

Kubernetes-based Stock Backtesting Platform

1 Backtest = 1 Kubernetes Job



Project Overview

BACKTEST TERMINAL

COMMAND INPUT

CONFIG

> SECURITY

TICKER

AAPL

> STRATEGY

RULE

RSI (Relative Strength : ▾

> PARAMETERS

RSI PERIOD

14

OVERSOLD

30

OVERBOUGHT

70

> DATE RANGE

FROM

2020-01-01

TO

2024-12-31

> TRADING

CAPITAL

100000

FEE RATE (DECIMAL)

0.001

e.g. 0.001 = 0.1%

EXECUTE BACKTEST

> PRESETS

-- NO PRESET --

SAVEDELETE

SYS K8s Backtest Platform v1.0

Backtest completed successfully!

RESULTS

RUN 1cc34935

STATS

EQUITY

DRAWDOWN

PORTFOLIO

TRADES

TOTAL RETURN

12.39%

SHARPE RATIO

0.14

MAX DRAWDOWN

30.30%

TOTAL TRADES

14

FINAL VALUE

\$112,394

WIN RATE

64.3%

SORTINO RATIO

0.13

PROFIT FACTOR

2.85

입력종목, 전략, 기간, 파라미터

실행백테스트 Job 생성 및 수행

출력수익률, 거래 수, 낙폭, 리스크 지표

01

Problem & Goals

Before

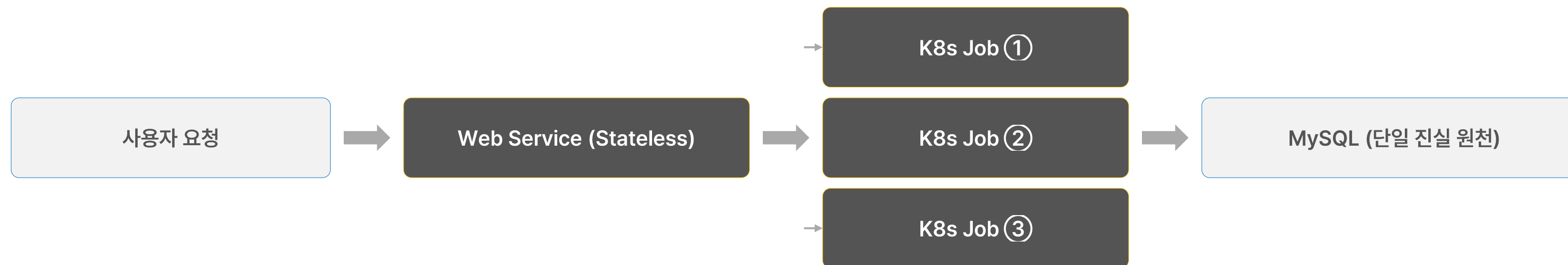
- 로컬 스크립트 실행, 격리 없음
- 실행마다 환경 불일치
- 결과 저장 및 감사 이력 부재
- 수동 실행, 확장성 부족

After

- 컨테이너 기반 격리 실행
- 1 Backtest = 1 Kubernetes Job
- MySQL 기반 결과 저장 및 run 추적
- GitOps 기반 선언형 배포
- 관측 가능한 플랫폼 (Prometheus + Grafana)

02 Core Concept

1 Backtest = 1 Kubernetes Job



워크로드 격리



수평 확장



K8s 수명주기 관리



리소스 효율성

03

Design Constraints

1 

엔진 불변성

- 레거시 엔진 = 불변 객체
- 코어 엔진 절대 수정 불가
- 모든 확장은 Adapter 계층으로
- 제약 → 설계 원칙으로 전환

2 

데이터셋 전략

- 데이터를 Docker 이미지에 내장
- OHLCV CSV → 이미지에 COPY
- 외부 API 의존성 제거
- 매 실행마다 동일한 데이터 보장

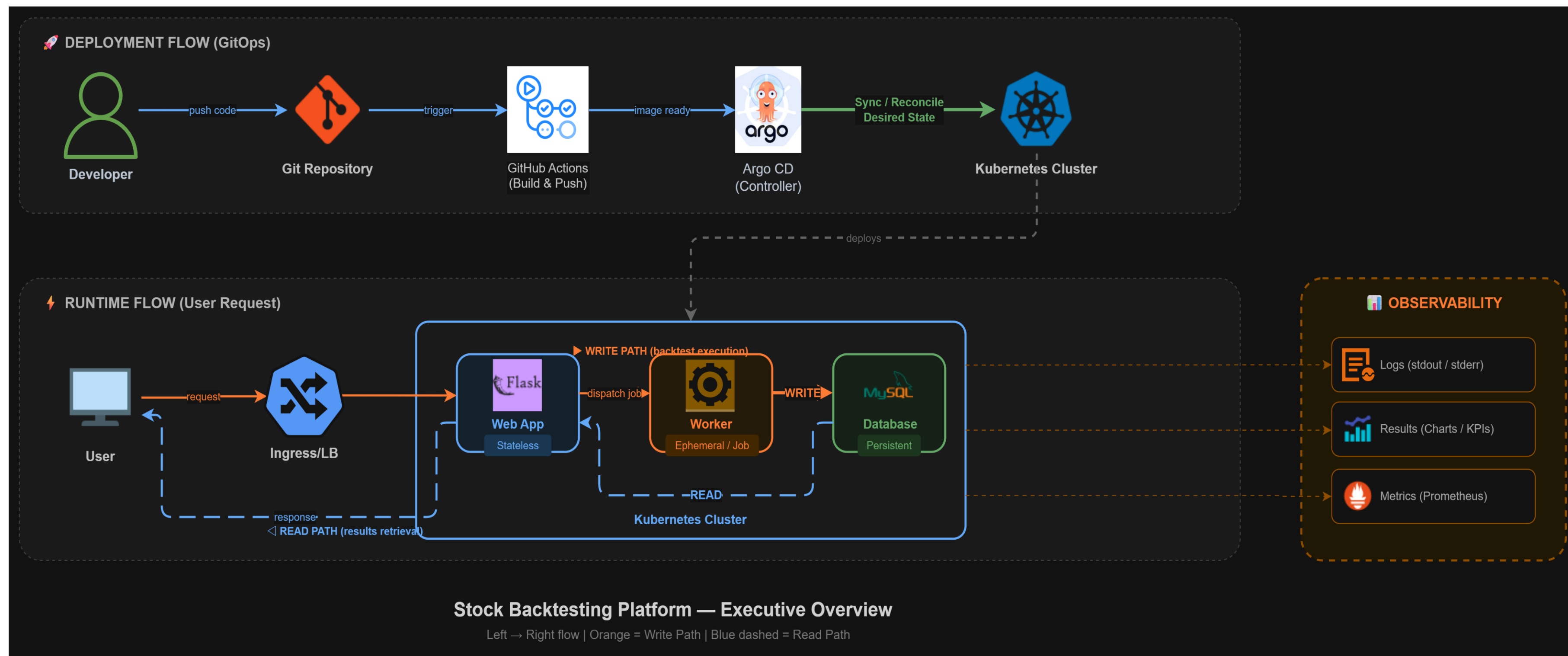
3 

재현성

- 동일 입력 = 동일 출력
- data_hash (SHA-256) 검증
- image_tag (Git SHA) 추적
- rule_type + params 기록

04

System Architecture



Web: 무상태

Worker: 일회성

MySQL: 영구 저장

Why Kubernetes Job?

왜 Celery가 아닌가?

기준	Celery	Kubernetes Job
격리 수준	프로세스 수준	Pod (컨테이너)
인프라 의존성	Redis / RabbitMQ	Kubernetes 자체
수명주기 관리	직접 구현 필요	backoffLimit, TTL
확장성	Worker 추가	인프라 수준 스케일링
정리 작업	수동 관리	TTL 자동 정리

"Kubernetes가 스케줄러이자 큐이다."
각 백테스트는 K8s Job이 관리하는 격리된 Pod에서 실행됩니다.

06

State Machine & Lifecycle



단방향 전이만 허용. 롤백 없음.

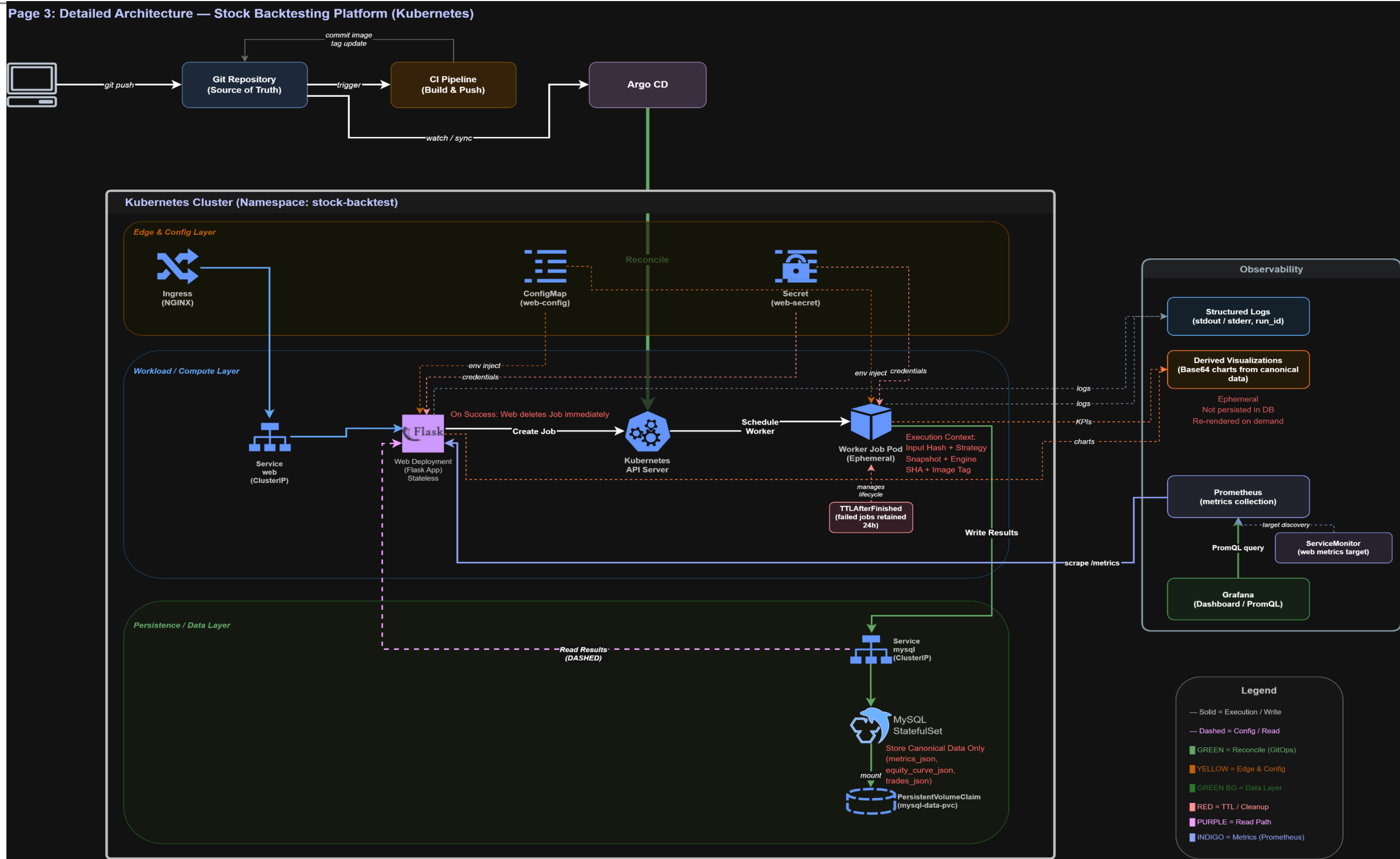
Job Lifecycle

- 성공 → Web이 Job 즉시 삭제
- 실패 → 24시간 유지 (TTL: 86400s)

JobLauncher 추상화

- ENV: JOB_LAUNCHER_MODE
- Local: subprocess (개발용)
- K8s: BatchV1Api (운영용)

07

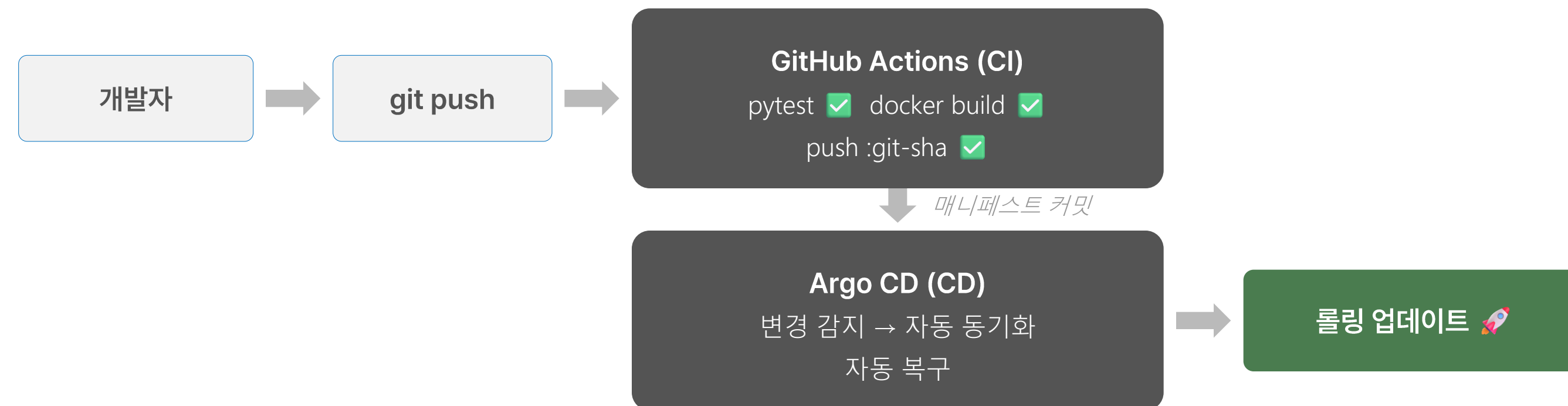


동일 흐름: CI/CD → Cluster → Web → Job Pod → MySQL → Observability
추가 세부사항: ConfigMap, Secret, RBAC, Job 수명주기

08

CI/CD & GitOps Pipeline

"배포 = git push. 롤백 = git revert."

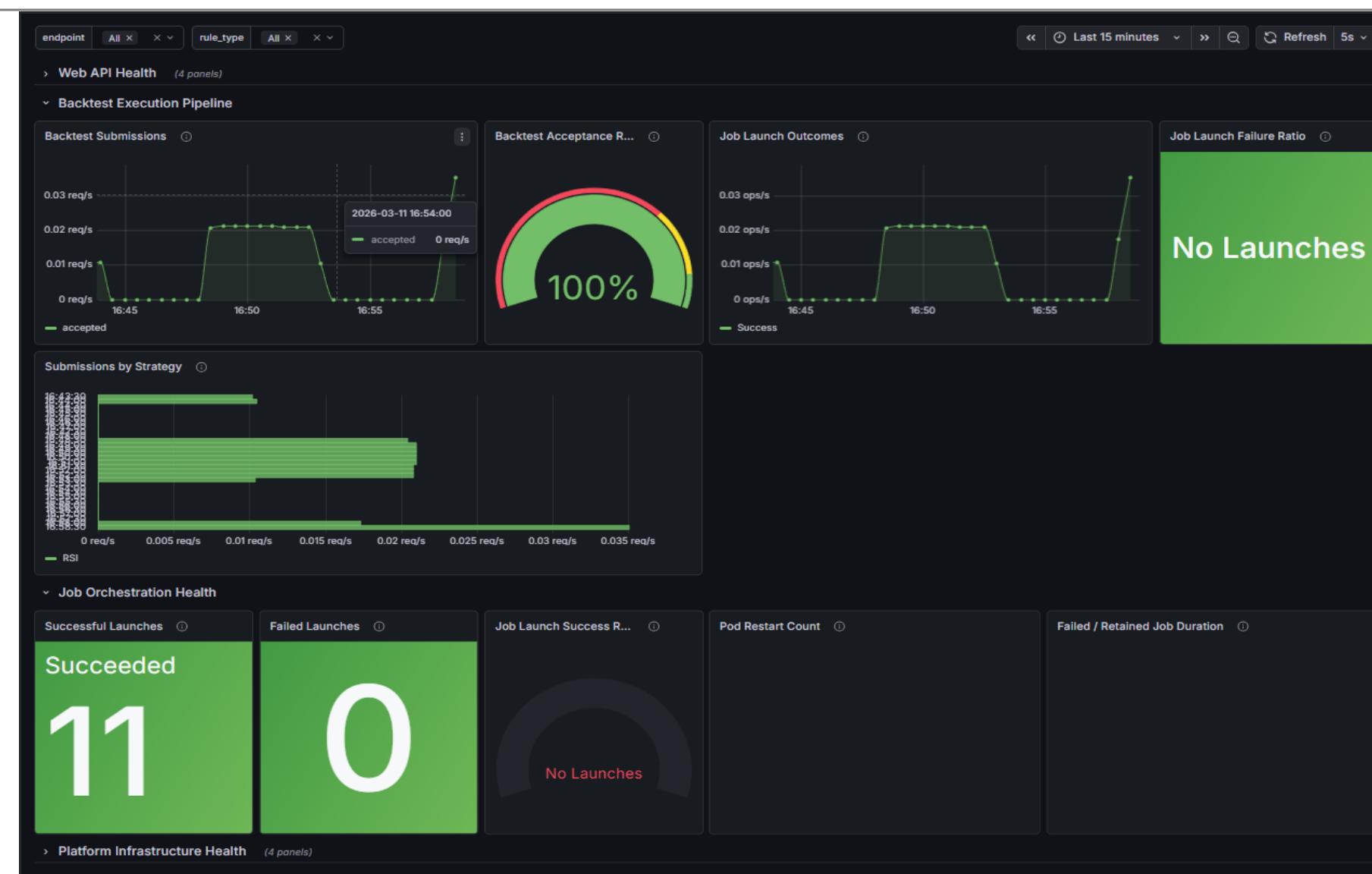


불변 태그 (:git-sha)
= 재현성 보장

CI가 매니페스트 커밋
→ CD가 동기화

자동 복구 활성화

09 Observability



로그 = 요청별 추적. 메트릭 = 집계 모니터링.

구조화 로그 (요청별 추적) 📋

- Web → Worker → MySQL 전 구간
- 모든 로그에 [run_id=UUID] 태깅
- kubectl logs | grep run_id=xxx

Prometheus 메트릭 (집계 모니터링) 📊

- /metrics → ServiceMonitor → Grafana
- ⚠️ run_id는 절대 메트릭 라벨로 사용 금지

10

Engineering Results

83

테스트 통과 (pytest)



Job 오케스트레이션 검증



GitOps 파이프라인 운영 중



E2E 추적 확인 완료

Live Demo

End-to-End Platform Walkthrough

① 요청 제출



② K8s Job 생성



③ Grafana 대시보드 결과 확인



④ 로그 추적



⑤ 결과 확인

BACKTEST TERMINAL

CONNECTED 10:09:15

COMMAND INPUT

CONFIG

SECURITY

TICKER AAPL

STRATEGY

RULE RSI (Relative Strength :)

PARAMETERS

RSI PERIOD 14

OVERSOLD 30

OVERBOUGHT 70

DATE RANGE

FROM 2020-01-01

TO 2024-12-31

TRADING

CAPITAL 100000

FEE RATE (DECIMAL) 0.001

e.g. 0.001 = 0.1%

EXECUTE BACKTEST

PRESETS

NO PRESET

SAVE

DELETE

SYS K8s Backtest Platform v1.0

ENV Development

ENG BacktestEngine (immutable)

AWAITING BACKTEST EXECUTION

Configure parameters and press EXECUTE to begin.

Select a s

adjust

www.BANDICAM.com

stock_backtest [WSL: Ubuntu-22.04] [관리자]

AAPL.csv

data > AAPL.csv > data

1 Date,open,high,low,close,adj_close,volume

256 2001-01-04,0.3239400088787079,0.33035698533058167,0.3002229928970337,0.3046880066394806,0.25610941648483276,739396000

257 2001-01-05,0.30245500802993774,0.31026801466941833,0.28683000802993774,0.2924109995365143,0.24578988552093506,412356000

258 2001-01-08,0.30245500802993774,0.30329200625419617,0.2845979928970337,0.295758992433548,0.2486042082309723,373699200

259 2001-01-09,0.3002229928970337,0.31501099467277527,0.295758992433548,0.30691999197006226,0.2579856216907501,588929600

260 2001-01-10,0.297991007566452,0.3035709857940674,0.28683000802993774,0.295758992433548,0.2486042082309723,580781600

문제 출력 디버그 콘솔 터미널 포트 17

(stock_k8s) jongwon@DESKTOP-72EMS11:~/personal_project/cloud_project/stock_b

acktest\$

(stock_k8s) jongwon@DESKTOP-72EMS11:~/personal_project/cloud_project/stock_b

acktest\$

BANDICAM ADMIN UNREGISTERED

1920x1080 - (0, 0), (1920, 1080) - 디스플레이 1

홈

시작하기

비디오

이미지

오디오

일반

비디오

이미지

정보

화면 녹화 모드 - 전체 화면

모니터 화면 전체를 녹화할 때 사용하는 방식입니다.

1. 녹화대상 모니터(디스플레이)를 선택합니다.

2. 녹화시작 단축키나 'REC' 버튼을 눌러 녹화를 시작합니다.

REC 녹화 시작 하기

온라인 도움말 보기

녹화 시작/정지

F12

이미지 캡처

F11

BANDICUT >>>

반디캠+반디컷 패키지로 구매하시면 더 저렴합니다.

WSL: Ubuntu-22.04

dev

01 115t

0 0 0

17

CSVLint

Query

Align

Col 4: low

JJong-03 (1개월 전)

줄 273, 열 61

공백: 4

UTF-8

LF

{ } CSV

오전 10:09

2026-03-16

Live Demo

CI/CD test

① Git push



② main에 merge



③ PR 생성



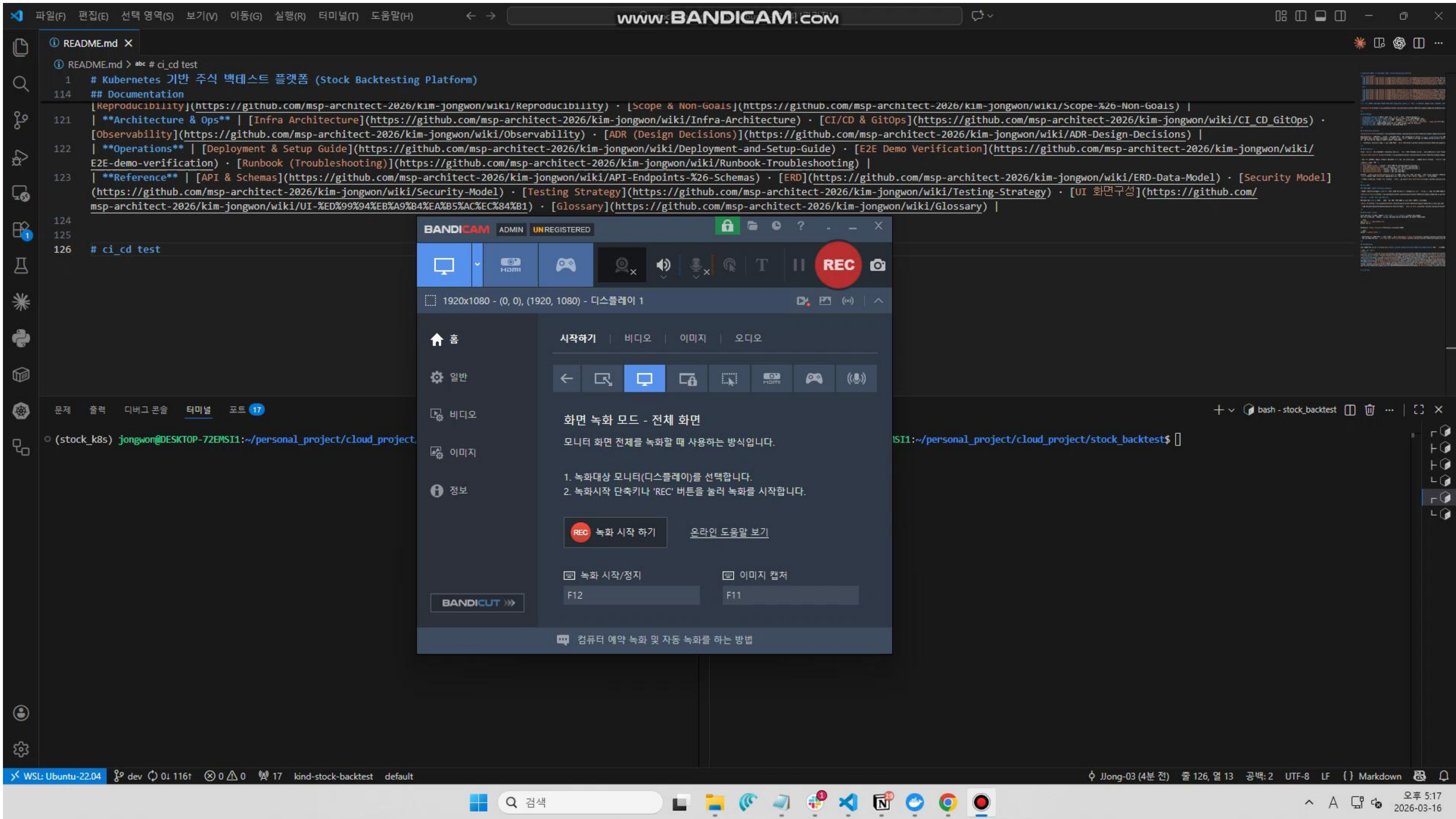
④ PR 승인 후 merge



⑤ New git-sha 생성



⑥ Manifest 업데이트



13

Lessons Learned

01 

엔진 불변성이 설계를 단순화

제약이 곧 설계 원칙이 되었다.
엔진 수정 대신 Adapter로 확장

02 

쓰기/읽기 경로 분리

Worker는 쓰고, Web은 읽는다.
MySQL만이 두 경로의 교차점

03 

K8s Job 오케스트레이션의 복잡성

상태 머신, RBAC, TTL, 정리 작업 필요
생각보다 "그냥 Job 생성"이 아니었다.

04 

GitOps = 운영 안정성

배포는 commit, 롤백은 revert
수동 kubectl apply를 제거했다.

14

Future Work

1

멀티노드 클러스터 + HPA

Control Plane 1 + Worker 2 구성
Worker 노드에 HPA 적용, 부하 기반 자동 확장

2

Helm / Kustomize

환경별 매니페스트 관리 (dev/stg/prod)
운영 환경 차이를 더 일관되게 관리

3

MetalLB

Bare-metal LoadBalancer로 사내 접근 가능
Kind 없이 팀 전체가 서비스 이용 가능

Thank You

Q&A

GitHub: <https://github.com/msp-architect-2026/kim-jongwon>

