Data Works

Data Works 관리자 가이드

버전 v0.9.52

작성일 2026-09-12

문서 Data Works 관리자 가이드

Data Works 관리자 가이드

> 적용 버전: **v0.9.52**
> 서비스 관리자 화면: `http://<host>:8080/dataworks/settings`
> 일반 사용 방법은 사용자 가이드를 참고하세요.

이 문서는 폐쇄망 설치, 최초 관리자 로그인, AI·MCP, Keycloak SSO, 키 정책과 데이터 상품 운영 절차를 실제 v0.9.52 구현 기준으로 설명합니다.

> 이 문서의 모든 화면은 v0.9.52을 실제로 띄워 가상 데이터로 촬영했습니다. 실제 운영 키와 고객 정보가 아닙니다.

1. 구성 요소

Data Works 배포 이미지는 React SPA와 Go API를 하나의 바이너리로 포함합니다. 운영 저장소는 PostgreSQL을 사용하며, 브라우저는 `/dataworks/`, API는 같은 호스트의 `/admin/*`, `/v1/*`, `/mcp*` 경로를 사용합니다.

컨테이너와 의존 서비스

구성 요소	필수	무엇을 주고받나
`dataworks` 컨테이너	필수	SPA 제공, 관리 API, 상품 런타임 API, MCP 엔드포인트. distroless 이미지에서 `nonroot` 로 실행됩니다.
PostgreSQL	운영 필수	사용자·역할·설정·자산·상품·승인·계약·엔타이틀먼트·감사 이력 전부. `POSTGRES_DSN` 으로 연결합니다.
컨테이너 볼륨 `/data`	필수	로그 적재 실패 시 예비 파일(`fallback.ndjson`). SQLite 모드에서는 `gateway.db` 도 여기에 놓입니다.
OpenAI 호환 AI 서버	선택	코파일럿·상품 공장·MCP 에이전트가 호출합니다. 폐쇄망에서는 내부 vLLM/Qwen 주소를 등록합니다.
Keycloak	선택	OIDC 로그인. 켜지 않으면 로컬 이메일·비밀번호 로그인만 사용합니다.
ClickHouse	선택	분석 팩트 테이블 내보내기. `CLICKHOUSE_URL` 이 비어 있으면 전송하지 않습니다.

포트·볼륨·자원

항목	값	비고
컨테이너 포트	`8080`	이미지의 `LISTEN_ADDR=:8080` . 외부에 열 포트는 이것 하나입니다.
볼륨	`/data`	compose 기본값은 호스트의 `./data` .
PostgreSQL 포트	`5432`	컨테이너에서 접근 가능해야 합니다. 외부에 공개하지 마세요.
최소 자원	vCPU 2 · 메모리 2 GiB · 디스크 10 GiB	실제 사용량은 로그 보존 기간과 상품 수에 좌우됩니다.

서비스 접속 경로

주소	용도와 동작
`/dataworks/`	새 React 기반 사용자·관리자 Workbench의 **표준 진입점**
`/dataworks`	`/dataworks/` 로 `308 Permanent Redirect`
`/admin`	기존(레거시) 관리자 콘솔
`/`	UI 라우트가 아니며 기본 구성에서는 `404 Not Found`

운영 안내에는 `http://<host>:8080/dataworks/` 를 서비스 주소로 사용하세요. 대표 도메인의 루트(`/`)를 진입점으로 사용할 경우 리버스 프록시에서 `/` 를 `/dataworks/` 로 리다이렉트합니다. 프록시는 SPA와 로그인·API 기능이 함께 동작하도록 `/dataworks/*`, `/auth/*`, `/admin/*`, `/v1/*`, `/mcp*`, `/openapi.json`, `/swagger` 를 동일한 Data Works 서비스로 전달해야 합니다.

GitHub Release의 운영 산출물은 다음 하나의 custom asset입니다.

```
dataworks-v0.9.52.tar.gz
```

압축 파일을 적재하면 다음 이미지가 생성됩니다.

```
dataworks:v0.9.52
```

2. 폐쇄망 설치

필수 환경변수 네 개

이름	설명
`POSTGRES_DSN`	운영 PostgreSQL 연결 문자열
`BOOTSTRAP_ADMIN`	최초 최고 관리자 이메일
`BOOTSTRAP_ADMIN_PASSWORD`	최초 최고 관리자 비밀번호
`ENCRYPTION_KEY`	JWT 서명과 저장 비밀 암호화에 사용할 키

운영자가 컨테이너에 전달해야 하는 설정은 이 네 항목입니다. AI 공급자, Keycloak과 런타임 정책은 기동 후 관리자 화면에서 저장합니다. 나머지 환경 변수 전체는 4.6 환경 변수 전수 표에 있습니다.

`ENCRYPTION_KEY` 는 32바이트 난수를 64자리 16진수로 표현하는 방식을 권장합니다.

```
openssl rand -hex 32
```

키는 백업 가능한 비밀 저장소에 보관하고 운영 중 임의로 변경하지 마세요. 변경하면 기존 암호화 비밀을 해독할 수 없고 로그인 세션에도 영향을 줍니다.

이미지 적재

```
gzip -t dataworks-v0.9.52.tar.gz
gunzip -c dataworks-v0.9.52.tar.gz | docker load
docker image inspect dataworks:v0.9.52
```

컨테이너 실행

PostgreSQL은 폐쇄망 안에서 컨테이너가 접근할 수 있어야 합니다.

```
docker run -d --name dataworks --restart=always \
  -p 8080:8080 \
  -v "$PWD/data:/data" \
  -e POSTGRES_DSN='postgres://dataworks:change-me@postgres.internal:5432/dataworks?sslmode=require' \
  -e BOOTSTRAP_ADMIN='admin@dataworks.local' \
  -e BOOTSTRAP_ADMIN_PASSWORD='replace-with-a-strong-password' \
  -e ENCRYPTION_KEY='replace-with-64-hex-characters' \
  dataworks:v0.9.52
```

저장소의 `docker-compose.yml` 을 함께 반입한 환경에서는 같은 네 값을 `.env` 에 저장한 뒤 실행할 수 있습니다.

```
POSTGRES_DSN=postgres://dataworks:change-me@postgres.internal:5432/dataworks?sslmode=require
BOOTSTRAP_ADMIN=admin@dataworks.local
BOOTSTRAP_ADMIN_PASSWORD=replace-with-a-strong-password
ENCRYPTION_KEY=0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef
```

```
docker compose config --images
# dataworks:v0.9.52
docker compose up -d
```

기동 확인

```
curl -fsS http://127.0.0.1:8080/health
curl -fsS http://127.0.0.1:8080/ready
docker logs --tail=100 dataworks
```

정상 응답:

```
{"status":"ok"}
{"status":"ready"}
```

로그에 다음 줄이 보이면 기동이 끝난 것입니다.

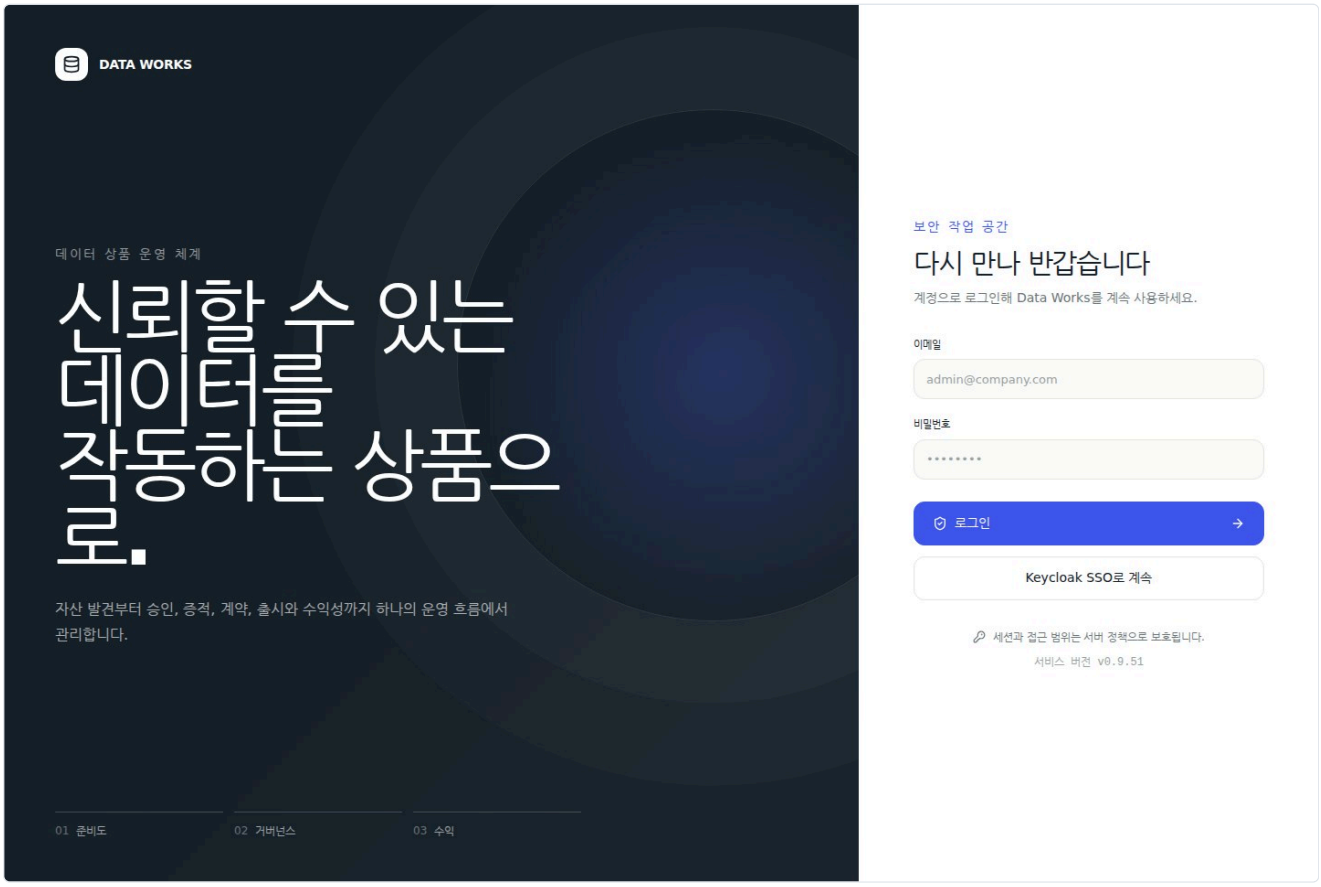
```
level=INFO msg="Data Works listening" addr=:8080 database=postgres
```

PostgreSQL 마이그레이션은 기동 시 자동 실행됩니다. 외부 공개 환경에서는 TLS 종료와 접근 제어가 적용된 리버스 프록시 뒤에 배치하세요.

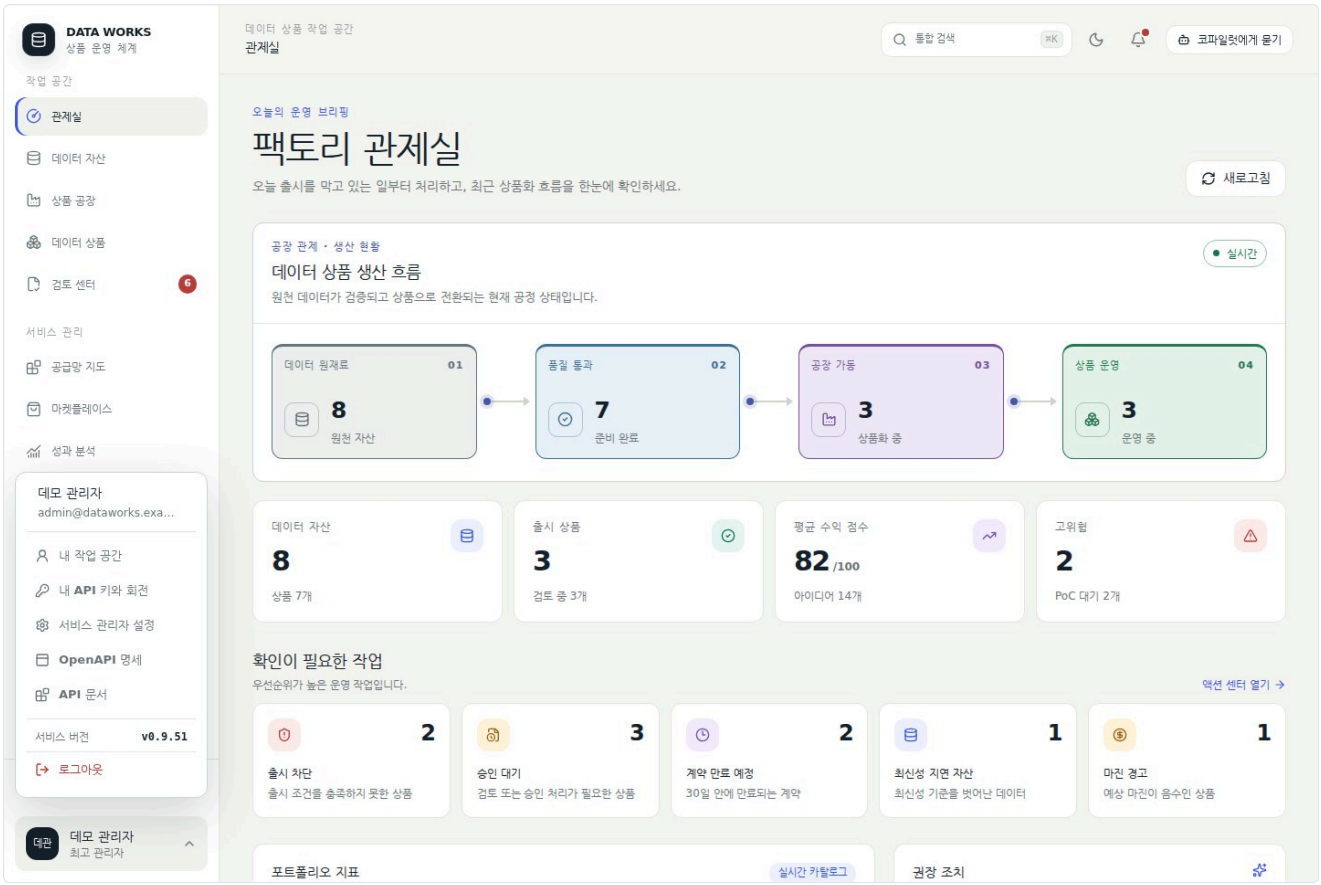
3. 최초 관리자 로그인

`BOOTSTRAP_ADMIN` 계정이 DB에 없으면 최초 기동 시 `super_admin` 으로 생성됩니다. 이미 같은 이메일이 있으면 재생성하거나 비밀번호를 덮어쓰지 않습니다.

1. `http://<host>:8080/dataworks/` 를 엽니다.
2. Bootstrap 이메일과 비밀번호로 로그인합니다.
3. 로그인 화면 또는 프로필 메뉴에서 `v0.9.52` 를 확인합니다.
4. 왼쪽 아래 `관리자 설정` 을 엽니다.



로그인 화면 — 로컬 로그인과 Keycloak SSO 버튼, 서비스 버전이 함께 보인다



프로필 메뉴 — 로그인 후 현재 서비스 버전과 관리자 설정 진입점을 확인한다

4. 설정

관리자 영역과 개인화 영역은 별도 메뉴와 권한으로 분리됩니다.

- **관리자 설정** `/dataworks/settings` : AI 공급자, AI·MCP 정책, Keycloak과 전체 런타임 설정
- **내 작업 공간** `/dataworks/personal` : 현재 사용자의 사용량·비용·품질과 개인 신호
- **내 API 키** `/dataworks/personal/keys` : 현재 사용자가 소유한 개인 키

설정 쓰기는 서버가 역할과 설정 카테고리별 권한을 다시 검사합니다.

4.1 AI 공급자

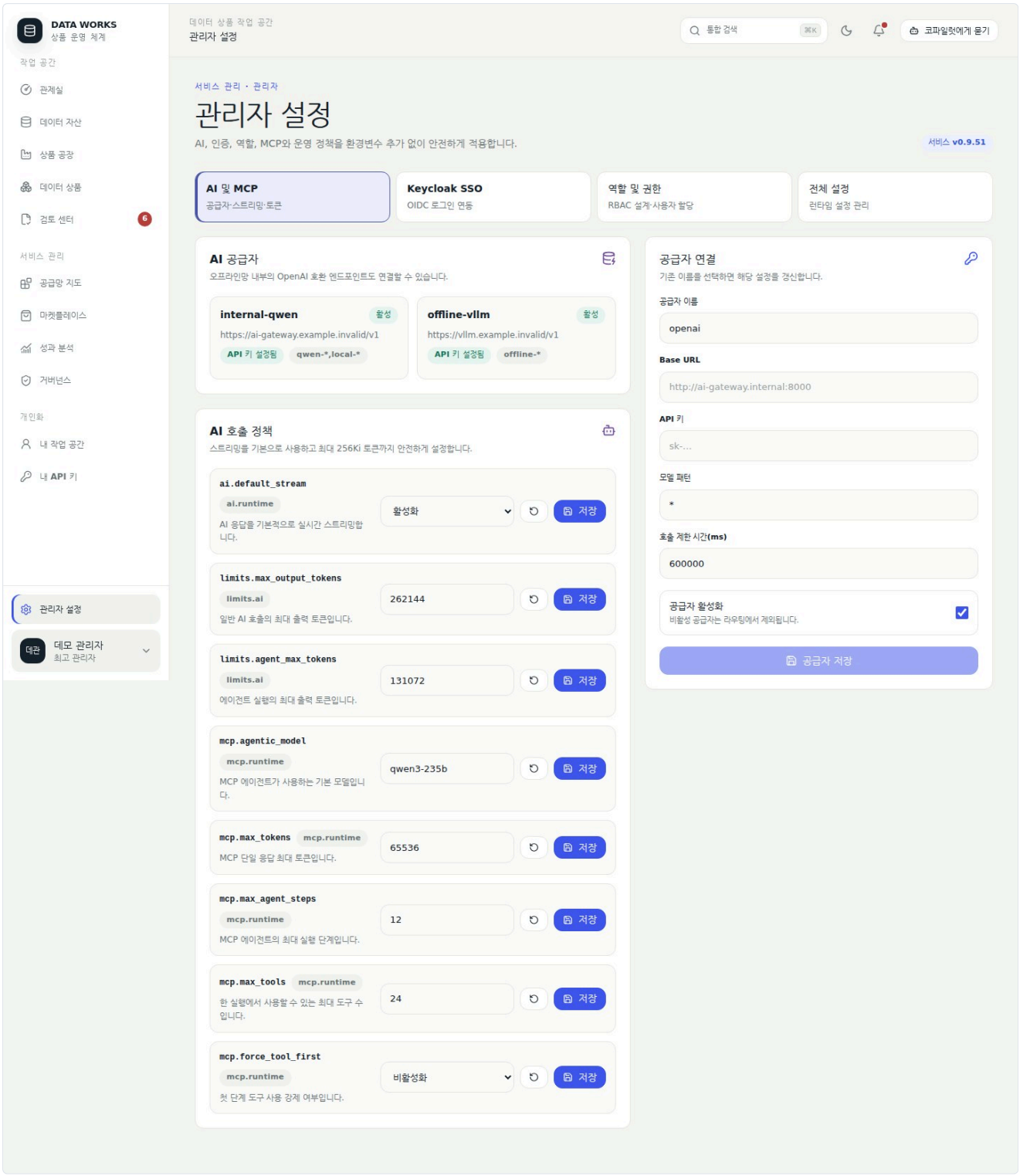
`관리자 설정 → AI 및 MCP` 에서 공급자를 등록합니다.

필수 또는 주요 필드:

- 공급자 이름
- OpenAI 호환 Base URL
- API 키
- 모델 패턴: 예 `qwen-*,local-*`
- 호출 제한 시간
- 활성 상태

API 키는 암호화되어 DB에 저장되며 다시 평문으로 표시되지 않습니다. 기존 공급자를 수정할 때 API 키를 비우면 저장된 키를 유지합니다.

폐쇄망에서는 내부 vLLM, Qwen 또는 다른 OpenAI 호환 서버 주소를 Base URL로 지정합니다. 저장 후 실제 `/v1/models` 와 테스트 호출로 모델 라우팅을 검증하세요.



AI 및 MCP 설정 — 공급자 Base URL·모델 패턴과 스트리밍·토큰 정책을 한 화면에서 관리한다

4.2 AI 스트리밍과 토큰 정책

같은 화면에서 다음 설정을 관리합니다.

설정	의미
`ai.default_stream`	Chat Completions에서 `stream` 생략 시 기본값, 기본 `true`
`limits.max_output_tokens`	일반 응답 출력 상한, `0` 은 관리 상한 비활성
`limits.agent_max_tokens`	내부 Agent 출력 상한
`mcp.agentic_model`	Agentic MCP에 사용할 모델
`mcp.max_tokens`	MCP 각 LLM 턴의 출력 토큰
`mcp.max_agent_steps`	Agentic MCP 최대 LLM 턴
`mcp.max_tools`	모델에 노출할 최대 MCP 도구 수
`mcp.force_tool_first`	첫 턴 도구 호출 강제 여부

토큰 관련 서비스 절대 상한은 `262144` 입니다.

- Chat Completions: `max_tokens` , `max_completion_tokens`
- Responses API: `max_output_tokens`
- 관리 상한이 양수이면 요청 값은 상한 이하로 조정됩니다.
- 호출자가 `stream:false` 를 명시하면 기본 스트리밍 설정이 이를 덮어쓰지 않습니다.
- 연결 모델 자체 한도가 더 낮으면 모델 제한이 우선합니다.

4.3 Keycloak SSO/OIDC

`관리자 설정 → Keycloak SSO` 에서 Issuer URL, Client ID와 Client Secret만 입력하면 OIDC Discovery와 PKCE 로그인 흐름을 자동 구성합니다.

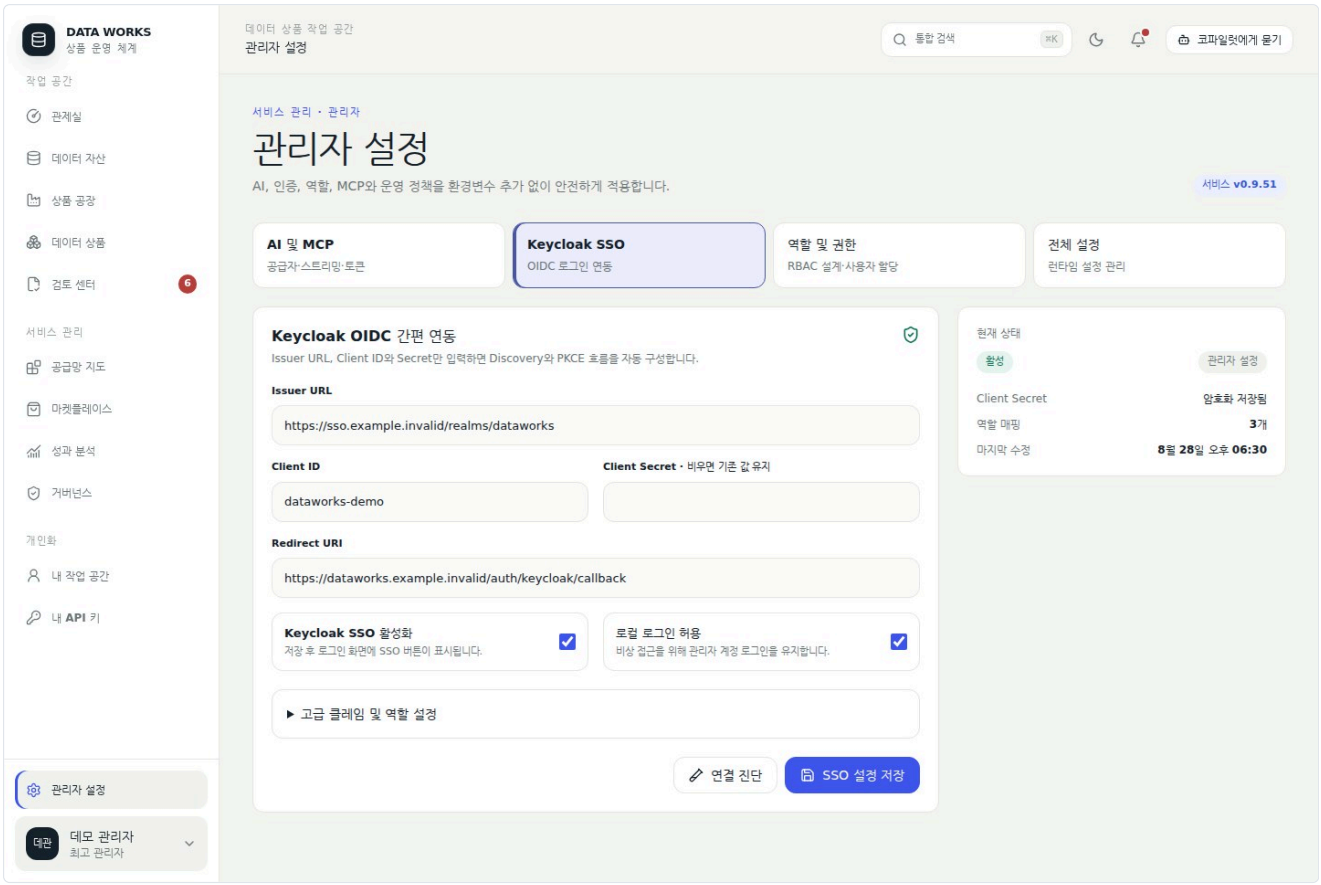
Keycloak에 confidential OIDC client를 만들고 다음 Redirect URI를 허용합니다.

```
https://<service-host>/auth/keycloak/callback
```

HTTP 내부망 시험 환경에서는 실제 서비스 origin의 HTTP URI를 사용할 수 있지만 운영 환경은 HTTPS를 권장합니다.

1. Issuer URL을 입력합니다. 예: `https://keycloak.internal/realms/dataworks`
2. Client ID와 Client Secret을 입력합니다.
3. Redirect URI가 서비스 공개 주소와 일치하는지 확인합니다.
4. `Keycloak SSO 활성화` 를 켭니다.
5. 비상 접근 정책에 따라 `로컬 로그인 허용` 을 선택합니다.
6. `SSO 설정 저장` 을 누릅니다.
7. `연결 진단` 에서 Issuer와 RSA 서명 키 발견을 확인합니다.
8. 로그아웃 후 로그인 화면의 `Keycloak SSO로 계속` 을 시험합니다.

Client Secret은 암호화 저장됩니다. 수정 화면에서 Secret을 비우면 기존 값을 유지합니다.



Keycloak SSO 설정 — Issuer·Client 정보와 역할 매핑, 연결 진단을 한 화면에서 처리한다

고급 역할 매핑 항목:

- OIDC Scope 기본값: `openid profile email`
- 기본 내부 역할
- 역할 Claim, 기본 `realm_access.roles`
- 그룹 Claim, 기본 `groups`
- Keycloak 역할→Data Works 역할 JSON

역할 매핑은 권한 상승 방지 검사를 적용합니다. 일반 관리자는 자신과 동급 이상의 새 관리자 또는 `super_admin` 매핑을 임의로 만들 수 없습니다. SSO를 켜기 전에 Bootstrap 로컬 계정으로 복구할 수 있는지 확인하고, 로컬 로그인을 끄기 전 별도 관리자 SSO 계정을 시험하세요.

관리 API:

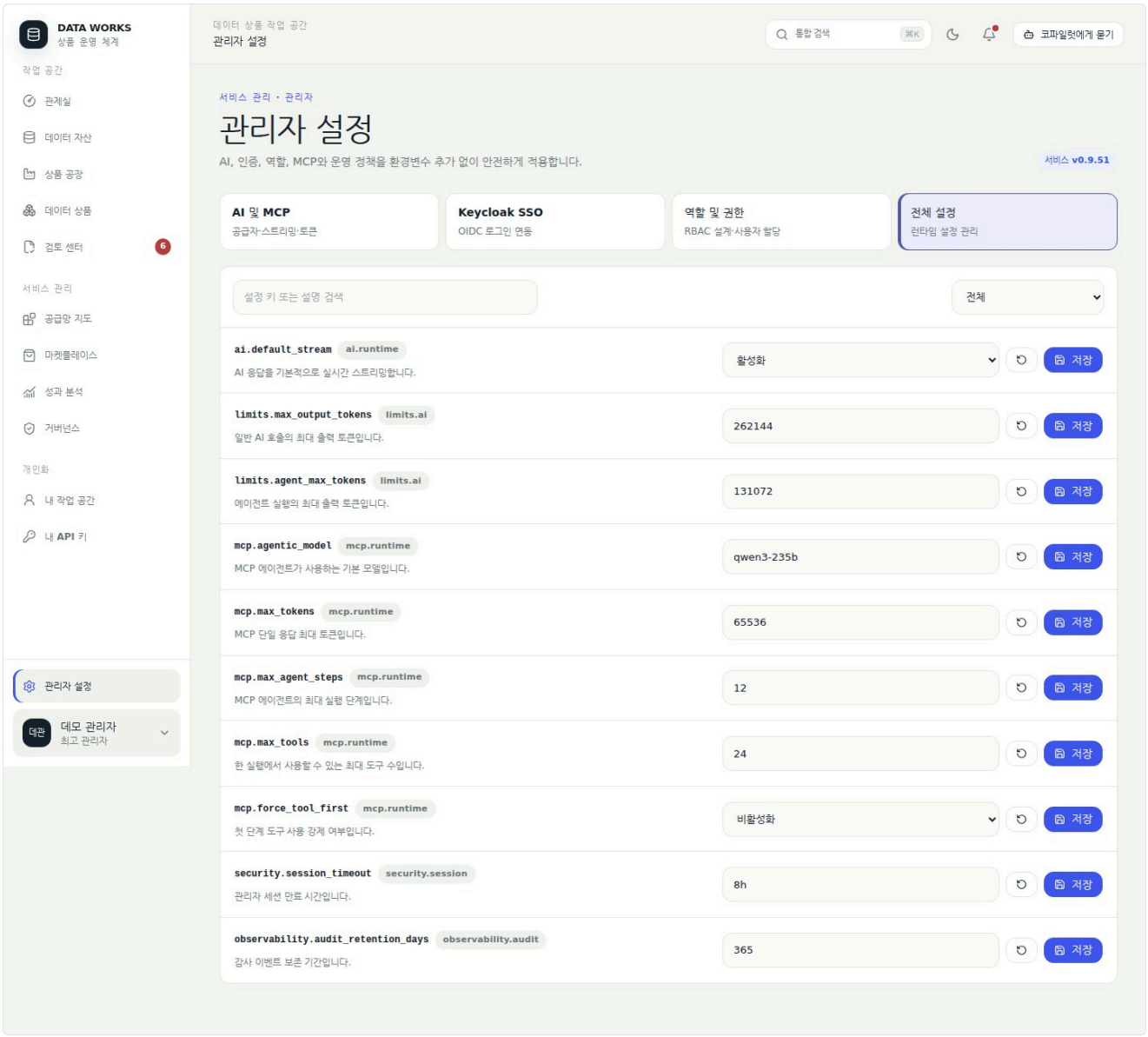
```
GET/PUT /admin/sso/keycloak/config
POST    /admin/sso/keycloak/test
GET     /auth/sso/status
```

4.4 전체 런타임 설정

`관리자 설정 → 전체 설정`에서 설정 키와 설명을 검색하고 카테고리로 필터링합니다.

- 현재 값과 값 출처를 확인합니다.
- `저장`으로 DB 기반 관리자 override를 적용합니다.
- 되돌리기 버튼으로 관리자 override를 삭제하고 기본값을 복원합니다.
- 비밀 설정은 암호화하고 응답에서 마스킹합니다.
- 읽기 전용, 재시작 필요와 역할별 쓰기 가능 여부가 표시됩니다.

```
GET    /admin/settings/effective
PUT    /admin/settings/by-key/{key}
DELETE /admin/settings/by-key/{key}
```



전체 런타임 설정 — 설정 키를 검색해 현재 값과 출처를 확인하고 override를 저장·복원한다

DB에 저장된 관리자 override는 `SETTINGS_RELOAD_INTERVAL` (기본 `10s`)마다 다시 읽습니다.

4.5 방문 추적 스크립트

`관리자 설정 → 방문 추적` 에서 어떤 화면이 실제로 쓰이는지 재는 추적 도구를 붙입니다. 기본값은 **꺼짐**이며, 새로 설치한 곳은 켜기 전까지 아무것도 달라지지 않습니다. 설정은 다른 런타임 설정과 같이 DB에 저장되고 화면에서 바꿉니다(환경변수 없음, 재배포 불필요).

설정	뜻	기본값
`tracking.enabled`	스니펫 삽입 여부. 나머지를 저장한 뒤 **마지막에** 켭니다	`false`
`tracking.provider`	`none` · `momento` · `ga4` · `gtm` · `matomo` · `custom`	`none`
`tracking.momento_url` · `tracking.momento_site_id`	Momento 수집기 주소와 사이트 id	없음
`tracking.momento_proxy`	같은 오리진 프록시(`/momento/*`) 사용	`true`
`tracking.momento_environment`	Momento `data-environment` 값	`prd`
`tracking.measurement_id`	GA4(`G-…`) 또는 GTM(`GTM-…`) id	없음
`tracking.matomo_url` · `tracking.matomo_site_id`	Matomo 주소와 사이트 id	없음
`tracking.custom_snippet`	붙여넣는 스니펫. 8KB 를 넘으면 저장되지 않습니다	없음
`tracking.allowed_hosts`	스니펫에서 자동으로 읽지 못한 출처를 더하는 자리(쉼표 구분)	없음
`tracking.include_admin`	관리 화면(`/admin` , `/dataworks/settings`)에서도 추적할지	`false`
`tracking.placement`	`head` 또는 `body`	`head`

Momento 를 먼저 씁니다. 사내 자체 호스팅 수집기라 데이터가 밖으로 나가지 않는 유일한 선택지입니다. 순서는 다음과 같습니다.

1. `tracking.provider = momento` , `tracking.momento_url = https://momento.internal` , `tracking.momento_site_id` = 발급받은 id 를 저장합니다.
2. `tracking.enabled = true` 로 켭니다. 오른쪽 "현재 상태"가 `추적 중` 이면 끝입니다. `설정 미완료` 면 빠진 값이 그 아래에 적혀 있습니다.
3. 브라우저에서 워크벤치를 한 번 열고 Momento 쪽에 페이지뷰가 들어오는지 확인합니다.

같은 오리진 프록시가 기본입니다. 서버가 `/momento/*` 를 수집기로 넘기고 스니펫에 `data-endpoint="/momento"` 를 주므로 tracker.js 와 수집 호출이 모두 이 서비스 주소로만 나갑니다. 프록시는 추적이 켜져 있고 provider 가 `momento` 일 때만 열리며, 방문자의 세션 쿠키와 Authorization 헤더는 수집기로 전달하지 않습니다. 브라우저가 수집기에 직접 닿아야 하는 설치라면 `tracking.momento_proxy = false` 로 두면 수집기 출처가 정책에 자동으로 추가됩니다.

CSP 와 nonce

워크벤치(`/dataworks/*`)는 `script-src 'self'` 로 잠긴 콘텐츠 보안 정책(CSP)을 답니다. 스니펫을 그냥 붙이면 브라우저가 조용히 막고 화면은 비어 있는 이유를 알려 주지 않습니다. 그래서 서버는 다음을 자동으로 합니다.

- 요청마다 nonce 를 새로 만들어 스니펫의 **모든** `<script>` 태그에 붙이고, 같은 값을 `script-src 'nonce-…'` 로 정책에 넣습니다. 정책을 `'unsafe-inline'` 으로 풀지 않습니다 — 한 번 풀면 추적을 끈 뒤에도 느슨한 채 남기 때문입니다.
- provider 가 필요로 하는 출처(GA4·GTM 의 googletagmanager, Matomo 주소, 직접 연결 모드의 Momento 주소)와 `custom` 스니펫 안에 적힌 `http(s)` 출처를 읽어 `script-src` · `connect-src` · `img-src` 에 더합니다.
- 추적이 켜져 있는 동안만 `report-uri /tracking/csp-report` 를 정책에 넣어 브라우저가 막은 요청을 신고하게 하고, 그 출처와 지시어를 메모리에 최대 100개까지 기억합니다.
- 추적을 끄면 정책은 원래대로 좁아지고(nonce 도 report-uri 도 사라짐) 프록시는 닫힙니다.

`관리자 설정 → 방문 추적 → 차단된 출처` 에 "이 출처가 막혔습니다" 가 뜨면 `허용` 을 눌러 `tracking.allowed_hosts` 에 넣습니다. 이미 허용된 출처는 `허용됨` 으로 표시됩니다. 스니펫을 고친 뒤 휴지통 버튼으로 기록을 비우면 아직 막히는 것이 남았는지 다시 볼 수 있습니다.

```
GET    /admin/tracking/status             # 유효 설정, 스니펫, 정책 출처, 미완료 사유
GET    /admin/tracking/violations         # 차단된 출처 목록
DELETE /admin/tracking/violations         # 기록 비우기
POST   /admin/tracking/violations/allow   # {"origin":"https://…"} → tracking.allowed_hosts 에 추가
POST   /tracking/csp-report               # 브라우저 신고 수신(무인증, 추적이 꺼져 있으면 버림)
```

붙지 않는 곳:

- `/v1/*` · `/admin/*` API · `/healthz` · `/metrics` 같은 비화면 경로. 이쪽은 `default-src 'none'` 으로 오히려 더 좁습니다.
- 관리 화면은 `tracking.include_admin` 이 켜졌을 때만. 워크벤치는 한 문서(SPA)라 브라우저가 처음 연 주소 기준으로 판단합니다.
- 레거시 콘솔(`/admin`)은 인라인 핸들러로 만들어져 nonce 정책을 달 수 없으므로 CSP 없이 그대로이며, 스니펫은 `include_admin` 일 때만 들어갑니다.
- 로그인 화면에 붙더라도 스니펫에 개인 식별 값을 넣지 마세요.

4.6 환경 변수 전수 표

아래는 서비스 바이너리(`cmd/dataworks`)가 읽는 환경 변수 전부입니다. **필수** 표시가 없는 값은 설정하지 않아도 기본값으로 동작합니다. 비밀값 예시는 모두 가짜 값입니다.

기동과 데이터베이스

이름	기본값	필수	설명
`LISTEN_ADDR`	`:9090` (배포 이미지는 `:8080`)		HTTP 수신 주소. 숫자만 넣으면 `:` 을 앞에 붙입니다.
`POSTGRES_DSN`	(없음)	**필수**	운영 PostgreSQL DSN. `postgres://` 로 시작하면 드라이버를 자동으로 PostgreSQL로 정합니다.
`DATABASE_URL`	(없음)		`POSTGRES_DSN` 의 대체 이름. 같은 자동 감지 규칙을 적용합니다.
`DB_DRIVER`	`sqlite`		`postgres` 또는 `postgresql` 이면 PostgreSQL을 씁니다. DSN 자동 감지가 이 값보다 우선합니다.
`DB_DSN`	`data/gateway.db`		드라이버별 DSN. SQLite 모드의 파일 경로이기도 합니다. 배포 이미지는 `/data/gateway.db`.
`SETTINGS_RELOAD_INTERVAL`	`10s`		DB에 저장된 런타임 설정을 다시 읽는 주기.

인증과 세션

이름	기본값	필수	설명
`BOOTSTRAP_ADMIN`	(없음)	**필수**	최초 `super_admin` 이메일. 같은 이메일이 이미 있으면 아무것도 덮어쓰지 않습니다.
`BOOTSTRAP_ADMIN_PASSWORD`	(없음)	**필수**	최초 `super_admin` 비밀번호.
`AUTH_ADMIN_BOOTSTRAP_EMAIL`	(없음)		`BOOTSTRAP_ADMIN` 의 예전 이름. 짧은 이름이 없을 때만 쓰입니다.
`AUTH_ADMIN_BOOTSTRAP_PASSWORD`	(없음)		`BOOTSTRAP_ADMIN_PASSWORD` 의 예전 이름.
`ENCRYPTION_KEY`	(없음)	**필수**	저장 비밀 암호화 키이자 JWT 서명 기본 키. 예: `0123…cdef` (64자리 16진수).
`AUTH_ENABLED`	Bootstrap 이메일·비밀번호가 모두 있으면 `true`		계정 로그인 사용 여부. 끄면 레거시 토큰 모드가 됩니다.
`AUTH_JWT_SECRET`	`ENCRYPTION_KEY` 값		JWT 서명 키를 암호화 키와 분리하고 싶을 때만 지정합니다.
`AUTH_ACCESS_TOKEN_TTL`	`15m`		액세스 토큰 수명.
`AUTH_REFRESH_TOKEN_TTL`	`168h`		리프레시 토큰 수명(7일).
`AUTH_API_KEY_PREFIX`	`vc_sk_`		개인 API 키 접두사.
`AUTH_SERVICE_KEY_PREFIX`	`vc_sa_`		서비스 계정 키 접두사.
`SELF_SERVICE_KEYS_ENABLED`	`AUTH_ENABLED` 와 같음		사용자가 직접 개인 키를 발급할 수 있는지 여부.
`SESSION_INFERENCE_ENABLED`	`true`		요청을 세션 단위로 묶어 추론할지 여부.
`SESSION_IDLE_TIMEOUT`	`30m`		세션 묶음을 끊는 유휴 시간.
`GATEWAY_SECRET`	`ENCRYPTION_KEY` → 내장 기본값		게이트웨이 내부 서명용 비밀. 운영에서는 `ENCRYPTION_KEY` 를 그대로 씁니다.
`ADMIN_TOKEN`	(없음)		레거시 관리자 토큰. 계정 로그인을 쓰는 배포에서는 **비워 두세요**.
`ADMIN_READONLY_TOKEN`	(없음)		레거시 읽기 전용 관리자 토큰. 같은 이유로 비워 두는 것을 권장합니다.
`PROXY_API_KEYS`	(없음)		레거시 정적 프록시 키 목록(쉼표 구분). 개인 키 발급으로 대체되었습니다.
`ATTRIBUTE_EXTERNAL_KEYS`	`true`		외부에서 들어온 키 사용량을 사용자에게 귀속시킬지 여부.

`AUTH_ENABLED=true` 인데 `ENCRYPTION_KEY` 와 `AUTH_JWT_SECRET` 이 모두 비어 있으면 기동이 `AUTH_JWT_SECRET is required when AUTH_ENABLED=true` 로 실패합니다.
`LOG_QUEUE_SIZE` 는 양수여야 합니다.

Keycloak SSO (환경 변수로 초기값 주입)

관리자 화면에서 저장한 값이 우선하며, 아래 변수는 화면을 열기 전 초기 구성을 넣을 때 씁니다.

이름	기본값	필수	설명
SSO_KEYCLOAK_ENABLED	false		SSO 사용 여부.
SSO_KEYCLOAK_ISSUER_URL	(없음)		예: https://keycloak.internal/realms/dataworks . 끝의 / 는 제거됩니다.
SSO_KEYCLOAK_CLIENT_ID	(없음)		OIDC client id.
SSO_KEYCLOAK_CLIENT_SECRET	(없음)		OIDC client secret. 예: replace-with-client-secret .
SSO_KEYCLOAK_REDIRECT_URI	(없음)		예: https://dataworks.example/auth/keycloak/callback .
SSO_KEYCLOAK_SCOPES	openid profile email		공백 또는 쉼표로 구분합니다.
SSO_KEYCLOAK_DEFAULT_ROLE	developer		매핑되지 않은 사용자에게 줄 내부 역할.
SSO_KEYCLOAK_ROLE_CLAIM	realm_access.roles		역할이 담긴 claim 경로.
SSO_KEYCLOAK_GROUP_CLAIM	groups		그룹이 담긴 claim 이름.
SSO_KEYCLOAK_ALLOW_LOCAL_LOGIN	true		로컬 이메일·비밀번호 로그인을 함께 허용할지 여부. 비상 접근 경로입니다.

기본 업스트림 AI

관리자 화면에 공급자를 등록하면 그 설정이 우선합니다. 아래는 화면 등록 전 기본 업스트림입니다.

이름	기본값	필수	설명
UPSTREAM_PROVIDER	openai		기본 공급자 이름.
UPSTREAM_BASE_URL	https://api.openai.com		OpenAI 호환 Base URL. 폐쇄망에서는 내부 주소로 바꿉니다.
UPSTREAM_API_KEY	(없음)		업스트림 API 키. 예: sk-replace-me .
OPENAI_API_KEY	(없음)		UPSTREAM_API_KEY 가 비어 있을 때 대신 쓰는 값.
UPSTREAM_TIMEOUT	10m		업스트림 호출 제한 시간.
UPSTREAM_DEFAULT_MODEL	(없음)		모델을 지정하지 않은 요청의 기본 모델.

한도·스트리밍·MCP

이름	기본값	필수	설명
AI_DEFAULT_STREAM	true		stream 을 생략한 Chat Completions의 기본값.
LIMITS_MAX_OUTPUT_TOKENS	0		일반 응답 출력 상한. 0 은 관리 상한 비활성. 양수는 262144 로 상한 제한됩니다.
LIMITS_AGENT_MAX_TOKENS	16384		내부 Agent 실행의 출력 상한.
LIMITS_MAX_REQUEST_BYTES	0		요청 본문 크기 상한. 0 은 비활성.
LIMITS_MAX_MESSAGES	0		한 요청의 메시지 수 상한. 0 은 비활성.
MCP_AGENTIC_MODEL	(없음)		Agentic MCP가 쓸 모델 이름.
MCP_MAX_AGENT_STEPS	8		Agentic MCP 최대 LLM 턴.
MCP_MAX_TOKENS	2048		MCP 각 LLM 턴의 출력 토큰.
MCP_MAX_TOOLS	32		모델에 노출할 최대 도구 수.
MCP_FORCE_TOOL_FIRST	true		첫 턴에 도구 호출을 강제할지 여부.

기능 플래그

이름	기본값	필수	설명
FEATURE_DATAWORKS	true		데이터 상품 기능. 끄면 이 문서의 대부분이 사라집니다.
FEATURE_AI_GATEWAY	true		OpenAI 호환 API와 AI 게이트웨이 기능.
FEATURE_K8S	false		레거시 K8s 운영 기능. 기본 비활성입니다.

로깅과 보존

이름	기본값	필수	설명
`LOG_RAW_PROMPTS`	`false`		프롬프트 원문 저장 여부. **개인정보가 남으므로 기본값을 유지하세요.**
`LOG_RAW_BODIES`	`false`		요청·응답 본문 원문 저장 여부. 같은 이유로 기본값 권장.
`LOG_RESPONSE_TEXT`	`false`		응답 텍스트 저장 여부.
`LOG_RESPONSE_MAX_BYTES`	`1048576`		저장할 응답 텍스트의 최대 바이트(1 MiB).
`LOG_QUEUE_SIZE`	`4096`		비동기 로그 큐 크기. 넘치면 예비 파일로 흘립니다.
`LOG_FALLBACK_PATH`	`data/fallback.ndjson` (이미지 `/data/fallback.ndjson`)		DB 적재 실패 시 기록할 예비 파일 경로.
`RETENTION_REQUEST_DAYS`	`90`		요청 이력 보존 일수.
`RETENTION_PROMPT_DAYS`	`30`		프롬프트 보존 일수.
`RETENTION_RESPONSE_DAYS`	`30`		응답 보존 일수.
`RETENTION_TEXT2SQL_REPLAY_DAYS`	`30`		Text2SQL 재현 번들 보존 일수.
`RETENTION_INTERVAL`	`1h`		보존 정책 정리 작업 주기.

캐시

이름	기본값	필수	설명
`CACHE_EMBEDDING_ENABLED`	`true`		임베딩 결과 캐시 사용 여부.
`CACHE_EMBEDDING_TTL`	`24h`		임베딩 캐시 보존 시간.
`CACHE_EMBEDDING_MAX_BYTES`	`1048576`		항목당 최대 크기(1 MiB).
`CACHE_EMBEDDING_PROVIDER`	(없음)		임베딩 전용 공급자 이름.
`CACHE_EMBEDDING_BASE_URL`	(없음)		임베딩 전용 Base URL.
`CACHE_EMBEDDING_API_KEY`	(없음)		임베딩 전용 API 키. 예: `sk-replace-me`.
`CACHE_CHAT_ENABLED`	`false`		채팅 응답 캐시. 응답이 비결정적이라 기본은 꺼져 있습니다.
`CACHE_CHAT_TTL`	`1h`		채팅 캐시 보존 시간.
`CACHE_CHAT_SEMANTIC_ENABLED`	`false`		의미 유사 질문까지 캐시로 처리할지 여부.
`CACHE_CHAT_SEMANTIC_MODEL`	(없음)		의미 캐시에 쓸 임베딩 모델.
`CACHE_CHAT_SEMANTIC_THRESHOLD`	`0.95`		캐시 적중으로 볼 유사도 하한.
`CACHE_CHAT_SEMANTIC_MAX_CANDIDATES`	`200`		유사도 비교 후보 수.
`CACHE_CHAT_SEMANTIC_MULTITURN`	`false`		다중 턴 대화에도 의미 캐시를 적용할지 여부.

비용·탄소·SLA

이름	기본값	필수	설명
PRICING_FALLBACK_MODEL	qwen-plus		단가 정보가 없는 모델에 적용할 대체 단가 모델.
PRICING_USD_KRW	1380		USD→KRW 환산율.
MODEL_PRICING_KRW_PER_1M	{}		모델별 100만 토큰 단가 JSON. 형식이 깨지면 기동이 실패합니다.
CARBON_WH_PER_1K_TOKENS	0.4		1000토큰당 소비 전력(Wh) 기본값.
CARBON_MODEL_WH_PER_1K	(없음)		모델별 Wh 값(모델=값 목록).
CARBON_PUE	1.2		데이터센터 PUE.
CARBON_GRID_INTENSITY_G	475		전력 1kWh당 탄소 배출량(g).
INSURANCE_SLA_TARGET	0.99		SLA 목표 성공률.
INSURANCE_FAST_BURN	14.4		빠른 소진 경보 배수.
INSURANCE_SLOW_BURN	3.0		느린 소진 경보 배수.
SKILLS_ENFORCEMENT	warn		스킬 정책 위반 처리 방식.

ClickHouse 분석 내보내기 (선택)

CLICKHOUSE_URL 이 비어 있으면 아무것도 전송하지 않습니다.

이름	기본값	필수	설명
CLICKHOUSE_URL	(없음)		ClickHouse HTTP 주소. 끝의 / 는 제거됩니다.
CLICKHOUSE_DB	default		데이터베이스 이름.
CLICKHOUSE_TABLE	analytics_daily		일자별 집계 테이블.
CLICKHOUSE_USER	(없음)		접속 계정.
CLICKHOUSE_PASSWORD	(없음)		접속 비밀번호. 예: replace-with-password .
CLICKHOUSE_SINK_INTERVAL	0		집계 전송 주기. 0 은 비활성.
CLICKHOUSE_SINK_DAYS	3		한 번에 다시 계산할 일수.
CLICKHOUSE_BATCH_SIZE	200		전송 배치 크기.
CLICKHOUSE_FLUSH_INTERVAL	5s		배치 플러시 주기.
CLICKHOUSE_MAX_QUEUE_SIZE	10000		전송 대기 큐 상한.
CLICKHOUSE_REQUEST_FACT_TABLE	(없음)		요청 팩트 테이블 이름. 비우면 전송하지 않습니다.
CLICKHOUSE_TOOL_FACT_TABLE	(없음)		도구 호출 팩트 테이블.
CLICKHOUSE_ROUTING_FACT_TABLE	(없음)		라우팅 팩트 테이블.
CLICKHOUSE_EVAL_FACT_TABLE	(없음)		평가 팩트 테이블.
CLICKHOUSE_FEEDBACK_FACT_TABLE	(없음)		피드백 팩트 테이블.
CLICKHOUSE_SKILL_FACT_TABLE	(없음)		스킬 팩트 테이블.
CLICKHOUSE_MULTIMODEL_FACT_TABLE	(없음)		멀티모델 비교 팩트 테이블.
CLICKHOUSE_POLICY_FACT_TABLE	(없음)		정책 판정 팩트 테이블.
CLICKHOUSE_TEXT2SQL_FACT_TABLE	(없음)		Text2SQL 팩트 테이블.

Text2SQL (기본 비활성, 선택)

이름	기본값	필수	설명
`TEXT2SQL_ENABLED`	`false`		자연어→SQL 기능 사용 여부.
`TEXT2SQL_DIALECT`	`PostgreSQL`		생성할 SQL 방언.
`TEXT2SQL_SCHEMA`	(없음)		대상 스키마 이름.
`TEXT2SQL_PREVIEW_MODEL`	`gpt-4.1-mini`		미리보기 생성 모델.
`TEXT2SQL_EXECUTE_MODEL`	`gpt-4.1-mini`		실행용 생성 모델.
`TEXT2SQL_ACCURATE_MODEL`	`claude-sonnet-4`		정확도 우선 모델.
`TEXT2SQL_LOCAL_MODEL`	`qwen-coder`		폐쇄망 로컬 모델.
`TEXT2SQL_SUMMARY_MODEL`	`gpt-4.1-mini`		결과 요약 모델.
`TEXT2SQL_EXEC_DRIVER`	`postgres`		실행 대상 DB 드라이버.
`TEXT2SQL_EXEC_DSN`	(없음)		실행 대상 DSN. 조회 전용 계정을 쓰세요.
`TEXT2SQL_TWIN_DRIVER`	`postgres`		검증용 트윈 DB 드라이버.
`TEXT2SQL_TWIN_DSN`	(없음)		검증용 트윈 DB DSN.
`TEXT2SQL_DEFAULT_LIMIT`	`100`		생성 쿼리의 기본 `LIMIT`.
`TEXT2SQL_MAX_LIMIT`	`1000`		허용 최대 `LIMIT`.
`TEXT2SQL_MAX_EXPLAIN_COST`	`0`		허용 최대 실행 계획 비용. `0`은 비활성.
`TEXT2SQL_STATEMENT_TIMEOUT`	`15s`		실행 제한 시간.
`TEXT2SQL_WORK_MEM`	(없음)		세션 `work_mem` 값.
`TEXT2SQL_MASK_RESULTS`	`true`		결과 마스킹 여부.
`TEXT2SQL_REQUIRE_DATE_FILTER`	`false`		날짜 조건을 필수로 요구할지 여부.
`TEXT2SQL_CLARIFY_ENABLED`	`false`		모호한 질문에 되묻기 사용 여부.
`TEXT2SQL_CACHE_ENABLED`	`true`		생성 결과 캐시 사용 여부.
`TEXT2SQL_CACHE_TTL`	`1h`		캐시 보존 시간.
`TEXT2SQL_SHADOW_MODELS`	(없음)		그림자 비교 모델 목록(쉼표 구분).
`TEXT2SQL_SHADOW_SAMPLE_RATE`	`0`		그림자 비교 표본 비율.
`TEXT2SQL_REPLAY_BUNDLES`	`false`		재현 번들 저장 여부.
`TEXT2SQL_DAILY_RISK_LIMIT`	`20`		하루 위험 쿼리 허용 수.
`TEXT2SQL_DAILY_RISK_WARN`	`0`		위험 쿼리 경고 기준. `0`은 비활성.

VCS 연동 (선택)

이름	기본값	필수	설명
`VCS_WEBHOOK_SECRET`	(없음)		웹훅 서명 검증 비밀. 예: `replace-with-webhook-secret`.
`VCS_INFER_FROM_CONTENT`	`true`		본문에서 저장소·변경 정보를 추론할지 여부.

5. 계정과 권한

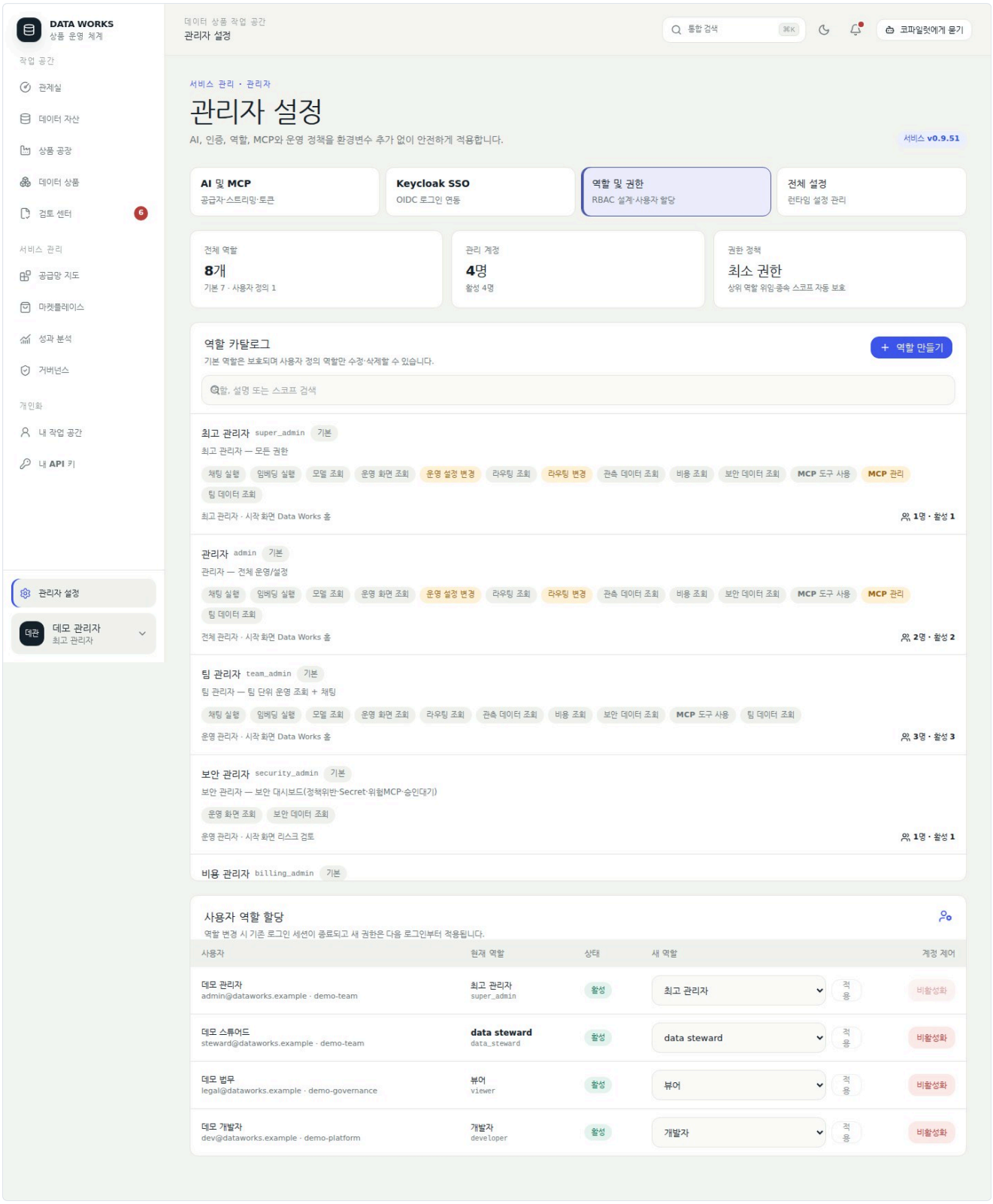
주요 내장 역할에는 `super_admin`, `admin`, `team_admin`, `team_manager`, `developer`, `viewer`, `service_account`, `ops_admin`, `ai_admin`, `security_admin`, `billing_admin`, `readonly_admin` 이 있습니다. Bootstrap 계정은 모든 Scope를 가진 `super_admin` 입니다.

역할이 할 수 있는 일

역할	등급	Scope	할 수 있는 일
`super_admin`	5	전체	모든 관리·설정·역할 위임. 마지막 활성 계정은 강등·비활성화할 수 없습니다.
`admin`	4	전체	전체 운영과 설정 변경. 자기와 동급 이상 역할은 위임할 수 없습니다.
`team_admin`	3	`admin:read`, `routing:read`, `observability:read`, `costs:read`, `security:read`, `team:read`, 채팅·임베딩·모델·MCP	팀 단위 운영 조회와 AI 사용. 설정 쓰기는 없습니다.
`ops_admin`	3	`admin:read`, `observability:read`, `costs:read`, `models:read`	관측·비용 조회와 담당 카테고리 설정 쓰기.
`ai_admin`	3	`admin:read`, `models:read`, `routing:read`, `observability:read`	모델·라우팅 설정 쓰기.
`security_admin`	3	`admin:read`, `security:read`	보안 대시보드. 로그인 후 리스크 검토 화면으로 이동합니다.
`billing_admin`	3	`admin:read`, `costs:read`, `observability:read`, `models:read`	비용 대시보드.
`team_manager`	2	`team:read`, 관측·비용·채팅·임베딩·모델·MCP	팀 대시보드. 운영 화면은 없습니다.
`developer`	2	채팅·임베딩·모델·라우팅 조회·관측·비용·MCP	AI 사용 중심. 운영 화면은 없습니다.
`service_account`	2	채팅·임베딩·모델·MCP	자동화용 계정.
`viewer`	1	`admin:read` 등 조회 Scope	운영 조회 전용.
`readonly_admin`	1	`admin:read`, `observability:read`, `costs:read`, `security:read`	운영 조회 전용, 변경 불가.

역할 및 권한 화면

`관리자 설정 → 역할 및 권한`은 기본 역할과 사용자 정의 역할, 역할별 사용자 수와 실제 할당 계정을 한 화면에서 관리합니다.



역할 및 권한 — 기본·사용자 정의 역할의 스코프와 각 계정의 역할 할당을 한 화면에서 관리한다

- 기본 역할은 제품 권한 기준이므로 수정하거나 삭제할 수 없습니다.
- 사용자 정의 역할은 영문 소문자로 시작하는 식별자, 설명, 로그인 시작 화면과 필요한 Scope를 조합해 생성합니다.
- `admin:write`, `routing:write`, `mcp:admin` 을 선택하면 각각 필요한 조회·사용 Scope가 자동 포함되며 서버도 같은 종속성을 검증합니다.
- 관리자는 자기 역할보다 낮은 등급만 설계·할당할 수 있고 `super_admin` 만 관리자 동급 역할을 위임할 수 있습니다.
- 사용자에게 할당된 역할은 먼저 다른 역할로 이동해야 삭제할 수 있습니다.
- 사용자 역할 또는 사용자 정의 역할의 Scope를 변경하면 기존 로그인 세션이 즉시 종료되어 다음 로그인부터 새 권한이 적용됩니다.
- 마지막 활성 `super_admin` 은 강등하거나 비활성화할 수 없습니다.

역할 변경은 확인 대화상자를 거쳐 적용되고 감사 로그에 남습니다. Keycloak 역할 매핑의 내부 역할 이름에도 기본 역할과 여기서 만든 사용자 정의 역할을 사용할 수 있습니다.

```
GET    /admin/roles
POST   /admin/roles
DELETE /admin/roles?role={role}
GET    /admin/users
PATCH  /admin/users/{user_id}
```

개인 API 키 정책 운영

계정 로그인 사용자는 `/dataworks/personal/keys` 에서 자신의 역할 범위 안에 개인 키를 발급할 수 있습니다. 비밀값은 발급·회전 직후 한 번만 표시됩니다.

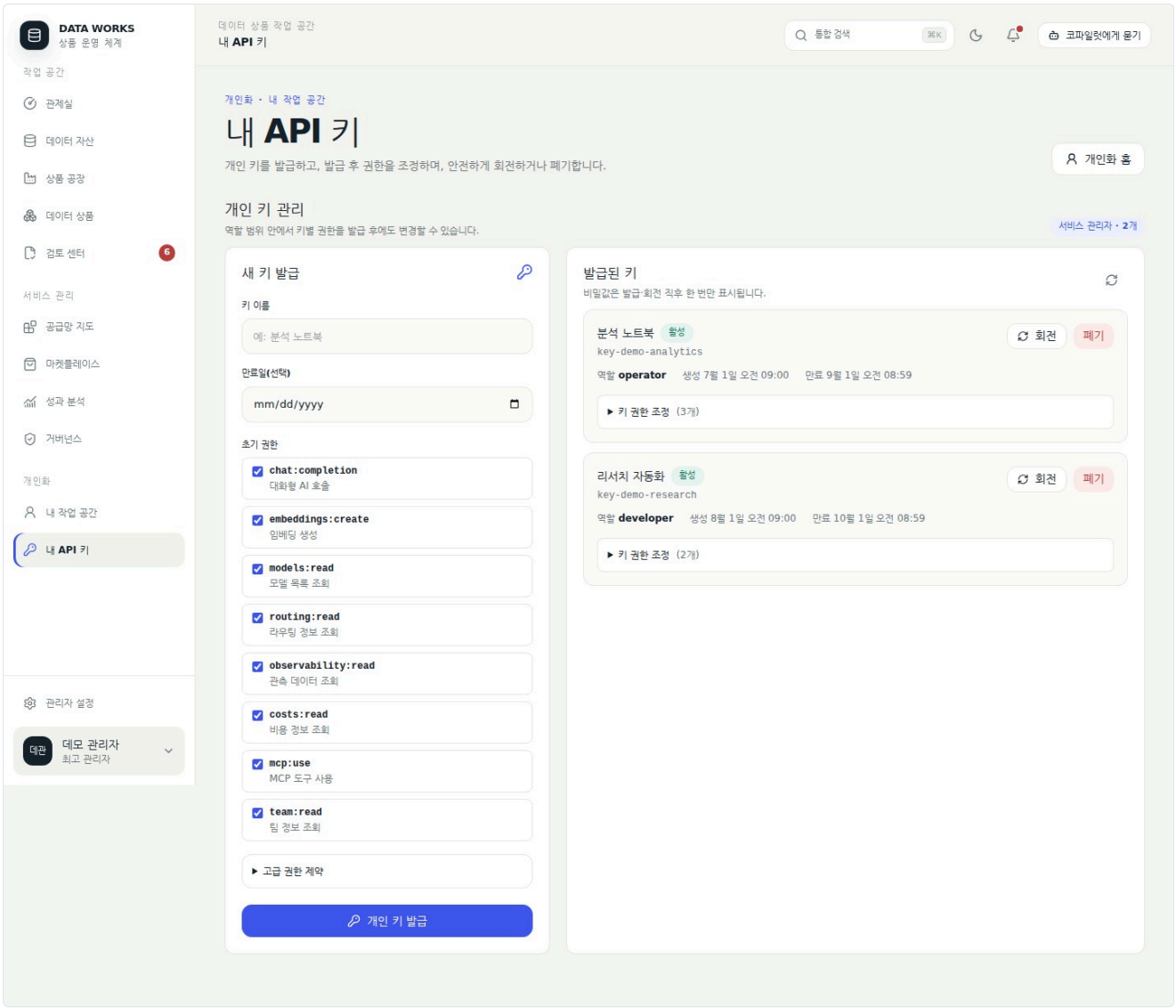
발급 후 변경 가능한 정책은 정확히 다음 8개입니다.

1. `scopes`
2. `allowed_ips`
3. `allowed_models`
4. `denied_models`
5. `allowed_providers`
6. `denied_providers`
7. `budget_limit_krw`
8. `expires_at`

서버는 최종 정책이 호출자 정책의 부분집합인지 검사합니다. 상위 키나 사용자 역할보다 넓은 Scope, 모델, 공급자, IP, 예산 또는 만료 범위로 확대할 수 없습니다.

회전은 정책을 그대로 보존한 새 키를 원자적으로 만들고 기존 키를 폐기합니다. 폐기와 다른 사용자의 키 ID 조회는 소유권 검사를 적용합니다.

```
GET    /me/keys
POST   /me/keys
PATCH  /me/keys/{id}
POST   /me/keys/{id}/rotate
DELETE /me/keys/{id}
```



내 API 키 — 발급 후에도 Scope·IP·모델·예산·만료를 조정하고 회전할 수 있다

6. MCP 운영

Data Works에는 서로 다른 두 MCP 엔드포인트가 있습니다.

엔드포인트	용도
`/mcp`	등록된 외부 MCP 서버의 도구·프롬프트·리소스 집약
`/mcp/gateway`	Data Works 자체 기능을 MCP 도구로 제공

React `AI 및 MCP` 설정에서는 Agentic MCP 모델과 토큰·턴·도구 수 정책을 관리합니다. 외부 MCP 업스트림 등록, Bearer 인증, allowlist·차단 정책, 도구별 역할 Scope, Route Explain, 연결 시험과 호출 관측은 기존 `/admin` 콘솔의 MCP 메뉴에서 관리합니다.

MCP 업스트림 인증 토큰은 암호화 저장됩니다. 개인 키에는 `mcp:use` Scope가 필요하고, MCP 관리에는 `mcp:admin` 등 관리자 권한이 필요합니다.

주요 API:

```
GET/POST    /admin/mcp/upstreams
GET/PUT/DELETE /admin/mcp/upstreams/{id}
GET/POST    /admin/mcp/policies
GET/DELETE  /admin/mcp/policies/{server}
GET/POST    /admin/mcp/tool-scopes
POST        /admin/mcp/route/explain
POST        /admin/mcp/test
```

사용자 연결 템플릿과 진단:

```
GET  /me/onboarding-pack?client=mcp|cursor|roo|cline|openai-sdk
POST /me/connection-doctor
```

7. 데이터 상품 운영 흐름

React Workbench는 현황 탐색과 출시 게이트 외에도 쓰기 권한이 있는 사용자에게 자산 등록·수정·준비도 재평가·안전 삭제, 상품 생성·수정·상태 전환·초안/보관 삭제, Canvas 저장, 승인 추적 추가·갱신과 append-only 계약 버전 생성을 제공합니다. AI 상품 공장 실행 생성·재생과 일부 고급 운영은 REST API 또는 기존 관리자 콘솔을 사용합니다.

7.1 데이터 자산

```
GET/POST/DELETE /admin/dataworks/assets
GET/POST /admin/dataworks/assets/readiness
POST     /admin/dataworks/assets/{asset_key}/readiness/check
GET      /admin/dataworks/assets/{asset_key}/lineage
```

화면에서 자산의 키, 이름, 도메인, 담당자, 컬럼 요약, 민감도와 갱신 주기를 등록·수정하고 개별 또는 전체 준비도를 재평가합니다. 삭제는 확인 절차를 거치며, 상품이 원천 자산으로 참조 중이면 `409 Conflict`로 차단됩니다. 민감 상품 출시 전에 준비도 70 이상인지 확인합니다.

7.2 아이디어와 상품 정의

```
POST /admin/dataworks/factory/ideas
POST /admin/dataworks/factory/definitions
POST /admin/dataworks/scoring/evaluate
POST /admin/dataworks/similarity/check
GET/POST/DELETE /admin/dataworks/products
POST /admin/dataworks/products/{product_key}/{submit|approve|reject|archive}
```

React 상품 공장은 현재 실행 관찰 화면이며 새 실행 생성 UI가 아닙니다. 대신 `데이터 상품` 목록에서 상품을 생성·수정하고, 상품 작업 공간에서 제출·승인·반려·보관 상태 전환을 제어합니다. 화면의 삭제 동작은 `draft` 또는 `archived` 상태에서만 제공되며, 삭제한 상품 키는 보존된 과거 승인·계약 이력의 오연결을 막기 위해 재사용할 수 없습니다.

7.3 Product Canvas

```
GET/POST /admin/dataworks/products/{product_key}/canvas
POST     /admin/dataworks/products/{product_key}/canvas/generate
```

고객 문제, 구매자, 사용 사례, 제공 데이터, 차별점, 가격 모델, 위험 참고, PoC 성공 기준과 예상 수익을 관리합니다. 상품 작업 공간의 `블루프린트` 탭에서 Canvas를 편집해 저장할 수 있습니다.

7.4 위험 검토와 승인

```
POST /admin/dataworks/risk/check
GET  /admin/dataworks/reviews
POST /admin/dataworks/reviews/{product_key}/approve
POST /admin/dataworks/reviews/{product_key}/reject
GET/POST /admin/dataworks/products/{product_key}/approvals
```

필수 승인 step은 `data_owner`, `legal`, `compliance` 입니다. 승인 `expires_at`이 지나면 출시 증적으로 인정되지 않습니다. React 검토 센터는 필터와 상품 이동을 제공하고, 연결된 상품 작업 공간의 `승인` 탭에서 승인 추적 항목을 추가하거나 기존 결정·증적·만료 정보를 갱신할 수 있습니다.

7.5 Evidence Pack

```
GET/POST /admin/dataworks/products/{product_key}/evidence-pack
```

상품, Canvas, 원천 자산, 준비도, 위험 검토, 승인, PoC, API 계약, 계약 버전과 출시 게이트 결과를 감사 가능한 JSON으로 묶습니다. 본문 없이 `POST` 하면 현재 저장 정보를 기준으로 생성합니다.

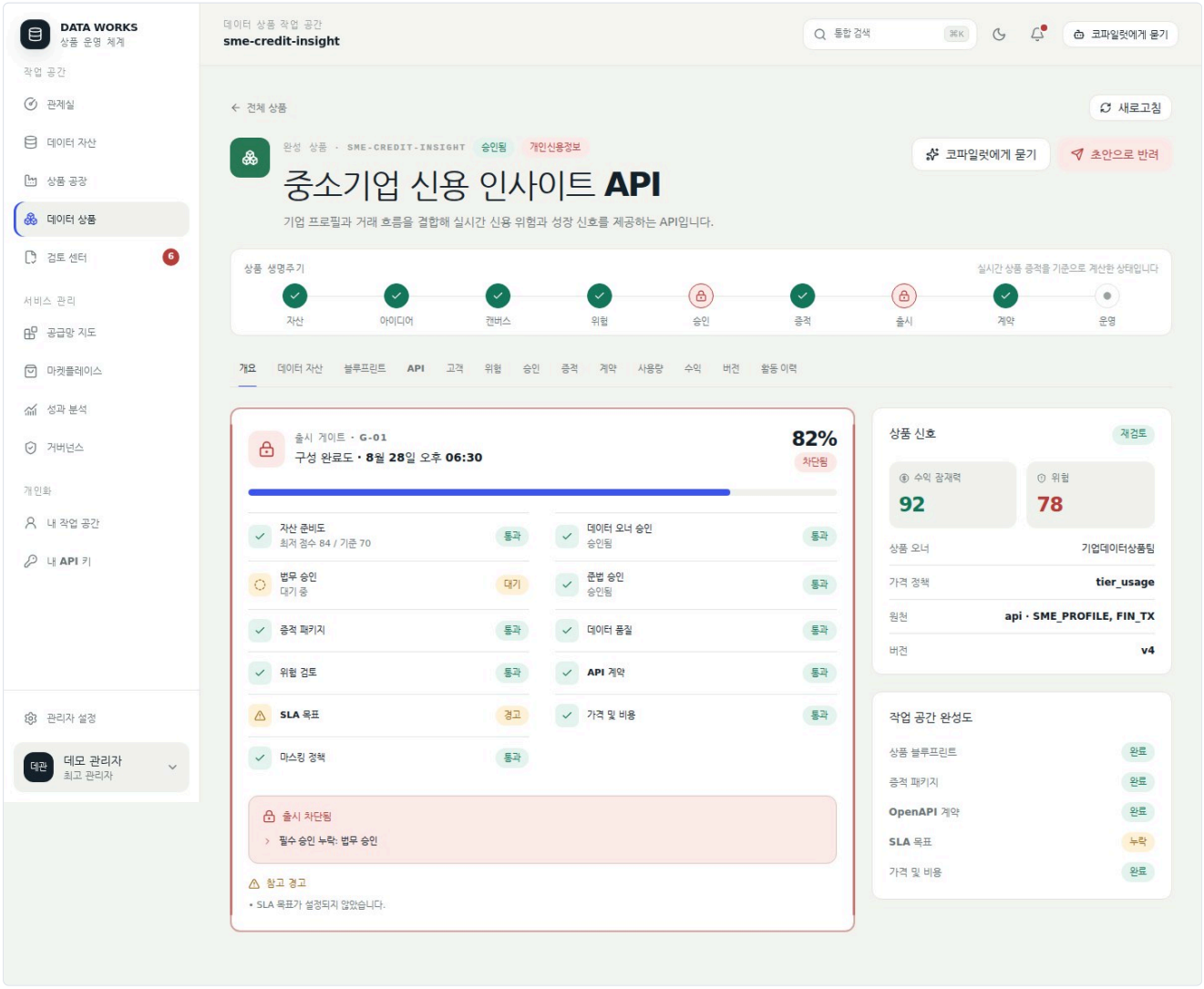
7.6 Publish Gate와 출시

`risk_score >= 70` 또는 `restricted`, `personal_credit`, `pseudonymized` 등 민감 프로필에는 엄격 게이트를 적용합니다.

- 모든 원천 자산 준비도 70 이상
- 데이터 오너·법무·준법 승인 또는 면제
- Evidence Pack 존재
- 승인 증적 미만료

```
GET  /admin/dataworks/products/{product_key}/publish-gate
POST /admin/dataworks/products/{product_key}/publish
```

조건 미충족 시 `409 Conflict` 와 차단 사유를 반환합니다.



상품 출시 게이트 — 조건별 통과·차단 사유와 생명주기 단계를 함께 보여준다

7.7 계약, Entitlement와 상품 API

```
GET/POST /admin/dataworks/products/{product_key}/contract-versions
GET/POST /admin/dataworks/products/{product_key}/contract-scopes
GET/POST /admin/dataworks/products/{product_key}/entitlements
GET/POST /admin/dataworks/products/{product_key}/sla
GET/POST /admin/dataworks/products/{product_key}/watermarks
GET/POST /admin/dataworks/products/{product_key}/costs
POST     /v1/data-products/{product_key}/query
```

권장 순서:

1. 상품 작업 공간의 `계약` 탭에서 계약 정의를 새 버전으로 추가하고, 고객별 허용 필드, 호출 한도, 기간과 목적을 관리합니다.
2. Entitlement로 고객 API 키를 계약 Scope에 연결합니다.
3. SLA, 최신성 Watermark와 비용·마진을 설정합니다.
4. 런타임 상품 API를 호출해 출시, 계약 범위와 만료 검사를 확인합니다.

쓰기 경로는 런타임이 절대 서빙할 수 없는 계약·권한을 저장 시점에 거부합니다. `400 invalid_allowed_fields` (허용 필드 없음), `400 invalid_masking_policy` (런타임이 수행하지 않는 마스킹), `400 invalid_access_window` (`valid_from` 이 `valid_to` 보다 뒤), `400 invalid_contract_status` · `400 invalid_entitlement_status` (허용값은 `active` · `draft` · `suspended` · `revoked`)를 확인하세요. 자세한 판정 규칙은 운영 가이드 5절에 있습니다.

Product Workspace의 API·고객·사용량·수익·버전·활동 이력 탭은 현재 안내 화면입니다. 위 API가 React 탭에서 편집 가능하다고 가정하지 마세요.

계약 버전은 감사 가능성을 유지하기 위한 append-only 이력입니다. React UI와 관리 API는 새 버전 생성을 지원하지만 기존 계약 버전이나 과거 이력의 수정·삭제는 지원하지 않습니다.

7.8 만료 예정 계약과 권한 확인

```
GET /admin/dataworks/action-center?expiring_within=30d
```

기본 예고 창은 30일이며 `expiring_within` 으로 `13w` , `45d` , `72h` 처럼 지정할 수 있습니다. 해석할 수 없거나 0 이하인 값은 `400 invalid_expiring_within` 으로 거부합니다. 응답의 `expiring_within` 은 실제로 적용된 창입니다. 활성 계약과 활성 엔타이틀먼트만 예고 대상이며 이미 닫은 `revoked` · `draft` · `suspended` 계약은 집계하지 않습니다.

7.9 Factory Run 재현성과 평가

```
GET      /admin/dataworks/factory/runs
POST     /admin/dataworks/factory/runs/{id}/replay
POST     /admin/dataworks/factory/runs/{id}/evaluate
GET/POST /admin/dataworks/prompt-templates
```

재실행은 원본 `input_hash` 와 부모 실행 계보를 보존합니다. 평가는 정확도, 유용성, 위험 통제와 출력 품질을 기록합니다. 현재 React 상품 공장 화면은 실행 생성·재생·평가 버튼이 아닌 조회 화면입니다.

7.10 OpenAPI와 개발자 문서

- 서비스 전체 명세: `/openapi.json`
- Swagger UI: `/swagger`
- 상품별 OpenAPI: `/admin/dataworks/products/{product_key}/openapi`
- 상품 런타임: `/v1/data-products/{product_key}/query`

서비스 전체 OpenAPI에는 개인 키 정책 변경과 회전 응답도 문서화되어 있습니다. 프로필 메뉴에서 전체 OpenAPI와 Swagger를 바로 열 수 있습니다.

8. 운영

8.1 상태 점검 엔드포인트

메서드·경로	인증	응답
`GET /health` , `GET /healthz`	불필요	`{"status":"ok"}`
`GET /ready` , `GET /readyz`	불필요	DB `Ping` 성공 시 `{"status":"ready"}` , 실패 시 `503` 과 `{"status":"not_ready","error":"…"}`
`GET /metrics`	불필요	Prometheus 텍스트 형식. 로그 큐 깊이·유실·기록 수를 포함합니다.

`/health` 는 프로세스 생존만, `/ready` 는 DB 연결까지 확인합니다. 로드밸런서 헬스체크에는 `/ready` 를 씁니다. `/metrics` 는 인증이 없으므로 외부에 노출하지 말고 내부망 또는 프록시 ACL 뒤에 두세요.

8.2 로그 위치

- 컨테이너 표준 출력: `docker logs dataworks` . `log/slog` 텍스트 형식이며 `level=INFO msg="Data Works listening" …` 같은 줄이 나옵니다.
- 요청·프롬프트·응답 이력: PostgreSQL. 보존 기간은 `RETENTION_*` 변수로 정합니다.
- DB 적재 실패분: `LOG_FALLBACK_PATH` (배포 이미지는 `/data/fallback.ndjson`). 이 파일이 커지고 있으면 DB 쓰기가 막힌 것입니다.
- 감사 이력: DB에 적재되며 역할 변경·설정 변경·상품 조회 거부가 남습니다.

8.3 백업과 복구

PostgreSQL 운영에서는 표준 유틸리티를 씁니다.

```
# 백업
pg_dump "$POSTGRES_DSN" > dataworks-$(date +%F).sql

# 복구 (빈 데이터베이스에 적용)
psql "$POSTGRES_DSN" < dataworks-YYYY-MM-DD.sql
```

`ENCRYPTION_KEY`를 함께 보관하지 않으면 복구해도 저장된 AI 공급자 키, Keycloak Secret, MCP 업스트림 토큰을 해독할 수 없습니다. **DB 백업과 암호화 키를 같은 절차로 관리하세요.**

백업 대상 테이블 목록과 SQLite 모드 절차는 운영 가이드 7절, 컨테이너/원격 PostgreSQL 명령 예시는 PostgreSQL 가이드 6절에 있습니다. 저장소의 `scripts/backup.sh`는 SQLite 파일과 `fallback.ndjson`을 묶는 스크립트이므로 PostgreSQL 운영에서는 위 `pg_dump`를 쓰세요.

8.4 업그레이드와 롤백

1. PostgreSQL을 백업하고 현재 이미지 태그와 서비스 버전을 기록합니다.
2. 새 릴리즈 archive를 적재합니다: `gunzip -c dataworks-<새 버전>.tar.gz | docker load`
3. 네 환경변수를 그대로 두고 이미지 태그만 새 버전으로 바꿔 다시 기동합니다.
4. `/ready`와 로그인, 프로필 메뉴의 버전 표시를 확인합니다.

기동 시 DB 마이그레이션이 자동 실행됩니다. 롤백은 이전 이미지 태그로 컨테이너를 다시 실행하는 방식이지만, **마이그레이션이 적용된 DB를 이전 버전이 읽을 수 있는지 릴리즈 노트를 먼저 확인해야 합니다.** 하위 호환이 보장되지 않으면 3-1단계에서 받아 둔 백업으로 DB를 함께 되돌립니다. 상세 절차는 릴리즈 가이드를 따릅니다.

8.5 점검표

일일

- `/health`, `/ready` 확인
- 관제실의 출시 차단, 승인 대기, 만료와 마진 경고 확인
- AI 공급자 오류와 MCP 도구 오류 확인
- 키 만료와 비정상 사용량 확인

변경 전

- PostgreSQL 백업
- 현재 이미지 태그와 서비스 버전 기록
- Keycloak 로컬 비상 로그인 확인
- 관리자 런타임 설정과 암호화 키 보관 상태 확인

변경 후

- 로그인과 프로필 버전 확인
- `/openapi.json`과 `/swagger` 확인
- 개인 키 발급→정책 변경→회전→폐기 시험
- Keycloak 연결 진단과 SSO 로그인 시험
- AI 스트리밍과 출력 토큰 상한 시험
- MCP `initialize`, `tools/list`와 정책 시험
- 상품 출시 게이트 차단·허용 경로 시험

9. 장애 대응

증상	확인할 곳	조치
컨테이너가 바로 죽는다	`docker logs dataworks` 에 `level=ERROR msg="invalid configuration"`	필수 네 변수 중 빠진 값이나 `MODEL_PRICING_KRW_PER_1M` JSON 형식 오류를 고칩니다.
컨테이너가 바로 죽는다	`msg="open database"` 또는 `msg="migrate database"`	DSN, 네트워크, DB 계정 권한을 확인합니다. 마이그레이션 실패는 백업 후 재시도합니다.
`/ready` 가 `503 not_ready`	응답의 `error` 문자열, PostgreSQL 상태	DB 연결이 끊긴 상태입니다. DB 기동과 방화벽, 커넥션 수를 확인합니다.
로그인은 되는데 화면이 `접근 권한이 없습니다`	`관리자 설정 → 역할 및 권한` 의 해당 계정 역할	역할을 올리고 사용자에게 다시 로그인하도록 안내합니다(역할 변경 시 세션이 종료됩니다).
`Keycloak SSO로 계속` 버튼이 안 보인다	`GET /auth/sso/status` 응답	SSO가 꺼져 있거나 저장에 실패한 상태입니다. `POST /admin/sso/keycloak/test` 로 Issuer·RSA 다.
모든 계정이 로그인하지 못한다	`로컬 로그인 허용` 설정	로컬 로그인을 끈 상태에서 SSO가 고장난 경우입니다. 환경 변수 `SSO_KEYCLOAK_ALLOW_LOCAL_LOG` 해 Bootstrap 계정으로 복구합니다.
상품 출시가 계속 막힌다	`GET /admin/dataworks/products/{key}/publish-gate` 의 차단 사유	준비도·승인·Evidence Pack·승인 만료 중 무엇이 빠졌는지 보고 그 항목을 채웁니다.
고객이 `403 inactive_entitlement`	`GET /admin/dataworks/products/{key}/entitlements`	권한 `status` 가 `active` 인지, `expires_at` 이 지나지 않았는지 확인합니다.
고객이 `403 contract_scope_inactive`	계약의 `status` 와 `valid_from` · `valid_to`	유효 기간 밖이거나 `draft` · `suspended` · `revoked` 상태입니다. 새 계약 창을 저장합니다.
고객이 `429 contract_rate_limited`	계약의 `rate_limit`	분당 한도를 넘었습니다. 한도를 조정하거나 고객에게 호출 간격을 안내합니다.
계약 저장이 `400` 으로 거부된다	응답의 코드	`invalid_allowed_fields` · `invalid_masking_policy` · `invalid_access_window` · `invalid_` 는 런타임이 서빙할 수 없는 값이라 거부한 것입니다. 7.7절의 허용값을 확인합니다.
AI 호출이 전부 실패한다	`관리자 설정 → AI 및 MCP` 의 공급자 활성 상태와 Base URL	폐쇄망에서 외부 주소가 남아 있으면 내부 주소로 바꿉니다. 저장 후 `/v1/models` 로 확인합니다.
`fallback.ndjson` 이 계속 커진다	`/metrics` 의 로그 큐 지표, DB 쓰기 상태	DB 적재가 막힌 상태입니다. DB 용량·연결을 복구한 뒤 파일을 보관 이동합니다.
Evidence Pack 생성 실패	상품 존재 여부, 최신 definition/risk/poc 조회 오류	상품과 선행 증적을 확인하고 마이그레이션 상태를 점검합니다.

더 많은 사례는 운영 가이드 8절에 있습니다.

10. 보안

기본값 중 반드시 바꿔야 하는 것

- `BOOTSTRAP_ADMIN_PASSWORD` : 최초 로그인 후 비밀번호를 바꾸고, 환경 변수에 남은 값은 비밀 저장소로 옮깁니다.
- `ENCRYPTION_KEY` : 예시 값을 그대로 쓰지 말고 `openssl rand -hex 32` 로 생성합니다. 운영 중 변경하지 않습니다.
- `UPSTREAM_BASE_URL` : 기본값은 외부 `https://api.openai.com` 입니다. 폐쇄망에서는 내부 주소로 바꾸거나 관리자 화면에서 공급자를 등록합니다.
- `ADMIN_TOKEN` · `ADMIN_READONLY_TOKEN` · `PROXY_API_KEYS` : 계정 로그인을 쓰는 배포에서는 비워 둡니다. 값이 있으면 역할 검사를 우회하는 별도 경로가 살아 있게 됩니다.

외부에 열면 안 되는 것

대상	이유
PostgreSQL `5432`	데이터 원본. 컨테이너 네트워크 안에서만 접근하게 합니다.
`GET /metrics`	인증이 없습니다. 내부망 또는 프록시 ACL 뒤에 둡니다.
`/admin` 레거시 콘솔	필요한 경우에만 노출합니다. 표준 진입점은 `/dataworks/` 입니다.

외부에 공개하는 것은 리버스 프록시의 `8080` 하나이며, TLS 종료와 접근 제어를 프록시에서 적용합니다.

개인정보와 로그

`LOG_RAW_PROMPTS` , `LOG_RAW_BODIES` , `LOG_RESPONSE_TEXT` 는 기본이 모두 `false` 입니다. 켜면 프롬프트·요청 본문·응답 원문이 DB에 남으므로, 조사 목적으로 한시적으로만 켜고 `RETENTION_PROMPT_DAYS` · `RETENTION_RESPONSE_DAYS` 를 함께 낮추세요. 원문을 볼 수 있는 역할은 `super_admin` , `admin` , `security_admin` 뿐이고 나머지 역할은 마스킹된 텍스트만 봅니다.

민감 데이터 상품은 계약의 마스킹 정책으로 응답 값을 가립니다. 런타임이 실제로 수행하는 정책은 `redact` 와 `hash` 뿐이며, 다른 문자열은 쓰기 시점에 `400 invalid_masking_policy` 로 거부합니다. 서술형 문구를 마스킹 근거로 저장할 수 없습니다.

인증 연동

Keycloak 역할 매핑은 권한 상승 방지 검사를 거칩니다. 일반 관리자는 자신과 동급 이상의 매핑을 만들 수 없고 `super_admin` 매핑은 `super_admin` 만 만들 수 있습니다. SSO를 켠 뒤에도 로컬 로그인을 완전히 끄기 전에 관리자 SSO 계정으로 로그인이 되는지 반드시 확인하세요.

관련 문서

- 사용자 가이드
- 운영 가이드
- 안전 및 보안 가이드
- PostgreSQL 가이드
- 릴리즈 가이드