

HaaS

High Availability Streaming as a Service

RAPA KAKAO Cloud School 2nd Toy Project



INDEX

01. 프로젝트 개요

02. 구성도

03. 기술 설명

04. 시연 결과

05. Next Step

1

프로젝트 개요



프로젝트 개요

프로젝트 목표

프로젝트 주제

CI/CD 및 K8S를 활용하여 자동화된 클라우드 인프라 환경을 구축하고, 고객 요구 사항을 반영한 음원 스트리밍 서비스를 제공한다.

프로젝트 기간

2024.03.15 (금) - 2024.04.03 (수)

< 프로젝트 배경 및 필요성 >

클라우드 기반 인프라와 K8S 기술을 활용한 자동화는 서비스의 안정적 운영과 지속적 개선에 필수적이다. 이를 통해 인프라의 유연성 및 확장성을 강화하고, 효율적인 리소스 관리 및 빠른 시장 대응력을 실현할 계획이다.

< 프로젝트 목표 >

- 사용자 경험 향상: 직관적인 UI/UX 설계를 통해 사용자 만족도 극대화
- CI/CD 파이프라인 구축: 지속적인 통합 및 배포를 통한 신속한 반복 개발 및 빠른 대응
- 모니터링 및 성능 측정: 실시간 데이터 분석을 통한 서비스의 안정성 보장 및 최적화
- 데이터 스토리지: AWS RDS 및 S3를 사용한 효율적인 데이터 관리 및 확장 가능한 스토리지 솔루션
- 운영 효율성: Kubernetes를 활용한 클라우드 자원의 자동화 및 최적화

팀원 소개



김정우

팀장



허지웅

웹 개발



이동준

K8S



김혜림

모니터링



노현

성능 측정

프로젝트 개요

팀 소개 및 구성

팀 소개

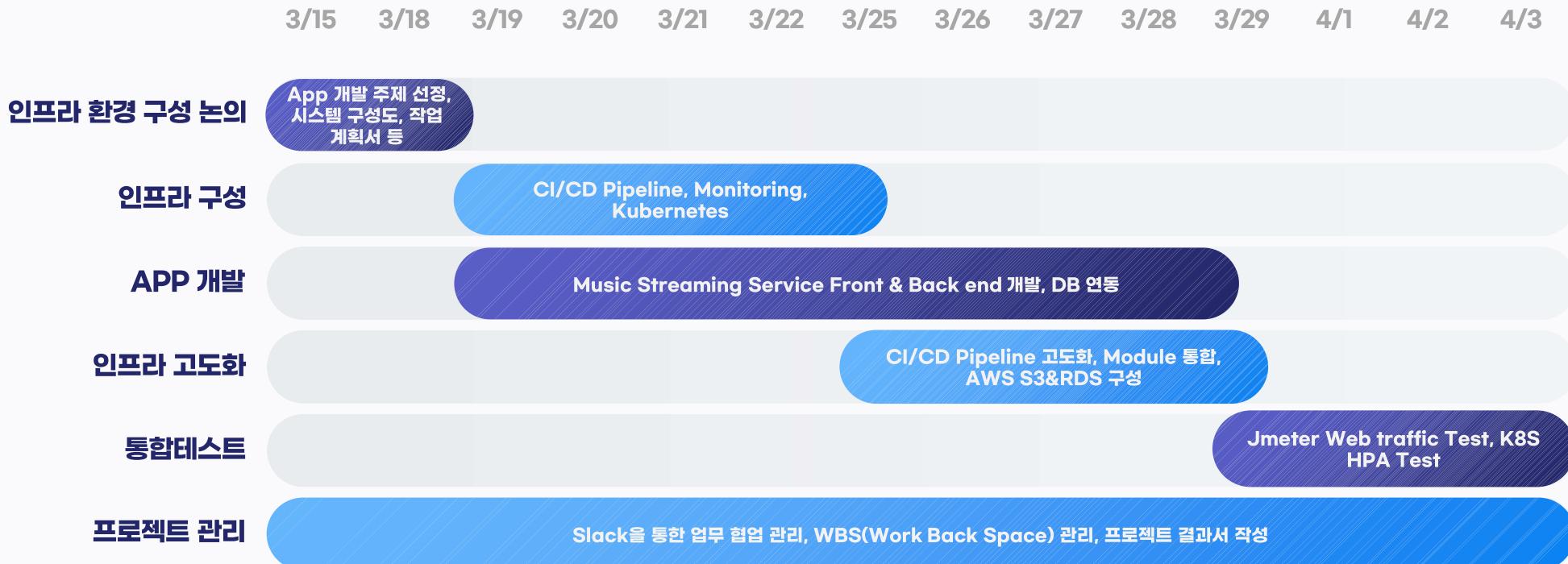
유칼립투스(Eucalyptus)는 강인함과 지속 가능성을 상징합니다.
팀 명과 같이 급변하는 클라우드 환경에서 견고하고, 신뢰할 수 있는 인프라 서비스를 제공합니다.

팀 구성

역할	이름	이메일	역할
팀장 (PM)	김정우	jwook_7@naver.com	클라우드 인프라 환경 구성 및 CI/CD 파이프 라인 구축
팀원 (PL)	허지웅	koreapower98@naver.com	음원 스트리밍 서비스 APP 개발
팀원	이동준	dzdzaa7@gmail.com	음원 스트리밍 서비스 DB 연동, K8S IaC 코드 작성
팀원	김혜림	pilothr12@gmail.com	Grafana & Prometheus & PromQL을 활용 Monitoring 시스템 구축
팀원	노현	hyunn915@gmail.com	Jmeter를 활용한 성능 테스트 및 프로젝트 결과서 작성

프로젝트 개요

WBS



프로젝트 개요

주제 선정 이유

K8S의 오토스케일링 기능이 실제 스트리밍 서비스 성능, 사용자 경험에 미치는 중요한 요소인 '버퍼링'에 어떤 영향을 미치는가?



**오토스케일링이
음악 스트리밍 버퍼링 시간 최소화 및 스트리밍 품질 유지에 어떻게 기여하는지
실질적인 사용자 경험을 통해 평가할 수 있는 기회를 제공**

Technology

기술 스택

< Orchestration >



kubernetes

< CI/CD >



Jenkins



argo



< Monitoring >



Prometheus



Grafana

< Performance Measurement >



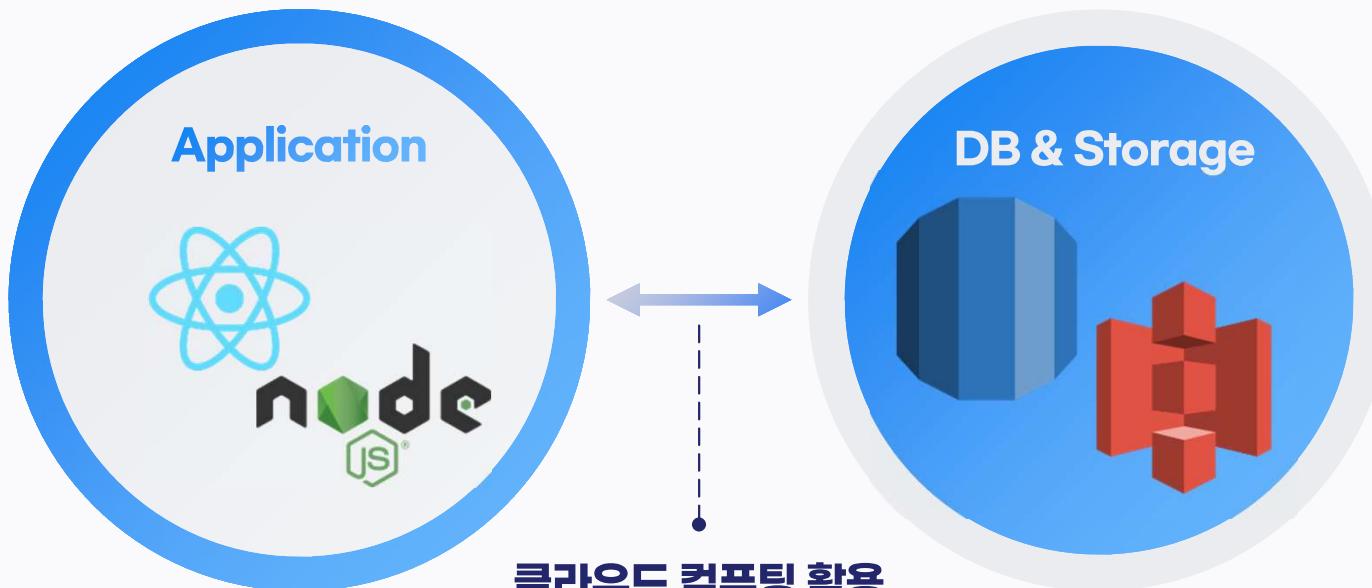
< Collaborative Software >



Google Workspace



Technology
기술 스택

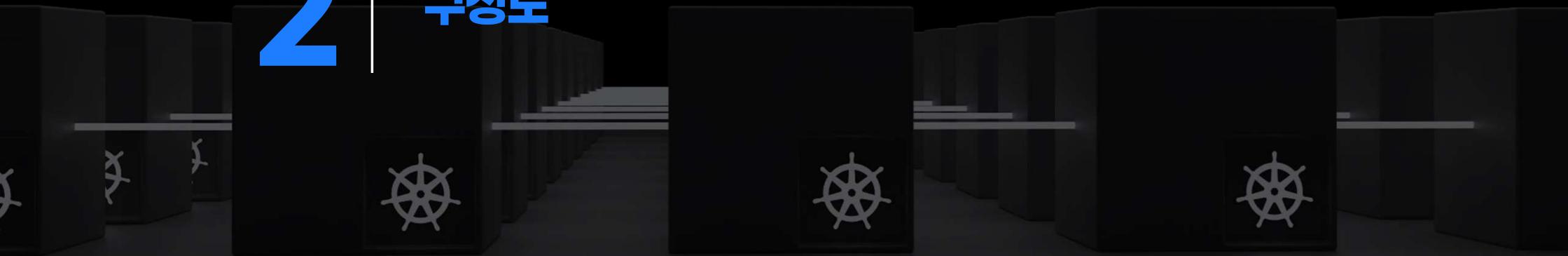


AWS의 서비스인 RDS, S3를 이용,
관리가 편리하고 확장성이 뛰어나기 때문에
운영 부담을 줄이고 성능을 최적화할 수 있습니다

Technology 기술 스택

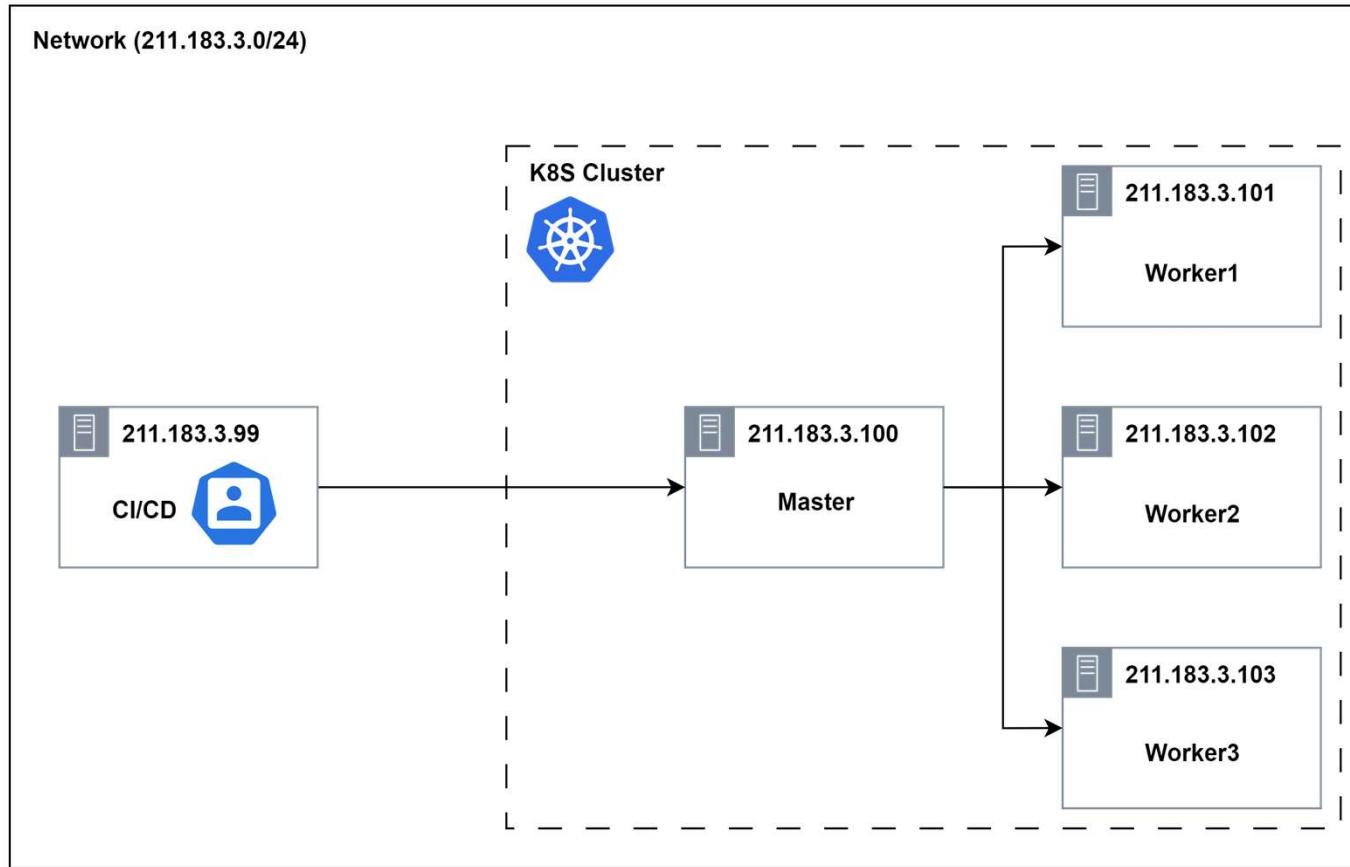


2 | 구성도



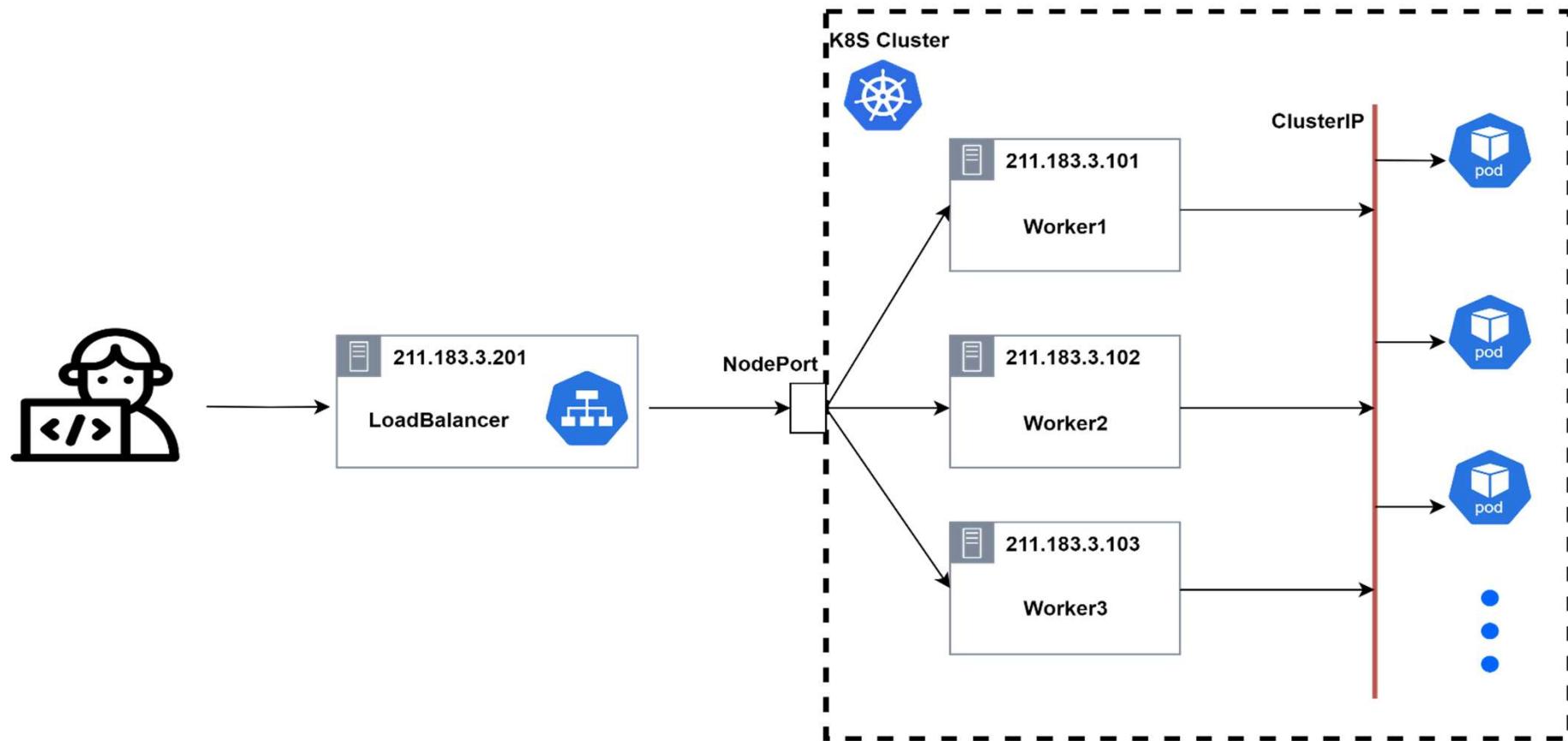
네트워크 구성도

구성도



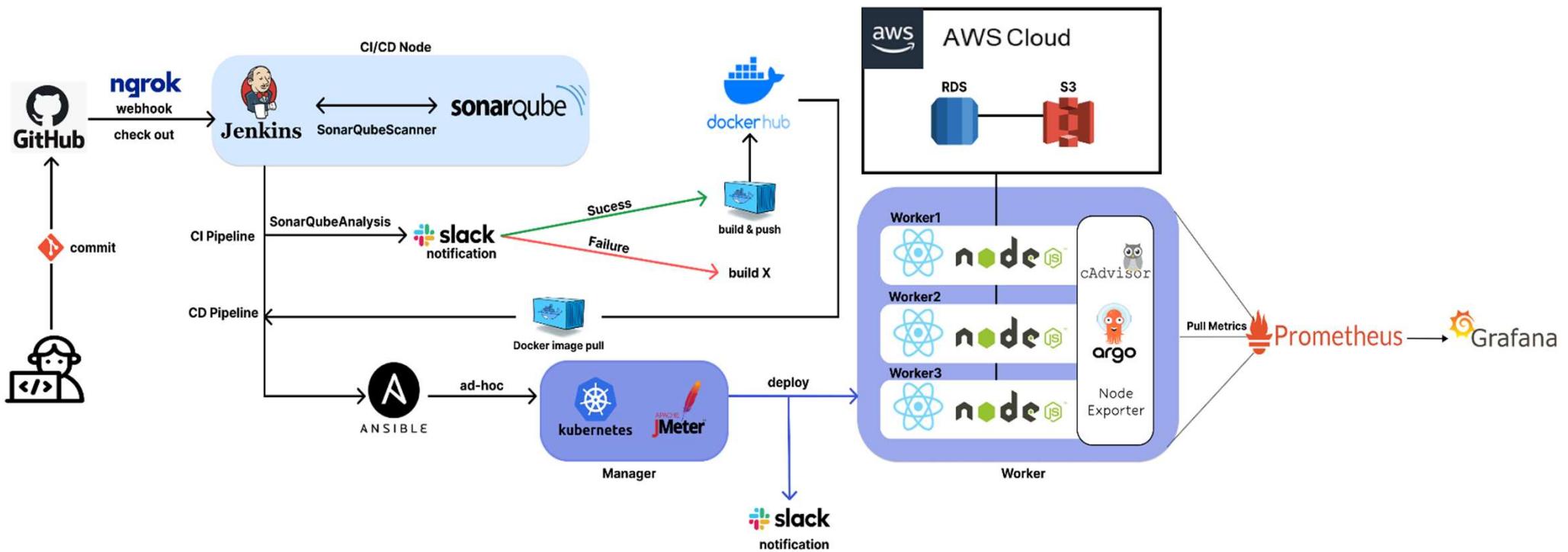
서비스 구성도

구성도



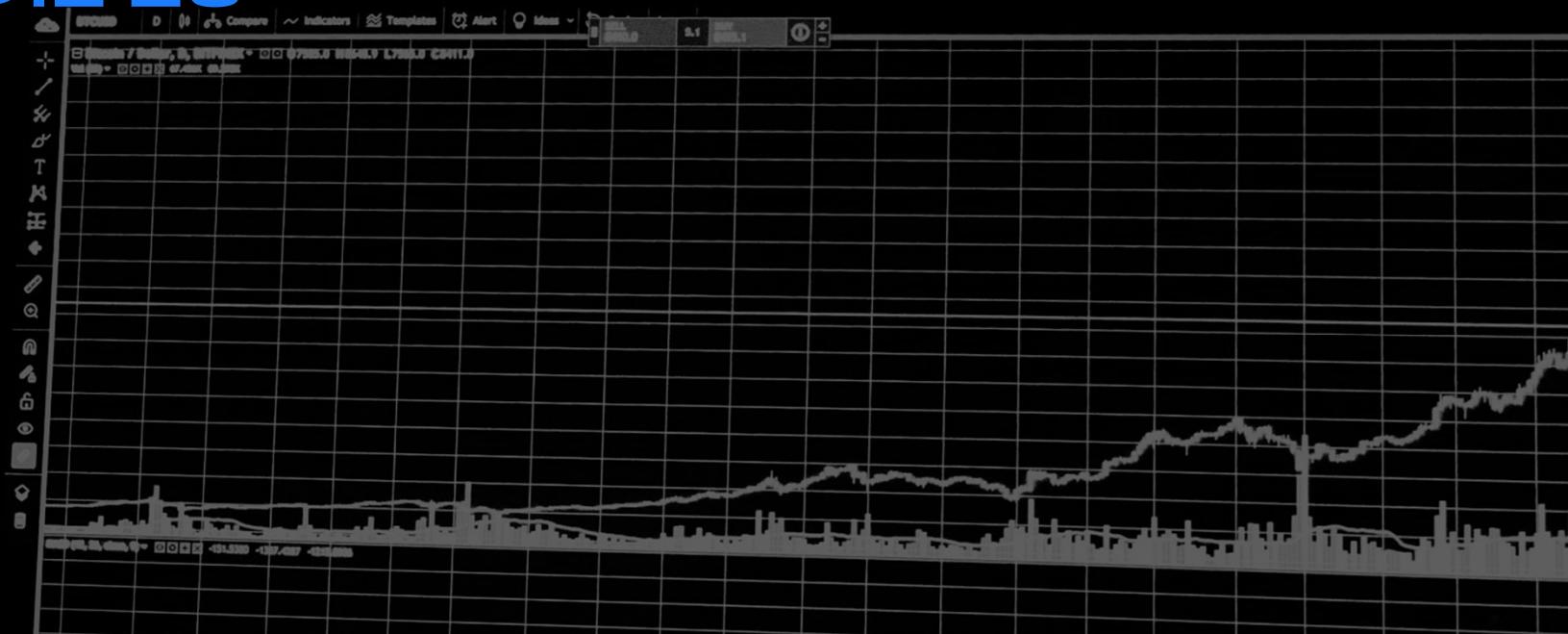
시스템 구성도

구성도



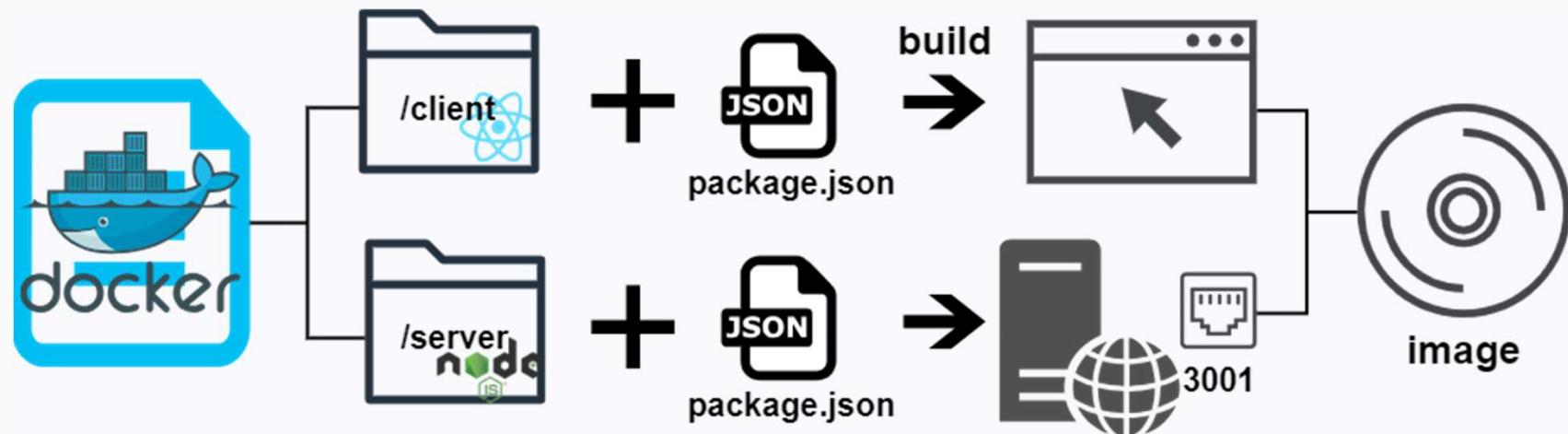
3

기술 설명





Web Application
이미지 빌드



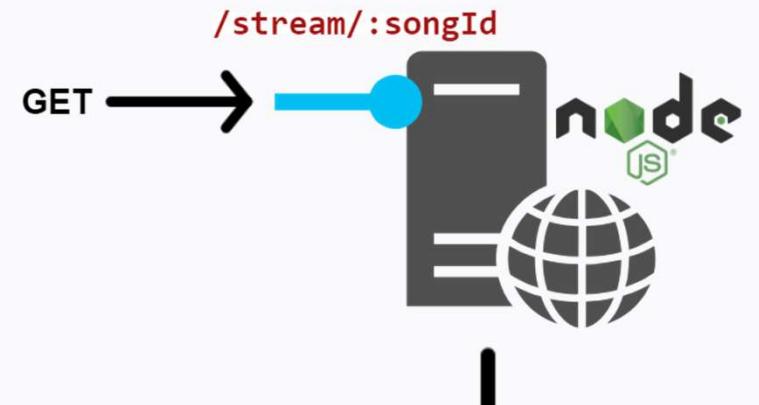
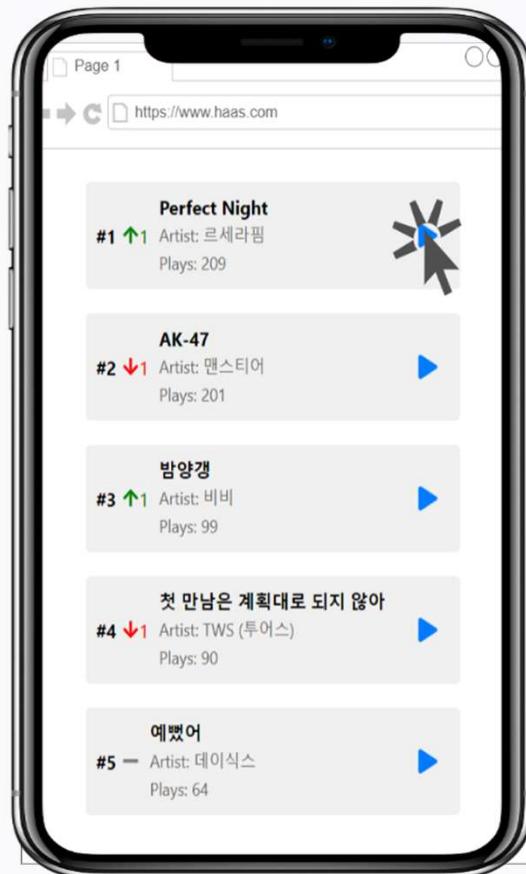
Web Application

프론트 & 백엔드



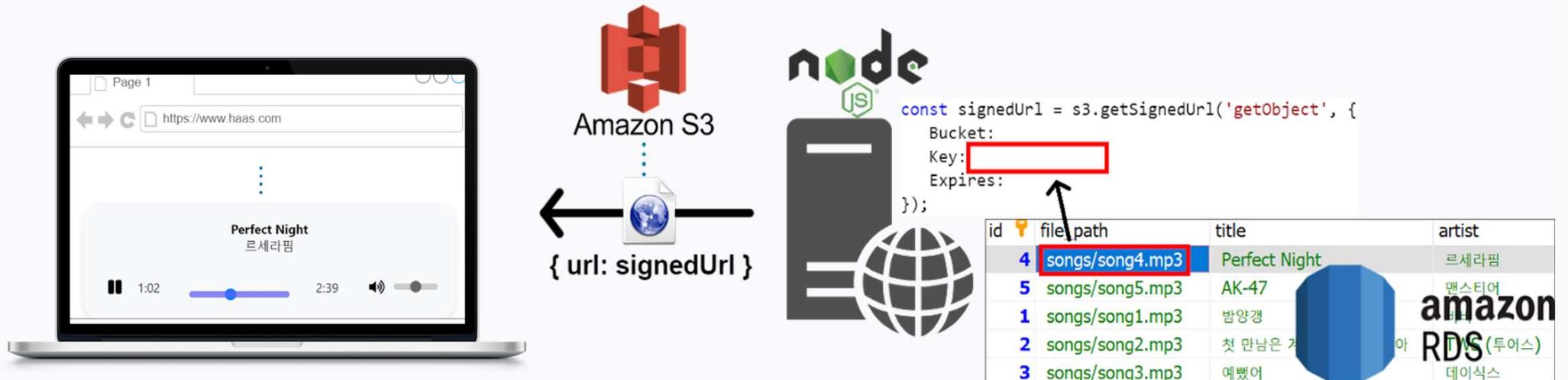
Web Application

음악 재생이 이루어지는 과정 (1)

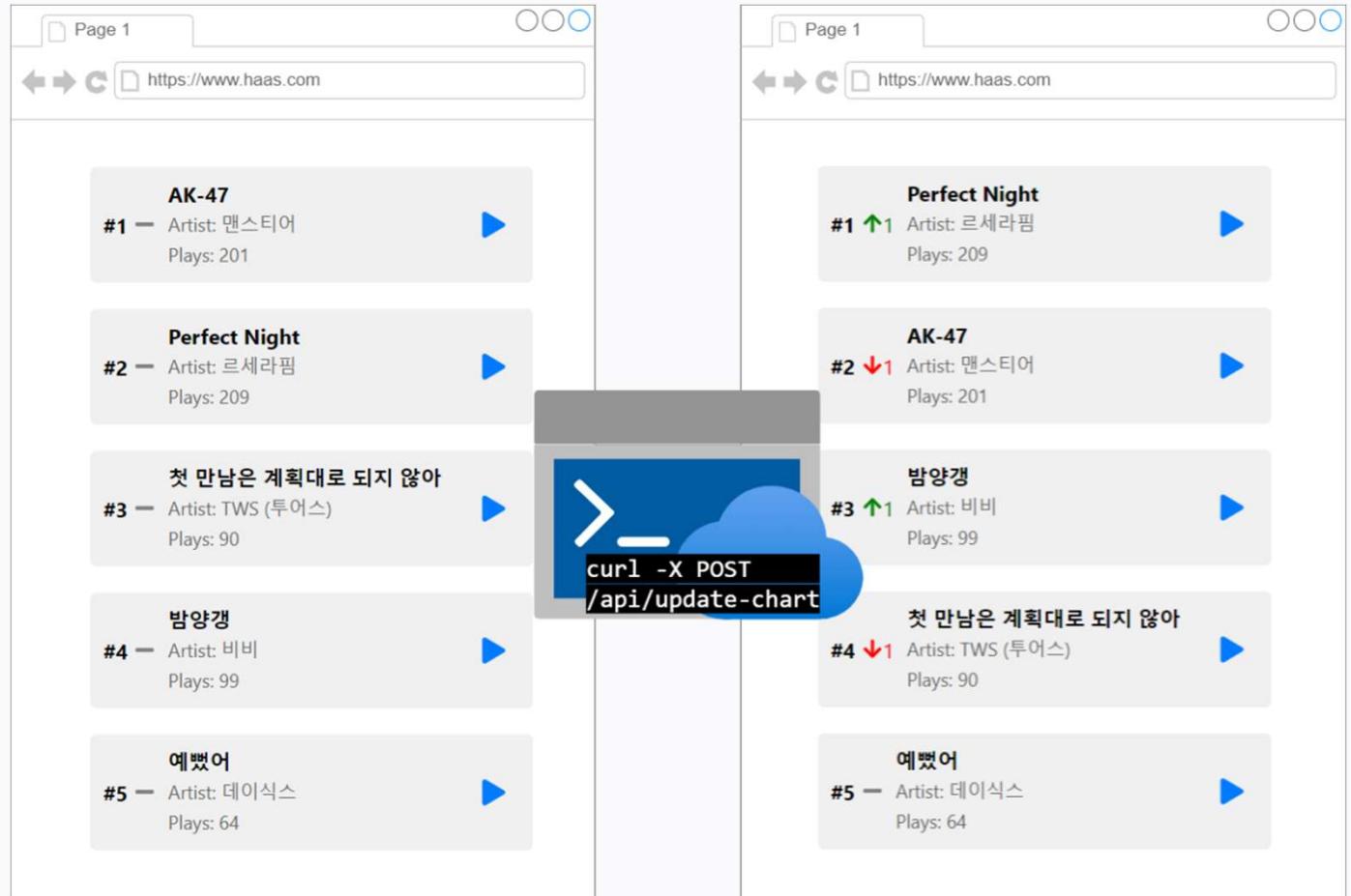
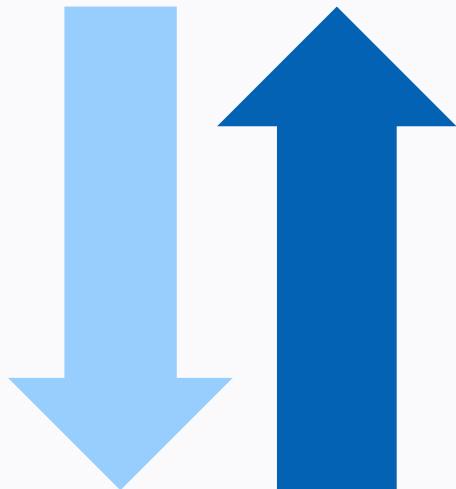


id	file_path	title	artist
4	songs/song4.mp3	Perfect Night	르세라핌
5	songs/song5.mp3	AK-47	맨스티어
1	songs/song1.mp3	밤양갱	
2	songs/song2.mp3	첫 만남은 계획대로 되지 않아	TWS (투어스)
3	songs/song3.mp3	예뻤어	데이식스

Web Application 음악 재생이 이루어지는 과정 (2)



Web Application 차트 업데이트





CI/CD 구성 목표

기술 설명 및 시연 결과 | CICD

지속적
통합



코드 형상 관리: 개발자는 변경된 코드를 Git hub의 업로드하며, 코드 형상 관리를 수행한다.
코드 품질 관리: 코드 품질을 개선하기 위해 Commit 즉시, SonarQube가 코드를 분석한다.
빌드 자동화: SonarQube 분석이 올바르게 수행되었을 때, Docker image build&Push를 수행한다.
알림: Pipeline 수행 결과를 Slack을 통해 채널 내 구성원에게 알림을 제공한다.

지속적
배포



배포 자동화: 새로운 코드 변경 사항이 성공적으로 테스트 통과 및 빌드된 경우, 자동으로 배포한다.
제로 다운타임 배포: K8S 배포 방식인 RollingUpdate를 통해 사용자 경험에 영향을 주지 않고 신규 버전을 배포한다.
배포 모니터링: argoCD를 통해 배포된 Pod를 시각화하고, Pod의 Health check와 같이 상태를 점검한다.
알림: 배포 결과를 Slack을 통해 채널 내 구성원에게 알림을 제공한다.

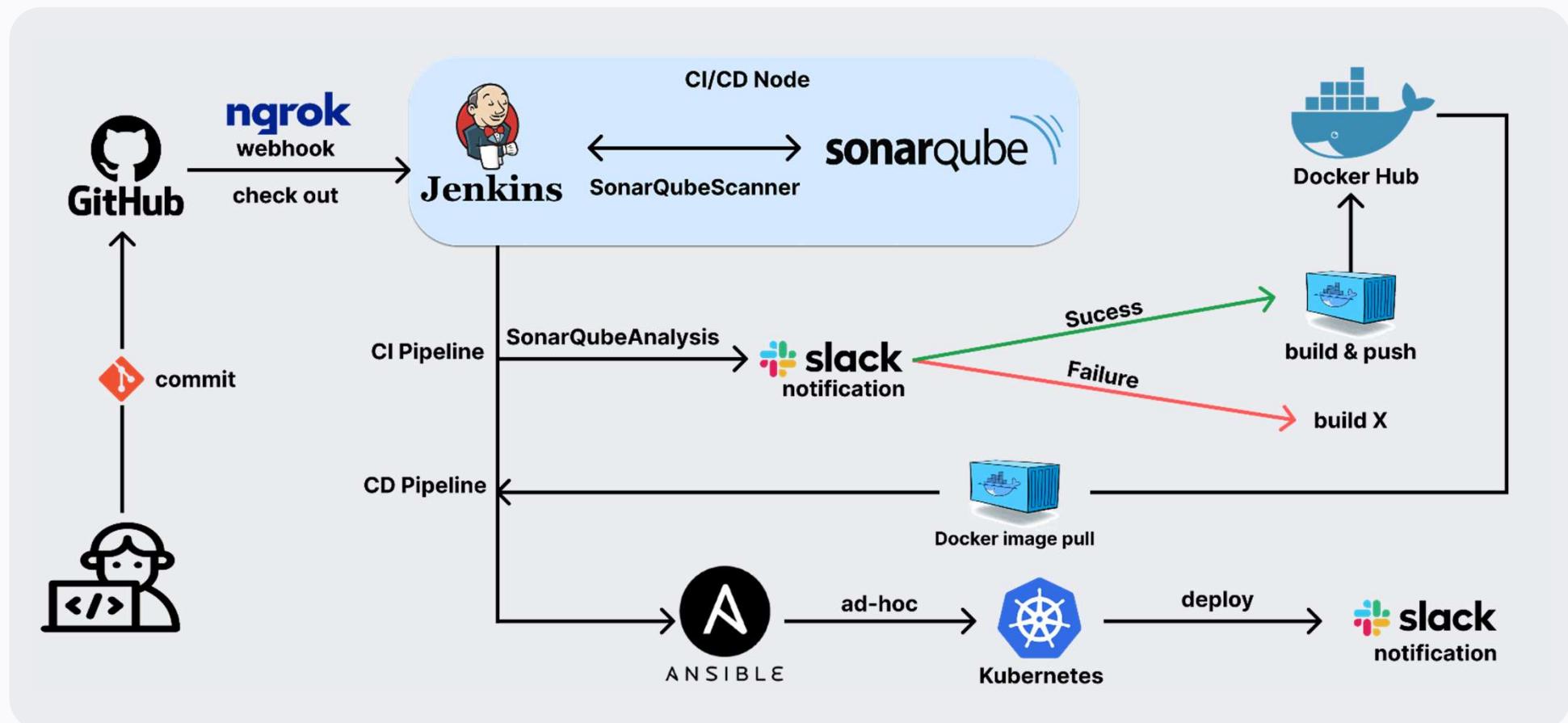
지속적
피드백



피드백 반영: 팀 내 피드백을 반영하여 Pipeline 개선하고, 더욱 효율적인 Pipeline을 구성한다.

CI/CD 구성도

기술 설명 및 시연 결과 | CICD



CI/CD 장점

기술 설명 및 시연 결과 | CICD



자동화된 워크플로우

개발자가 GitHub에 커밋을 하면 자동으로 CI 파이프라인이 동작하며, 결과에 따라 CD 까지 수행하는 개발 프로세스 자동화를 지원한다.



통합된 코드 품질 관리

SonarQube를 통해 자동화된 코드 품질 분석을 수행합니다. 지속적으로 코드의 품질을 관리할 수 있습니다.



배포 자동화&모니터링

CI 단계에서 성공적으로 테스트 통과 및 빌드된 경우 배포를 수행하며, argoCD를 통해 배포된 Pod를 시각화 및 모니터링을 수행할 수 있습니다.



운영 효율성

K8S를 사용하면 신규 버전을 자유롭게 배포하며, 컨테이너화된 애플리케이션을 오퍼스트레이션하여 가용성과 확장성을 높일 수 있습니다.



알림

Slack을 통해 빌드의 성공 및 실패에 대한 알림을 받을 수 있어, 팀 구성원들이 실시간으로 상태를 파악할 수 있습니다.

CI/CD 단점

기술 설명 및 시연 결과 | CICD



ngrok 사용

ngrok은 로컬 서버를 임시적으로 인터넷에 공개하는 도구로, 재부팅 시에는 새로운 주소를 발급 받아야 합니다.



보안 고려 사항

외부 서비스 사용 시 보안 설정은 중요하며, ansible SSH 통신 및 ngrok webhook 트리거는 취약점으로 인식될 수 있습니다.



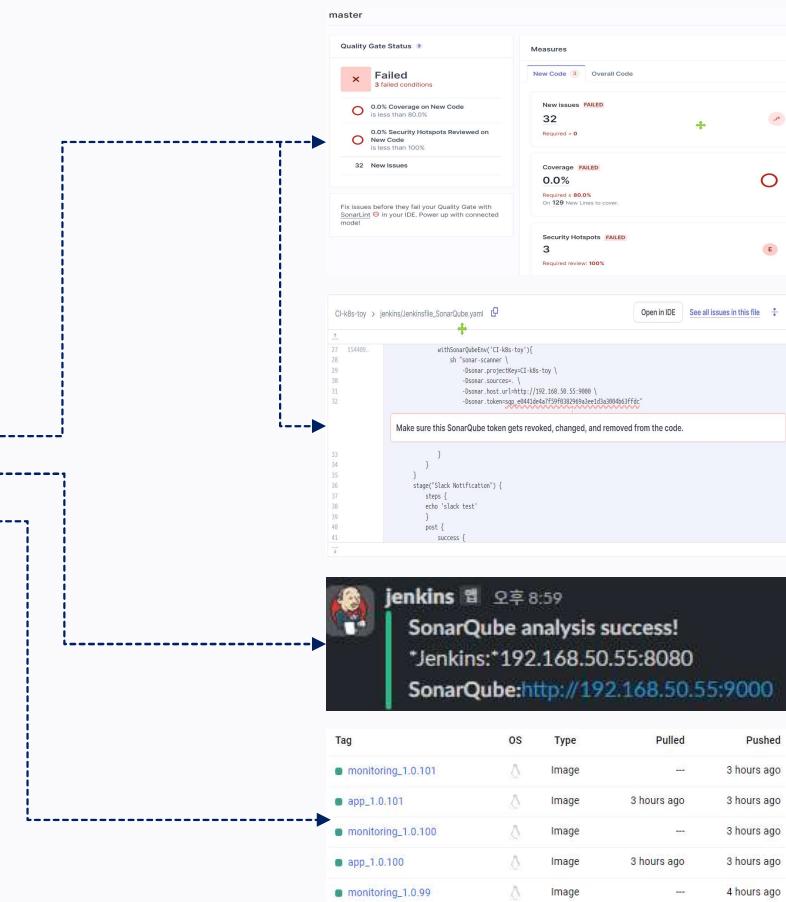
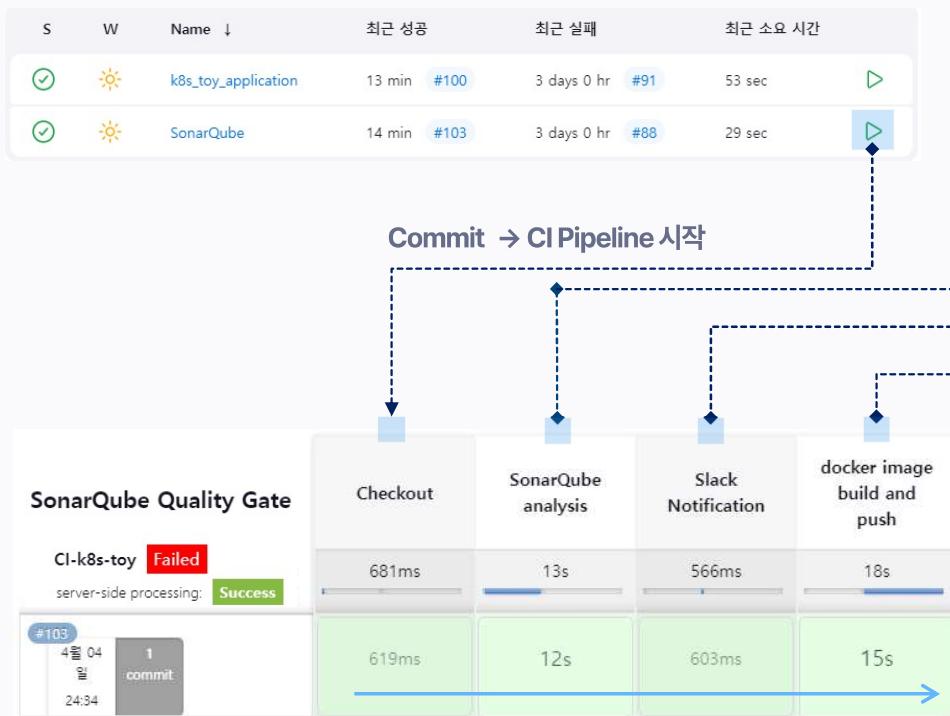
확장성 및 유지보수

현재 코드는 간결하게 작성되었지만, 파이프라인이 발전함에 따라 복잡성이 증가할 것으로 예상되며, 따라서 유지보수와 확장성에 대한 고려가 필요합니다.

CI/CD 결과 이미지

기술 설명 및 시연 결과 | CICD

Commit 발생 이후 SonarQube Analysis를 실시하고,
분석을 성공적으로 마친 경우, Docker image build를 수행한다.

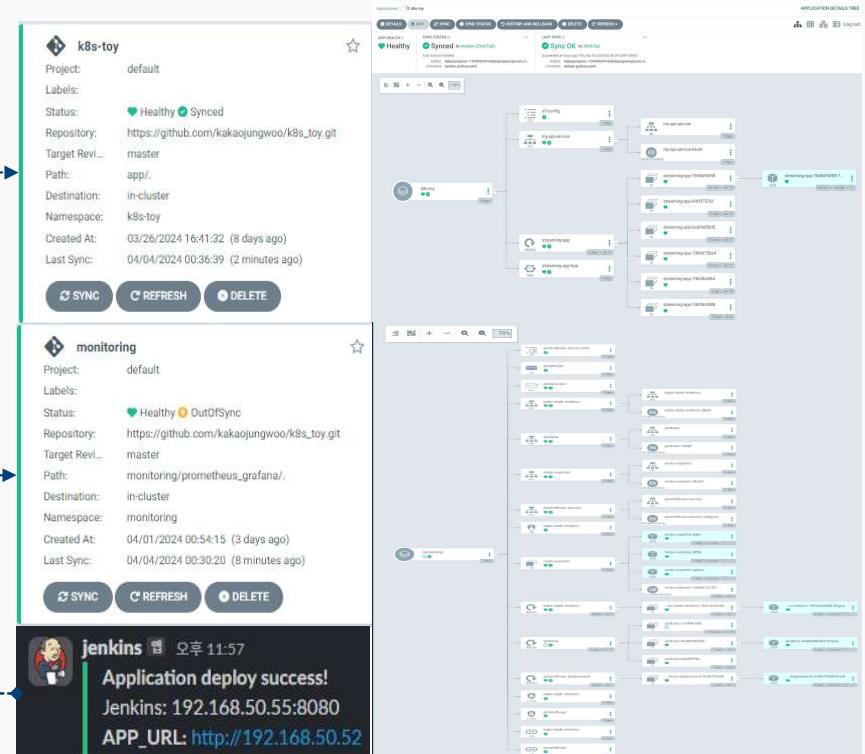
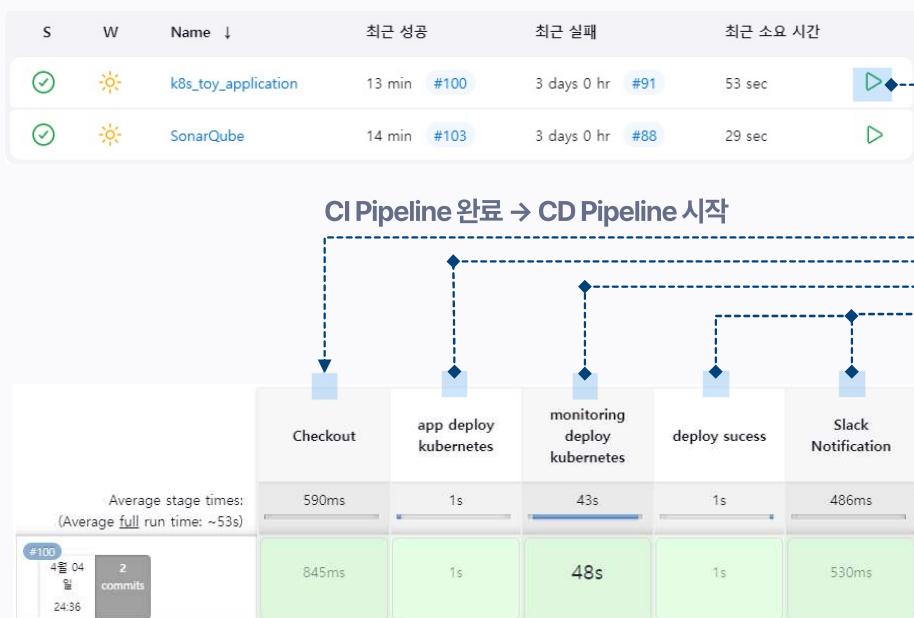


CI/CD 결과 이미지

기술 설명 및 시연 결과 | CICD

CI Pipeline이 성공적으로 수행된 경우, CD Pipeline을 실행한다.

순차적으로 Application → Monitoring → Notification으로 실행된다.

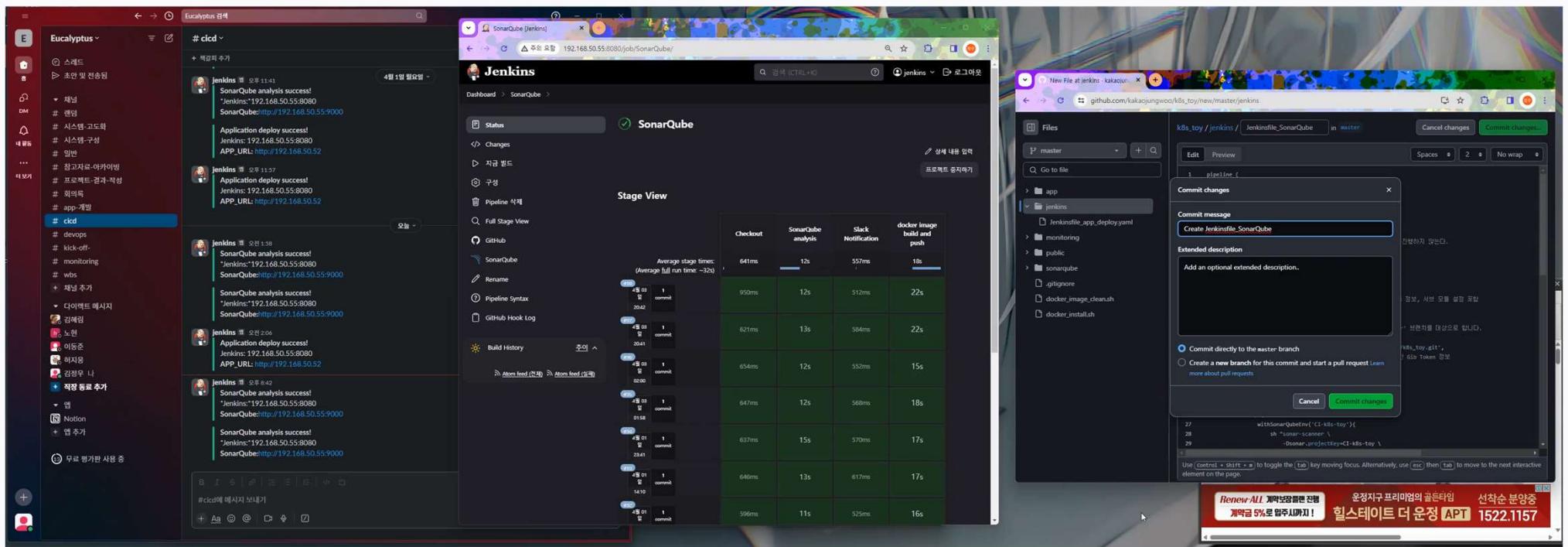


CI 시연 영상

기술 설명 및 시연 결과 | CICD

CI시연 시나리오

Commit 발생 이후 SonarQube Analysis를 실시하고, 분석을 성공적으로 마친 경우, Docker image build를 수행한다.



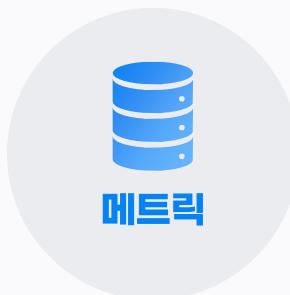


기술 설명 및 시연 결과 | Monitoring

Prometheus & Grafana 설정 이유

다양한 메트릭 수집

컨테이너, 네트워크 사용량, 메모리 및 CPU 사용량과 같은 다양한 시스템 메트릭을 수집한다.



대시보드 커스터마이징

사용자가 필요에 따라 대시보드를 손쉽게 커스터마이즈하고 Promql을 통해 다양한 메트릭 기반의 대시보드 생성 가능.



시계열 데이터에 최적화

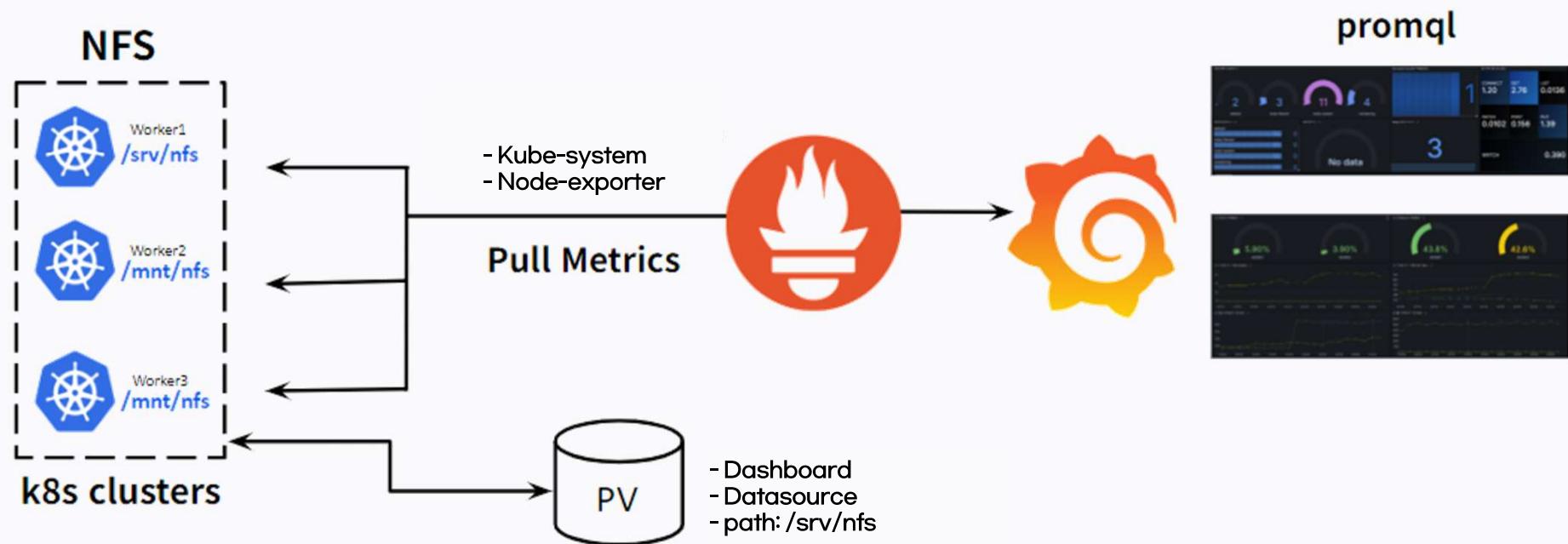
Prometheus는 시계열 데이터에 최적화되어 있어, Kubernetes 클러스터에서 발생하는 대량의 시계열 데이터를 효과적으로 처리하고 저장할 수 있다.

실시간 데이터 처리

그라파나는 데이터 소스에서 실시간으로 데이터를 수집하고, 이를 실시간으로 처리하여 대시보드에 반영할 수 있다. 이를 통해 실시간 상태를 모니터링하고, 문제가 발생했을 때 신속하게 대응할 수 있도록 도와준다.

기술 설명 및 시연 결과 | Monitoring

Monitoring 구성도



운영 효율성을 위한 NFS + PVC 구성

기술 설명 및 시연 결과 | Monitoring

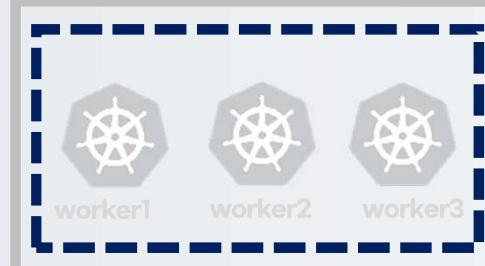


데이터 유실: 노드 재시작 시 대시보드/데이터 소실.

가용성 저하: 노드 장애 시 전체 시스템 영향.

효율성 부족: 데이터/설정 관리 복잡화.

NFS + PVPVC



데이터 보존: 노드를 재시작해도 데이터 유지.

가용성 향상: 장애 시 빠른 복구.

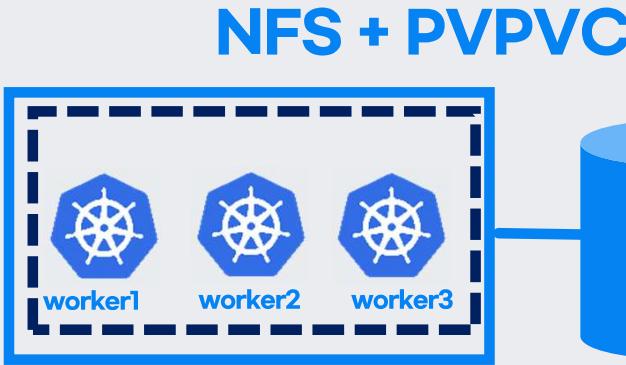
효율성 향상: 스토리지를 활용한 효율적 관리.

운영 효율성을 위한 NFS + PVC 구성

기술 설명 및 시연 결과 | Monitoring



데이터 유실: 노드 재시작 시 대시보드/데이터 소실.
가용성 저하: 노드 장애 시 전체 시스템 영향.
효율성 부족: 데이터/설정 관리 복잡화.



데이터 보존: 노드를 재시작해도 데이터 유지.
가용성 향상: 장애 시 빠른 복구.
효율성 향상: 스토리지를 활용한 효율적 관리.

Metrics

기술 설명 및 시연 결과 | Monitoring

Kube state metrics

Kubernetes 클러스터의 오브젝트 상태 정보 (파드, 노드, 디플로이먼트 등)를 수집합니다.

Kubernetes pods

Node-exporter가 포함되어 있으며 각 노드의 운영 체제 및 하드웨어 수준의 상세한 성능 지표를 수집합니다.

Kubernetes nodes

K8S API를 통해 직접 수집하는 노드 관련 정보이며 K8S 내장 metrics를 사용하여 노드의 상태, CPU, 메모리 사용량을 수집합니다.

Kubernetes cAdvisor

노드별 컨테이너 자원 사용량과 성능 메트릭스를 수집합니다.

Kubernetes apiservers

Kubernetes API 서버의 성능과 요청 처리 데이터를 수집합니다.

Kubernetes service endpoints

K8S 서비스 엔드포인트에서 발생하는 metrics 수집하며 서비스의 가용성, 엔드포인트의 상태 및 성능을 모니터링하는데 사용된다



4 | 시연 결과



프로젝트 시연 및 결과

JMETER를 활용한 성능 측정

테스트 기준 설정

- HTTP Request 기준
- Number of Threads: 가상의 생성자
- Ramp-up Period (in seconds): 1회 실행 초
- Loop Count: 반복 횟수

테스트 시나리오

시나리오 A:

1개의 Pod에서 몇 명의 유저까지 수용이 가능한가?

시나리오 B:

다수의 Pod에서 몇 명의 유저까지 수용이 가능한가?



JMETER 측정 결과

Scale-Up할 경우,
Pod의 Pending 없이

효율적으로 Pod 생성



노드에 메트릭 부하량 감소

프로젝트 시연 및 결과

JMETER를 활용한 성능 측정

시나리오 A: 1개의 Pod에서 몇 명의 유저까지 수용이 가능한가?

Scenario A	Pod Spec		Users/Seconds /Loop Count	Expected user load	HPA/Pod/Result	Pod 운영
-	MAX	CPU:1 Memory: 512 Mi	6,000u / 120s / 1c	50u/1s	X / 1 / 안정	약 1초당 70~80명 수용 예상
	MIN	CPU:0.2 Memory: 256 Mi	12,000u / 120s / 1c	100u/1s	X / 1 / 보통	
Scale UP	MAX	CPU:1.5 Memory: 1024 Mi	6,000u / 120s / 1c	50u/1s	X / 1 / 안정	약 1초당 130~150명 수용 예상
	MIN	CPU:0.5 Memory: 512 Mi	12,000u / 120s / 1c	100u/1s	X / 1 / 안정	
			18,000u / 120s / 1c	150u/1s	X / 1 / 안정	

프로젝트 시연 및 결과

JMETER를 활용한 성능 측정

노드별 CPU 사용량

12.30%

Worker1

13.10%

Worker2

12.05%

Worker3

노드별 Memory 사용량

31.4%

Worker1

38.8%

Worker2

27.7%

Worker3

01

1개의 Pod에서
몇 명의 유저까지
수용이 가능한가?

* 360,000u/120s/1c

시나리오 A

* 중간 값 기준으로 측정함

프로젝트 시연 및 결과

JMETER를 활용한 성능 측정

시나리오 B: **다수의 Pod에서 몇 명의 유저까지 수용이 가능한가?**

*** Max scale Replicas: 10

Scenario B	Pod Spec		Users/Seconds /Loop Count	Expected user load	Max Replicas /Pod/Result	Pod 운용		
-	MAX	CPU:1 Memory: 512 Mi	120,000u / 120s / 1c	1,000u/1s	10 / 6 / 안정	약 1초당 1000명 수용 예상		
			240,000u / 120s / 1c	2,000u/1s	10 / 10 / 보통			
	MIN	CPU:0.2 Memory: 256 Mi	360,000u / 120s / 1c	3,000u/1s	10 / 10 / 보통			
			600,000u / 120s / 1c	5,000u/1s	10 / 10 / 나쁨			
Scale UP	MAX	CPU:1.5 Memory: 1024 Mi	1,200,000u / 120s / 1c	10,000u/1s	10 / 4 / 안정	약 1초당 3000명 수용 예상		
			120,000u / 120s / 1c	1,000u/1s	10 / 6 / 안정			
	MIN	CPU:0.5 Memory: 512 Mi	240,000u / 120s / 1c	2,000u/1s	10 / 8 / 안정			
			360,000u / 120s / 1c	3,000u/1s	10 / 7 / 보통			
				600,000u / 120s / 1c	5,000u/1s			
				1,200,000u / 120s / 1c	10,000u/1s			

* Scale UP = Pod의 사양을 증가시키고, Max Replicas 수는 유지

프로젝트 시연 및 결과

JMETER를 활용한 성능 측정

시나리오 B: **다수의 Pod에서 몇 명의 유저까지 수용이 가능한가?**

*** Max scale Replicas : 30

Scenario B	Pod Spec		Users/Seconds /Loop Count	Expected user load	Max Replicas /Pod/Result	Pod 운용
-	MAX	CPU:1 Memory: 512 Mi	240,000u / 120s / 1c 360,000u / 120s / 1c	2,000u/1s 3,000u/1s	30 / 12 / 안정 30 / 20 / 안정	약 1초당 3000명 수용 예상
		MIN	CPU:0.2 Memory: 256 Mi	600,000u / 120s / 1c 1,200,000u / 120s / 1c	5,000u/1s 10,000u/1s	
	MAX	CPU:1.5 Memory: 1024 Mi	240,000u / 120s / 1c 360,000u / 120s / 1c	2,000u/1s 3,000u/1s	30 / 4 / 안정 30 / 6 / 안정	약 1초당 3000명 수용 예상
		MIN	CPU:0.5 Memory: 512 Mi	600,000u / 120s / 1c 1,200,000u / 120s / 1c	5,000u/1s 10,000u/1s	

* Scale UP = Pod의 사양을 증가시키고, Max Replicas 수는 유지

프로젝트 시연 및 결과

JMETER를 활용한 성능 측정

노드별 CPU 사용량



노드별 Memory 사용량



02

다수의 Pod에서
몇 명의 유저까지
수용이 가능한가?

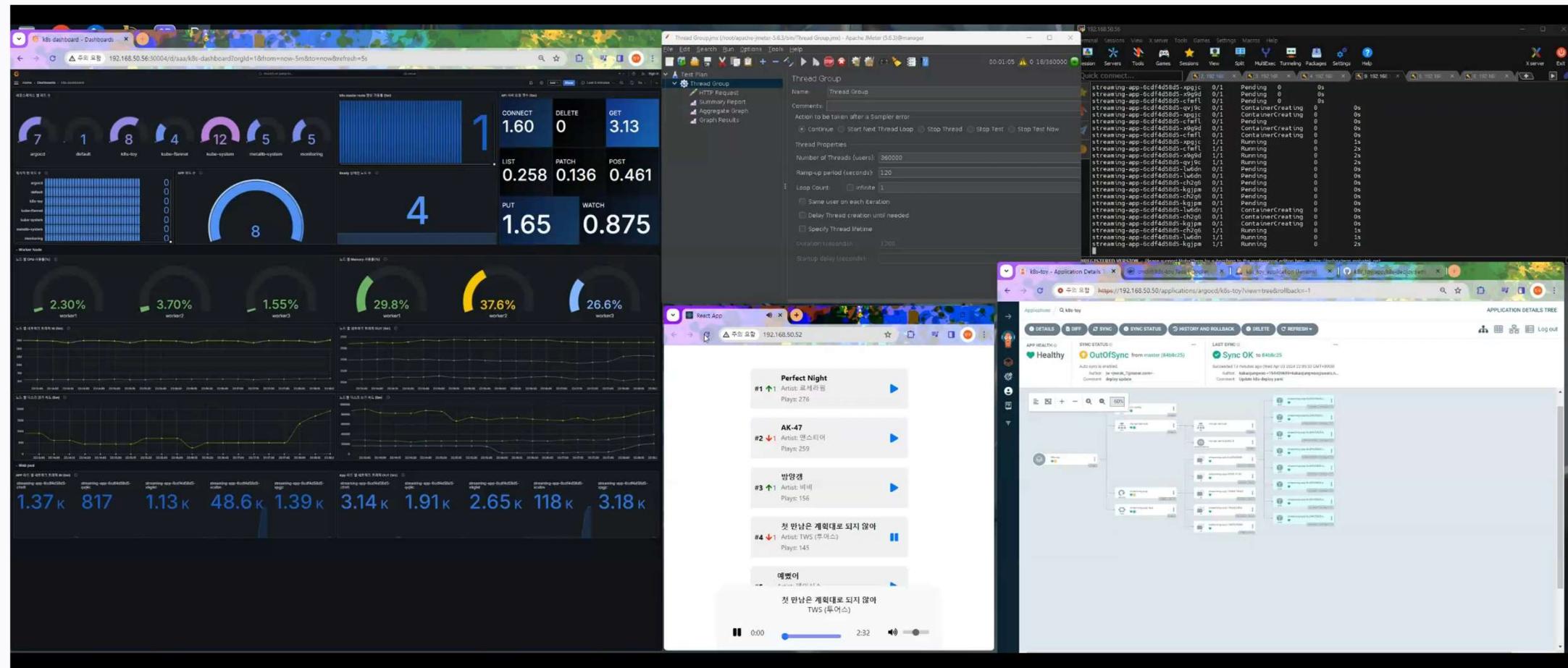
* 1,200,000u/120s/1c

시나리오 B

* 중간 값 기준으로 측정함

프로젝트 시연 및 결과

JMETER를 활용한 성능 측정



테스트 케이스

구분	테스트 시나리오	Pod Spec.		Users	Seconds	Loop Count	예상 유저	Replicas	Auto Scale	Result	pod	Test Result
		min	max									
1	한 개의 Pod에서 몇 명의 유저까지 수용이 가능한가?	cpu: "0.2" memory: "256Mi"	cpu: "1" memory: "512Mi"	6,000	120	1	1초당 50명	1	-	안정	1	매트릭 수치가 안정적이며, 정상적으로 Streaming Service를 제공한다.
				12,000	120	1	1초당 100명	1	-	보통	1	매트릭 수치가 안정적이지만, 파드가 <u>간헐적으로 Pending 상태</u> 를 오고 간다.
				종합 결과							한 개의 파드만을 운용하였을 때 <u>약 1초당 70~80명을 수용</u> 이 예상된다.	
		cpu: "0.5" memory: "512Mi"	cpu: "1.5" memory: "1024Mi"	6,000	120	1	1초당 50명	1	-	안정	1	매트릭 수치가 안정적이며, 정상적으로 Streaming Service를 제공한다.
				12,000	120	1	1초당 100명	1	-	안정	1	매트릭 수치가 안정적이며, 정상적으로 Streaming Service를 제공한다.
				18,000	120	1	1초당 150명	1	-	안정	1	매트릭 수치가 안정적이며, 정상적으로 Streaming Service를 제공한다.
				종합 결과							한 개의 파드만을 운용하였을 때 <u>약 1초당 130~150명을 수용</u> 이 예상된다.	

테스트 케이스

구분	테스트 시나리오	Pod Spec.		Users	Seconds	Loop Count	예상 유저	Replicas	Auto Scale	Result	pod	Test Result	
		min	max										
2 다수의 Pod에서 몇 명의 유저까지 수용이 가능한가?	cpu: "0.2" memory: "256Mi" cpu: "1" memory: "512Mi"	120,000 240,000 360,000 600,000 1,200,000 240,000 360,000 600,000 1,200,000	120 120 120 120 120 120 120 120 120	1 1 1 1 1 1 1 1 1	1초당 1000명 1초당 2000명 1초당 3000명 1초당 5000명 1초당 10000 1초당 2000명 1초당 3000명 1초당 5000명 1초당 10000	1 1 1 1 1 30 30 30 30	10 10 10 10 10 30 30 30 30	안정 보통 보통 보통 나쁨 안정 안정 보통 나쁨	6 10 10 10 10 12 20 19 19	매트릭 수치가 안정적이며, Scale UP/down이 정상적으로 수행된다. 매트릭 수치가 안정적이나, Scale UP(정상) down(약 2~3분 지연)이 발생된다. 매트릭 수치가 안정적이나, Scale UP(정상) down(약 5분 지연)이 발생된다. 매트릭 수치가 안정적이나, Scale UP(정상) down(약 5분 지연)이 발생된다. 파드가 지속적으로 Pending 상태를 오고가며, Scale down(약 10분 이상 지연)이 발생한다. 매트릭 수치가 안정적이며, Scale UP/down이 정상적으로 수행된다. 매트릭 수치가 안정적이나, Scale UP(정상) Scale down(약 2~3분 지연)이 발생된다. 매트릭 수치가 안정적이나, Scale UP(정상) Scale down(약 7~10분 지연)이 발생된다. 매트릭 수치가 안정적이나, Scale UP(정상) Scale down(약 15분 지연)이 발생된다.			
										종합 결과			다 수(HPA Max 10)의 파드를 운용하였을 때 약 1초당 1000명을 수용이 예상된다. 다 수(HPA Max 30)의 파드를 운용하였을 때 약 1초당 3000명을 수용이 예상된다.
		120,000 240,000 360,000 600,000 1,200,000 240,000 360,000 600,000 1,200,000	120 120 120 120 120 120 120 120 120	1 1 1 1 1 1 1 1 1	1초당 1000명 1초당 2000명 1초당 3000명 1초당 5000명 1초당 10000 1초당 2000명 1초당 3000명 1초당 5000명 1초당 10000	1 1 1 1 1 30 30 30 30	10 10 10 10 10 30 30 30 30	안정 안정 안정 보통 보통 4 6 8 9	매트릭 수치가 안정적이며, Scale UP/down이 정상적으로 수행된다. 매트릭 수치가 안정적이며, Scale UP/down이 정상적으로 수행된다. 매트릭 수치가 안정적이며, Scale UP(정상)/down(약 5분 지연)이 정상적으로 수행된다 매트릭 수치가 안정적이며, Scale UP(정상)/down(약 5분 지연)이 정상적으로 수행된다 매트릭 수치가 안정적이며, Scale UP(정상)/down(약 10분 지연)이 정상적으로 수행된다. 매트릭 수치가 안정적이며, Scale UP/down이 정상적으로 수행된다. 매트릭 수치가 안정적이며, Scale UP/down이 정상적으로 수행된다. 매트릭 수치가 안정적이며, Scale UP(정상)/down(약 5분 지연)이 정상적으로 수행된다 매트릭 수치가 안정적이며, Scale UP(정상)/down(약 5분 지연)이 정상적으로 수행된다				
										종합 결과			다 수(HPA Max 10)의 파드를 운용하였을 때 약 1초당 3000명 수용이 예상된다. 다 수(HPA Max 30)의 파드를 운용하였을 때 약 1초당 3000명 수용이 예상된다.

5

Next step



Next Step

프로젝트 개선 사항

노드 오토스케일링

AWS@카펜터를 활용하여,
Pod 뿐만 아니라 노드 HPA
서비스를 제공

도메인 사용

테스트 환경을 넘어서
누구나 접속 가능하도록
도메인과 공인 IP를 구매하여
실제 서비스를 배포

Web Application

회고

“

아쉬운 점

혼자 웹 개발을 맙아서 개발 협업을 경험해보지 못함!
스트리밍 서비스에서 발생할 수 있는 여러 시나리오들을
테스트해보지 못해서 아쉬움!

프로젝트 기간 내 인프라 환경 구성 안정화를 위하여,
JMETER 성능 측정 시나리오의 다양성과 구체화 부분
을 보완하지 못한 점

EKS까지 사용해보지 못한 점

”

“

배운 점

개발한 소스들을 직접 배포해보면서 경험 UP!
프론트에서 자주 쓰는 리액트와 친해짐

성능 측정과 최적화 작업에 대한 접근 방식을
개선하는데 귀중한 통찰력을 경험함

”

Web Application

회고

“

아쉬운 점

프로젝트 초반 작업 시
Git Repository를
팀원들과 공유하지 못한 점

인프라 구성도 초안 설계를
충분히 고민하는 시간이 부족했던 점

”

“

배운 점

초기 단계 명확한 소통 구조와
공유 메커니즘 설정의 중요성

더 많은 시간을 할애하여
다양한 설계 옵션을 검토하는 것이
프로젝트의 성공적인 결과에 기여함을 배움

”

감사합니다!

**THANK
YOU**

RAPA KAKAO Cloud School 2nd Toy Project

Eucalyptus DevOps Team

부록

Storage → AWS S3

The screenshot shows the AWS S3 console interface. On the left, there's a sidebar with various navigation options like 'Access Grants', 'Storage Lens', and 'AWS Marketplace'. The main area displays a bucket named 'songs/'. Inside the bucket, there are five objects listed:

이름	유형
song1.mp3	mp3
song2.mp3	mp3
song3.mp3	mp3
song4.mp3	mp3
song5.mp3	mp3

Advantage

1. 내구성과 가용성
2. 확장성
3. 접근성
4. 보안성

부록

DB → AWS RDS

The screenshot shows the Amazon RDS console interface. On the left, there's a sidebar with various navigation options: 대시보드 (Dashboard), **데이터베이스** (Database) which is highlighted with a red box, 쿼리 편집기 (Query Editor), 성능 개선 도우미 (Performance Advisor), 스냅샷 (Snapshot), Amazon S3에서 내보내기 (Export to S3), 자동 백업 (Automatic Backup), 예약 인스턴스 (Reserved Instances), and 프록시 (Proxy). The main content area has a banner about Aurora I/O 최적화 소개 (Aurora I/O Optimization Introduction). Below it, there's a note about minimizing downtime during upgrades. The '데이터베이스 (1)' section lists a single database named 'streaming-app'. The database row includes columns for DB 식별자 (Identifier), 상태 (Status), and 인스턴스 (Instances). The 'streaming-app' row is selected, indicated by a red box around its identifier.

Advantage

1. 관리 용이성
2. 확장성
3. 고가용성 및 재해 복구
4. 보안

부록

구분	사용 기술 및 스택	목적	버전
OS	Ubuntu	서비스 지원 기간이 긴 Ubuntu를 선정	20.04 LTS
모니터링	Grafana	Metric 정보를 시각화하며, 모니터링을 위한 대시보드를 제공한다.	
	Prometheus & PromQL	Metric 정보를 pull로 가져오며, 원하는 정보를 필터링한다.	
	node-exporter	Node Metric 정보를 수집한다.	
	cAdvisor	Pod Metric 정보를 수집한다.	
	metic-server	Metric 정보를 수집한다.	
성능부하	JMeter	오픈 소스 기반인 JMeter를 활용하여 웹 서버 성능을 측정한다.	
CI/CD		Pipe line을 구축하여, 자동화 배포 환경을 구성한다.	
	Jenkins	트리거를 통해 git 변경점, 스케줄 등을 확인하여 자동 배포하며, 주된 역할은 도커를 활용한 이미지 생성 및 K8S Pod 배포이다.	
	SonarQube	정적코드분석을 수행하며, git의 업로드된 문서의 대한 코드를 분석한다.	
	Argo CD	Pod를 배포하거나, Pod 배포 현황을 시각화 한다.	
네트워크	ngrok	Ngrok은 로컬 서버를 인터넷 상에서 접근 가능하게 해주는 터널링 서비스입니다. 보통 개발자들이 로컬에서 개발한 웹 애플리케이션을 외부에서 테스트하거나 공개적으로 접근할 수 있도록 할 때 사용 됩니다.	
DB Storage	AWS RDS	관리형 관계형 데이터베이스 서비스로, 쉽고 효율적으로 데이터베이스를 설정, 운영, 그리고 확장할 수 있게 한다.	
	AWS S3	확장성이 뛰어난 객체 스토리지 서비스로, 안전하게 대량의 데이터를 저장하고 검색할 수 있게 한.	
app	music streaming service	음원 스트리밍 서비스를 제공한다.	
	react	music streaming service의 Front-End를 담당한다.	
	node.js	music streaming service의 Back-End를 담당한다.	
	mariadb	music streaming service의 데이터베이스를 담당한다.	
패키징 및 패키징 관리	Docker	Pod의 배포하기위한 image를 생성한다.	20.10.24
	Docker Hub	생성된 image를 Public Docker Registry의 업로드 한다.	-
버전 관리	GitHub	소스코드 파일 집합체를 Git의 업로드 한다.	-
실행	Kubernetes	application을 경량화된 pod로 배포, 삭제, HPA 등 전반적인 provisioning을 담당한다.	1.28.8
협업	Slack	팀 내 채널을 개설하여, 프로젝트 진행 상황을 공유한다. 또한, CI/CD 배포 이후 notification 알림을 담당한다.	-
	Google WorkSpace	프로젝트 관련 산출물을 작성한다.	-

부록

1. 테스트 환경

- 시스템 구성이 완료된 상태에서 jmeter를 활용하여 성능 테스트를 수행한다.

2. 테스트 목적

- 자사의 음악 스트리밍 서비스가 사용자에게 효율적으로 제공하기 위해 테스트 케이스를 작성하여 테스트를 수행한다.
- SaaS 서비스로의 전환을 고려하여 서비스가 수용할 수 있는 사용자 수와 관련된 가용성 측면에서 테스트를 수행한다.

3. 테스트 결과

- 테스트 기준은 노드의 리소스 사용량이 여유가 있는 가정하에 진행하였으며, 테스트 수행을 통해 유의미한 결과를 알게 되었다. (자세한 테스트 결과는 아래 이미지를 첨부)
- 현재 구성된 클라우드 인프라 환경 구성에서는 Scale Out 보다 Scale Up이 더 효율적인 시스템 구성인 것을 알게 되었다.

*Scale OUT = Pod의 사양은 동일하게 유지하고, AutoScale 수를 증가시킨다.

*Scale UP = Pod의 사양을 향상시키고, AutoScale 수는 유지한다.

*Pod의 관점에서는 Scale Up이 효율적 (적은 파드수로 동일한 트래픽을 처리하기 때문)

*Node의 관점에서는 Scale UP, Scale OUT 동일한 사용자를 처리

4. 테스트 종합 결과

결론적으로 현 인프라 환경에서는 매트릭 수치로는 안정적이나 네트워크 기준으로 볼 때 1초당 약 3,000명의 사용자에게 서비스를 제공할 수 있는 것을 확인하였습니다. 서버의 사양은 충분하였으나 1Gbps의 네트워크 속도에서는 예상보다 많은 사용자 처리가 어려웠으며, 과도한 트래픽이 발생하는 순간에 병목 현상이 나타났습니다. JMeter는 정해진 시간 내에 트래픽을 처리하기 위해 계속 시도하며, 트래픽 증가로 인한 파드의 증가와 감소 패턴을 반복하는 현상을 관찰할 수 있었습니다.

부록

구분	테스트 시나리오	Pod Spec.		Users	Seconds	Loop Count	예상 유저	Replicas	Auto Scale	Result	pod	Test Result	
		min	max										
1 한 개의 Pod에서 몇 명의 유저까지 수용이 가능한가?	cpu: "0.2" memory: "256Mi"	cpu: "1" memory: "512Mi"		6,000	120	1	1초당 50명	1	-	안정	1	매트릭 수치가 안정적이며, 정상적으로 Streaming Service를 제공한다.	
				12,000	120	1	1초당 100명	1	-	보통	1	매트릭 수치가 안정적이지만, 파드가 <u>간헐적으로 Pending 상태</u> 를 오고 간다.	
				18,000	120	1	1초당 150명	1	-	나쁨	1	파드가 지속적으로 Pending 상태를 오고 간다.	
				종합 결과						한 개의 파드만을 운용하였을 때 <u>약 1초당 70~80명을 수용</u> 이 예상된다.			
	cpu: "0.5" memory: "512Mi"	cpu: "1.5" memory: "1024Mi"		6,000	120	1	1초당 50명	1	-	안정	1	매트릭 수치가 안정적이며, 정상적으로 Streaming Service를 제공한다.	
				12,000	120	1	1초당 100명	1	-	안정	1	매트릭 수치가 안정적이며, 정상적으로 Streaming Service를 제공한다.	
				18,000	120	1	1초당 150명	1	-	안정	1	매트릭 수치가 안정적이며, 정상적으로 Streaming Service를 제공한다.	
				종합 결과						한 개의 파드만을 운용하였을 때 <u>약 1초당 130~150명을 수용</u> 이 예상된다.			

부록

구분	테스트 시나리오	Pod Spec.		Users	Seconds	Loop Count	예상 유저	Replicas	Auto Scale	Result	pod	Test Result	
		min	max										
2 다수의 Pod에서 몇 명의 유저까지 수용이 가능한가?	cpu: "0.2" memory: "256Mi"	cpu: "1" memory: "512Mi"		120,000	120	1	1초당 1000명	1	10	안정	6	매트릭 수치가 안정적이며, Scale UP/down이 정상적으로 수행된다.	
				240,000	120	1	1초당 2000명	1	10	보통	10	매트릭 수치가 안정적이나, Scale UP(정상) down(약 2~3분 지연)이 발생된다.	
				360,000	120	1	1초당 3000명	1	10	보통	10	매트릭 수치가 안정적이나, Scale UP(정상) down(약 5분 지연)이 발생된다.	
				600,000	120	1	1초당 5000명	1	10	보통	10	매트릭 수치가 안정적이나, Scale UP(정상) down(약 5분 지연)이 발생된다.	
				1,200,000	120	1	1초당 10000명	1	10	나쁨	10	파드가 지속적으로 Pending 상태를 오고가며, Scale down(약 10분 이상 지연)이 발생한다.	
				240,000	120	1	1초당 2000명	1	30	안정	12	매트릭 수치가 안정적이며, Scale UP/down이 정상적으로 수행된다.	
				360,000	120	1	1초당 3000명	1	30	안정	20	매트릭 수치가 안정적이나, Scale UP(정상) Scale down(약 2~3분 지연)이 발생된다.	
				600,000	120	1	1초당 5000명	1	30	보통	19	매트릭 수치가 안정적이나, Scale UP(정상) Scale down(약 7~10분 지연)이 발생된다.	
				1,200,000	120	1	1초당 10000명	1	30	나쁨	19	매트릭 수치가 안정적이나, Scale UP(정상) Scale down(약 15분 지연)이 발생된다.	
				종합 결과								다 수(HPA Max 10)의 파드를 운용하였을 때 약 1초당 1000명을 수용이 예상된다. 다 수(HPA Max 30)의 파드를 운용하였을 때 약 1초당 3000명을 수용이 예상된다.	
2 다수의 Pod에서 몇 명의 유저까지 수용이 가능한가?	cpu: "0.5" memory: "512Mi"	cpu: "1.5" memory: "1024Mi"		120,000	120	1	1초당 1000명	1	10	안정	4	매트릭 수치가 안정적이며, Scale UP/down이 정상적으로 수행된다.	
				240,000	120	1	1초당 2000명	1	10	안정	6	매트릭 수치가 안정적이며, Scale UP/down이 정상적으로 수행된다.	
				360,000	120	1	1초당 3000명	1	10	안정	8	매트릭 수치가 안정적이며, Scale UP(정상)/down(약 5분 지연)이 정상적으로 수행된다.	
				600,000	120	1	1초당 5000명	1	10	보통	7	매트릭 수치가 안정적이며, Scale UP(정상)/down(약 5분 지연)이 정상적으로 수행된다.	
				1,200,000	120	1	1초당 10000명	1	10	보통	9	매트릭 수치가 안정적이며, Scale UP(정상)/down(약 10분 지연)이 정상적으로 수행된다.	
				240,000	120	1	1초당 2000명	1	30	안정	4	매트릭 수치가 안정적이며, Scale UP/down이 정상적으로 수행된다.	
				360,000	120	1	1초당 3000명	1	30	안정	6	매트릭 수치가 안정적이며, Scale UP/down이 정상적으로 수행된다.	
				600,000	120	1	1초당 5000명	1	30	보통	7	매트릭 수치가 안정적이며, Scale UP(정상)/down(약 5분 지연)이 정상적으로 수행된다.	
				1,200,000	120	1	1초당 10000명	1	30	보통	6	매트릭 수치가 안정적이며, Scale UP(정상)/down(약 5분 지연)이 정상적으로 수행된다.	
				종합 결과								다 수(HPA Max 10)의 파드를 운용하였을 때 약 1초당 3000명 수용이 예상된다. 다 수(HPA Max 30)의 파드를 운용하였을 때 약 1초당 3000명 수용이 예상된다.	