QualiJournal – Cloud Run 운영 핸드오프

1) 현재 상태 요약

서비스: QualiJournal Admin (FastAPI + Uvicorn, Docker, Google Cloud Run / asia-northeast3)

서비스 URL: https://quali-journal-665693030029.asia-northeast3.run.app

핵심 성공 지표

/health가 {"status": true}로 200 OK

/api/tasks/recent?limit=5 200 OK + items가 비어있거나/존재

Admin UI에서 "커뮤니티 실행/키워드 실행" 시 하단 로그가 스트리밍됨

최근 수정의 핵심

/api/tasks/recent를 **동적 경로**(/api/tasks/{job_id}) **보다 먼저** 선언(경로 충돌 방지)

/health 엔드포인트 추가 (Cloud Run 스타트업 프로브/헬스체크용)

작업 실행 시 **sys.executable/python3** 사용(컨테이너에서 py 명령 없음)

Cloud Run에서 **컨테이너 명령어/인수 비우기** → 이미지의 start.sh 실행 보장

---

2) 코드/파일 구성(핵심만)

server_quali.py

app = FastAPI(...)

@app.get("/health") → {"status": true}

@app.get("/api/tasks/recent") → 최근 작업 로그 목록

@app.get("/api/tasks/{job_id}") → 작업 상세

@app.post("/api/tasks/flow") → orchestrator 실행(community/daily/keyword)

orchestrator.py

실제 수집/처리 작업. 로그는 logs/tasks/&job-id&.log에 적재.

start.sh (컨테이너 엔트리포인트)

exec python -m uvicorn server_quali:app --host 0.0.0.0 --port "$PORT" --workers 1

권장: 자동 감지 버전(아래 7절) 사용하면 admin/server_quali.py 위치도 자동 지원

Dockerfile

pip install -r requirements.txt → CMD ["sh","-c","exec /app/start.sh"]

requirements.txt

fastapi, uvicorn, 기타 실행에 필요한 패키지

---

3) 배포 파이프라인 요약

git push (main) → Cloud Build가 Docker 이미지 빌드/푸시
Cloud Run이 새 리비전 배포
트래픽 100%가 최신 리비전에 할당됐는지 확인(버전 탭)

---

4) 로컬 실행(빠른 점검)

```
# 프로젝트 루트에서
python -m uvicorn server_quali:app --reload --port 8080

# 점검
curl http://localhost:8080/health
curl "http://localhost:8080/api/tasks/recent?limit=5"
```

---

5) 운영 점검 체크리스트(초간단)

브라우저: …/health → {"status": true}
브라우저: …/api/tasks/recent?limit=5 → 200 OK
Admin UI: “커뮤니티 실행” 클릭 → 하단 스트림 로그가 흐르는지
Cloud Run **버전** 탭: 최신 리비전에 **트래픽 100%**
Cloud Run **컨테이너 명령어/인수**: 비어있음(이미지의 start.sh 사용)
포트: 8080

---

6) 장애/이슈 → 처방 매뉴얼(필살기 3개)

경로 충돌
증상: /api/tasks/recent가 404 또는 "job not found"
원인: /api/tasks/{job_id}가 **먼저** 선언됨
처방: server_quali.py에서 **고정 경로**/api/tasks/recent를 **위**, 동적 경로를 **아래**
컨테이너 시작 실패(8080 리슨 실패/STARTUP TCP probe failed)
원인:
Cloud Run “명령어/인수”에 uvicorn …를 넣어 start.sh가 **무시**됨
모듈 경로 불일치(server_quali:app vs admin.server_quali:app)
처방:
“명령어/인수” **비우기**
start.sh**자동 감지 버전** 사용(7절)
작업 실행 즉시 &END& error
원인: 컨테이너에는 py 명령이 없음
처방: 작업 실행은 **sys.executable/python3** 사용(이미 코드 반영됨)

7) start.sh 권장(자동 감지) 스니펫

admin/server_quali.py 또는 루트 server_quali.py 어느 쪽이든 안전하게 기동

```
#!/bin/sh
set -eu
PORT="${PORT:-8080}"

TARGET="server_quali:app"
if [ -f "admin/server_quali.py" ]; then
  TARGET="admin.server_quali:app"
elif [ ! -f "server_quali.py" ] && [ -f "/app/admin/server_quali.py" ]; then
  TARGET="admin.server_quali:app"
fi

exec python -m uvicorn "$TARGET" --host 0.0.0.0 --port "$PORT" --workers 1
```

8) 로그 빠르게 보는 법(Cloud Run → 관측 가능성 → 로그)

필터 예시(검색창):
Traceback / Exception / startup
Default STARTUP TCP probe failed
Container called exit(1)
GET /api/tasks/recent
“최근 N분” 범위 조정하며 확인

9) 환경 변수/비밀

PORT (Cloud Run 제공)
APP_ENV=prod, QUALI_DB_MODE=prod (현재 사용)
OPENAI_API_KEY (비밀) — 서버가 직접 외부 호출 필요 시 사용

10) 남은 개선 후보(백로그)

start.sh**자동 감지** 정식 채택(운영 표준화)

로그 삭제/순환 정책(Cloud Run/스토리지 비용 최소화)
/api/tasks/recent 페이지네이션/정렬 옵션
작업 실패 시 알림(Slack/Webhook)
GitHub Actions 등 CI/CD 파이프라인 메트릭/게이트 강화

👉 새 채팅용 “컨텍스트 블록”(이걸 통째로 붙여넣고 시작하세요)

[컨텍스트/핸드오프]

프로젝트: QualiJournal Admin (FastAPI + Uvicorn, Docker, Cloud Run: asia-northeast3)
서비스 URL: https://quali-journal-665693030029.asia-northeast3.run.app

핵심 파일:
- server_quali.py: FastAPI app. /health, /api/tasks/recent(고정) → /api/tasks/{job_id}(동적) 순서 중요.
- orchestrator.py: 작업 실행 및 logs/tasks/&job-id&.log 생성.
- start.sh: python -m uvicorn &모듈경로&:app --port $PORT. (권장: 자동 감지 버전)
- Dockerfile: pip install -r requirements.txt → CMD exec /app/start.sh
- requirements.txt: fastapi, uvicorn 등

운영 체크:
1) /health → {"status": true}
2) /api/tasks/recent?limit=5 → 200 OK
3) Admin UI에서 커뮤니티/키워드 실행 → 하단 스트림 로그
4) Cloud Run: 최신 리비전에 트래픽 100%
5) Cloud Run 컨테이너 명령어/인수 비움, 포트 8080

장애 패턴 & 처방:
- recent 404: 경로 순서 역전 → 고정 경로를 동적 경로보다 위로
- 8080 리슨 실패: 명령어/인수 오버라이드→ 비우기 / start.sh 사용, 모듈경로 점검
- &END& error: 'py' 미존재 → sys.executable/python3 사용

요청:
- 이전 맥락 그대로 이어서 작업 시작.
- 먼저 현재 리비전/트래픽, /health, /api/tasks/recent 상태 확인.
- start.sh 자동 감지 적용 여부 점검 후 필요 시 반영.

[/컨텍스트/핸드오프]