블릭 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

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.

리소스에 연결된 태그가 없습니다.

새로운 태그 추가

최대 50개의 태그를 더 추가할 수 있습니다.

취소

할당

4. 탄력적 IP 인스턴스와 연결

- 생성된 탄력적 IP 선택 후 우측 상단 작업 → 탄력적 IP 주소 연결 클릭

탄력적 IP 주소 (1/1)

Q 탄력적 IP 주소 필터링

퍼블릭 IPv4 주소: 3.39.144.26

X

필터 지우기

작업

탄력적 IP 주소 할당

세부 정보 보기

탄력적 IP 주소 릴리스

탄력적 IP 주소 연결

탄력적 IP 주소 연결 해제

역방향 DNS 업데이트

전송 활성화

전송 비활성화

전송 수락

DNS 레코드

<input checked="" type="checkbox"/>	Name	할당된 IPv4 주소	유형	할당 ID
<input checked="" type="checkbox"/>	-	3.39.144.26	퍼블릭 IP	eipalloc-0e368

- 인스턴스, 프라이빗 IP 주소 선택 후 연결 클릭

11_AWS 웹 호스팅을 이용한 프로젝트 배포

10

탄력적 IP 주소 연결 정보

이 탄력적 IP 주소에 연결할 인스턴스 또는 네트워크 인터페이스를 선택합니다. (3.39.144.26)

탄력적 IP 주소: 3.39.144.26

리소스 유형

탄력적 IP 주소를 연결할 리소스의 유형을 선택합니다.

☒ 인스턴스
☐ 네트워크 인터페이스

⚠ 탄력적 IP 주소가 이미 연결된 인스턴스와 탄력적 IP 주소를 연결하면 이전에 연결된 탄력적 IP 주소가 연결 해제되지만 주소는 여전히 계정에 할당됩니다. 자세히 알아보기

프라이빗 IP 주소를 지정하지 않으면 탄력적 IP 주소가 기본 프라이빗 IP 주소와 연결됩니다.

인스턴스

프라이빗 IP 주소

탄력적 IP 주소를 연결할 프라이빗 IP 주소입니다.

재연결

이미 리소스에 연결되어 있는 탄력적 IP 주소를 다른 리소스에 재연결할 수 있는지 여부를 지정합니다.

☐ 이 탄력적 IP 주소를 재연결하도록 허용

취소

연결

5. 좌측 메뉴 탭 → 인스턴스 → 생성한 인스턴스 우클릭 → 재부팅 후 퍼블릭 IPv4와 탄력적 IP 주소가 같은지 확인

- 인스턴스 재시작

- 퍼블릭 IP, 탄력적 IP가 같은지 확인

11_AWS 웹 호스팅을 이용한 프로젝트 배포

11

퍼블릭 IPv4 주소

3.39.144.26 | [개방 주소법](#)

인스턴스 상태

실행 중

프라이빗 IP DNS 이름(IPv4만 해당)

ip-172-31-35-181.ap-northeast-2.compute.internal

인스턴스 유형

t2.micro

프라이빗 IPv4 주소

172.31.35.181

퍼블릭 IPv4 DNS

ec2-3-39-144-26.ap-northeast-2.compute.amazonaws.com | [개방 주소법](#)

탄력적 IP 주소

3.39.144.26 [퍼블릭 IP]

EC2 인스턴스 연결(접속) 및 JDK, git 설치

- 생성된 인스턴스에 연결하여 명령어를 입력하여 JDK, git 설치
- git을 이용해 github에 업로드된 프로젝트를 clone
- gradle을 이용해서 프로젝트를 build
- jdk를 이용해 build된 프로젝트 jar 파일을 실행

1. 인스턴스 선택 후 연결 클릭 → 다음 화면에서 바로 연결 클릭 → 커맨드 입력 화면

인스턴스 (1/1) 정보

🔄

연결

인스턴스 상태 ▼

작업 ▼

인스턴스 시작 ▼

☑

Name

▼

☑

my server

✎

i-0ed346f323cc49d64

실행 중

t2.micro

2/2개 검사 통과...

경보 없음

ap-northeast-2c

ec2-3-39-144-

정보

다음 옵션 중 하나를 사용하여 인스턴스 i-0ed346f323cc49d64 (my server)에 연결

연결

```

#
~\#### Amazon Linux 2023
~~\#####\
~~\###|
~~\#/ https://aws.amazon.com/linux/amazon-linux-2023
~~v~'~>
~~~
~~~
~~~
~/m/'~>
Last login: Mon Nov 20 05:30:52 2023 from 13.209.1.61
[ec2-user@ip-172-31-35-181 ~]$

```

처음 접속 시 /home/ec2-user/ 폴더에 위치해있음!

2. JDK 설치

```
sudo yum install java-17-amazon-corretto
```

(설치 완료 후 `java -version` 으로 설치 확인)

3. git 설치

```
sudo yum install git
```

(설치 완료 후 `git --version`으로 설치 확인)

4. github에 업로드된 프로젝트 clone 하기

```
git clone [클론주소]
```

(클론 주소는 github 페이지에서 복사)



클론한 프로젝트에 변경 사항 내용이 있을 경우 pull 받는 방법!

`git pull origin`

✨config.properties EC2 서버에 생성 후 clone한 프로젝트에 추가하기



수업 때 사용한 프로젝트에는 계정 정보, 주소, 경로가 작성된 config.properties가 누락되어 있기 때문에 프로젝트 build 전에 추가해주는 작업이 필요하다!

1) 기존 config.properties 에서 경로 관련된 내용을 수정

```
--
40 # resource path : 자원이 저장된 프로젝트 외부 폴더 경로 (window)
41 #회원(프로필이미지) 저장 경로
42 #게시판 이미지 저장 경로
43 # 파일의 임시 저장 경로(메모리 임계값 초과 시 임시 저장할 폴더)
44 # my.images.resourcepath=file:///C:/workspace/project-resources/images/
45 # my.member.location=C:/workspace/project-resources/images/member/
46 # my.board.location=C:/workspace/project-resources/images/board/
47 # spring.servlet.multipart.location=C:/workspace/project-resources/temp/
48
49
50 # resource path : 자원이 저장된 프로젝트 외부 폴더 경로 (linux)
51 my.images.resourcepath=file:///home/ec2-user/project-resources/images/
52 my.member.location=/home/ec2-user/project-resources/images/member/
53 my.board.location=/home/ec2-user/project-resources/images/board/
54 spring.servlet.multipart.location=/home/ec2-user/project-resources/temp/
```

2) EC2 서버에 config.properties 파일 생성

[/home/ec2-user 폴더에서 진행]

vim config.properties

또는

vi config.properties

3) "i" 키를 눌러 INSERT 모드로 변경 (왼쪽 하단에서 확인 가능)



4) 기존 config.properties 파일의 내용을 복사한 후
EC2 서버 vi 또는 vim 화면에 붙여넣기

! ctrl + v 단축키가 적용되지 않음 (우클릭 → 붙여넣기 또는 ctrl + shift + insert)

5) esc를 눌러 모드 선택으로 이동 → :wq 입력 후 엔터

(: wq → 저장 후 종료 (write, quit) 를 의미하는 vi, vim 명령어)

```
# HTTP 요청당 최대 크기
# 기본값: 10MB
# 52,428,800 = 50MB
spring.servlet.multipart.max-request-size = 52428800
:wq
```

6) EC2 서버에 생성된 config.properties 파일을 clone한 프로젝트 폴더로 복사

cp config.properties [클론한프로젝트경로]/src/main/resources/config.properties

(cp → 복사(copy)를 의미하는 리눅스 명령어)

7) 복사한 위치로 이동해서 복사 확인

```
cd [클론한프로젝트경로]/src/main/resources/  
ls
```

(cd → 디렉토리 변경(change directory)를 의미하는 리눅스 명령어)

(ls → 디렉토리 목록(list) 보기를 의미하는 리눅스 명령어)

```
[ec2-user@ip-172-31-35-181 ~]$ cd copy/boardProject-boot/src/main/resources/  
[ec2-user@ip-172-31-35-181 resources]$ ls  
application.properties  'config.properties 템플릿'  messages.properties  static  
config.properties      mappers                     mybatis-config.xml   templates  
[ec2-user@ip-172-31-35-181 resources]$
```

5. gradle을 이용해서 프로젝트 build 및 배포하기

- 프로젝트 최상위 폴더로 이동

```
cd /home/ec2-user/[프로젝트 폴더]
```

- gradlew 파일에 권한 부여

```
sudo chmod 777 ./gradlew
```

- 프로젝트 빌드

```
sudo ./gradlew clean build
```

- 프로젝트 배포

```
sudo java -jar build/libs/[빌드된 .jar파일].jar
```

- 퍼블릭 IP 주소를 브라우저에 작성하면 배포된 프로젝트 이용 가능
- 서버 종료는 ctrl + c