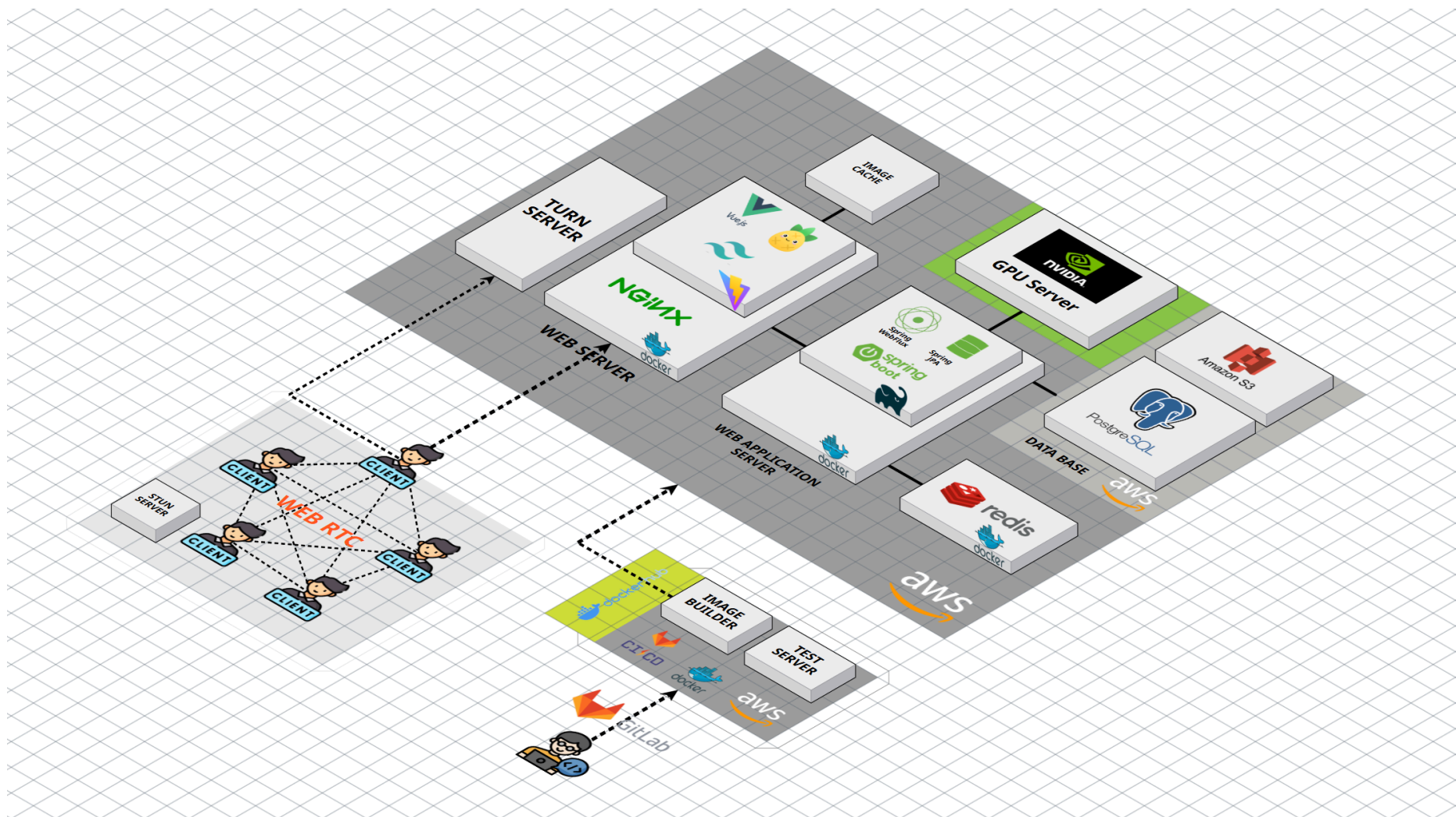




# 시스템 아키텍처



“아주 먼 옛날”은 WebRTC 기반의 실시간 유저 통신, LLM 모델과 FLUX 모델을 활용한 이미지 생성 그리고 **Spring Boot** 기반의 백엔드 애플리케이션 등을 포함한 아키텍처 갖추고 있습니다. Docker 컨테이너를 기반으로 구축되었으며, CI/CD 및 클라우드 인프라를 활용하여 효율적이며 확장 가능한 서비스를 제공하고자 하였습니다.

# 도커를 쓴 이유

## 1. 일관된 배포 환경

- ⇒ 일관된 배포 환경을 제공하여 시스템 안정성을 높이고자 하였음
- ⇒ 세팅값을 미리 설정

## 2. 의존성 관리 용이

- ⇒ 애플리케이션과 그에 필요한 라이브러리를 컨테이너 안에 포함시켜서 외부에 영향을 받지도 주지도 않도록 함.
- ⇒ **방학 프로젝트 때, 이 문제 때문에 배포 실패함... 실제 경험임**

## 3. 리소스 효율성

- ⇒ 빌드와 실행에 필요한 리소스(java-alpine, nginx-alpine 등)만을 사용하여 성능 최적화 용이  
(VM 보다 도커가 가벼움)

## 4. 손쉬운 확장

- ⇒ 이미지(~~세팅이 끝난 컨테이너 찍어내는 빵틀~~)를 통하여 손쉬운 확장이 가능함.

## 5. 버전 관리

- ⇒ 도커 허브로 이미지 버전을 관리하여 오류가 났을시 신속하게 롤백 가능

## 6. 컨테이너 오케스트레이션

- ⇒ 전체 시스템을 한 손에 넣고? 관리하기 좋음

# WebRTC

1. 서버 리소스를 최소한으로 줄이고자 하였다.

2. webRTC를 통하여 실시간으로 유저들끼리 데이터를 주고 받으면서 서버에 보내는 유저들의 request수를 획기적으로 줄이고자 했다.

3. 또한,