About Me

Information

- 이름 : 윤순상
- 생년월일 : 1998.08.18
- 주소지 : 서울특별시 노원구

Awards

- 2018 Hallym SW Week Bug-fix 금상

Education

- 2018.02 ~ 2024.02
  강원도 춘천 한림대학교 소프트웨어융합대학 졸업
- 컴퓨터공학과 전공 (빅데이터 / 콘텐츠IT)
  4.13 / 4.5
- 2023.12.06 ~ 2024.06.11
  RAPA 카카오 스쿨 클라우드 엔지니어 4기

Contact

- Tel : 010-4657-0097
- Email : tnstkd98@gmail.com
- Github : https://github.com/Yunsoonsang

Skills

language

- Python, Shell
- C, Java, SQL
- Javascript

Monitoring

- ELK Stack
- Prometheus, Grafana
- Node-Exporter, CAdvisor

Cloud

- Docker, Dokcer-compose
- Dokcer-swarm, Kubernetes
- Terraform(KVM, Openstack, AWS)
- KVM, AWS, Openstack

CI/CD

- Gitlab, Github, Privary Registry
- Jenkins, Sonarqube
- Ansible, Argocd

쿠버네티스 자동화 환경 배포 서비스

소개

- 쿠버네티스와 자동화 환경을 즉시 제공하는 프로젝트입니다.
- 기간 & 인원 : 2024.03.15 ~ 2024.04.03 / 5명

사용 기술

Kubernetes

Kubeadm(1.28), Metallb

Prometheus & Grafana

CI & CD

Github, Ngrok, Argocd

Jenkins, Sonarqube

App

Lan : HTML, CSS, JS

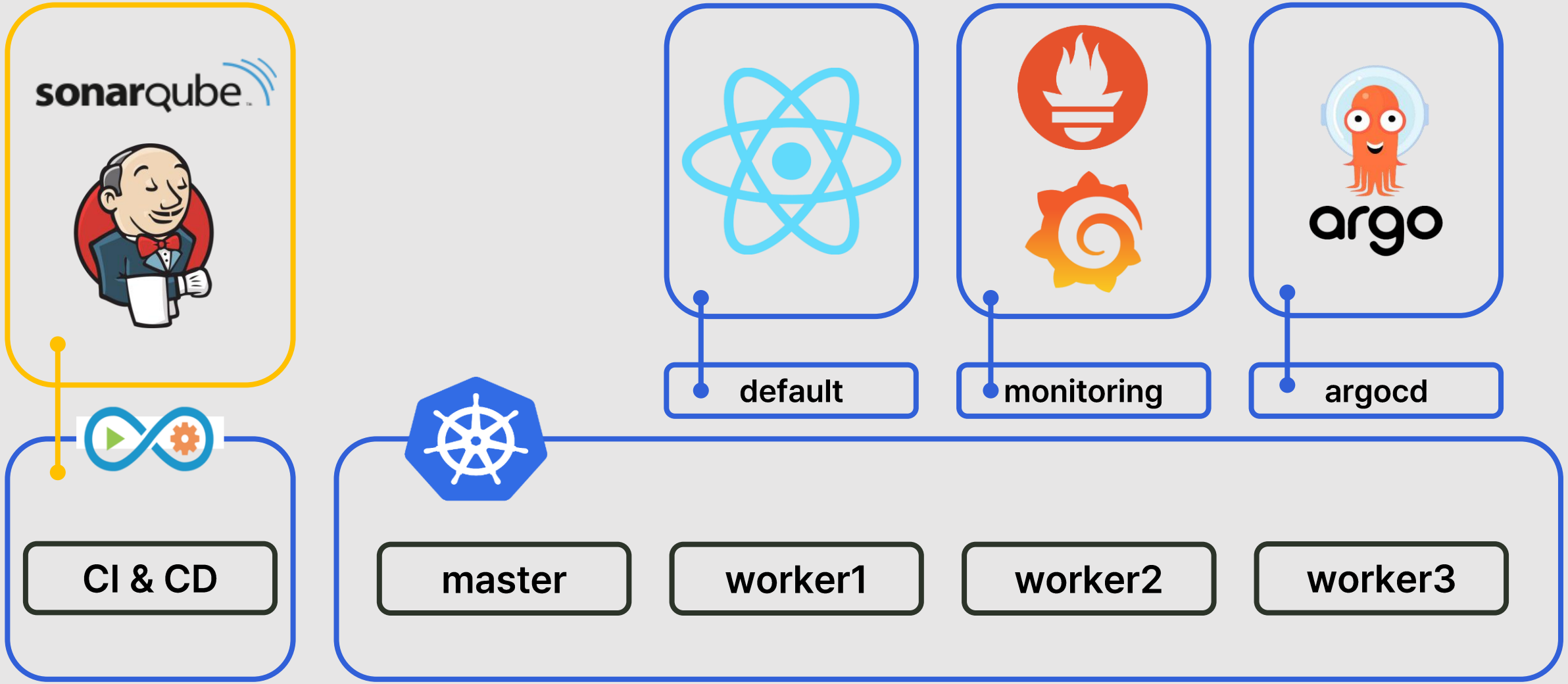
Framework : React

Build : Dockerfile

수행 역할

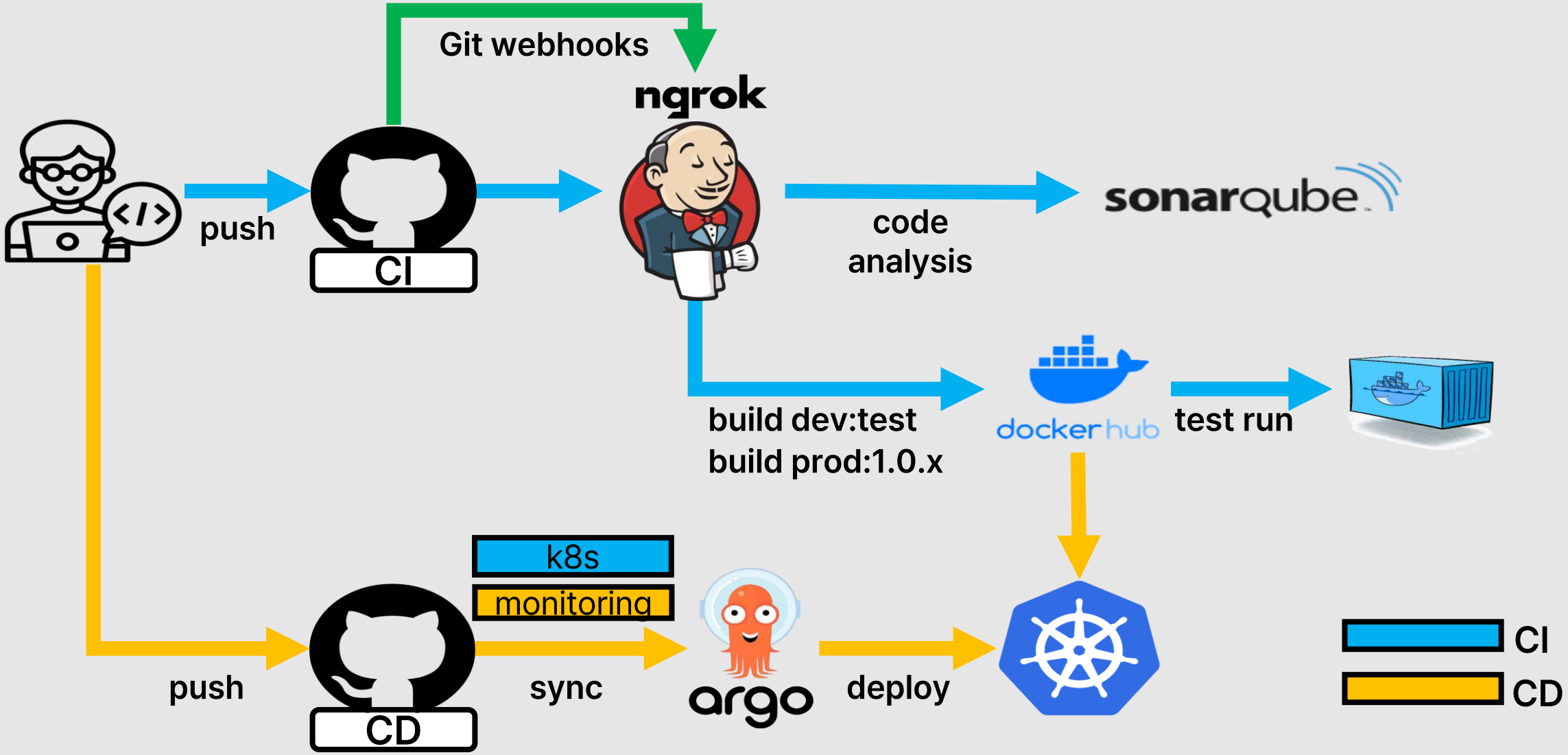
- **프로젝트 PM, CI & CD, 자동화 파이프라인**

Project Infra



github - rapaeng4-CamuCamu

Project CI & CD



github - rapaeng4-CamuCamu

CI 자동화

- VM속 Jenkins와 Webhook을 구성하기 위해서는 반드시 공인 IP, 도메인 주소가 필요했다. 이를 Ngrok으로 해결하여 CI 자동화를 구현할 수 있었습니다.
- **github - CI-pipeline**

CD 구성

- 빠른 배포를 위해 Argocd를 사용하였습니다.
- 다양한 배포 파일을 폴더로 구분하여 관리할 수 있도록 구성하였습니다.
- **github - CD-repo**

dev & prod build file

- CI 테스트 그리고 배포용 이미지는 분리되어 한다는 점을 고려하여 2가지 도커 파일을 제작하여 프로젝트를 진행했습니다. 또한 이미지 경량화를 위해 노력하였습니다.
- **github - Dockerfile**

DOCKER 배포 전략 모니터링

소개

- 컨테이너 오케스트레이션 도구인 Docker-Swarm을 사용해 대표적인 배포전략 중 Blue/Green 배포와 Canary배포 환경을 구현하고 각 배포 전략의 성능 지표를 확인해보는 프로젝트입니다.
- 기간 & 인원 : 2024.02.05 ~ 2024.02.13 / 1명

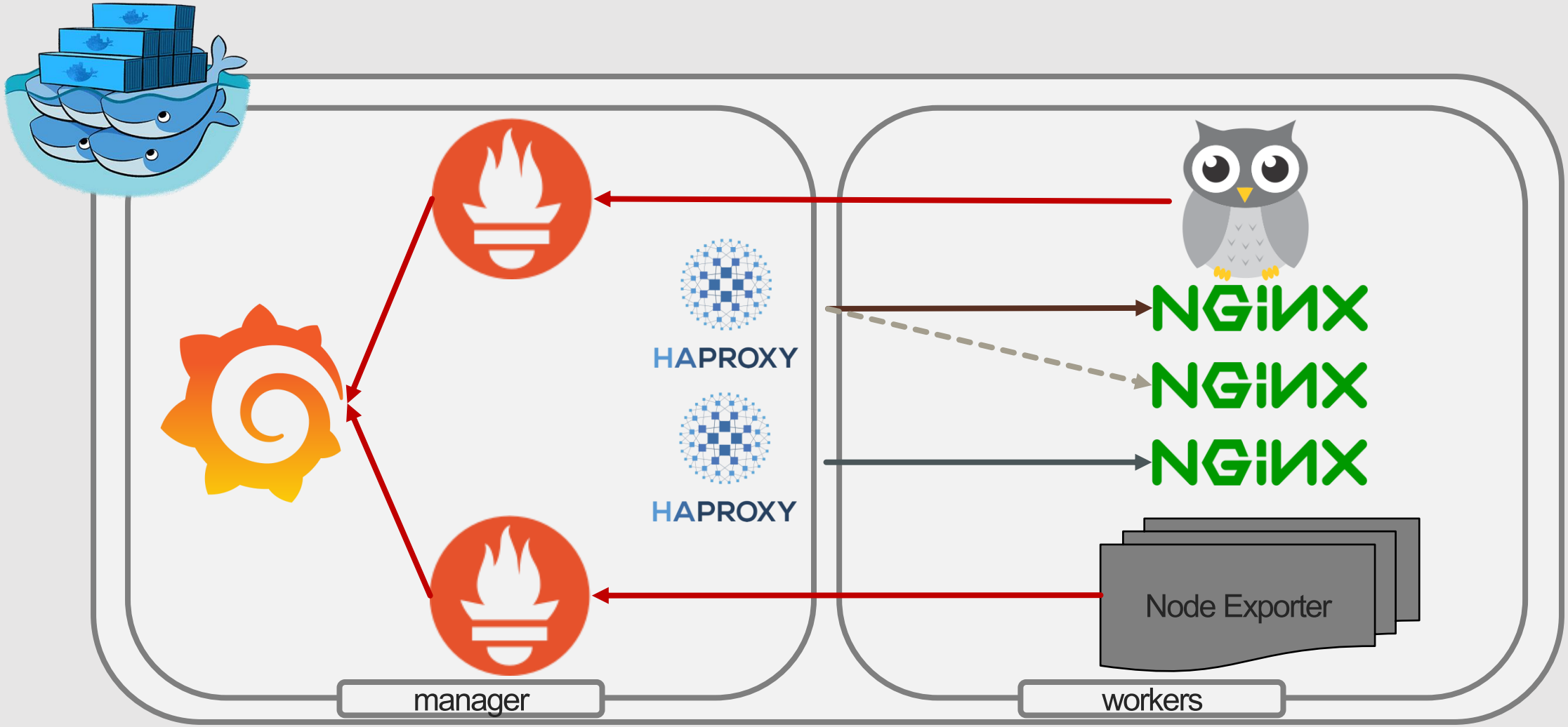
사용 기술

- **오케스트레이션 도구 – Docker-Stack**
- **모니터링 도구 - Prometheus & Grafana**
- **Metric 수집 도구 - Node Exporter & CAdvisor**
- **HAproxy**

수행 역할

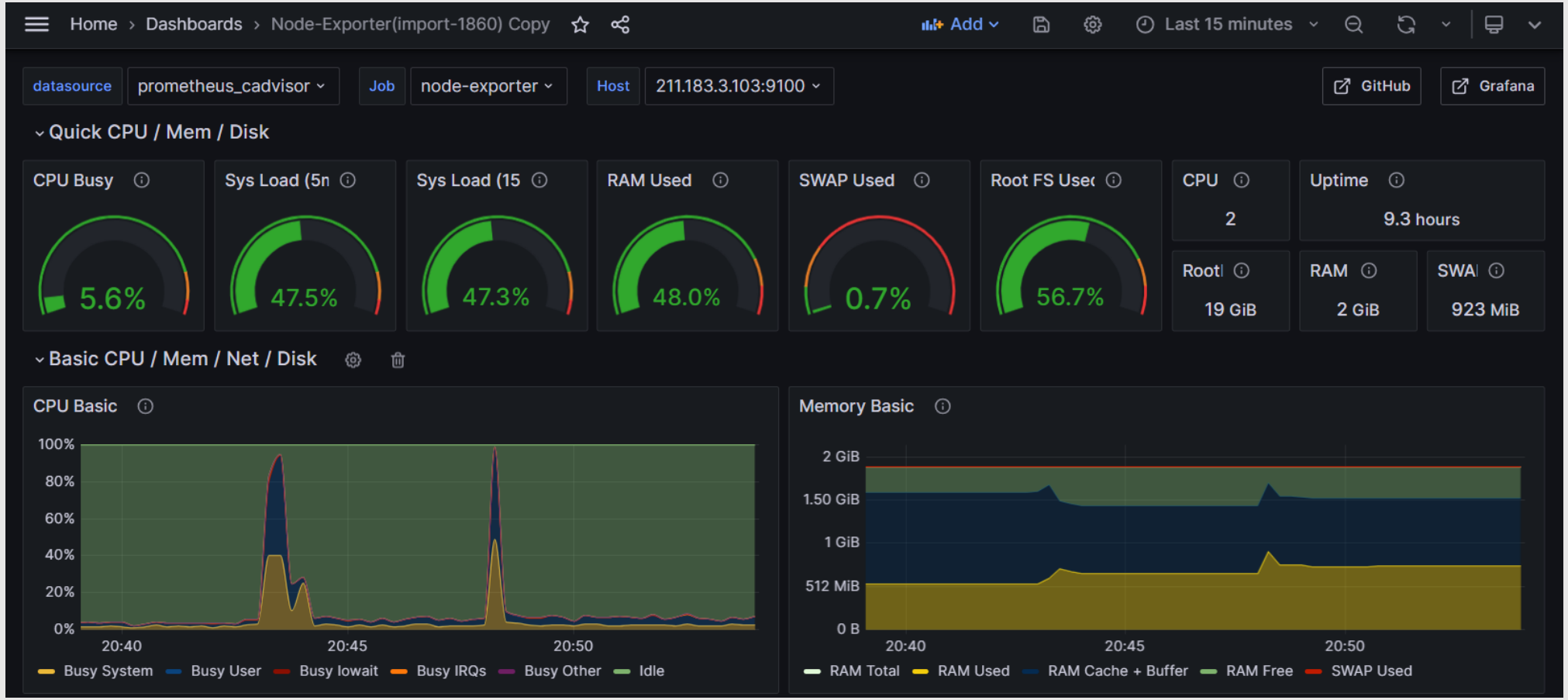
- **개인 프로젝트**

Project



github - docker-toyproject

Node Exporter – Worker 하나의 Metric 정보



HA프록시 이미지 만들기

```
andrewyss/canary      default   4840dd92eff5   8 minutes ago   122MB
andrewyss/proxy       green     8e487e822d8e   20 hours ago    584MB
andrewyss/proxy       blue      cffd900d8b72   20 hours ago    584MB
haproxy:latest로 더 간편하게 만드는 것이 가능하다. 용량도 훨씬 작다!
```

```
FROM haproxy:latest
COPY haproxy.cfg /usr/local/etc/haproxy/haproxy.cfg
EXPOSE 80
EXPOSE 8080
# HAProxy 실행 명령어
CMD ["haproxy", "-f", "/usr/local/etc/haproxy/haproxy.cfg"]
```

이미지 생성 시간 단축

- 처음에 OS컨테이너를 만들어 프록시로 활용했는데 이미지를 생성하는 시간이 오래 걸려 프록시 업데이트에 소요되는 시간이 오래 걸렸습니다.

해결 방법

- 프로젝트 이전에 사용했던 HA프록시 이미지를 분석해본 결과 HA프록시 도커 이미지를 베이스로 설정하면 된다는 점을 확인할 수 있었습니다. 결과적으로 이미지의 크기를 크게 줄이고 이미지를 생성하고 업데이트하는 시간을 크게 줄일 수 있었습니다.

KVM 클라우드 서비스

소개

- Shell로 구현해본 가상의 클라우드 서비스입니다.
- 총 5개의 노드를 사용하고 KVM을 이용해 Public Cloud에서 제공하는 가상 인스턴스를 생성하는 서비스를 구현했습니다.
- 기간 & 인원 : 2024.01.04 ~ 2024.01.08 / 5명

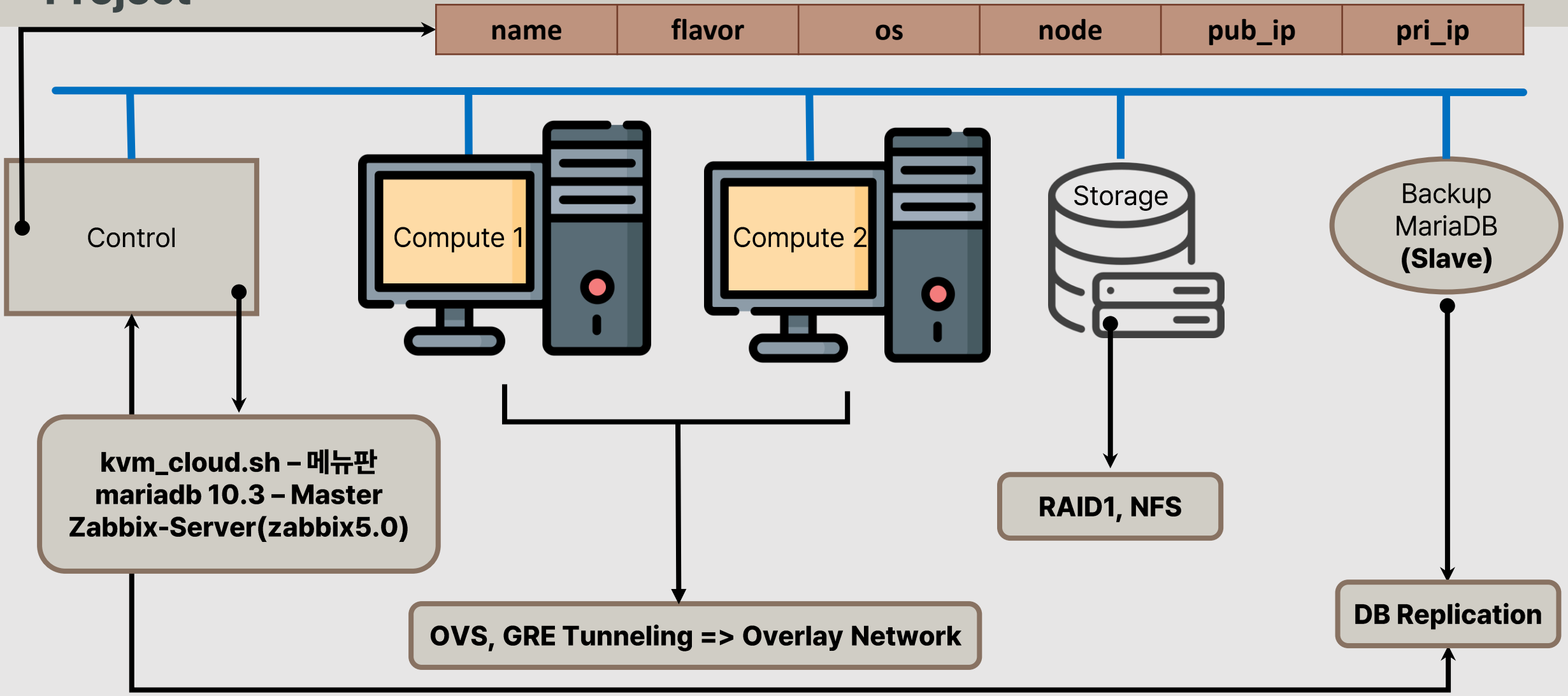
사용 기술

- **Control : Zabbix, MariaDB(Master), SSH PK, Shell Script**
- **Compute : KVM, OVS와 GRE Tunneling**
- **Storage : DISK RAID 1, NFS Server**
- **Backup : MariaDB(Slave) - DB Replication 구성**

수행 역할

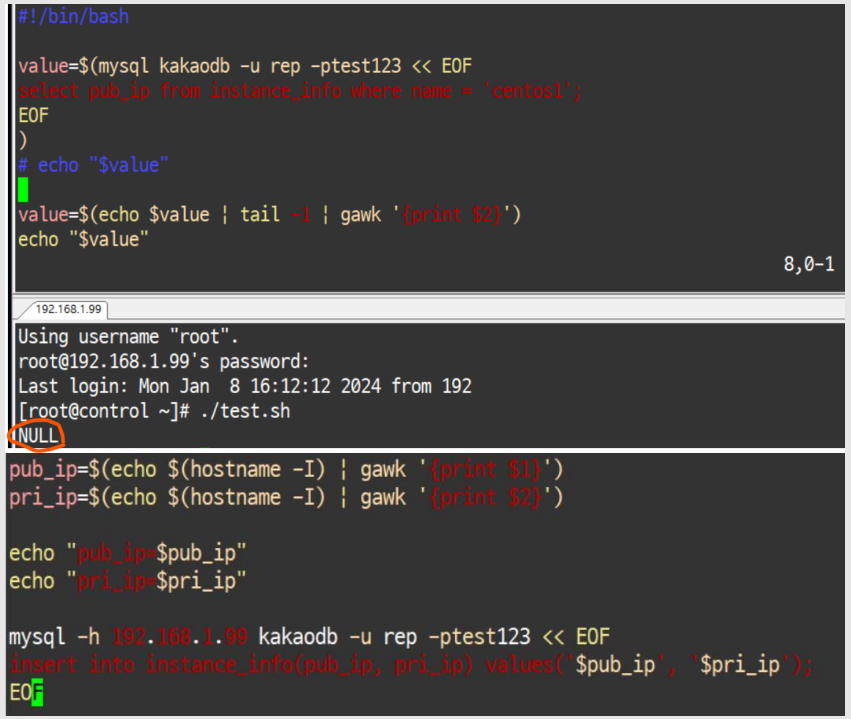
- **Shell Script 작성 및 DataBase 구성**

Project



github - kvm cloud

KVM 인스턴스의 IP 갱신



VM의 IP

- Compute에 생성되는 VM은 Compute의 vswitch에 의해서 DHCP 방식으로 IP가 자동할당되기 때문에 실행됐을 때 알 수 있습니다.

해결 방법

- VM에 사용되는 Volume을 Customize하여 해결할 수 있었습니다.
- Volume에 DB-Client를 설치하고 작성된 Shell Script를 넣어 VM이 시작됐을 때 실행되도록 설정합니다.

POSTRUE FLOW

소개

- COVID 이후 홈트레이닝 늘어남에 따라 AI기술을 통해 운동 자세를 교정 받을 수 있는 웹 서비스입니다.
- 기간 & 인원 : 2023.9.1 ~ 2023.11.21 / 5명

사용 기술

Front-end

Lan : Javascript

Framwork : React

Design : Figma

AI

Model : YOLOv4

Back-end

Lan : Python

Framwork : Flask

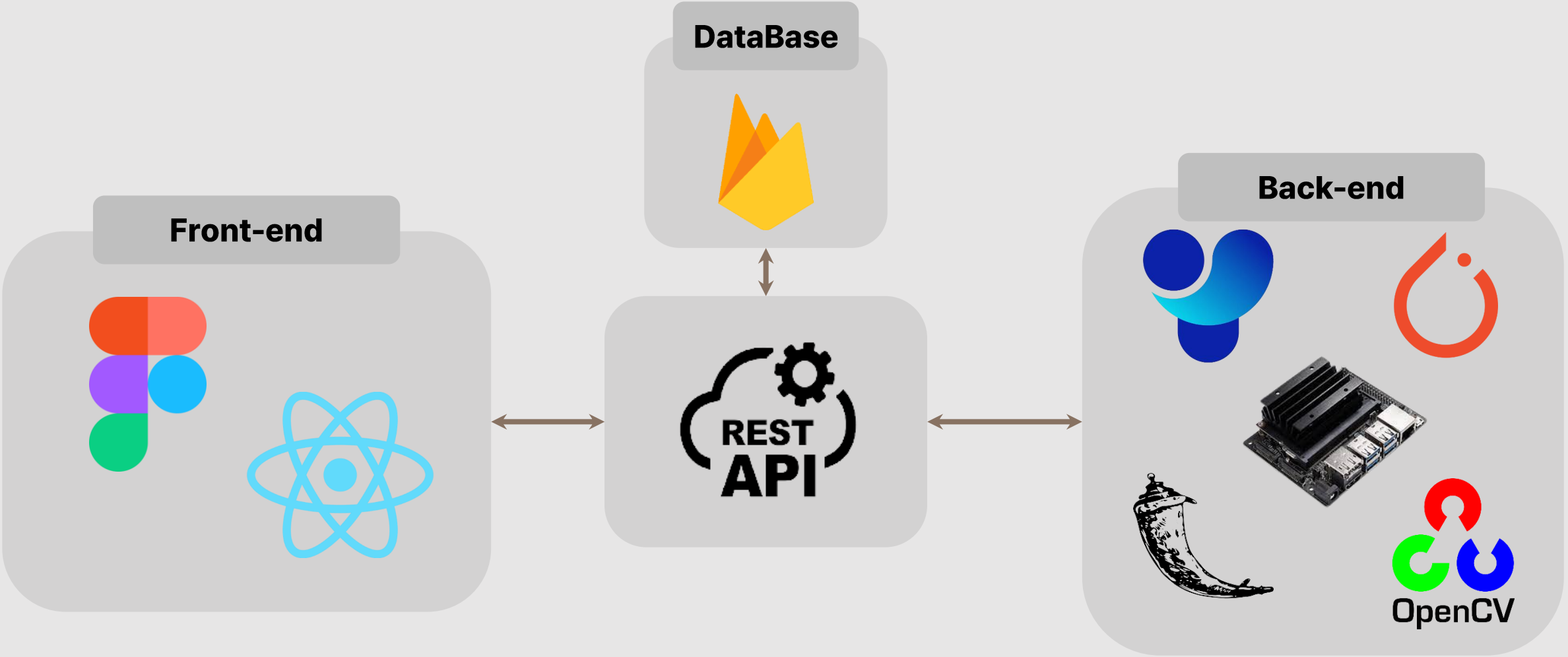
DB : Firebase

ENV : Embedded linux

수행 역할

- **Back-end, AI 최적화**

Project



github - backend
github - Team_PATI-캡스톤

환경 설정 문제

OpenCV

AI

```
yss@yss-desktop:~$ python3 check.py
OpenCV version :  4.5.4
torch version :  1.7.0
torchvision version :  0.9.0
Cuda is available? :  True
yss@yss-desktop:~$
```

Version 호환 이슈

- AI모델을 위한 라이브러리 설치가 매끄럽지 못하는 이슈로 너무 낮은 버전의 OpenCV, yolov4 모델을 사용하기 위한 ultralyics의 설치가 제대로 되지 않았습니다.

해결 방법

- Python 3.8 가상환경을 생성하여 필요한 버전의 OpenCV, ultralyics, pytorch를 모두 설치하여 해결할 수 있었습니다.

젯슨 나노 & AI

Memory 부족

- Jetson nano의 Memory는 4GB로 낮은 메모리 스펙입니다.
- 완성된 AI 모델을 이식하여 테스트할 때 멈춤 혹은 크게 버벅거리는 현상이 프로젝트를 진행할 수 없었습니다.

해결 방법

- 모델 경량화와 함께 Swap 메모리를 확보하였지만 그럼에도 성능저하를 피할 수 없었습니다.
- 따라서 AI팀과 협업하여 모델의 학습량과 학습 매커니즘을 단순한 Vertorize로 변경하여 해결할 수 있었습니다.

유명인의 작품 찾기 - AWS

소개

- AWS의 Rekognition 서비스를 사용하여 유명인의 사진 혹은 이름만으로 참여 작품들과 프로그램들을 출력해주고 추천해주는 기능이 있으면 좋을 것 같아 진행한 프로젝트입니다.
- 기간 & 인원 : 20.10.05 ~ 20.12.x (2개월) / 2명

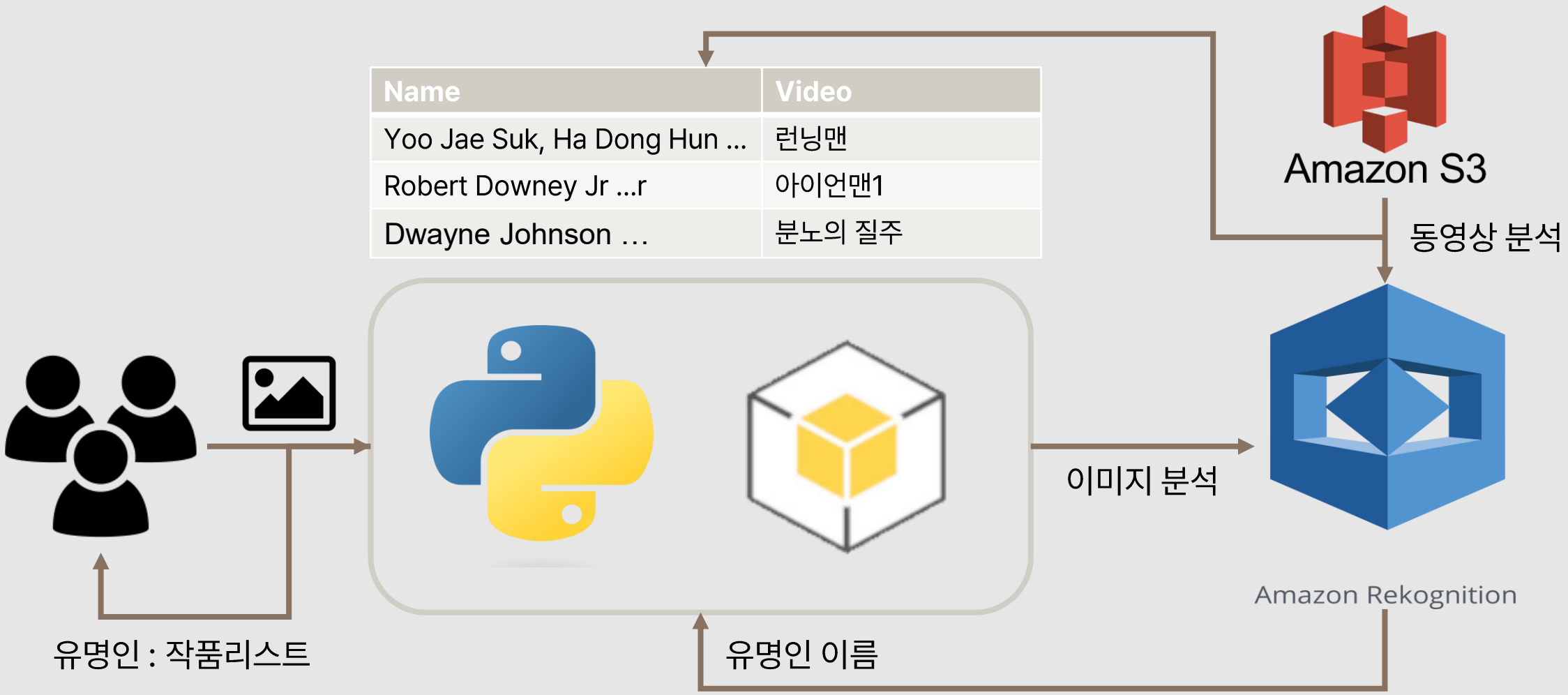
사용 기술

- **AWS S3 – 동영상 저장소**
- **AWS Rekognition – 이미지 데이터를 넘겨받아 처리**
- **AWS SDK – Python + Boto3**

수행 역할

- **아이디어 구현 담당**

Project



github - CC2020-2_Project

THANK YOU

Yun Soon Sang
tnstkd98@gmail.com
https://github.com/Yunsoonsang