

24 클라우드 기반 크레딧 거래 플랫폼

소속 정보컴퓨터공학부

분과 C

팀명 클라우드나인

참여학생 김민승, 김민중, 송재원

지도교수 김원석

과제 소개



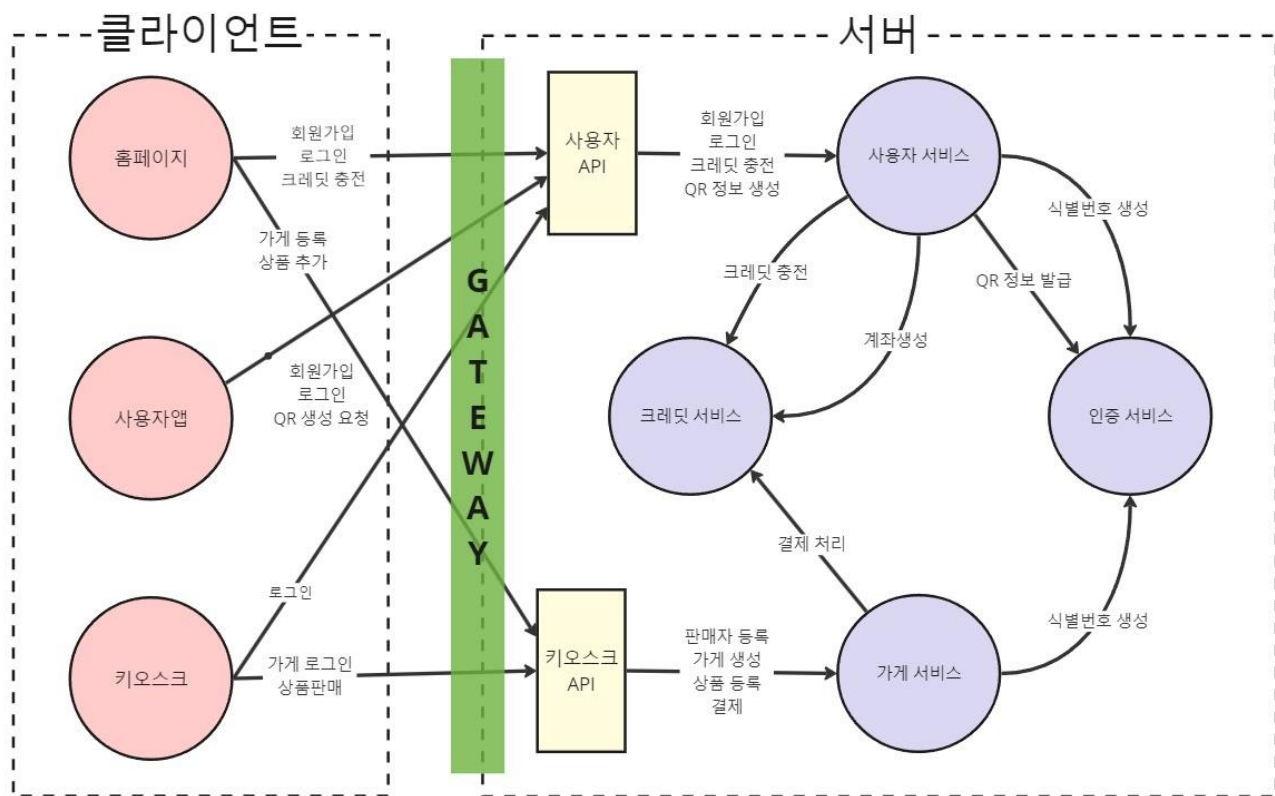
과제 개요

2022 기준 전체 약 59%의 결제가 모바일을 통해 이루어질 정도로 전자 결제가 크게 증가했다. 그리고 다양한 서비스의 등장과 규제 완화로 결제 방식 또한 빠르게 변화하고 있다. 이러한 상황에 맞춰 자유로운 확장이 가능한 클라우드 환경에 유연성이 장점인 MSA 설계를 적용하여 크레딧 결제 플랫폼을 구현하고자 한다.

과제 목표

- 클라우드 환경의 장점인 자유로운 서비스 확장을 고려한 크레딧 거래 플랫폼 개발
- Microservice Architecture 설계를 적용하여 확장성과 유연성 확보
- 컨테이너 오케스트레이션 기술인 kubernetes를 사용하여 운영

과제 구현



MSA(MicroService Architecture) 적용

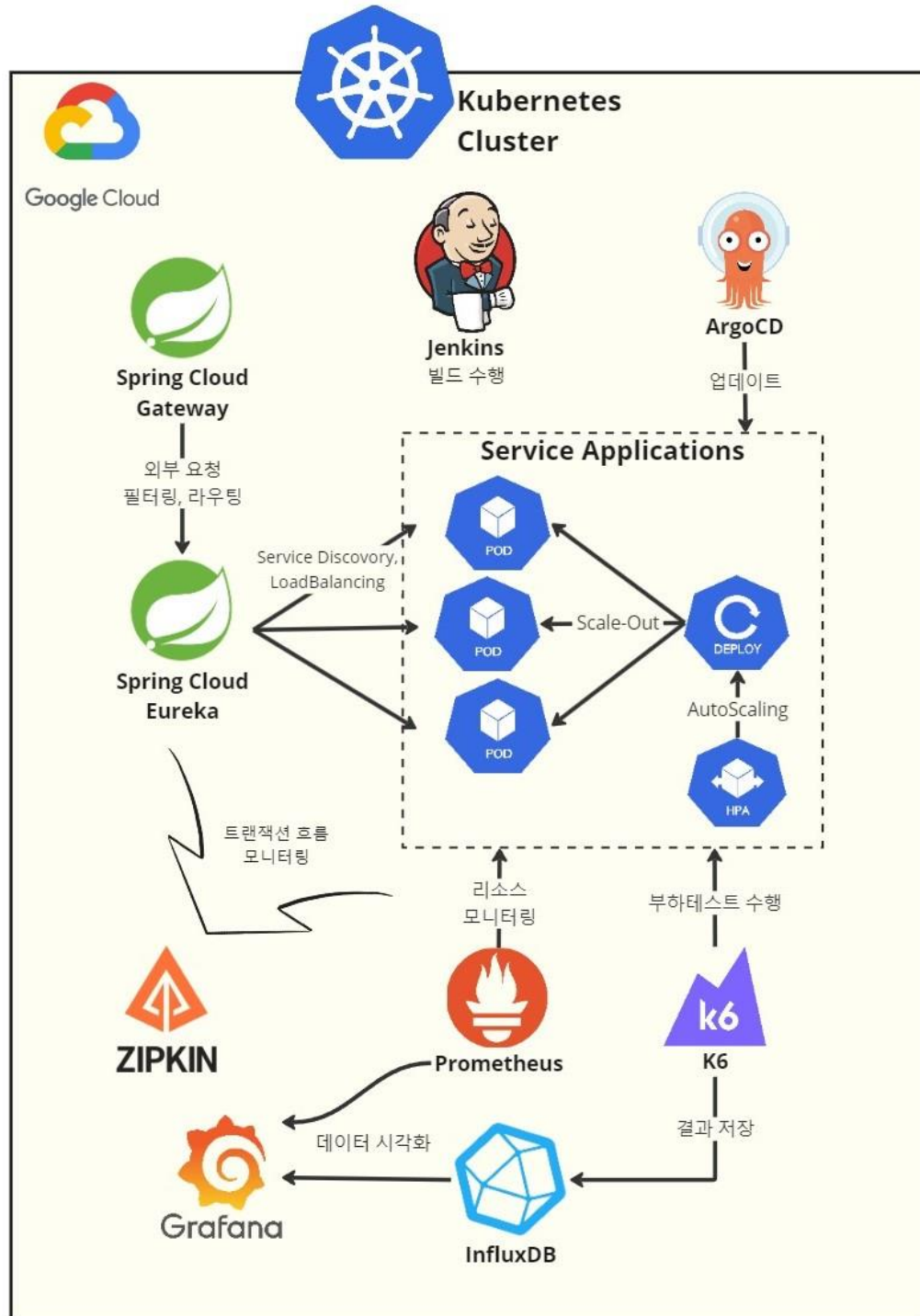
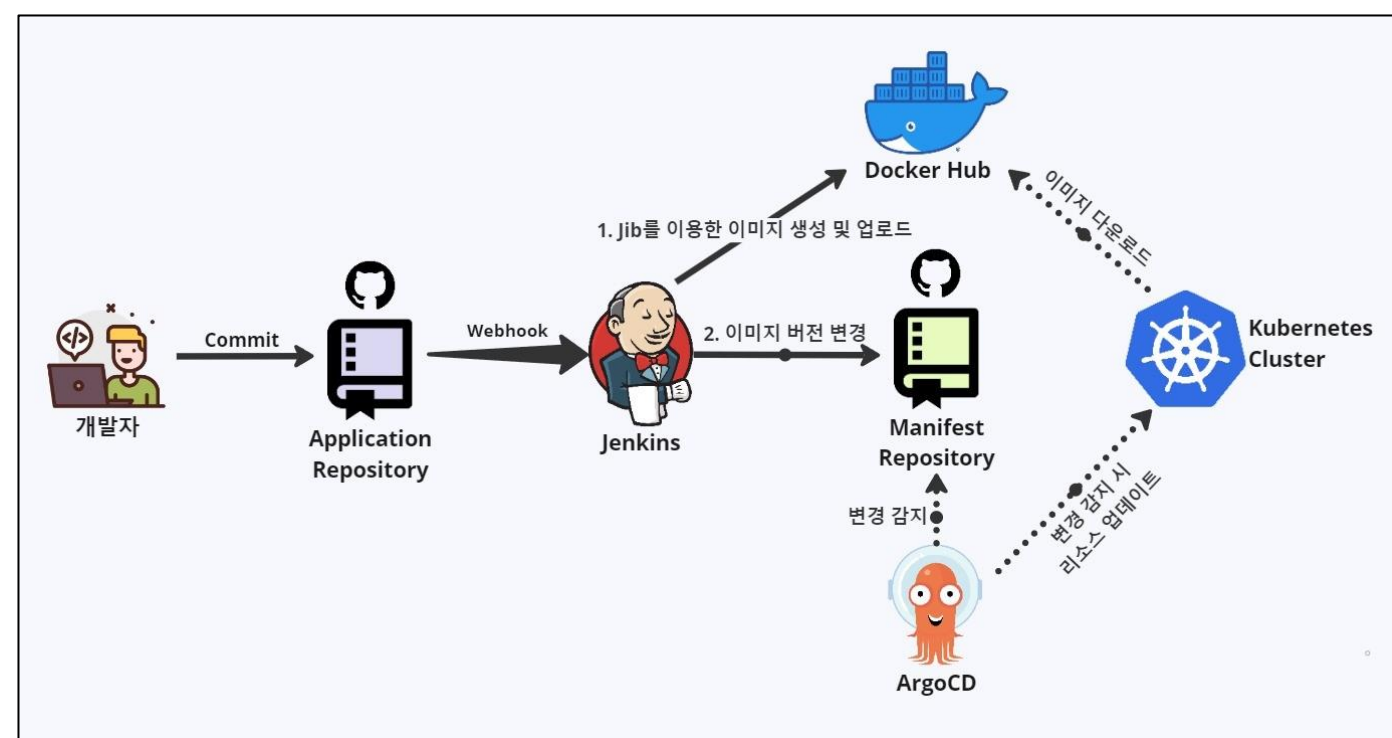
- 유연한 확장을 위한 MSA 설계
- 3개의 클라이언트, 4개의 서버로 시스템을 구성
- 자유로운 Scale-Out을 위해 무상태로 구현

보안을 고려한 설계

- 두 가지 접근 권한에 대한 서로 다른 방식의 JWT 탈취 방지 적용
- 비대칭 키를 이용한 서명으로 QR 코드 위·변조 방지
- Gateway를 이용해 단일 진입점을 제공하고 내부 API와 외부 API를 분리하여 API 접근 제어

CI/CD 파이프라인 구성

- Github webhook을 통한 자동 배포 시작
- Jenkins Dynamic Agent를 이용하여 병렬적으로 빌드 수행
- 빠른 배포 및 복원, 변경이력 관리, 편리한 클러스터 인프라 구성 공구를 위한 GitOps 방식 적용



클라우드 환경 K8S 클러스터

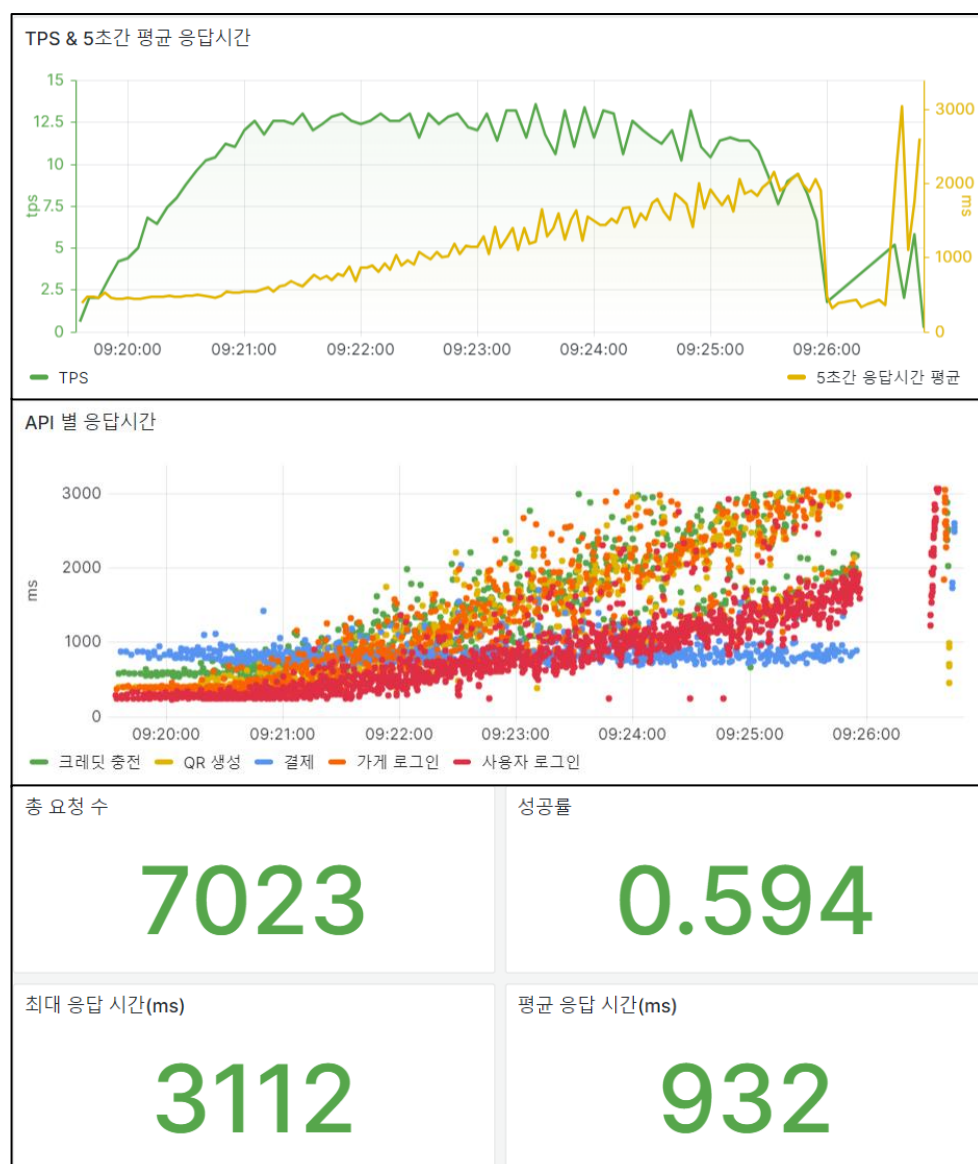
- Spring Cloud Eureka를 사용하여 서비스 디스커버리 및 로드밸런싱
- 파드별 CPU 사용량을 기준으로 Autoscaling 적용
- Prometheus로 클러스터의 리소스를 모니터링하고 Zipkin으로 트랜잭션 흐름 모니터링

부하 테스트

K6를 이용한 부하 테스트

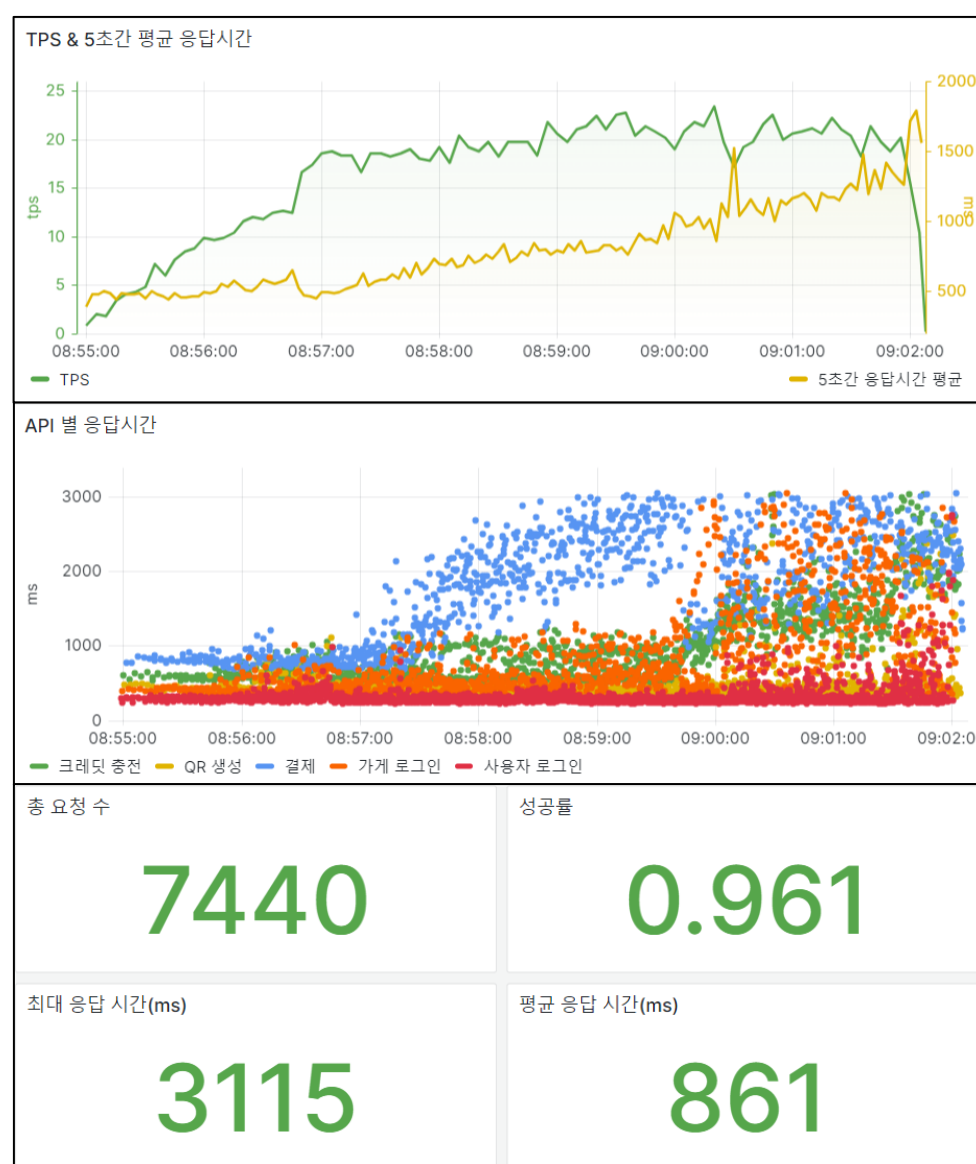
- K6는 javascript로 간편하게 테스트 코드 작성 가능한 부하 테스트 도구이다.
- Autoscaling이 성능에 미치는 영향을 확인하기 위해 서비스별로 파드가 하나씩만 존재하는 단일 파드 환경과 Autoscaling이 적용된 환경으로 나누어 테스트했다.
- 시스템의 대표적인 API를 사용하여 테스트 시나리오를 작성하고 병렬적으로 시나리오를 수행하는 테스트의 수를 1에서 30까지 7분간 늘리며 진행했다.
- TPS(Transaction Per Second) 초당 처리하는 트랜잭션의 수로 서비스의 성능을 나타내는 대표적인 지표이다.

단일 파드 환경

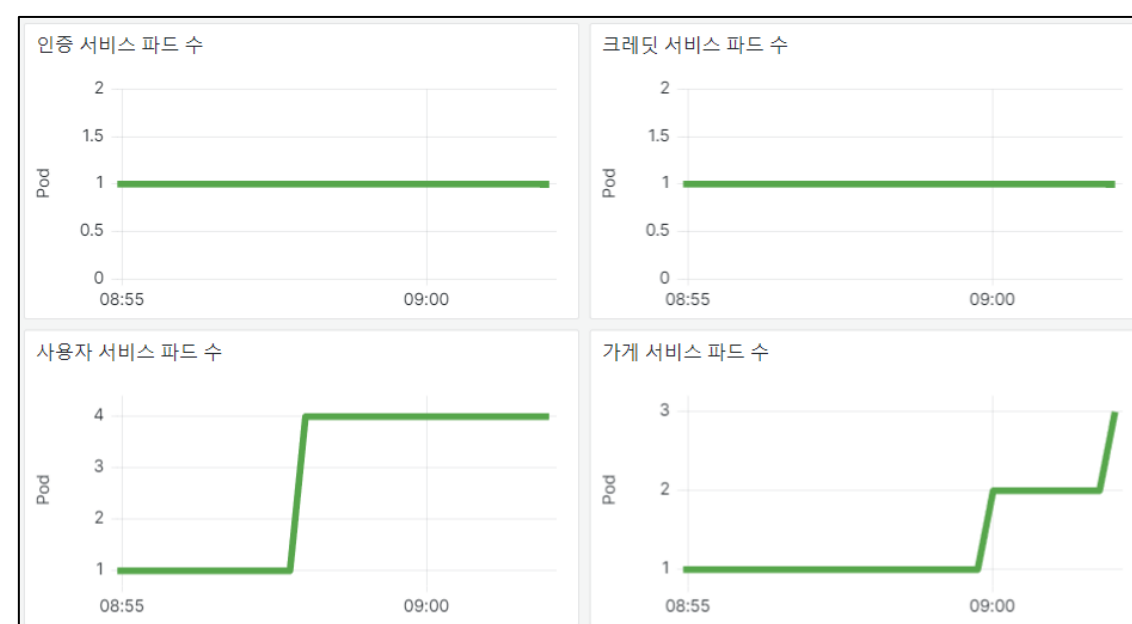


VS

Autoscaling 환경



Autoscaling 환경 파드 수 변화



결과

- Circuit Breaker의 동작으로 인해 평균 응답시간은 큰 차이가 없으나 상태코드 200 응답의 비율인 성공률은 Autoscaling 환경이 매우 높다.