Updated:
Nov, 2020

쿠버네티스환경에서 머신러닝파이프라인을 통한 개인화추천시스템 구축

메가존 **AILab** 팀장 손영제

Google Cloud

Cloud Innovator MEGAZONE

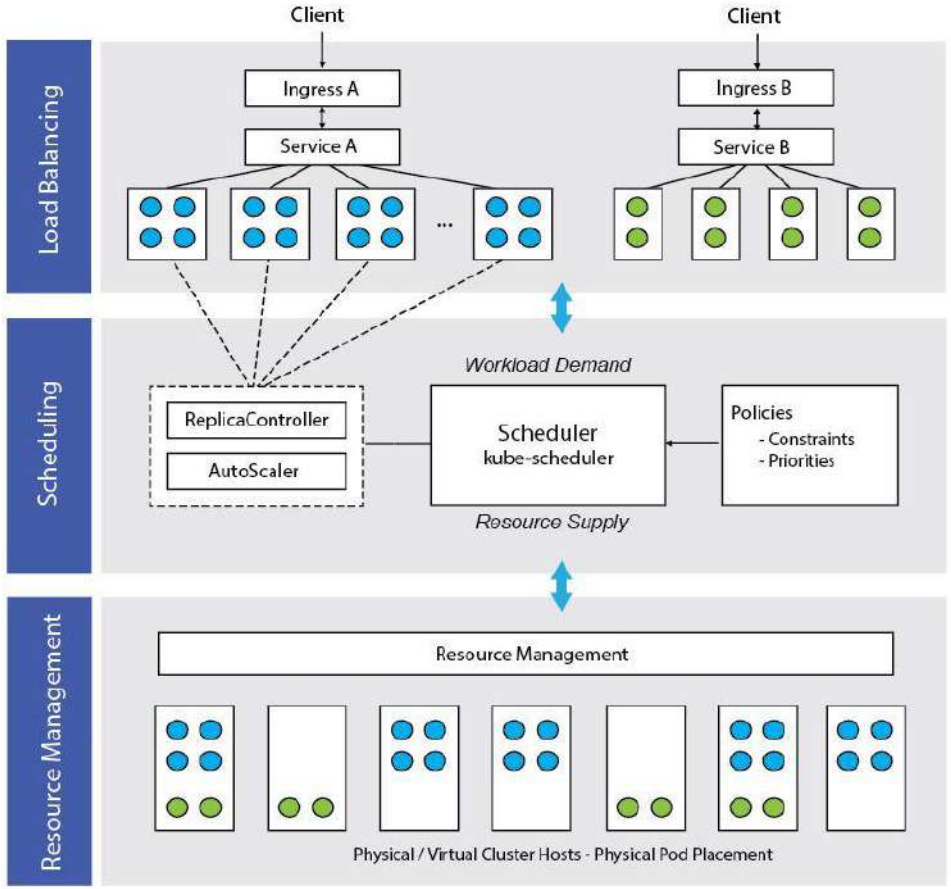
Index

Google Cloud

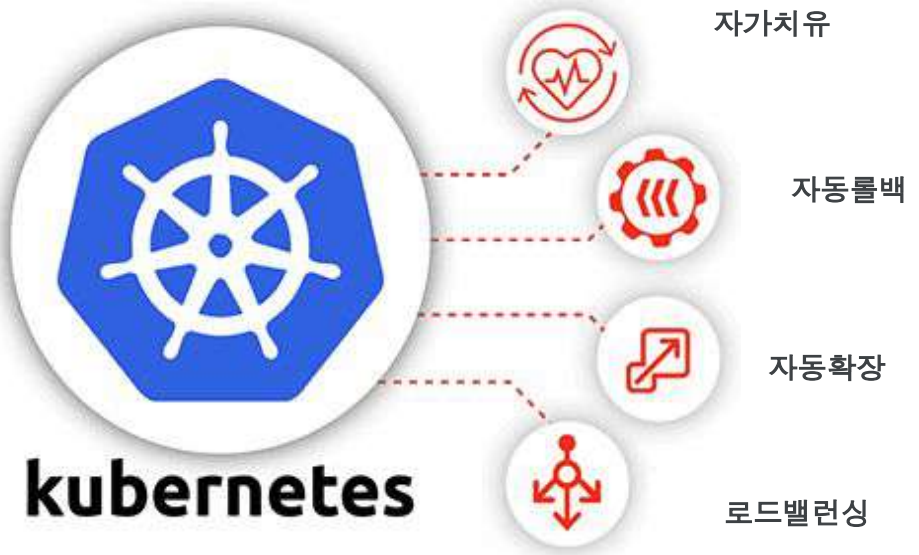
Cloud Innovator MEGAZONE

쿠버네티스:

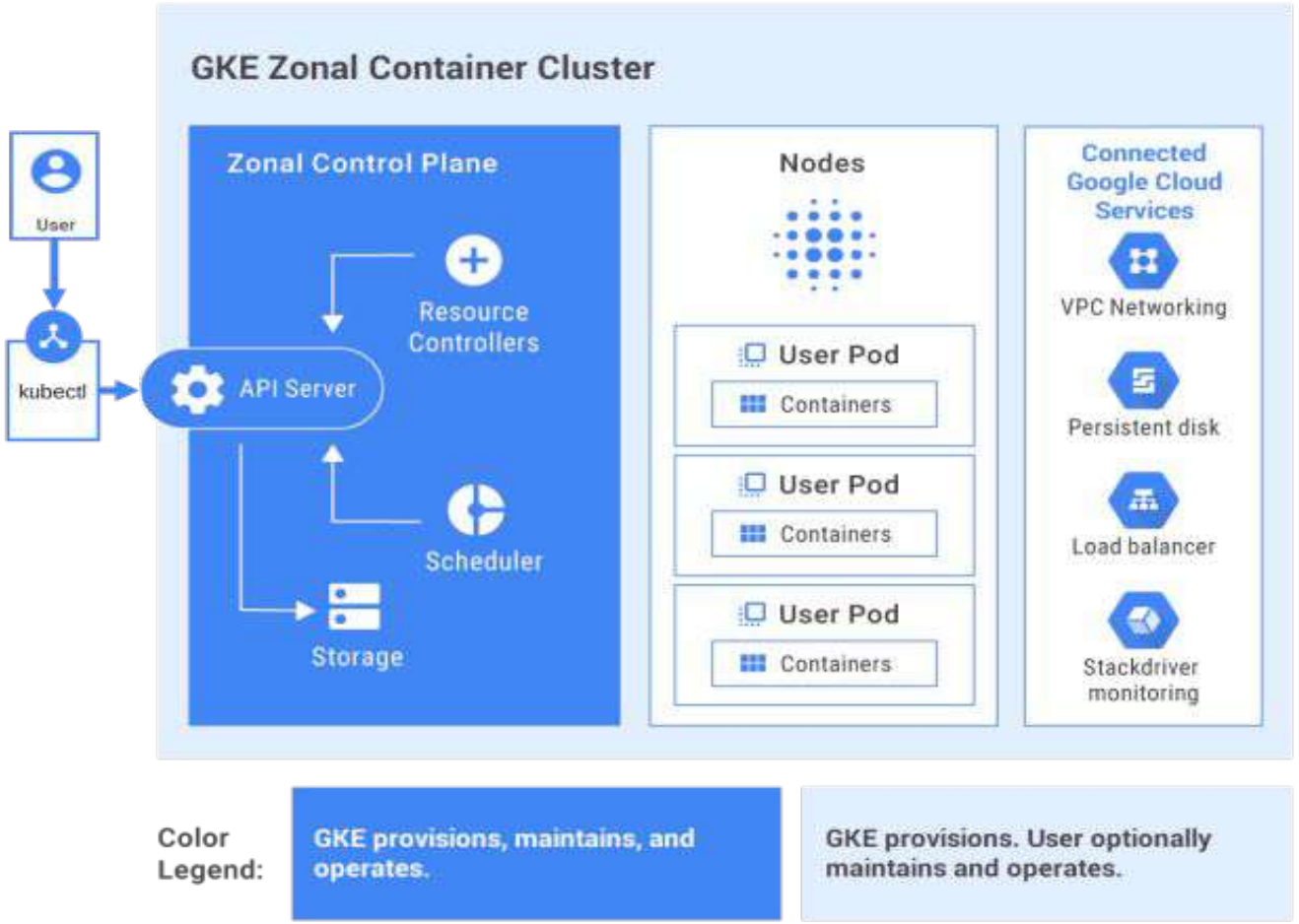
컨테이너화 된 애플리케이션 및
서비스를 제어하고 관리하는 플랫폼;
애플리케이션을 일관성 있게
만들고 요청을 받을 수 있도록 해준다.



쿠버네티스 특징:



쿠버네티스 아키텍처 (GKE)



Google Cloud

Cloud Innovator MEGAZONE

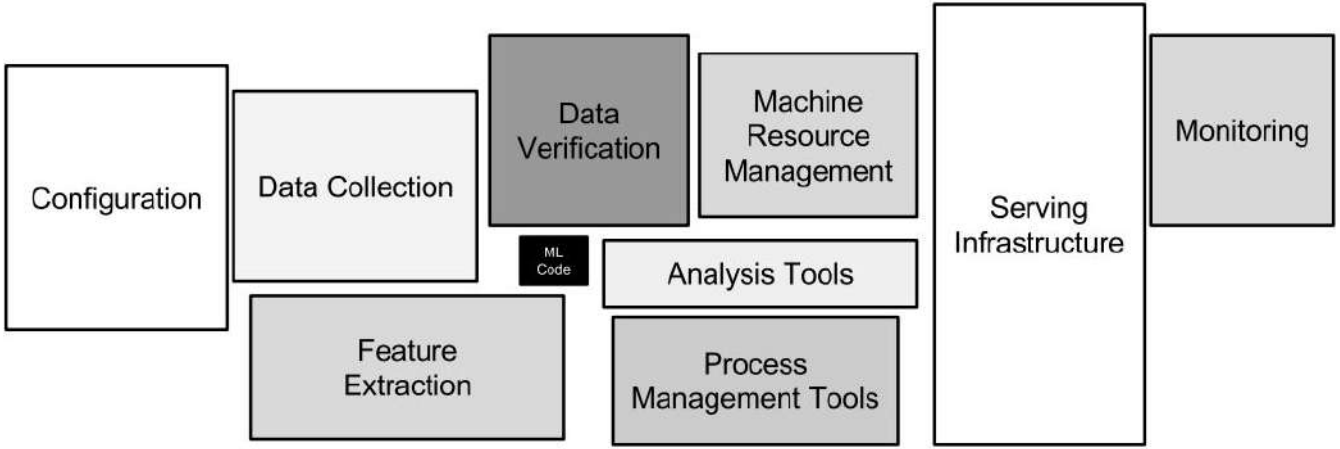
쿠버네티스 + 빅데이터/머신러닝 장점

클라우드나 On-Prem (데이타 센터), 개인 개발 환경에 상관 없이 동일한 머신러닝 플랫폼을 손쉽게 만들 수 있기 때문에 특정 벤더나 플랫폼에 종속되지 않는다

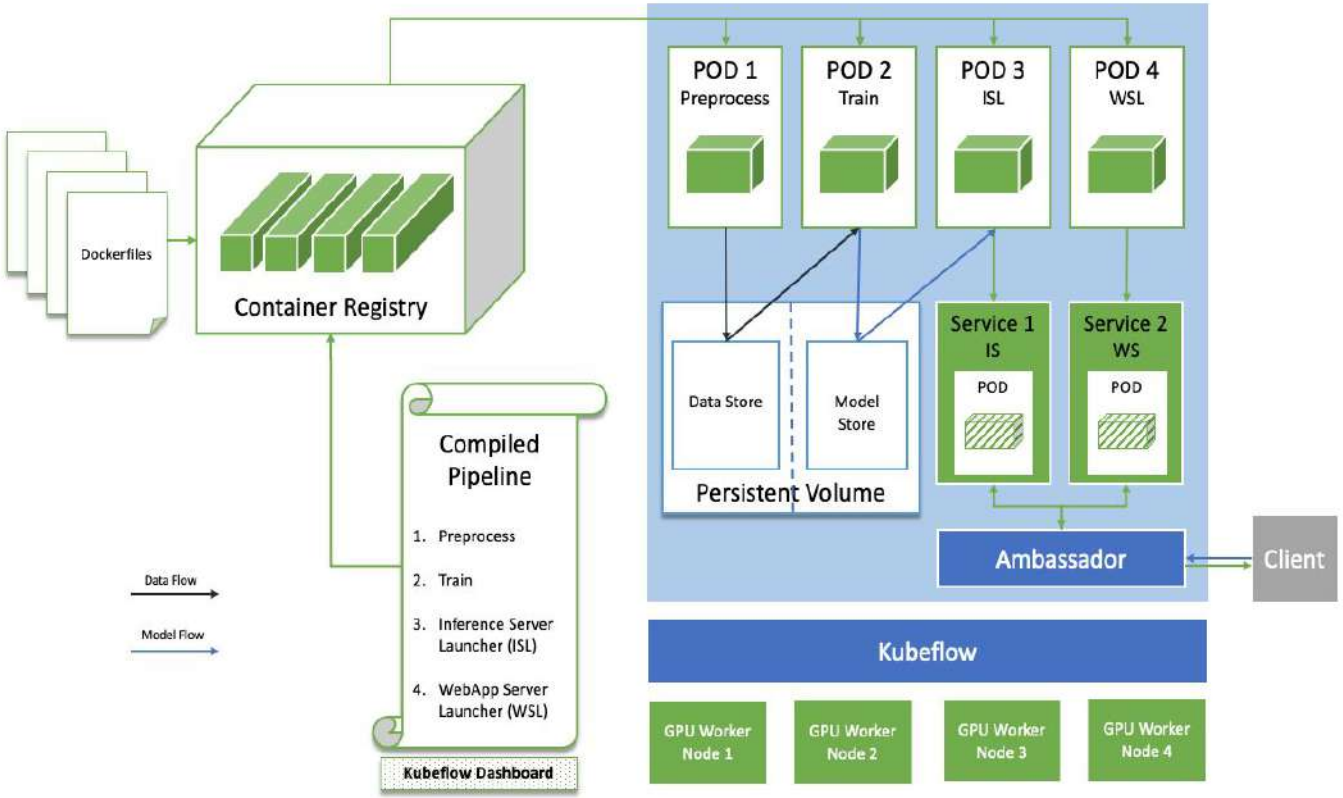
컨테이너 기술을 이용해서 필요한 경우에만 컨테이너를 생성해서 사용하고, 사용이 끝나면 컨테이너를 삭제하는 방식이기 때문에 자원 활용율이 매우 높다.

컨테이너로 패키징이 되어있기 때문에 내부 구조를 알 필요가 없이 단순하게 컨테이너만 배포하면 된다.

머신러닝 파이프라인


Configuration
Data Collection
Data Verification
Machine Resource Management
Serving Infrastructure
Monitoring
ML Code
Analysis Tools
Feature Extraction
Process Management Tools

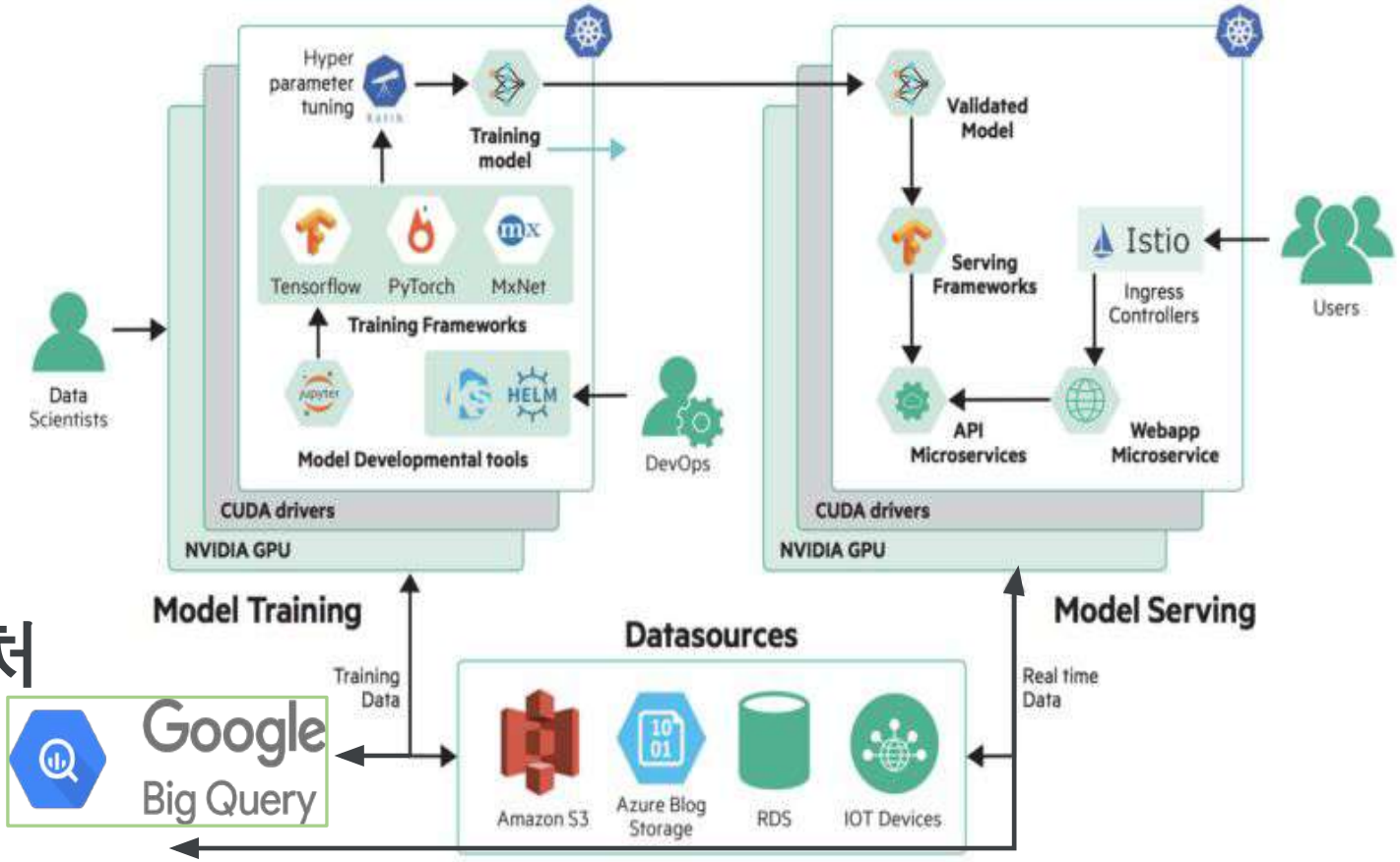
쿠버네티스 머신러닝 아키텍처



Google Cloud

Cloud Innovator MEGAZONE

머신러닝 솔루션 아키텍처

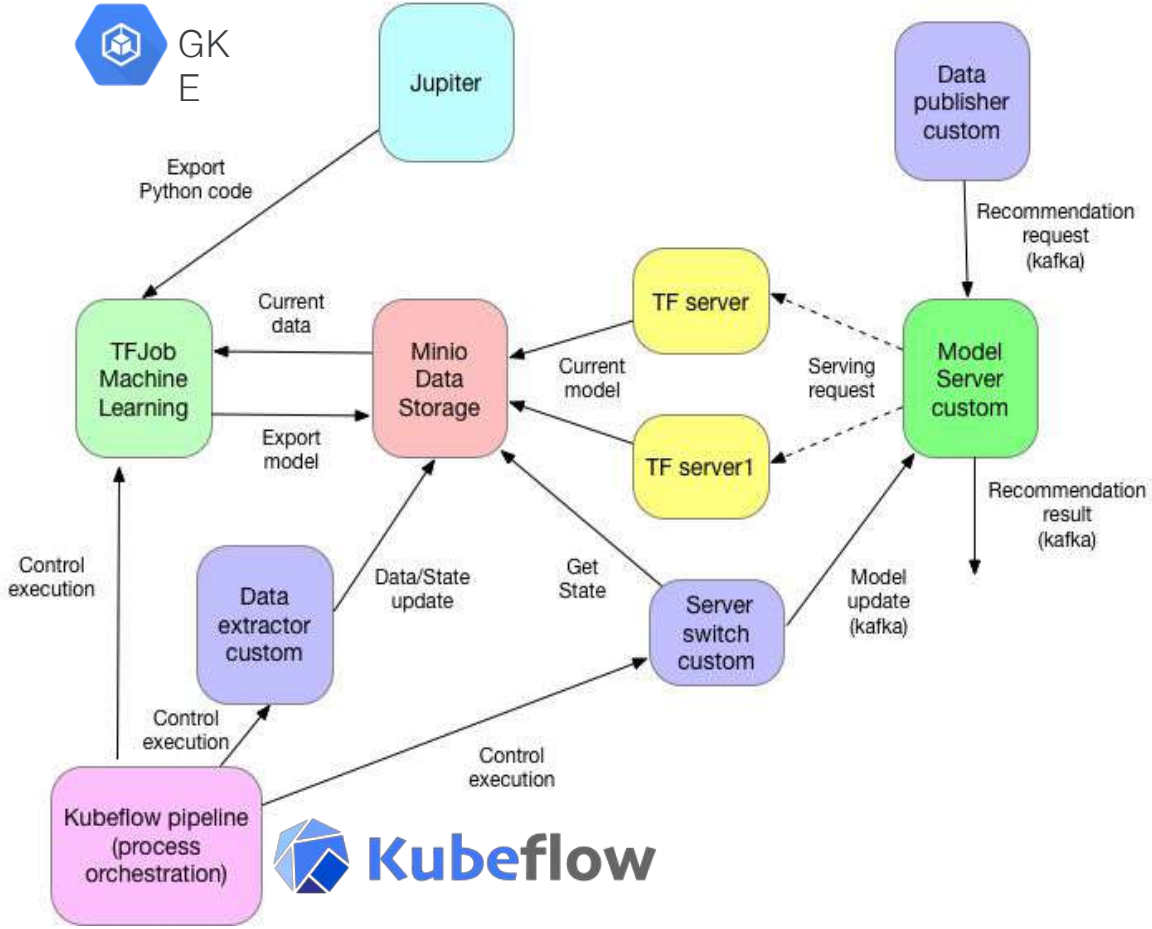


Google Cloud

Cloud Innovator MEGAZONE

데모: 개인화추천

사용자 선호도에 따르는 매트릭스 분해
(협업 필터링) 기반 추천시스템;
온라인 상점에서 제품을 본 시간을
기준으로 사용자에게 추천 할
다른 제품을 식별하는 방식



Google Cloud

Cloud Innovator MEGAZONE

데모 과정

학습 데이터 준비 : 실제 이커머스 쇼핑몰에서 발생한 사용자 로그와 상품데이터 (userId , itemId , rating 으로 labeling)

모델학습 및 평가 : 딥러닝모델 선택, 하이퍼파라미터 튜닝, 모델 학습, 모델 테스트 및 평가

모델배포 : 모델 운영환경에 배포

모델서비스 : 배포된 딥러닝 모델로 개인화 추천 서비스

1 학습 데이터 준비

2 모델학습 및 평가

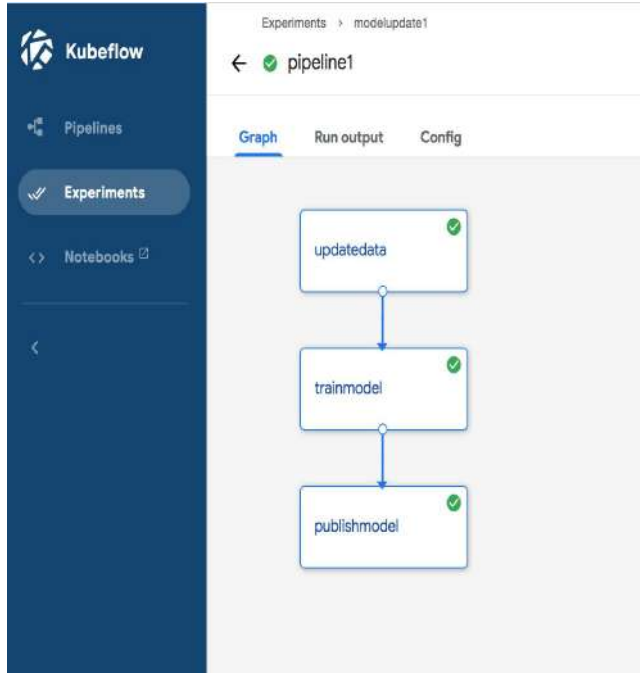
3 모델배포

4 모델서비스

Google Cloud

Cloud Innovator MEGAZONE

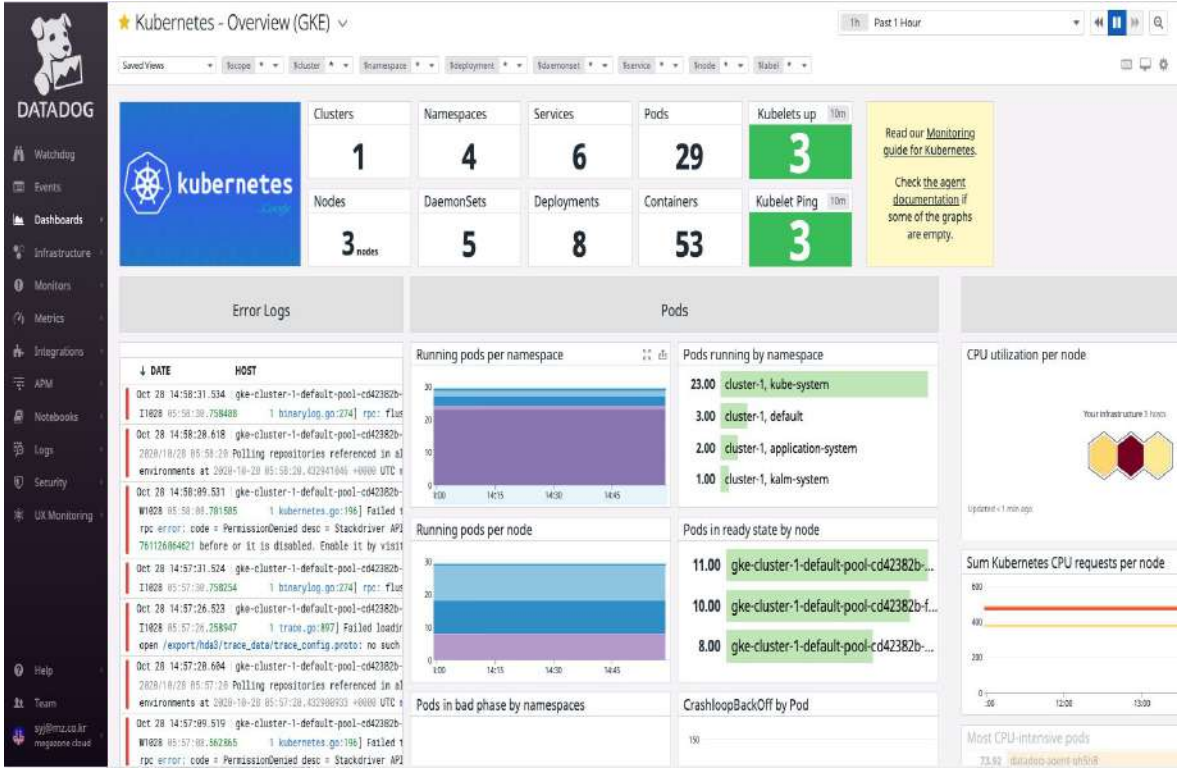
```
@dsl.pipeline(
  name='Recommender model update',
  description='Demonstrate usage of pipelines for multi-step model update'
)
def recommender_pipeline():
    # Load new data
    # kubectl port-forward pods/minio-97967bd9-4tvn5 9000:9000
  data = dsl.ContainerOp(
      name='updatedata',
      output_artifact_paths={
        'mlpipeline-ui-metadata': '/output/mlpipeline-ui-metadata.json',
        'mlpipeline-metrics': '/output/mlpipeline-metrics.json',
      },
      image='lightbend/kubeflow-datapublisher:0.0.1') \
    .add_env_variable(k8s_client.V1EnvVar(name='MINIO_URL',value='http://minio-service.default.svc.cluster.local:9000')) \
    .add_env_variable(k8s_client.V1EnvVar(name='MINIO_KEY', value='minio')) \
    .add_env_variable(k8s_client.V1EnvVar(name='MINIO_SECRET', value='minio123')) \
    .add_volume(k8s_client.V1Volume(name='outputs', empty_dir=k8s_client.V1EmptyDirVolumeSource())) \
    .add_volume_mount(k8s_client.V1VolumeMount(name='outputs', mount_path='/output'))
    # Train the model
  train = dsl.ContainerOp(
      name='trainmodel',
      output_artifact_paths={
        'mlpipeline-ui-metadata': '/output/mlpipeline-ui-metadata.json',
        'mlpipeline-metrics': '/output/mlpipeline-metrics.json',
      },
      image='lightbend/ml-tf-recommender:0.0.1') \
    .add_env_variable(k8s_client.V1EnvVar(name='MINIO_URL',value='minio-service.default.svc.cluster.local:9000')) \
    .add_env_variable(k8s_client.V1EnvVar(name='MINIO_KEY', value='minio')) \
    .add_env_variable(k8s_client.V1EnvVar(name='MINIO_SECRET', value='minio123')) \
    .add_volume(k8s_client.V1Volume(name='outputs', empty_dir=k8s_client.V1EmptyDirVolumeSource())) \
    .add_volume_mount(k8s_client.V1VolumeMount(name='outputs', mount_path='/output'))
  train.after(data)
    # Publish new model model
  publish = dsl.ContainerOp(
      name='publishmodel',
      output_artifact_paths={
        'mlpipeline-ui-metadata': '/output/mlpipeline-ui-metadata.json',
        'mlpipeline-metrics': '/output/mlpipeline-metrics.json',
      },
      image='lightbend/kubeflow-modelpublisher:0.0.1') \
    .add_env_variable(k8s_client.V1EnvVar(name='MINIO_URL',value='http://minio-service.default.svc.cluster.local:9000')) \
    .add_env_variable(k8s_client.V1EnvVar(name='MINIO_KEY', value='minio')) \
    .add_env_variable(k8s_client.V1EnvVar(name='MINIO_SECRET', value='minio123')) \
    .add_env_variable(k8s_client.V1EnvVar(name='KAFKA_BROKERS', value='kafka-bootstrap.default.svc.cluster.local:9092')) \
    .add_env_variable(k8s_client.V1EnvVar(name='DEFAULT_RECOMMENDER_URL', value='http://recommendermodelserver1.default.svc.cluster.local:8501')) \
    .add_env_variable(k8s_client.V1EnvVar(name='ALTERNATIVE_RECOMMENDER_URL', value='http://recommendermodelserver2.default.svc.cluster.local:8501')) \
    .add_volume(k8s_client.V1Volume(name='outputs', empty_dir=k8s_client.V1EmptyDirVolumeSource())) \
    .add_volume_mount(k8s_client.V1VolumeMount(name='outputs', mount_path='/output'))
  publish.after(train)
```



Google Cloud

Cloud Innovator MEGAZONE

Datadog을 통한 모니터링



Google Cloud

Cloud Innovator MEGAZONE

감사합니다.

Google Cloud

Cloud Innovator MEGAZONE