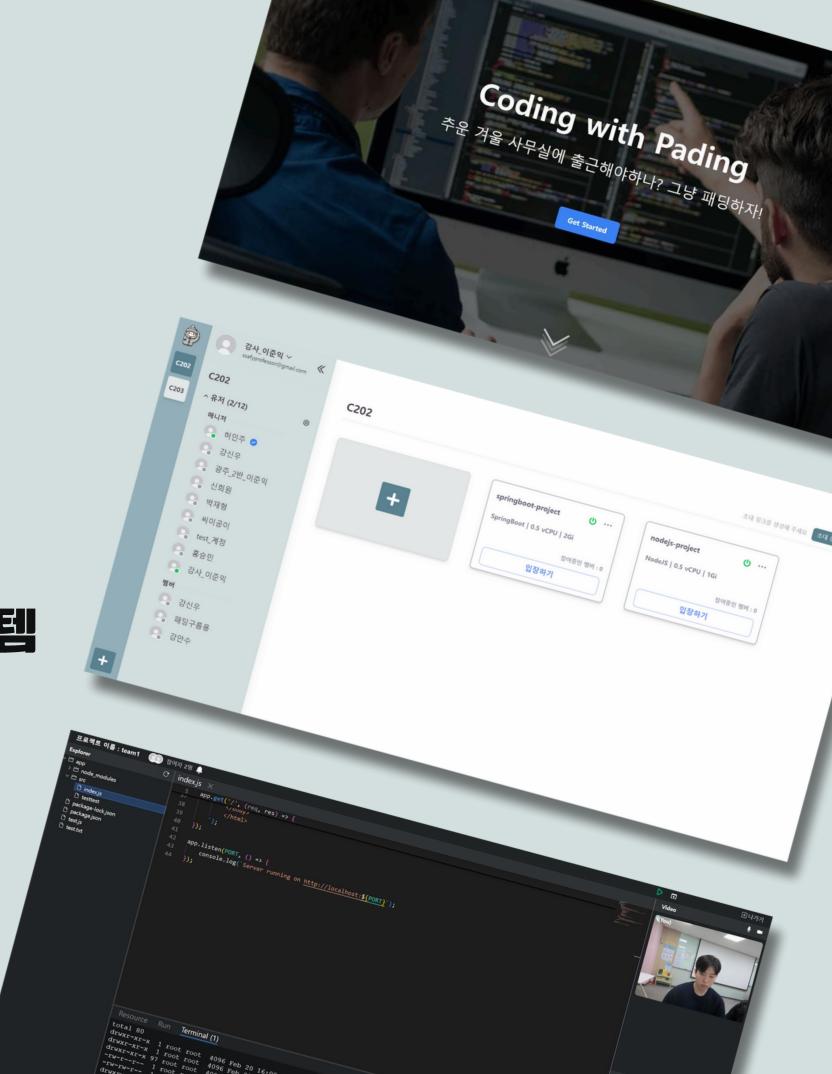
SSAFY 공통 프로젝트 C202팀



페어 프로그래밍을 위한 웹 IDE 및 관리 시스템



# Contents

01 페어프로그래밍

04 시연

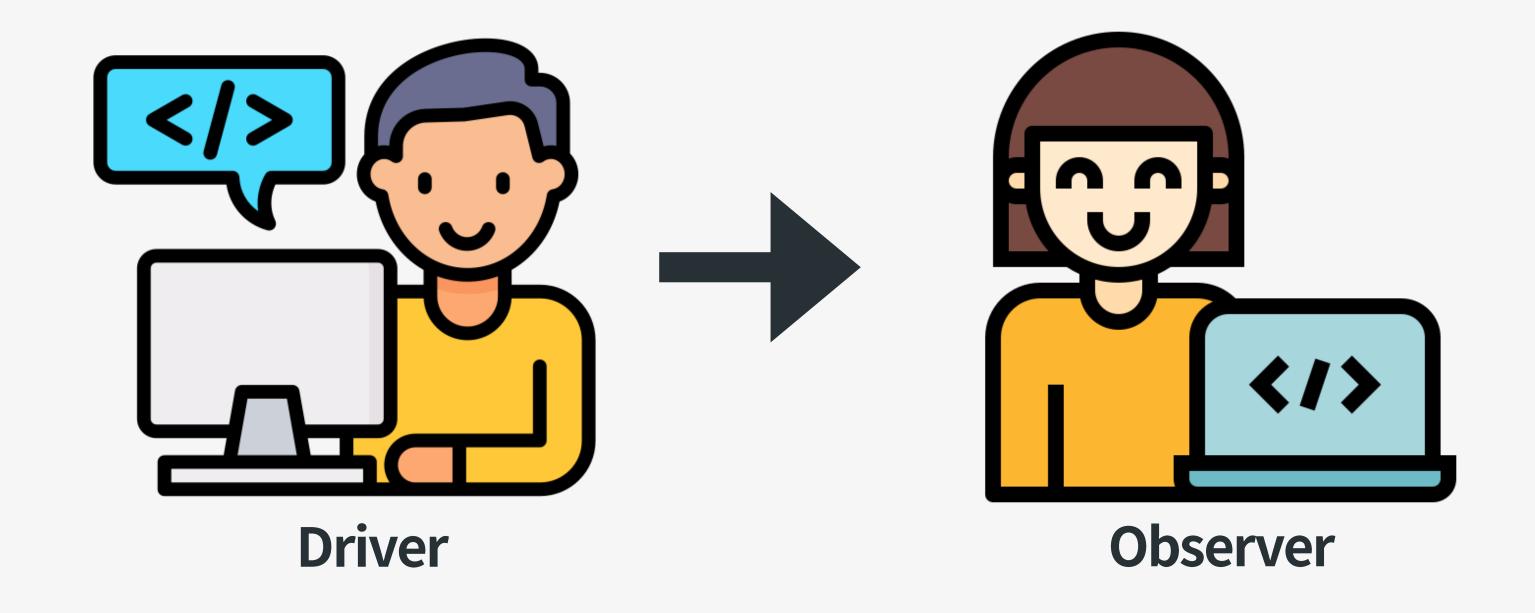
02 기획배경

05 주요기술구현

03 기능

06 실사용후기

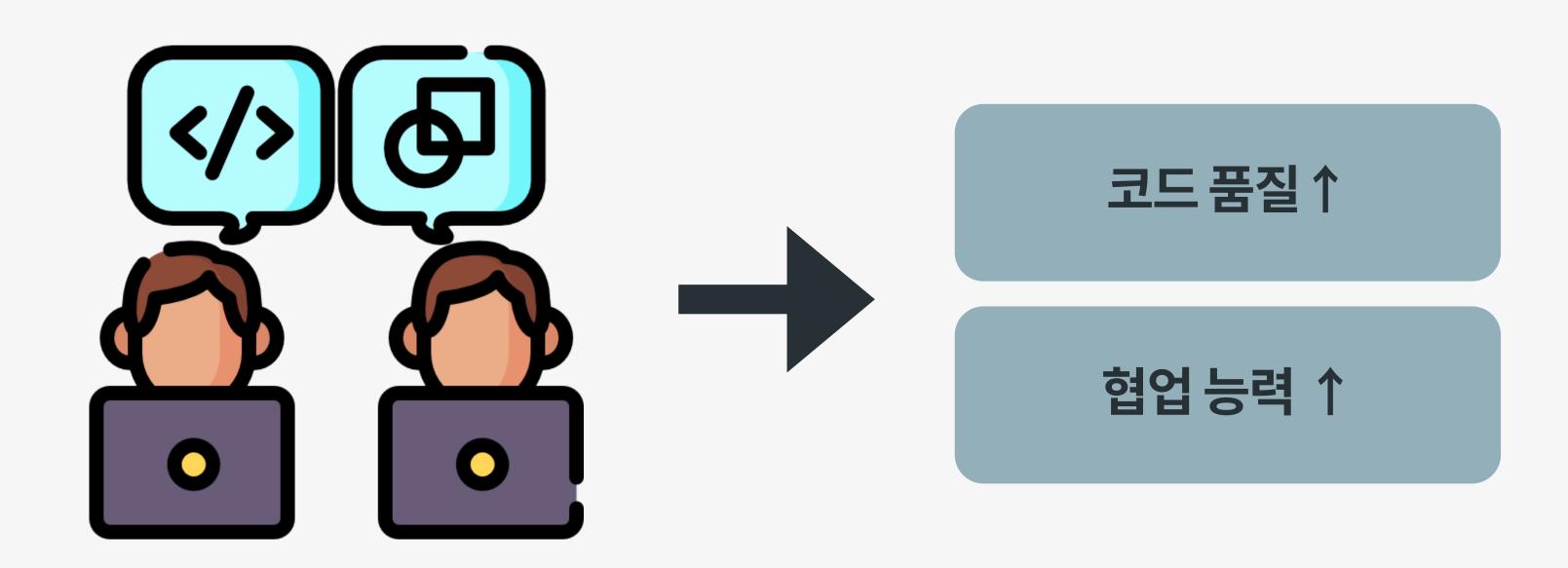
# 01 페어프로그래밍이란?



### 01 페어 프로그래밍의 장점



### 01 페어 프로그래밍의 장점



### 01 페어 프로그래밍 사례













# 02 문제 정의



### 02 문제 정의

화상 채팅

동시 편집

매니징 시스템

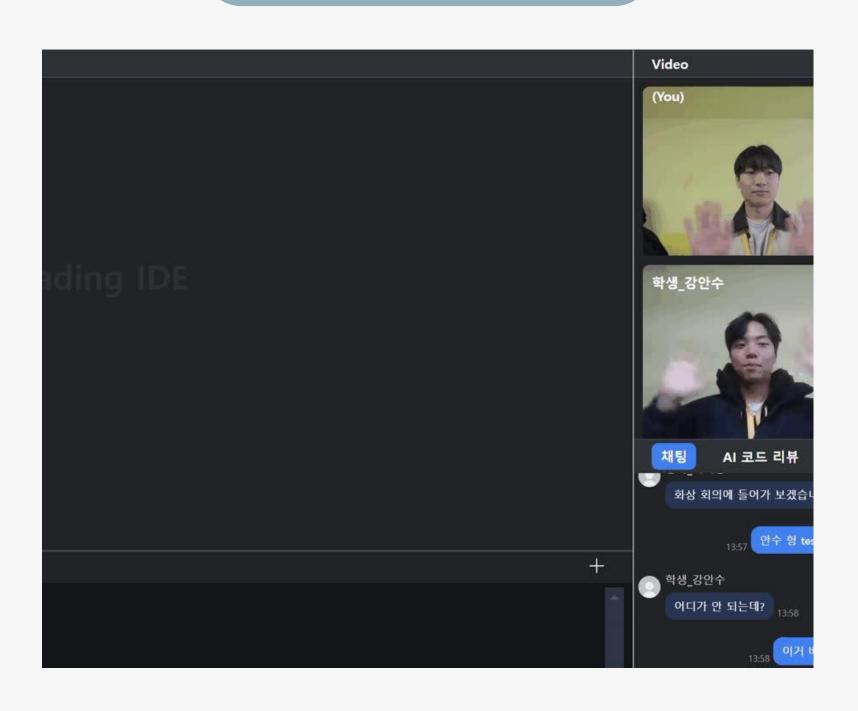
자동 배포



Pair + Coding

### 화상 채팅

### 동시 편집



```
// 변수 선언 및 할담
let message = "Hello, world!";
let count = 0;
7/ 함수 정의
function greet(name) {
   return 'Hello, ${name}!';
// 반복문 테스트
for (let i = 0; i < 5; i++) {
   console.log(greet("User"));
    count++;
// 객체 및 배열 활용
const user = {
   name "John Doe",
    age:
};
console.log(`User: ${user.name}, Age: ${user.age}`);
console.log("Hobbies:", user.hobbies.join(", "));
// 비동기 함수 테스트
async function fetchData() {
   return new Promise(resolve => {
           setTimeout(() => {
            resolve("Data loaded successfully!")
```

### 자동 배포

```
장여자 3명 🖡
     const express = require('express');
     const app = express();
const PORT = 3000;
     app.use(express.urlencoded({ extended: true }));
    function logRequest(req) {
   const timestamp = new Date().toISOString();
   console.log(`[LOG] $(timestamp) - $(req.method) $(req.url)`);
     // 최면 일레이 메서드 (서버 무하 느낌)
function randomDelay() {
  const delay = Math.floor(Math.random() * 300) + 100; // 180-400ms 편설 지연
  return new Promise(resolve -> setTimeout(resolve, delay));
     // ★ 서버 시작 시 멋있어 보이는 ASCII 이트 출력
function showBanner() {
  console.log(` ✔ Server is launching...`);
     // 출청 되일 제공s
app.get('/', async (req. res) => {
    logRequest(req);
    // await randomDelay(); // 현점 일레이 추가
    res.sendFile(path.join(__dirname, "views", 'index.html'));
      app.listen(PORT, () => {
           console.log( 

Server running on http://localhost:${PORT} );
t@team1-yhhx-55fb6bb4c9-4wvh5:/app# npm start
                                                                                                                                                                                                                                                                                 학<sup>학 학생</sup>
이거
 ode-ubuntu@1.0.0 start
ode src/index.js
erver is launching...
erver running on http://localhost:3000
G] 2025-02-25T06:07:58.244Z - GET /
```

### 매니징 시스템

	되인주	jjoonior97@gmail.com	오너
학생_3	당신우	sinukang333@gmail.com	멤버
강사_0	미준익	zz26Zzzc@gmail.com	매니저
학생_선	브피원	shinheewon0107@gmail.com	멤버
학생_8	암안수	ansoo97†@gmail.com	멤버
강사_박	박 <mark>재</mark> 형	ssafyprofessor@gmail.com	매니저
학생_2	김하준	student_1@pair-coding site	멤버
학생_0	미서윤	student_2@pair-coding.site	멤버
학생_박	박도현	student_3@pair-coding.site	멤버
학생_장	전지우·	student_4@pair-coding.site	멤버
학생_조	희민서	student_5@pair-coding site	멤버

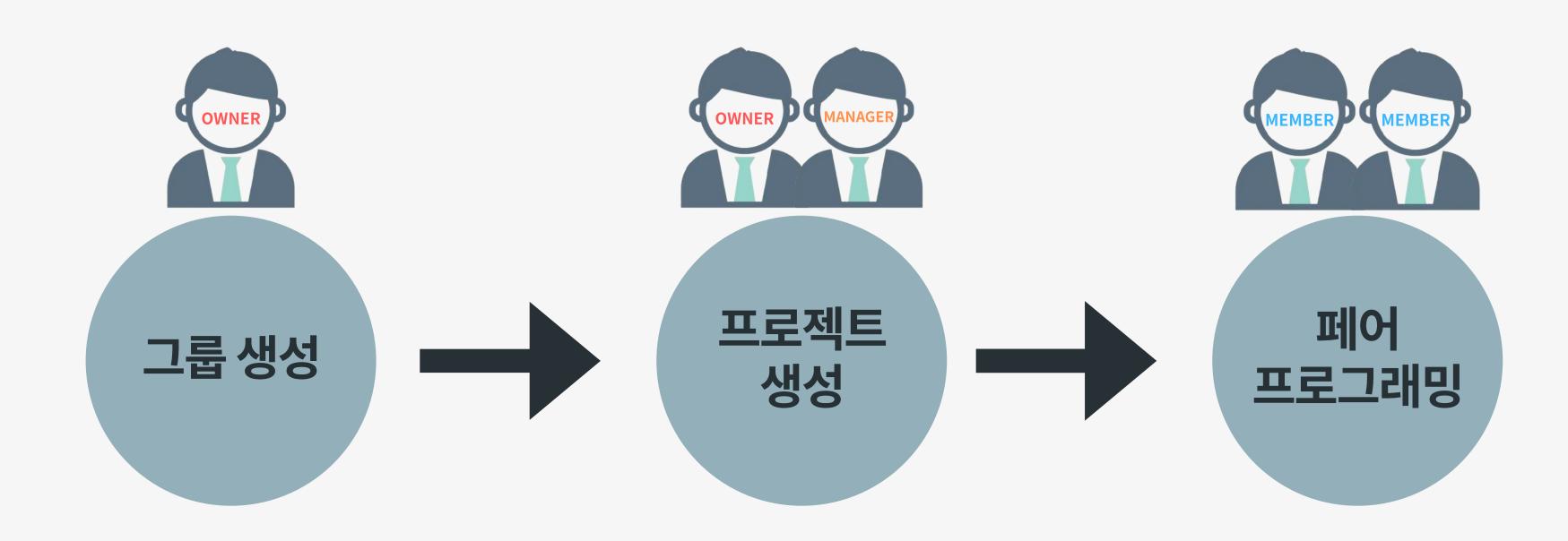
# 03 매니징시스템







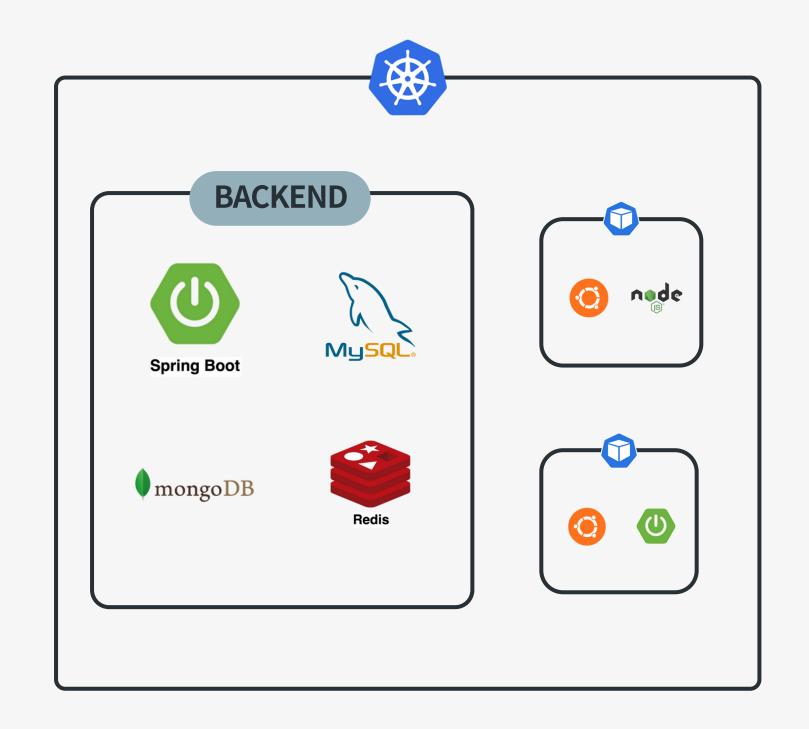
### 03 매니징시스템

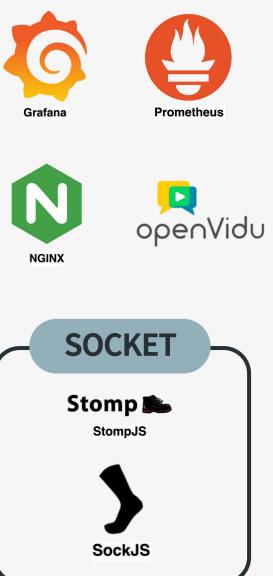


04 시연

# LIVE A CE

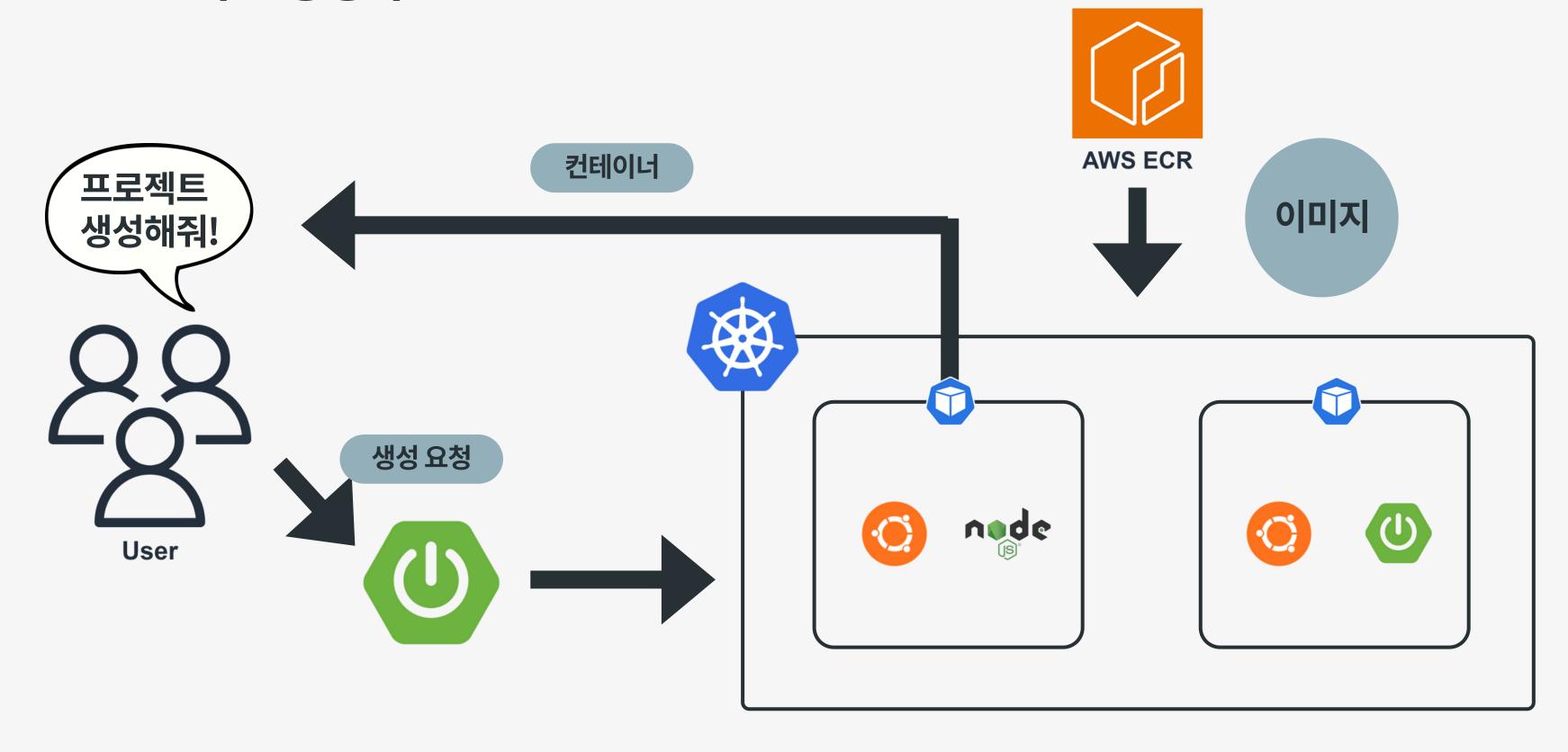
### 05 전체 아키텍처

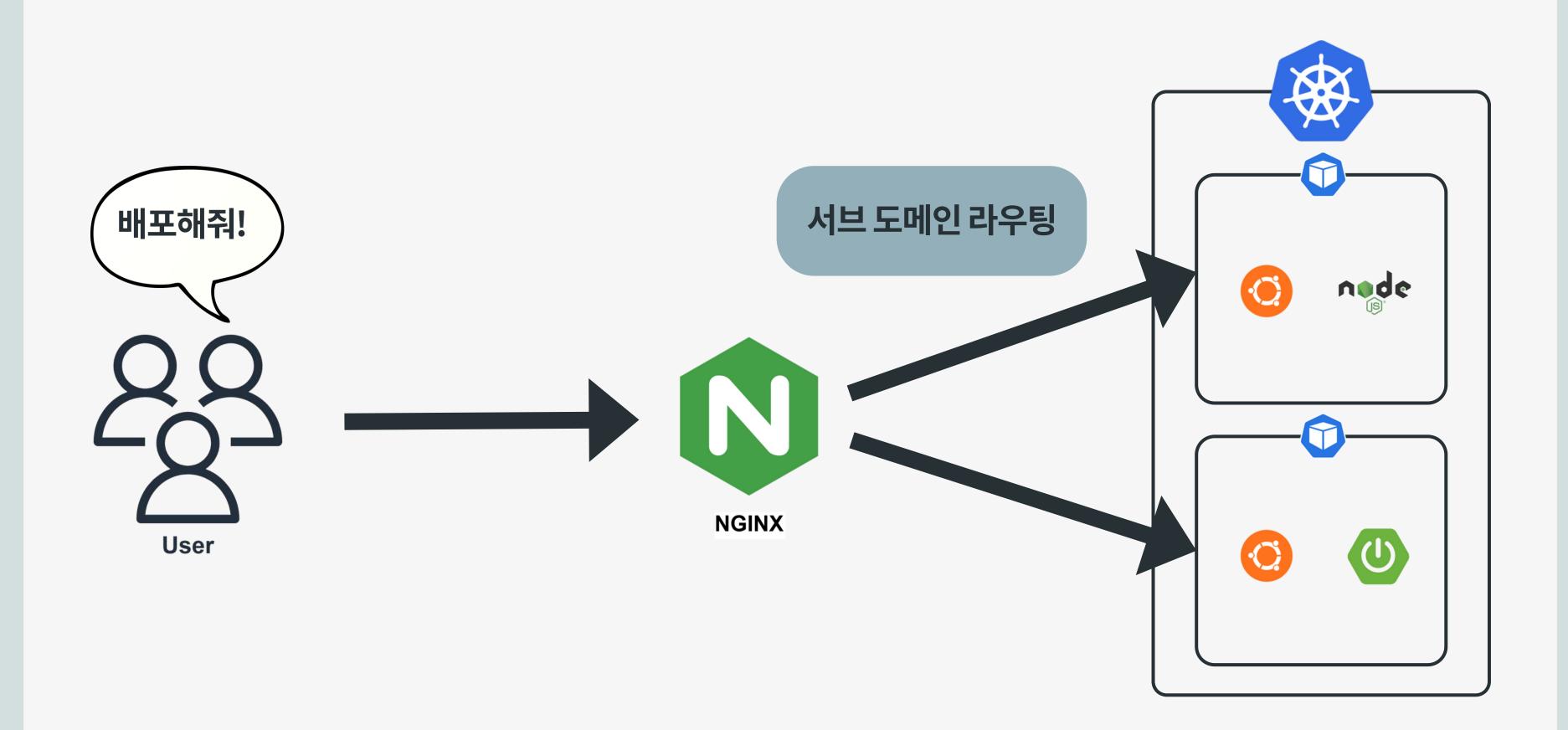






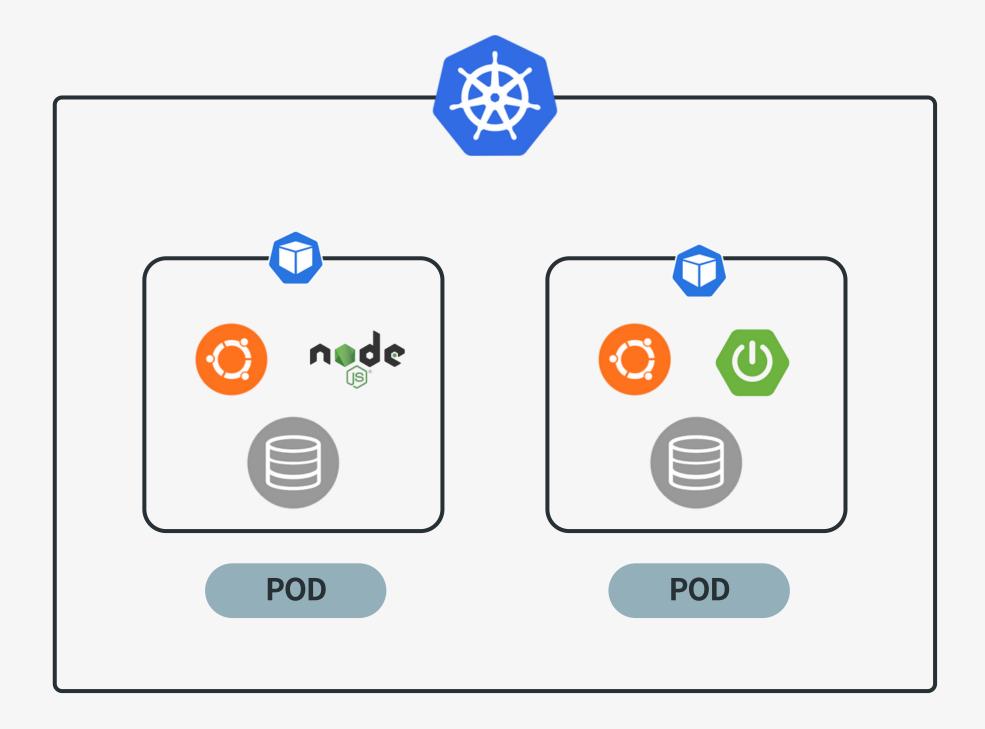
### 05 프로젝트 생성 구현





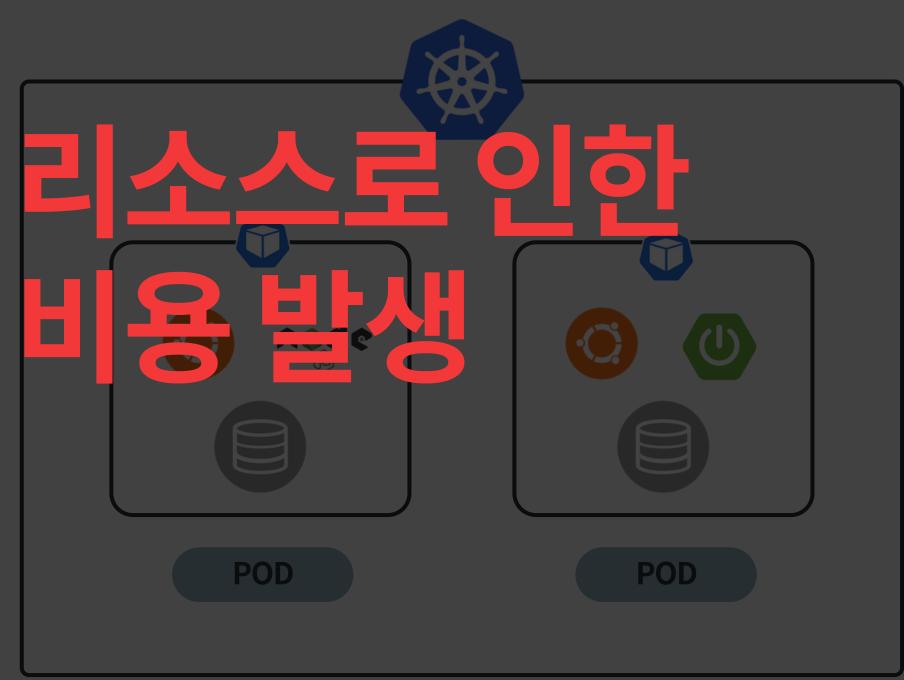
# 05 컨테이너 관리 (초기 버전)

컨테이너 자원을 관리하기 위해 쿠버네티스와 POD 도입

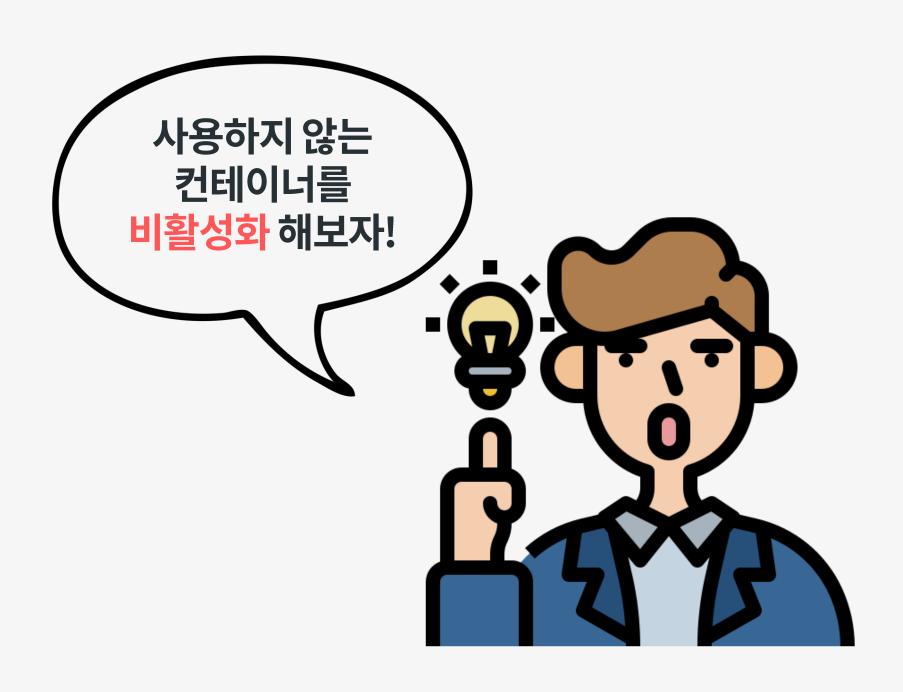


### 05 컨테이너 관리 (초기 버전)

전테이너 자원을 사이 관리하기 위해 보고 무버네티스와 POD 도입



# 05 컨테이너 관리 아이디어



### 05 여전한문제

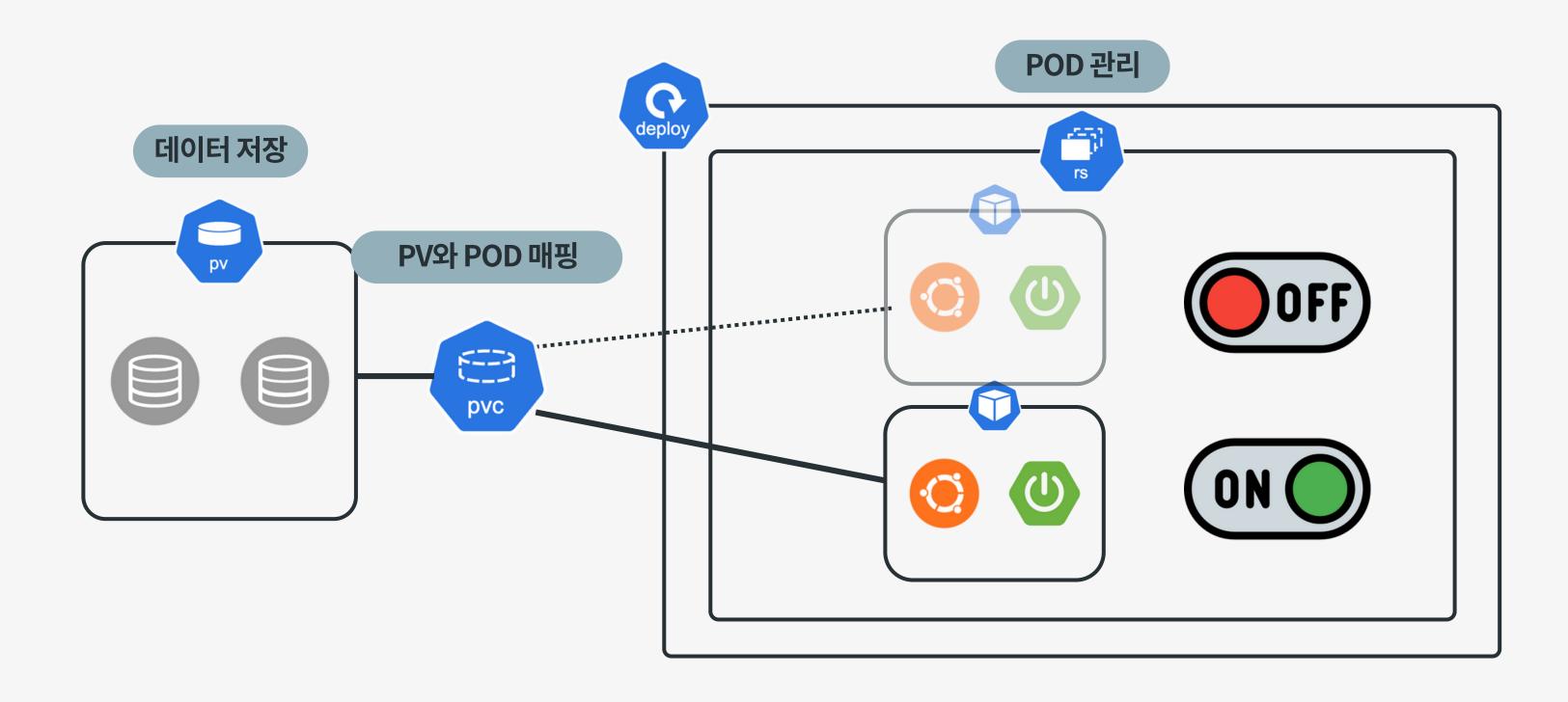


Deployment Replica Set (RS)

Н

Persistent Volume (PV)
Persistent Volume Claim (PVC)

# 05 컨테이너 관리 (개선 버전)

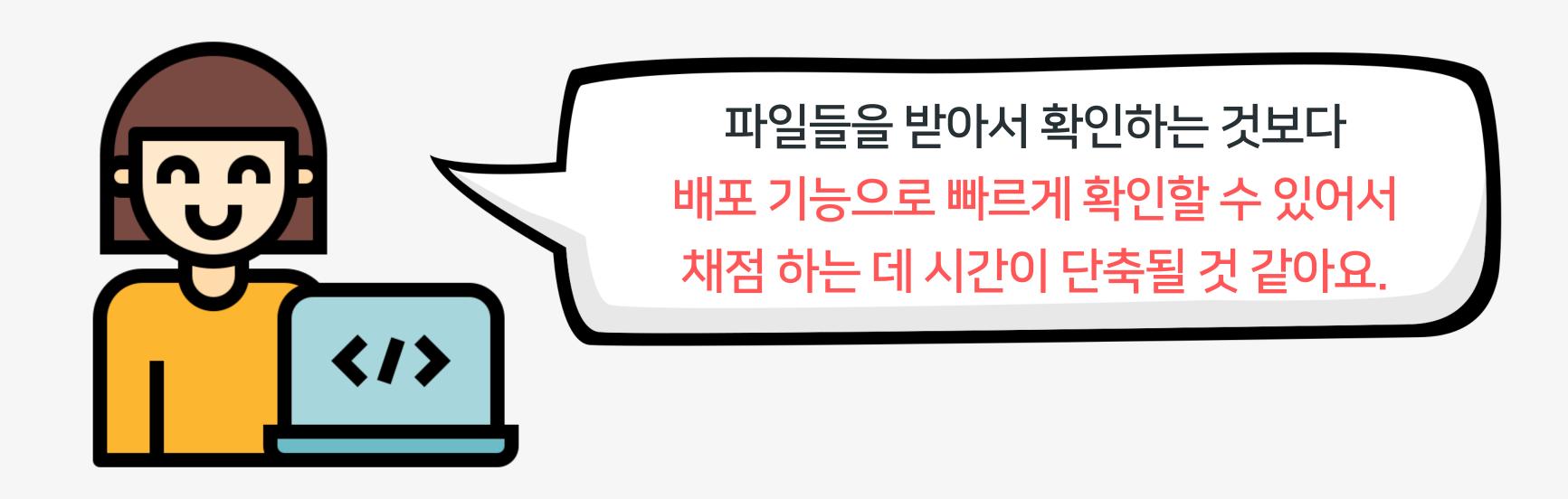


### 06 실사용후기



팀 프로젝트 시, 하나의 코드로 같이 작업을 하기 때문에 병합오류가 발생하는 번거로움이 사라집니다.

### 06 실사용후기



# Thank you

감사합니다.