시스템 아키텍쳐 설계서

1. 시스템 개요

1.1 주요 컴포넌트

- **Frontend**
  - 공통 컴포넌트
    - header

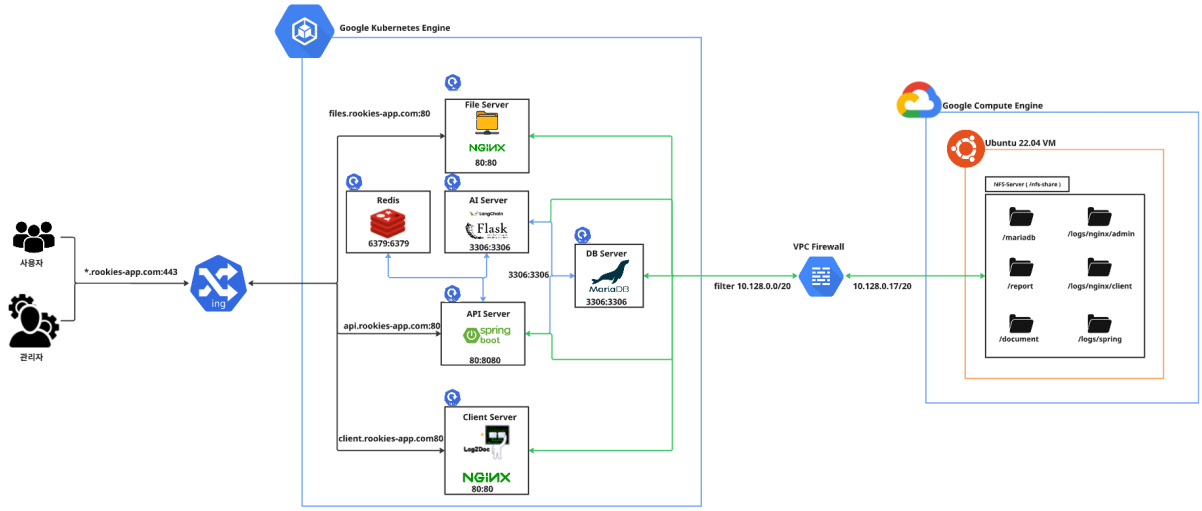
  - 일반 사용자 컴포넌트
    - DocumentCard: 단일 문서를 요약 카드 형태로 표시
    - DocumentGrid: 문서들을 그리드 형태로 배치
    - UploadModal: 문서 업로드를 위한 모달 창
    - Sidebar: 문서 탐색용 사이드바 메뉴
    - FilterControls: 필터 및 정렬 컨트롤 UI
    - Pagination: 페이지 이동 컨트롤러

  - 관리자 컴포넌트
    - MemberListToolbar: 회원 검색, 필터, 액션 버튼 등 툴바
    - MemberViewForm: 개별 회원 정보 상세 뷰/수정 폼
    - ErrorReportTable: 에러 리포트 테이블
    - AttackErrorReportTable: 공격 에러 리포트 테이블
    - AttackErrorReportChart: 공격 에러 리포트 시각화 차트
    - CategoryPieChart: 에러 유형별 카테고리 파이차트
    - ReportTrendChart: 에러 발생 추세 차트
    - StatusBarChart: 상태별(미처리/진행 중/완료 등) 에러 바차트
    - WeeklyReportChart: 주간 에러 리포트 차트
    - Sidebar: 관리자 페이지용 사이드바 메뉴
    - FilterControls: 필터 및 정렬 컨트롤 UI
    - Pagination: 페이지 이동 컨트롤러
    - MemberListTable: 회원 목록 테이블 컴포넌트

- **Backend**
  - Flask web Application
    REST API 서버 및 HTTP 요청 처리
    - SecurityAnalysisSystem
      - 핵심 보안 분석 엔진
      - 로그 데이터 패턴 분석
      - 공격 유형 탐지, 위험도 평가 및 분류

        - Analysis Workflow Engine
            - 분석 워크플로우 관리
            - `base_analysis_node`: 기본 패턴 분석
            - `llm_analysis_node`: AI 기반 고급 분석
            - `report_generation_node`: 보고서 생성
            - `db_save_node`: 데이터베이스 저장
        - LLM Integration Module
            - AI 기반 보안 분석
            - OpenAI GPT-3.5-turbo 모델 연동
            - LangChain을 통한 프롬프트 관리
            - 컨텍스트 기반 분석 결과 생성
    - Spring Boot
        - controller/ (컨트롤러)
            - AuthController - JWT 기반 인증/인가 (로그인, 토큰 갱신, 로그아웃)
            - DocumentController - 문서 업로드/조회/다운로드 및 권한 관리
            - ErrorReportController - AI 생성 에러 리포트 조회/관리/통계
            - CeoUserController - CEO 전용 사용자 관리 API
        - service/ (서비스)
            - DocumentService - 파일 저장/권한 검증/DTO 변환
            - ErrorReportService - AI 리포트 데이터 조회/상태 관리
            - FlaskReportService - Flask AI 서버 연동/로그 분석 요청
            - CeoUserService - CEO 권한 검증/사용자 관리 (Redis 캐싱)
            - RefreshTokenService - JWT 토큰 생성/검증/관리
- AI
    - LangChain
- **Database**
    - MariaDB
        - 테이블
            - users - 사용자 정보 및 계정 관리
            - roles - 계층적 직급 시스템 (인턴~CEO, 레벨 1-11)
            - documents - 문서 메타데이터 및 파일 정보
            - error_report - AI 분석 에러 리포트 저장소
    - Redis
        - Authentication & Session (인증 및 세션)
- **Web Server**
    - nginx
        -
- ETC
    - nfs server
        - /app/document (문서 파일 저장 디렉토리)
        - file.upload.path.nfs=/app/document (application.properties 설정)
        - FileStorageConfig (NFS 전용 파일 저장 설정 클래스)

2. 시스템 구성도



3. 기술 스택

Backend

- Framwork
  - Springboot 3.5.3
  - Spring security
  - Spring JPA
  - Spring Web (Rest API)
  - Spring Data Redis
  - Flask 2.x
- 언어: Java, Python 3.1x
- AI Framwork
  - LangChain
  - LangGraph
  - OPENAI API
- DB, 저장소
  - MariaDB 11.4
  - Redis Latest
  - NFS (Network File System)
- 통신: RestClient
- 보안: JWT Token 인증 기반
- 빌드: Maven

Frontend

- Framework: React 19
- 언어: JavaScript
- 상태 관리: Zustand
- 디자인: Figma
- 스타일링: CSS, Lucide React, react-datepicker
- 데이터 시각화: Chart.js + react-chartjs-2, Recharts
- 라우팅 라이브러리: React Router v7
- Lint 도구: ESLint
- 빌드: Vite

DevOps

- 배포: kubernetes 1.32, gcp vm
- kubernetes cluster:
    - node : 4
    - cpu : 8
    - memory : 16gb
    - control plain location : us-central1-a
- 이미지:
    - file server
    - mariadb server
    - ai server
    - api server
    - redis server
    - client server
- HTTPS
    - IP : VPC Network
    - Domain : Square Space
    - SSL : GTS CA WR3
- Storage
    - GCP VM Ubuntu Server 22.04 NFS Server
- Health Check
    - kubernetes backend config
    - nginx /health endpoint

4. 컴포넌트 구조

Backend

- /config
    - WebSecurityConfig (Spring Security 설정/CORS)
    - RedisConfig (Redis 연결/직렬화 설정)
    - SwaggerConfig (API 문서화 설정)
    - FileStorageConfig (NFS 파일 저장소 설정)
    - AsyncConfig (비동기 작업 스레드 풀)
    - DataInitializer (기본 데이터 초기화)
- /controller
    - AuthController (JWT 인증/로그인/로그아웃)
    - DocumentController (문서 업로드/조회/다운로드)
    - ErrorReportController (AI 리포트 조회/통계)
    - CeoUserController (CEO 전용 사용자 관리)
- /dto
    - /request
        - LoginRequest (로그인 요청)
        - TokenRefreshRequest (토큰 갱신 요청)
        - DocumentCreateRequest (문서 생성 요청)
    - /response
        - JwtResponse (JWT 토큰 응답)
        - ApiResponse (표준 API 응답 래퍼)
        - DocumentResponseDTO (문서 상세 응답)
        - ErrorReportDTO (에러 리포트 응답)
        - UserDetailResponse (사용자 상세 정보)
- /entity
    - User (사용자 정보/계층적 직급)
    - Role (직급 시스템 인턴~CEO)
    - Document (문서 메타데이터/파일 경로)
    - ErrorReport (AI 분석 리포트)
    - RefreshToken (JWT 토큰 정보)
    - CategoryType (문서 분류)
- /exception
    - GlobalExceptionHandler (통합 예외 처리)
    - PermissionDeniedException (권한 부족 예외)
    - UnauthenticatedException (인증 실패 예외)
    - TokenRefreshException (토큰 갱신 실패)
    - CustomException (애플리케이션 전용 예외)
- /log
    - LoggingInterceptor (요청/응답 통합 로깅)
    - LogBuilder (공통 로그 데이터 구성)
    - LogSender (Flask AI 서버 로그 전송)

- /repository
    - UserRepository (사용자 CRUD/권한 조회)
    - DocumentRepository (문서 메타데이터/권한 필터링)
    - ErrorReportRepository (AI 리포트 통계/JPQL 쿼리)
    - RefreshTokenRepository (JWT 토큰 관리)
    - RoleRepository (직급 정보 조회)
    - CategoryTypeRepository (문서 카테고리 관리)
- /scheduler
    - TokenCleanupScheduler (만료 토큰 정리 스케줄러)
- /security
    - /jwt
        - JwtUtils (JWT 토큰 생성/검증)
        - AuthTokenFilter (JWT 토큰 필터)
        - AuthEntryPointJwt (인증 실패 핸들러)
    - /services
        - UserDetailsServiceImpl (Spring Security 사용자 서비스)
        - UserDetailsImpl (사용자 인증 정보 구현체)
- /service
    - DocumentService (문서 관리/파일 저장/권한 검증)
    - ErrorReportService (AI 리포트 데이터 조회/상태 관리)
    - CeoUserService (CEO 권한 검증/Redis 캐싱)
    - RefreshTokenService (JWT 토큰 생성/검증/관리)
    - FlaskReportService (Flask AI 서버 연동)

Frontend

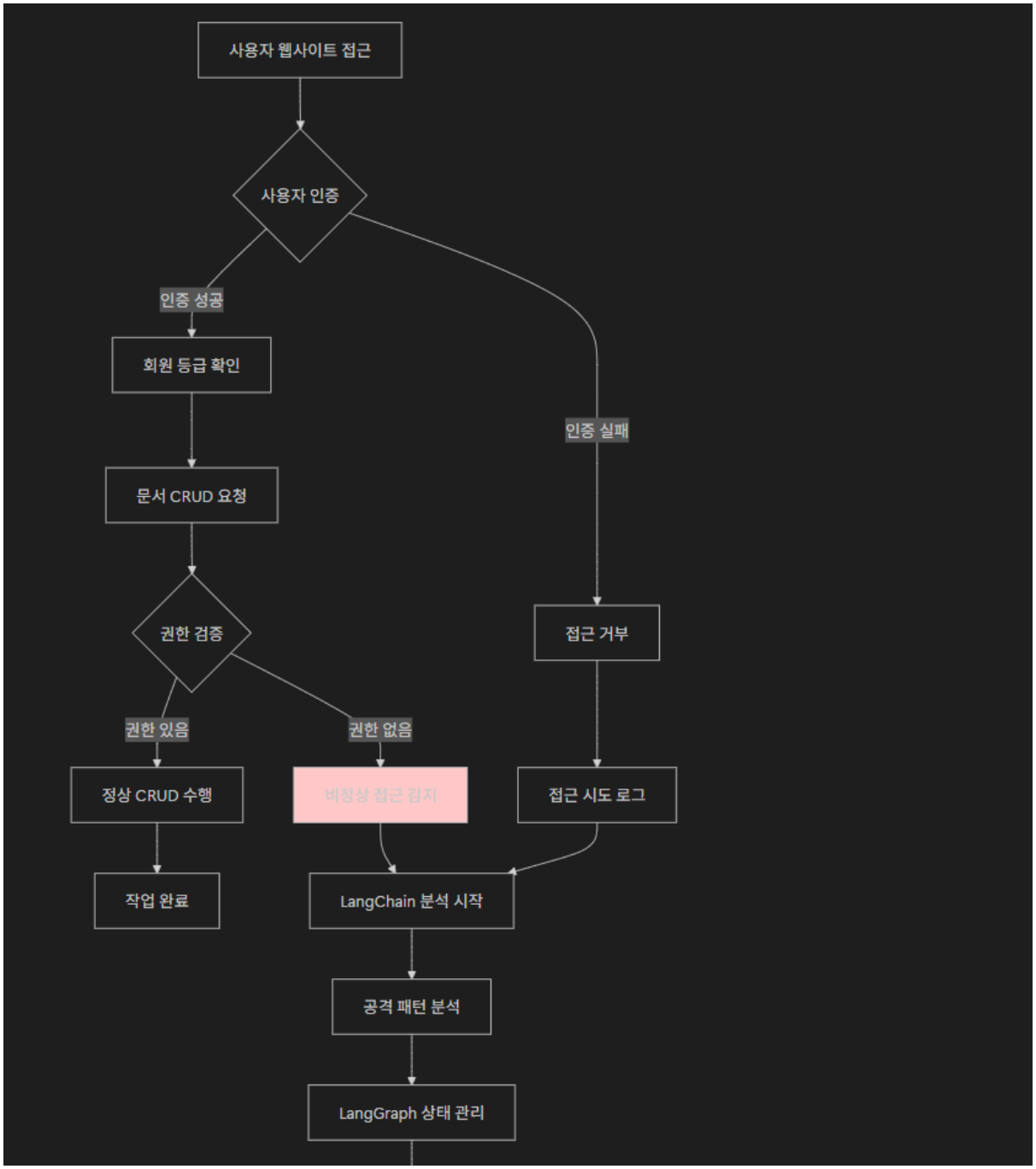
- /admin (관리자 전용 모듈)
    - /api (관리자 API 호출 모듈)
    - /components
        - /layout (레이아웃 컴포넌트)
        - /member (회원 관리 페이지 컴포넌트)
        - /report (에러 리포트 관리 페이지 컴포넌트)
        - /ui (공통 UI)
    - /pages (메인 대시보드, 회원 관리 페이지, 에러 리포트 페이지, 공격 에러 리포트 페이지)
    - /stores (회원 정보, 에러 리포트 정보 상태 관리 저장소)
    - /styles (css)
- /api (일반 사용자 API 호출 모듈)
- /assets (로고 이미지)
- /components (로그인 페이지 및 문서 페이지 전용 컴포넌트)
- /constants (이미지, 카테고리, 직급 매핑)
- /pages (로그인 페이지, 문서 페이지)
- /stores (로그인 정보 상태 관리 저장소)

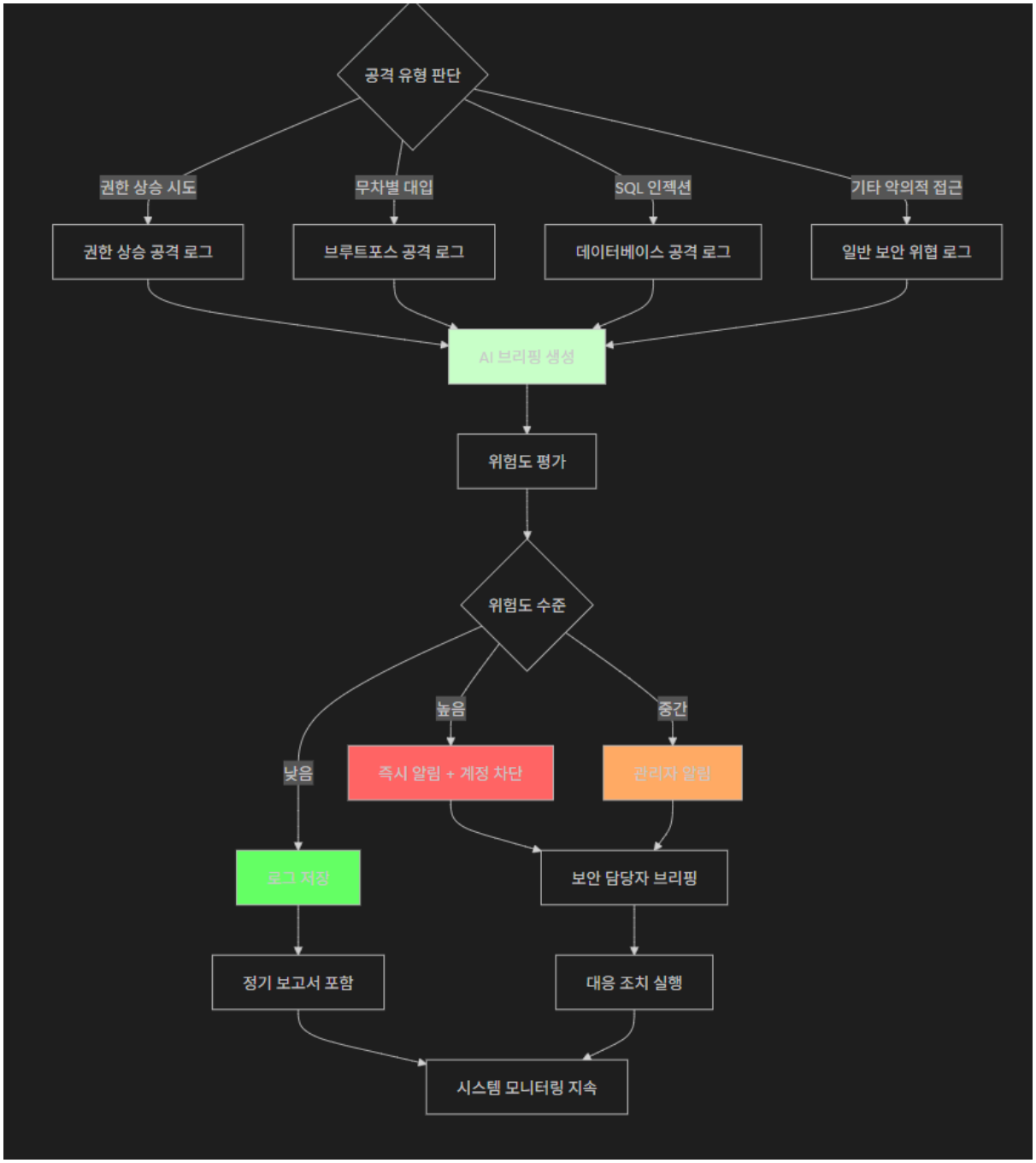
DevOps

- /ai_server (Dockerfile, ai-server-deployment.yaml)

- /api_server (Dockerfile, api-server-deployment.yaml)
- /client_server (Dockerfile, nginx.conf client-server-deployment.yaml)
- /file_server(Dockerfile, nginx.conf, file-server-deployment.yaml)
- /mariadb_server (mariadb-deployment.yaml)
- /redis_server(redis-deployment.yaml)
- gke-backend-config.yaml
- gke-managed-certificate.yaml
- gke-namespace-settings.yaml
- gke-pv-pvc-settings.yaml
- gke-secret-settings.yaml
- gke-service-settings.yaml

5. 데이터 플로우


사용자 웹사이트 접근
사용자 인증
인증 성공
회원 등급 확인
문서 CRUD 요청
권한 검증
권한 있음
권한 없음
정상 CRUD 수행
작업 완료
비정상 접근 감지
인증 실패
접근 거부
접근 시도 로그
LangChain 분석 시작
공격 패턴 분석
LangGraph 상태 관리


공격 유형 판단
권한 상승 시도
무차별 대입
SQL 인젝션
기타 악의적 접근
권한 상승 공격 로그
브루트포스 공격 로그
데이터베이스 공격 로그
일반 보안 위협 로그
AI 브리핑 생성
위험도 평가
위험도 수준
높음
중간
낮음
즉시 알림 + 계정 차단
관리자 알림
로그 저장
보안 담당자 브리핑
정기 보고서 포함
대응 조치 실행
시스템 모니터링 지속