

ReSSO

관리자 가이드

오프라인 설치부터 백업·장애 대응까지, 서비스를 지키는 사람을 위한 안내

버전	v0.9.77
작성일	2026-09-10
문서	관리자 가이드

ReSSO 관리자 가이드

이 문서는 ReSSO를 설치하고 지키는 사람을 위한 것입니다. 화면을 쓰는 사람에게는 [사용자 가이드](#)를 안내하세요.

- 대상 버전: v0.9.77
- 화면 캡처는 모두 v0.9.77을 실제로 띄워 찍었으며, 등장하는 이름·이메일·주소는 모두 가짜입니다.
- 더 깊은 운영 절차(키 회전, 승인 프로세스, 세션 수명, 지표 해석)는 [운영 가이드](#)에, OIDC 사양 준수 범위는 [호환성 문서](#)에, LDAP 세부 설정은 [User Federation 문서](#)에 있습니다. 여기서는 그쪽을 가리키고 겹쳐 쓰지 않습니다.

1. 구성 요소

구성 요소	형태	역할	필수
resso	단일 Docker 컨테이너 (<code>resso:vX.Y.Z</code>)	OIDC Endpoint, 관리·개인 REST API, MCP, React 콘솔 을 모두 한 바이너리에서 제공	필수
PostgreSQL	외부 서비스	사용자·Client·Session·Refresh Token digest·감사 이벤 트·암호화된 Signing Private Key 저장	필수
Reverse Proxy	외부 서비스	TLS 종료. 운영에서는 이 뒤에 둥니다	운영 권장
LDAP / Active Directory	외부 서비스	User Federation을 쓸 때만	선택
Prometheus	외부 서비스	<code>/metrics</code> 수집	선택

주고받는 것:

방향	경로	내용
애플리케이션(RP) → ReSSO	/realms/{realm}/protocol/openid-connect/*	Authorization, Token, UserInfo, JWKS, Logout, Introspection, Revocation
ReSSO → 애플리케이션	Client에 등록한 Back-Channel Logout URI	세션 종료 시 서명된 logout_token POST
브라우저 → ReSSO	/, /api/v1/*, /api/admin/v1/*	콘솔과 그 API
도구·자동화 → ReSSO	/mcp, /api/openapi.json	개인 API 키(Bearer)로 읽기 전용 접근
ReSSO → PostgreSQL	POSTGRES_DSN	모든 영속 상태
ReSSO → LDAP/AD	연결 URL	로그인 시 검증과 주기 동기화

컨테이너는 상태를 갖지 않습니다 — 볼륨이 필요 없고, 읽기 전용 루트 파일시스템(`read_only: true`)과 `/tmp tmpfs` 16MB로 동작합니다. 백업 대상은 PostgreSQL과 Keyring 값 두 가지뿐입니다.

2. 설치

오프라인망 기준 절차입니다. 온라인이라면 2-1을 건너뛰고 이미지를 그대로 받으면 됩니다.

2-1. 이미지 반입

GitHub Release에서 `resso-v0.9.77.tar.gz` 와 `release-sha256.txt` 를 함께 내려받아 옮깁니다. 공식 오프라인 아카이브는 `linux/amd64` 전용입니다.

```
./scripts/offline-load.sh resso-v0.9.77.tar.gz
docker image inspect resso:v0.9.77
```

체크섬 파일이 없으면 반입을 거절합니다 — 이동식 매체로 들어온 아카이브야말로 확인할 값어치가 있기 때문입니다. 파일을 다른 곳에 두었다면 두 번째 인자로 경로를 넘기고, 확인 없이 진행해야 한다면 `--no-verify` 를 붙입니다.

```
./scripts/offline-load.sh resso-v0.9.77.tar.gz /media/usb/release-sha256.txt
```

2-2. PostgreSQL 준비

전용 Database와 최소 권한 사용자를 만들고, 연결에 TLS를 씁니다. 관리 화면의 사용자 검색과 감사 이벤트 행위자 검색은 선행 Wildcard를 쓰므로 `pg_trgm` 을 설치해 두세요(없어도 동작하지만 전체 스캔이 됩니다 — [운영 가이드](#)의 <<검색 색인>>).

```
CREATE EXTENSION IF NOT EXISTS pg_trgm;
```

2-3. 설정 파일

```
cp .env.example .env
```

두 Keyring은 각각 서로 다른 32바이트 키가 필요합니다.

```
printf 'DATA_ENCRYPTION_KEYS=data-2026-09:%s\n' "$(openssl rand -base64 32)" >> .env
printf 'DIGEST_KEYS=digest-2026-09:%s\n' "$(openssl rand -base64 32)" >> .env
```

.env 에서 최소한 다음 네 개를 실제 값으로 바꿉니다: POSTGRES_DSN , BOOTSTRAP_ADMIN ,
BOOTSTRAP_ADMIN_PASSWORD (12자 이상), 그리고 위에서 만든 두 Keyring. 전체 목록은 [3장](#)에 있습니다.

2-4. 기동

```
docker compose -f compose.offline.yaml up -d
docker compose -f compose.offline.yaml ps
curl -fsS http://127.0.0.1:8080/health/ready
```

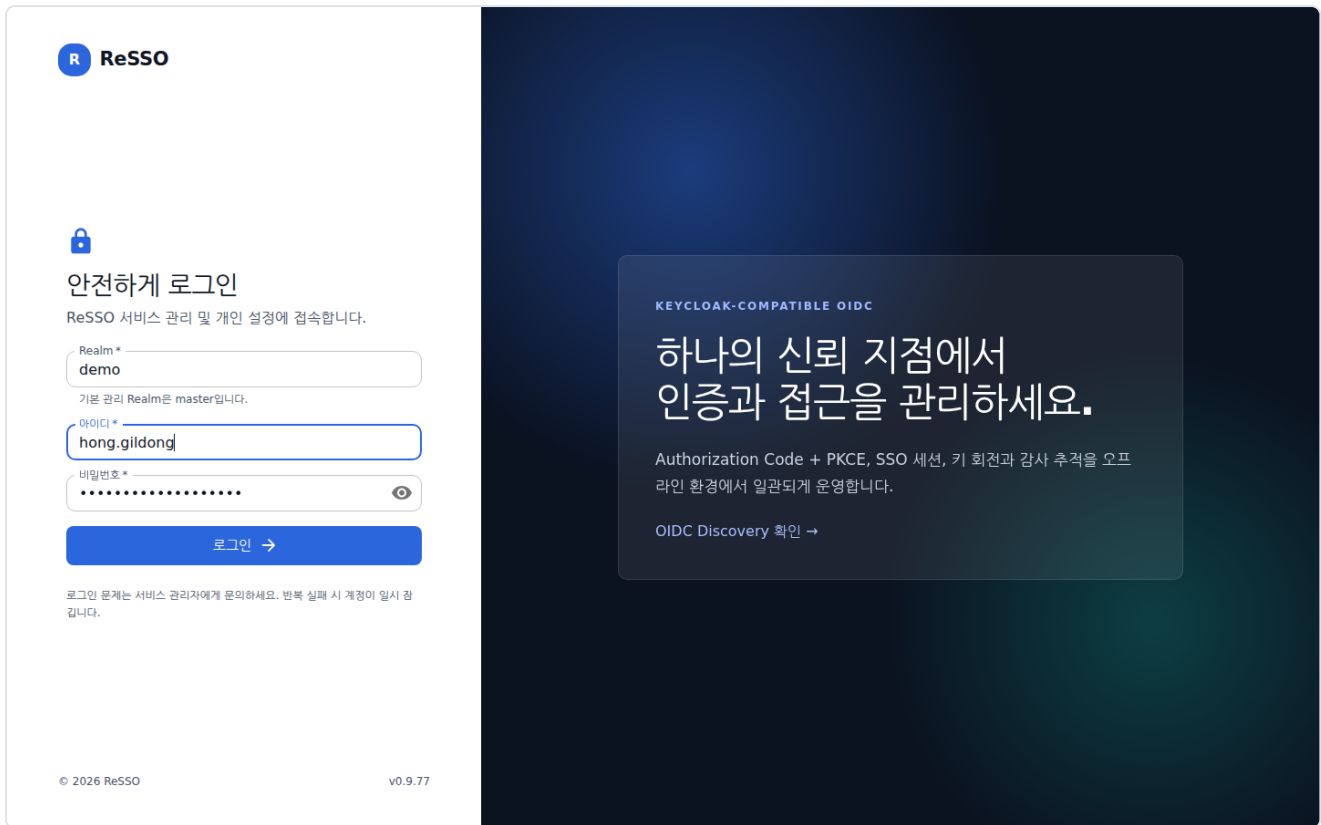
{"status": "ready"} 가 나오면 준비된 것입니다. 뜨지 않으면 [6장](#)의 기동 실패 표를 보세요.

필요한 자원:

항목	값
컨테이너 포트	컨테이너 안에서는 항상 :8080 . 호스트 게시 포트는 .env 의 PUBLISHED_PORT 로 바꿉니다
볼륨	없음. 루트 파일시스템은 읽기 전용이고 /tmp 만 tmpfs 16MB
권한	cap_drop: ALL , no-new-privileges:true
Health Check	이미지에 포함된 /resso healthcheck 를 30초 간격으로 실행
외부 연결	PostgreSQL(필수), LDAP/AD(선택)

2-5. 최초 관리자 계정

BOOTSTRAP_ADMIN / BOOTSTRAP_ADMIN_PASSWORD 로 만들어진 계정이 master Realm의 서비스 관리자입니다. 브라우저에서 게시 포트로 접속해 Realm master , 그 아이디로 로그인합니다.



로그인 — Realm·아이디·비밀번호 세 칸. 애플리케이션에서 넘어오면 Realm 칸은 나타나지 않는다

로그인 직후 반드시 다음 두 가지를 합니다.

1. 개인 설정 → 로그인 보안 에서 Bootstrap 비밀번호를 바꿉니다.
2. 서비스 관리 → Realm → master 의 Issuer URL을 외부 HTTPS 주소로 바꿉니다. 기본값은 `http://localhost:8080/realms/master` 이고, 이 값이 그대로 남아 있으면 대시보드의 외부 Issuer HTTPS 출이 HTTP Issuer 확인 필요 로 경고합니다.

`BOOTSTRAP_ADMIN_PASSWORD` 는 계정을 처음 만들 때만 쓰입니다. 재시작하거나 값을 바꿔도 기존 비밀번호가 재설정되지 않고, 이미 있는 계정을 관리자로 승격시키지도 않습니다.

3. 설정

모든 설정은 환경 변수입니다. 아래는 코드 (`internal/config/config.go`)가 읽는 전부입니다.

이름	기본값	필수	설명
POSTGRES_DSN	—	필수	PostgreSQL 연결 문자열. 운영에서는 <code>sslmode=require</code> 이상. 예: <code>postgres://resso:***@db.example.com:5432/resso?sslmode=require</code>
BOOTSTRAP_ADMIN	—	필수 (서버 기동 시)	최초 master Realm 서비스 관리자 아이디. <code>/</code> , <code>\</code> , 공백, 개행, 탭은 쓸 수 없습니다
BOOTSTRAP_ADMIN_PASSWORD	—	필수 (서버 기동 시)	최초 관리자 비밀번호. 12자 이상 . 계정 생성에만 쓰입니다
DATA_ENCRYPTION_KEYS	—	권장	암호화 Keyring. <code>key-id:base64-또는-hex-32바이트</code> 를 쉼표로 구분하며 첫 키가 신규 Signing Private Key·LDAP Bind Credential 암호화에 쓰입니다. 예: <code>data-2026-09:AAAA...</code> , <code>data-2026-03:BBBB...</code>
DIGEST_KEYS	—	권장	Session·Token·API Key HMAC Keyring. 형식은 위와 같고 첫 키가 신규 Digest에 쓰입니다
ENCRYPTION_KEY	—	선택	v0.2.0 호환 단일 32바이트 키. 분리형 Keyring이 없으면 양쪽 용도로 쓰이고, Keyring과 함께 설정하면 legacy 읽기 키로 자동 추가됩니다
TRUSTED_PROXY_CIDRS	비어 있음	선택	쉼표로 구분한 Reverse Proxy CIDR. 예: <code>10.0.0.0/8,192.168.1.0/24</code> . 등록된 Proxy에서 온 <code>X-Forwarded-For</code> · <code>X-Forwarded-Proto</code> 만 신뢰합니다
LISTEN_ADDRESS	<code>:8080</code>	선택	<code>host:port</code> 형식. 바이너리를 직접 실행할 때 씁니다 — compose 배포에서는 컨테이너가 항상 <code>:8080</code> 에 바인딩하므로 대신 <code>PUBLISHED_PORT</code> 를 바꿉니다

compose 파일이 읽는 변수 두 개가 더 있습니다(애플리케이션 설정이 아니라 배포 설정입니다).

이름	기본값	설명
RESSO_VERSION	v0.9.77	실행할 이미지 태그. 되돌릴 때만 지정합니다
PUBLISHED_PORT	8080	호스트가 게시할 포트

키 규칙:

- DATA_ENCRYPTION_KEYS 와 DIGEST_KEYS 는 함께 설정해야 합니다. 한쪽만 두면 기동을 거절합니다.
- 둘 다 없고 ENCRYPTION_KEY 도 없으면 기동하지 않습니다.
- Key ID는 영숫자로 시작하고 A-Za-z0-9._- 64자 이내입니다. 암호문에 ID가 저장되므로 ID는 키 재료와 함께 불변 식별자로 관리하고, 같은 ID를 다른 키에 재사용하지 마세요.
- 비밀값은 로그·이슈·문서에 남기지 마세요. 이 문서의 예시 값은 모두 가짜입니다.

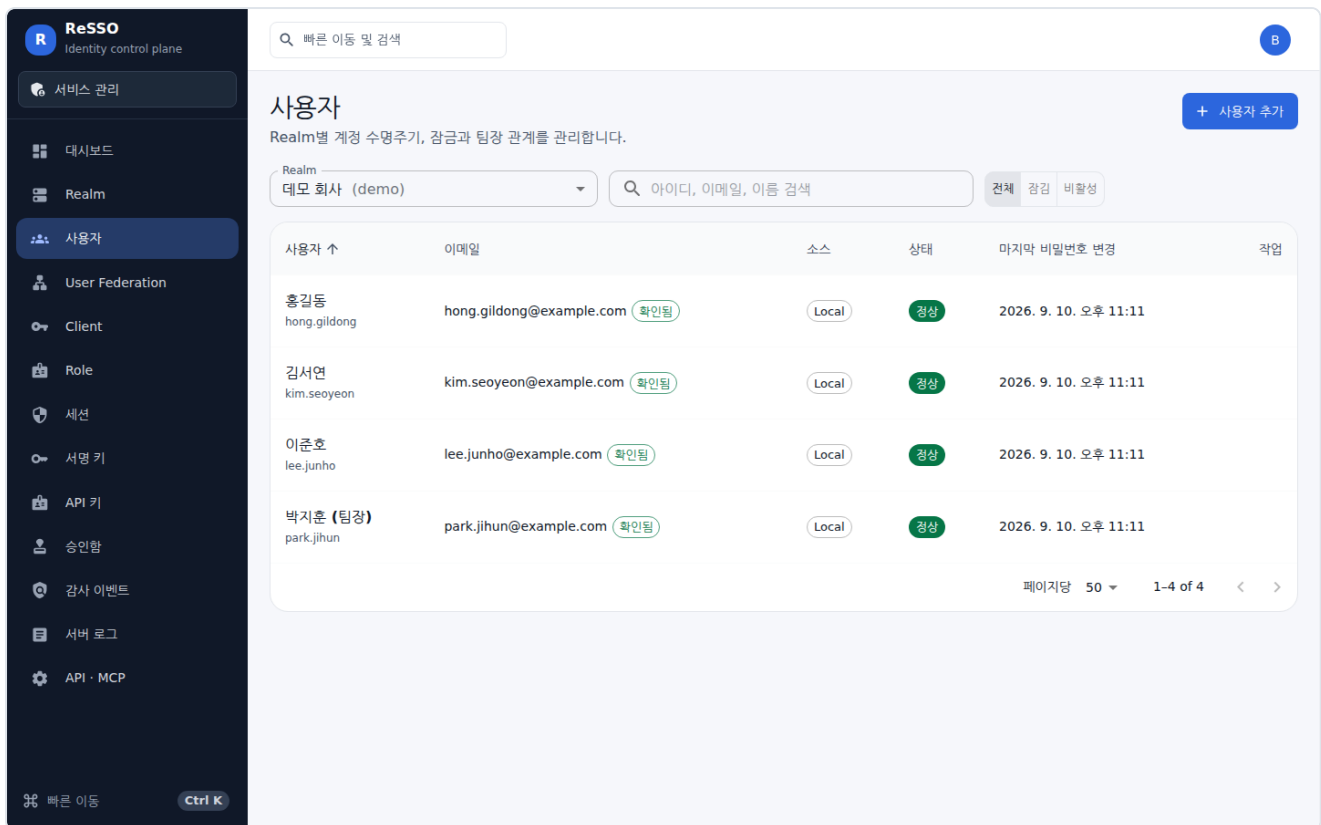
4. 계정과 권한

권한은 세 가지뿐이고, 부여하는 방법이 각각 다릅니다.

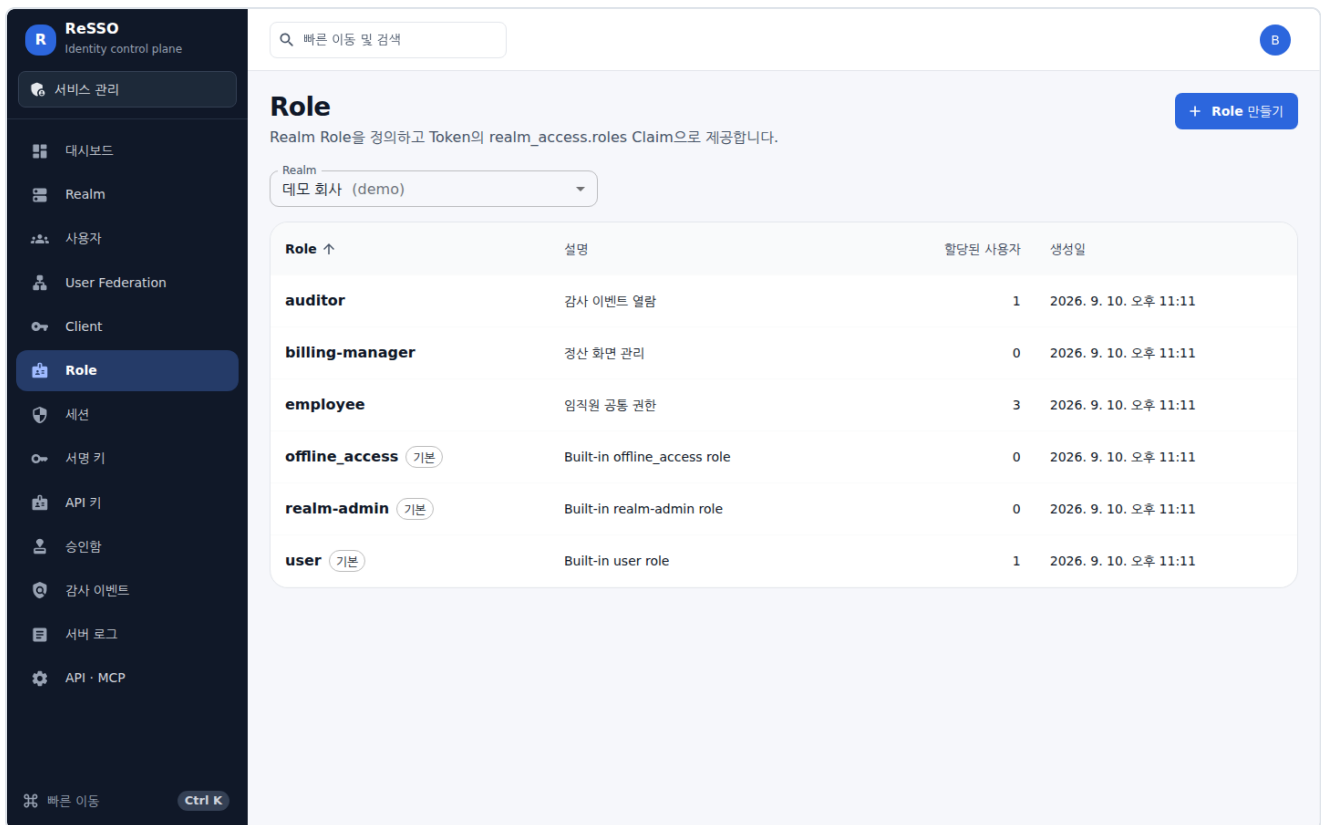
권한	무엇을 할 수 있나	어떻게 부여하나
서비스 관리자 (platform admin)	모든 Realm의 관리 화면. Realm 생성과 서버 로그 화면은 이 권한만 가능	users.platform_admin 컬럼. API·화면으로는 부여할 수 없습니다 — BOOTSTRAP_ADMIN 으로 처음 만들어진 계정이거나, admin recover 로 복구한 계정입니다
Realm 관리자	자기 Realm의 사용자·Client·Role·세션·서명 키·감사 이벤트	그 Realm의 realm-admin Role을 할당
일반 사용자	개인 설정 다섯 화면(사용자 가이드)	기본값

Realm을 만들면 user , realm-admin , offline_access 세 Role이 자동으로 생깁니다.

사용자 관리

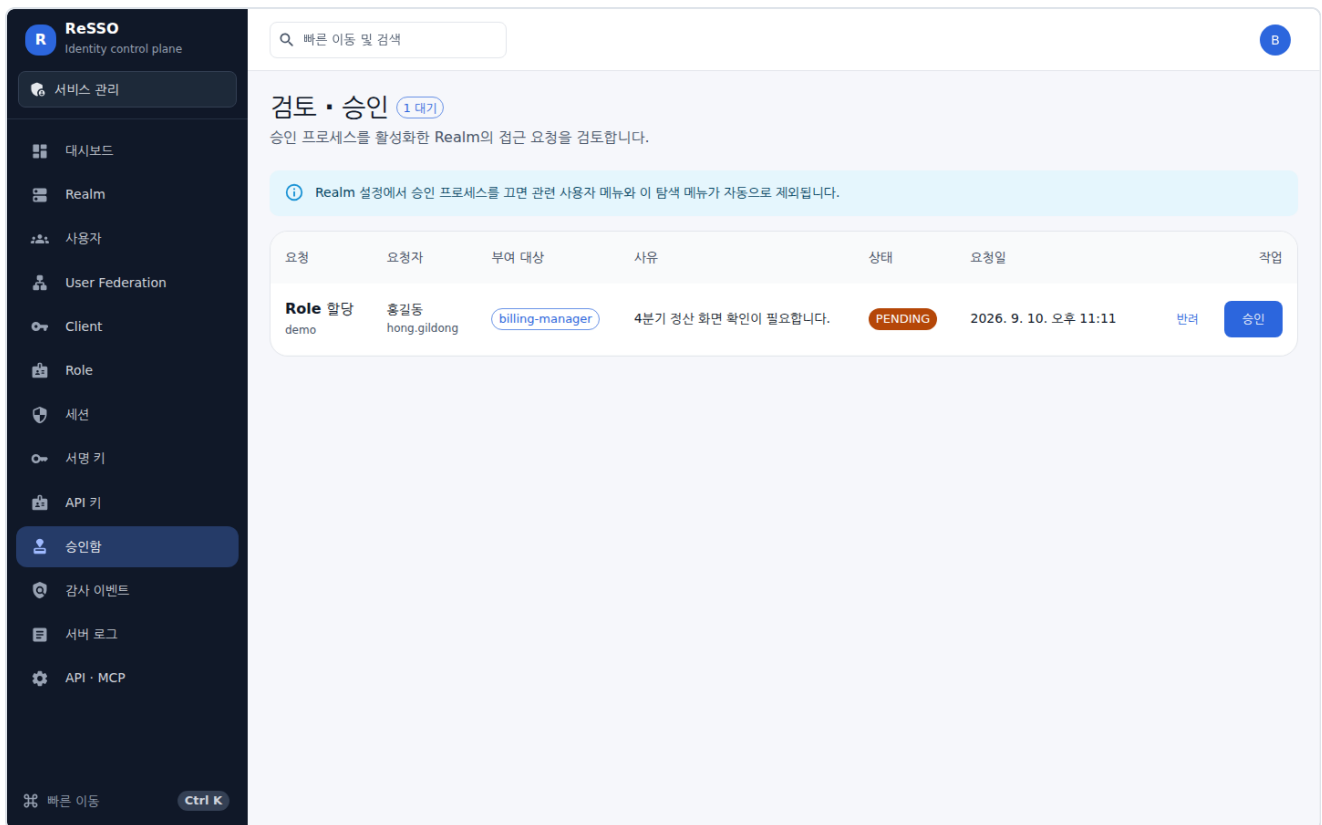


- 위쪽 Realm 선택이 이 화면의 범위입니다. 선택은 주소창의 ?realm= 값으로 함께 이동하므로 화면 링크를 그대로 공유할 수 있습니다.
- 전체 / 잠금 / 비활성 필터로 지금 잠긴 계정만 골라 볼 수 있습니다.
- 각 줄에서 비밀번호 재설정, 계정 활성화/비활성, **잠금 즉시 해제**, 팀장 지정, Role 할당을 합니다.
- 소스 가 Local 이면 ReSSO가 비밀번호를 보관하고, LDAP이면 디렉터리 쪽이 원본입니다.



Realm Role은 Token의 `realm_access.roles` Claim으로 나옵니다. Client별 Role은 `Client` 화면의 각 Client 안에서 관리하며 `resource_access` Claim으로 나옵니다.

Realm 설정에서 승인 절차를 켜면 사용자가 Role을 직접 요청하고 팀장이 검토합니다.

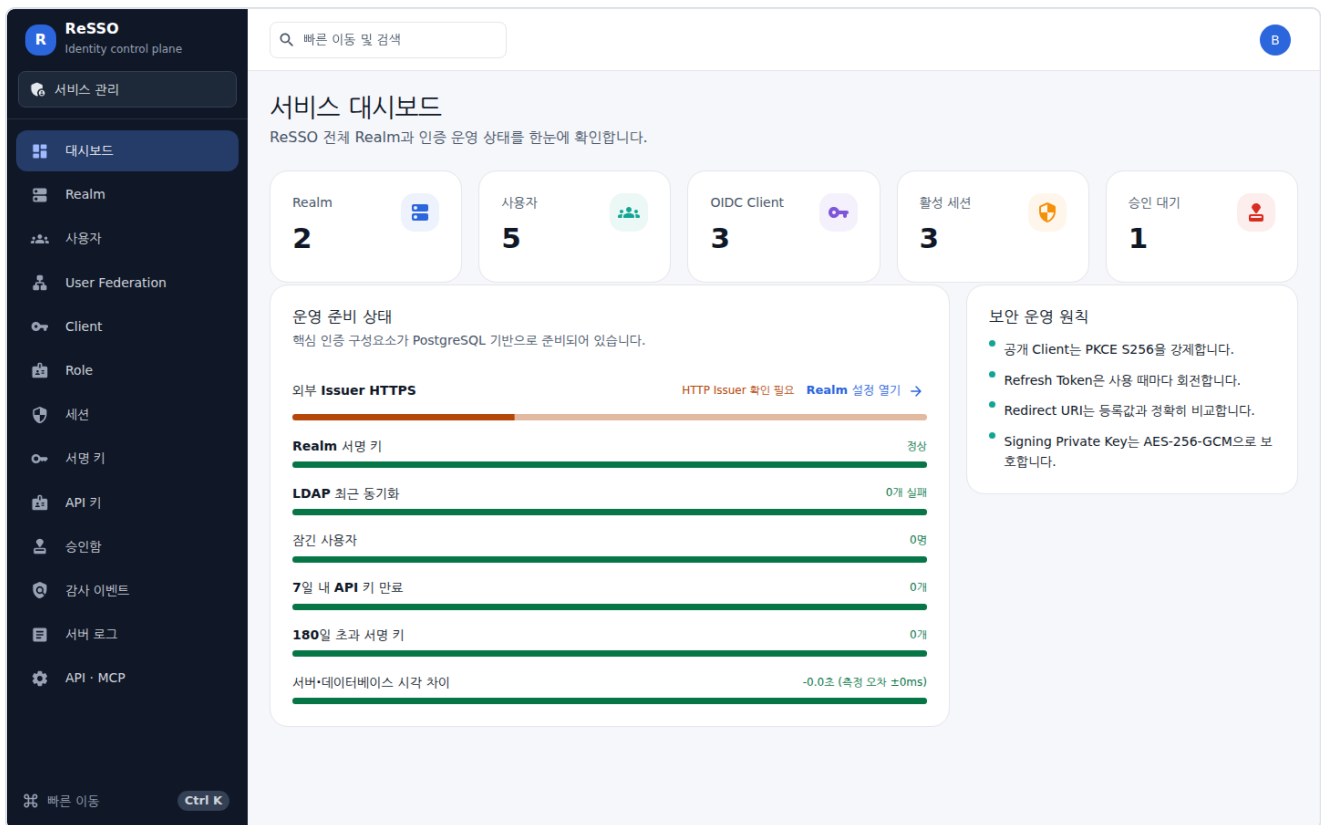


검토·승인 — 요청자·부여 대상 Role·사유를 보고 승인하거나 반려한다

- 검토자는 요청자 계정에 지정된 **팀장**입니다. 팀장이 없으면 Realm 관리자가 결정합니다.
- 승인하면 Role이 즉시 부여되고, 결정과 검토 의견이 감사 이벤트에 남습니다.
- Realm 설정에서 승인 절차를 끄면 이 메뉴와 사용자 쪽 **내 요청** 메뉴가 함께 사라집니다.

5. 운영

5-0. 대시보드부터 본다

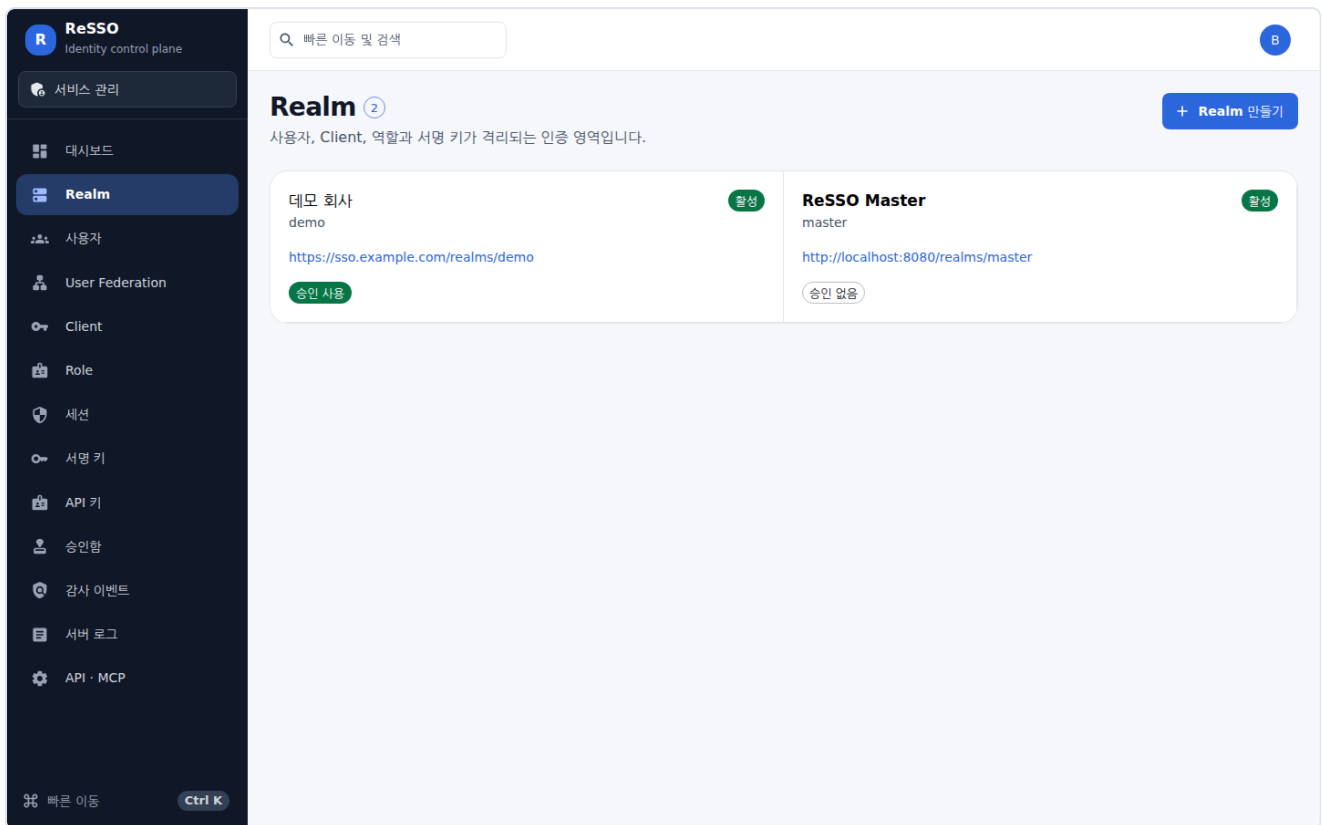


서비스 대시보드 — 왼쪽 '운영 준비 상태'의 일곱 줄이 모두 초록이면 손댈 것이 없다

로그인하면 처음 보이는 화면이고, 아침에 한 번 볼 화면입니다. 운영 준비 상태 의 각 줄은 실제 데이터베이스 상태를 읽은 값입니다.

줄	초록이 아니면
외부 Issuer HTTPS	HTTP Issuer 확인 필요 — Realm의 Issuer URL이 아직 localhost 거나 HTTP입니다 (2-5)
Realm 서명 키	ACTIVE 서명 키가 없는 Realm이 있습니다(5-3)
LDAP 최근 동기화	실패한 동기화가 있습니다(5-10)
잠긴 사용자	지금 잠긴 계정 수. 사용자 화면의 잠김 필터가 같은 집합입니다
7일 내 API 키 만료	곧 멈출 연동이 있습니다(5-5)
180일 초과 서명 키	회전을 검토할 때입니다
서버·데이터베이스 시각 차이	차이가 크면 세션·Token 수명 판정이 흔들립니다(운영 가이드)

5-1. Realm과 Client



Realm — 각 Realm의 Issuer URL과 승인 절차 사용 여부

Realm은 계정·Client·Role·서명 키가 완전히 격리되는 단위입니다. Issuer URL은 애플리케이션이 Discovery로 읽는 주소이므로 외부에서 보이는 HTTPS 주소여야 합니다.

R

ReSSO

Identity control plane

서비스 관리

대시보드

Realm

사용자

User Federation

Client

Role

세션

서명 키

API 키

승인함

감사 이벤트

서버 로그

API · MCP

빠른 이동

Ctrl K

빠른 이동 및 검색

B

OIDC Client

업무 애플리케이션의 Redirect URI, Grant와 Token 정책을 관리합니다.

Realm

데모 회사 (demo)

Client ID, 이름 검색

Client ↑

유형

PKCE

Grant

상태

모바일 근태 앱
mobile-app

public

S256

authorization_code, refresh_token

활성

사내 포털
portal-web

confidential

S256

authorization_code, refresh_token

활성

정산 배치
billing-batch

confidential

선택

client_credentials

활성

+ Client 등록

OIDC Client — 유형·PKCE·Grant를 한 줄로 확인한다

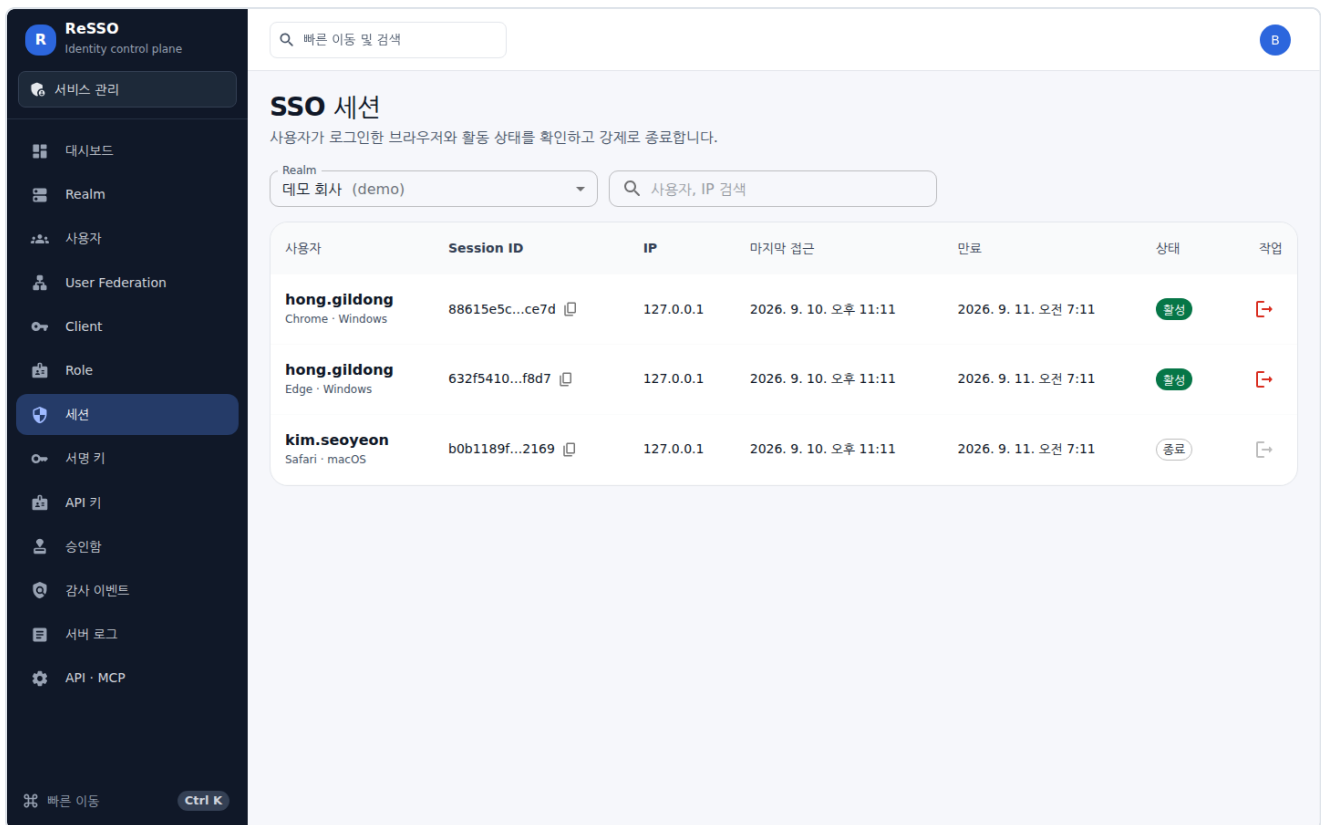
Client 등록 시 지켜야 하는 것:

- Public Client는 **PKCE S256**이 강제됩니다.
- Redirect URI와 Post Logout Redirect URI는 등록값과 **정확히 일치**해야 합니다.
- 브라우저 SPA는 등록된 정확한 Web Origin에서만 Token·UserInfo의 CORS 응답을 받습니다.
- Client Secret은 생성·회전 직후 한 번만 표시됩니다.

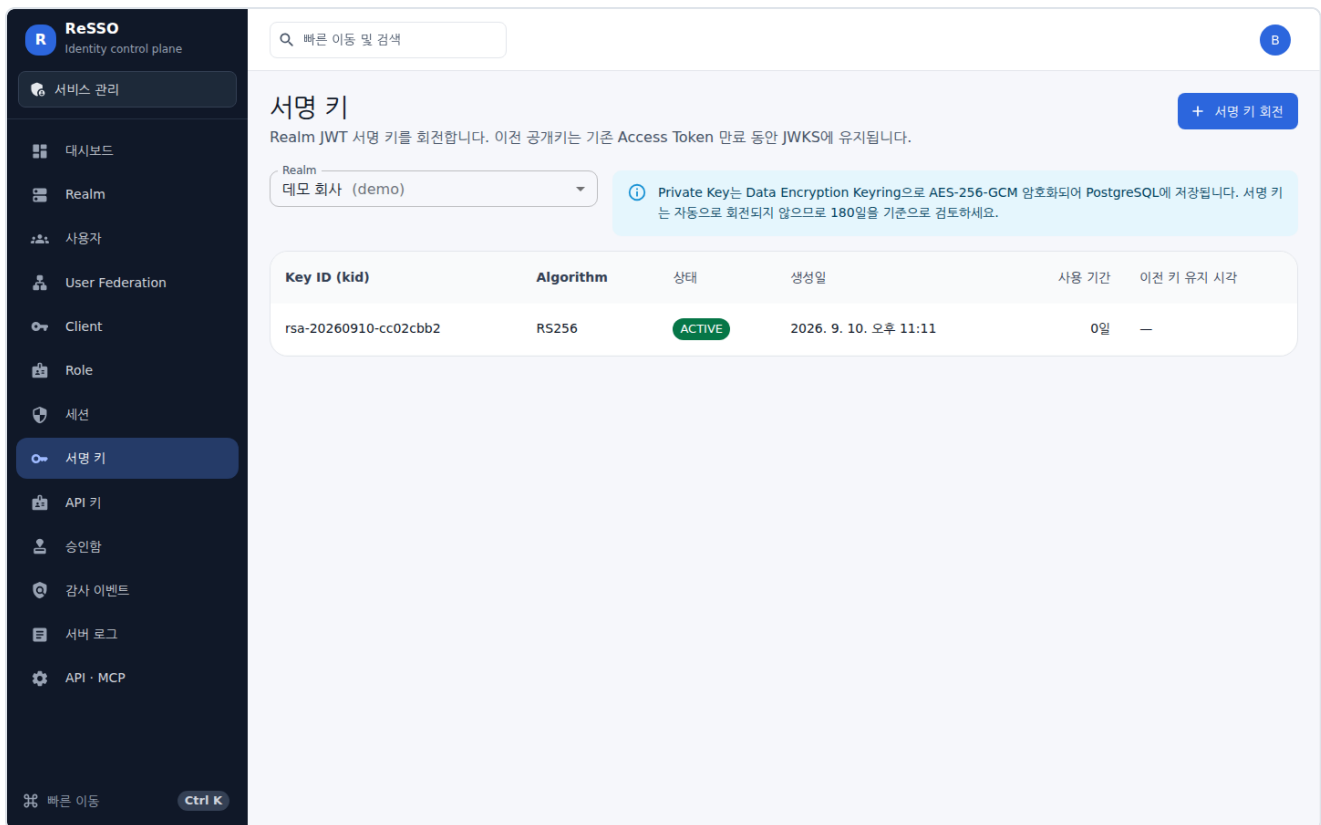
애플리케이션에 알려 줄 주소:

```
https://sso.example.com/realms/{realm}/.well-known/openid-configuration
```

5-2. 세션



회사·사고 대응에서 세션을 즉시 끊는 화면입니다. 종료하면 그 세션에 참여한 Client 중 Back-Channel Logout URI가 등록된 곳에 통지가 나갑니다(전달은 Best effort이며 결과는 감사 이벤트와 [/metrics](#)에서 확인합니다).



- Private Key는 Data Encryption Keyring으로 AES-256-GCM 암호화되어 PostgreSQL에 저장됩니다.
- **자동으로 회전하지 않습니다.** 180일을 기준으로 검토하세요 — 대시보드가 180일 초과 서명 키를 쉼니다.
- 회전해도 이전 공개키는 기존 Access Token 만료 동안 JWKS에 남으므로 즉시 끊기지 않습니다.

5-4. 로그와 감사

R ReSSO
Identity control plane

서비스 관리

대시보드

Realm

사용자

User Federation

Client

Role

세션

서명 키

API 키

승인함

감사 이벤트

서버 로그

API · MCP

감사 이벤트 365일 보론

로그인, Token, 관리자 변경과 키 회전을 보안 감사 관점에서 추적합니다.

이벤트 ▾

결과 ▾

시간 ↓	이벤트	결과	행위자	대상	IP	Trace
26. 9. 10. 오후 11:11:42	LOGIN_SUCCESS	SUCCESS	hong.gildong	session 88615e5c... ce7d	127.0.0.1	8877bcf6... ffb0
26. 9. 10. 오후 11:11:41	APPROVAL_REQUEST_CREATE	SUCCESS	hong.gildong	approval 083c8030... 2aed	127.0.0.1	6253d2ed... 5815
26. 9. 10. 오후 11:11:41	API_KEY_CREATE	SUCCESS	hong.gildong	api_key 6e7515a9... 27d7	127.0.0.1	482facd1... 46a8
26. 9. 10. 오후 11:11:41	API_KEY_CREATE	SUCCESS	hong.gildong	api_key 67da83f2... 7141	127.0.0.1	66c40d53... 73e8
26. 9. 10. 오후 11:11:41	LOGIN_SUCCESS	SUCCESS	hong.gildong	session 632f5410... f8d7	127.0.0.1	8634bee2... c54e
26. 9. 10. 오후 11:11:41	LOGOUT	SUCCESS	kim.seoyeon	session b0b1189f... 2169	127.0.0.1	f15aa990... 07dc
26. 9. 10. 오후 11:11:41	LOGIN_SUCCESS	SUCCESS	kim.seoyeon	session b0b1189f... 2169	127.0.0.1	1b0e4c10... e468

페이지당 100 ▼
1-24 of 24
<
>

ReSSO
Identity control plane

서비스 관리

대시보드

Realm

사용자

User Federation

Client

Role

세션

서명 키

API 키

승인함

감사 이벤트

서버 로그

API · MCP

서버 로그 30일 보존

ReSSO 서버의 구조화 로그를 Trace ID와 함께 확인합니다. Password, Secret, Token 필드는 저장 전에 마스킹됩니다.

Level ▾

[새로고침](#)

시간	Level	Component	메시지	Trace ID
26. 9. 10. 오후 11:12:00	INFO	resso	http request	99d8be3a...d15e 🔗
26. 9. 10. 오후 11:12:00	INFO	resso	http request	f67adf2f...1cf7 🔗
26. 9. 10. 오후 11:12:00	INFO	resso	http request	a54c5b3d...0867 🔗
26. 9. 10. 오후 11:11:59	INFO	resso	http request	313cf640...edf0 🔗
26. 9. 10. 오후 11:11:59	INFO	resso	http request	d90edb60...6718 🔗
26. 9. 10. 오후 11:11:59	INFO	resso	http request	9ef33b35...8102 🔗
26. 9. 10. 오후 11:11:59	INFO	resso	http request	8b8991d5...d050 🔗
26. 9. 10. 오후 11:11:59	INFO	resso	http request	2a394e42...7ce8 🔗
26. 9. 10. 오후 11:11:59	INFO	resso	http request	aa73a823...6503 🔗
26. 9. 10. 오후 11:11:59	INFO	resso	http request	5ef0a4d0...72ab 🔗

어디	무엇	보존
컨테이너 표준 출력	JSON 구조화 로그	수집 Agent에 위임
서버 로그 화면 (서비스 관리자 전용)	같은 로그를 DB에 보관, Trace ID 검색	30일
감사 이벤트 화면	로그인·Token·설정·키 변경	365일

더 긴 보존이 필요하면 표준 출력을 SIEM으로 전달하세요. 화면의 로그 저장은 장애 분석 편의를 위한 보조 계층입니다.

5-5. API 키

API 키
이 Realm의 구성원이 발급한 개인 API 키입니다. Secret은 저장되지 않으므로 여기에도 표시되지 않습니다.

키가 만료되면 그 키를 쓰는 연동은 예고 없이 멈춥니다. 만료가 가까운 키는 소유자에게 회전을 요청하세요.

Realm: 데모 회사 (demo) 전체 7일 내 만료

소유자	이름	Prefix	Scope	마지막 사용	만료	상태
hong.gildong	근태 조회 스크립트	rk_tSWIIC2CPdIX	api:read	—	2026. 12. 9. 오후 11:11	활성
hong.gildong	MCP 클라이언트	rk_TmkkquzjwPdb	mcp:read	—	2027. 3. 9. 오후 11:11	활성

API 키 — 이 Realm 구성원이 발급한 개인 키의 소유자·범위·만료

Secret은 저장되지 않으므로 이 화면에도 표시되지 않습니다. 7일 내 만료 필터가 대시보드의 같은 이름 카운터와 같은 집합입니다 — 만료되면 그 키를 쓰는 연동이 예고 없이 멈추므로, 소유자에게 회전을 요청하세요.

5-6. Health Check와 지표

경로	메서드	인증	내용
/health/live	GET	없음	프로세스 생존
/health/ready	GET	없음	PostgreSQL 연결 준비 상태
/metrics	GET	필요 (admin:read 개인 API 키 또는 관리자 세션)	Prometheus text format
/api/v1/meta	GET	없음	제품·버전·커밋

Prometheus는 Bearer token으로 붙입니다.

```
scrape_configs:
  - job_name: rosso
    metrics_path: /metrics
    authorization:
      credentials: rk_xxxxx.yyyyy
    static_configs:
      - targets: ['sso.example.com']
```

지표 목록과 각 계열을 어떻게 읽는지는 [README의 《운영 지표》](#)와 [운영 가이드](#)에 있습니다.

5-7. 백업과 복구

백업 대상은 두 가지뿐이고, 둘 다 있어야 복구됩니다.

1. PostgreSQL — 조직의 RPO/RTO에 맞춘 PITR 가능한 백업
2. DATA_ENCRYPTION_KEYS · DIGEST_KEYS 의 그 시점 값 — DB와 다른 보안 경계에 보관

백업 시점에 쓰이던 읽기 키가 빠진 채로 기동하면 기존 Private Key·Session·Token·API Key를 쓸 수 없습니다. 복구 훈련은 격리 환경에서 분기별로 권장합니다.

5-8. 업그레이드와 롤백

```
# 1. 새 이미지 반입
./scripts/offline-load.sh resso-v0.9.78.tar.gz

# 2. 데이터베이스 백업 (마이그레이션 전)

# 3. 태그를 올리고 재기동 — 마이그레이션은 기동 시 자동 적용됩니다
sed -i 's/^# *RESSO_VERSION=.* /RESSO_VERSION=v0.9.78/' .env
docker compose -f compose.offline.yaml up -d
curl -fsS http://127.0.0.1:8080/health/ready
```

되돌리기: `.env` 의 `RESSO_VERSION` 을 이전 태그로 되돌리고 다시 `up -d` 합니다. 되돌린 이미지는 자기가 모르는 Migration이 이미 적용된 데이터베이스를 만나게 되는데, **기동이 막히지는 않고** 기동 로그에 경고가 남으며 `admin diagnose` 의 `migrations_ahead_of_binary` 에도 나타납니다. 경고가 보이면 그 Migration을 넣은 버전의 Upgrade notes에서 DB 호환성과, 되돌린 동안 적용되지 않는 설정이 무엇인지 확인하세요 — 자세한 판단 기준은 [운영 가이드](#)의 《업그레이드와 롤백》에 있습니다. 되돌린 뒤에는 `Discovery`·`로그인`·`Token`·`JWKS`·`UserInfo`를 한 번씩 확인합니다 (`./scripts/smoke-test.sh`).

5-9. 유지보수 CLI

HTTP 서버를 띄우지 않고 PostgreSQL에 직접 붙는 명령입니다. `POSTGRES_DSN` 과 현재 Keyring이 필요하며 Bootstrap 환경 변수는 필요하지 않습니다.

```
# 상태 진단
docker compose -f compose.maintenance.yaml --profile maintenance run --rm \
    resso-maintenance admin diagnose

# Signing Private Key와 LDAP Bind Credential을 활성 Data Encryption Key로 재암호화
docker compose -f compose.maintenance.yaml --profile maintenance run --rm \
    resso-maintenance crypto rewrap

# 마지막 관리자가 잠겼을 때의 break-glass 복구
read -rsp 'New recovery password: ' RESSO_RECOVERY_PASSWORD; echo
printf '%s\n' "$RESSO_RECOVERY_PASSWORD" | docker compose -f compose.maintenance.yaml \
    --profile maintenance run --rm -T resso-maintenance \
    admin recover --username recovery-admin --password-stdin
unset RESSO_RECOVERY_PASSWORD
```

`admin recover` 는 master Realm의 로컬 관리자를 만들거나 비밀번호·잠금·권한을 복구하고, **그 계정의 기존 Session·Refresh Token·개인 API 키를 모두 폐기합니다**. LDAP 계정을 로컬 계정으로 바꾸지는 않습니다.

5-10. User Federation (선택)

ReSSO Identity control plane

서비스 관리

대시보드

Realm

사용자

User Federation

Client

Role

세션

서명 키

API 키

승인함

감사 이벤트

서버 로그

API · MCP

빠른 이동 Ctrl K

빠른 이동 및 검색

User Federation

Realm 사용자 계정을 LDAP 또는 Active Directory와 연결하고 로그인·동기화 정책을 중앙 관리합니다.

Realm: 데모 회사 (demo)

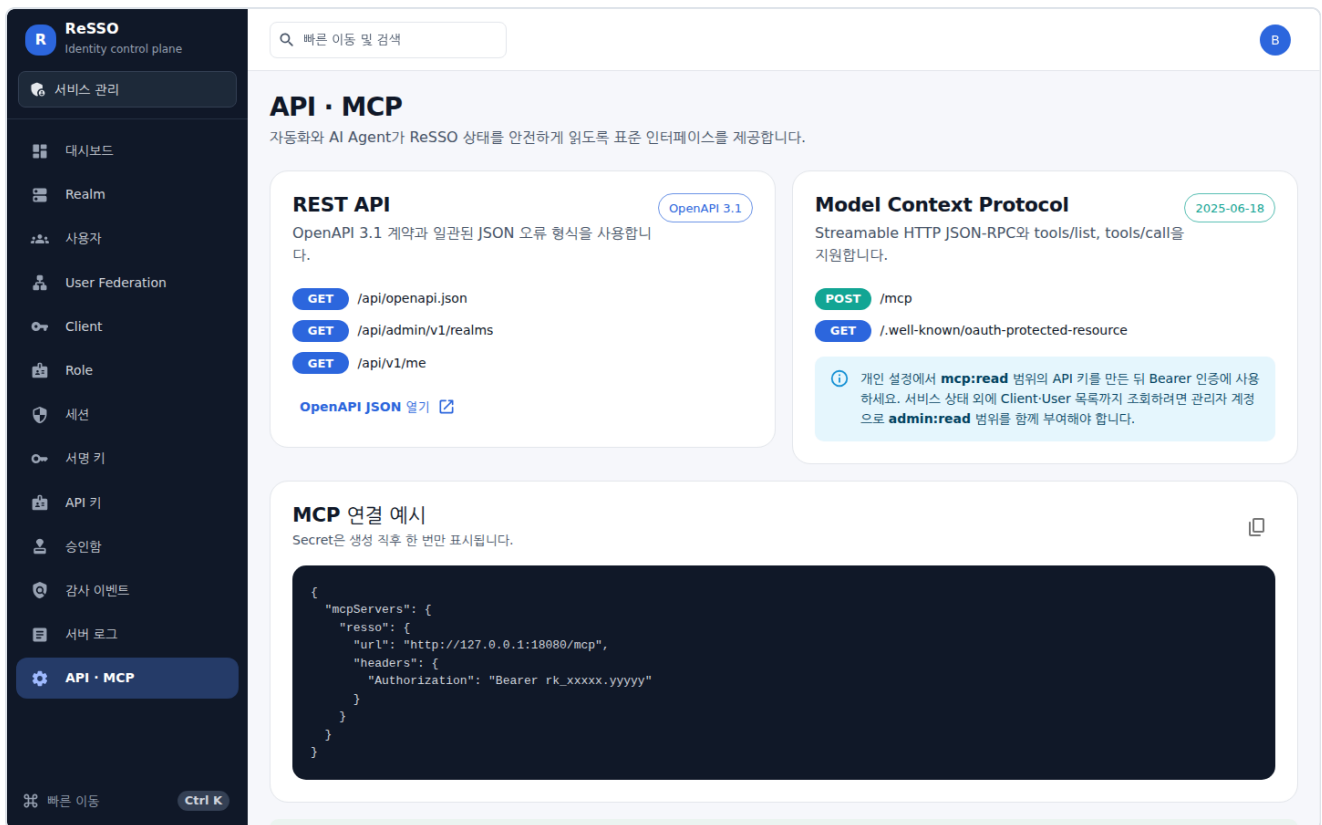
낮은 우선순위 숫자의 공급자부터 로그인 사용자를 검색합니다. 운영 전 연결 및 인증 테스트를 실행하세요.

공급자	연결	편집 모드	마지막 동기화	상태
본사 Active Directory Active Directory · 우선순위 0	ldaps://ad.example.com:636	READ_ONLY	실행 전	활성

User Federation — 등록된 디렉터리 공급자의 연결 주소·편집 모드·마지막 동기화

Realm별로 LDAP/Active Directory를 연결합니다. 등록 직후에는 마지막 동기화 가 실행 전 이며, **운영 전에 화면의 연결 테스트와 인증 테스트를 반드시 실행하세요.** 낮은 우선순위 숫자의 공급자부터 로그인 사용자를 검색합니다. 속성 매핑·Edit mode·Group→Role 매핑 등 세부 설정은 [User Federation 문서](#)에 있습니다.

5-11. API와 MCP



API · MCP — OpenAPI와 MCP endpoint, 연결 예시를 화면에서 그대로 복사한다

경로	메서드	내용
/api/openapi.json	GET	OpenAPI 3.1 문서
/mcp	POST	MCP Streamable HTTP (GET은 405)
/.well-known/oauth-protected-resource	GET	OAuth Protected Resource Metadata

변경 API는 CSRF가 적용된 브라우저 세션만 허용합니다. 개인 API 키는 어떤 범위든 읽기 전용입니다.

6. 장애 대응

기동하지 않을 때

컨테이너 표준 출력의 마지막 `ERROR` 한 줄이 원인을 그대로 말합니다.

로그 메시지	원인	조치
configuration validation failed + required environment variables are missing: ...	필수 변수 누락	나열된 변수를 .env 에 채웁니다
configuration validation failed + BOOTSTRAP_ADMIN_PASSWORD must contain at least 12 characters	비밀번호가 짧음	12자 이상으로 바꿉니다
configuration validation failed + ... must be exactly 32 bytes encoded as base64 or hexadecimal	Keyring 값의 길이·인코딩 오류	openssl rand -base64 32 결과를 그대로 씁니다
configuration validation failed + DATA_ENCRYPTION_KEYS and DIGEST_KEYS must be configured together	한쪽만 설정	두 Keyring을 함께 설정합니다
database initialization failed / database migration failed	PostgreSQL에 붙지 못했거나 마이그레이션 실패	DSN·네트워크·권한을 확인합니다. 마이그레이션 실패는 백업을 확보한 뒤 로그의 SQL 오류를 그대로 확인합니다
encryption service initialization failed	Keyring 값이 형식은 맞으나 초기화 실패	Key ID 규칙(A-Za-z0-9._- , 64자 이내) 과 중복 ID를 확인합니다
service bootstrap failed	최초 관리자 생성 실패	BOOTSTRAP_ADMIN 문자 제약(/ , \ , 공백 불가)을 확인합니다

뜯 뒤에 생기는 증상

증상	어디를 보나	조치
애플리케이션이 unknown client_id 를 받는다	Client 화면에서 그 client_id 의 존재와 활성화 여부	이 답은 등록되지 않았거나 꺼진 Client 에만 나옵니다. 등록하거나 활성화하세요 — 데이터베이스 장애는 이 답을 만들지 않습니다
애플리케이션이 갑자기 500 server_error 를 받는다	서버 로그 에서 the client named in the authorization request could not be looked up	데이터베이스 장애입니다. 등록 오류가 아니므로 Client 설정을 건드리지 마세요
로그아웃 뒤 브라우저가 애플리케이션이 아니라 ReSSO 페이지에 남는다	서버 로그 에서 the client named at logout could not be looked up	이 줄이 있으면 데이터베이스 장애이고, 없으면 애플리케이션이 등록되지 않은 post_logout_redirect_uri 를 보낸 것입니다
사용자가 "잠겼다"고 문의한다	사용자 화면 → 잠김 필터	목록에 있으면 그 줄에서 즉시 잠금 해제합니다. 없으면 잠긴 것이 아니므로 비밀번호 문제입니다
로그인 화면에서 로그인 요청을 확인하지 못했습니다... 를 봤다는 문의	서버 로그 의 그 Trace ID	사용자를 애플리케이션으로 돌려보내지 마세요. 요청은 살아 있으며 이쪽 장애가 겹치면 다시 시도 로 이어집니다
인가가 조용히 실패한다(사용자가 로그인 화면에 도달하지 못한다)	/metrics 의 resso_authorization_errors_total{stage=...}	이 계열은 대부분 302로 나가 성공한 인가와 HTTP 상태가 같습니다. stage 값이 어느 조화가 멈췄는지 가리킵니다
Token 발급이 실패한다	/metrics 의 resso_token_errors_total{stage=...}	400 invalid_grant (호출자 문제)와 500 server_error (이쪽 문제)의 구분은 운영 가이드 에 있습니다
Prometheus가 /metrics 에서 401·403을 받는다	스크래이프 설정의 Bearer token	admin:read 범위 개인 API 키인지, 만료되지 않았는지 확인합니다
대시보드가 HTTP Issuer 확인 필요를 표시한다	Realm 화면의 Issuer URL	외부 HTTPS 주소로 바꿉니다
클라이언트 IP가 전부 Proxy 주소로 보인다	TRUSTED_PROXY_CIDRS	실제 Proxy 네트워크를 등록합니다. 등록되지 않은 발신지의 전달 헤더는 무시됩니다

증상별 더 깊은 절차(계정 잠금 운영, 로그인 요청 제한, 승인 프로세스, 서버·DB 시각 차이)는 [운영 가이드](#)에 있습니다.

7. 보안

기본값 중 바꿔야 하는 것

항목	기본값	해야 할 일
master Realm Issuer URL	<code>http://localhost:8080/realms/master</code>	외부 HTTPS 주소로 변경
Bootstrap 관리자 비밀번호	<code>.env</code> 에 적은 임시 값	최초 로그인 직후 변경. 이름을 붙인 관리자를 만든 뒤에는 이 계정을 비활성화해도 됩니다 — 재시작해도 다시 켜지지 않습니다
TRUSTED_PROXY_CIDRS	비어 있음	Reverse Proxy 뒤에 둔다면 그 네트워크만 등록
POSTGRES_DSN 의 sslmode	<code>.env.example</code> 은 <code>require</code>	그대로 두거나 더 강하게

외부에 열면 안 되는 것

- PostgreSQL 포트를 서비스 네트워크 밖으로 열지 마세요.
- `compose.maintenance.yaml` 의 유지보수 컨테이너는 필요할 때만 `run --rm` 으로 띄웁니다.
- `/metrics` 는 이미 관리 권한을 요구하지만, 스크레이프 경로도 내부망으로 제한하는 편이 낫습니다.
- 콘솔(`/`)과 OIDC Endpoint(`/realms/...`)는 같은 포트를 쓰므로 Reverse Proxy에서 `/`, `/api`, `/realms`, `/mcp`, `/well-known` 경로를 변경 없이 전달해야 합니다.

지켜야 할 습관

- `POSTGRES_DSN`, Bootstrap 비밀번호, Keyring 값을 로그·티켓·저장소에 남기지 마세요.
- Keyring과 PostgreSQL 백업은 서로 다른 보안 경계에 보관하세요.
- 개인 API 키와 Client Secret은 생성 직후 한 번만 표시됩니다. 잃으면 회전하세요.
- 감사 이벤트 와 서버 로그 를 정기적으로 검토하세요.
- Digest Key는 재작성할 수 없습니다. 그 키로 만든 Session·Token·API Key가 모두 만료·회전되고 그 키로 만든 Client Secret이 모두 회전될 때까지 읽기 키로 유지해야 합니다. Client Secret은 만료되지 않으므로 사실상 그것이 제거 시점을 정합니다.

8. 화면 캡처 다시 찍기

이 문서와 사용자 가이드의 그림은 저장소의 `scripts/guide-screenshots.mjs` 가 찍습니다. 데모 데이터를 직접 만들어 넣으므로 버려도 되는 인스턴스에만 씁니다 — 대상은 전용 변수 `RESSO_GUIDE_URL` 로만 받고, loopback 주소가 아니면 그 자리에서 멈춥니다.

```
make build

docker run -d --name rosso-guide-pg -p 127.0.0.1:55480:5432 \
  -e POSTGRES_USER=rosso -e POSTGRES_PASSWORD=rosso -e POSTGRES_DB=rosso postgres:17-alpine
docker exec rosso-guide-pg psql -U rosso -d rosso -c 'CREATE EXTENSION IF NOT EXISTS pg_trgm'

POSTGRES_DSN='postgres://rosso:rosso@127.0.0.1:55480/rosso?sslmode=disable' \
BOOTSTRAP_ADMIN=admin BOOTSTRAP_ADMIN_PASSWORD='demo-admin-pass-1234' \
DATA_ENCRYPTION_KEYS="data-demo:${openssl rand -base64 32}" \
DIGEST_KEYS="digest-demo:${openssl rand -base64 32}" \
LISTEN_ADDRESS=127.0.0.1:18080 ./build/rosso &

RESSO_GUIDE_URL=http://127.0.0.1:18080 \
RESSO_GUIDE_ADMIN=admin RESSO_GUIDE_ADMIN_PASSWORD='demo-admin-pass-1234' \
node scripts/guide-screenshots.mjs
```

PDF는 저장소마다 만들지 않고 공용 도구를 씁니다.

```
node <aidev>/tools/guide/md2pdf.mjs docs/USER_GUIDE.md docs/USER_GUIDE.pdf \
  --title "사용자 가이드" --subtitle "..." --project "ReSSO" --version "v0.9.77"
```