

시스템 아키텍처 (Self-Hosted + Caddy)

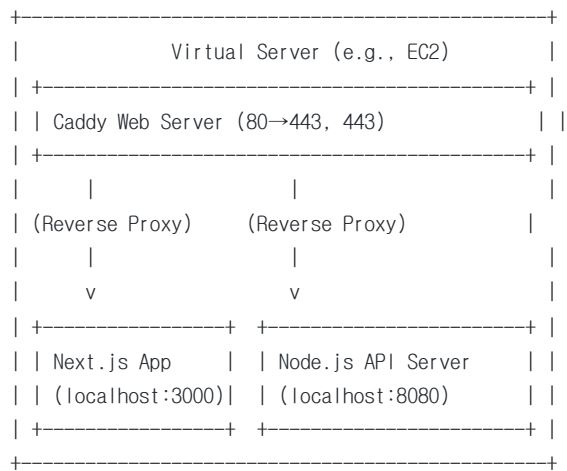
구성 개요

- Caddy: HTTPS 종단, 리버스 프록시, 로그, 압축
- Next.js(프론트): UI, 카메라/업로드/프롬프트/상태 관리
- Node.js(Express, 백엔드): Gemini 연동, 이미지 변환 오케스트레이션, 정책/보안

트래픽 흐름

사용자 브라우저 ⇄ 인터넷(HTTPS) ⇄ Caddy

- `/api/*` → `localhost:8080` (Express)
- 그 외 → `localhost:3000` (Next.js)



Caddyfile 예시

```
my-4cut-app.com {
  handle_path /api/* {
    reverse_proxy localhost:8080
  }

  handle {
    reverse_proxy localhost:3000
  }

  log {
    output file /var/log/caddy/access.log
  }

  encode gzip zstd
}
```

상태/데이터 보존 전략

- 변환 결과(4컷), 프롬프트, 선택 프리셋을 `localStorage` 에 보관해 네비게이션/새로고침에도 유지
- 서버 저장소 사용 시 개인정보/저작권 고려 필요(기본은 비저장)

보안

- HTTPS 기본화(Caddy 자동 인증서)
- 백엔드에서 입력 검증, Rate limiting, CORS 제한(프론트 도메인 화이트리스트)
- 비밀키는 `.env` 또는 시크릿 매니저 사용