

---

## System Test 목적

- LogMate 시스템의 전체 통합 동작 검증
- 각 서버 간 통신, 로그 수집 및 분석 등 기능 정상 동작 확인

## System Test 범위

- Agent, Streaming Server, API Server, AI Server, Frontend

## System Test 절차

1. 각 서버 배포 및 구동
2. API 서버 인증 및 설정 Pull 수행 확인
3. 로그를 발생시키는 별도의 앱을 설치하여 Agent 로그 수집을 확인
4. 수집된 로그를 Streaming 서버로 전송
5. Kafka - OpenSearch - AI 서버 데이터 플로우 확인
6. Frontend 실시간 확인

---

## System Test 환경

항목	상세 내용
테스트 환경	Netlify + AWS 통합 환경 (실제 배포 버전 기준)
테스트 도구	Postman, OpenSearch DashBoard, Kafka UI
Frontend 환경	Netlify
Backend 연결 서버	API Server (AWS EC2), Streaming Server (AWS EC2), AI Server (AWS EC2)

## System Test 결과

TC ID	테스트 항목	테스트 환경 / 데이터	절차 요약	예상 결과	실제 결과	결과	관련 FR
ST-001	Agent ↔ API Server 인증 통신	Agent 실행, API Server 가동	Agent가 실행 시 API 서버 /api/v1/agents/auth에 ID/PW로 인증 요청	HTTP 200 응답, AccessToken 수신	AccessToken 정상 발급 및 로그 기록 확인	P	FR-1.1.5, FR-3.2.6
ST-002	Agent ↔ API Server 설정 Pull	API 서버에 Dashboard Config 존재	Agent 주기적 Pull 요청 수행	eTag 비교 후 설정 변경 시 JSON 설정 수신	변경 시 설정 반영 로그 출력	P	FR-1.2.1~1.2.4
ST-003	Agent ↔ Streaming Server 로그 전송	Agent tailing 활성	Agent가 수집 로그를 /api/v1/streaming/logs/springboot/{agentId}로 전송	Streaming 서버 로그 수신 및 ACK(200 OK) 반환	수신 로그 개수와 일치 확인	P	FR-1.3.1, FR-2.1.1
ST-004	Streaming Server ↔ Kafka Messaging	Kafka broker 연결	Streaming 서버가 수신 로그를 log-stream 토픽에 produce	Kafka consumer에서 동일 메시지 수신 확인	Kafka UI 및 로그에서 확인됨	P	FR-2.2.1

## System Test 결과

TC ID	테스트 항목	테스트 환경 / 데이터	절차 요약	예상 결과	실제 결과	결과	관련 FR
ST-005	Streaming Server ↔ AI Server 로그 분석 요청	AI 서버 REST API /api/v1/analyze 활성	Streaming 서버가 로그 batch를 AI 서버로 POST	AI 서버로부터 anomaly score 응답 수신	JSON 응답(score, label) 정상	P	FR-2.3.1, FR-5.1.1
ST-006	Streaming Server ↔ OpenSearch 로그 저장	OpenSearch 인덱스 생성됨	Streaming 서버가 Kafka에서 consume한 로그를 OpenSearch에 저장	log-yyyy.MM.dd 인덱스에 로그 적재	OpenSearch UI 검색 시 데이터 확인됨	P	FR-2.3.2
ST-007	Streaming Server ↔ Frontend 실시간 로그 스트리밍	WebSocket 연결	Frontend가 /ws/logs 구독 요청	Streaming 서버가 새 로그 발생 시 실시간 push	Frontend 로그 창에 실시간 표시	P	FR-2.3.3, FR-4.7.1
ST-008	Frontend ↔ API Server 사용자 인증	로그인 UI에서 POST /api/v1/auth/login	이메일/비밀번호 전송	JWT 토큰 및 사용자 정보 반환	로그인 성공 후 대시보드 이동	P	FR-3.1.1, FR-4.1.1

## System Test 결과

TC ID	테스트 항목	테스트 환경 / 데이터	절차 요약	예상 결과	실제 결과	결과	관련 FR
ST-009	Frontend ↔ API Server 설정 조회	로그인된 상태	GET /api/v1/dashboard/config 호출	JSON Config 응답 수신	대시보드 설정 값 일치	P	FR-3.2.5, FR-4.2.4
ST-010	API Server ↔ Frontend Webhook 연동	Slack Webhook 등록	API 서버가 알림 이벤트 발생 시 Slack POST 전송	Slack 메시지 수신 확인	Slack에 정상 알림 수신	P	FR-3.6.1, FR-3.6.3
ST-011	Frontend ↔ Streaming Server 로그 검색	로그 저장 완료 상태	/api/v1/streaming/logs/search?agentId=... 호출	검색 결과 JSON 리스트 반환	UI 결과 일치	P	FR-2.4.1
ST-012	AI Server ↔ Streaming Server 이상탐지 후 알림	비정상 로그 전송	AI 서버 anomaly score > 0.8 시 알림 생성	Streaming 서버가 알림 큐로 전달	Slack 및 UI 알림 표시	P	FR-2.3.1, FR-5.1.5

## System Test 분석

구분	분석 내용
통합 안정성	Agent-API-Streaming-AI-Frontend 전 구간에서 통신 성공률 100%. 로그 수집부터 시각화까지 단일 장애 없이 흐름 완성됨.
데이터 일관성	Kafka를 통한 로그 전송 시 송신/수신 로그 개수 일치, OpenSearch 인덱스 생성 및 OpenSearch UI에서의 검증으로 데이터 손실 없음 확인.
AI 분석 기능 검증	AI Server의 이상탐지(Anomaly Detection) 로직이 정상 작동, score 0.8 이상 로그에 대해 Slack/Web 알림이 정확히 표시됨.
구성관리 검증	eTag 기반 Config Pull 구조로 설정 변경 시 즉시 반영됨을 확인, 동적 재설정 기능의 실효성 입증.
보안 및 인증	Agent의 인증이 정상적으로 작동, 무단 접근 차단 확인.

## 보완점

구분	내용	개선 방향
1. 성능	대용량 로그 입력과 같은 처리의 지연 및 OpenSearch 인덱싱 및 Kafka 부하 검증 미흡	부하테스트 수행
2. 복원력	네트워크 단절, 서버 재시작 등 장애 상황에서 로그 재전송 로직 미검증	DLQ, Fallback Log File에 대한 로그 재전송에 대해 테스트 수행
3. 테스트 자동화	수동 수행, 반복 테스트 시 효율성 낮음	Postman/Newman 기반 자동화 테스트 구축 및 CI/CD 연동