

분산 Edge Cloud 환경에서의

Function as a Service 기능 개발

김도현, 송수현, 윤창섭, 송지원

김영한
지도 교수님



CONTENTS

1

서론

- 클라우드 컴퓨팅
- FaaS 인프라

2

본론

- 인프라 구축
- 서비스 제안

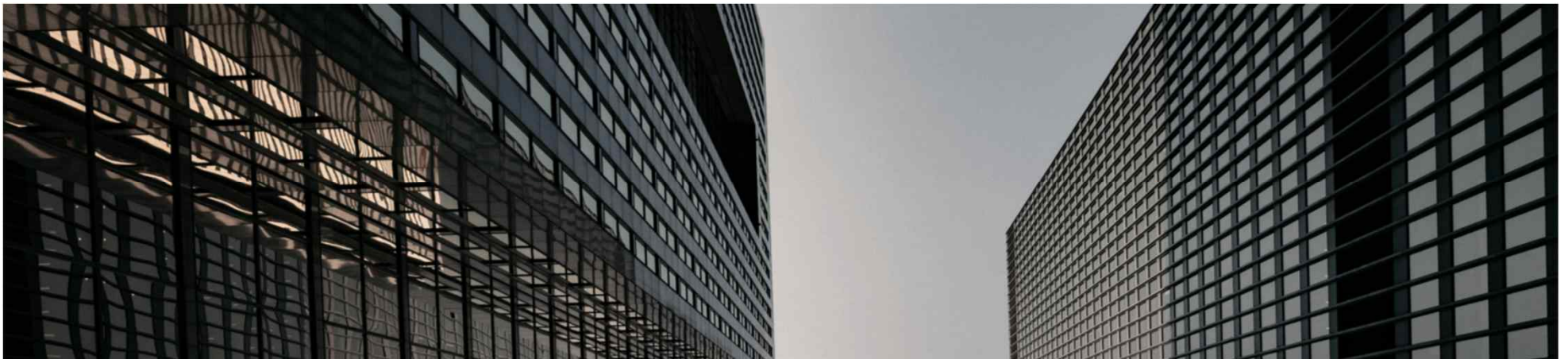
3

결론

- 성능 및 자원 효율성
- 인프라 활용 기대 방안

서론

부산 Edge환경에서의 Function as a Service 기능 개발



Container 기반 클라우드 컴퓨팅

Containers are packages of software that contain all of the necessary elements to run in any environment.

유연성

자신이 원하는 자원을 자유롭게 선택하여 운용할 수 있다.



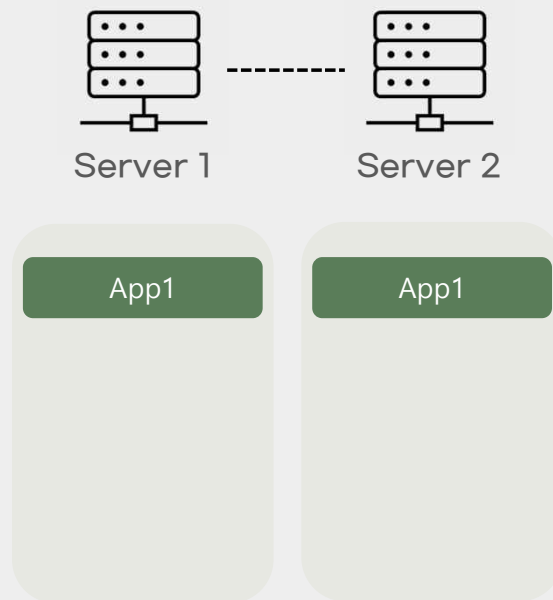
이식성

컨테이너 내에 애플리케이션과 필요한 라이브러리가 모두 존재하여 이식성이 높다.

Serverless

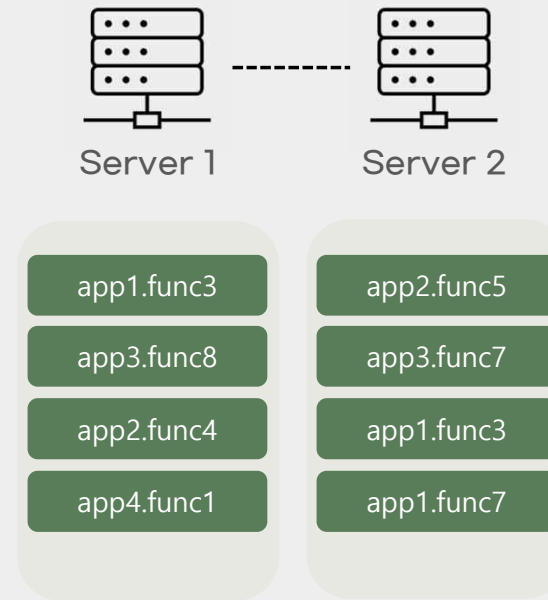
소비자에게 인프라 구축, 관리 필요 없이 개발에 대한 개발환경, 실행 환경, 운영 환경을 제공하는 클라우드 컴퓨팅의 일종

Traditional Architecture



vs

Serverless Architecture



“FaaS (Function-as-a-Services)”

Function as a Service

분산된 대형 인프라 자원에 함수를 등록하고 함수가 실행되는 횟수만큼 비용을 지불하는 방식의 서비스



프로젝트 특징

Kubernetes, KEDA, Grafana

Event-Driven

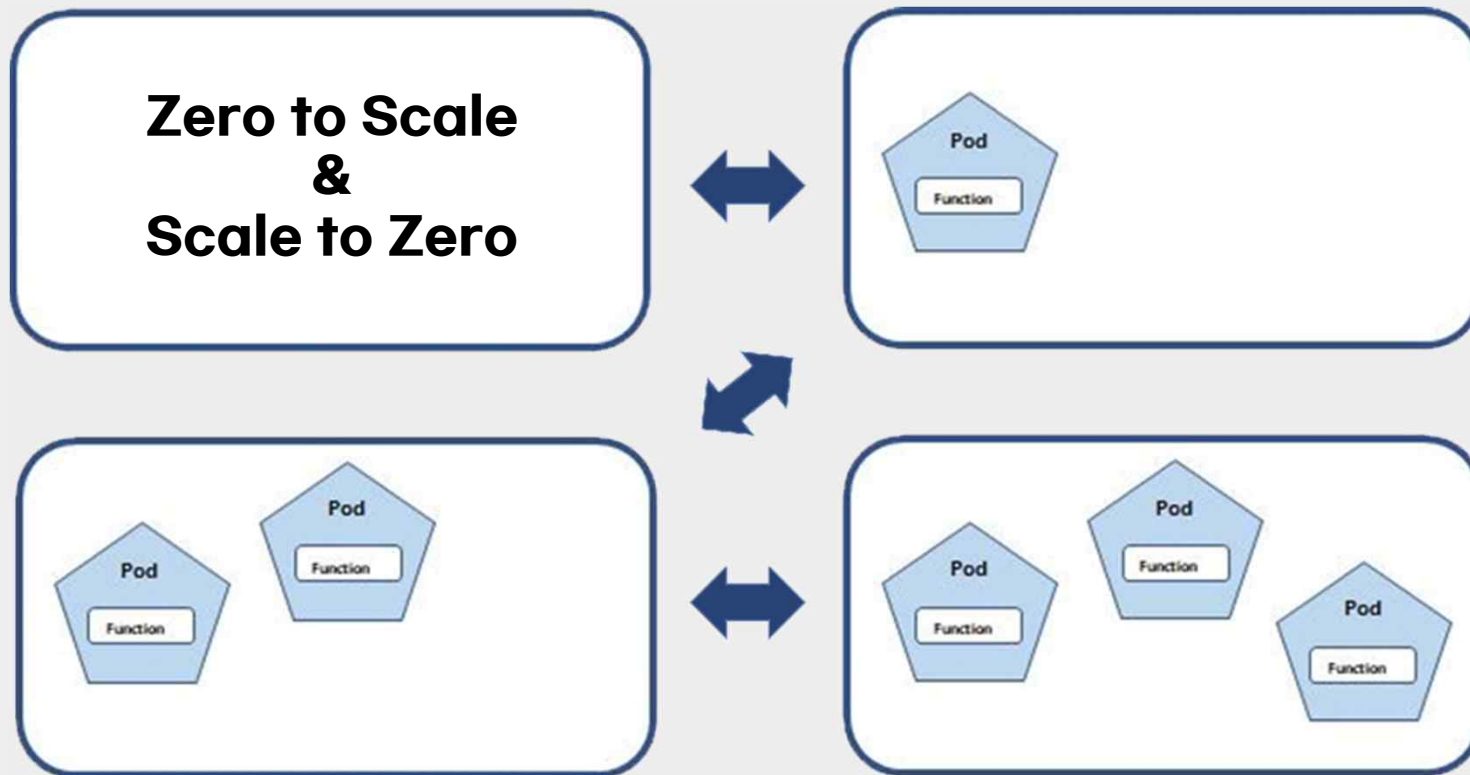
Scale to Zero, Zero to Scale

Grafana를 통한 모니터링

Auto Scaling

Function as a Service

소비자에게 인프라 구축, 관리 필요 없이 개발에 대한 개발환경, 실행 환경, 운영 환경을 제공하는 클라우드 컴퓨팅의 일종



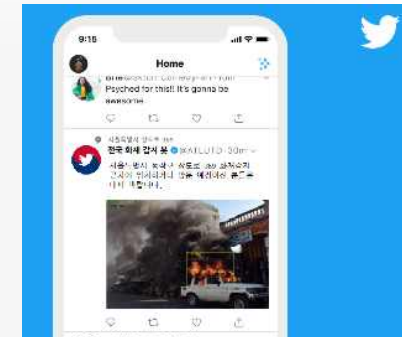
서비스 개요



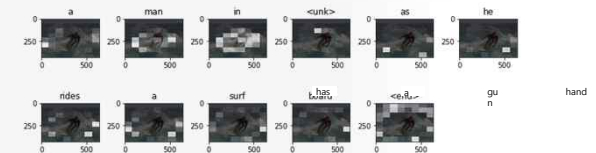
HPC에 딥러닝 Pod 배포



실시간 사고 위치
표시 서비스



전국 화재 발생 알림
SNS 서비스



용의자 식별에 유리한
Image Captioning 서비스

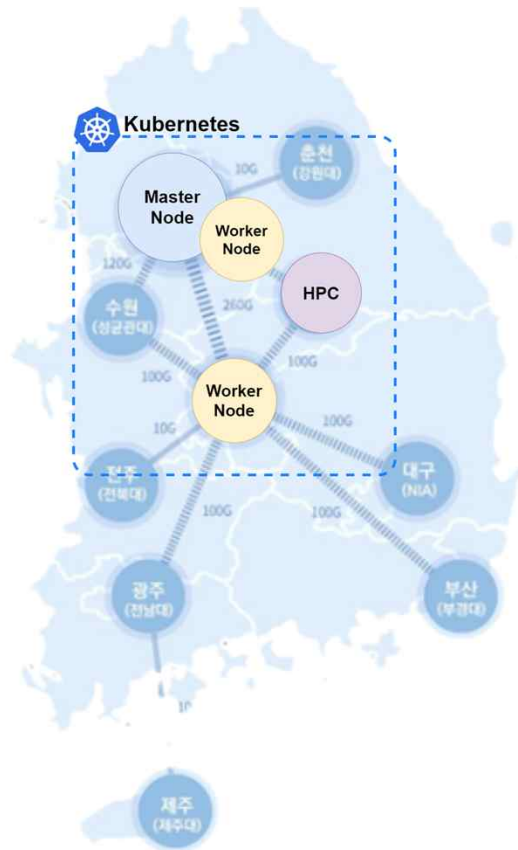
딥러닝을 통한 위험 상황 감지
(이벤트 발생)

본론

부산 Edge환경에서의 Function as a Service 기능 개발



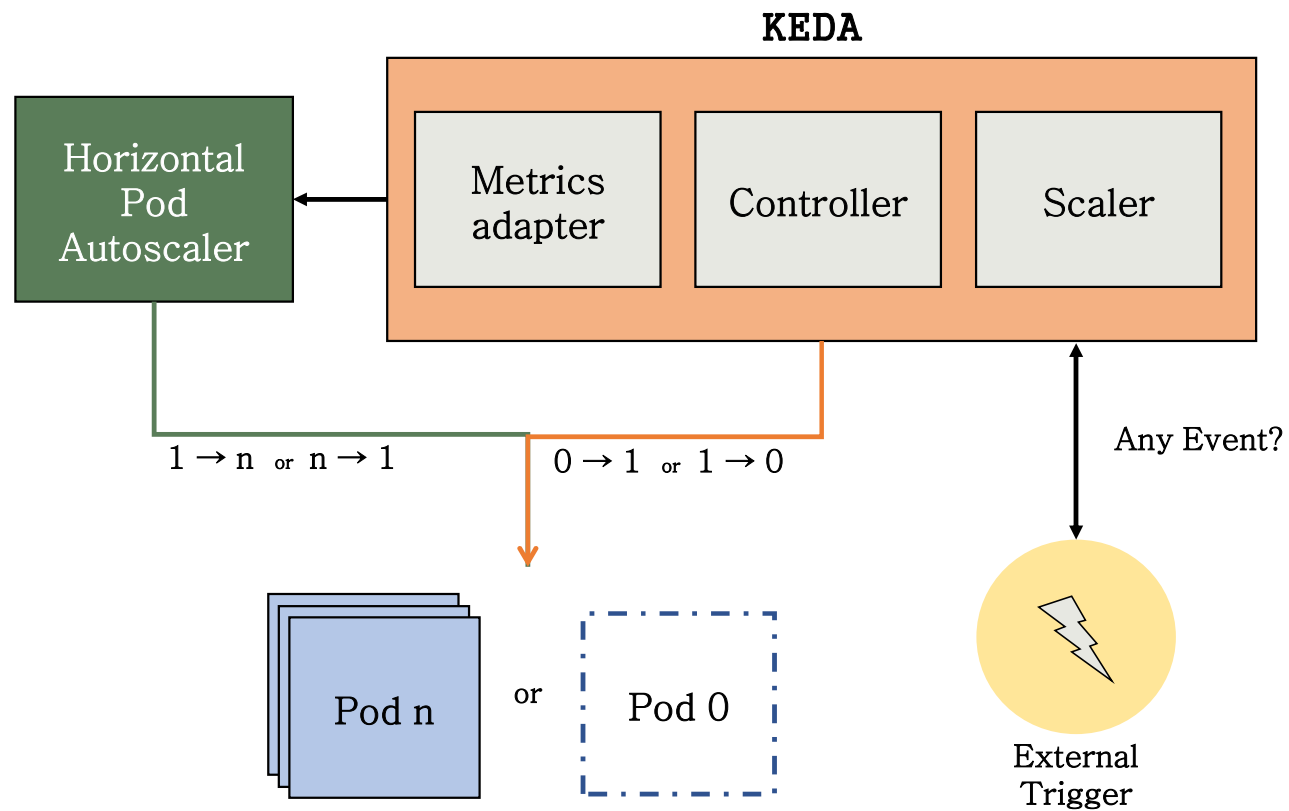
인프라 구축 - 클러스터 구성



Kubernetes 클러스터화

- Master Node - 서울
- Worker Node - 서울, 대전, 성남(HPC)

인프라 구축 - KEDA

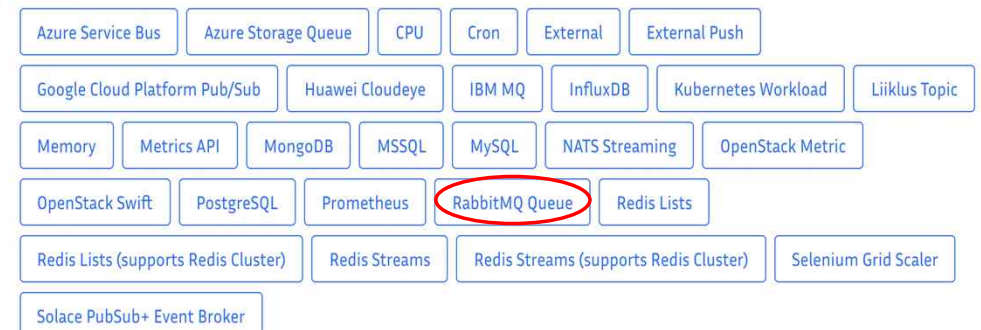


KEDA

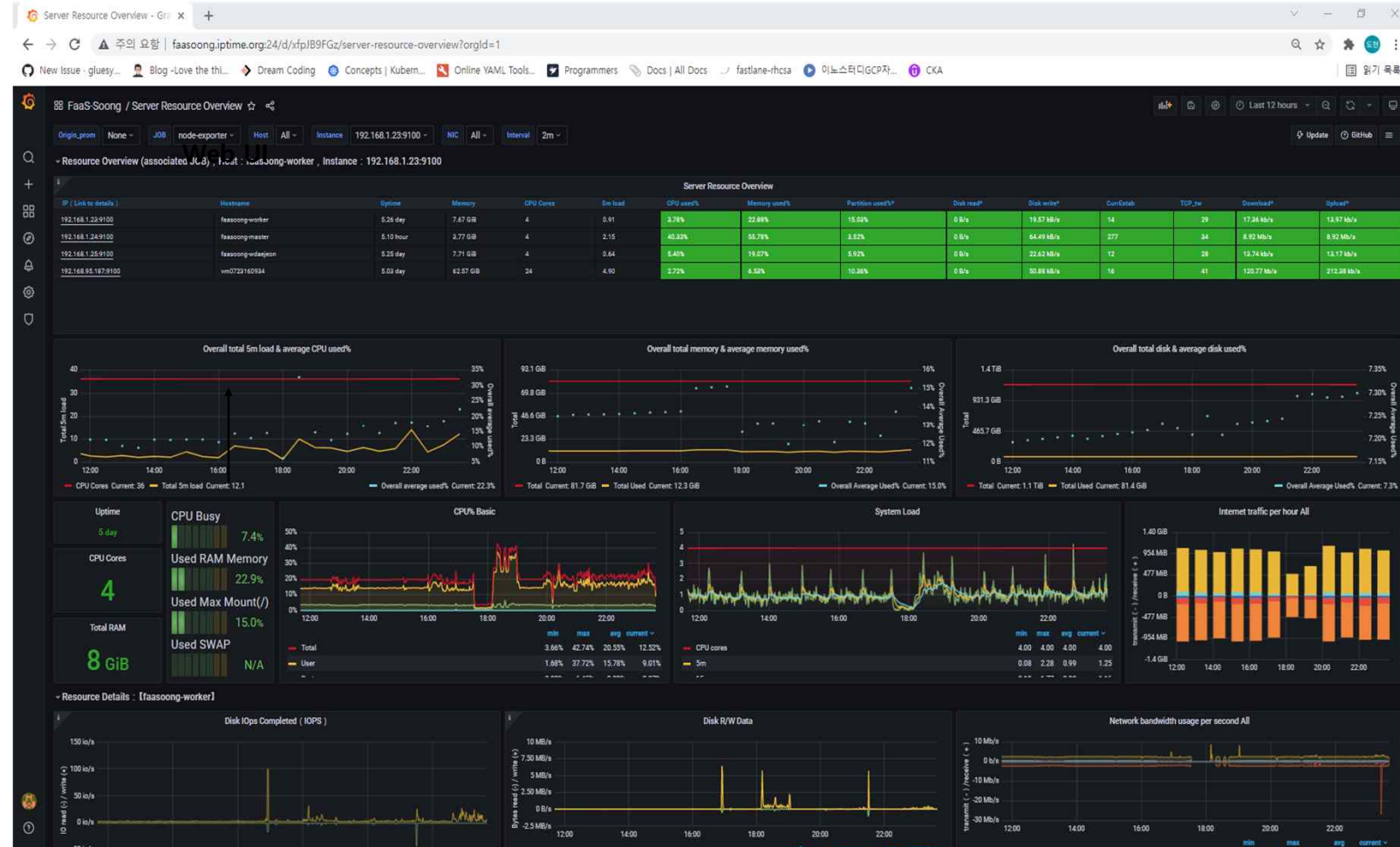
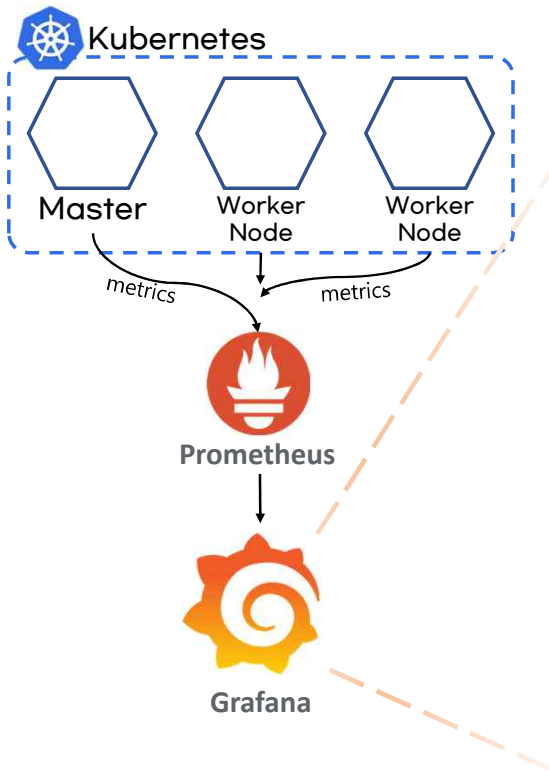
Scaled-Object 정의

```
apiVersion: keda.sh/v1alpha1
kind: ScaledObject
metadata:
  name: darknet-detail-detection-scaling
  namespace: detail-detection
spec:
  scaleTargetRef:
    apiVersion: apps/v1 # Optional. Default: apps/v1
    kind: Deployment    # Optional. Default: Deployment
    name: darknet-detail-detection-model # Mandatory. Must be in the same namespace as the ScaledObject
    envSourceContainerName: darknet-detail-detection # Optional. Default: .spec.template.spec.containers[0]
  pollingInterval: 10 # Optional. Default: 30 seconds
  cooldownPeriod: 100 # Optional. Default: 300 seconds 후에 다시 zero상태로 돌아가는데 까지 걸리는 시간
  minReplicaCount: 0 # Optional. Default: 0
  maxReplicaCount: 1 # Optional. Default: 100
  advanced:
    restoreToOriginalReplicaCount: false # Optional. Default: false
    horizontalPodAutoscalerConfig:
      behavior: # Optional. Section to specify HPA related options
        # Optional. Use to modify HPA's scaling behavior
      scaleDown:
        stabilizationWindowSeconds: 10 # 스케일 다운하는데 걸리는 시간.
      policies:
        - type: Percent
          value: 100
          periodSeconds: 10
  triggers:
    - type: rabbitmq
      metadata:
        host: amqp://faasoong:tnd@faasoong.iptime.org:5672/
        protocol: amqp
        mode: QueueLength
        value: "500"
```

Scaler - RabbitMQ

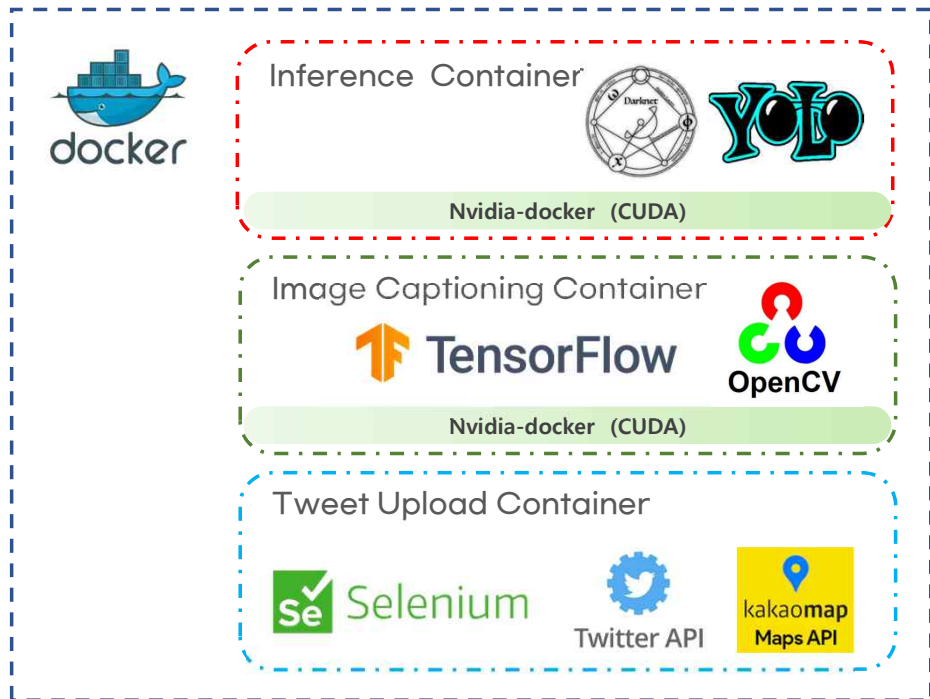


Prometheus, Grafana

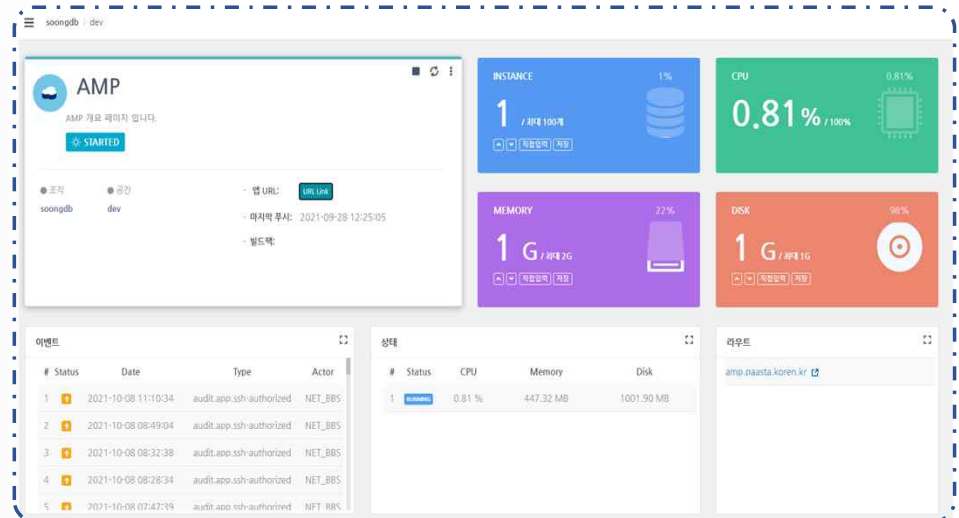


Docker Image

Service Containers



AMP ( **APACHE** /  **MySQL** /  **php**)

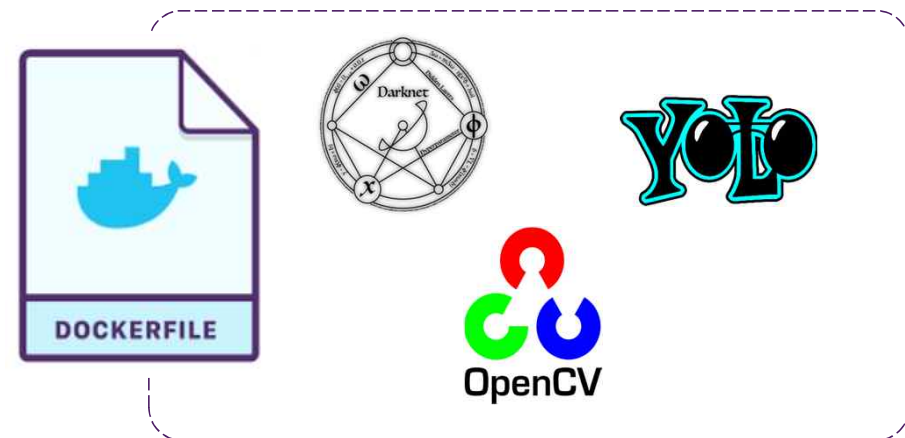
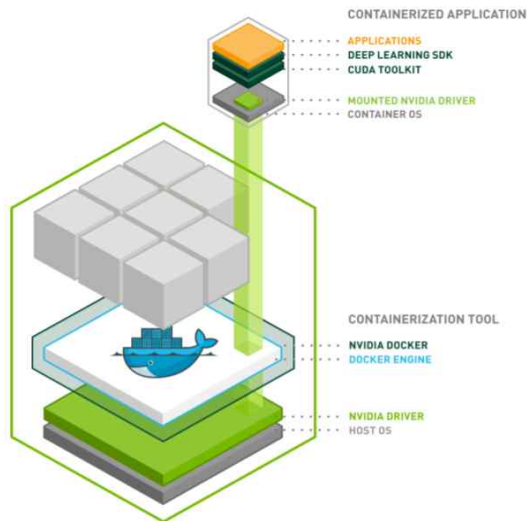


CLOUD  **FOUND** **RY**

Docker Image

Darknet Framework Yolov4

NVIDIA-DOCKER
github.com/NVIDIA/nvidia-docker



ryunchang/darknet:5.1

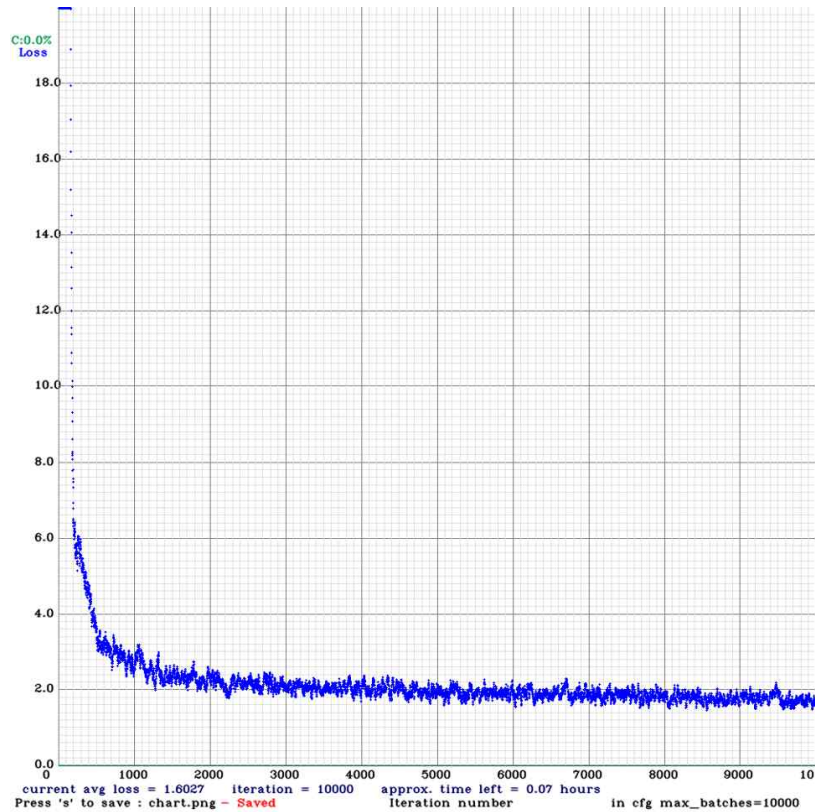
DIGEST: sha256:8d7049d9da9a72754a5f623fc1c32bdbc94bc140ac5860dadab66912a230f548

OS/ARCH
linux/amd64

COMPRESSED SIZE
3.53 GB

LAST PUSHED
4 days ago by [ryunchang](#)

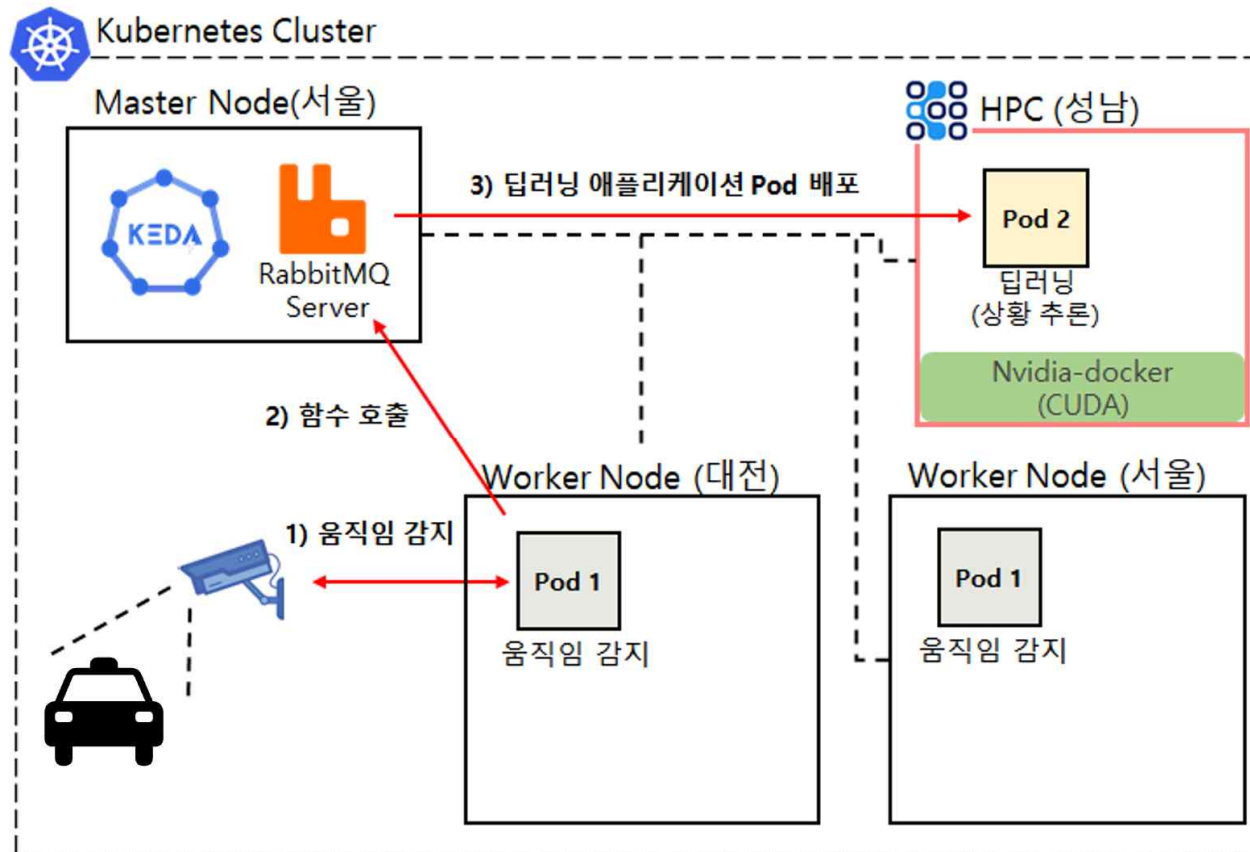
Darknet Framework



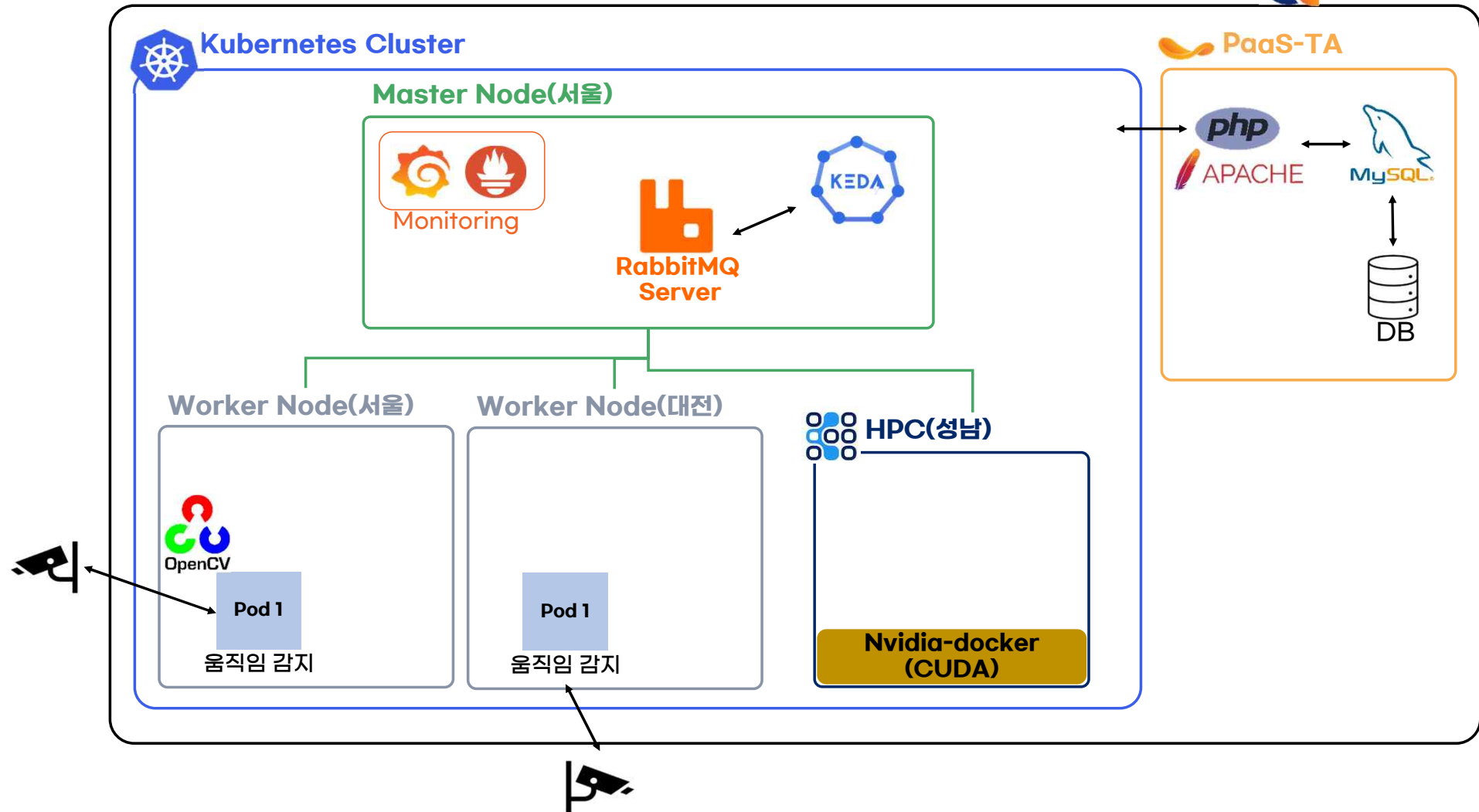
- 총 4개 class (교통사고, 화재, 총, 칼)
- 5000여장의 Training Dataset 생성
- 약 11시간의 training 후 weight 생성

서비스 흐름도

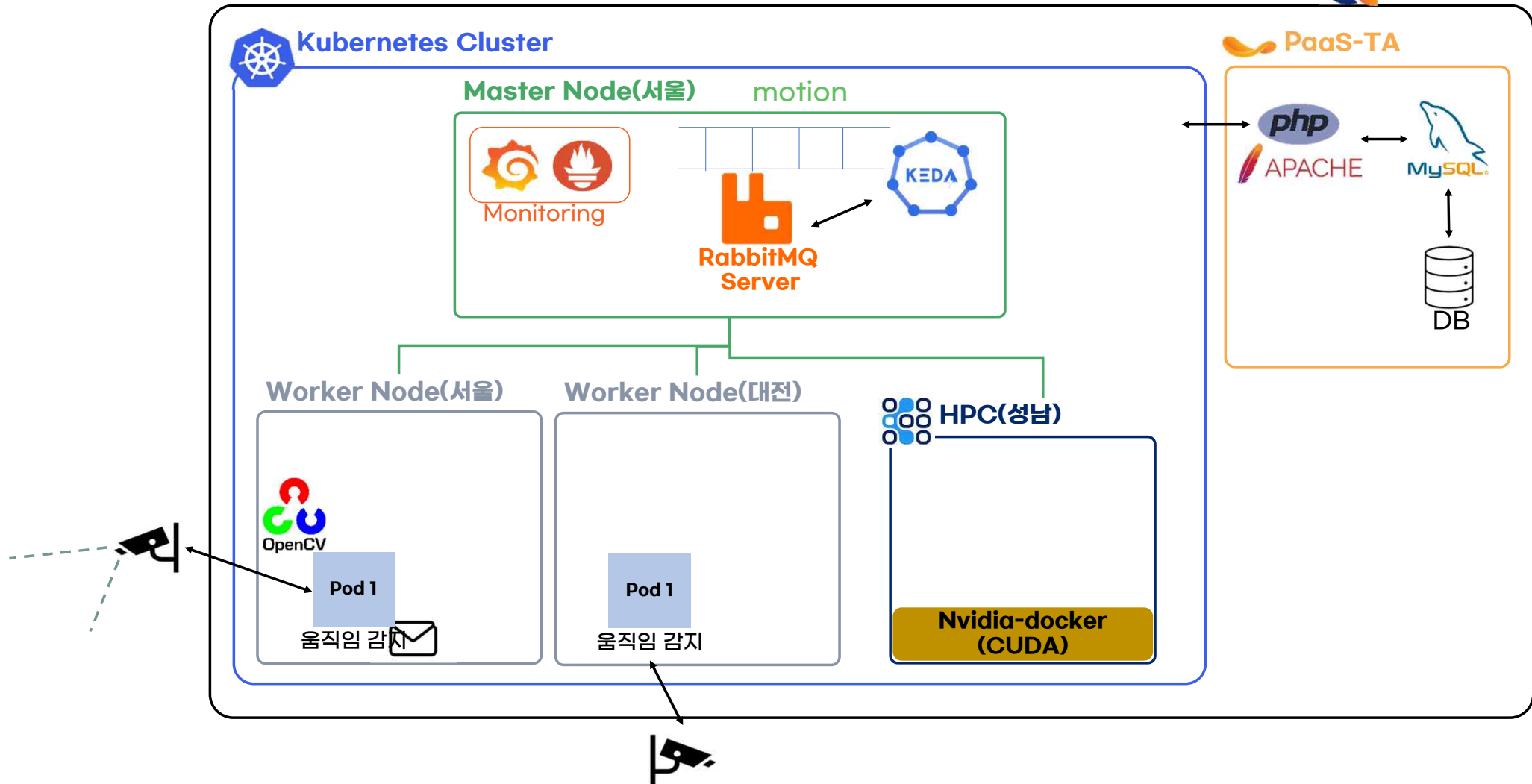
엣지 클라우드를 활용한 연쇄적인 컴퓨팅 워크플로우를 통해 자원 활용성 극대화



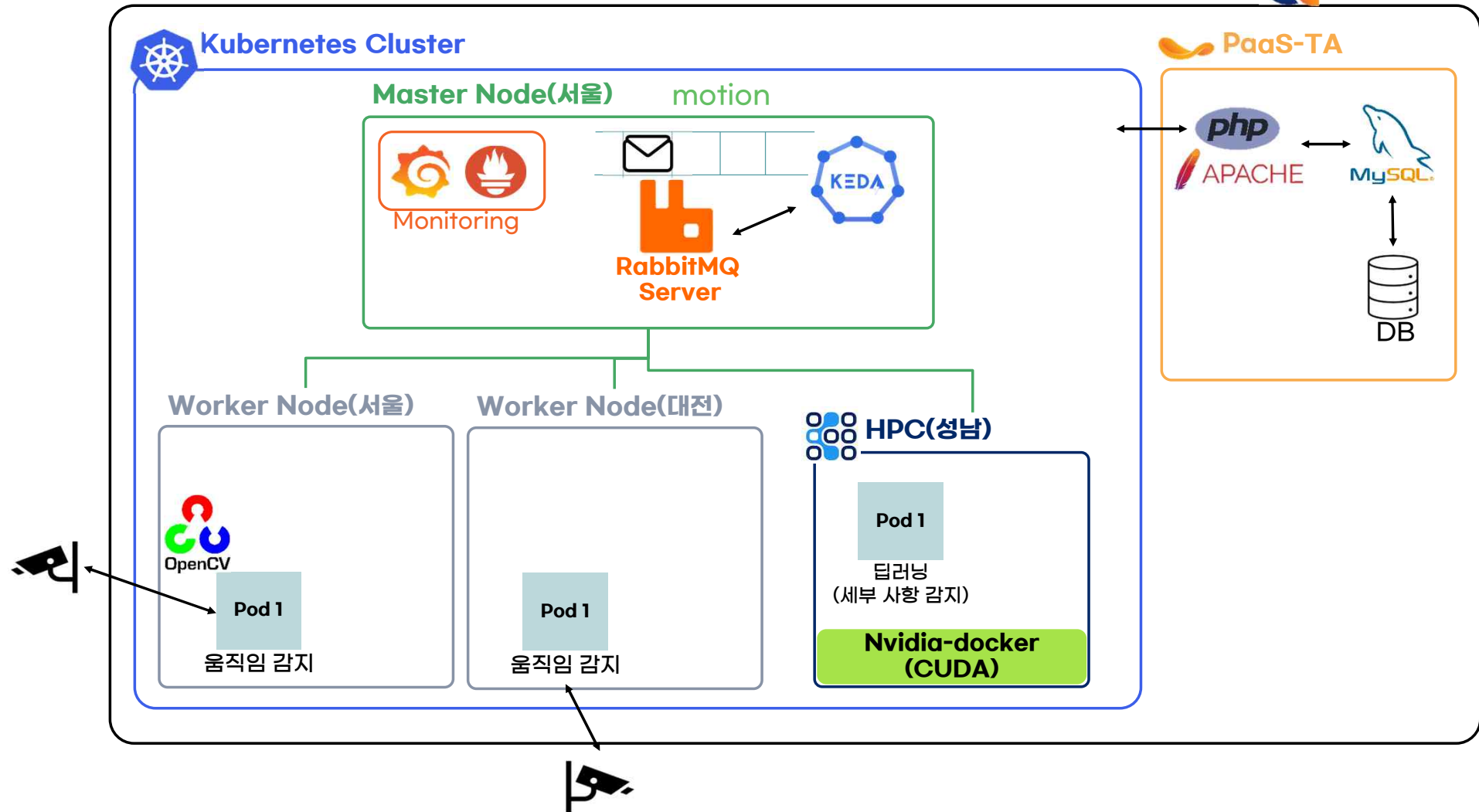
(1) 워커 노드에서 CCTV 영상을 수신하면서 움직임 감지 Pod 항시 동작



(2) 움직임 감지 후 RabbitMQ의 Motion 큐에 메시지 전송



(3) KEDA가 RabbitMQ 메시지를 받아 HPC에 세부 상황 감지 Pod 배포

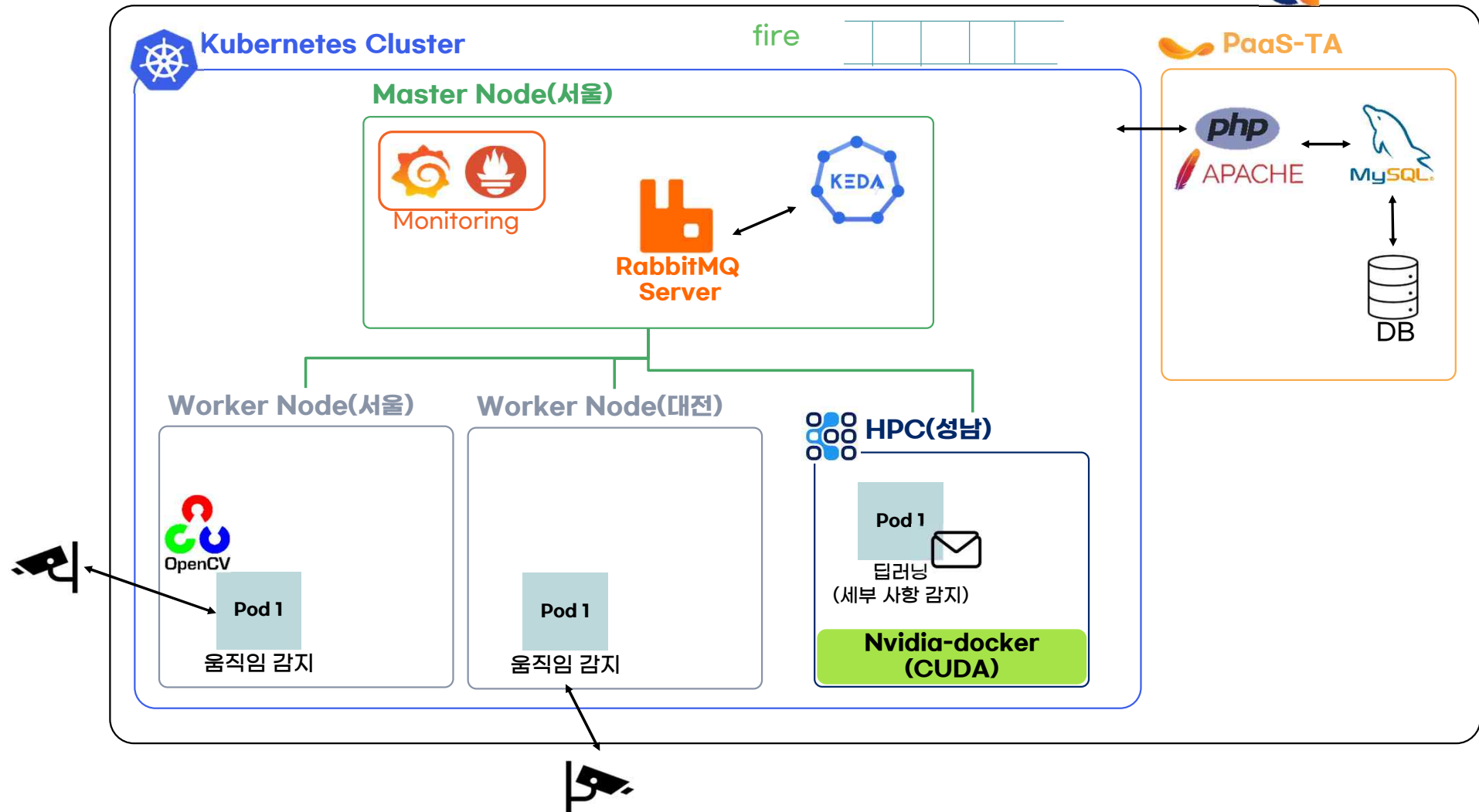


(4) 세부 상황 감지 후 RabbitMQ의 해당 큐에 메시지 전송

accident

knife

fire



(5) 사고 데이터를 DB에 저장 후 실시간 위험 상황 알림 서비스 배포

accident

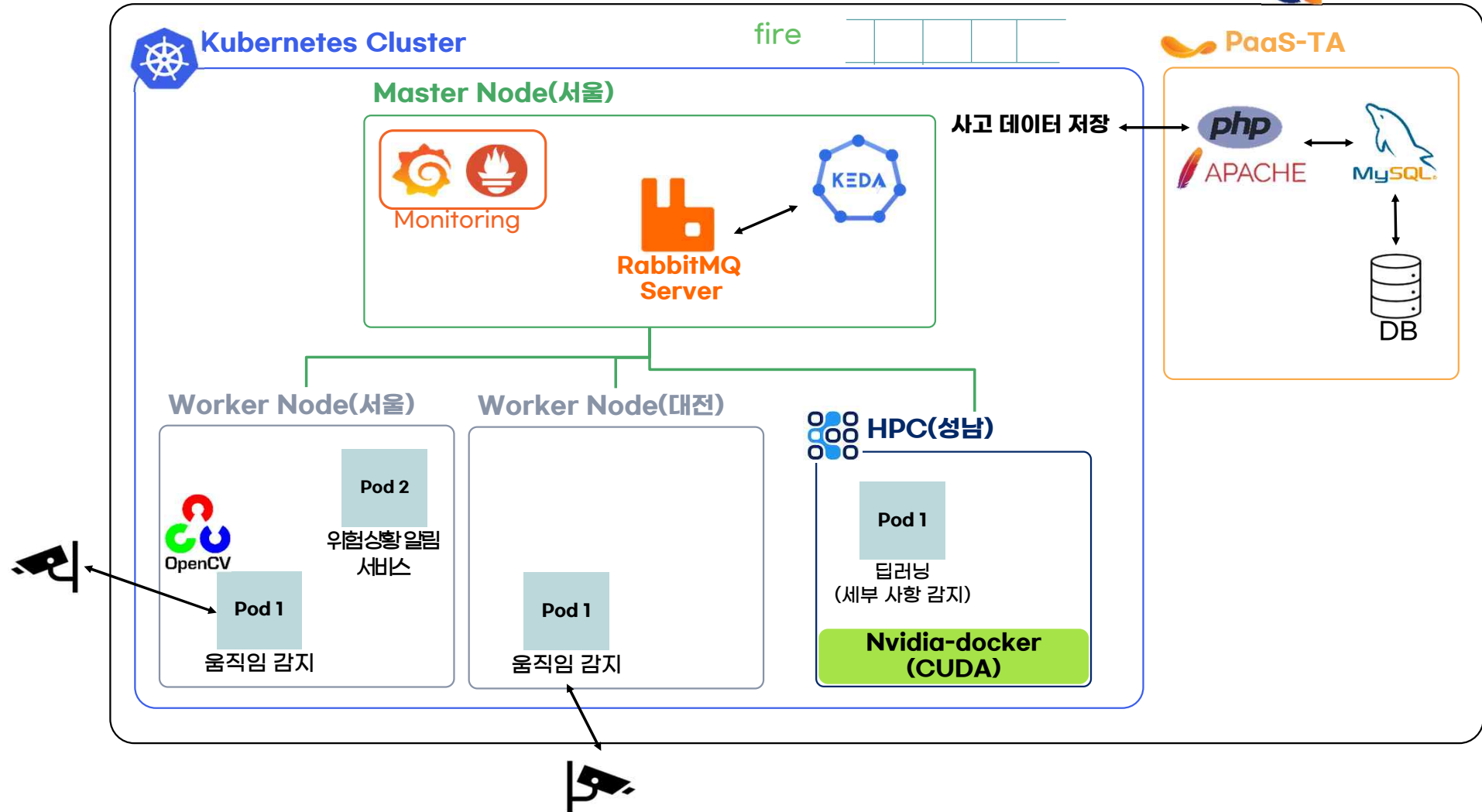
--	--	--	--

knife

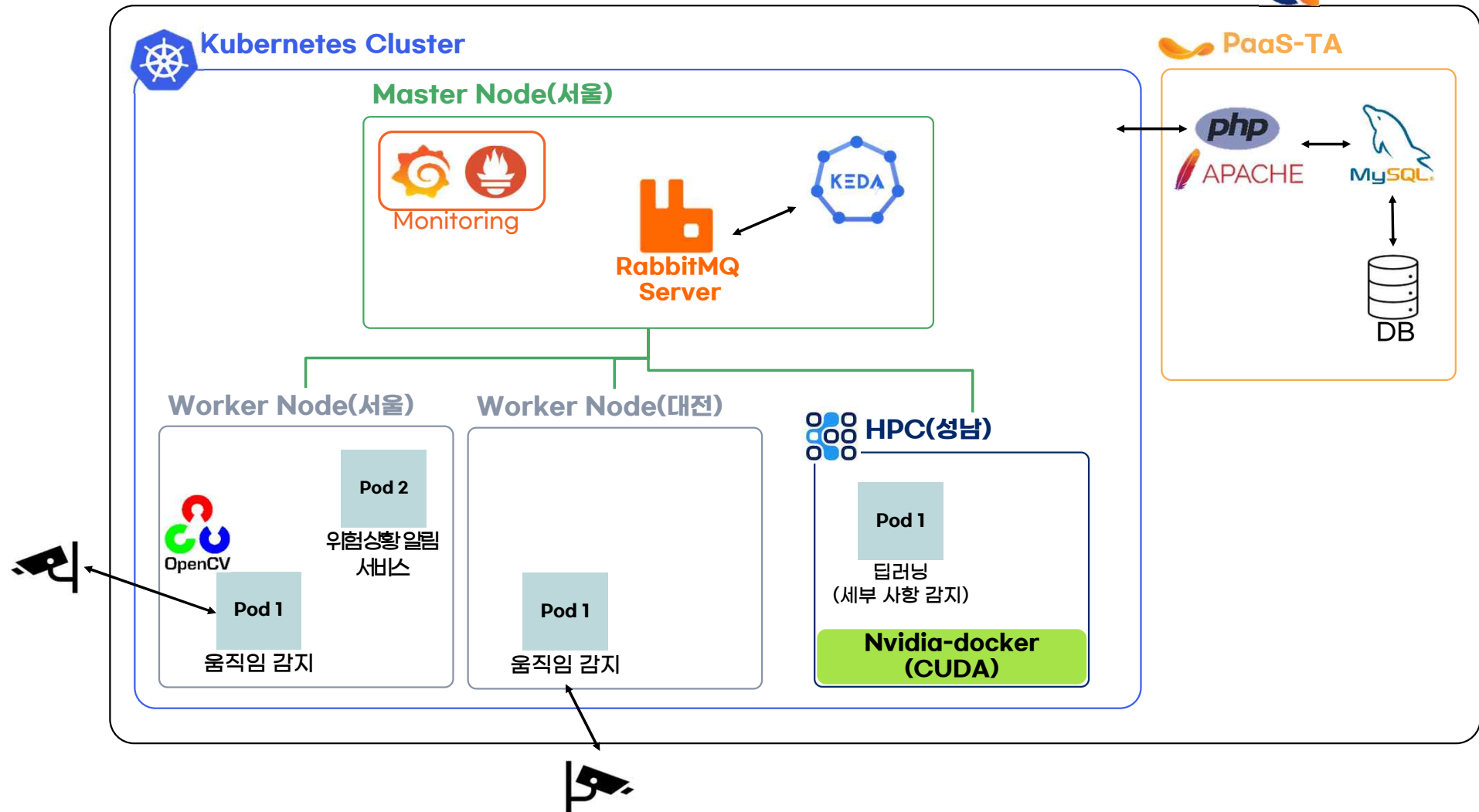
--	--	--	--

fire

--	--	--	--



(6) 서비스 종료 후 포드 소멸

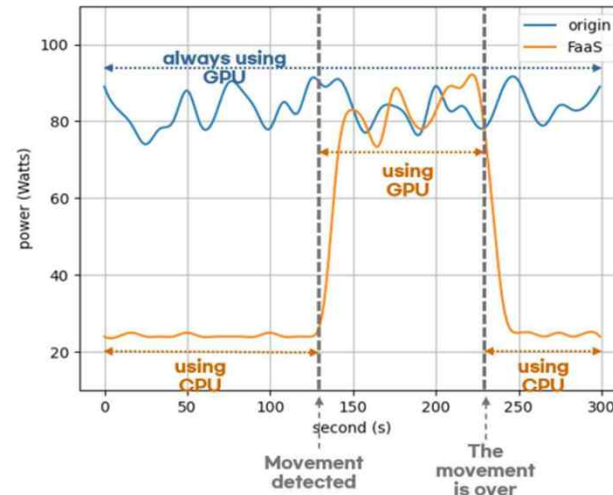
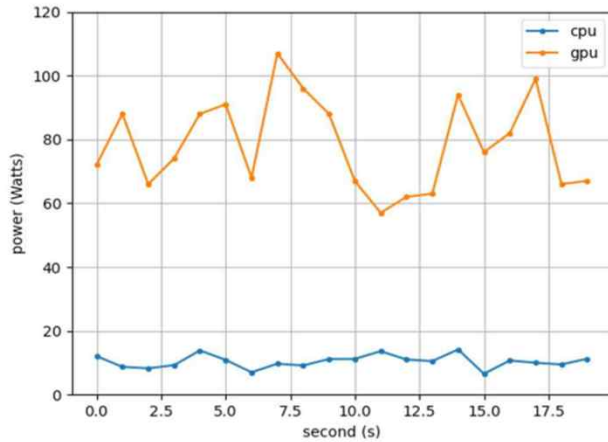


결론

부산 Edge환경에서의 Function as a Service 기능 개발



자원 절약 성능에 관한 연구



FaaS 인프라 적용 전후 소비전력량 비교

	세부 상황 판단 딥러닝 애플리케이션 (NVIDIA Tesla V100)	움직임 감지 애플리케이션 (CPU + RAM)
시작 전 메모리 사용량	239 MiB/ 16160 MiB	5846 MiB/ 167954 MiB
시작 후 메모리 사용량	1237 MiB/ 16160 MiB	5986 MiB/ 167954 MiB
평균 사용량	998 MiB	140 MiB

딥러닝 애플리케이션과 움직임 감지 애플리케이션의 메모리 사용량 비교

자원 절약 성능에 관한 연구



쿠버네티스 컨테이너 이미지로 개발

The screenshot displays the Docker Hub interface with three repository cards overlaid. The background shows the 'dohyunkim12' profile page with tabs for General, Tags, Builds, Collaborators, Webhooks, and Settings. An 'Advanced Image Management' section is visible at the top left.

Repository 1: dohyunkim12/amp
By dohyunkim12 • Updated a month ago
Container
Overview | Tags

Repository 2: ryunchang/darknet
By ryunchang • Updated a day ago
Container
Overview | Tags

Repository 3: dohyunkim12/tweet_upload
By dohyunkim12 • Updated a day ago
Container
Overview | Tags

Tags and Scans for dohyunkim12/amp:

- 8.0-with-clear
- 7.0
- 6.0
- 5.0
- 4.0

Tags and Scans for ryunchang/darknet:

- 8.0-fire (Last pushed a day ago by ryunchang)
- 8.0-rtsp (Last pushed a day ago by ryunchang)
- 8.0-accident (Last pushed 5 days ago by ryunchang)

Tags and Scans for dohyunkim12/tweet_upload:

- test (Last pushed 4 days ago by dohyunkim12)
- 1.0 (Last pushed 4 days ago by dohyunkim12)

Instructions for dohyunkim12/amp:

- root로 변경 `su -`
- apache2 start
`service apache2 start`
- (mysql은 docker commit으로 반영안됨)
- plugin 방식 확인
`mysql > SELECT User, Host, plugin FROM mysql.user;`
- plugin 변경
`mysql> alter user root@localhost identified with mysql;`
`mysql> flush privileges;`
`mysql> select user, host, plugin from user;`

Tags and Scans for dohyunkim12/tweet_upload:

TAG	DIGEST	OS/ARCH	COMPRESSED SIZE
test	b9fb2723e8de	linux/amd64	414.75 MB
1.0	449d9e0a6218	linux/amd64	415.21 MB

인프라를 통한 차후 이익 모델

Scale-to-Zero를 통한 자원 활용성 극대화

Auto-Scaling을 통한 안정적인 트래픽 관리

인프라 구축에 대한 부담 절감

기능 개발 및 수정에 용이

통합 모니터링

Self-Healing을 통한 운용 용이성

감사합니다