
오프라인 멀티 게임의 클라우드 서
비스 환경 구축
추억의구름빵
START

〈프로젝트 소개〉

클라우드 기반 고전 게임 서비스 제작하기

공간의 제약

Problem1

- 고전 게임 중 멀티 플레이 게임들의 경우 오프라인 멀티 플레이 게임이 대부분.
- 친구와 만나서 게임하기에는 공간적 제약이 많음.

복잡한 설치

Problem2

- 고전게임을 플레이 하기 위해서는 대부분 에뮬레이터를 설치해야함.
- 하지만 설치 과정이 복잡하고 버전 호환의 문제가 있음.

보안상 문제

Problem3

- 고전 게임들의 경우 인터넷에서 파일을 받기 때문에 악성 바이러스의 위협에 노출됨.

‹ 프로젝트 목표 ›

클라우드 기반
고전 게임 서비스
제작하기

목표 1 ☑ **웹**에서의 플레이를 통한 높은 접근성!

목표 2 ☑ 동일한 **키 입력**에서도 동작하는 2P 플레이!

목표 3 ☑ **실제 맵남** 없이 플레이할 수 있는 로비 시스템!

목표 4 ☑ 고전 게임의 **개체 보존** 및 홍보!

〈 프로젝트 진행 〉

아키텍처 구상

게임 구동 부분

게임

실제 게임을 구동하는 부분
가상 화면과 오디오 장치 이용

REC 녹화

FFMpeg를 사용한
화면 및 소리 녹화

송출

녹화한 화면 과 오디오 정보를
실시간으로 송출

트래픽

인그레스를 활용한
트래픽 제어

클러스터 매니페스트

- 로그 수집 및 중앙 관리
- 클러스터 상태 모니터링 대시보드 구성
- git ops 기반 CI/CD Workflow 구축

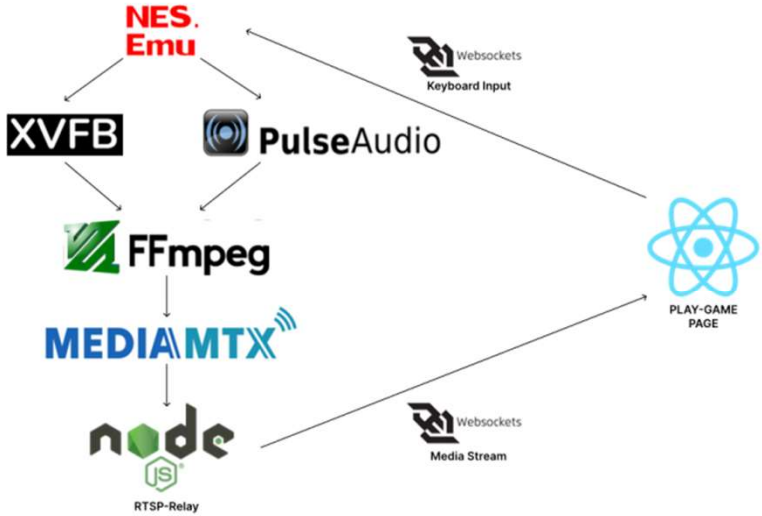
웹 서비스 부분

- Sock.io의 Room을 이용한 로비 기능
- in memory 데이터베이스 Redis 사용

아키텍처 세부 사항

게임 스트리밍

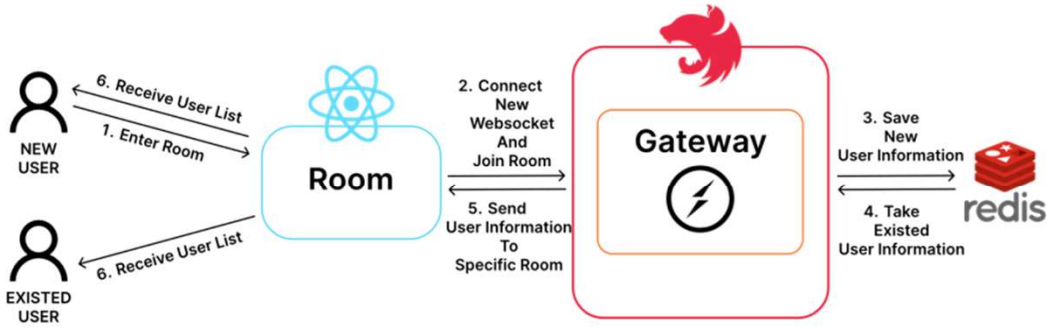
- 게임 구동 및 FFMpeg를 통한 녹화, MediaMtx를 사용한 스트리밍 생성, JSMpeg를 사용한 영상 생성으로 구성
- Nes Emulator 등의 게임 컨테이너에서 게임을 구동한 뒤, FFMpeg를 사용하여 화면과 소리를 녹화.
- 녹화된 스트림은 MediaMtx를 사용하여 스트리밍을 생성 후 사용자에게 전달.
- 사용자의 각 웹 브라우저에서 JSMpeg를 이용해 수신받은 영상 스트리밍 정보를 송출.



아키텍처 세부 사항

로비 시스템

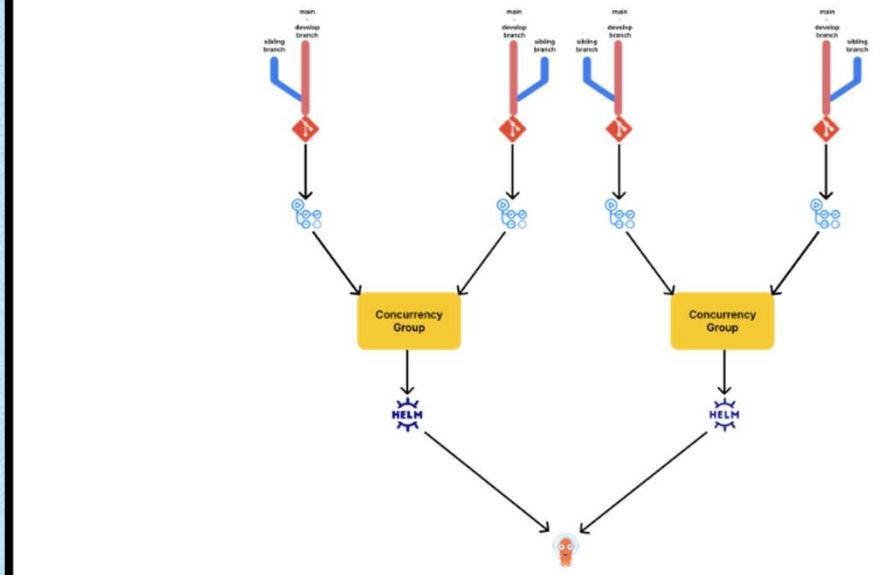
- 로비 시스템을 구축하기 위해 socket.io의 room을 사용.
- room의 정보를 저장하기 위해 in memory 데이터베이스인 redis를 사용
- redis를 활용한 빠른 접근을 보장함과 동시에 socket.io의 room에서 보관할 수 없는 추가적인 정보 보관



아키텍처 세부 사항

gitops 기반의 CI/CD Workflow

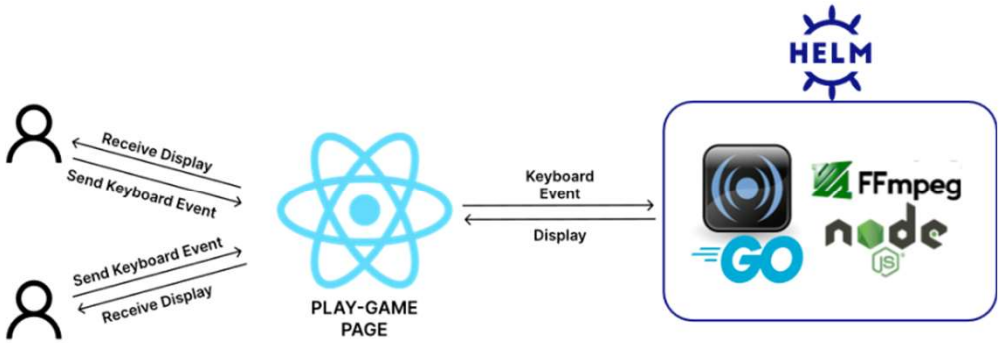
- 개발의 편리성을 보장하기 위해 CI / CD 파이프라인을 구축
- GitOps 방식의 CI / CD 구성하여 단일 진실 공급원(SSOT)의 원칙 준수
- github action에서 헬름 차트 레포를 업데이트 하는 부분의 동시성 문제를 해결하기 위해서 Concurrecy Group 설정



아키텍처 세부 사항

gitops 기반의 CI/CD Workflow

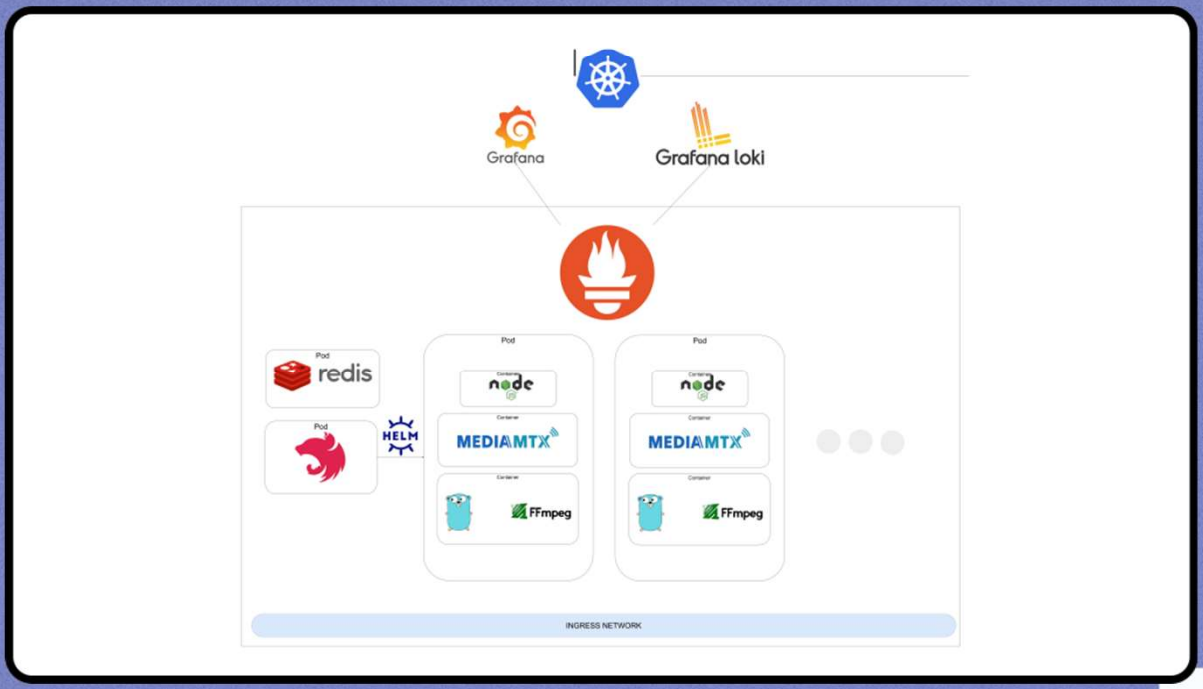
- 방에 들어온 유저가 모두 Ready 상태일 경우 쿠버네티스에 대한 배포 권한을 가지고 있는 서버가 동적으로 게임 서버 헬름 차트를 배포할 수 있도록 구성
- 각 게임 서버는 독립적인 url을 가지고 ingress controller에 추가됨



아키텍처 세부 사항

리버스 프록시를 통한 트래픽 제어

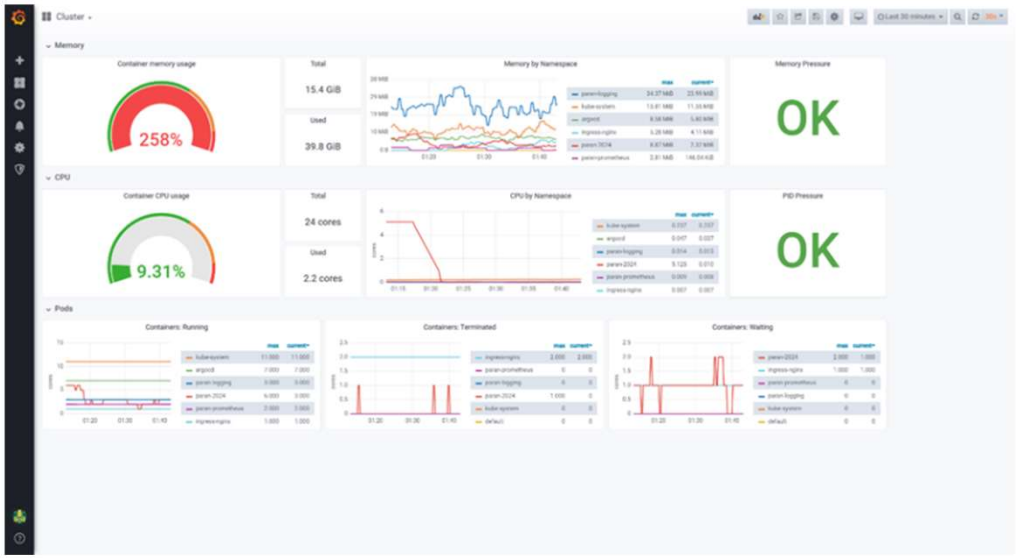
- ☑ Ingress Nginx를 사용하여 클러스터 내부에 리버스 프록시를 배치. 외부에서 들어오는 모든 트래픽은 이 리버스 프록시를 거쳐 내부에 존재하는 컨테이너들로 전달되도록 구성.
- ☑ 보안적 이점 확보 및 클러스터 매니페스트 관련 부분과 같이 외부로 노출되면 안되는 페이지들의 접근 제어.



아키텍처 세부 사항

컨테이너 자원 모니터링

- 컨테이너가 여러 개 배치될 경우 클러스터에 과부화가 올 수 있음. 이에 따라 클러스터의 현재 상태를 지속적으로 모니터링할 도구가 필요.
- 프로메테우스를 이용해 cadvisor와 kube-system-metrics 파드에서 클러스터의 상태 정보를 가져오도록 구성.
- 그라파나를 이용해 프로메테우스에서 수집한 정보를 대시보드 형태로 송출




THANK YOU
YOU WIN!!