

jikim

관리자 가이드

설치·설정·운영·장애 대응·보안

버전 v0.2.11
작성일 2026-09-12
문서 관리자 가이드

jikim 관리자 가이드

이 문서는 jikim v0.2.11 을 설치하고 지키는 사람을 위한 것입니다. 화면 캡처는 모두 jikim:v0.2.9 이미지를 전용 PostgreSQL과 함께 찍은 뒤 데모 데이터를 넣고 찍은 것입니다.

화면을 쓰는 방법은 [사용자 가이드](#)에 있습니다. 같은 내용을 두 번 적지 않았으니 사용자 문의를 받으면 그쪽을 먼저 보내십시오. 폐쇄망 반입 절차는 [오프라인 설치 가이드](#), 위협 모델과 암호화 경계는 [보안 운영 가이드](#), OpenBao 호환 상한은 [호환성 프로파일](#)에 있습니다.

1. 구성 요소

jikim은 컨테이너 하나와 외부 PostgreSQL 하나로 동작합니다. 서비스가 자체적으로 데이터베이스를 번들링하지 않으므로 사내 표준 PostgreSQL을 먼저 준비해야 합니다.

구성 요소	필수	제공 방식	하는 일
jikim 컨테이너	필수	릴리스 이미지 jikim:v0.2.11	Go 서버 하나가 관리 API, OpenBao 제한 호환 API, MCP, React 정적 자산을 모두 제공
PostgreSQL	필수	사내 표준 인스턴스	사용자·정책·시크릿 암호문·감사 로그 저장
TLS Reverse Proxy	운영 권장	사내 표준	외부 HTTPS 종단, Host · X-Forwarded-Proto 전달
Keycloak	선택	사내 표준	OIDC SSO 로그인
OpenAI-compatible AI Gateway	선택	사내 표준	AI 보안 도우미의 Upstream
Webhook 수신기	선택	사내 표준	서명된 운영 이벤트 수신

컨테이너가 주고받는 것은 다음과 같습니다.

방 향	상대	프로토콜	비고
수 신	브라우저·API·bao 호환 클라이언트·MCP 클라이언트	HTTP 8080	고정 포트. 컨테이너 내부에서 변경할 수 없습니다.
발 신	PostgreSQL	TCP 5432	POSTGRES_DSN 이 가리키는 주소
발 신	Keycloak	HTTPS	Discovery·Token·JWKS·UserInfo·End Session
발 신	AI Gateway	HTTPS(SSE)	/v1/chat/completions 로 정규화
발 신	Webhook 수신기	HTTPS	HMAC-SHA256 서명 전송

React 정적 자산은 이미지 안의 /app/web 에 들어 있고 Go 서버가 SPA fallback을 제공합니다. 런타임에 CDN이나 npm·Go 저장소가 필요하지 않습니다.

2. 설치

릴리스 자산으로 처음부터 끝까지 올리는 절차입니다. 아래 명령은 그대로 붙여 넣을 수 있습니다.

2.1 필요한 자원

항목	값
컨테이너 포트	8080 (호스트 노출은 Reverse Proxy 뒤로 두는 것을 권장)
볼륨	없음. 컨테이너는 read_only 로 뜨고 /tmp 만 tmpfs 64MB
내부 CA 파일(선택)	/app/certs/internal-ca.crt 에 읽기 전용 bind mount
PostgreSQL	전용 데이터베이스와 최소 권한 계정 하나
이미지 용량	압축 번들 약 13MB, 적재 후 약 43MB
컨테이너 자원	pids_limit 256 , 종료 유예 30초 (docker-compose.yml 기준)

2.2 번들 검증과 적재

GitHub Release에 올라오는 자산은 두 개입니다.

```
sha256sum --check jikim-v0.2.11.tar.gz.sha256
docker load --input jikim-v0.2.11.tar.gz
docker image inspect jikim:v0.2.11
```

체크섬 확인 없이 적재하지 마십시오. 반입 경로에서 파일이 바뀌었는지 확인할 유일한 수단입니다.

2.3 PostgreSQL 준비

전용 데이터베이스와 계정을 만들고 그 계정에만 권한을 줍니다. 스키마 마이그레이션은 서비스가 기동할 때 직접 수행하므로 별도 도구가 필요하지 않습니다.

```
CREATE ROLE jikim_app LOGIN PASSWORD '내부-정책에-맞는-값';
CREATE DATABASE jikim OWNER jikim_app;
```

2.4 암호화 키 준비

ENCRYPTION_KEY 는 32바이트 원문, 64자리 hex 또는 32바이트 base64 중 하나입니다.

```
openssl rand -hex 32
```

이 값을 잃으면 저장된 암호문을 복구할 수 없습니다. 만들자마자 키 관리 시스템에 넣고 PostgreSQL 백업과 다른 통제 영역에 두십시오.

2.5 Compose 기동

```
export POSTGRES_DSN='postgres://jikim_app:REDACTED@postgres.internal:5432/jikim?sslmode=verify-full'
export BOOTSTRAP_ADMIN='bootstrap-admin'
export BOOTSTRAP_ADMIN_PASSWORD='REDACTED-AT-LEAST-12-CHARS'
export ENCRYPTION_KEY='REDACTED-64-HEX-CHARS'
docker compose up --detach
```

사내 CA로 Keycloak·AI·Webhook의 TLS를 검증해야 하면 인증서를 `certs/internal-ca.crt` 에 두고 환경변수를 늘리지 않고 override를 함께 씁니다.

```
docker compose -f docker-compose.yml -f docker-compose.internal-ca.yml up --detach
```

2.6 기동 확인

```
curl --fail http://127.0.0.1:8080/healthz
curl --fail http://127.0.0.1:8080/readyz
curl --fail http://127.0.0.1:8080/v1/sys/health
```

- GET /healthz 는 프로세스가 살아 있지만 답합니다.
- GET /readyz 는 PostgreSQL에 2초 제한으로 ping 한 뒤 답합니다. 저장소에 닿지 못하면 503 과 not_ready , 데이터베이스 연결을 확인할 수 없습니다 를 돌려줍니다.
- GET /v1/sys/health 는 OpenBao 클라이언트와 Load Balancer가 쓰는 경로입니다. v0.2.11 은 저장소 미도달 을 seal과 같은 운영 상태로 보아 "sealed": true 와 503 을 반환합니다. probe 설정을 이식할 때는 activecode * sealedcode 쿼리로 상태 코드를 바꿀 수 있고, 100~599 정수가 아니면 조용히 무시하지 않고 400 으로 거부합니다.

2.7 최초 관리자 계정

1. /readyz 가 성공한 뒤 https://<서비스 주소>/login 으로 들어갑니다.
2. BOOTSTRAP_ADMIN 과 BOOTSTRAP_ADMIN_PASSWORD 로 로그인합니다.
3. 서비스 관리 → 사용자 관리에서 개인 관리자 계정을 만들거나 OIDC 역할을 매핑합니다.
4. 초기화 전용 계정의 비밀번호를 회전하고 사용 범위를 제한합니다.
5. 로그인 화면과 프로필 메뉴의 버전이 배포한 이미지 태그와 같은지 확인합니다.

부트스트랩 값은 기존 관리자의 비밀번호를 매 기동마다 덮어쓰기 위한 수단이 아닙니다. 복구 절차를 검증하기 전에 컨테이너 환경에서 임의로 바꾸지 마십시오.

3. 설정

3.1 환경변수 전수 표

jikim 애플리케이션이 읽는 환경변수는 정확히 다음 네 개입니다 (internal/config/config.go). 나머지 운영 설정은 모두 관리 화면에 있습니다.

이름	기본값	필수	설명
POSTGRES_DSN	없음	필수	메타데이터와 암호문 저장소 연결 문자열. 운영에서는 <code>sslmode=verify-full</code> 과 최소 권한 계정을 씁니다. 예: <code>postgres://jikir_app:REDACTED@postgres.internal:5432/jikir?sslmode=verify-full</code>
BOOTSTRAP_ADMIN	없음	필수	최초 관리자 아이디. 3~128자. 개인 실명 계정보다 초기화 전용 계정을 권장합니다. 예: <code>bootstrap-admin</code>
BOOTSTRAP_ADMIN_PASSWORD	없음	필수	최초 관리자 비밀번호. 12자 이상. Secret 파일이나 안전한 주입 수단으로 전달합니다. 예: <code>REDACTED-AT-LEAST-12-CHARS</code>
ENCRYPTION_KEY	없음	필수	AES-256-GCM 마스터 키(KEK). 32바이트 원문, 64자리 hex 또는 32바이트 base64. 예: <code>REDACTED-64-HEX-CHARS</code>

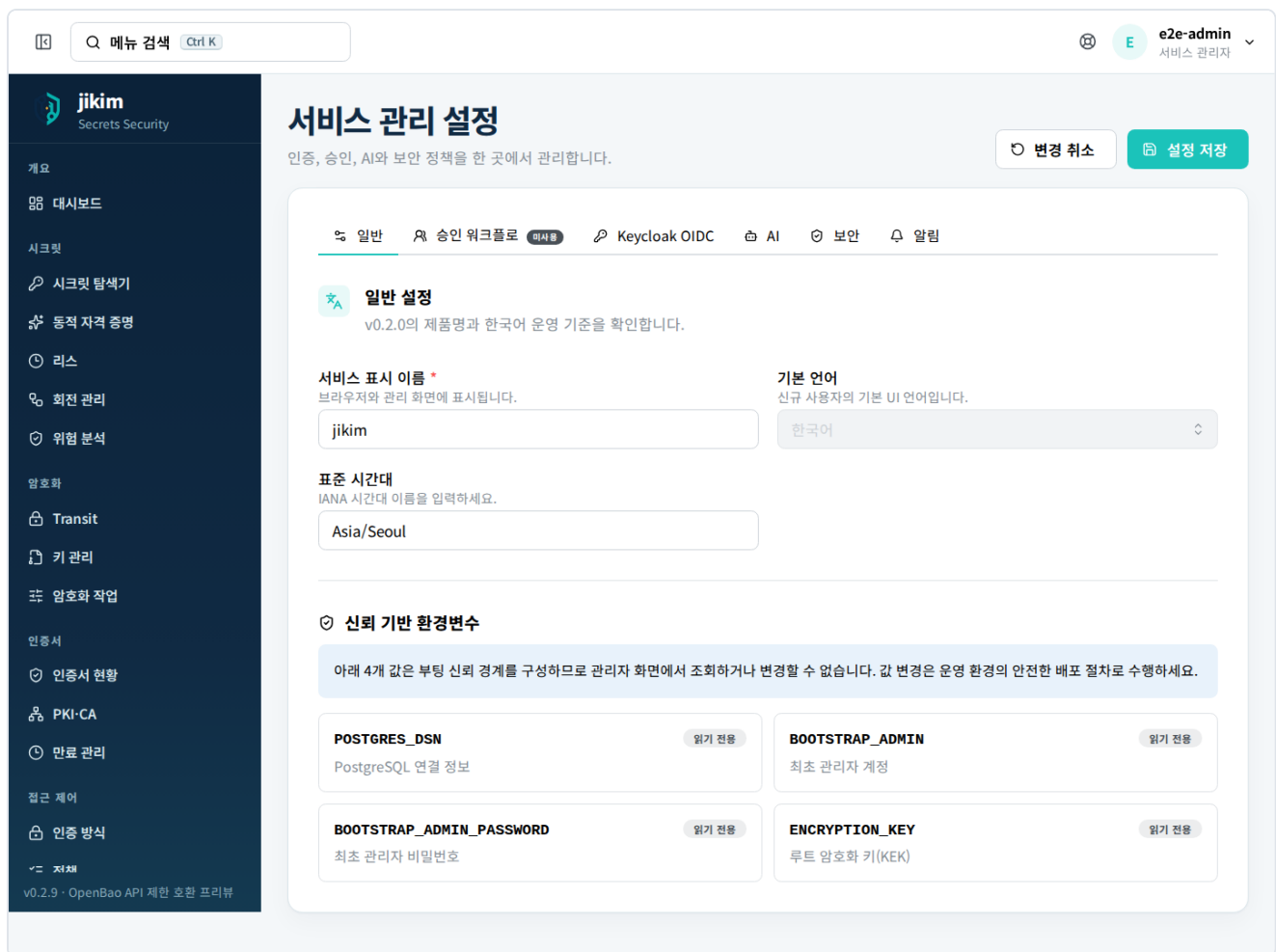
네 값 중 하나라도 비어 있으면 서버는 기동하지 않고 필수 환경변수가 없습니다: <이름 목록> 을 남기고 종료합니다. 길이 조건을 어기면 `BOOTSTRAP_ADMIN`은 3~128자여야 합니다 또는 `BOOTSTRAP_ADMIN_PASSWORD`는 12자 이상이어야 합니다 가 나옵니다.

컨테이너 자체는 위 네 개 외에 하나만 더 봅니다. 파일 하나이지 환경변수는 아닙니다.

경로	필수	설명
<code>/app/certs/internal-ca.crt</code>	선택	PEM 인증서가 있으면 outbound TLS 신뢰 목록에 추가합니다. 없으면 시스템 신뢰 저장소만 씁니다.

3.2 관리 화면 설정

일상 설정은 서비스 관리 → 서비스 설정 (`/admin/settings`) 한 화면에 모여 있습니다. `admin` 역할만 열 수 있습니다.



서비스 관리 설정 — 일반·승인 워크플로·Keycloak OIDC·AI·보안·알림 탭과 읽기 전용 환경변수 목록

일반 탭 아래쪽 신뢰 기반 환경변수 카드는 위 네 값을 이름과 용도만 보여 줍니다. 값은 화면에서 조회하거나 변경할 수 없습니다. 부팅 신뢰 경계를 화면에서 만질 수 없게 만든 것이 의도된 설계입니다.

탭	설정하는 것
일반	서비스 표시 이름, 기본 언어, 표준 시간대(IANA 이름)
승인 워크플로	워크플로 사용 여부, 기본 검토 역할, 필수 승인 수, 승인 적용 작업
Keycloak OIDC	Issuer URL, Client ID·Secret, Scopes, 사용자명·그룹·역할 Claim, Callback URL 확인, 내부 HTTP 허용
AI	사용 여부, Base URL, 모델, 인증 방식, API Key, 최대 출력 토큰, 요청 제한 시간, 내부 HTTP 허용
보안	세션 제한 시간(5~1440분), 최소 비밀번호 길이(12~128), 감사 로그 보존 기간(1~3650일, 프리뷰), 허용 네트워크(프리뷰), 로컬 로그인 허용, 최초 비밀번호 변경 요구(프리뷰)
알림	서명 Webhook 사용 여부, Webhook URL, 서명 Secret, 전송 이벤트, 내부 HTTP 허용
방문 추적	사용 여부, 제공자(Momento·GA4·GTM·Matomo·직접 붙여넣기), 수집기 주소와 사이트 ID, 같은 오리진 프록시, 삽입 위치, 추가 허용 출처, 관리 화면 포함, 수집기 내부 HTTP 허용, 정책이 차단한 출처 목록

값의 타입이 맞지 않으면 조용히 버리지 않고 400 으로 거부합니다. 저장이 성공했다는 것은 그 값이 실제로 적용되었다는 뜻입니다. 단, **프리뷰**로 표시된 항목은 값을 저장하고 검증하지만 강제하지 않습니다 — 감사 로그 보존 기간은 자동 삭제를 수행하지 않고, 허용 네트워크는 Reverse Proxy나 방화벽에서 강제해야 하며, 최초 비밀번호 변경 요구는 로그인 시 강제되지 않습니다. 화면의 안내 문구가 이를 그대로 적어 두었습니다.

3.3 Keycloak OIDC 연결

Keycloak 쪽에서 먼저 준비합니다.

1. 전용 Realm 또는 기존 보안 Realm에 OIDC Client를 만듭니다.
2. Client authentication을 켜고 Authorization Code Flow를 씁니다.
3. 관리 화면의 **Callback URL** 칸에 표시되는 절대 URL을 Keycloak의 Valid redirect URIs에 그대로 등록합니다. 형식은 `https://<서비스 Origin>/api/v1/oidc/callback` 이며 프론트엔드 경로 `/oidc/callback` 과 혼동하지 않습니다.
4. Web origins는 서비스의 실제 HTTPS Origin으로 제한합니다. 와일드카드는 쓰지 않습니다.
5. 그룹·역할 claim을 토큰에 넣되 최소 정보만 전달합니다.

jikim의 **Keycloak OIDC** 탭에서 Issuer URL, Client ID·Secret, Scopes(보통 openid profile email groups), 사용자명·그룹·역할 Claim 이름을 넣습니다. Issuer의 `/.well-known/openid-configuration` 으로 Authorization·Token·JWKS·UserInfo·End Session 메타데이터를 발견합니다.

Discovery 연결 테스트가 확인하는 범위는 다음뿐입니다.

- TLS 인증서 체인을 컨테이너가 신뢰하는지
- Issuer가 정확히 일치하는지
- Discovery 문서를 읽고 표시할 메타데이터가 있는지

Client ID·Secret의 실제 인증, JWKS 다운로드와 ID Token 서명 검증, Keycloak에 등록된 Redirect URI, claim·역할 매핑은 이 테스트로 확인되지 않습니다. **저장 후 전용 테스트 계정으로 실제 SSO 로그인을 한 번 완료하십시오.**

브라우저가 로그인을 시작하면 프론트엔드는 같은 Origin의 `/oidc/callback` 을 복귀 화면으로 요청합니다. 공급자는 위의 Backend Callback으로 돌아오고, jikim은 state·nonce·PKCE와 ID Token을 검증한 뒤 짧은 TTL의 일회용 code 만 프론트엔드에 넘깁니다. 프론트엔드는 `POST /api/v1/oidc/exchange` 로 이를 교환하며 세션은 응답 JSON이 아니라 HttpOnly 쿠키로 설정됩니다. Reverse Proxy는 원래 Host 와 X-Forwarded-Proto 를 정확히 전달해야 합니다.

Client Secret은 민감 설정으로 암호화 저장되고 조회 응답에는 설정 여부만 나옵니다. 비상 로컬 로그인을 검증하기 전에 OIDC를 유일한 관리 경로로 만들지 마십시오.

3.4 승인 워크플로

기본값은 비활성입니다. 꺼져 있으면 생성·변경 작업에 검토 단계를 만들지 않고 즉시 반영합니다. `v0.2.11` 에서 설정할 수 있는 것은 워크플로 사용 여부, 적용 작업(시크릿 생성·변경, 시크릿 폐기), 검토 역할(`manager` 또는 `admin`)입니다.

승인은 1인 검토이며 요청자 본인 승인은 항상 금지됩니다. 환경·위험 등급별 조건, 다단계 승인, 요청 만료, 키 회전·정책 변경 승인은 후속 범위입니다. 저장한 뒤 일반 사용자 계정과 검토자 계정으로 각각 한 번 시험하십시오. 요청자가 자기 요청을 승인할 수 없어야 하고, 승인·반려 결과가 감사 로그에 남아야 합니다.

3.5 AI 연동

기본 비활성입니다. 활성화하면 Base URL은 서버에서 `/v1/chat/completions` 형태로 정규화됩니다.

auth_type	Upstream 요청 헤더
bearer	Authorization: Bearer <API key>
api-key	api-key: <API key>
none	인증 헤더 없음

- API Key는 암호화 저장되고 조회 응답에는 설정 여부만 나옵니다. `none` 이 아닌 방식을 쓰려면 API Key를 먼저 저장해야 하며, 없으면 선택한 AI 인증 방식에는 API Key가 필요합니다 로 거부합니다.
- `max_tokens` 는 1~262144, `timeout_seconds` 는 10~3600입니다. 모델의 컨텍스트 상한과 출력 상한은 다른 개념이므로 제공자의 실제 출력 한도보다 낮게 잡으십시오.
- 요청 제한은 사용자별 인메모리 기준으로 분당 30회, 동시 2개이며 초과 시 429 와 Retry-After: 60 을 돌려줍니다. 서버를 재시작하면 카운터가 초기화되므로 외부 API Gateway 제한을 함께 두는 것을 권장합니다.

저장한 뒤 **저장된 AI 연결 테스트**를 실행하십시오. `POST /api/v1/integrations/ai/test` 가 저장된 설정으로 `stream: true` , `max_tokens: 1` 요청을 보내고 Upstream이 HTTP 2xx, Content-Type: text/event-stream , 하나 이상의 data: 이벤트를 반환하는지 확인합니다. 단순 TCP 연결 확인이 아닙니다.

3.6 서명 Webhook 알림

알림 탭에서 Webhook URL, 전송 이벤트, 서명 Secret, 내부 HTTP 허용 여부를 관리합니다. HTTPS가 기본이고 비 loopback HTTP는 **Webhook 내부 HTTP 허용**을 명시적으로 켜는 경우에만 저장할 수 있습니다. 서명 Secret은 32자 이상이어야 하며 처음 저장할 때 비어 있으면 안전한 값이 자동 생성됩니다. 회전하면 이후 전송부터 새 Secret으로 서명 되므로 수신 측을 함께 전환하십시오.

지원 이벤트는 여덟 개입니다.

```
approval.requested approval.approved approval.rejected
secret.created      secret.updated      secret.rotated
secret.deleted      rotation.failed
```

본문에는 delivery_id, event, occurred_at, resource, actor_id, request_id 와 민감하지 않은 data 가 들어갑니다. 시크릿 평문은 들어가지 않습니다. 수신 요청 헤더는 다음과 같습니다.

```
X-Jikim-Delivery: <delivery UUID>
X-Jikim-Event: <event name>
X-Jikim-Timestamp: <Unix seconds>
X-Jikim-Signature-256: sha256=<hex digest>
```

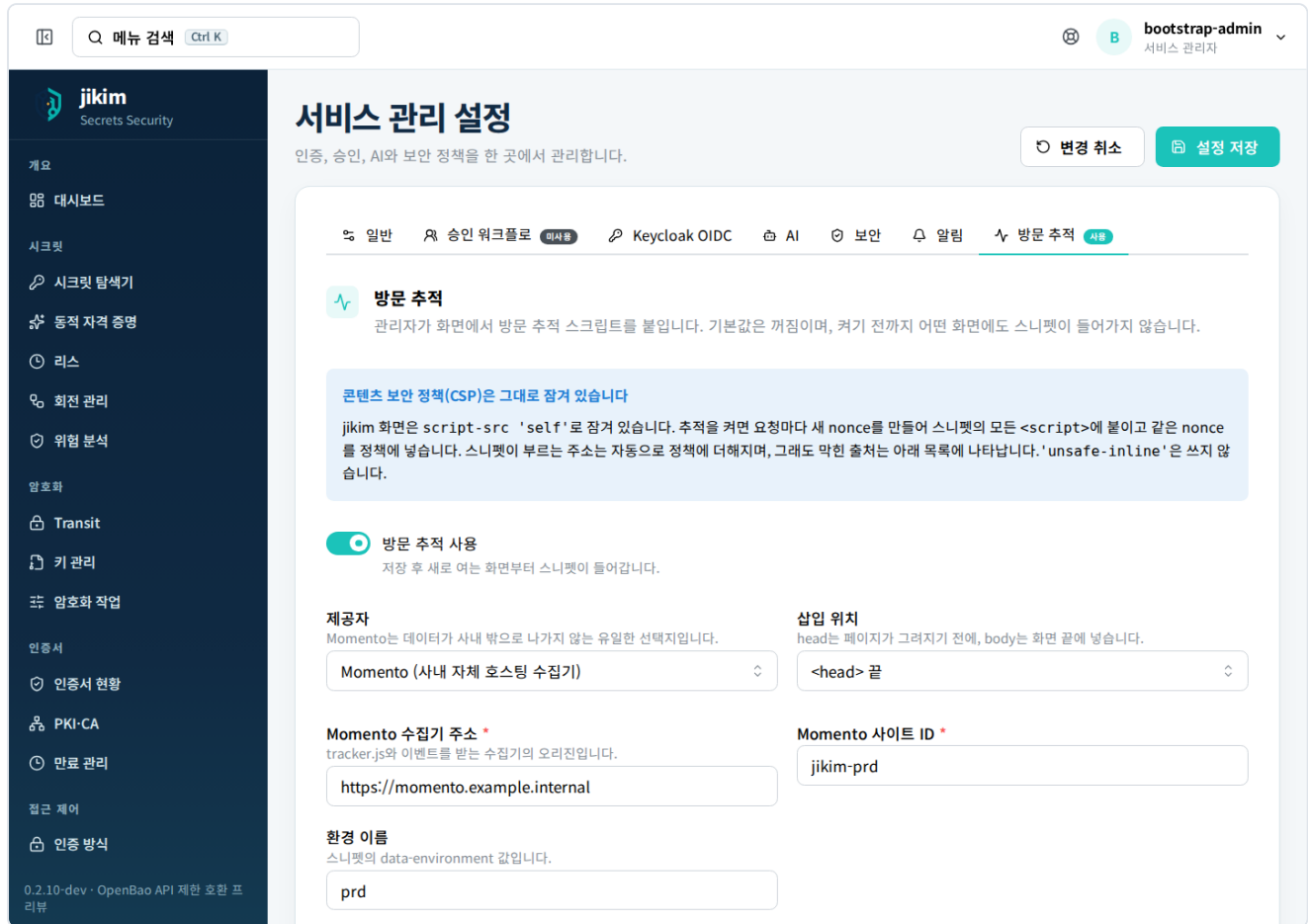
서명 입력은 X-Jikim-Timestamp + "." + raw HTTP body 이고 HMAC-SHA256을 씁니다. 수신기는 JSON을 다시 직렬화하기 전에 원문 body로 상수 시간 비교를 하고, timestamp 허용 오차와 delivery_id 중복을 검사해야 합니다. HTTP 2xx만 성공으로 처리하며 일반 이벤트 전송 제한 시간은 10초입니다.

작업	API	역할
서명된 테스트 전송	POST /api/v1/integrations/webhook/test	admin
전송 이력 조회	GET /api/v1/integrations/webhook/deliveries? status=failed&limit=50&offset=0	admin , auditor
저장된 payload 재전송	POST /api/v1/integrations/webhook/deliveries/{id}/retry	admin

이력 상태는 pending · delivered · failed 이며 시도 횟수, 응답 상태, 최근 오류와 시각을 기록합니다. 자동 백오프 재시도는 없습니다. 실패 건은 원인을 해소한 뒤 위 API로 수동 재시도하며, 재시도에는 현재 저장된 URL과 서명 Secret을 씁니다. 동시 비동기 전송 슬롯 16개가 모두 차면 그 이력은 실패로 기록됩니다.

3.7 방문 추적 스크립트

방문 추적 탭에서 화면에 방문 추적 스크립트를 붙입니다. 기본값은 꺼짐이며, 켜기 전까지 어떤 화면에도 스니펫이 들어가지 않고 응답 헤더도 이전 릴리스와 같습니다. 재배포 없이 화면에서 켜고 끄며, 값은 다른 관리 설정과 같이 PostgreSQL의 `settings` 테이블(`tracking` 키)에 저장됩니다.



서비스 관리 설정 — 방문 추적 탭. CSP 안내, 제공자 선택, Momento 수집기 주소·사이트 ID·환경, 같은 오리진 프록시 스위치

항목	뜻
방문 추적 사용	기본 OFF. 저장 후 새로 여는 화면부터 스니펫이 들어갑니다.
제공자	momento (첫 자리, 권장) · ga4 · gtm · matomo · custom (직접 붙여넣기)
Momento 수집기 주소 · 사이트 ID · 환경 이름	사내 Momento 수집기의 오리진, 사이트 id, data-environment 값(기본 prd)
같은 오리진 프록시 사용	기본 ON. jikim이 /momento/* 를 수집기로 넘기고 스니펫에 data-endpoint="/momento" 를 줍니다.
측정 ID · 컨테이너 ID	GA4·GTM의 id
Matomo 주소 · 사이트 ID	Matomo 설치 주소와 사이트 id
추적 스니펫	직접 붙여넣는 <script> 태그. 8KB를 넘으면 400 으로 거부합니다.
추가 허용 출처	스니펫에서 자동으로 읽지 못한 출처를 실험표로 구분해 적는 자리
관리 화면도 추적	기본 OFF. /admin/*, /infrastructure/*, /access/authentication 으로 들어온 요청에는 넣지 않습니다.
삽입 위치	head (기본) 또는 body 끝
수집기 내부 HTTP 허용	기본 OFF. 다른 연동과 같은 규칙으로, 비 loopback HTTP 주소는 커야만 저장됩니다.

Momento를 먼저 고르십시오. Momento는 사내 자체 호스팅 수집기라 데이터가 밖으로 나가지 않는 유일한 선택지입니다. 같은 오리진 프록시를 켜 두면(기본) 브라우저는 jikim 오리진만 보므로 콘텐츠 보안 정책에 외부 출처가 아예 등장하지 않습니다. 프록시는 추적이 켜져 있고 제공자가 Momento일 때만 열리며, 그 밖에는 /momento/* 가 404 입니다. 프록시는 GET · HEAD · POST 만 넘기고, 브라우저가 붙인 Cookie · Authorization · X-Vault-Token 을 지운 뒤 전달하며, 수집기가 돌려준 Set-Cookie 는 브라우저에 넘기지 않습니다. 본문은 64KB, 제한 시간은 10초입니다.

```
<!-- 프록시를 켜올 때 화면에 들어가는 스니펫 (nonce는 요청마다 다릅니다) -->
<script nonce="..." async src="/momento/tracker.js"
    data-site-id="<사이트 ID>" data-environment="prd"
    data-contract-version="1" data-endpoint="/momento"></script>
```

콘텐츠 보안 정책(CSP)과 nonce

jikim 화면은 script-src 'self' 로 잠겨 있습니다. 스니펫을 그냥 붙이면 브라우저가 조용히 차단하고 관리자는 화면이 왜 비어 있는지 알 길이 없습니다. 그래서 추적을 켜면 서버가 다음을 함께 합니다.

1. 화면(SPA 셸)을 내려 줄 때마다 새 nonce를 만들어 스니펫의 모든 <script> 태그에 붙이고, 같은 nonce를 응답 헤더의 script-src 에 'nonce-...' 로 넣습니다. 'unsafe-inline' 은 쓰지 않습니다 — 한 번 풀면 그 앱의 모든 인라인 스크립트가 함께 허용되고, 추적을 끈 뒤에도 정책이 느슨한 채로 남기 때문입니다.

2. 제공자가 정해 주는 출처(GA4·GTM의 Google 주소, Matomo 주소, 프록시를 끈 Momento 주소)와 직접 붙여 넣은 스크립트 문자열 안의 http(s) 출처를 읽어 script-src · connect-src · img-src 에 더합니다.
3. 추적이 켜져 있는 동안에만 report-uri /api/v1/tracking/csp-report 를 정책에 넣습니다. 브라우저가 차단한 요청을 여기로 신고하면 서버가 출처와 지시어를 메모리에 기억합니다. 같은 차단이 페이지마다 반복되므로 횟수가 아니라 서로 다른 출처가 중요하며, 100개를 넘으면 가장 오래된 것부터 버립니다. 이 신고는 감사 로그에 남기지 않습니다.

탭 아래쪽 정책이 차단한 출처 표가 그 기록입니다. 막힌 출처 옆의 허용을 누르면 그 출처가 추가 허용 출처에 더해져 다음 화면 요청부터 정책에 반영됩니다. 기록 지우기로 비운 뒤 화면을 다시 열어 아직 막히는 것이 있는지 확인하십시오. 이미 허용된 출처는 허용됨으로 표시됩니다.

추적을 끄면 정책은 이전 릴리스와 바이트 단위로 같은 값으로 돌아갑니다. /api/* , /v1/* , /mcp , /healthz , /readyz , /momento/* 처럼 브라우저가 그리지 않는 응답에는 스크립트를 넣지 않고 정책도 default-src 'none' 으로 더 좁힙니다. 설정 저장소에 닿지 못하면 그 응답은 추적 없이(이전 정책으로) 내려가므로 로그인 화면이 함께 멈추지는 않습니다.

주의할 점 두 가지입니다. 화면은 SPA이므로 "관리 화면도 추적"이 꺼져 있어도 /dashboard 로 들어온 뒤 화면 안에서 관리 메뉴로 이동하면 처음 실린 스크립트가 계속 동작합니다. 제외는 브라우저가 처음 연 주소 기준입니다. 그리고 로그인 화면에도 스크립트가 들어가므로 개인 식별 값을 보내는 스크립트는 붙이지 마십시오.

작업	API	역할
차단 출처 목록	GET /api/v1/tracking/violations	admin
차단 기록 지우기	DELETE /api/v1/tracking/violations	admin
차단 출처를 허용 목록에 추가	POST /api/v1/tracking/violations/allow ({ "origin": "https://..." })	admin
브라우저 CSP 신고 수신	POST /api/v1/tracking/csp-report	인증 없음(브라우저가 보냄), 항상 204

4. 계정과 권한

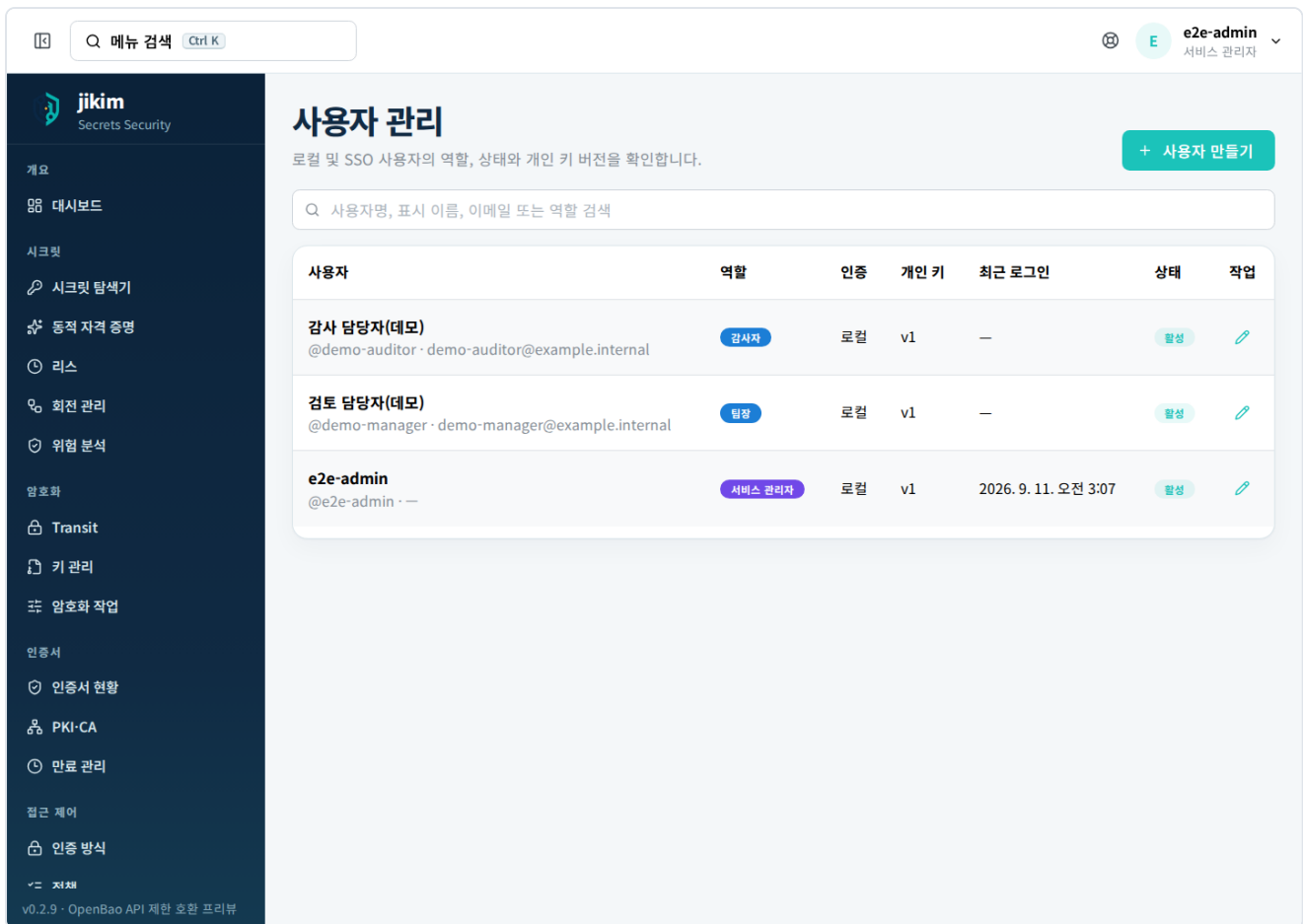
4.1 역할 네 종류

역할	화면 표시	할 수 있는 일
admin	서비스 관리자	모든 시크릿 경로 통과, 사용자·정책·설정·연동 관리, 감사 조회, 인프라·Namespace 화면
manager	팀장	모든 시크릿 경로 통과, 애플리케이션·정책 생성·변경, 승인 검토, 감사 조회. 설정과 사용자 생성은 못 합니다.
user	일반 사용자	붙은 정책이 허용하는 경로만. 자기 토큰과 개인 키 관리
auditor	감사자	목록 조회(capability list)만 통과. 감사 로그·정책·사용자 조회 가능, 값 복호화 불가

admin 과 manager 는 정책 평가를 통과합니다(internal/store/resources.go 의 canAccessWith). 즉 이 두 역할에게는 경로 정책이 적용되지 않습니다. auditor 는 list 만 허용됩니다. 정책으로 권한을 좁혀야 하는 계정에는 user 역할을 주십시오.

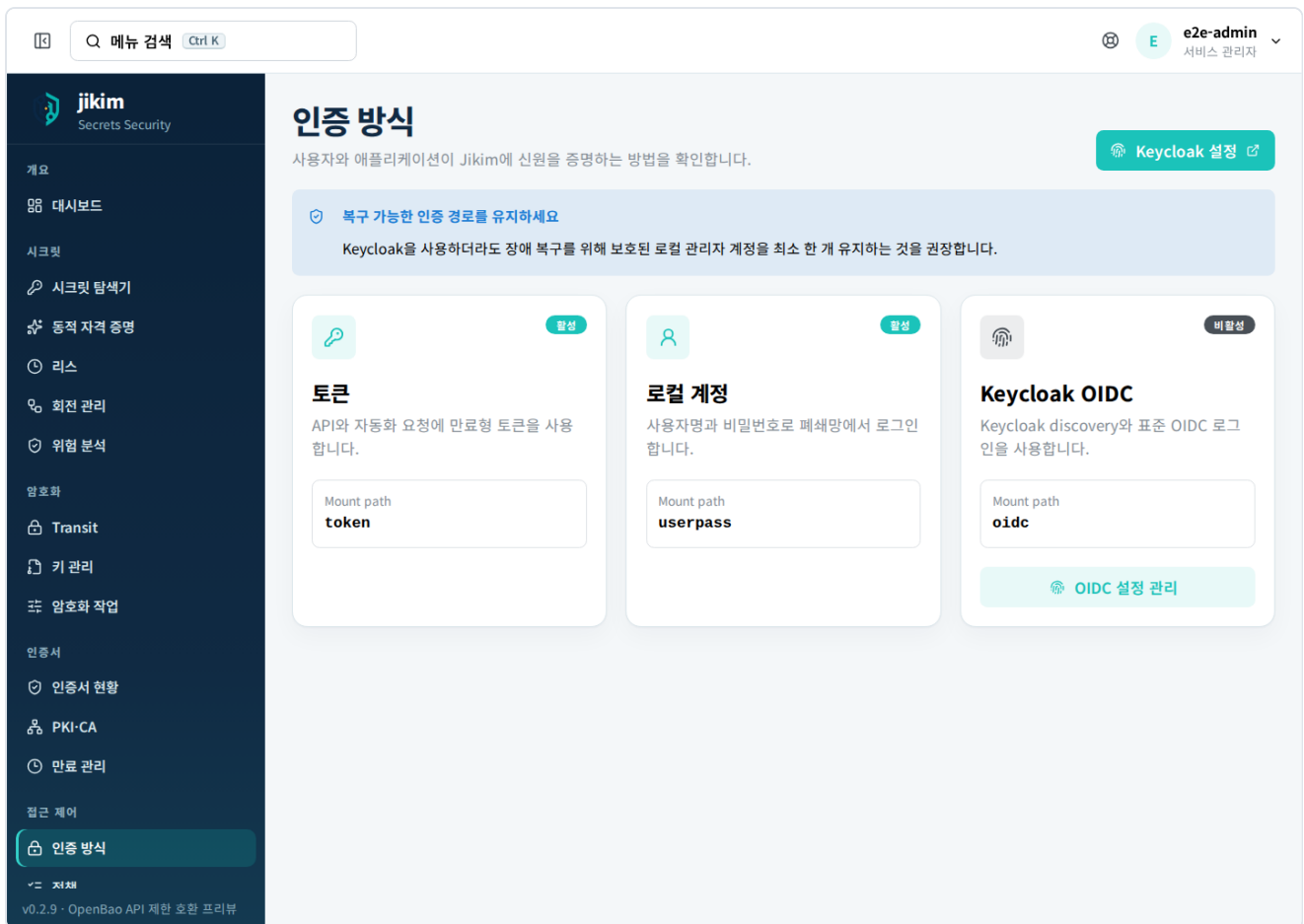
4.2 사용자 만들기

서비스 관리 → 사용자 관리(/admin/users)입니다. 조회는 admin · manager · auditor , 생성·변경·삭제는 admin 만 가능합니다.



- 사용자 만들기로 아이디, 표시 이름, 이메일, 역할, 비밀번호를 넣습니다. 비밀번호가 보안 탭의 최소 길이보다 짧으면 비밀번호가 최소 길이 정책보다 짧습니다 로 거부합니다.
- 인증 열이 로컬 인지 SSO인지 알려 줍니다. 개인 키 열은 그 사용자의 개인 키 버전입니다.
- 사용자를 만들면 개인 키가 함께 만들어집니다. 별도 발급 절차는 없습니다.

접근 제어 → 인증 방식 (/access/authentication)에서 현재 켜져 있는 로그인 경로를 한눈에 봅니다. admin 만 열 수 있습니다.

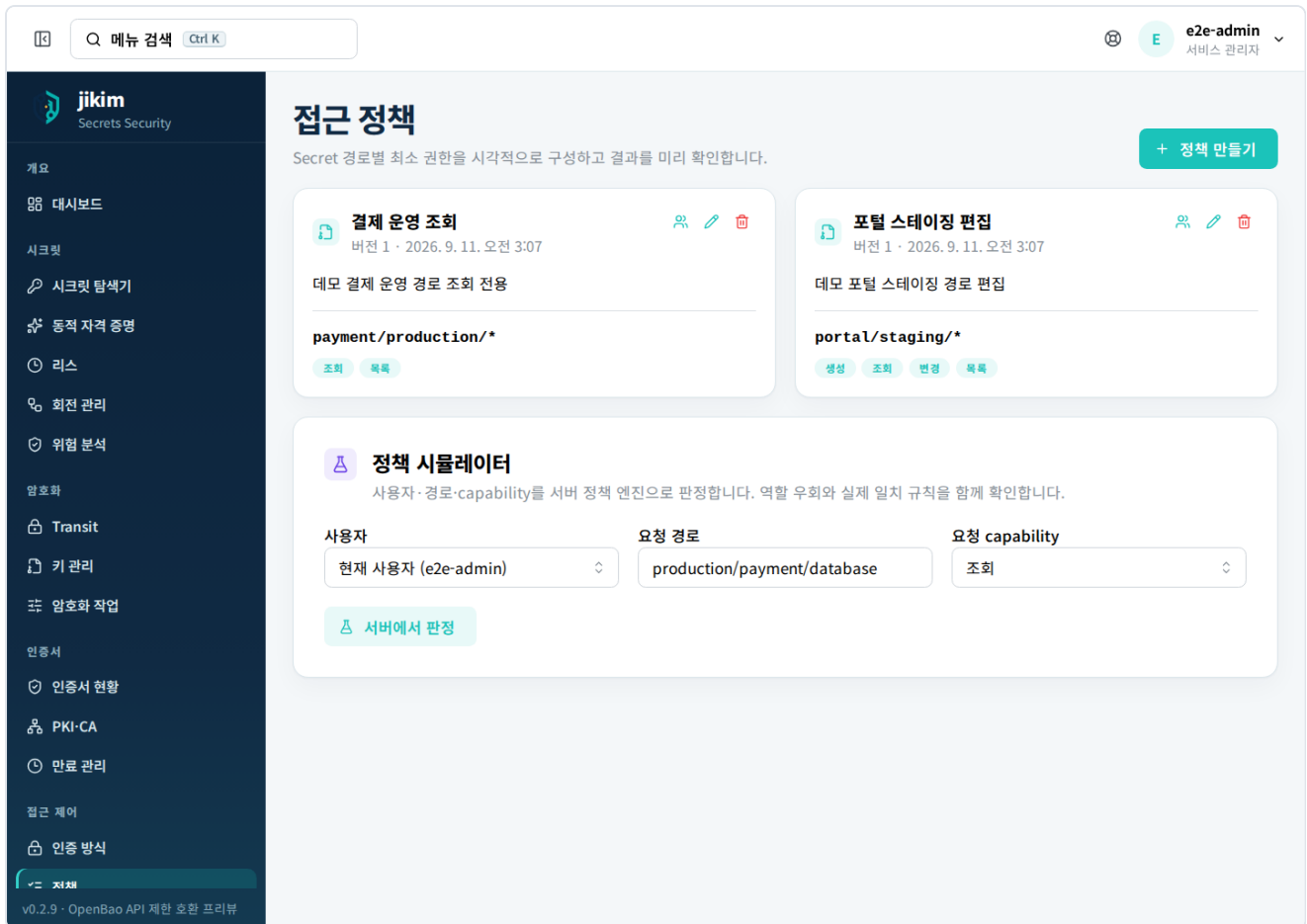


인증 방식 — 토큰·로컬 계정·Keycloak OIDC의 활성 여부와 mount path

화면 위쪽 안내대로, Keycloak을 쓰더라도 보호된 로컬 관리자 계정을 최소 한 개 유지하십시오. 보안 탭의 로컬 로그인 허용을 끄고 OIDC가 고장 나면 들어갈 방법이 없습니다.

4.4 정책으로 경로 권한 줘주기

접근 제어 → 정책 (/access/policies)에서 경로 패턴과 capability를 묶고, 카드의 사람 아이콘으로 적용할 사용자를 지정합니다.



접근 정책 — 경로별 capability와 서버 정책 엔진으로 판정하는 시뮬레이터

capability는 create · read · update · delete · list · rotate · encrypt · decrypt 여덟 개이며 그 밖의 값은 저장 시 알 수 없는 capability <이름>으로 거부됩니다. 정책을 만든 뒤 **정책 시뮬레이터**에서 실제 사용자·경로·capability로 판정을 돌려 의도한 대로 좁혀졌는지 확인하십시오. 역할 우회 여부까지 함께 알려 줍니다.

4.5 키 인벤토리

암호화 → **키 관리**(/encryption/keys)는 admin · manager 만 볼 수 있고 서비스 키(transit)와 사용자별 개인 키(personal)를 함께 보여 줍니다.

5. 운영

5.1 상태 점검 엔드포인트

경로	메서드	인증	용도
/healthz	GET	없음	프로세스 생존. 감사 로그에 남지 않습니다.
/readyz	GET	없음	PostgreSQL 2초 ping 포함. 감사 로그에 남지 않습니다.
/v1/sys/health	GET	없음	OpenBao 호환 probe. 저장소 미도달 시 "sealed": true + 503. 감사 로그에 남지 않습니다.
/api/v1/version	GET	없음	배포된 버전 확인
/api/v1/capabilities	GET	없음	구현 역량 조회
/api/openapi.json	GET	없음	OpenAPI 3.1 문서

docker-compose.yml 의 healthcheck는 /healthz 를 3초 제한으로 30초마다 봅니다. Load Balancer에는 저장소 상태까지 보는 /readyz 또는 /v1/sys/health 를 쓰십시오.

5.2 로그

로그는 표준 출력으로 나가는 JSON 한 줄씩(slog JSON handler, 레벨 INFO)입니다. 파일로 쓰지 않으므로 docker logs 또는 사내 수집기로 받습니다.

```
docker compose logs --follow jikim
```

기동할 때 다음이 순서대로 나옵니다.

```
{"level":"INFO","msg":"bootstrap 관리자 준비","username":"...","created":true}
{"level":"INFO","msg":"웹 UI 활성화","directory":"/app/web"}
{"level":"INFO","msg":"jikim 시작","address":":8080","version":"v0.2.11"}
```

종료 신호를 받으면 종료 신호 수신 을 남기고 최대 20초 동안 진행 중인 요청을 마무리합니다.

5.3 백업과 복구

서비스 상태의 원본은 PostgreSQL 암호문과 마스터 ENCRYPTION_KEY 두 개입니다. 둘을 서로 다른 통제 영역에 백업하십시오. 한쪽만 있으면 복구할 수 없습니다.

1. PostgreSQL 일관 백업을 수행합니다.
2. 마스터 키를 키 관리 시스템에서 따로 백업합니다.
3. 복구 격리망에 같은 버전 이미지를 적재합니다.
4. DB를 복구한 뒤 같은 키로 /readyz 와 로그인을 확인합니다.
5. 시크릿 샘플을 복호화하고 감사 이벤트가 남는지 확인합니다.
6. OIDC 로그인, AI SSE, 서명 Webhook 전송과 이력 조회를 시험합니다.

백업만 만들고 복구를 시험하지 않은 상태는 복구 가능 상태가 아닙니다.

5.4 업그레이드와 롤백

1. 새 버전의 체크섬과 [호환성 프로파일](#)을 확인합니다.
2. DB를 백업하고 복구 지점을 기록합니다.
3. 별도 환경에서 마이그레이션과 주요 API를 시험합니다.
4. 기존 이미지 태그를 지우지 않은 채 새 태그를 적재합니다.
5. 유지보수 시간에 컨테이너를 교체하고 /readyz ·로그인·버전 표시를 확인합니다.

되돌릴 때는 docker-compose.yml 의 이미지 태그를 이전 값으로 바꿔 다시 올립니다. 단, DB 마이그레이션이 하위 호환 인지 확인하기 전에는 이미지 교체만으로 롤백된다고 가정하지 마십시오. 마이그레이션이 스키마를 바꿨다면 DB도 함께 되돌려야 합니다.

5.5 정기 점검

주기	볼 것
매일	/healthz · /readyz , 로그인 실패, 승인 대기, 고위험 시크릿, 실패한 Webhook 전송
매주	관리자·감사자 권한, 미사용 세션과 API 토큰, 수동 회전 기록, OIDC 연결
매월	복구 시험, 관리자 계정 검토, 감사 로그 증가량, 인증서 만료
릴리스 전	make release-check , 버전 표시, 모바일·새로 고침 동작, 전체 화면 캡처 계약

감사 로그는 자동으로 지워지지 않습니다(3.2의 프리뷰 항목). 증가량을 매월 확인하고 필요하면 데이터베이스 쪽 보존 정책으로 처리하십시오.

6. 장애 대응

증상에서 시작해 확인할 곳과 조치를 적었습니다. 로그 문구는 실제로 찍히는 것을 그대로 실었습니다.

6.1 컨테이너가 뜨지 않는다

로그에 찍히는 문구	원인	조치
필수 환경변수가 없습니다: POSTGRES_DSN, ...	네 값 중 일부가 비었습니다.	Compose 환경변수를 채웁니다.
BOOTSTRAP_ADMIN은 3~128자여야 합니다	아이디 길이 위반	값을 고칩니다.
BOOTSTRAP_ADMIN_PASSWORD는 12자 이상이어야 합니다	비밀번호 길이 위반	값을 고칩니다.
jikim 서버 종료 + error 필드	기동 단계 실패(키 형식, DB 연결, 마이그레이션, 마스터 키 검증)	error 값을 읽고 아래 항목으로 넘어갑니다.

error 에 마스터 키 검증 실패가 보이면 **이전과 다른** ENCRYPTION_KEY 로 기존 데이터베이스에 붙은 것입니다. 키를 원래 값으로 되돌리십시오. 새 키로 계속 올리면 기존 암호문을 읽을 수 없습니다.

6.2 화면은 뜨는데 로그인이 안 된다

증상	확인할 곳	조치
/readyz 가 503 + 데이터베이스 연결을 확인할 수 없습니다	PostgreSQL 도달성, DSN, TLS 모드, 계정 권한	저장소를 복구합니다. v0.2.11 은 DB 장애를 자격증명 거부로 접지 않고 500 으로 구분해 보고합니다.
아이디 또는 비밀번호가 올바르지 않습니다	사용자 관리 화면의 계정 상태	계정·비밀번호를 확인합니다. DB 장애일 때는 이 문구가 아니라 500 이 나옵니다.
사용자 계정이 비활성화되었습니다	사용자 관리 화면	계정을 활성화합니다.
로컬 로그인이 비활성화되었습니다	보안 탭의 로컬 로그인 허용	OIDC로 들어가거나, 그것도 막혔으면 설정 값을 되돌려야 합니다.
로그인 시도가 너무 많습니다. 잠시 후 다시 시도하세요	로그인 실패 제한	Retry-After 이후 재시도합니다. 계정+IP 조합 기준입니다.
SSO만 실패	POST /api/v1/oidc/test (관리 화면의 Discovery 연결 테스트), Keycloak Valid redirect URIs	Callback URL이 https://<Origin>/api/v1/oidc/callback 로 등록됐는지 확인합니다.
OIDC callback Redirect URL이 설정되지 않았거나 올바르지 않습니다	OIDC 탭의 Callback URL	절대 URL로 다시 저장합니다.
OIDC 상태가 만료되었거나 올바르지 않습니다	Reverse Proxy가 쿠키·Host·X-Forwarded-Proto 를 전달하는지	프록시 설정을 고칩니다.

6.3 시크릿 값이 열리지 않는다

증상	원인	조치
Secret 값 조회 권한이 없습니다	정책에 read 가 없습니다.	정책 시뮬레이터로 확인한 뒤 정책을 조정합니다.
감사 로그를 저장할 수 없어 Secret 값을 표시하지 않습니다	감사 기록에 실패했습니다. 기록 없이 평문을 내주지 않는 설계입니다.	저장소 상태를 확인합니다. 재시도로 해결되지 않습니다.
failed to check permissions (/v1/*) 또는 500	정책 조회 자체가 실패했습니다. 거부 아닙니다.	저장소 상태를 확인합니다. 정책 오설정을 쫓지 마십시오.
로그에 감사 로그 저장 실패 + request_id	감사 write 실패	저장소 용량·연결을 확인합니다.

6.4 연동이 실패한다

증상	확인할 곳	조치
AI API에 연결할 수 없습니다	AI Gateway 도달성, 내부 CA	egress 허용 대상과 /app/certs/internal-ca.crt 를 확인합니다.
AI API가 SSE 스트림을 반환하지 않았습니다	Upstream이 text/event-stream 을 주지 않습니다.	Upstream 설정이나 중간 프록시의 버퍼링을 확인합니다.
저장된 AI API 설정이 보안 정책에 맞지 않습니다	Base URL이 HTTP인데 내부 HTTP 허용이 꺼져 있습니다.	HTTPS로 바꾸거나 내부 HTTP 허용을 명시적으로 켭니다.
로그에 내부 CA 파일을 읽었지만 유효한 PEM 인증서가 없습니다	mount한 파일이 PEM이 아닙니다.	인증서 파일을 다시 넣습니다.
로그에 내부 CA 파일을 읽지 못했습니다	mount 경로·권한 문제	bind mount 대상이 /app/certs/internal-ca.crt 인지 확인합니다.
Webhook 이력이 failed	이력 API의 응답 상태·최근 오류	원인을 해소하고 retry API로 수동 재전송합니다. 자동 재시도는 없습니다.
로그에 webhook 전송 실패 + delivery_id	수신기 장애·타임아웃(10초)	수신기를 확인한 뒤 재전송합니다.
로그에 웹 UI 파일을 찾지 못했습니다	이미지가 손상됐거나 잘못된 이미지입니다.	번들 체크섬을 다시 확인하고 재적재합니다.

6.5 오류 하나를 추적한다

모든 응답에는 X-Request-ID 헤더가 붙습니다(클라이언트가 보낸 값이 있으면 그대로 반향합니다). 사용자가 요청 ID 나 감사 화면의 이벤트 ID를 가져오면,

1. 감사·관측 → 감사 로그에서 그 값으로 검색합니다.
2. 같은 request_id 로 컨테이너 로그를 찾습니다. 서버 오류는 {"level":"ERROR","msg":"request failed","request_id":"...","path":"..."} 로 남습니다.
3. 패닉이면 {"level":"ERROR","msg":"panic","request_id":"...","stack":"..."} 가 함께 있습니다.

7. 보안

7.1 배포 직후 바뀌어야 하는 것

항목	기본값	해야 할 일
BOOTSTRAP_ADMIN 계정	기동 시 생성	개인 관리자 계정을 만든 뒤 비밀번호를 회전하고 사용 범위를 제한합니다. 화면이 강제하지 않으므로 사람이 해야 합니다.
최소 비밀번호 길이	12	조직 정책에 맞게 올립니다(12~128).
세션 제한 시간	관리 화면 값	조직 정책에 맞게 조정합니다(5~1440분).
TLS	없음(HTTP 8080)	Reverse Proxy로 HTTPS를 종단하고 8080 을 직접 노출하지 않습니다.
PostgreSQL TLS	DSN에 따름	sslmode=verify-full 을 씁니다.
내부 HTTP 허용 (OIDC·AI·Webhook·방문 추적 수집기)	모두 OFF	켜지 않습니다. TLS를 적용할 수 없는 신뢰된 내부 개발망에서만 예외로 씁니다.
승인 워크플로	비활성	조직이 검토 절차를 요구하면 켜고 두 계정으로 시험합니다.
AI 연동	비활성	필요할 때만 켜고 승인된 내부 Gateway 주소만 씁니다.
방문 추적	꺼짐	컨다면 Memento를 같은 오리진 프록시로 붙입니다. 외부 제공자는 데이터가 밖으로 나가므로 폐쇄망에서는 쓰지 않습니다.

7.2 외부에 열지 않을 것

- 컨테이너 포트 8080 을 인터넷이나 사무망에 직접 노출하지 않습니다. 앞에 TLS Reverse Proxy를 둡니다.
- PostgreSQL 포트를 애플리케이션 네트워크 밖으로 열지 않습니다.
- 컨테이너 egress를 PostgreSQL과 승인된 Keycloak·AI·Webhook 주소로 제한합니다. 릴리스 검증에 egress 차단 스모크 테스트가 들어 있는 이유가 이것입니다.
- POST /mcp 는 Origin 가드를 통과한 인증 요청만 받습니다. MCP 클라이언트를 붙일 때 [보안 운영 가이드](#)의 MCP 절을 함께 보십시오.

7.3 흘리지 말 것

- 시크릿 평문, 세션 토큰, API 토큰, `POSTGRES_DSN`, Client Secret, AI API Key를 로그·이슈·AI 프롬프트·화면 캡처에 넣지 않습니다.
- `ENCRYPTION_KEY` 를 잃으면 저장된 암호문을 복구할 수 없습니다. 재암호화 절차 없이 바꾸면 기존 시크릿을 읽을 수 없습니다.
- 관리자·복호화·키 회전 권한은 최소 인원에게만 줍니다. `admin` 과 `manager` 는 경로 정책을 통과하므로, 권한을 좁혀야 하는 계정에는 `user` 역할과 정책을 씁니다.
- 이미지 체크섬, digest와 배포 버전을 기록해 두십시오.

취약점은 공개 Issue가 아니라 저장소 `SECURITY.md` 의 비공개 절차로 신고합니다.