



Kubernetes 3-Tier 고가용성 시스템 구축과 개발자를 위한 올인원 협업 플랫폼 개발

2025-01-09

발표자 : 박용현



Gition 개발 배경

> Git + Notion = “Gition”

> 기존의 git 과 Notion이 분리 되어 업무 효율성이 떨어진다고 판단되어

> 개발자를 위해 기존 git와 Notion을 통합하여
“ALL-in-One Collaboration Platform” 을
개발하게 되었습니다.

SteelCrab's Workspace

Repos Search

REPOSITORIES

Search repositories...

All 61 Public 55 Private 6

> pista-megazoncloud

> gition

> pipesql-test

> pipecloud

> pipesql-website

> gluesql

> pipesql

> rs-db-terraform

> mz-student-json

> stopwatch

> crossterm

> jex

> rsmd

> aws-sdk-rust

> Seoul-Doko-Collector-Serivce

> relay

> Programmers_coding_Test

> Heart-rate-monitor

> nushell

> coreutils

> linux_Rust

> coffee-count

> smithy-rs

> kvm

> SteelCrab.github.io

> ia-codysey-personal

> ia-codysey

> ai-codvssev-team

SteelCrab / workspace

Gition

Developer's All-in-One Collaboration Platform

Welcome to gition. This repository demonstrates a block editor and integrated CI/CD pipelines.

CLI installation

```
git clone https://github.com/gition/gition.git
cd gition && pnpm install
pnpm dev
```

Below is an interactive pipeline block. Trigger build manually.

>_ production-rollout (isomorphic-git)

Build

npm install && npm run build

2s

Deploy

aws s3 sync ./dist s3://...

Running...

+ Click to add blocks, or type "/" for commands



Why Gition ?

코드작성 → 문서화 → 배포 도구 사용

문제	기존 방식	Gition
코드와 문서가 분리됨	GitHub + Notion 별도 사용	하나의 워크스페이스
문서 버전 관리 어려움	수동 백업 또는 없음	Git 기반 자동 버전 관리
CI/CD 상태 확인	탭 전환 필요	문서 내 실시간 표시



기술스택

계층	기술
Frontend	React + Vite + TypeScript
Backend	Python FastAPI + GitPython
Database	MySQL InnoDB Cluster
Auth	GitHub OAuth 2.0
Infra	Kubernetes + MetalLB + NFS
CI/CD	GitHub Actions + GitLab CI



기능 요구사항 (1)

인증 및 계정 관리

기능	구현 방식
GitHub OAuth	OAuth 2.0
세션 관리	Token 기반
로그인/로그아웃	localStorage 저장

저장소 관리

기능	설명
저장소 목록	GitHub API 연동
저장소 복제	서버로 Clone
브랜치 전환	로컬/리모트 브랜치
파일 탐색	디렉토리 네비게이션

에디터

기능	설명
블록 에디터	Notion 스타일
마크다운 렌더링	MarkdownRenderer
코드 블록	구문 강조
.gition 저장소	브랜치별 로컬 페이지

Git 연동

기능	설명
커밋 히스토리	브랜치별 조회
파일 뷰어	내용 조회/편집
코드 검색	저장소 내 검색
Issues/PR	GitHub 연동 표시

기능 요구사항 (2)

CI/CD 파이프라인

플랫폼	기능
GitHub Actions	ESLint / PyLint 검사
	Unit Test 실행
	Gitleaks / Trivy 보안 스캔
GitLab CI	GitLab Mirror 동기화
	Docker 이미지 빌드
	Container Registry 푸시
	수동 배포 트리거

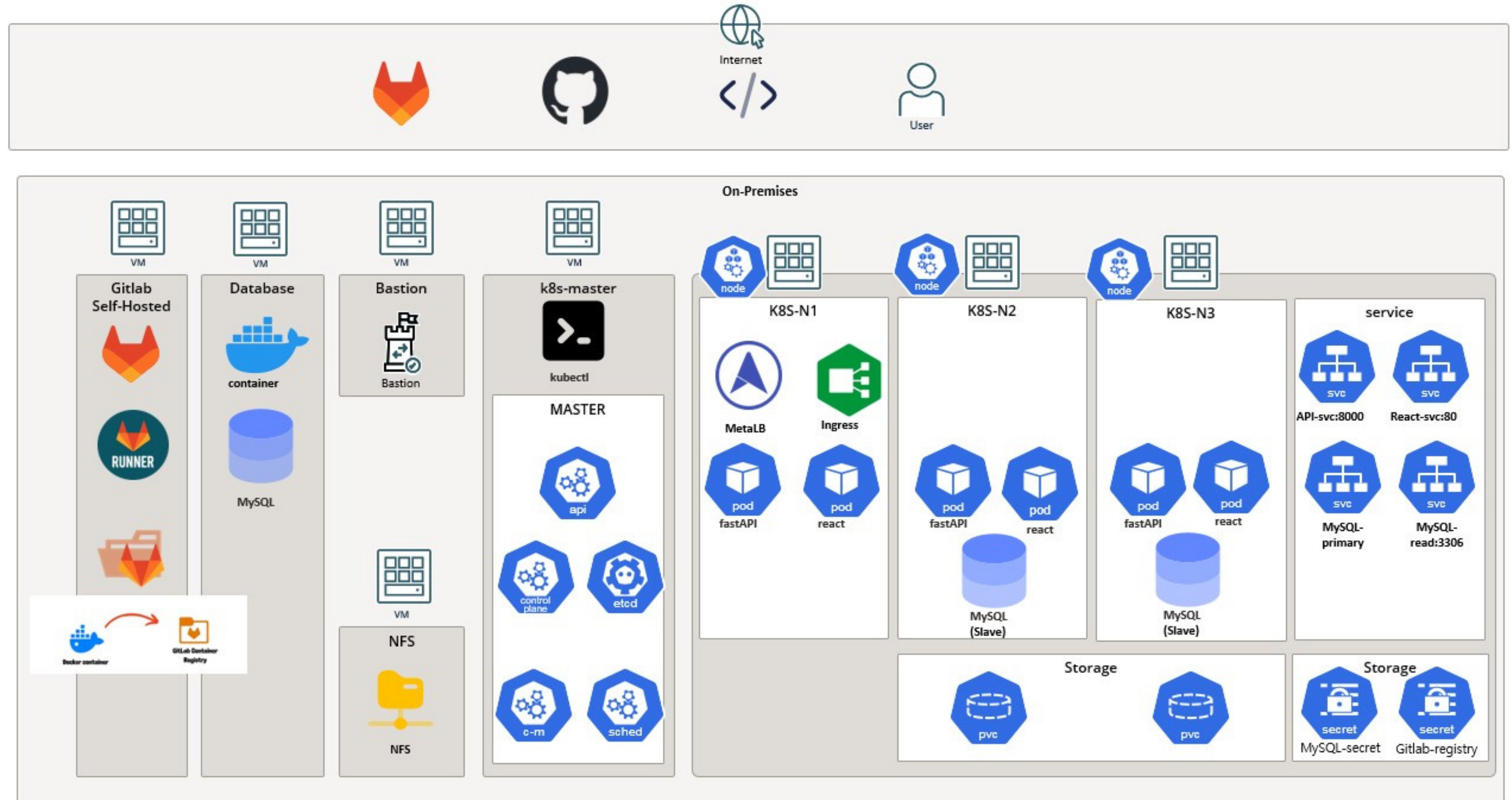
Kubernetes 인프라

구성 요소	설명
Ingress Controller	Host/Path 기반 라우팅
MetalLB	LoadBalancer IP 할당
NFS Provisioner	동적 PVC 생성
FastAPI Deployment	3 replicas + 공유 볼륨
Frontend Deployment	2 replicas (React + Nginx)
MySQL Master	외부 Docker (svc Endpoints)
MySQL Slave StatefulSet	K8s 내 2 replicas (GTID 복제)
imagePullSecrets	GitLab Registry 인증

주요 테스트

Day	날짜	주요 작업	결과물
Day 1	12/24	K8s 클러스터 초기 구축	kubeadm, Calico CNI 설치
Day 2	12/29	기본 인프라 구성	NFS, Docker, K8s manifests
Day 3	12/30	GitLab CI/CD 구축	Runner 설정, .gitlab-ci.yml
Day 4	12/31	Container Registry 연동	insecure-registry 설정
Day 5	01/01	Frontend/Backend 배포	Deployment, Ingress
Day 6	01/02	MySQL Master-Slave 구성	ExternalName → Endpoints 전환
Day 7	01/03	Replication 설정	GTID 복제, server-id 자동화
Day 8	01/04	고가용성 테스트	Failover 검증
Day 9	01/05	OAuth/DB 트러블슈팅	UFW, 복제 계정 생성
Day 10	01/06	Read/Write Split	복제 지연 대응
Day 11	01/07	MetalLB 설정, 문서 정리	LoadBalancer IP 할당
Day 12	01/08	MySQL InnoDB Cluster	Operator, Local PV, Backup CronJob

시스템 구성



리소스 할당

노드	vCPU	RAM	역할
Bastion	2	1GB	SSH Gateway
GitLab	2	4GB	CI/CD, Registry
NFS	1	1GB	Persistent Storage, Backup
MySQL	2	4GB	Master Database (Docker)
k8s-m	4	4GB	Control Plane
k8s-n1	4	4GB	Worker (Frontend, API, MySQL)
k8s-n2	4	4GB	Worker (Frontend, API, MySQL)
k8s-n3	4	4GB	Worker (API, MySQL)



구성(2)

Pod 구성

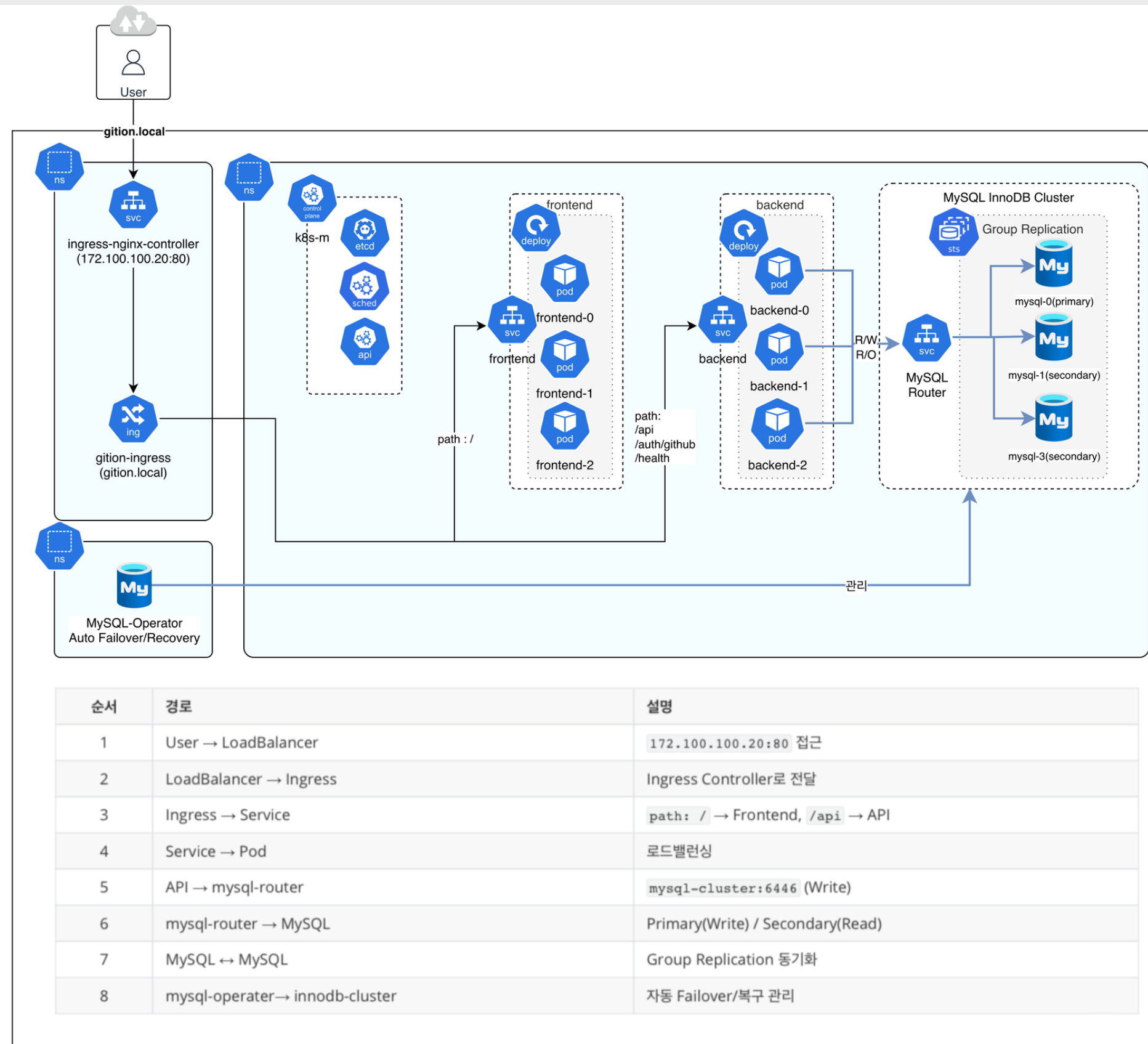
Deployment/StatefulSet	Replicas	Image	Port
frontend	2	172.100.100.8:5050/root/gition/frontend	80
backend	3	172.100.100.8:5050/root/gition/backend	3001
mysql-cluster	3	mysql/mysql-server:8.0	3306, 6446, 6447
mysql-router	2	mysql/mysql-router:8.0	6446, 6447

Ingress 라우팅

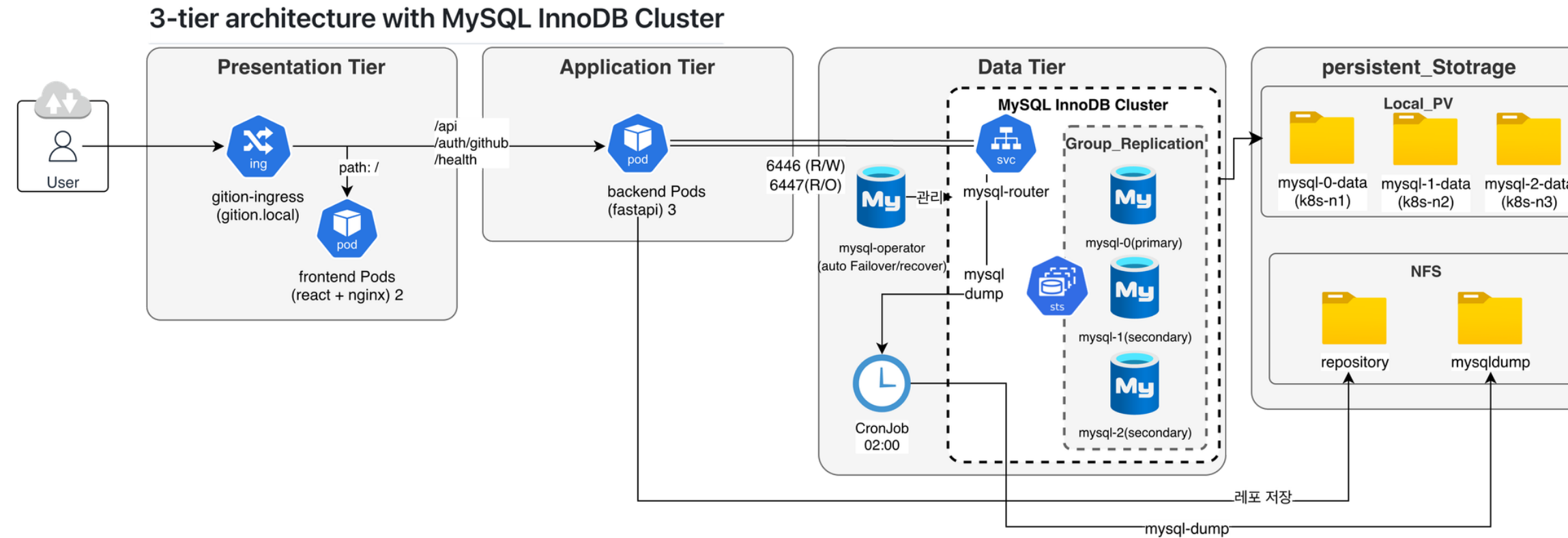
Path	Service	Port
/	frontend-svc	80
/api	api-svc	3001
/auth/github	api-svc	3001
/health	api-svc	3001



architecture (k8s)



architecture (3tier)



순서	경로	설명
1	User → Ingress	gition.local 접속
2	Ingress → Frontend	path: / → React 앱
3	Ingress → API	path: /api, /auth → FastAPI
4	API → mysql-router	:6446 (Write) 또는 :6447 (Read)
5	mysql-router → MySQL	Primary(Write) / Secondary(Read)
6	MySQL ↔ MySQL	Group Replication 동기화
7	MySQL → Local PV	각 노드별 데이터 저장
8	mysql-operator → Cluster	자동 Failover/복구 관리

포트	용도	대상
6446 (R/W)	INSERT, UPDATE, DELETE	Primary (mysql-0)
6447 (R/O)	SELECT	Secondary (mysql-1, mysql-2)

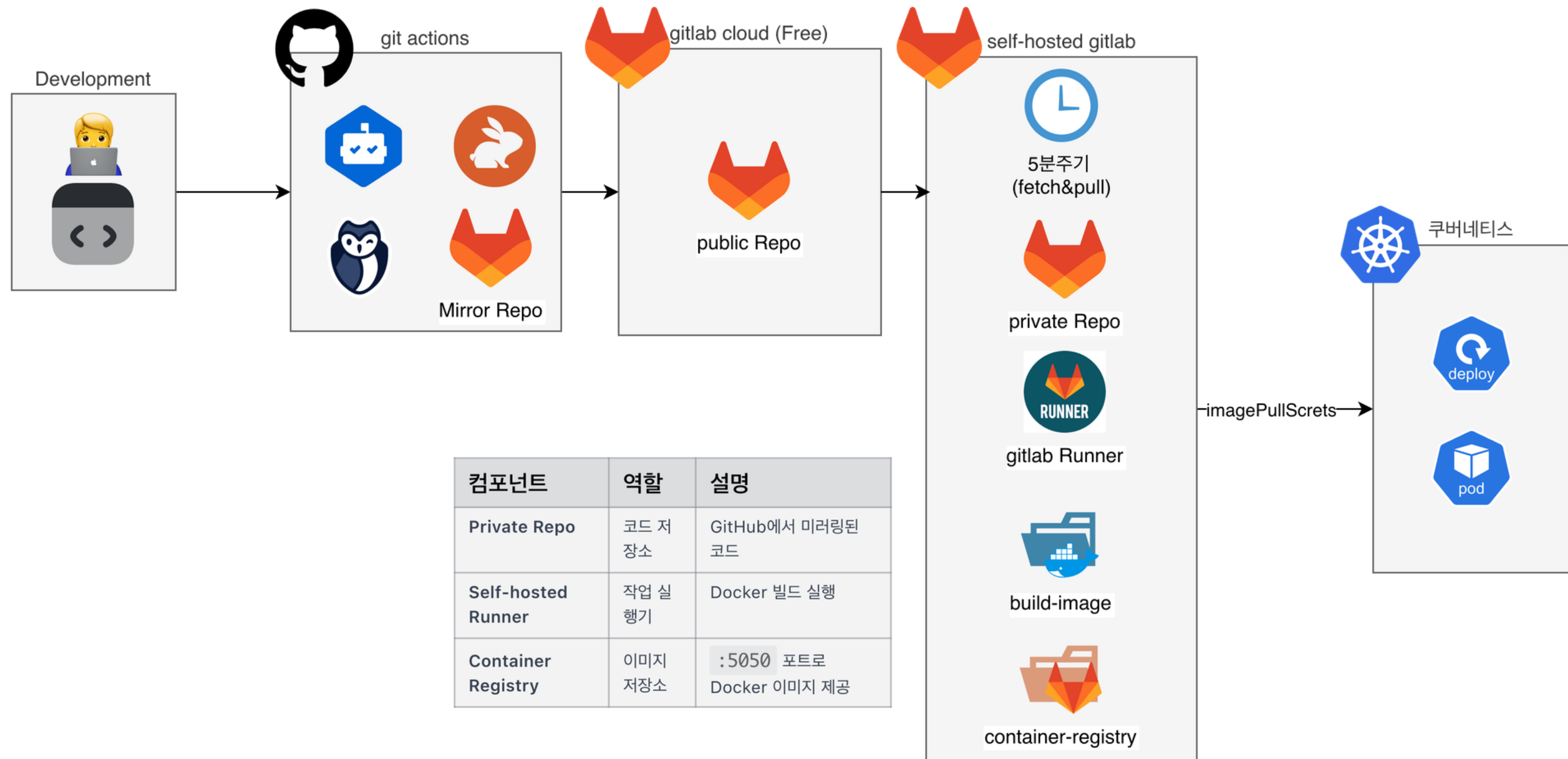


테스트 - MySQL InnoDB Cluster Auto Failover

```
sudo ssh home-gition
home-gition (ssh) ㄹ1
pista@k8s-m:~/gition/k8s$ kubectl exec -it mysql-cluster-0 -n gition -c mysql -- \
mysql -uroot -p -e "SELECT member_host, member_role FROM performance_schema.replicat
ion_group_members"
```

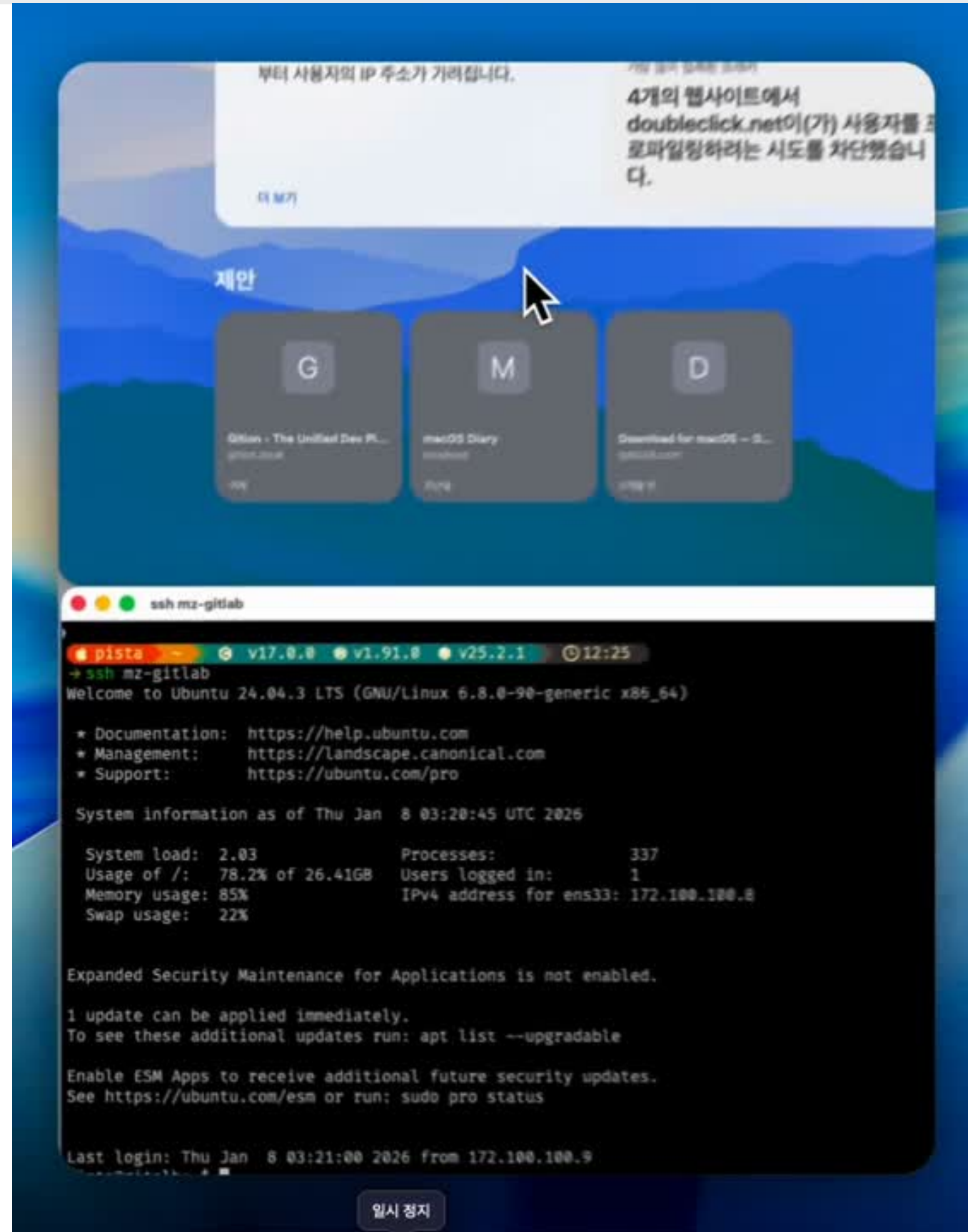


architecture (CI/CD)

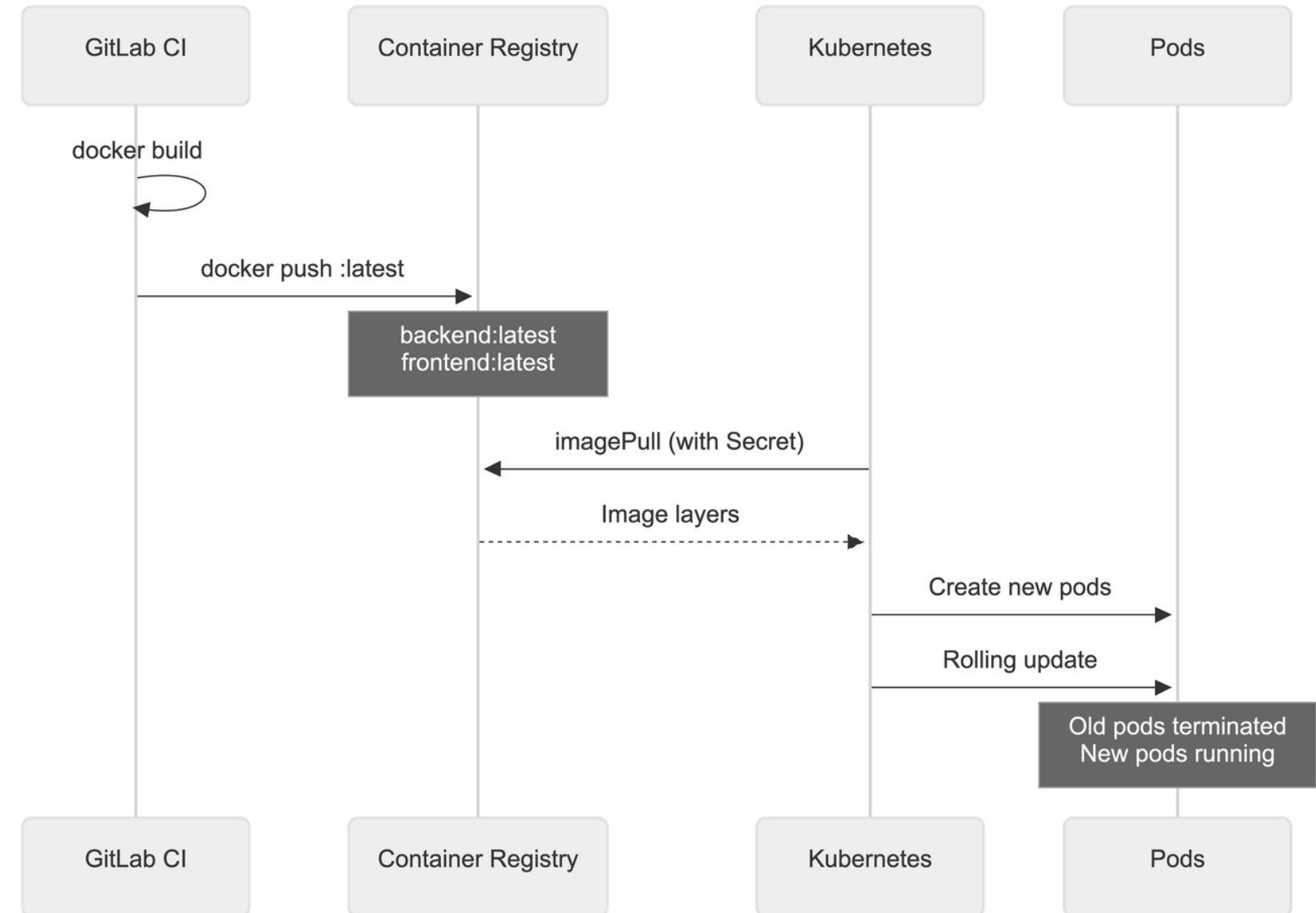
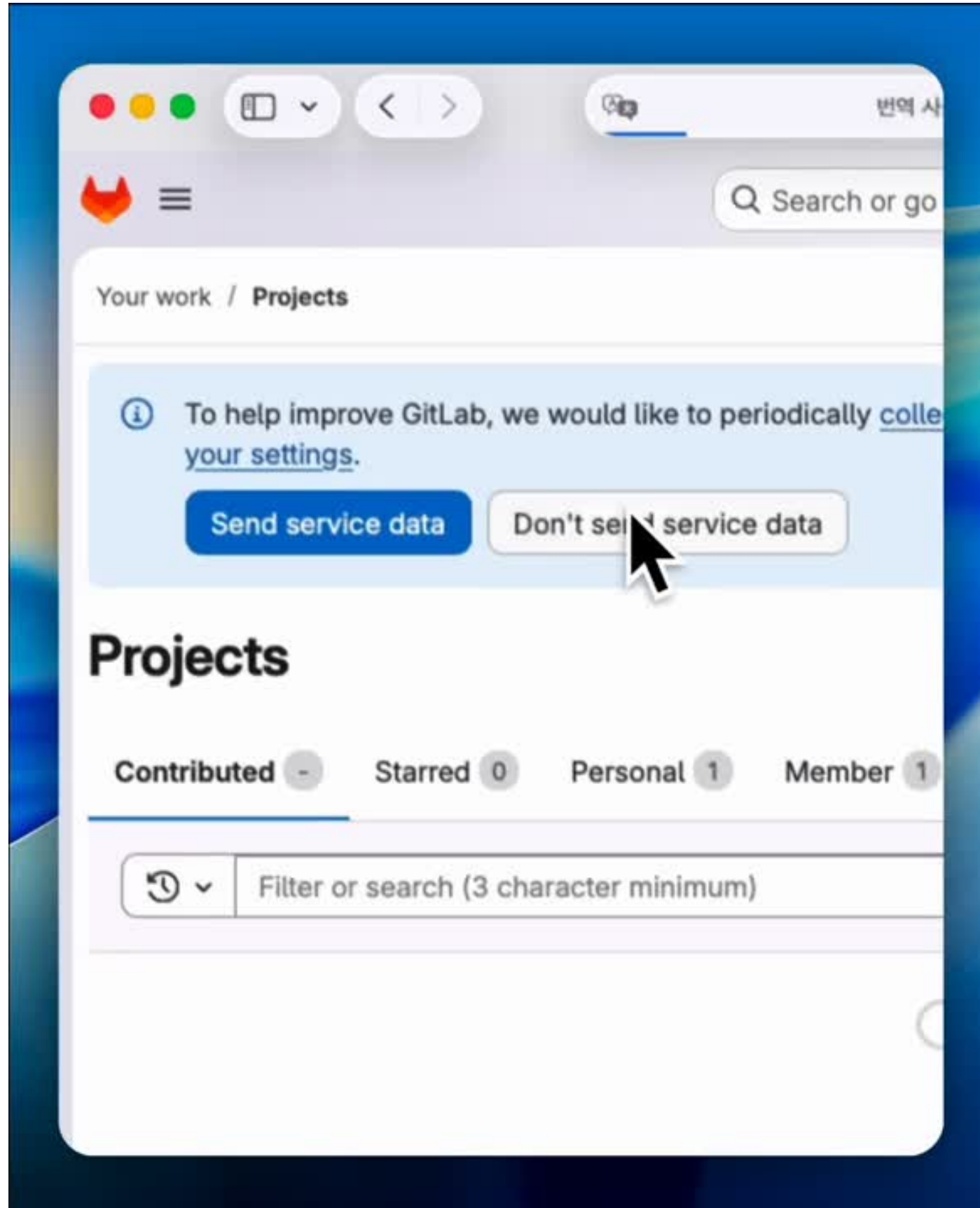




gitlab 로그인



gitlab 배포과정

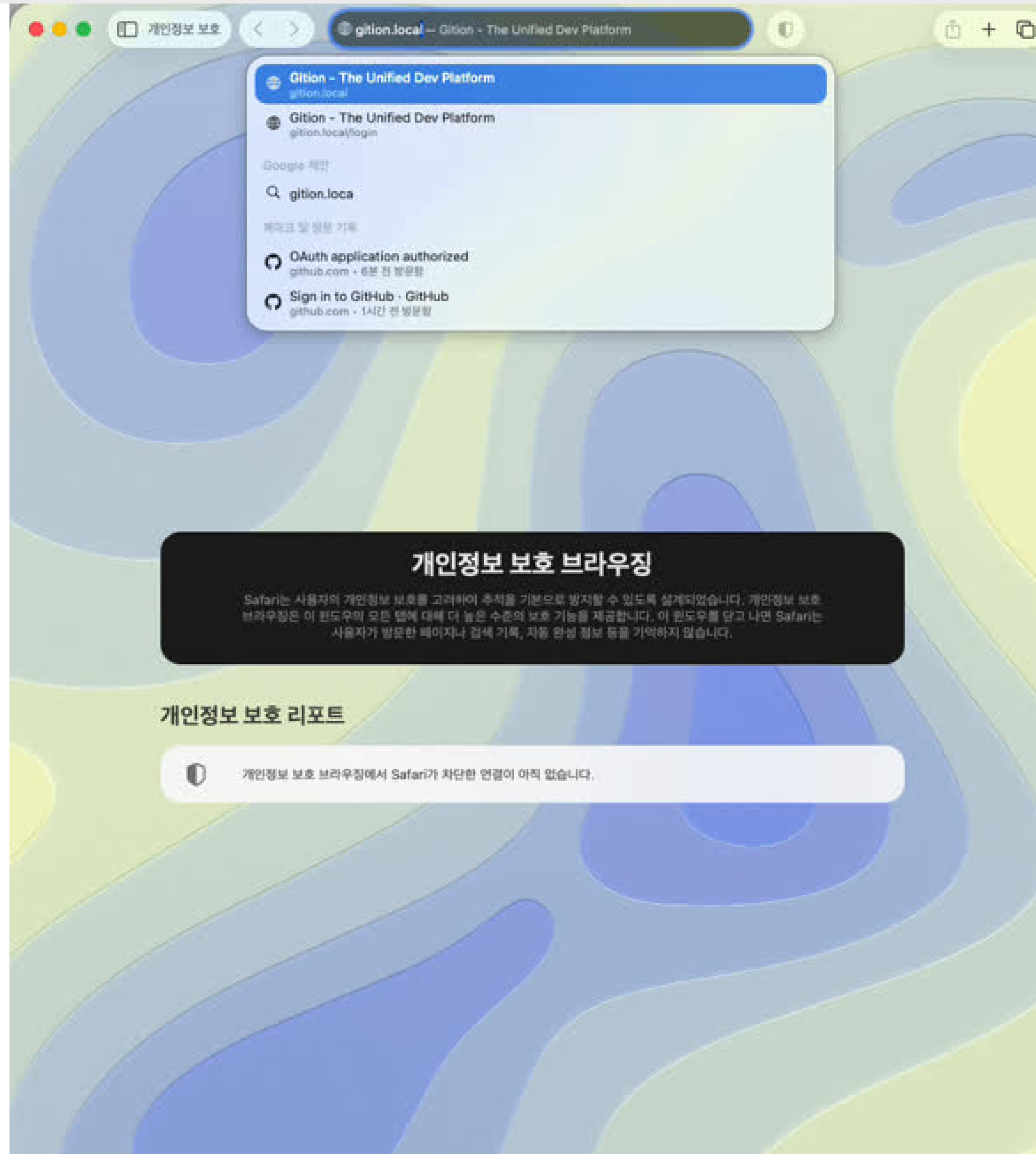


새 이미지를 배포 하는 경우 (frontend,api)

```
kubectl rollout restart deployment/frontend -n gition &&
kubectl rollout status deployment/frontend -n gition
```

```
kubectl rollout restart deployment/api -n gition &&
kubectl rollout status deployment/api -n gition
```

데모 - GitionOauth 연동





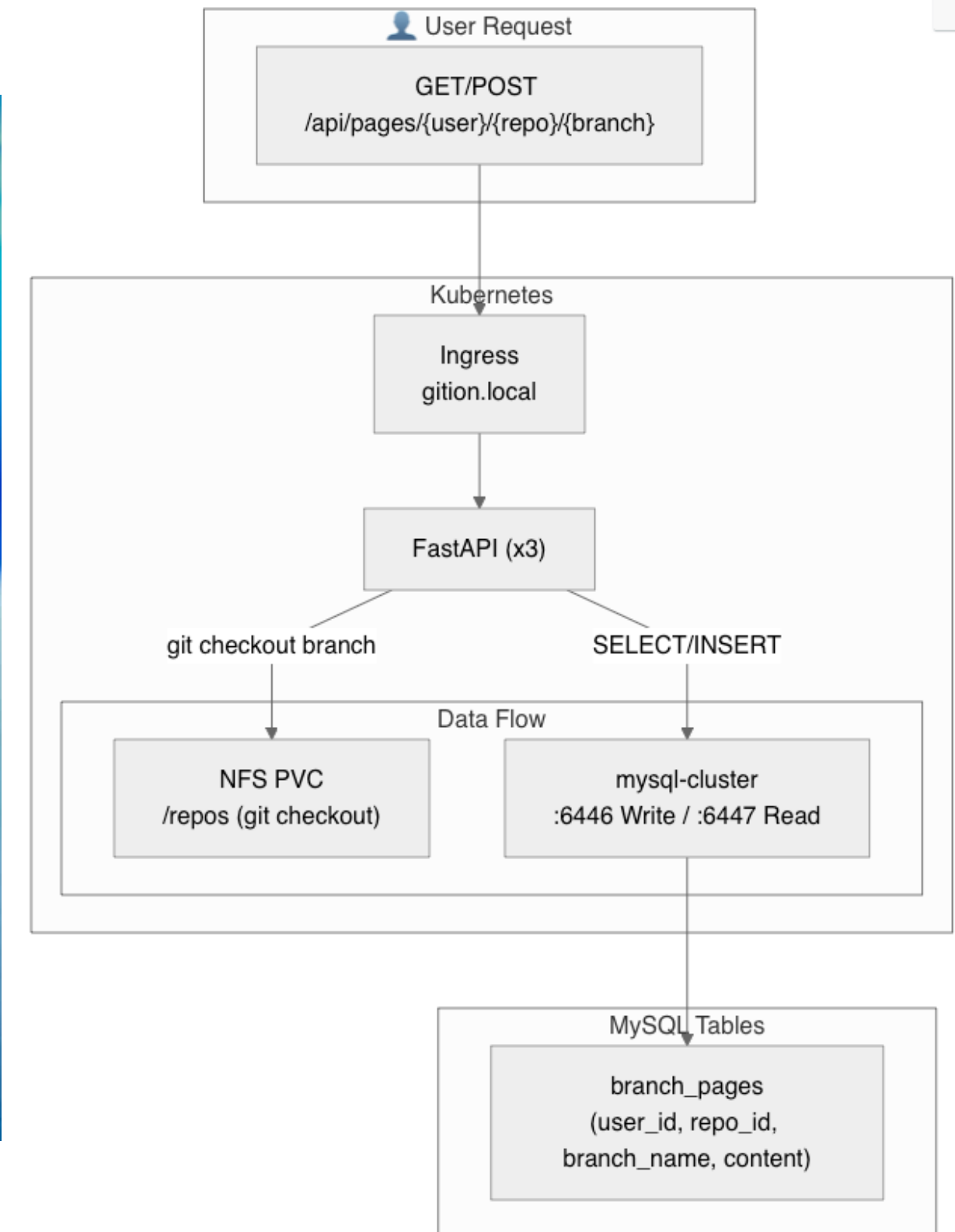
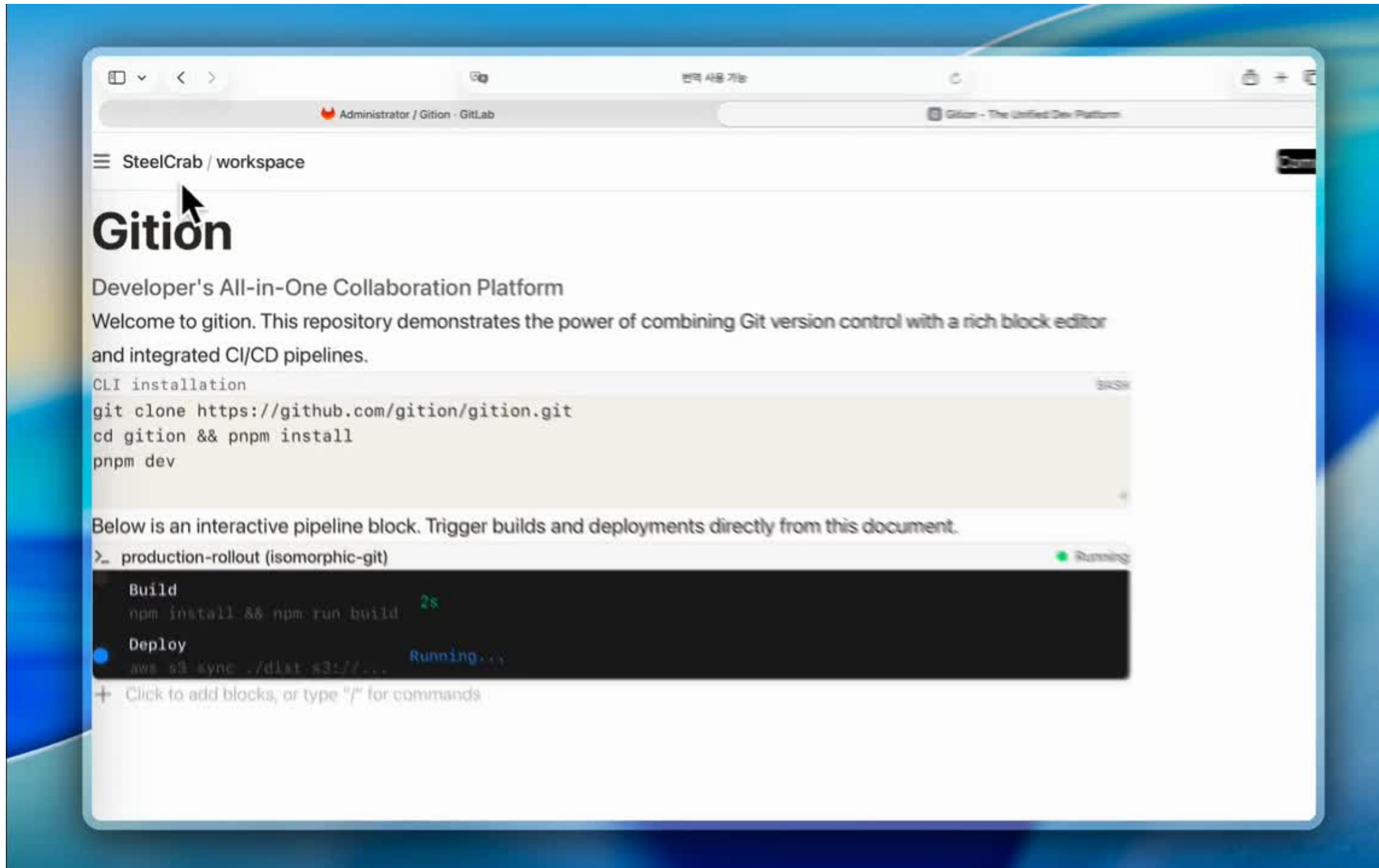
데모 - Gition repository clone

The screenshot displays the Gition web interface in a browser window. The address bar shows 'gition.local'. The page title is 'SteelCrab's Workspace'. The left sidebar, titled 'REPOSITORIES', lists various repositories with status indicators (e.g., 'pista-megazoncloud', 'gition', 'pipesql-test', 'pipesql-website', 'gluesql', 'pipesql', 'rs-db-terraform', 'mz-student-json', 'stopwatch'). The main content area is titled 'Gition' and describes it as a 'Developer's All-in-One Collaboration Platform'. It includes a welcome message and a 'CLI installation' section with the following commands:

```
git clone https://github.com/gition/gition.git
cd gition && npm install
npm dev
```

Below the installation instructions, there is an interactive pipeline block titled 'production-rollout (isomorphic-git)' with a 'Run' button. The block contains two steps: 'Build' (npm install && npm run build) and 'Deploy' (aws s3 sync ./dist s3://...). The 'Deploy' step is currently running, indicated by a 'Running...' status.

데모 - 브랜치노트





데모 - PR과Issues 조회

The screenshot shows the Gition web interface. The left sidebar displays the 'SteelCrab's Workspace' with a 'COMMIT HISTORY' section listing recent commits. The main area shows a file explorer with a list of files and folders.

COMMIT HISTORY

- chore: update gitleaks to v8.30.0 (986d00c Dec 27)
- chore: update gitleaks to v8.30.0 (9e0f9fb Dec 27)
- Update .pre-commit-config.yaml (e3521cd Dec 27)
- Update .pre-commit-config.yaml (e7543ea Dec 27)
- chore: add pre-commit hooks configura... (44cc6f2 Dec 27)
- Merge pull request #75 from SteelCrab/... (810952e Dec 27)
- Merge pull request #73 from SteelCrab/... (d54778e Dec 27)
- docs: update roadmap with branch pag... (982b121 Dec 27)
- Merge pull request #74 from SteelCrab/... (4a174a8 Dec 27)

Files

- .md_kr
- backend
- docs
- frontend
- mysql
- terminal
- .dockerignore
- CONTRIBUTING.md
- docker-compose.yml
- Dockerfile
- eslint.config.js
- index.html
- LICENSE
- nginx.conf
- package-lock.json
- package.json
- README.md
- ROADMAP.md
- tsconfig.json
- tsconfig.node.json
- vite.config.ts



트러블해결 - CI/CD

★ Day 3: Runner에서 GitLab 접근 불가

- 상황: GitLab CI 파이프라인 시작 시 Runner가 소스 코드를 Clone 하지 못함
- 에러: `Could not connect to server`
- 원인: Runner가 GitLab 내부 URL을 해석 못함 (Docker 네트워크 격리)
- 해결: `/etc/gitlab-runner/config.toml` 수정

```
clone_url = "http://172.100.100.8"
network_mode = "host"
```

Day 4: Docker API 버전 오류

- 상황: GitLab CI에서 Docker 빌드 Job 실행 시 즉시 실패
- 에러: `client version 1.43 is too old. Minimum supported API version is 1.44`
- 원인: CI Job의 Docker 클라이언트 이미지와 Runner의 Docker daemon 버전 불일치
- 해결: `.gitlab-ci.yml` 에서 Docker 이미지 업그레이드

```
image: docker:27 # 24.0.5 → 27
```

Day 11: Docker 빌드 시 DNS 해석 실패

- 상황: `docker build` 중 `pip install` 또는 `npm install` 에서 외부 패키지 다운로드 실패
- 에러: `Temporary failure in name resolution`
- 원인: 빌드 컨테이너가 외부 DNS 해석 불가 (pypi.org, registry.npmjs.org 접근 필요)
- 해결: `/etc/docker/daemon.json` 에 Google DNS 추가

```
{"insecure-registries": ["172.100.100.8:5050"], "dns": ["8.8.8.8", "8.8.4.4"]}
```



트러블해결 - kubernetes

★ Day 4: HTTP Registry → HTTPS 오류

- 상황: K8s 노드에서 GitLab Registry 이미지 Pull 시 실패
- 에러: `http: server gave HTTP response to HTTPS client`
- 원인: containerd가 기본적으로 HTTPS로 접근하는데 GitLab Registry는 HTTP 사용
- 해결: 모든 K8s 노드에 containerd 설정 추가

```
sudo tee /etc/containerd/certs.d/172.100.100.8:5050/hosts.toml <<EOF
server = "http://172.100.100.8:5050"
[host."http://172.100.100.8:5050"]
  capabilities = ["pull", "resolve", "push"]
  skip_verify = true
EOF
sudo systemctl restart containerd
```

Day 9: Repository not cloned

- 상황: FastAPI Pod 3개 중 일부만 Git 저장소에 접근 가능, 나머지는 에러
- 에러: `Repository not cloned`
- 원인: 각 Pod가 자신만의 로컬 emptyDir 볼륨 사용 → Pod마다 저장소 상태 불일치
- 해결: NFS 기반 공유 PVC 생성으로 모든 Pod가 동일 저장소 접근

```
volumes:
- name: repos-volume
  persistentVolumeClaim:
    claimName: repos-pvc
```



트러블해결 - Database & Network

Day 9: Replication 실패

- 상황: MySQL Slave Pod가 Master에 복제 연결을 시도하지만 인증 실패
- 에러: `Replica_IO_Running: Connecting`
- 원인: Master DB에 복제 전용 계정이 없음
- 해결: Master에서 복제 계정 생성

```
CREATE USER 'repl_pista'@'%' IDENTIFIED BY '<PASSWORD>';  
GRANT REPLICATION SLAVE ON *.* TO 'repl_pista'@'%';
```

★ Day 10: Duplicate Key 에러

- 상황: 사용자가 페이지 저장 시 간헐적으로 중복 키 에러 발생
- 에러: `IntegrityError: Duplicate entry for key 'unique_user_repo_branch'`
- 원인: 복제 지연 중 Master에서 데이터 생성 → Slave에서 읽기 시 없음 → 다시 생성 시도
- 해결: Read/Write 모두 Master 사용 (임시 조치), 향후 InnoDB Cluster로 전환

Day 9: OAuth 콜백 후 404

- 상황: GitHub 로그인 후 콜백 URL에서 404 에러
- 에러: `/auth/callback` 에서 404 Not Found
- 원인: Ingress가 `/auth/*` 를 모두 백엔드로 라우팅 (프론트엔드 콜백 페이지 접근 불가)
- 해결: `/auth/github/*` 만 백엔드, 나머지는 프론트엔드로 라우팅

진행 현황

- **v0.1 Core Platform:** 85% 완료
- **Infrastructure:** 100% 완료

배운 점

- Kubernetes 클러스터 운영 경험
- MySQL Master-Slave 복제 구성 -> MySQL InnoDB Cluster 구성
- GitHub/GitLab 하이브리드 및 CI/CD 파이프라인 구성



로드맵

v0.1 Core Platform (85%)

기능	상태	설명
GitHub OAuth 연동	O	로그인/로그아웃
저장소 목록/복제	O	GitHub API, Clone
파일 브라우저	O	네비게이션
브랜치 관리	O	전환 기능
커밋 히스토리	O	브랜치별 조회
블록 에디터	O	Notion 스타일
마크다운 렌더링	Δ	대부분 구현
코드 블록 구문 강조	-	Syntax Highlighting

v0.2 Visualization & Features (10%)

☰ ≡ ≡ ≡



기능	상태
커밋 그래프	-
Issue/PR 생성	-
웹 터미널	-
실시간 편집	-

기능	상태
ArgoCD GitOps	-
Helm Chart	-
멀티 클러스터	-

활발한 오픈소스 기여 이슈

☐	 Migrate application storage to MySQL and harden markdown/auth flows Bug fix good first issue #79 · qodo-code-review opened 2 days ago	4	
☐	 Promotion Page Development backend enhancement frontend good first issue #58 · SteelCrab opened 5 days ago · 🏷️ 2_DEADLINE	2	
☐	  Webpage Error 404 - Support Overview enhancement frontend #56 · SteelCrab opened 5 days ago · 🏷️ DEADLINE	1	
☐	 Bug: Logout button not visible on main page bug documentation frontend good first issue #50 · SteelCrab opened last week · 🏷️ DEADLINE		
☐	  Performance Engine enhancement good first issue #11 · SteelCrab opened last week · 🏷️ 2_DEADLINE		
☐	 Pipeline Monitoring enhancement good first issue #10 · SteelCrab opened last week · 🏷️ 2_DEADLINE		
☐	 Bi-directional Integration enhancement good first issue #9 · SteelCrab opened last week · 🏷️ 2_DEADLINE		
☐	  GitHub Actions Status enhancement #2 · SteelCrab opened last week · 🏷️ 2_DEADLINE		

Join Us

Currently, our team uses a private GitLab repository.

If you're interested in contributing or learning more, feel free to reach out via email or open an issue—we'd be happy to invite you!

We're open to AI-assisted contributions at the maintenance level. 😊

Contributing

 Check out our [ROADMAP](#) for a checklist of features you can contribute to! :)

See [CONTRIBUTING.md](#) for guidelines.



**Thank you
for listening!**

발표자: 박용현