

Ptium

관리자 가이드

설치·설정·운영·장애 대응·보안

버전	v1.69.32
작성일	2026-09-11
문서	관리자 가이드

Ptium 관리자 가이드

이 문서는 Ptium 을 띄워 놓고 지키는 사람을 위한 안내입니다. 화면을 쓰는 법은 [사용자 가이드](#)에 있고, 여기서는 되풀이하지 않습니다. 설계 배경과 보안 결정의 이유는 [architecture.md](#)·[security.md](#)에, 폐쇄망 반입 절차의 원문은 [offline-deployment.md](#)에 있습니다.

실린 화면은 모두 Ptium v1.69.32 릴리즈 이미지를 실제로 띄워 찍은 것입니다. 데이터는 예시(`hong@example.com`)입니다.

1. 구성 요소

Ptium 은 정적 바이너리 하나가 든 컨테이너 하나입니다. 워크스페이스(React 번들)·REST API·MCP 엔드포인트를 같은 포트에서 제공하고, 리버스 프록시나 별도 웹 컨테이너는 없습니다.

구성 요소	무엇	누가 제공
ptium 컨테이너	워크스페이스 / , API /api/v1 , MCP /mcp , 상태 /healthz · /readyz — 모두 8080	릴리즈 이미지 <code>ptium:<version></code>
PostgreSQL 14+	템플릿·덱·설정·자격 증명·오류 기록 전부. 기동 시 마이그레이션 자동 적용	배포 환경(번들에 없음)
이미지 볼륨 (선택)	<code>ASSET_STORAGE=filesystem</code> 일 때만 올린 그림 파일을 둬	배포 환경
OIDC 제공자 (선택)	Keycloak 등 표준 OIDC. 없으면 로컬 비밀번호 관리자로 운영 가능	배포 환경
AI 모델 (선택)	OpenAI 호환 Chat Completions API. 없으면 내장 작성기	배포 환경

컨테이너는 상태를 갖지 않습니다. 데이터베이스(와 볼륨을 쓰면 볼륨)만 남기면 컨테이너는 언제든 버리고 다시 띄울 수 있습니다. 복제본을 여러 개 띄워도 됩니다 — 생성 작업은 `SELECT ... FOR UPDATE SKIP LOCKED` 로 나눠 가져 같은 덱을 두 번 만들지 않습니다.

2. 설치

2.1 릴리즈 자산

인터넷이 되는 곳에서 v1.69.32 자산을 받아 반입합니다.

파일	용도
<code>ptium-1.69.32.tar.gz</code>	Docker 이미지 아카이브 (<code>ptium:1.69.32</code> + <code>ptium-1.69.32:latest</code> , Linux/AMD64)
<code>ptium-1.69.32.tar.gz.sha256</code>	체크섬
<code>docker-compose.ptium-1.69.32.yml</code>	폐쇄망 compose 파일
<code>ptium-1.69.32.env.example</code>	환경 변수 예시
<code>ptium-1.69.32.kubernetes.yaml</code>	Kubernetes 매니페스트
<code>load-ptium-1.69.32.sh</code> / <code>.ps1</code>	체크섬 확인 후 <code>docker load</code> 하는 도우미

2.2 이미지 반입

```
./load-ptium-1.69.32.sh ptium-1.69.32.tar.gz
```

도우미가 없으면 같은 일을 직접 합니다.

```
sha256sum -c ptium-1.69.32.tar.gz.sha256
gzip -dc ptium-1.69.32.tar.gz | docker load
docker image inspect ptium-1.69.32:latest ptium:1.69.32 >/dev/null
```

2.3 데이터베이스

이미 운영 중인 PostgreSQL 에 역할과 데이터베이스를 만듭니다. Ptium 은 자기 테이블을 만들 권한이 있으면 되고, 그 외 확장은 필요 없습니다.

```
CREATE ROLE ptium LOGIN PASSWORD 'change-me-to-a-long-random-password';
CREATE DATABASE ptium OWNER ptium;
```

평가용으로 PostgreSQL 까지 컨테이너로 띄우려면(운영에서는 권장하지 않습니다):

```
docker network create ptium
docker run -d --name ptium-db --network ptium \
  -e POSTGRES_DB=ptium -e POSTGRES_USER=ptium -e POSTGRES_PASSWORD='change-me' \
  -v ptium-postgres:/var/lib/postgresql/data postgres:16-alpine
```

2.4 환경 변수와 첫 관리자

ptium-1.69.32.env.example 을 .env 로 복사하고 최소 다음을 채웁니다. 첫 관리자는 환경 변수로 지정합니다 — OIDC 를 붙이기 전에도 로그인해 설정할 수 있어야 하기 때문입니다.

```
PTIUM_VERSION=1.69.32
DATABASE_URL=postgres://ptium:change-me@ptium-db:5432/ptium?sslmode=require
PUBLIC_BASE_URL=https://slides.example.com
KEY_ENCRYPTION_SECRET=<openssl rand -base64 32 결과>

BOOTSTRAP_ADMIN=admin@example.com
BOOTSTRAP_ADMIN_PASSWORD=<12자 이상, 여기서만 쓰는 값>
BOOTSTRAP_ADMIN_NAME=서비스 관리자
```

비밀번호는 계정을 처음 만들 때 한 번만 읽습니다. 제품 안에서 바꾼 비밀번호는 재시작해도 유지되고, 환경 변수는 다시 읽지 않습니다.

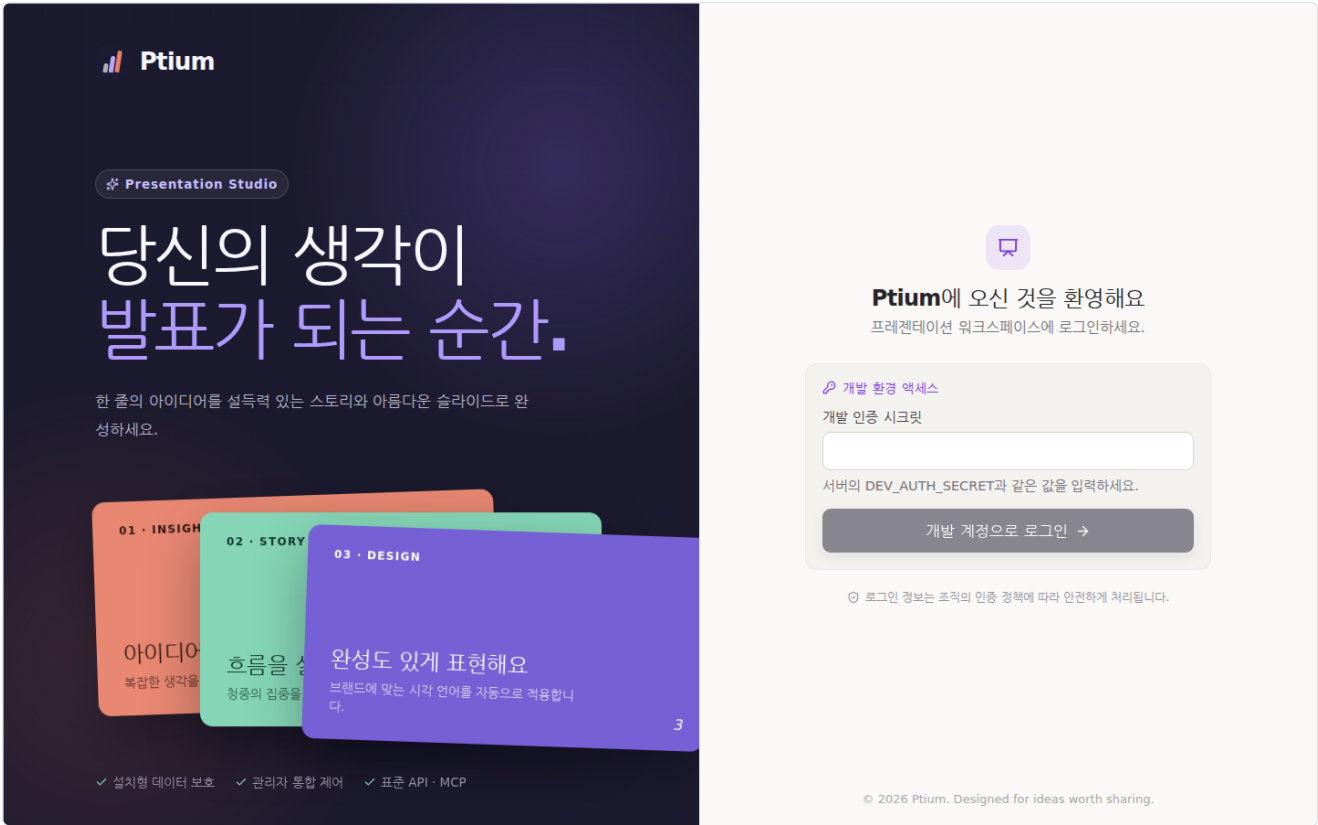
2.5 Compose 로 기동

```
docker compose --env-file .env -f docker-compose.ptium-1.69.32.yml up -d
docker compose --env-file .env -f docker-compose.ptium-1.69.32.yml ps
curl --fail http://localhost:8080/readyz
```

기동 로그에 다음 두 줄이 보이면 준비된 것입니다.

```
{"level":"INFO","msg":"database migrations ready","versions":64}
{"level":"INFO","msg":"Ptium server listening","address":":8080","version":"1.69.32","oidc_enabled":false,"dev_auth_enabled":false}
```

브라우저로 PUBLIC_BASE_URL 을 열면 로그인 화면이 나오고, BOOTSTRAP_ADMIN 계정으로 들어갑니다. 로그인 직후 비밀번호를 바꾸고(개인화 화면의 비밀번호 변경), OIDC 를 붙인 뒤에는 BOOTSTRAP_ADMIN_PASSWORD 를 환경에서 지웁니다.



로그인 — 배포가 허용한 인증 방식이 오른쪽에 보인다

2.6 Kubernetes 로 기동

```
kubectl create secret generic ptium \
  --from-literal=DATABASE_URL='postgres://ptium:change-me@postgres:5432/ptium?sslmode=require' \
  --from-literal=KEY_ENCRYPTION_SECRET="$(openssl rand -base64 32)" \
  --from-literal=BOOTSTRAP_ADMIN='admin@example.com' \
  --from-literal=BOOTSTRAP_ADMIN_PASSWORD='<12자 이상>'
kubectl apply -f ptium-1.69.32.kubernetes.yaml
```

매니페스트는 복제본 2개를 비루트 사용자(uid 65532)·읽기 전용 루트 파일시스템으로 띄우고, `/readyz` · `/healthz` 프로브를 겁니다. 인그레스가 앞에 있으면 요청 본문 한도를 `generation.max_template_mb` (기본 32 MiB) 이상으로 올립니다 — 번들 매니페스트는 `nginx` 용으로 `proxy-body-size: 64m` 을 넣어 둡니다.

2.7 포트 · 볼륨 · 자원

항목	값
포트	컨테이너 8080 하나. 워크스페이스·API·MCP·상태 점검 모두
볼륨	기본(<code>ASSET_STORAGE=database</code>)은 없음. <code>filesystem</code> 이면 <code>ASSET_DIR</code> (기본 <code>/var/lib/ptium/assets</code>), <code>uid/gid 65532</code> 가 쓸 수 있어야 함
메모리	요청 192Mi · 제한 1Gi (매니페스트 기준). 40장 텍 네 개를 사진과 함께 동시에 PDF 로 찍을 때 639MiB 까지
CPU	요청 100m
데이터베이스	PostgreSQL 14+ 한 개, 테이블 생성 권한

3. 설정

3.1 환경 변수 (기동 시)

`server/internal/config/config.go` 와 `server/internal/auth/config.go` 가 읽는 것 전부입니다. 기동 후에 바뀌지 않으며, 그 밖의 운영 설정은 데이터베이스 에 있고 관리자 콘솔에서 바꿉니다(3.2).

이름	기본값	필수	설명
DATABASE_URL	—	예	PostgreSQL DSN. 없으면 DATABASE_URL (or PTIUM_DATABASE_DSN) is required 로 멈춤
HTTP_ADDR	:8080		수신 주소
PUBLIC_BASE_URL	(없음)		사용자가 여는 주소. OIDC 리다이렉트와 공유 링크에 쓰임
WEB_DIR	이미지에서 /app/web		워크스페이스 번들 위치. 비우면 API 만 제공
CORS_ALLOWED_ORIGINS	http://localhost:3000,http://localhost:5173		허용할 브라우저 출처(선택 구분)
LOG_LEVEL	info		로그 수준
SHUTDOWN_TIMEOUT	15s		종료 시 진행 중 요청을 기다리는 시간
WORKER_POLL_INTERVAL	2s		생성 워커가 큐를 보는 간격
SESSION_LIFETIME	12h		브라우저 세션 수명. 절반이 지나면 다음 요청에서 갱신
KEY_ENCRYPTION_SECRET	(없음)	운영에서 예	민감 설정(AI 키·OIDC 시크릿)의 암호화 키. 없으면 DATABASE_URL 에서 유도하며 경고 로그를 남김
ASSET_STORAGE	database		올린 이미지 위치. database 또는 filesystem (volume · pvc · file · disk 도 같은 뜻)
ASSET_DIR	/var/lib/ptium/assets	filesystem 일 때	이미지 디렉터리. 없거나 쓸 수 없으면 기동을 거부
BOOTSTRAP_ADMIN	(없음)		비밀번호로 로그인하는 로컬 관리자 이메일
BOOTSTRAP_ADMIN_PASSWORD	(없음)	BOOTSTRAP_ADMIN 과 함께	12자 이상. 계정 생성 시 한 번만 읽음
BOOTSTRAP_ADMIN_NAME	Ptium Administrator		그 계정의 표시 이름
BOOTSTRAP_ADMIN_PASSWORD_RESET	false		true 로 한 번 기동하면 비밀번호를 덮어씀 (분실 복구). 끝나면 자을 것
BOOTSTRAP_ADMIN_EMAILS	(없음)		OIDC 로 처음 로그인할 때 관리자로 올릴 이메일(선택 구분)
BOOTSTRAP_ADMIN_SUBJECTS	(없음)		같은 일을 OIDC subject 로
OIDC_ISSUER_URL	(없음)		발급자 URL. 설정하면 OIDC 로그인 활성화. KEYCLOAK_URL + KEYCLOAK_REALM 으로 대신 지정 가능
OIDC_CLIENT_ID	(없음)	OIDC_ISSUER_URL 이 있으면 예	SPA 클라이언트 ID
OIDC_CLIENT_SECRET	(없음)		기밀 클라이언트일 때만. 설정하면 코드 교환을 서버가 대신 함
OIDC_AUDIENCE	(없음)		토큰의 audience 검사(선택 구분). OIDC_AUDIENCES 도 같음
OIDC_ALLOWED_ALGORITHMS	RS256		허용 서명 알고리즘
OIDC_ALLOW_HTTP	false		발급자 URL 에 http:// 허용(평가용)
OIDC_HTTP_TIMEOUT	10s		디스커버리·JWKS 요청 제한 시간
OIDC_JWKS_CACHE_TTL	15m		JWKS 캐시

이름	기본값	필수	설명
OIDC_CLOCK_SKEW	30s		토큰 시각 허용 오차
OIDC_MAX_TOKEN_BYTES	32768		받아들일 토큰 최대 크기
AUTH_ADMIN_ROLES	ptium-admin,admin		관리자로 볼 OIDC 역할. OIDC_ADMIN_ROLES 도 같음
DEV_AUTH_ENABLED	false		개발 인증(헤더 하나로 로그인). 운영에서는 끄
DEV_AUTH_SECRET	(없음)	개발 인증이 켜지면 예	32자 이상. DEV_AUTH_TOKEN 도 같음
DEV_AUTH_HEADER	X-Ptium-Dev-Secret		그 값을 담은 헤더 이름
DEV_AUTH_EMAIL · DEV_AUTH_NAME · DEV_AUTH_SUBJECT	(없음)		개발 계정의 정체
DEV_AUTH_ROLES · DEV_AUTH_SCOPES	(없음)		개발 계정의 역할과 API 범위
DEV_AUTH_ALLOW_REMOTE	false		루프백 밖에서 온 개발 인증 요청 허용

PTIUM_* 로 시작하는 옛 이름(PTIUM_DATABASE_DSN , PTIUM_OIDC_ISSUER_URL 등)도 같은 자리에 읽히지만 새 배포에서는 위 이름을 쓰세요.

APP_ENV · POSTGRES_* 는 compose 파일이 쓰는 값이고 서버는 읽지 않습니다.

3.2 서비스 설정 (관리자 콘솔)

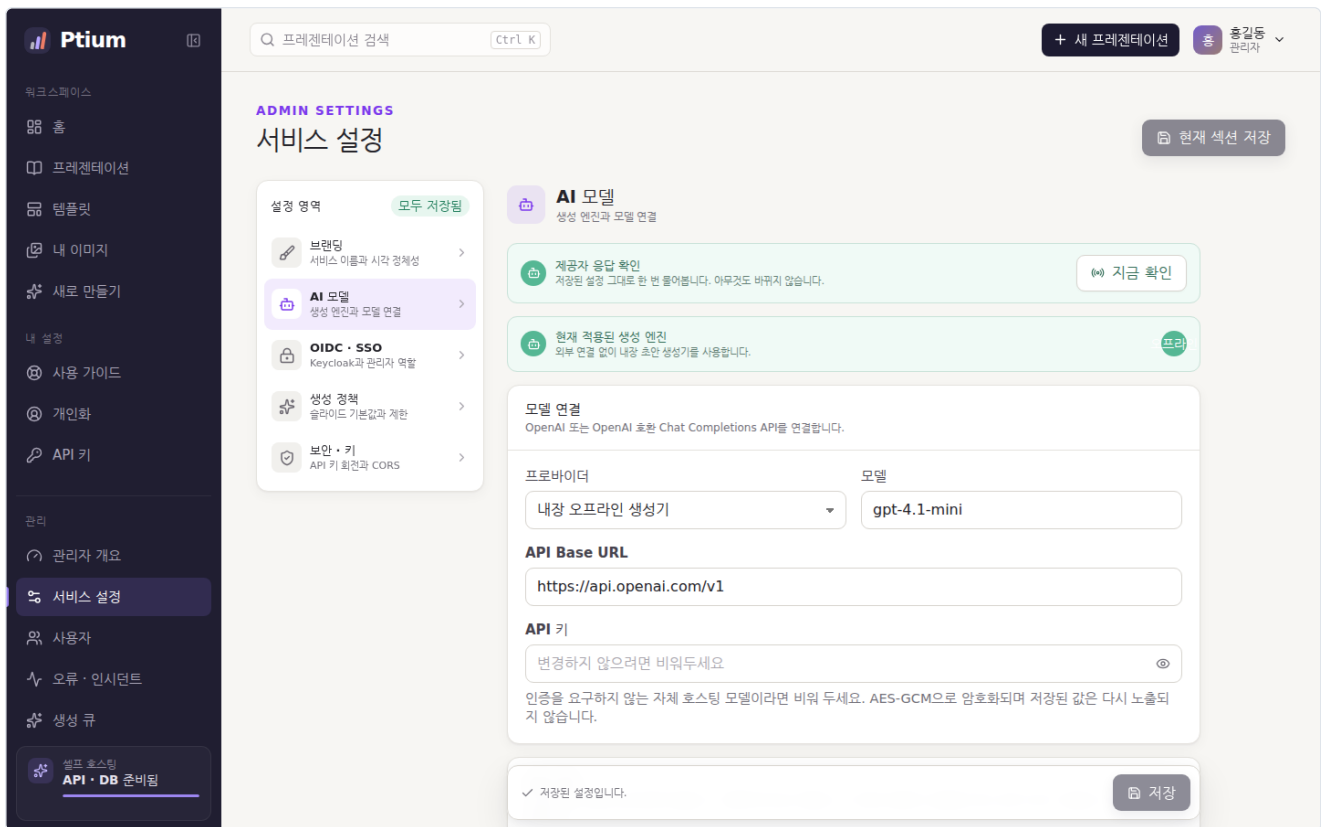
관리 → 서비스 설정 (/admin/settings)의 다섯 영역이 데이터베이스에 저장되는 운영 설정입니다. 저장은 PUT /api/v1/admin/settings/{key} 또는 영역 단위 PATCH /api/v1/admin/settings 로 이뤄지며, 누가 무엇을 바꿨는지는 감사 기록과 설정 화면의 변경 이력에 남고 되돌릴 수 있습니다 (POST /api/v1/admin/settings/changes/{id}/revert).

The screenshot shows the Ptium Admin Settings page. The left sidebar contains navigation links for various settings. The main content area is titled 'ADMIN SETTINGS' and '서비스 설정'. The 'Branding' section is active, showing fields for '서비스 정체성' (Service Identity), '서비스 이름' (Service Name), '로그 URL' (Login URL), and '대표 색상' (Representative Color). The '서비스 이름' field is set to 'Ptium'. The '로그 URL' field is set to 'https://...'. The '대표 색상' field is set to '#7C3AED'. Below these fields, there is a section for '최근 설정 변경' (Recent Settings Changes) showing a list of changes with a '되돌리기' (Revert) button. The bottom right corner has a '저장' (Save) button.

서비스 설정 — 브랜딩 영역. 왼쪽에서 영역을 고르고 오른쪽에서 저장한다

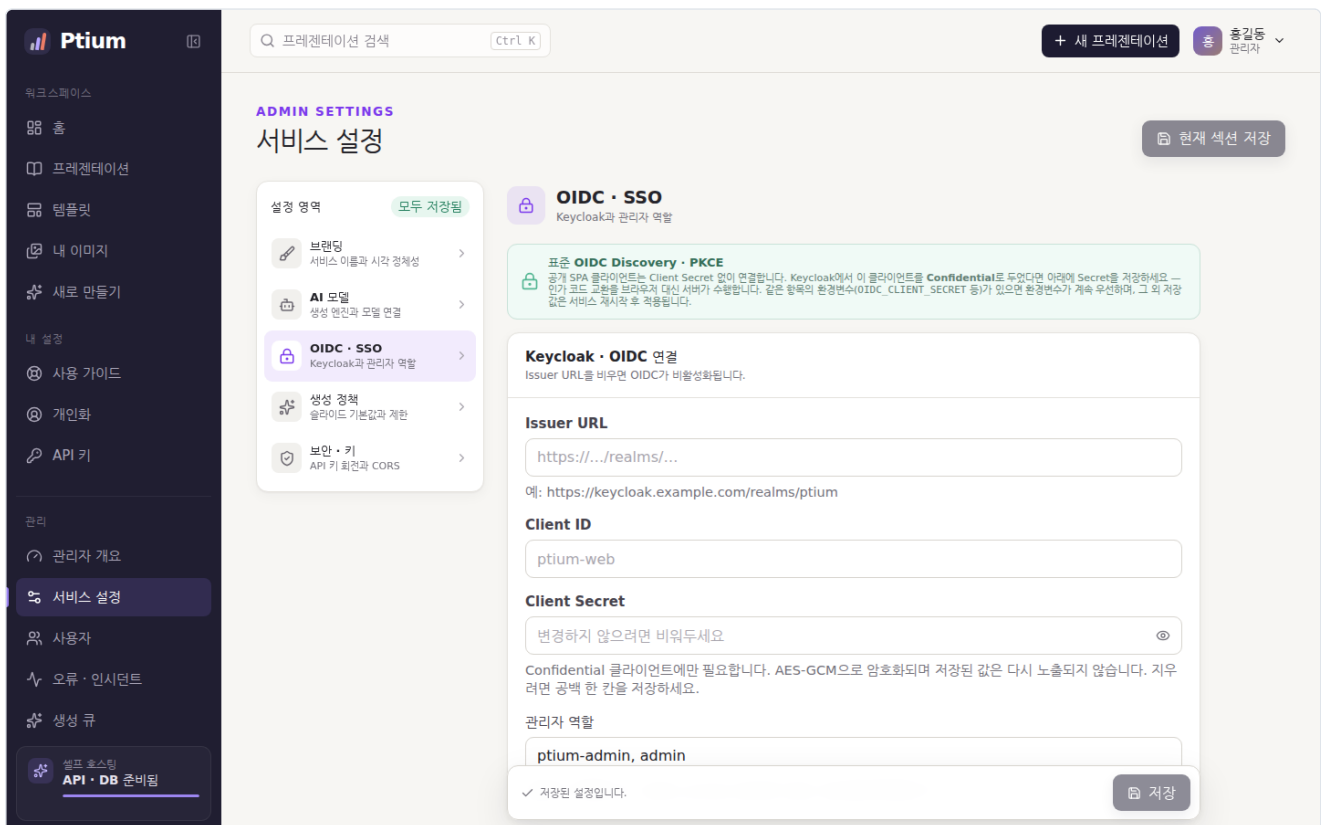
영역	키	기본값	뜻
브랜딩	branding.product_name	Ptium	로그인 후 화면과 브라우저 제목
	branding.logo_url	(없음)	로고 URL. 비우면 기본 마크
	branding.brand_color	#7C3AED	워크스페이스 대표 색. 사용자가 색을 고르지 않은 새 텍에도 적용
AI 모델	ai.provider	fallback	fallback (내장 작성기) 또는 openai-compatible
	ai.base_url	https://api.openai.com/v1	Chat Completions 호환 주소. vLLM·Ollama·llama.cpp 도 됨
	ai.model	gpt-4.1-mini	모델 이름
	ai.api_key	(없음)	제공자 키. AES-GCM 으로 암호화 저장, 다시 표시되지 않음. 인증 없는 자체 호스팅이면 비움
	ai.reasoning	auto	사고(thinking) 요청 방식: auto · off · on
	ai.max_output_tokens	8000	한 번에 요청할 최대 출력 토큰
	ai.timeout_seconds	300	한 번의 완성을 기다리는 시간
	ai.max_prompt_tokens	128000	한 번에 요청할 최대 프롬프트 토큰
OIDC · SSO	auth.oidc.issuer_url · auth.oidc.client_id · auth.oidc.client_secret	(없음)	환경 변수가 있으면 재시작 전까지 환경 변수가 우선
	auth.oidc.admin_roles	["ptium-admin", "admin"]	관리자로 볼 역할
생성 정책	generation.default_slide_count	10	기본 장수
	generation.max_slides	50	최대 장수
	generation.default_theme	slate-classic	기본 디자인
	generation.default_lang · default_tone · default_audience	ko · professional · general	기본 언어·톤·청중
	generation.outline_pass	true	문안 전에 개요를 먼저 짜기
	generation.repair_passes	3	넘친 슬라이드를 모델에 되돌려 고치는 횟수. 0 이면 끄
	generation.max_template_mb	32	업로드 한도 (MiB). 최대 64
	generation.allow_user_uploads	true	사용자가 자기 PowerPoint 템플릿을 올릴 수 있는지
보안 · 키	security.api_key_grace	24h	API 키 회전 시 이전 키가 함께 유효한 기간
	security.cors_origins	[]	추가로 허용할 브라우저 출처. 바꾸면 재시작 필요

AI 모델 연결. ai.provider 를 openai-compatible 로 바꾸고 주소·모델을 넣은 뒤 **지금 확인**을 누르면 저장된 설정 그대로 제공자에게 한 번 물어 응답 여부와 걸린 시간을 보여 줍니다 — 아무것도 바꾸지 않습니다. 화면 위 **현재 적용된 생성 엔진**이 지금 텍을 실제로 쓰는 작성기입니다.

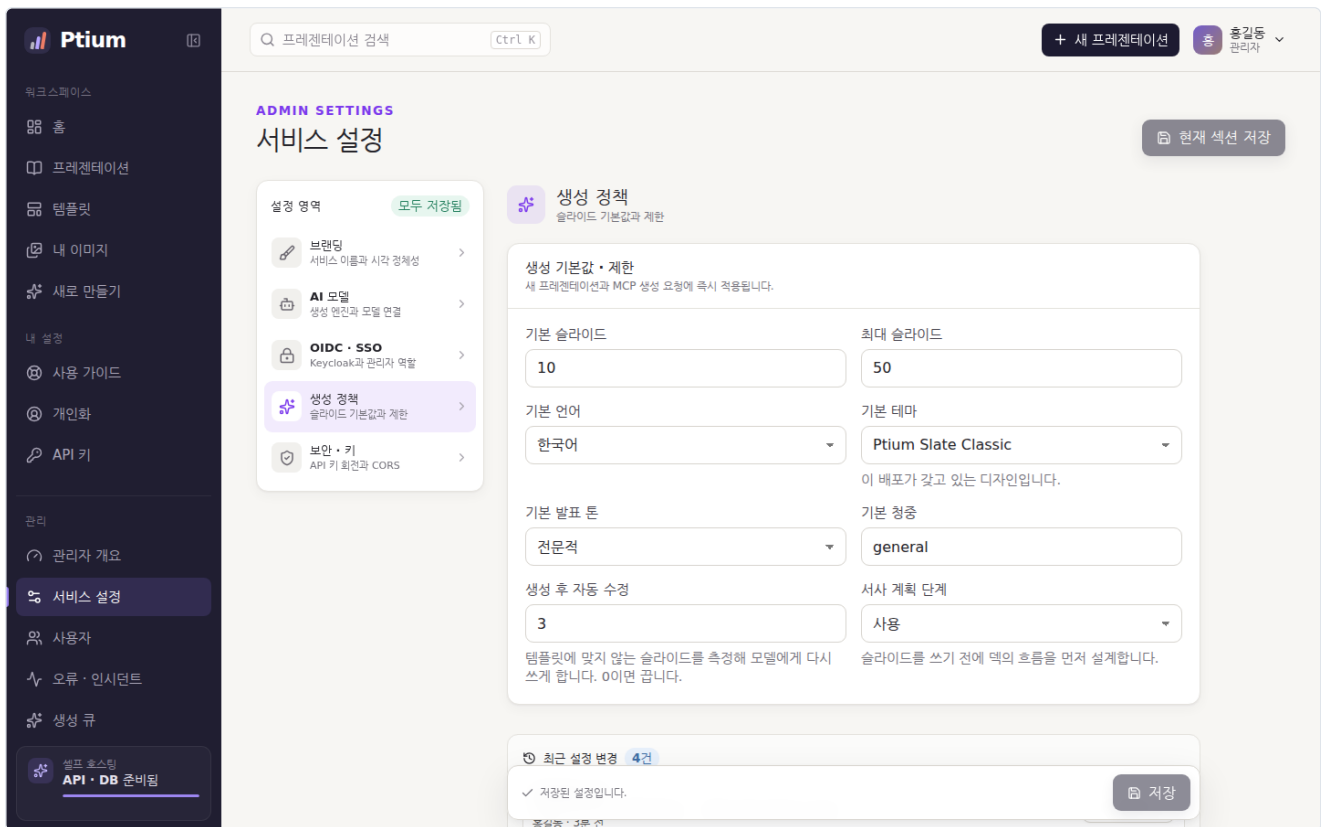


서비스 설정 — AI 모델, 제공자·모델·주소·키와 응답 확인

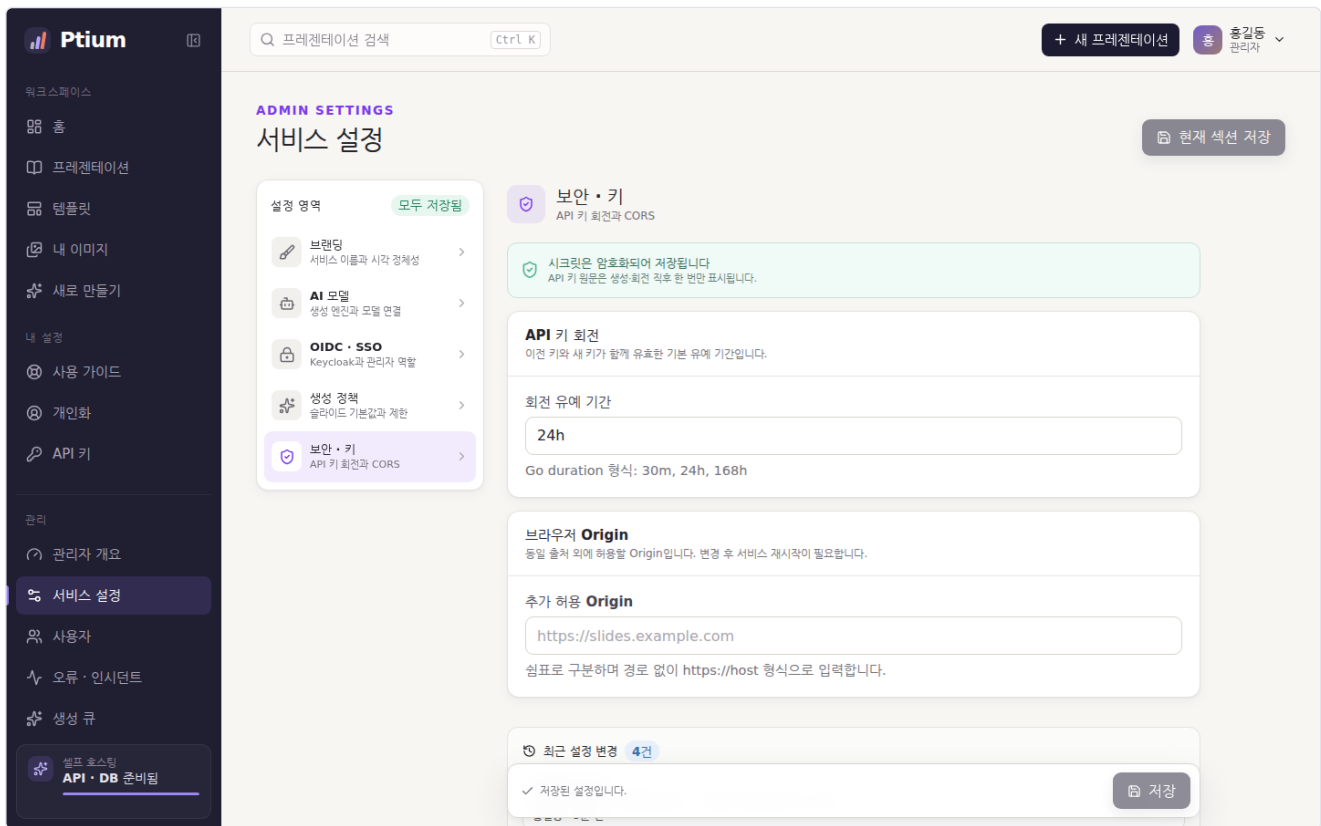
OIDC · SSO, Keycloak 이라면 realm 발급자(<https://sso.internal/realms/company>)와 SPA 클라이언트 ID 를 넣습니다. Keycloak 쪽에는 Ptium 의 정확한 origin 과 리다이렉트 URI (`PUBLIC_BASE_URL + /auth/callback`)를 허용해야 합니다. 기밀 클라이언트면 시크릿을 넣고, 공개 클라이언트(PKCE)면 비워 둡니다.



서비스 설정 — OIDC · SSO



서비스 설정 — 생성 정책



서비스 설정 — 보안 · 키

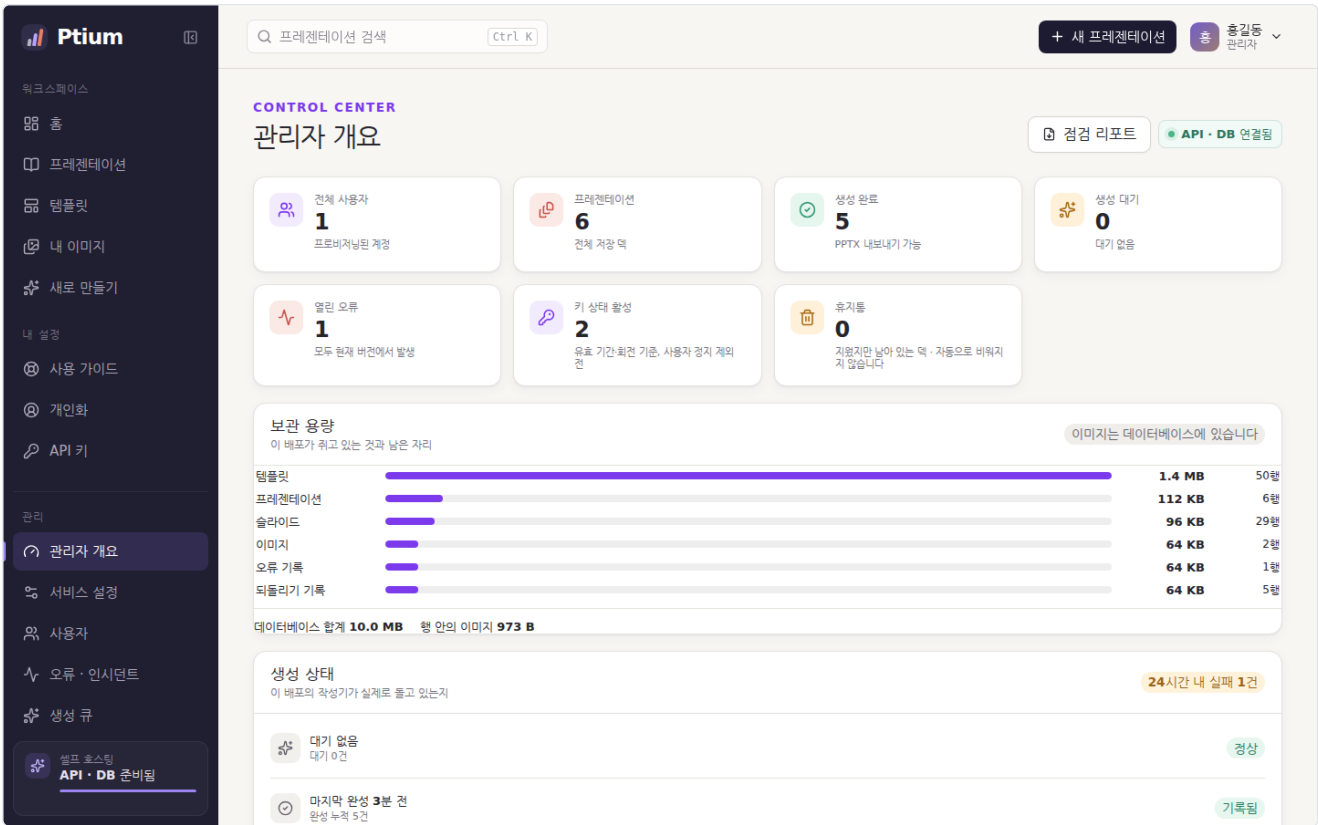
4. 계정과 권한

역할은 두 가지입니다.

5. 운영

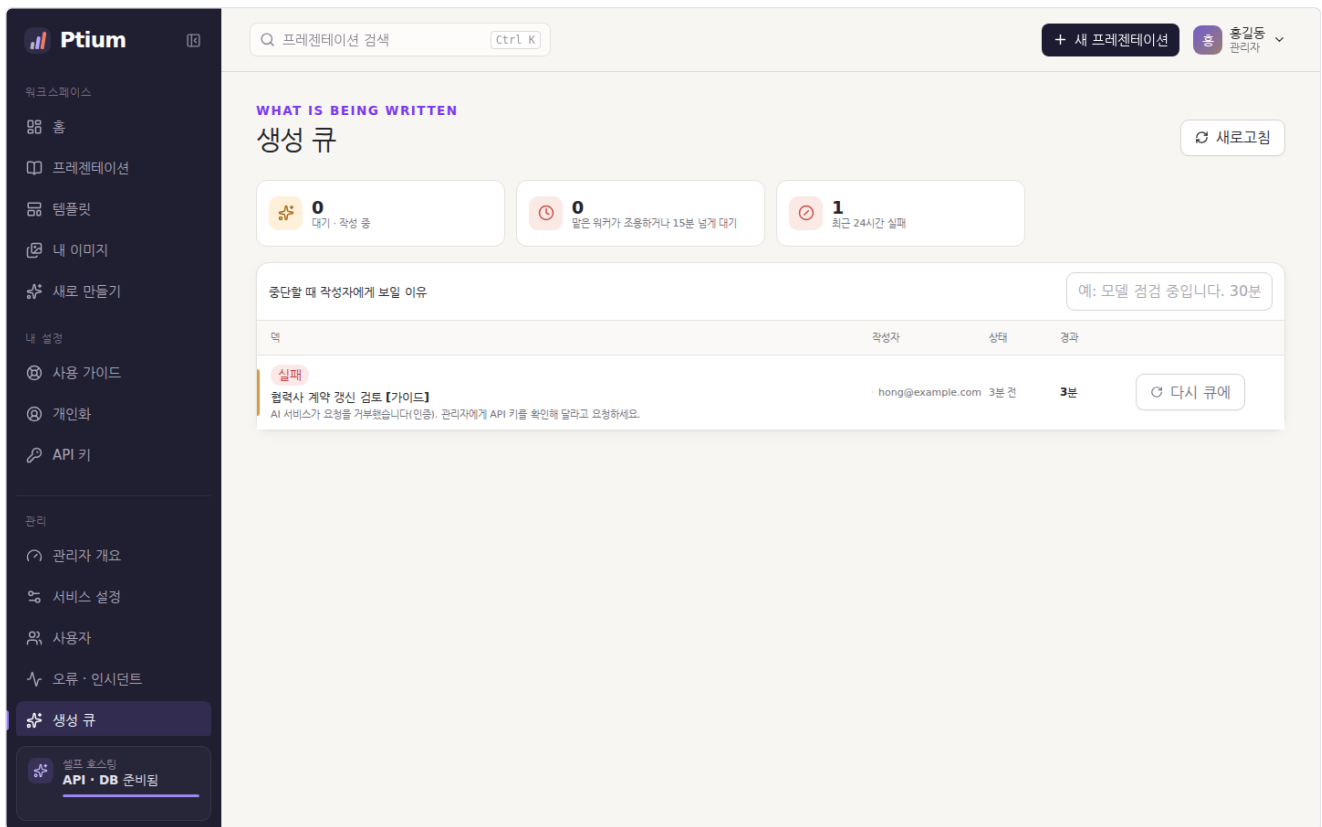
5.1 상태 점검

무엇	어디
프로세스가 살아 있나	GET /healthz
데이터베이스까지 준비됐나	GET /readyz → {"data":{"status":"ready"}}
한 화면 요약	관리 → 관리자 개요 (/admin): 사용자·텍·생성 대기·열린 오류·키·휴지통, 표별 보관 용량, 생성 상태
점검 리포트	관리자 개요의 점검 리포트 버튼 (GET /api/v1/admin/report)



관리자 개요 — 이 배포가 취고 있는 것과 생성 상태를 한 화면에

사용 현황 (/admin/usage)은 7·14·30·90일 동안 만든 텍·실패·가장 오래 걸린 텍을, 누가 어떤 디자인으로 만들었는지와 함께 보여 줍니다.



생성 큐 — 모델이 키를 거부해 실패한 텍 한 건. 이유가 작성자에게 보이는 문장 그대로 붙어 있다

5.3 로그

로그는 표준 출력으로 나가는 JSON 한 줄씩입니다(`docker logs ptium / kubectl logs deploy/ptium`). 모든 HTTP 요청은 `"msg": "http request"` 로 `request_id · method · path · status · duration_ms` 를 남기고, 클라이언트가 받은 응답의 `requestId` 와 오류 센터의 기록이 같은 값을 가리킵니다. 사용자가 오류를 신고하면 그 ID 를 받아 로그와 오류 · 인시던트에서 찾으면 됩니다. 인증 헤더 · 쿠키 · 비밀번호 · 토큰 · 제공자 키는 로그와 오류 기록에서 가려집니다.

5.4 백업과 복구

백업 대상은 데이터베이스 하나입니다. `ASSET_STORAGE=filesystem` 이면 `ASSET_DIR` 볼륨도 함께 받습니다.

```
pg_dump --format=custom --file=ptium-$(date +%F).dump "$DATABASE_URL"
# ASSET_STORAGE=filesystem 일 때
docker run --rm -v ptium_ptium-assets:/assets -v "$PWD":/backup alpine \
  tar czf /backup/ptium-assets-$(date +%F).tar.gz -C /assets .
```

복구는 빈 데이터베이스에 `pg_restore` 한 뒤 같은 버전(또는 더 새 버전)의 이미지를 띄우는 것입니다. 볼륨을 되살리지 않으면 이미지 요청이 410 과 "this image's file is missing from the image storage volume" 으로 답합니다 — 이미지가 있는 척하지 않습니다.

5.5 업그레이드

1. 데이터베이스(와 볼륨)를 백업합니다.
2. 새 아카이브를 `gzip -dc ... | docker load` 로 반입합니다.
3. `.env` 의 `PTIUM_VERSION` (또는 매니페스트의 이미지 태그)을 새 버전으로 바꿉니다.
4. `docker compose --env-file .env -f docker-compose.ptium-<new>.yaml up -d` 또는 `kubectl rollout restart deploy/ptium`.

마이그레이션은 기동 중에 앞으로만 적용되며 복제본 여럿이 동시에 돌려도 안전합니다. 내장 디자인은 기동마다 코드에서 다시 만들어지므로 새 릴리즈의 디자인 변경은 별도 단계 없이 반영됩니다. 오류 기록에는 그 오류를 본 빌드가 함께 남아, 업그레이드 뒤 열린 인시던트가 지금 버전의 것인지 알 수 있습니다.

되돌리기. 이전 태그로 다시 배포하면 됩니다. 어떤 마이그레이션도 데이터를 지우지 않아 옛 이미지가 새 이미지가 쓴 데이터베이스 위에서 기동하고 모든 텍이 열립니다. 옛 이미지가 모르는 것(그 뒤에 생긴 문법 · 기능)은 그대로 두거나 무시하지만 잃지는 않습니다. 데이터베이스 백업 복원은 마이그레이션 자체가 실패한 경우에만 필요합니다.

5.6 정리

관리 → 정리할 것 (/admin/tidy)은 휴지통의 텍, 실패로 남은 텍, 손대지 않은 초안, 기한 지남 공유 링크, 쓰이지 않는 이미지, 되돌리기 판본, 감사 기록이 얼마나 쌓였는지 보여 줍니다. 이 화면은 아무것도 지우지 않습니다 — 얼마나 오래 돌지는 배포가 정할 일이고, 자동 삭제는 없습니다.

Ptium

워크스페이스

홈

프레젠테이션

템플릿

내 이미지

새로 만들기

내 설정

사용 가이드

개인화

API 키

관리

관리자 개요

서비스 설정

사용자

오류 · 인시던트

생성 큐

선택 호스팅
API · DB 준비됨

Q 프레젠테이션 검색

Ctrl K

+ 새 프레젠테이션

출결동 관리자

WHAT HAS PILED UP

정리할 것

새로고침

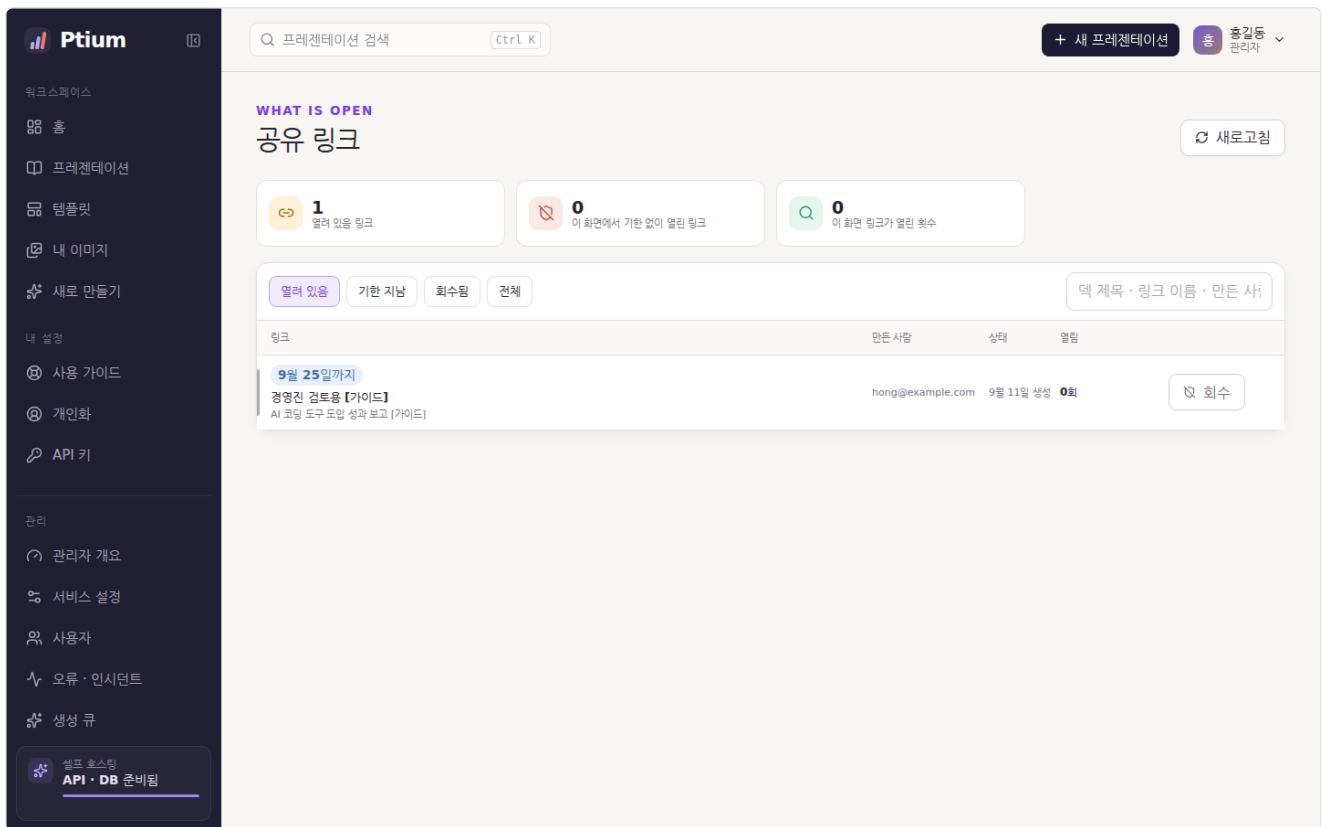
이 화면은 아무것도 지우지 않습니다.

이 화면은 아무것도 지우지 않습니다. 무엇을 얼마나 오래 두느냐는 정해야 하는 일이고, 무엇이 쌓였는지 모르면 정할 수 없습니다. 아래 숫자는 그 판단을 위한 것입니다. 자동 삭제는 없습니다.

휴지통에 있는 텍 작성자가 지운 텍입니다. 아무것도 자동으로 지워지지 않으므로 그대로 남아 있습니다. 없습니다
30일 넘게 실패로 남은 텍 생성에 실패한 뒤 아무도 다시 시도하지 않은 텍입니다. 없습니다
90일 넘게 손대지 않은 초안 슬라이드가 만들어지지 않은 채 남은 초안입니다. 없습니다
기한이 지난 공유 링크 이미 열리지 않는 링크입니다. 목록에는 남아 있습니다. 없습니다
어느 텍도 쓰지 않는 이미지 오늘 불러 아직 남지 않은 이미지도 여기 포함됩니다. 지를 대상으로 보기 전에 아래 항목을 보세요. 가장 오래된 것 2026년 9월 11일 2개1KB
한 달 넘게 쓰이지 않는 이미지 올린 지 30일이 지나도록 어느 텍에도 불러가지 않은 이미지입니다. 없습니다
텍을 고칠 때마다 쌓인 판본 되돌리기가 있고 있는 기록입니다. 텍을 고칠 때마다 한 판본씩 늘어납니다. 지우면 그만큼 되돌릴 수 없습니다. 가장 오래된 것 2026년 9월 11일 5개64KB
감사 기록 누가 무엇을 했는지 남긴 기록입니다. 줄수록 늘어납니다. 얼마나 오래 보관할지는 이 배포가 정할 일입니다. 가장 오래된 것 2026년 9월 11일 17개32KB

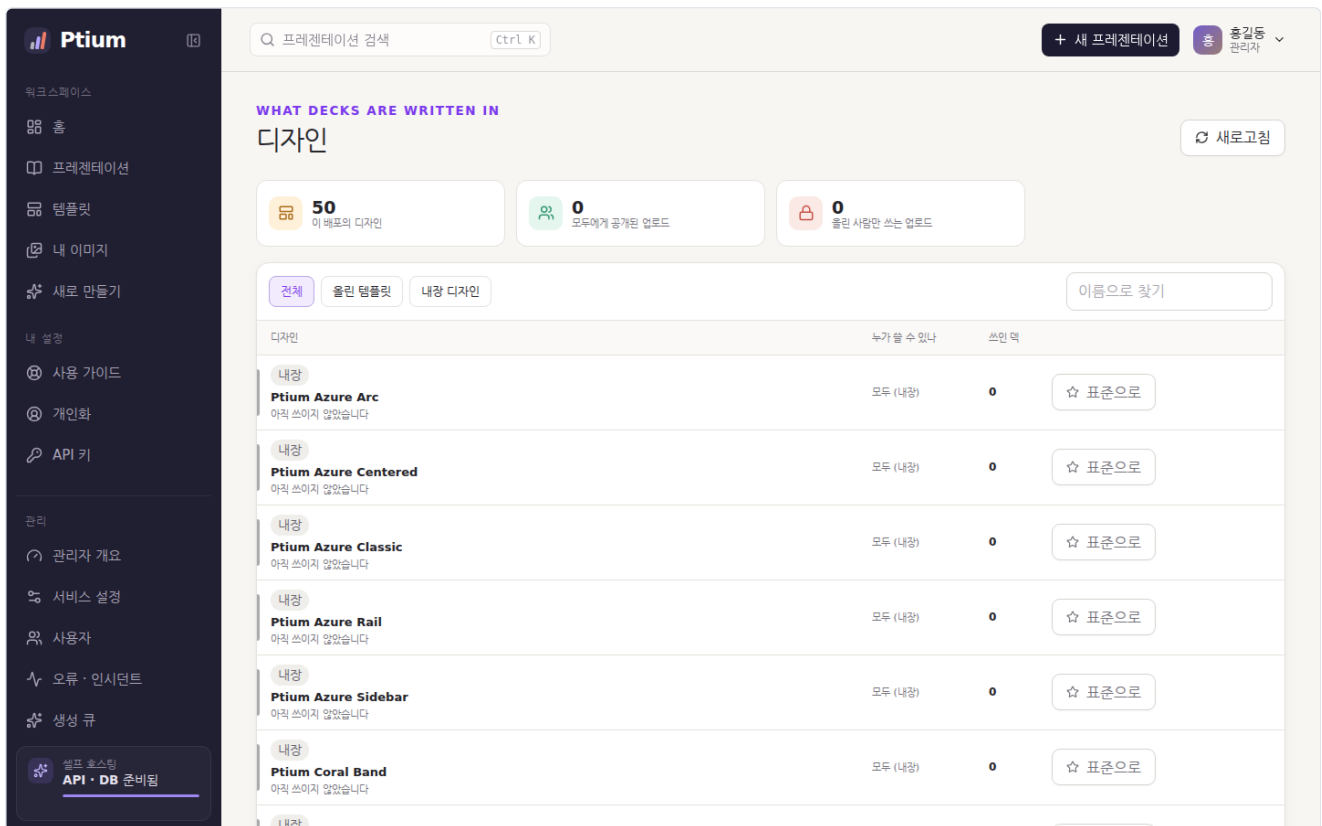
정리할 것 — 무엇이 얼마나 쌓였는지, 지우지는 않는다

공유 링크 (/admin/shares)에서는 배포 전체의 공유 링크를 열려 있음 · 기한 지남 · 회수됨 · 전체로 나눠 보고, 어느 텍의 링크를 누가 언제 만들어 몇 번 열렸는지 확인한 뒤 회수로 답습니다 (POST /api/v1/admin/shares/{id}/close). 회수한 주소는 그 즉시 열리지 않습니다. 위 칸의 이 화면에서 기한 없이 열린 링크는 만료 일 없이 만든 링크의 수로, 정리 대상을 고를 때 먼저 봅니다.



공유 링크 — 배포 전체의 링크를 상태별로 보고 회수한다

디자인 (/admin/designs)에서는 내장 50종과 사용자가 올린 템플릿을 보고 하나를 **표준**으로 지정하거나 올린 템플릿을 모두에게 공개합니다.

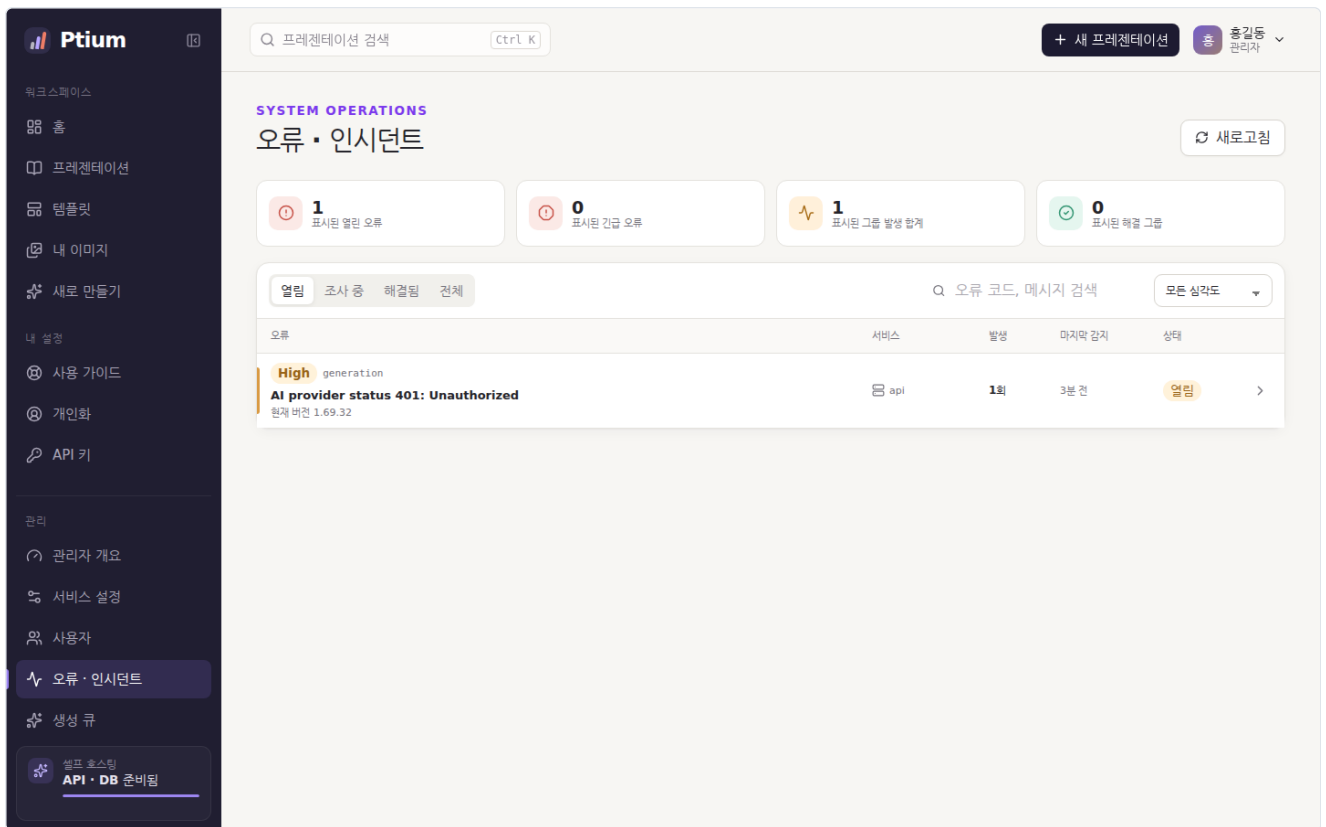


디자인 — 이 배포의 디자인과 어느 템이 쓰는지, 표준 지정

6. 장애 대응

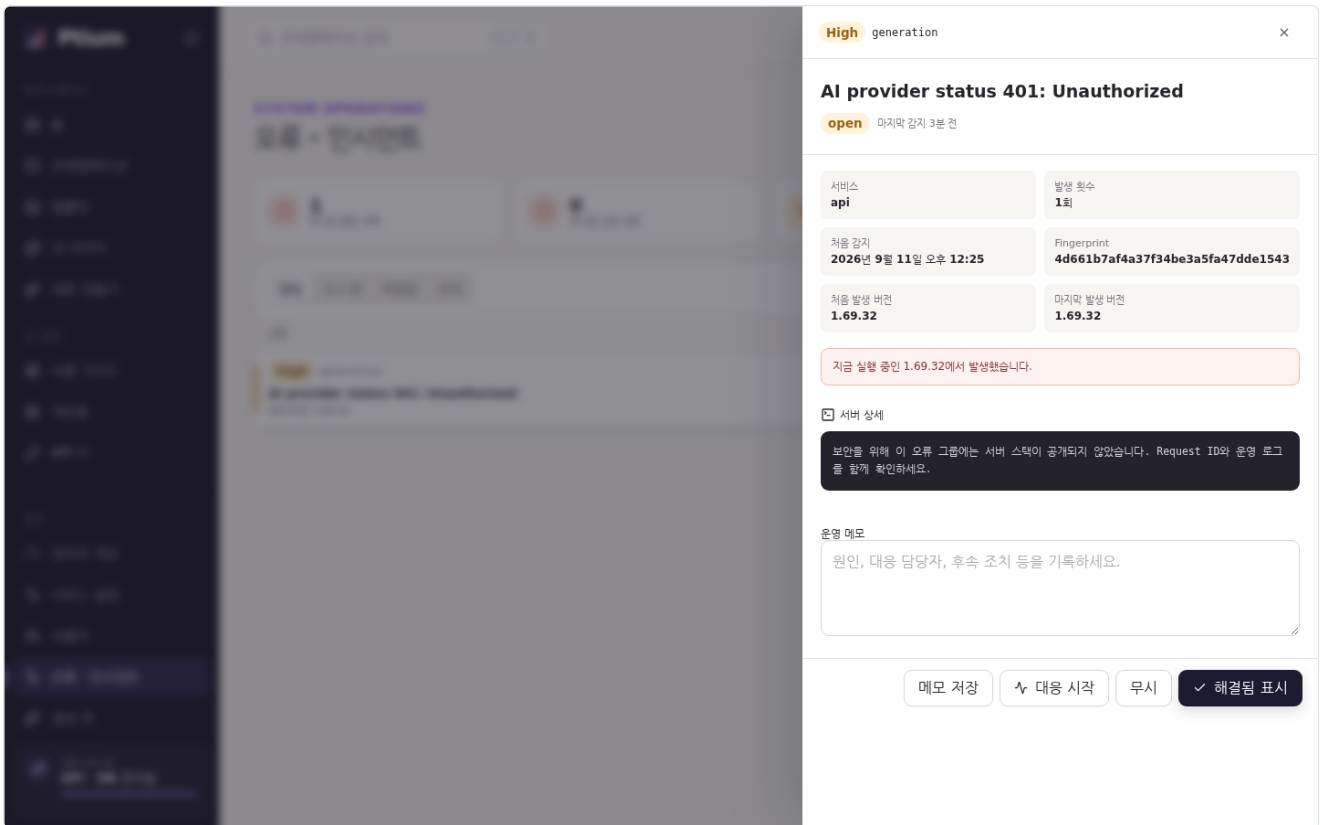
증상	확인할 곳	조치
컨테이너가 바로 종료	로그에 DATABASE_URL (or PTIUM_DATABASE_DSN) is required	.env 에 DATABASE_URL 을 넣었는지, compose 가 그 .env 를 읽는지(--env-file)
컨테이너가 바로 종료	로그에 auth config: DEV_AUTH_SECRET must contain at least 32 characters when development auth is enabled	DEV_AUTH_ENABLED=false 로 끄거나 32자 이상 시크릿을 넣음
컨테이너가 바로 종료	로그에 OIDC_CLIENT_ID is required when OIDC_ISSUER_URL is set 또는 URL must use HTTPS (HTTP requires OIDC_ALLOW_HTTP=true)	클라이언트 ID 를 넣거나, 평가 환경에서만 OIDC_ALLOW_HTTP=true
컨테이너가 바로 종료	로그에 image directory ... is not writable 또는 ... is not a directory	ASSET_DIR 볼륨이 마운트되고 uid/gid 65532 가 쓸 수 있는 지. 볼륨이 깨졌으면 첫 업로드가 아니라 기동이 멈추는 것이 의도된 동작
아무도 로그인할 수 없음	로그에 no interactive authentication is configured; set BOOTSTRAP_ADMIN and BOOTSTRAP_ADMIN_PASSWORD, or configure OIDC, before anyone can sign in	둘 중 하나를 설정하고 재시작
관리자 비밀번호 분실	—	BOOTSTRAP_ADMIN_PASSWORD 를 새 값으로, BOOTSTRAP_ADMIN_PASSWORD_RESET=true 로 한 번 기동. 로그에 bootstrap administrator ready . 끝나면 두 변수를 원래대로
비밀번호를 바꿨는데 재시작 후 그대로	로그에 the bootstrap administrator already has a password; set BOOTSTRAP_ADMIN_PASSWORD_RESET=true to overwrite it	정상. 환경 변수는 계정 생성 시 한 번만 읽힘
로그인에 Too many sign-in attempts. Try again shortly.	—	같은 주소에서 실패가 반복돼 지연이 걸린 것(2초에서 시작해 최대 5분). 기다리면 풀림
기동 시 KEY_ENCRYPTION_SECRET is unset; sensitive settings are encrypted with a key derived from DATABASE_URL ...	—	경고. 운영에서는 KEY_ENCRYPTION_SECRET 을 고정값으로 두고, DB 자격 증명을 바꾸기 전에 설정
생성이 실패로 끝남	관리 → 생성 큐(/admin/queue)와 오류 · 인시던트(/admin/errors); 로그 generation worker iteration failed	AI 모델 설정의 지금 확인으로 제공자 응답 확인. ai.timeout_seconds 가 모델에 비해 짧은지. 큐 화면의 사유 칸에 적어 두면 작성자에게 보임
덱이 "생성 중" 에서 멈춤	생성 큐의 "말은 워커가 조용하거나 15분 넘게 대기"	워커가 죽은 것. 재시작하면 다른 워커가 이어 받음(로그 this deck is no longer ours to write 는 정상 인계)
업로드가 413 또는 인그레스에서 끊김	인그레스 본문 한도	generation.max_template_mb 이상으로 올림(번들 매니페스트 proxy-body-size: 64m)
503 printing_busy · templates_busy · import_busy	로그 http request 의 status 503	동시 문서 작업이 예산을 넘은 것. 정상 보호 동작이며 잠시 뒤 재시도. 잦으면 메모리 한도와 복제본 수를 늘림
이미지가 410	로그 메시지 "this image's file is missing from the image storage volume"	ASSET_DIR 볼륨이 빠졌거나 복원되지 않음. 볼륨을 되살리거나 사용자가 다시 올림
사용자가 requestId 를 들고 옴	오류 · 인시던트에서 검색, 로그에서 request_id	인시던트를 조사 중 → 해결됨으로 옮기며 처리. 같은 원인은 지문으로 묶여 한 건으로 보임

오류 · 인시던트(/admin/errors)는 같은 원인을 지문(fingerprint)으로 묶어 한 줄로 보여 줍니다. 아래는 모델이 키를 거부해 생성이 실패한 뒤의 화면입니다 — 생성 큐에는 실패한 덱이, 여기에는 그 원인이 남고, 둘은 같은 사건입니다.



오류 · 인시던트 — 열림·조사 중·해결됨으로 나누어 처리한다. 모델의 401 거부한 건 열려 있다

줄을 누르면 서비스·발생 횟수·처음 감지·지문·처음과 마지막 발생 버전이 보이고, 지금 실행 중인 빌드에서 난 것인지 알려 줍니다. 요청에서 난 오류라면 Request ID 가 함께 있어 로그의 request_id 와 맞춰 봅니다. 운영 메모에 원인과 담당자를 적고 대응 시작(조사 중) → 해결됨 표시로 옮기며, 알고 있는 원인은 무시로 목록에서 내립니다(PATCH /api/v1/admin/errors/{id} 에 status · notes .API의 status 값은 open · acknowledged · resolved · ignored 이고, 화면의 조사 중이 acknowledged 입니다).



인시던트 상세 — 발생 버전·지문·운영 메모와 상태를 옮기는 버튼

7. 보안

바뀌어야 하는 기본값

- `DEV_AUTH_ENABLED=false` 인지 확인합니다. 개발 인증은 헤더 하나로 관리자가 되는 문이며, 폐쇄망 평가 호스트 밖에서는 절대 켜지 않습니다. 저장소의 `docker-compose.yml` (개발용)은 이것을 켜 두므로 운영에는 릴리즈 번들의 `docker-compose.ptium-<version>.yml` 을 씁니다.
- `KEY_ENCRYPTION_SECRET` 을 고정값으로 둡니다. 없으면 `DATABASE_URL` 에서 유도하므로 DB 비밀번호를 바꾸는 순간 저장된 AI 키·OIDC 시크릿을 풀 수 없게 됩니다.
- 첫 로그인 뒤 `BOOTSTRAP_ADMIN_PASSWORD` 를 바꾸고, OIDC 가 자리 잡으면 환경에서 지웁니다. `BOOTSTRAP_ADMIN_EMAILS` · `BOOTSTRAP_ADMIN_SUBJECTS` 도 관리자가 정해진 뒤에는 지웁니다.
- `PUBLIC_BASE_URL` 은 HTTPS 로, `CORS_ALLOWED_ORIGINS` 는 실제 출처만.
- PostgreSQL 은 Ptium 전용 계정과 `sslmode=require` , 예시 비밀번호는 전부 교체.

밖에 열면 안 되는 것

- PostgreSQL 포트. Ptium 컨테이너만 닿으면 됩니다.
- `/mcp` 와 `/api/v1` 은 워크스페이스와 같은 인그레스 정책으로 보호합니다. MCP 는 우회 통로가 아니라 같은 권한 계층을 지나는 또 하나의 입구입니다.

인증 연동

- OIDC 는 발급자 디스커버리와 JWKS 로 토큰을 검증하고, 발급자·audience·서명 알고리즘 (RS256) 을 확인합니다. Keycloak 에는 Ptium origin 과 리다이렉트 URI 를 정확히 등록하고, 공개 SPA 클라이언트는 Authorization Code + PKCE 를 씁니다.
- 브라우저 세션은 `ptium_session` 쿠키 (HttpOnly · SameSite=Lax · TLS 면 Secure) 에 있고, 교차 사이트 쓰기 요청은 거절됩니다. 비밀번호를 바꾸면 그 전에 발급된 세션은 모두 무효가 됩니다.
- API 키는 점두사와 해시만 저장하고 전체 값은 생성 직후 한 번만 보입니다. 회전하면 `security.api_key_grace` (기본 24h) 동안 이전 키가 함께 살아 있어 무종단 교체가 됩니다. 폐기는 즉시입니다.
- AI 제공자 키와 OIDC 시크릿은 관리자 설정 엔드포인트로만 받고, 읽기는 `configured` 표시만 돌려줍니다. 로그·인시던트·오류 응답은 인증 헤더·쿠키·비밀번호·토큰·키를 가립니다.

감사. 관리 → 감사 기록 (`/admin/audit`) 에 누가 무엇을 했는지 남습니다. 설정 변경은 설정 화면의 변경 이력에서 되돌릴 수 있습니다.

Ptium

워크스페이스

홈

프레젠테이션

템플릿

내 이미지

새로 만들기

내 설정

사용 가이드

개인화

API 키

관리

관리자 개요

서비스 설정

사용자

오류 · 인시던트

생성 큐

선택 중

API · DB 준비됨

Q 프레젠테이션 검색

Ctrl K

+ 새 프레젠테이션

홍길동 관리자

WHO DID WHAT

감사 기록

CSV로 내려받기

새로고침

오늘7일30일1년

Q 행위자, 대상, 기록 내용 검색

모든 동작

동작	행위자	대상	시각
설정 일괄 변경 settings.update_batch settings	hong@example.com	settings	3분 전
settings.change setting ai.base_	hong@example.com	setting	3분 전
settings.change setting ai.provi	hong@example.com	setting	3분 전
텍 생성 + 생성 요청 presentation.create_and_generate presentation 8e9a8c4e	hong@example.com	presentation	3분 전
설정 일괄 변경 settings.update_batch settings	hong@example.com	settings	3분 전
settings.change setting ai.base_	hong@example.com	setting	3분 전
settings.change setting ai.provi	hong@example.com	setting	3분 전
share.create share.create	hong@example.com	share	3분 전

감사 기록 — 누가 무엇을 언제 했는지