

[쿠버네티스] 프로젝트 1

☰ 카테고리	Docker
☰ 태그	
<input checked="" type="checkbox"/> 공개	<input type="checkbox"/>
■ 난이도	

Upbit Clone 프로젝트 최종 설계안 (Re-Revised)

1. 최종 기술 스택 및 데이터 정의

분류	기술 스택	세부 역할 및 저장 데이터
Infra	Ubuntu 24.04 / Tailscale	하이브리드 노드 통합, Tailscale Subnet Router(Bastion) 보안 구성
Platform	Kubernetes (v1.31+)	컨테이너 오케스트레이션 및 Patroni(DB HA) 운영
Backend	Node.js (Express)	업비트 Open API 수집, 비즈니스 로직, DB/Redis 커넥션 관리
Frontend	React	Tailwind CSS 기반의 실시간 시세 대시보드 및 차트 시각화
DB 1 (Cache)	Redis	휘발성: 실시간 호가, 현재가, API 요청 제한(Rate Limit) 관리
DB 2 (Main)	TimescaleDB	시계열: 분봉/일봉 캔들, 체결 내역 (Hypertable) 관계형: 유저 정보, 자산 현황, 마켓 코드
Ops	ArgoCD / Prometheus	(일정 후반) GitOps 배포 파이프라인 및 클러스터 리소스 모니터링

2. 장비별 리소스 및 서비스 배치 (Updated)

- *노트북(Control & View)**과 **데스크탑(Core Data & Logic)**으로 역할을 완전히 분리했습니다.

 노트북: Control & Frontend Zone (32GB)

사용자와 직접 맞닿는(UI) 역할과 클러스터 제어를 담당합니다.

노드 이름 (Hostname)	역할 (Role)	주요 컴포넌트 & 스펙 (할당량)
master	Control Plane	<ul style="list-style-type: none">K8s Control Plane (API Server, Scheduler)(4GB)
worker-2	Frontend	<ul style="list-style-type: none">React App (Web Server)Nginx Ingress Controller (외부 트래픽 유입)(10GB)
db-slave	DB Read Replica	<ul style="list-style-type: none">TimescaleDB Slave (Streaming Replication)읽기 전용 쿼리 처리 (차트 조회 등)(8GB)

💻 데스크탑: Backend & Data Core Zone (32GB)

무거운 연산(수집)과 데이터 저장(DB, Storage)을 전담합니다.

노드 이름 (Hostname)	역할 (Role)	주요 컴포넌트 & 스펙 (할당량)
worker-1	Backend	<ul style="list-style-type: none">Node.js App (Ingester/API)업비트 소켓 연결 유지 및 데이터 가공(8GB)
db-master	DB Master	<ul style="list-style-type: none">TimescaleDB Master (모든 쓰기/수정)Redis (인메모리 캐시)(12GB)
storage-nfs	Storage	<ul style="list-style-type: none">NFS Server (PV/PVC 영구 데이터 저장소)DB 백업 파일 관리(4GB)
monitor-1 (New)	Monitoring	<ul style="list-style-type: none">Prometheus (메트릭 수집)Grafana (대시보드)(4GB)

3. 데이터 흐름 (Data Flow)

변경된 노드 이름을 반영한 데이터 이동 경로입니다.

- 수집: worker-1 (Node.js)이 업비트에서 데이터를 받음.
- 저장/캐싱: worker-1 이 db-master (Redis/TimescaleDB)에 데이터를 씀(Write).
- 복제: db-master 의 데이터가 VPN을 타고 db-slave (노트북)로 실시간 복제됨.
- 조회: 사용자가 접속하면 worker-2 (React)가 db-slave (읽기) 또는 db-master (Redis)에서 데이터를 가져와 보여줌.

Flow: Upbit API → worker-1 → db-master (Write) ➔(Replication) ➔ db-slave
→ worker-2 (Read)

4. 상세 일정 (Schedule)

CI/CD 구축을 마지막(3-2단계)으로 미루어 현실적인 개발 시간을 확보했습니다.

- 1단계 (12/19 ~ 12/26): [인프라 & DB 기초]
 - Ubuntu/Tailscale/K8s 설치 완료.
 - TimescaleDB Master (`db-master`) 구축 및 테이블 생성 (현재 진행 중).
 - 노드 라벨링 재정리 (위의 이름대로).
 - 2단계 (12/27 ~ 01/02): [데이터 파이프라인 & 백엔드]
 - `db-slave` 복제 설정 (네트워크 이슈 해결 후).
 - Node.js 수집기 (`worker-1`) 개발 및 배포.
 - Redis 연동 및 실시간 시세 데이터 적재 테스트.
 - 3단계 (01/03 ~ 01/09): [프론트엔드 & 마무리]
 - 3-1 (필수): React (`worker-2`) 호가창/차트 UI 개발 및 API 연동.
 - 3-2 (선택/후순위): ArgoCD (CI/CD) 파이프라인 구축 및 Prometheus (`monitor-1`) 연동.
 - 최종 발표 자료 정리 및 시연 시나리오 점검.
-