

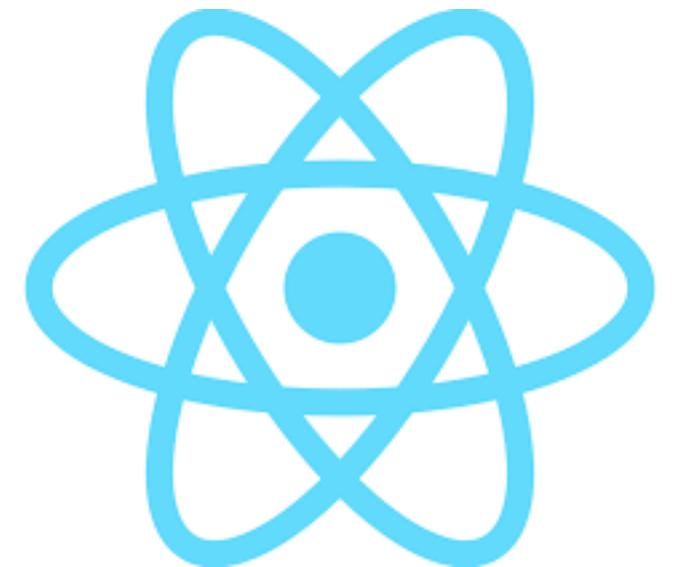
알아두면 쓸데있는 개발 상식

창업 ver

with 박지환

개발

프론트엔드

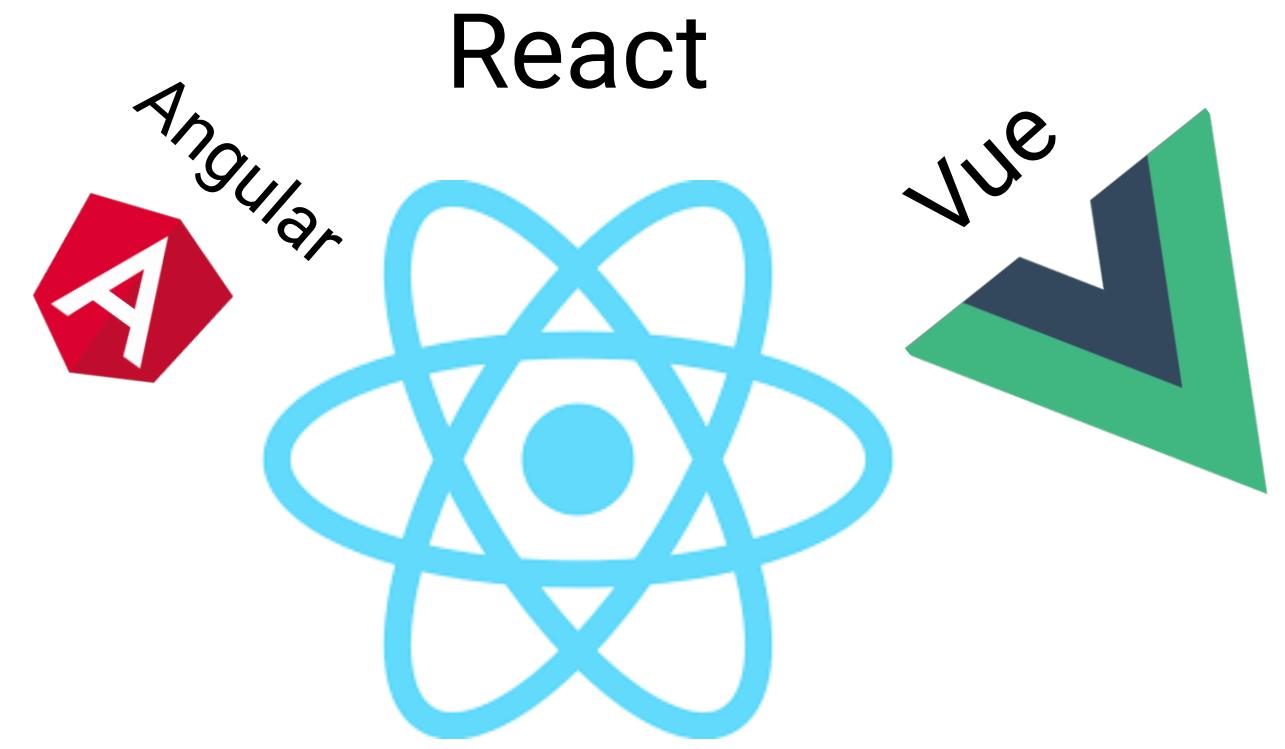


백엔드



알쓸창개상

개발(프론트엔드)



화면을 쉽게 조작할 수 있도록 도와줌

사용자 인터페이스를 만들기 위한 JavaScript 라이브러리 또는 프레임워크

앱처럼 부드러운 UX 제공

개발(프론트엔드)

화면 조작이 쉽다?
=> DOM 조작이 쉽다

Ex) 버튼 클릭시 숫자 올라가는 웹앱 만들기

0 1

개발(프론트엔드)

순수 Javascript

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <div id="root">
    <p id="display-number">0</p>
    <input id="up-button" type="button" value="올리기" />
    <script src=".//index.js"></script>
  </div>
</body>
</html>
```

index.html

```
const displayNumber = document.getElementById('display-number')

document.getElementById('up-button').addEventListener('click', function(){
  displayNumber.textContent = ((displayNumber.textContent * 1) + 1 + '')
})
```

index.js

일일이 요소를 찾고 eventListner 등록하고 값 바꿔서 넣고..



개발(프론트엔드)

with React

```
import './App.css';
import {useState} from 'react'

function App() {
  const [count, setCount] = useState(0)
  const onClickHandler = () => {
    setCount(count + 1);
  }
  return (
    <div className="App">
      <p>{count}</p>
      <input type="button" value="올리기" onClick={onClickHandler}></input>
    </div>
  );
}

export default App;
```

선언적이다

- 어떻게 동작하라가 아니라 이렇게 돼라!

개발자는 편안..



프론트엔드 고려 사항

브라우저 호환성

Chrome, Safari, Opera, ... <https://caniuse.com/>

Babel, Webpack 사용

렌더링 최적화

JS 코드가 많아지는데 어떻게 빨리 화면을 보여줘야 할까?

보안

XSS
(Cross-site-Scripting)

추천

1. 서비스가 보여지는 부분이 중요한지 판단

요즘은 대부분이 그러하다..

2. 타겟 고객층을 고려해서 서비스 환경 고려

기본은 Chrome, Safari

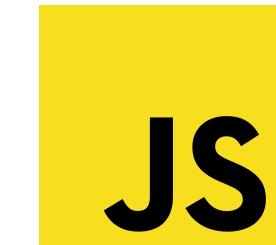
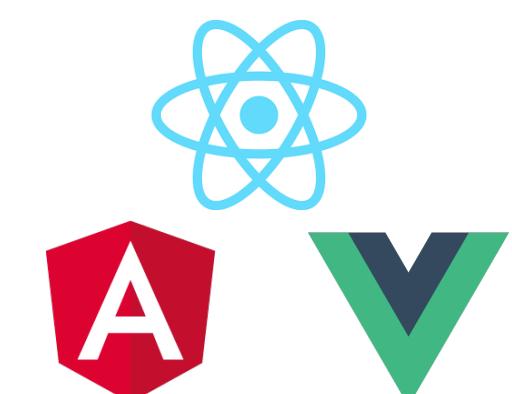
5, 60대가 많다면?

보안이 중요한가?

IE, Samsung Internet 고려

렌더링이 빨라야하는가?

3. 서비스에 맞는 스택 선택



개발(백엔드)

백엔드

- 서비스의 핵심 로직 담당
 - DB 조회, 변경,
 - 프론트엔드에게 정보 제공
 - 서버 관리
-

성능 높이기



Debug=False



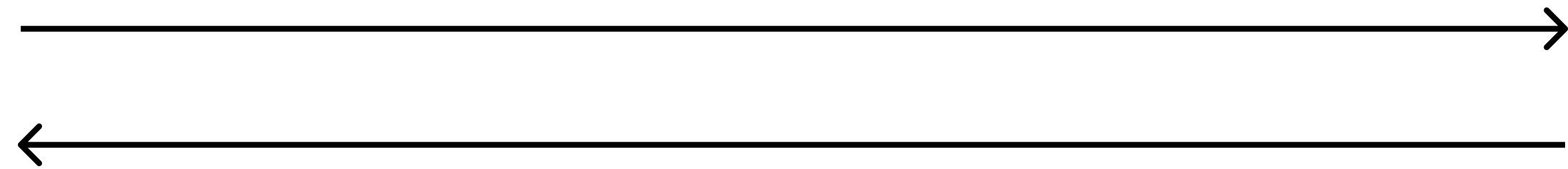
NODE_ENV=production

개발(배포)

1단계



Request



Response

HTML, CSS, JS, Img, ...



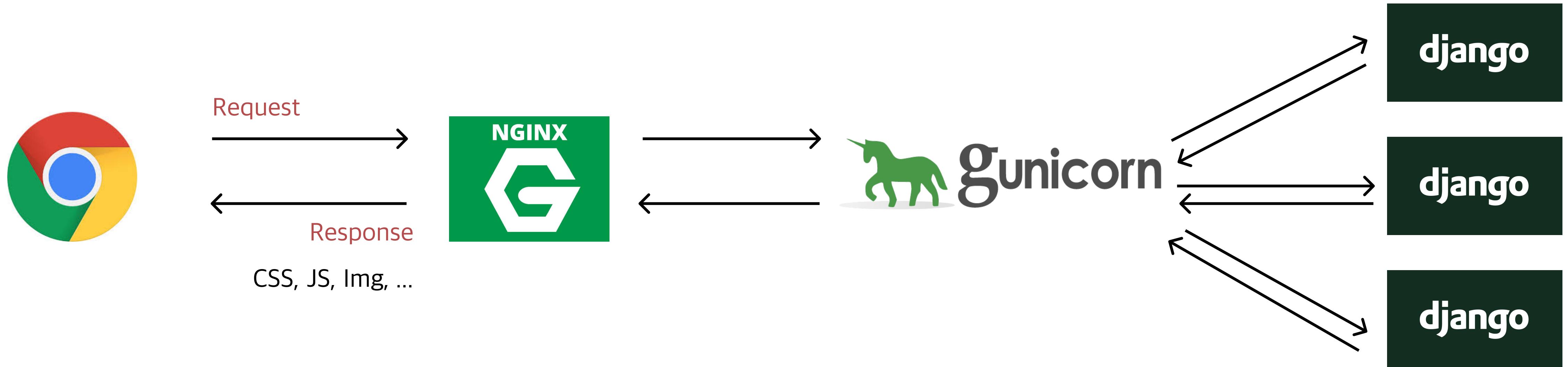
python manage.py runserver

개발(배포)

2단계

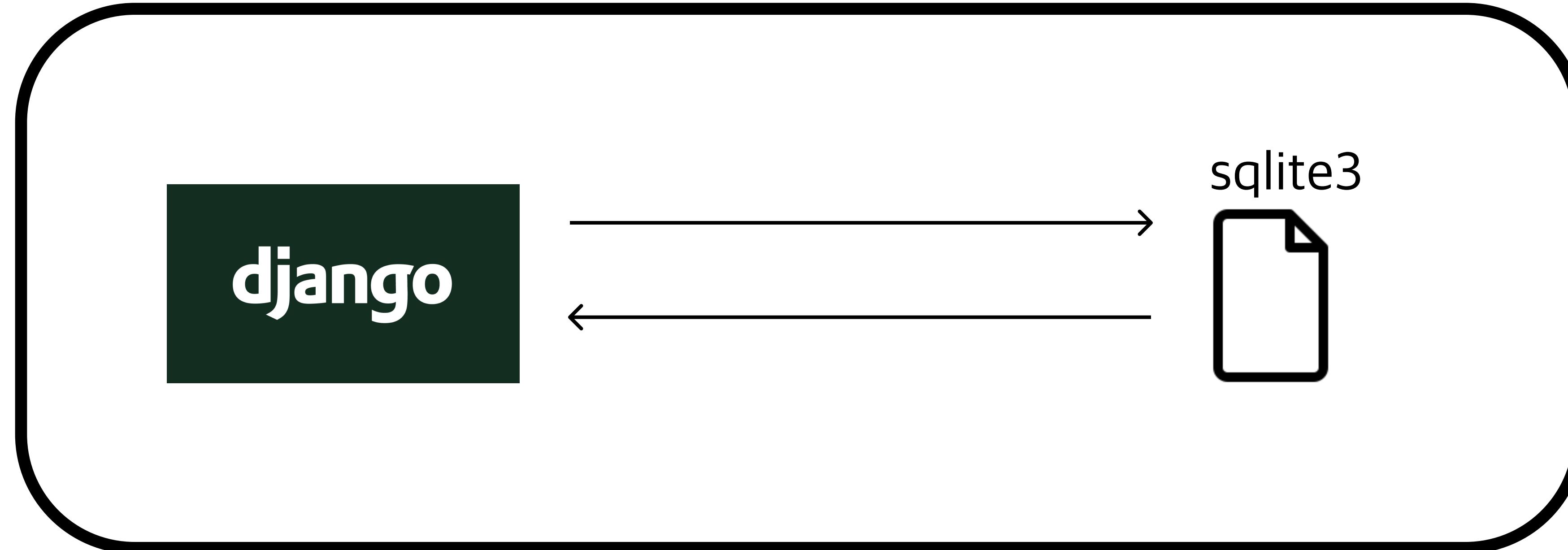


3단계(Production level)



DB 연결(1단계)

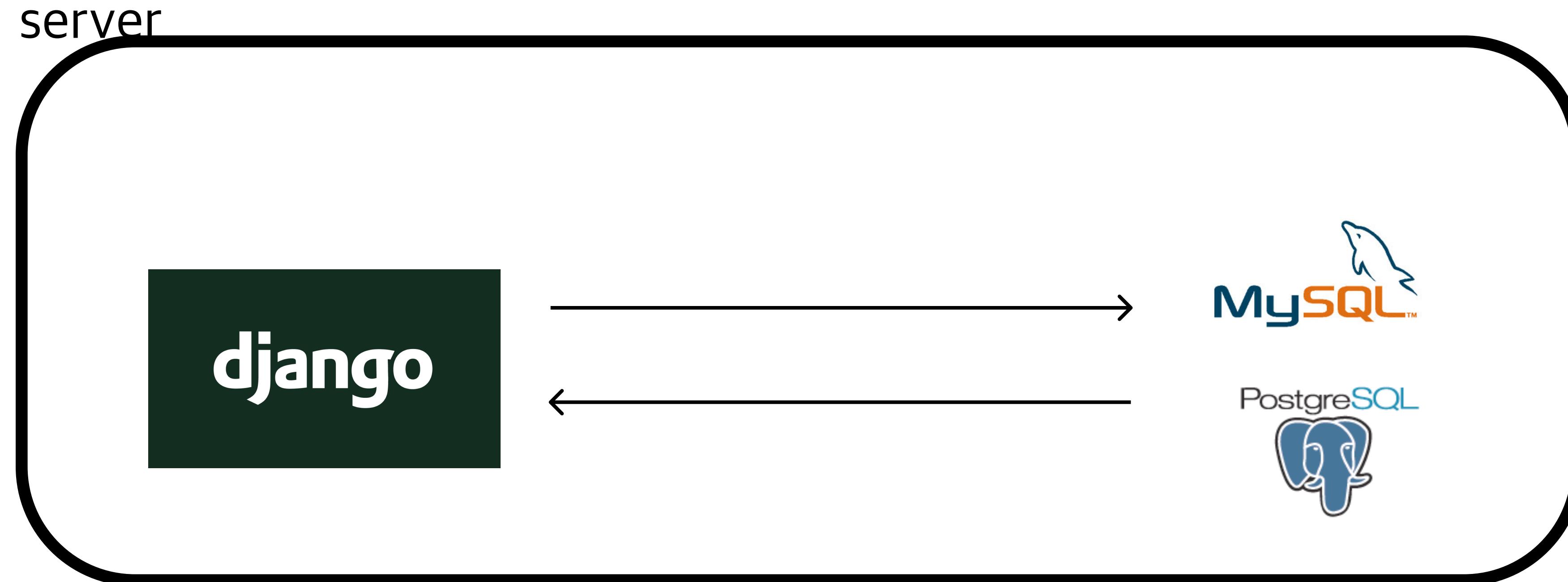
server



sqlite3

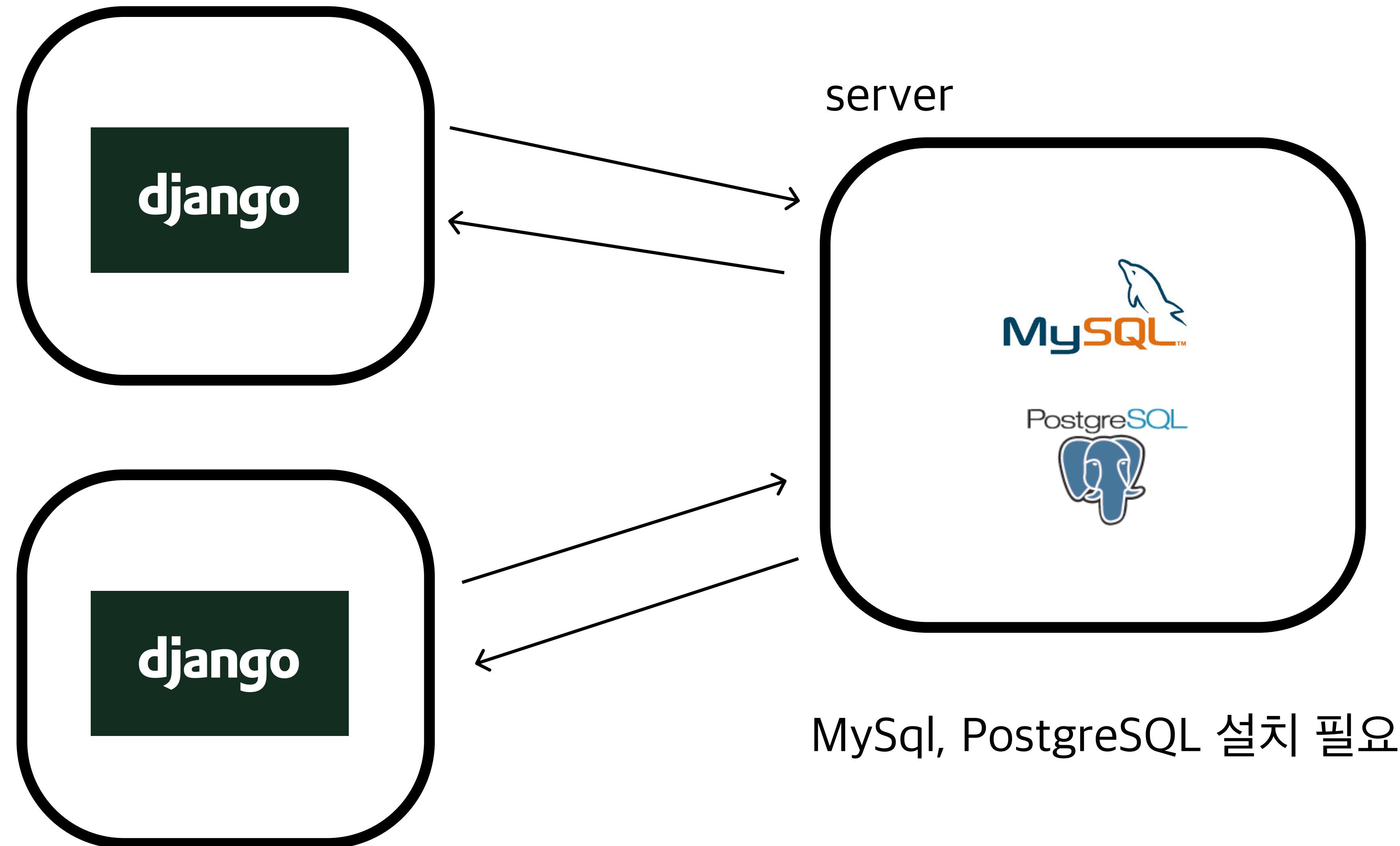
파일 기반의 데이터베이스 관리 시스템

DB 연결(2단계)



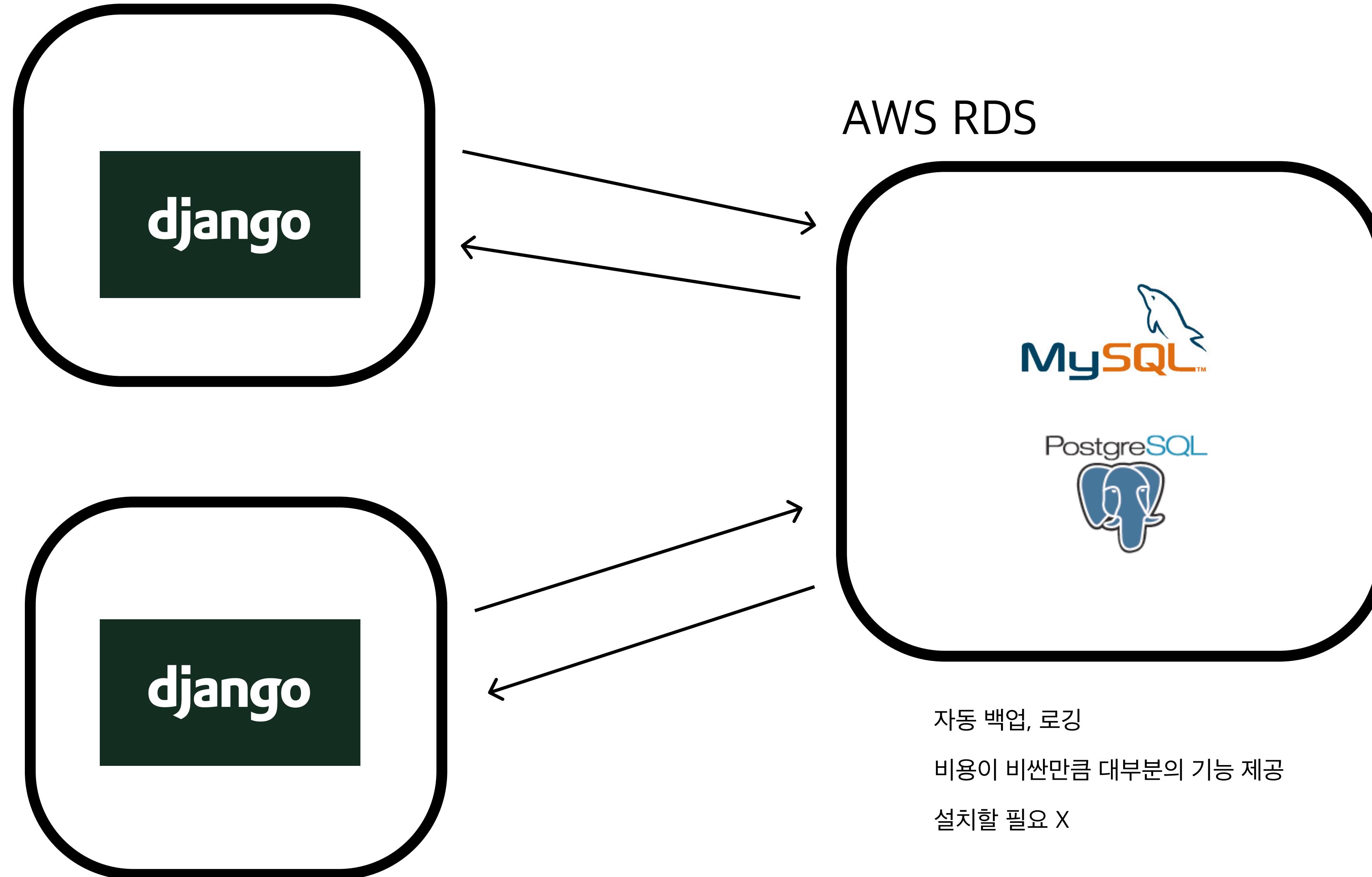
MySQL, PostgreSQL 설치 필요

DB 연결(3단계)



개발(배포)

DB 연결(3단계)



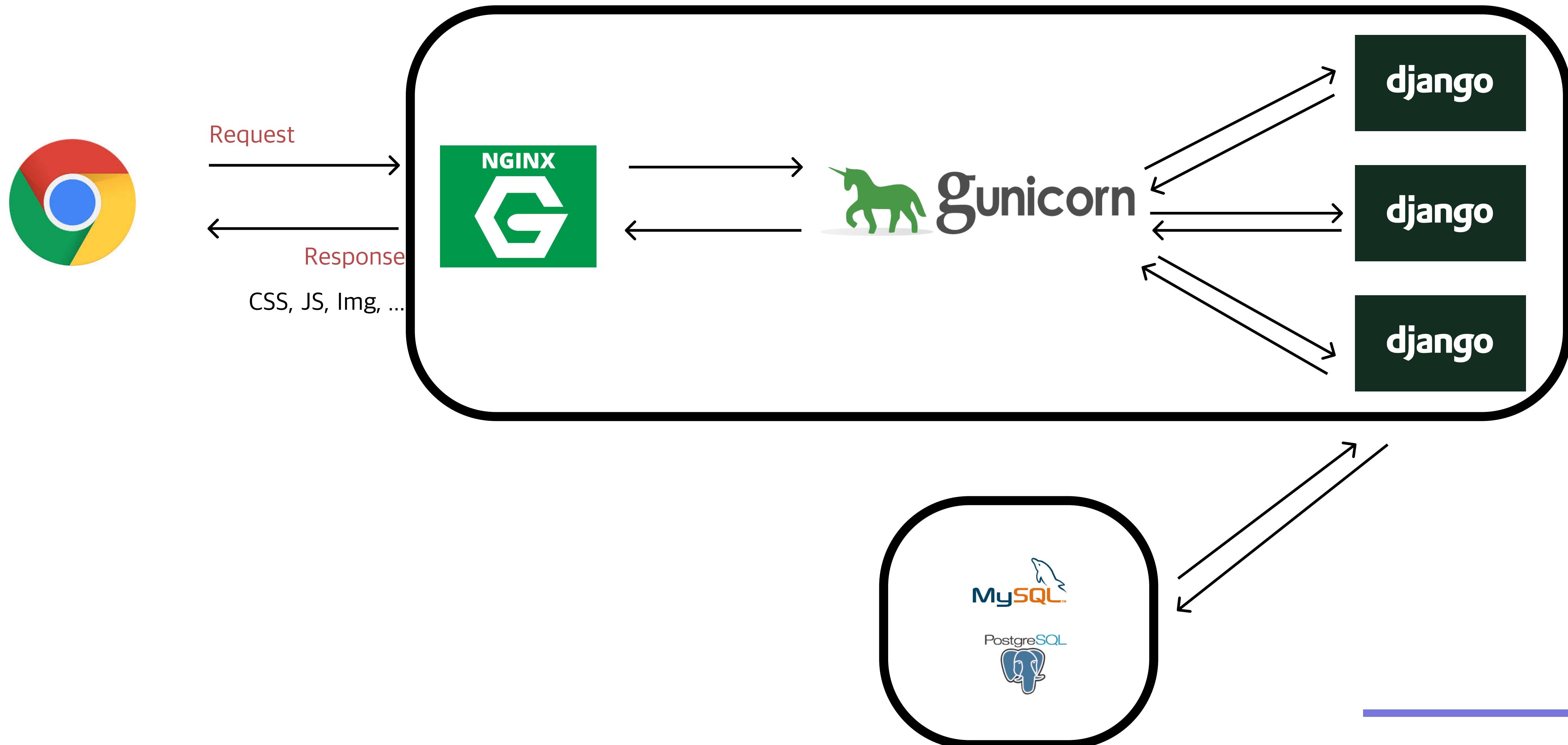
배포 환경에 맞게 settings.py 설정

```
 DATABASES = {  
     'default': {  
         'ENGINE': os.environ['DATABASE_ENGINE'],  
         'NAME': os.environ['DATABASE_NAME'],  
         'USER': os.environ['DATABASE_USER'],  
         'PASSWORD': os.environ['DATABASE_PASSWORD'],  
         'HOST': os.environ['DATABASE_HOST'],  
         'PORT': os.environ['DATABASE_PORT'],  
     }  
}
```

알쓸창개발

개발(배포)

이상적



추천

1. 서비스의 트래픽 예상

트래픽에 맞는 배포 환경 구성

2. 서비스의 확장 가능성 고려

트랙픽이 늘어남에 따라 서비스를 확장할 수 있는 구조인가?

3. 서비스의 긴급도 고려

바로 해야하는 서비스인가? 시간이 여유로운 서비스인가?

4. 비용 고려



개발(배포)

백엔드 고려 사항

속도, 성능

얼마나 빠르게, 많은 요청을 처리할 수 있는가?

배포

어떤 배포 환경이 서비스에 적합한가?

보안

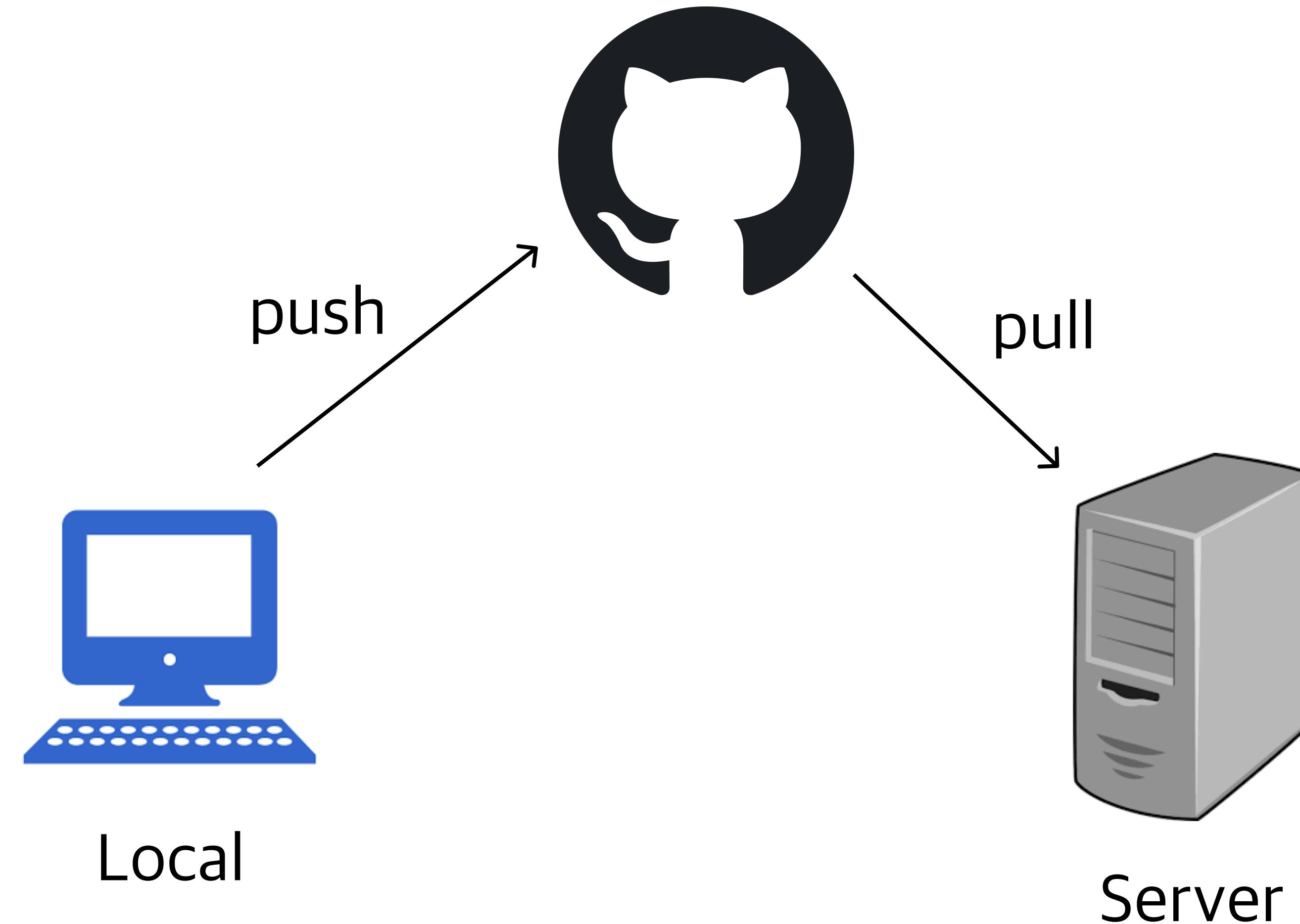
SQL Injection

DOS attack

CSRF

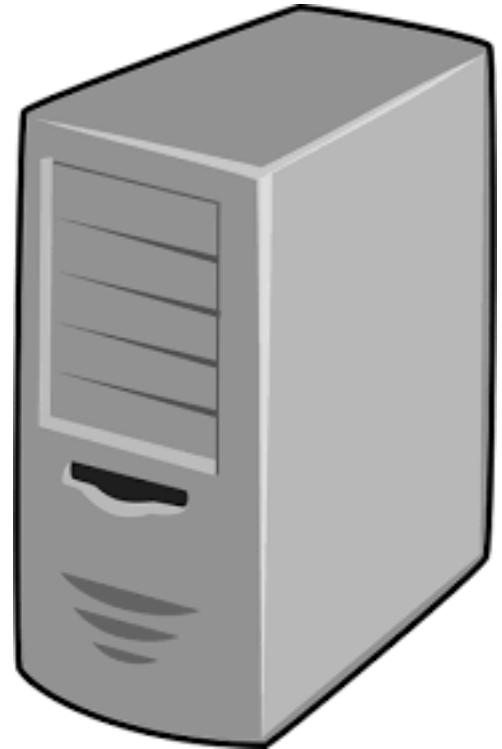
개발(배포)

배포 맛보기



개발(배포)

배포 맛보기

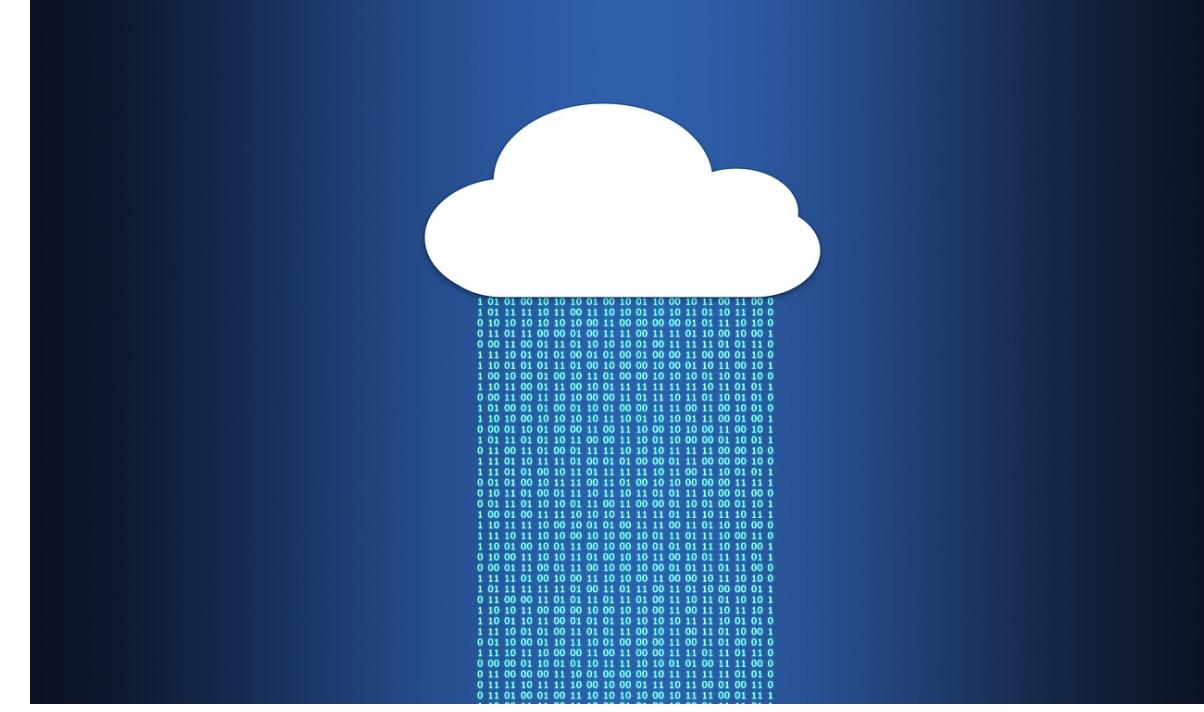


개비쌈;



서버 컴퓨터를 안사고 배포할 순 없을까?





cloud computing



EC2, S3, RDS...



Google Cloud

언제든, 필요한만큼 자원을 사용할 수 있다.

서버 관리를 일일이 하지 않아도 된다.(보안 등)

스타트업에 적합



클라우드 컴퓨팅 서비스 2004
김민철 저작



서비스 운영이 쉬워지는 **AWS 인프라 구축 가이드**

서버 구축부터 배포, 모니터링, 관리 자동화, 데브옵스까지



//

유카나

(추천) EC2, S3, RDS, AMI

(심화) CloudWatch, Route53, IAM

고려대학교 도서관

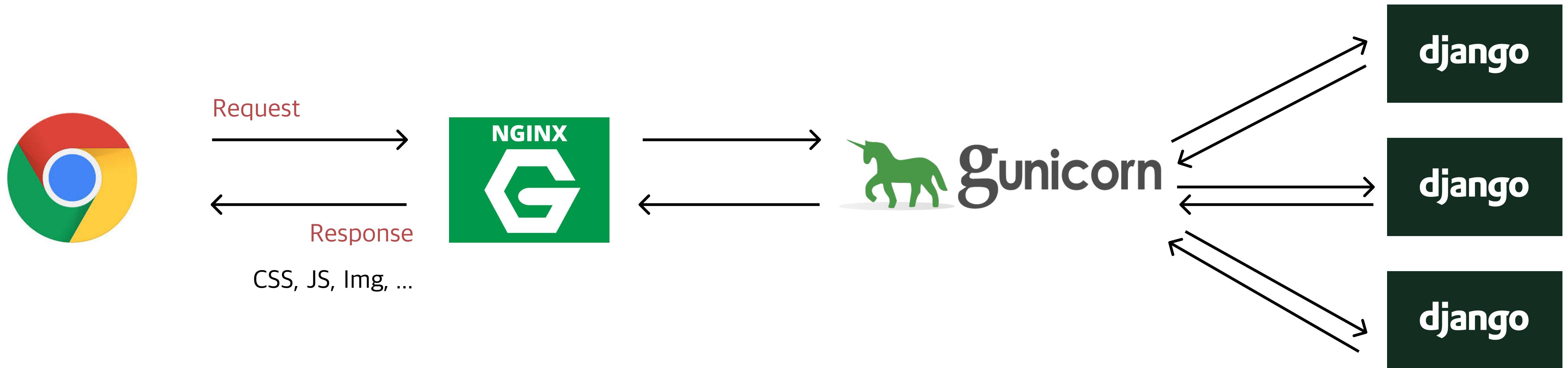
EC2

사용자가 가상 컴퓨터를 임대 받아 그 위에 자신만의 컴퓨터 애플리케이션들을 실행할 수 있게 하는 서비스

사용자는 필요하면 서버 인스턴스를 만들고 시작하고 종료할 수 있으며, 실행 중인 서버에 대해 시간당 지불

사용자는 필요하면 서버 인스턴스를 만들고 시작하고 종료할 수 있으며, 실행 중인 서버에 대해 시간 당 지불

EC2로 배포해보기



EC2 만들기

서울 지역 선택 확인..!

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

단계 1: Amazon Machine Image(AMI) 선택

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

검색어를 입력하여 AMI를 검색합니다. 예: 'Windows'

Systems Manager 파라미터로 검색

빠른 시작

	선택	64비트(x86)	64비트(Arm)
나의 AMI	Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0b827f3319f7447c6 (64비트 x86) / ami-065bbf792e2f70fd9 (64비트 Arm)	<input checked="" type="radio"/>	<input type="radio"/>
AWS Marketplace	Amazon Linux 2는 5년간 지원을 제공합니다. Amazon EC2에 성능 최적화된 Linux kernel 4.14와 systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, 최신 소프트웨어 패키지를 추가적으로 제공합니다.	<input checked="" type="radio"/>	<input type="radio"/>
커뮤니티 AMI	루트 디바이스 유형: ebs 가상화 유형: hvm ENA 활성화: 예		
<input type="checkbox"/> 프리 티어만 ⓘ			
Amazon Linux	Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-04876f29fd3a5e8ba (64비트 x86) / ami-0815e517ed50fd3f4 (64비트 Arm)	<input checked="" type="radio"/>	<input type="radio"/>
	Ubuntu Server 20.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).		
	루트 디바이스 유형: ebs 가상화 유형: hvm ENA 활성화: 예		
Ubuntu Server	Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0ba5cd124d7a79612 (64비트 x86) / ami-08b051fc14e6c551e (64비트 Arm)	<input checked="" type="radio"/>	<input type="radio"/>
	Ubuntu Server 18.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).		
	루트 디바이스 유형: ebs 가상화 유형: hvm ENA 활성화: 예		
SUSE Linux	SUSE Linux Enterprise Server 15 SP2 (HVM), SSD Volume Type - ami-097fc5cd098dd20d5 (64비트 x86) / ami-06fb0a38fed3dcdec (64비트 Arm)	<input checked="" type="radio"/>	<input type="radio"/>
	SUSE Linux Enterprise Server 15 SP2 (HVM), EBS General Purpose (SSD) Volume Type. Support available from SUSE (http://www.suse.com).		
	루트 디바이스 유형: ebs 가상화 유형: hvm ENA 활성화: 예		

Ubuntu 20.04 선택

EC2 만들기

인스턴스 유형
t2.micro 선택

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 <small>i</small>	볼륨 <small>i</small>	네트워크 인터페이스 <small>i</small>
Name	ec2-deploy-test	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

다른 태그 추가 (최대 50개 태그)

태그 습관화!

EC2 만들기



모든 IP 주소에서 인스턴스에 액세스하도록 허용

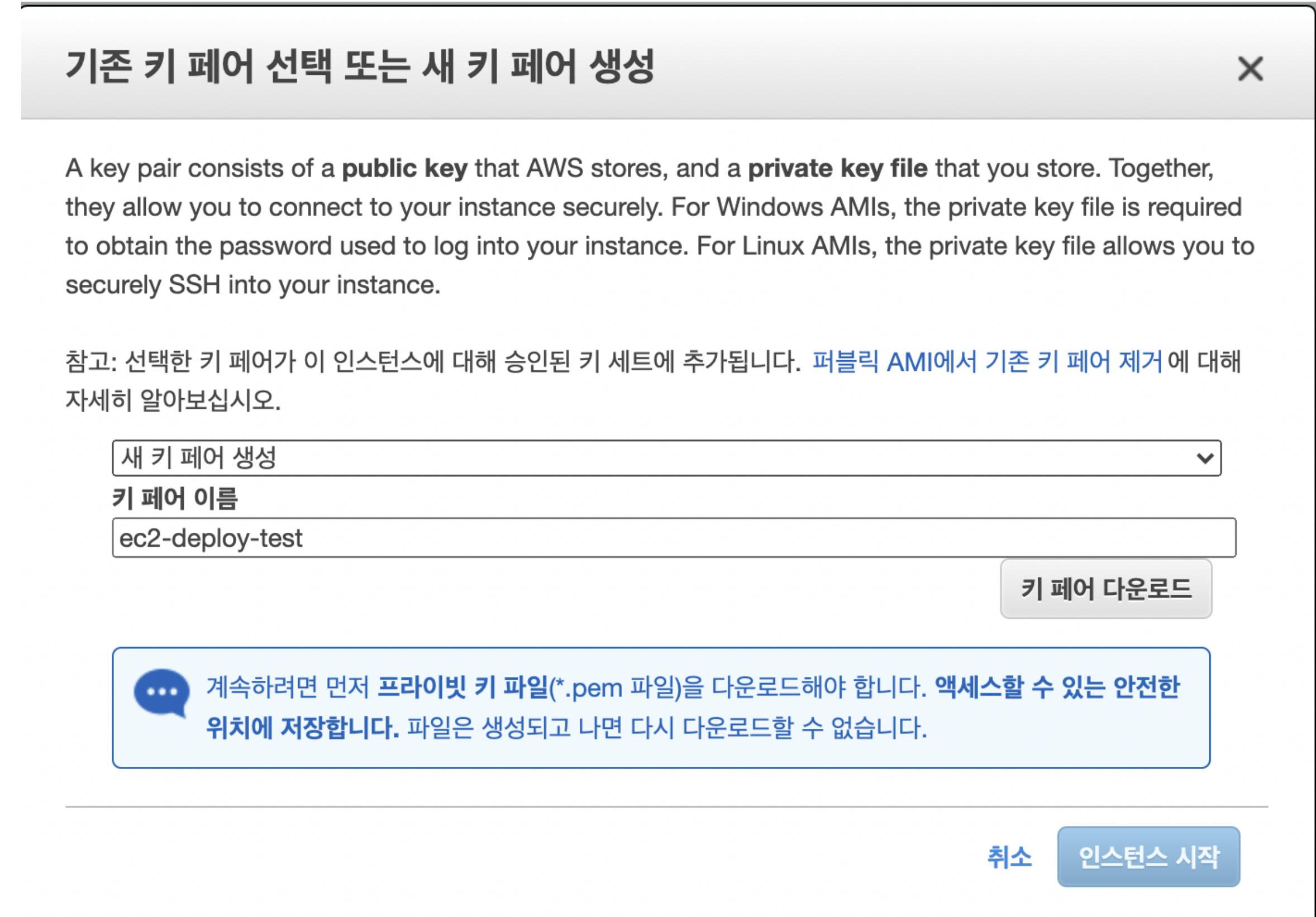
SSH(22번 포트), HTTP(80번 포트)에 대해 허용(서비스에 맞게)

실제 Production에서는 HTTPS를 쓰는게 좋다

실제 Production에서는 IP를 제한하는 것이 좋다

SSH -> 인스턴스에 들어갈 수 있는 컴퓨터(ex) 로컬 컴퓨터

EC2 만들기



ssh로 인스턴스에 들어갈 때 필요한 키 페어 생성
다운로드 후 기억할만한 곳으로 이동

EC2 만들기

인스턴스 (1/1) 정보

C 연결 인스턴스 상태 ▾ 작업 ▾ 인스턴스 시작 ▾

< 1 > | 설정

인스턴스 필터링

Name	인스턴스 ID	인스턴스 상태	인스턴스 유형	상태 검사	경보 상태	가용 영역
<input checked="" type="checkbox"/> ec2-deploy-test	i-0cefaf55039937b65	<input checked="" type="radio"/> 실행 중	<input checked="" type="radio"/> t2.micro	초기화	경보 없음	+ ap-northeast-2a

인스턴스: i-0cefaf55039937b65(ec2-deploy-test)

EC2 만들기

인스턴스에 연결 정보

다음 옵션 중 하나를 사용하여 인스턴스 i-0cefaf55039937b65 (ec2-deploy-test)에 연결

[EC2 인스턴스 연결](#) | [Session Manager](#) | **SSH 클라이언트**

인스턴스 ID
[i-0cefaf55039937b65](#) (ec2-deploy-test)

1. SSH 클라이언트를 엽니다.
2. 프라이빗 키 파일을 찾습니다. 이 인스턴스를 시작하는 데 사용되는 키는 ec2-deploy-test.pem입니다.
3. 필요한 경우 이 명령을 실행하여 키를 공개적으로 볼 수 있도록 합니다.
[chmod 400 ec2-deploy-test.pem](#)
4. 퍼블릭 DNS을(를) 사용하여 인스턴스에 연결:
[ec2-3-34-96-191.ap-northeast-2.compute.amazonaws.com](#)

예:
[ssh -i "ec2-deploy-test.pem" ubuntu@ec2-3-34-96-191.ap-northeast-2.compute.amazonaws.com](#)

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

chmod 400 ec2-deploy-test.pem

ssh -i "ec2-deploy-test.pem" ubuntu@ec2-3-34-96-191.ap-northeast-2.compute.amazonaws.com

ssh로 해당 인스턴스에 접속

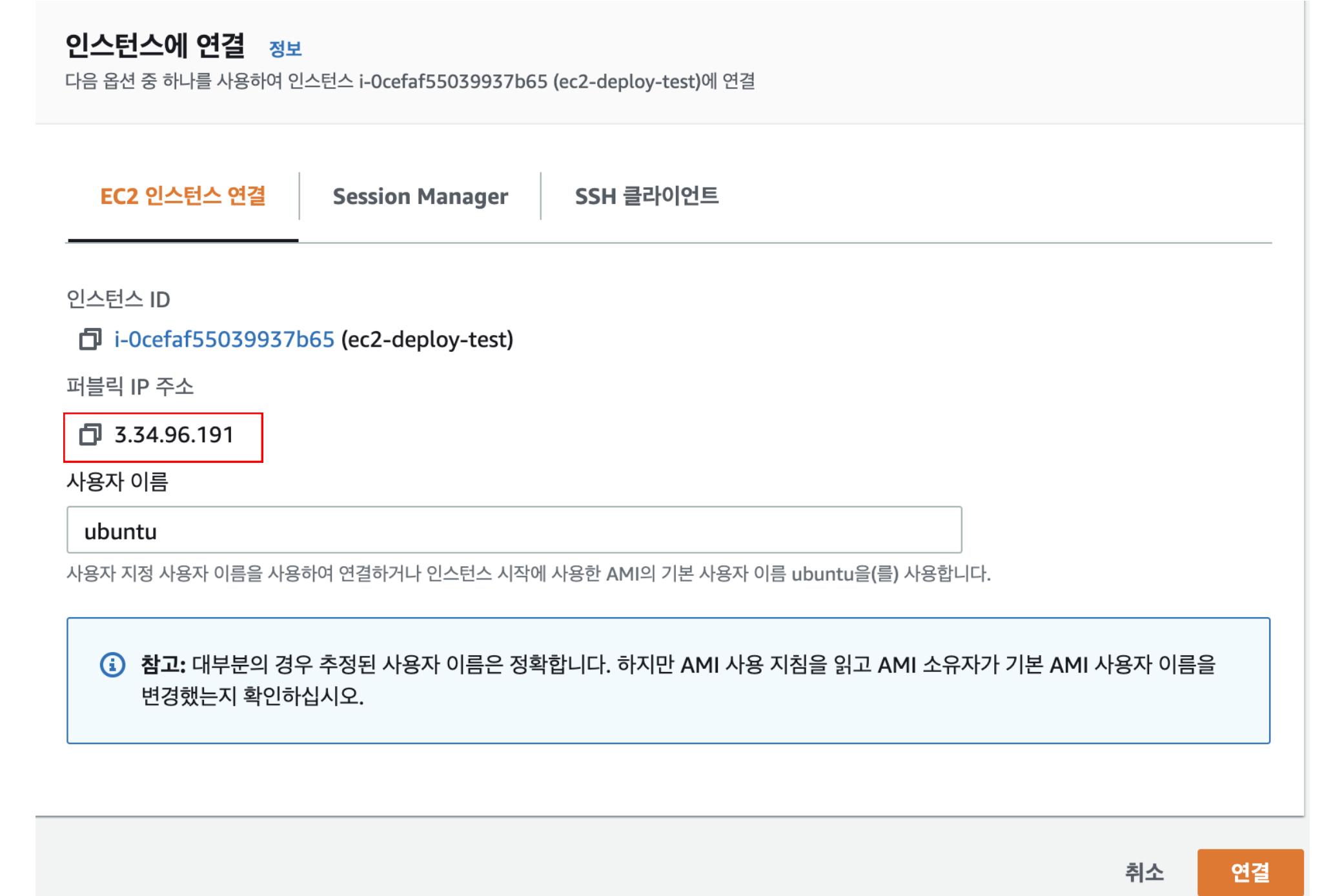
Nginx 설치

```
sudo apt-get update
```

```
sudo apt-get install nginx
```

```
cd /etc/nginx
```

```
sudo service nginx start(/etc/nginx/ 위치에서)
```



Nginx 실행 확인

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

가상 환경 설치

```
sudo apt-get update
```

```
sudo apt-get install python3-pip
```

```
export LANG=C.UTF-8 (환경 변수 설정)
```

```
export LC_ALL=C.UTF-8 (환경 변수 설정)
```

```
pip3 install pipenv
```

프로젝트 Clone

```
mkdir django
```

```
cd django
```

```
git clone https://github.com/parkjihwanjay/next-lion-session-django.git
```

```
cd next-lion-session-django
```

```
pipenv shell
```

```
pipenv install
```

```
cd deployTest
```

```
python3 manage.py runserver
```

접속 안된다 왜 why?

8000 포트에 대해 막아놓았기 때문에!

Nginx Django(8000 포트) 연결

sudo vim 파일 이름(ex) sudo vim django)

cd /etc/nginx/sites-available

cd /etc/nginx/sites-enabled/

sudo ln -s ..sites-available/파일이름(sites-available에 만들어 놓은 파일 이름)/심볼릭 링크 이름
ex) sudo ln -s ..sites-available/django django

```
● ● ● ec2 — ubuntu@ip-172-31-25-11: /etc/nginx/sites-available
server {
    listen 80;

    server_name 13.124.233.136;

    location / {
        proxy_pass http://127.0.0.1:8000/;
    }
}
```

EC2 instance public IP

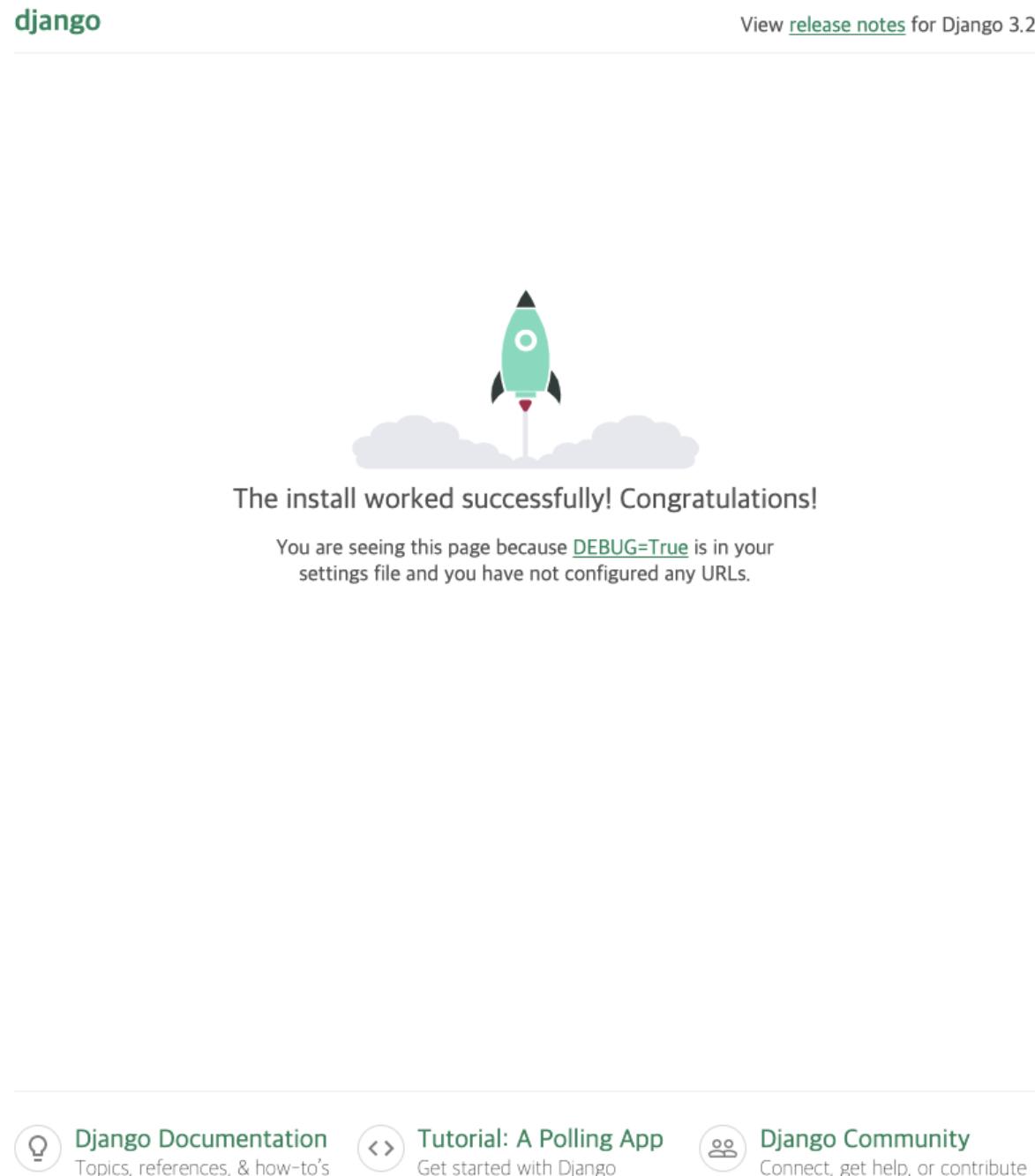
```
[ubuntu@ip-172-31-25-11:~$ cd /etc/nginx/sites-enabled/
[ubuntu@ip-172-31-25-11:/etc/nginx/sites-enabled$ sudo ln -s ..sites-available/django django
[ubuntu@ip-172-31-25-11:/etc/nginx/sites-enabled$ ls
default django
ubuntu@ip-172-31-25-11:/etc/nginx/sites-enabled$ ]
```

sudo rm default

```
(django) ubuntu@ip-172-31-15-18:/etc/nginx/sites-enabled$ sudo rm default
(django) ubuntu@ip-172-31-15-18:/etc/nginx/sites-enabled$ ls
django
```

sudo service nginx restart

django server



python3 manage.py runserver 0.0.0.0:8000

연결 왜 Why?

80번 포트로 연결된 Nginx가 8000번 포트로 연결해주기 때문

static 설정

```
ROOT_DIR = os.path.dirname(BASE_DIR)

STATIC_URL = '/static/'
STATIC_DIR = os.path.join(BASE_DIR, 'static')

STATICFILES_DIRS = [
    STATIC_DIR,
]

STATIC_ROOT = os.path.join(ROOT_DIR, '.static_root')
```

static 파일들(css, font, img, js...)이 앱 주변에 흩어져있다.

runserver할 경우 자동으로 모아주지만 production에서는 수동으로 모아줘야한다.

python3 manage.py collectstatic => .static_root에 다 모인다

nginx static 설정

```
server {
    listen 80;
    server_name 54.180.151.118;
    location / {
        proxy_pass http://127.0.0.1:8000/;
    }
    location /static/ {
        alias /home/ubuntu/django/next-lion-session-django/.static_root/;
    }
}
```

`sudo vi /etc/nginx/sites-available/django`

/static/으로 요청으면 django로 넘어가지 않고 .static_root에서 찾겠다.

웹서버로서의 역할

Gunicorn 설치

pipenv install gunicorn(Pipfile이 있는 위치에서)

nohup pipenv run gunicorn 프로젝트 이름.wsgi:application --bind=127.0.0.1:8000&

manage.py가 있는 위치에서

gunicorn

django [View release notes](#) for Django 3.2



The install worked successfully! Congratulations!

You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.

 [Django Documentation](#)
Topics, references, & how-to's

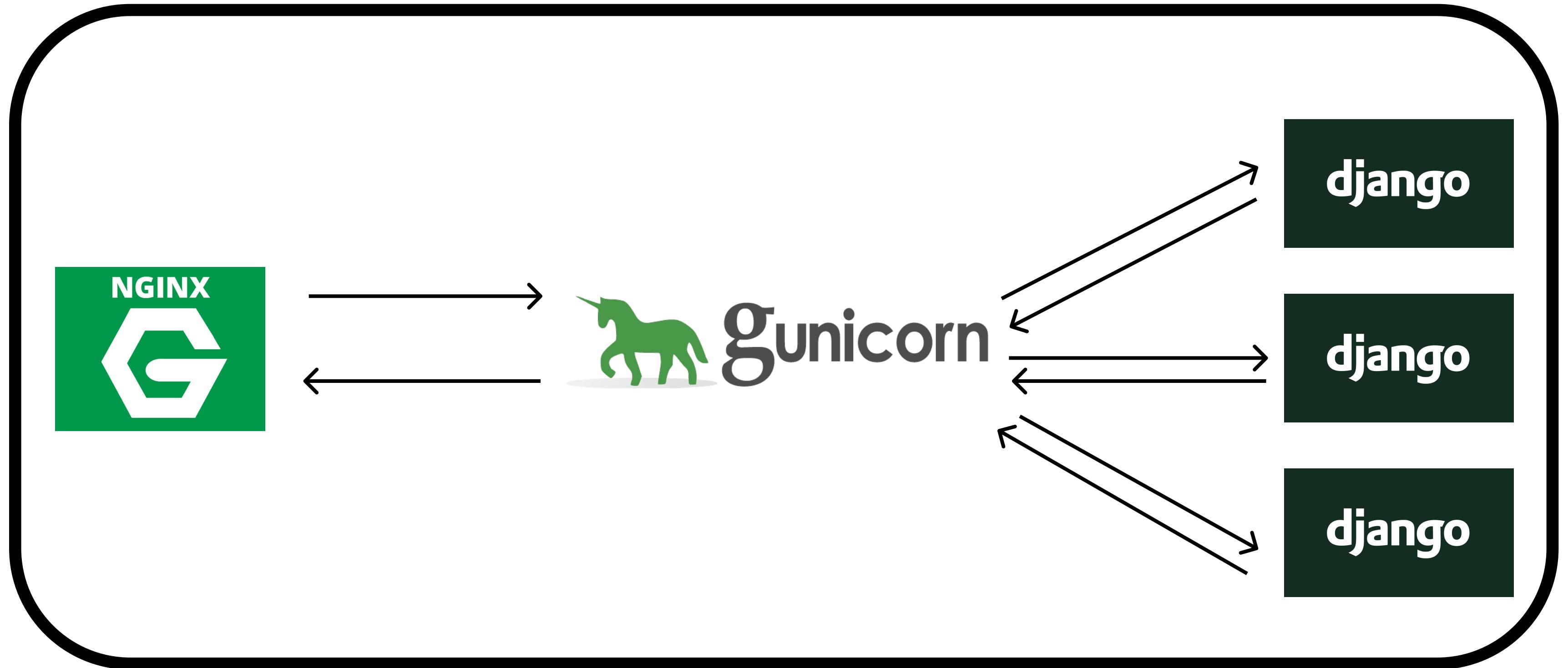
 [Tutorial: A Polling App](#)

 [Django Community](#)
Connect, get help, or contribute



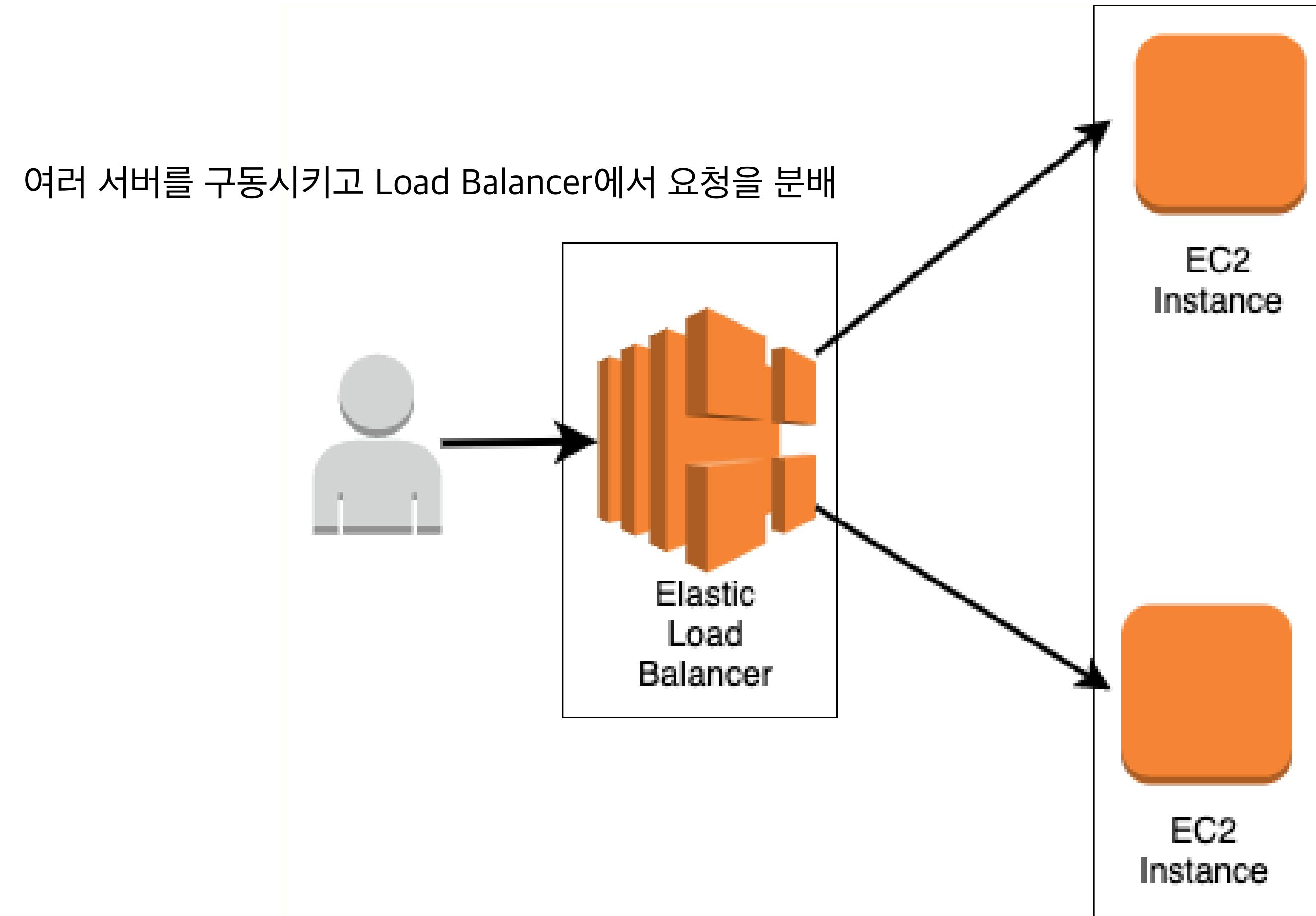


Request
→
← Response
CSS, JS, Img, ...



규모가 커지면?

서버 상황에 따라 서버를 자동으로 scale하는 auto scaling



서비스 상황, 타겟 고객층, 예상 트래픽 등 고려해서 정하기

마지막

생성한 EC2 인스턴스 삭제

감사합니다



하루빨리 코로나 시국이 끝나고 다 같이 모이는 날이 오기를..

하시는 일 모두 화이팅입니다~!!

참고

<https://parkjihwan.tistory.com/3>

<https://parkjihwan.tistory.com/8>
