

TACHYON Admin

TACHYON Admin

Version 0.6.2

Date: 2026년 02월 26일

Author: INCA Internet Co., Ltd.

© 2026 INCA Internet Co., Ltd.. All rights reserved.

목차

0. tkadmin 소개	1
0.1 tkadmin이란?	1
0.2 주요 기능 요약	1
0.3 시스템 요구사항	2
0.4 시스템 구조 다이어그램	3
0.5 화면 구성 미리보기	4
0.6 다음 단계	4
1. 설치 가이드	4
1.1 사전 요구사항	5
1.2 바이너리 배포	5
1.3 설치 명령어	6
1.4 설치 확인	6
1.5 포트 정보	7
1.6 서비스 제거	7
1.7 업데이트 절차	7
1.8 기타 CLI 옵션	8
1.9 다음 단계	9
2. 배포 가이드 (시스템 관리자)	10
2.1 배포 사전 요구사항	10
2.2 신규 설치 절차	10
2.3 업데이트 절차	12
2.4 로그 경로 검증	13
2.5 서비스 제거	14
2.6 배포 체크리스트	14
2.7 배포 수칙 요약	15
3. 최초 접속 가이드	16
3.1 TACHYON 대시보드에서 접근하기	16
3.2 직접 URL 접근	17

3.3 Recovery Mode 접근 (긴급 시)	17
3.4 화면 구성 소개	18
3.5 나가기 버튼	18
3.6 다음 단계	19
4. 대시보드	19
4.1 개요	20
4.2 상단 장애 알림 배너	20
4.3 알림 버튼 및 배지	21
4.4 카드 섹션	21
4.5 자동 갱신	24
4.6 조작 순서 요약	25
4.7 다음 단계	25
5. 내/외부 설정 편집기	25
5.1 개요	26
5.2 설정 항목 상세	26
5.3 설정 파일 위치	30
5.4 클라이언트 IP 표시 기능	30
5.5 저장 플로우	30
5.6 설정 취소	31
5.7 조작 순서 요약	31
5.8 다음 단계	32
6. 전문가 편집기	33
6.1 개요	33
6.2 화면 구성	33
6.3 파일 트리 탐색	33
6.4 파일 열기 및 편집	34
6.5 저장 프로세스	36
6.6 보안	36
6.7 주의사항	36
6.8 관련 API	37
6.9 다음 단계	38
7. 설정 파일 레퍼런스	39
7.1 설정 파일 경로	39

7.2 전체 설정 옵션	39
7.3 설정 파일 전체 예시	41
7.4 주요 설정 항목 상세	42
7.5 환경 탐색(Discovery) 우선순위	43
8. 서비스 관리	45
8.1 개요	45
8.2 서비스 목록 표시	45
8.3 서비스 제어	46
8.4 Watchdog 자동 복구	47
8.5 Watchdog 오탐 방지	48
8.6 주의사항	50
8.7 관련 API	51
8.8 다음 단계	51
9. 환경 체크	51
9.1 개요	52
9.2 화면 구성	52
9.3 OS Limits 확인 및 설정	52
9.4 SELinux 상태 제어	53
9.5 방화벽(Firewalld) 포트 관리	54
9.6 주의사항	55
9.7 관련 API	56
9.8 다음 단계	56
10. 시스템 로그	56
10.1 개요	57
10.2 화면 구성	57
10.3 실시간 로그 추적	58
10.4 필터링 기능	58
10.5 과거 로그 조회 (무한 스크롤)	59
10.6 로그 상세 데이터 확장	60
10.7 성능	60
10.8 주의사항	61
10.9 관련 API	61
10.10 다음 단계	62

11. 서비스 로그 뷰어	63
11.1 개요	63
11.2 화면 구성	63
11.3 주요 기능	64
11.4 대시보드에서 바로 가기	67
11.5 지원 서비스	67
11.6 비활성 탭 최적화	67
11.7 팁	67
12. 관리자 보고	69
12.1 개요	69
12.2 화면 구성	69
12.3 보고 유형	69
12.4 필터링 기능	70
12.5 읽음 관리	71
12.6 실시간 갱신	72
12.7 알림 배너 정책	72
12.8 알림 데이터 영속성	73
12.9 주의사항	73
12.10 관련 API	74
12.11 다음 단계	75
13. Recovery Mode (긴급 복구 모드)	76
13.1 개요	76
13.2 접근 방법	76
13.3 인증 방식	76
13.4 IP 기반 접근 제어 (ACL)	78
13.5 제공 API (제한적)	79
13.6 사용 시나리오	79
13.7 보안 고려사항	80
14. API 레퍼런스	81
14.1 공통 사항	81
14.2 설정 관리	84
14.3 외부 파일 편집	88
14.4 서비스 제어	90

14.5 알림 / 감사 로그	92
14.6 환경 체크	95
14.7 로그 조회	97
14.8 서비스 파일 로그 (v0.2.2)	99
15. 웹 터미널	105
15.1 개요	105
15.2 접근 방법	105
15.3 화면 구성	105
15.4 사용 방법	106
15.5 연결 관리	107
15.6 테마 지원	107
15.7 제한사항	108
15.8 트러블슈팅	108
99. 자주 묻는 질문 (FAQ)	109
99.1 Q1. 로그인에 안 됩니다	109
99.2 Q2. 서비스가 시작되지 않습니다	110
99.3 Q3. 설정 변경 후 반영되지 않습니다	111
99.4 Q4. 로그가 표시되지 않습니다	112
99.5 Q5. 알림 배너가 사라지지 않습니다	113
99.6 Q5-1. 서비스가 자체 업데이트 중인데 장애 알림이 발생합니다	114
99.7 Q6. 긴급 복구 모드로 접근하려면?	115
99.8 Q7. TACHYON 대시보드에 Admin 메뉴가 보이지 않습니다	116
99.9 Q8. 전문가 편집기에서 저장이 실패합니다	117

0. tkadmin 소개

| 0.1 tkadmin이란?

tkadmin은 TACHYON 솔루션의 고성능 설정 관리 및 운영 자동화 도구입니다. 복잡한 서버 환경 설정을 웹 인터페이스를 통해 직관적으로 제어하며, 단일 바이너리 배포를 통해 설치 및 운영 편의성을 극대화합니다.

tkadmin은 TACHYON 솔루션이 설치된 리눅스 서버에 배포되어, 운영자가 별도의 터미널 접속 없이도 웹 브라우저만으로 솔루션의 전반적인 설정과 상태를 관리할 수 있도록 지원합니다. 특히 폐쇄망(Air-gap) 환경에서도 외부 인터넷 연결 없이 100% 동작하도록 설계되었습니다.

- **제품 버전:** 0.6.1
 - **기술 스택:** Go (Gin) + Vanilla JS (Web Components) + Monaco Editor
-

| 0.2 주요 기능 요약

1. 단일 바이너리 배포 (Single-Binary)

모든 웹 리소스(HTML, CSS, JS, 라이브러리)를 Go의 `go:embed` 기술로 바이너리에 내장합니다. 외부 의존성 없이 바이너리 파일 하나만으로 전체 서비스를 실행할 수 있어 배포와 관리가 매우 간편합니다.

2. 환경 자동 탐색 (Environment Discovery)

TACHYON 서버의 운영 환경 정보를 동적으로 크롤링하여 MariaDB, Redis, OpenSearch, Kafka 등의 연결 정보를 자동으로 감지합니다. 설정 파일이 훼손되거나 존재하지 않는 극한 상황에서도 내장된 Fallback 정보를 통해 최소한의 관리 기능을 유지합니다.

3. Phoenix UI (다크 모드 기반 인터페이스)

Glassmorphism과 Vibrant Colors를 활용한 다크 모드 기반의 고급스러운 디자인을 제공합니다. 외부 프레임워크 없이 Vanilla JS와 Web Components만으로 구현되어 가볍고 빠른 반응 속도를 보장합니다.

4. Auth Bridge (인증 연동)

TACHYON SPA 대시보드와 세션을 물리적으로 동기화합니다. 운영자가 TACHYON 대시보드에 로그인된 상태라면, 별도의 추가 인증 없이 자연스럽게 tkadmin에 접근할 수 있습니다. Redis 기반의 실시간 세션 검증으로 TACHYON 로그아웃 시 즉시 연동 해제됩니다.

5. Watchdog 및 알림 시스템

서비스 상태를 실시간으로 감시하며, 장애 발생 시 자동 복구를 시도하고 관리자에게 알림(상단 배너, 뱃지)을 전송합니다. 모든 장애 및 복구 이력은 SQLite에 영구 보관되어 사후 감사 추적이 가능합니다.

6. CLI 통합 관리

한 줄의 명령어로 서비스 설치(`-i`), 제거(`-u`), 실행(`-r`), 종료(`-k`) 등을 처리합니다. systemd 서비스 등록, NGINX 리버스 프록시 설정, TACHYON 대시보드 인젝션까지 자동으로 수행됩니다.

7. 전문가 설정 편집기 (Monaco Editor)

TACHYON 솔루션 전반의 설정 파일(`.yaml` , `.properties` , `.conf` 등)을 웹 브라우저에서 직접 편집할 수 있습니다. 자동 백업, YAML 구문 검증, 경로 순회(Path Traversal) 차단 등 안전장치가 내장되어 있습니다.

0.3 시스템 요구사항

항목	요구사항
운영체제	Linux (RHEL/CentOS 7 이상, Rocky Linux 8/9 등)
서비스 관리	systemd 활성화 필수
사전 설치	TACHYON 솔루션 설치 완료 (기본 경로: <code>/usr/local/TACHYON/TTS40/</code>)
웹 서버	NGINX (리버스 프록시용, tkadmin이 자동 설정)
네트워크 포트	기본 포트 <code>13700</code> , Recovery 포트 <code>13701</code>
브라우저	Chrome, Edge 등 최신 Chromium 기반 브라우저 권장

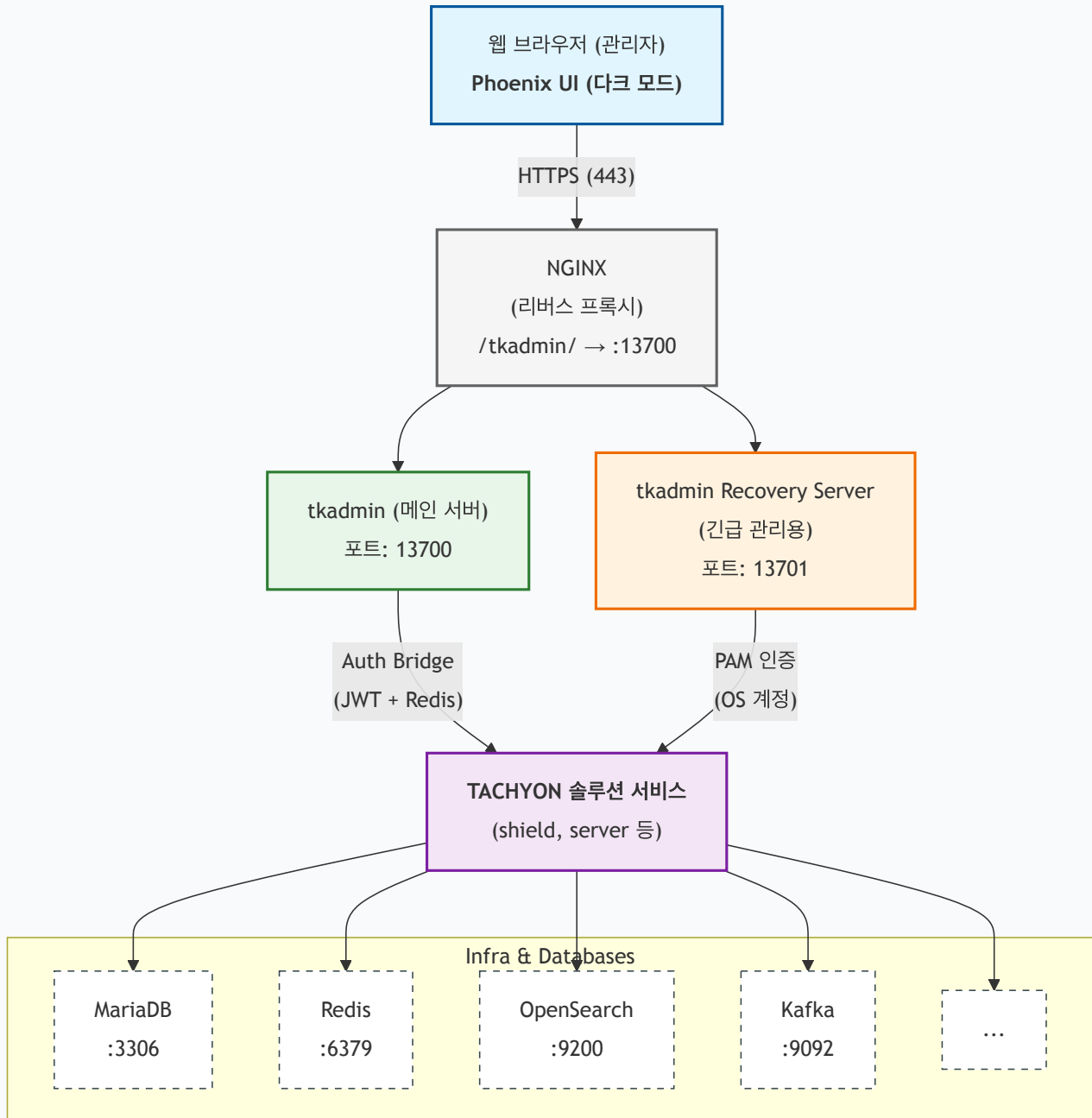


팁

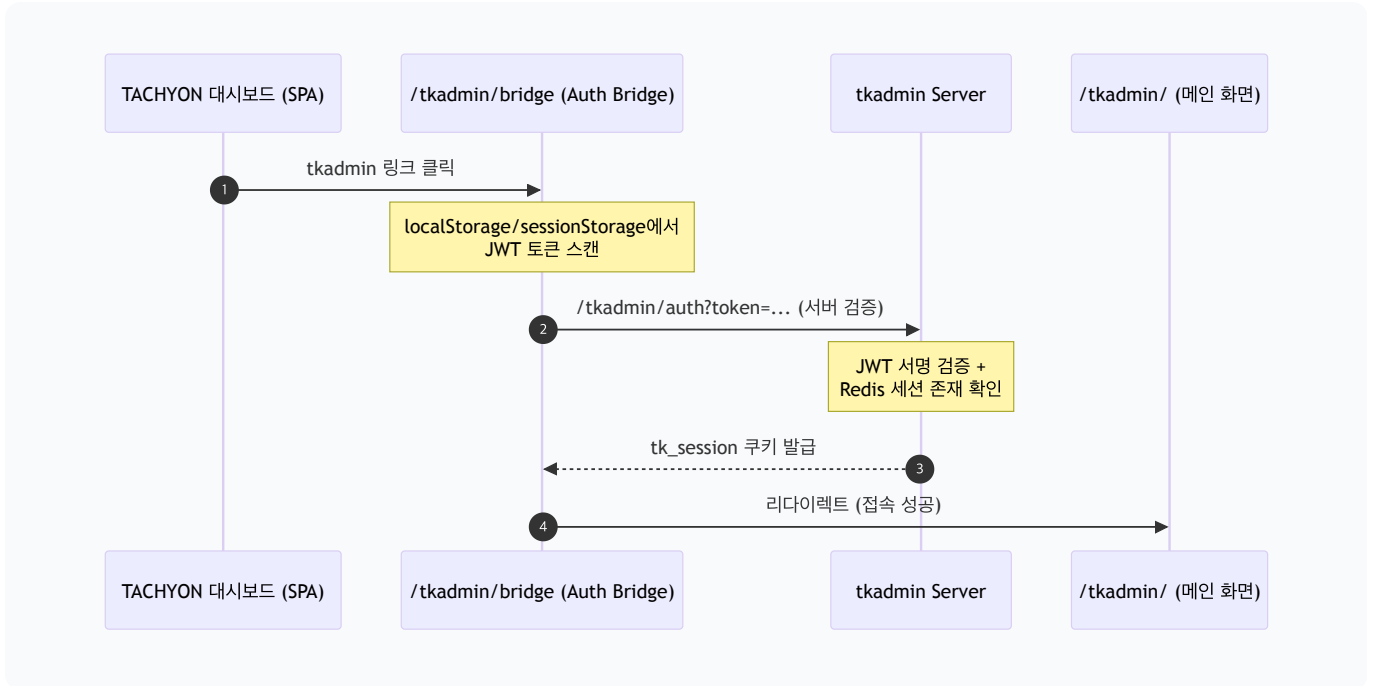
tkadmin은 폐쇄망(Air-gap) 환경을 완벽히 지원합니다. 외부 CDN이나 인터넷 연결이 필요하지 않습니다.

0.4 시스템 구조 다이어그램

전체 아키텍처



인증 흐름



0.5 화면 구성 미리보기

tkadmin의 메인 화면은 다음과 같은 영역으로 구성됩니다:

- **상단 헤더 바:** 로고, 알림 버튼(배지 포함), 나가기 버튼
- **좌측 사이드바:** 메뉴 네비게이션 (접힘/펼침 지원)
- **메인 콘텐츠 영역:** 선택된 메뉴에 따라 대시보드, 설정 편집기, 로그 뷰어 등이 표시

0.6 다음 단계

- [설치 가이드](#)로 이동하여 tkadmin을 설치하세요.
- [최초 접속](#)에서 로그인 및 화면 구성을 확인하세요.

1. 설치 가이드

1.1 사전 요구사항

tkadmin을 설치하기 전에 다음 조건이 충족되어야 합니다:

항목	요구사항
운영체제	Linux (RHEL/CentOS 7+, Rocky Linux 8/9 등)
TACHYON 솔루션	설치 완료 (기본 경로: <code>/usr/local/TACHYON/TTS40/</code>)
systemd	활성화 상태 (<code>systemctl</code> 명령 사용 가능)
NGINX	설치 및 구동 중 (리버스 프록시 설정 자동 주입을 위해 필요)
실행 권한	<code>root</code> 또는 <code>sudo</code> 권한 필요

1.2 바이너리 배포

tkadmin은 단일 바이너리 파일로 배포됩니다. 별도의 설치 패키지나 의존성 라이브러리가 필요하지 않습니다.

1. 배포받은 `tkadmin` , `tkctl` 바이너리 파일을 설치 경로에 복사합니다.

```
# 설치 디렉토리 생성
sudo mkdir -p /usr/local/tkadmin/bin

# 바이너리 복사 및 실행 권한 부여
sudo cp tkadmin tkctl /usr/local/tkadmin/bin/
sudo chmod +x /usr/local/tkadmin/bin/tkadmin /usr/local/tkadmin/bin/tkctl
```

2. 바이너리 버전을 확인합니다.

```
/usr/local/tkadmin/bin/tkadmin -v
```

1.3 설치 명령어

서비스 설치

다음 명령어로 tkadmin을 시스템 서비스로 등록합니다:

```
/usr/local/tkadmin/bin/tkadmin -i
```

-i 옵션은 다음 작업을 자동으로 수행합니다:

1. **표준 설치(Standard Install) 경로 생성:** `/usr/local/tkadmin/` 하위에 `bin`, `conf`, `data`, `logs` 등 표준 폴더 구조를 자동 생성합니다.
2. **설정 파일 마이그레이션:** 기존 루트에 있던 `tkadmin.yml` 을 `conf/tkadmin.yml` 로 자동 이동하여 표준을 준수합니다.
3. **systemd 서비스 등록:** `tkadmin.service` 유닛 파일을 생성하고 서비스를 활성화합니다.
 - `LimitNOFILE=65535` 설정이 포함됩니다.
 - `RuntimeDirectory=tkadmin` 설정을 통해 `/run/tkadmin/` 을 활용합니다.
4. **NGINX 리버스 프록시 설정 주입:** `/tkadmin/` 경로에 대한 프록시 패스 규칙을 NGINX 설정에 자동으로 추가합니다.
5. **TACHYON 대시보드 인젝터 스크립트 삽입:** TACHYON SPA 대시보드에 접근 메뉴를 자동으로 삽입합니다.
6. **서비스 시작:** 설치 완료 후 서비스가 자동으로 시작됩니다.

1.4 설치 확인

설치가 완료되면 다음 명령어로 서비스 상태를 확인합니다:

```
systemctl status tkadmin
```

정상적으로 설치된 경우 다음과 유사한 출력을 확인할 수 있습니다:

```
● tkadmin.service - tkadmin TACHYON Admin Console
   Loaded: loaded (/etc/systemd/system/tkadmin.service; enabled; ...)
   Active: active (running) since ...
   Main PID: 12345 (tkadmin)
   ...
```

추가로 NGINX 설정이 올바르게 적용되었는지 확인합니다:

```
nginx -t
systemctl status nginx
```

1.5 포트 정보

포트	용도	비고
13700	메인 서비스 포트	NGINX 리버스 프록시를 통해 HTTPS(443)로 접근
13701	Recovery 서비스 포트	긴급 관리용, 직접 HTTP 접근 (PAM 인증 + IP ACL)



팁

포트 번호는 `tkadmin.yml` 설정 파일에서 변경할 수 있습니다. Recovery 포트는 기본적으로 메인 포트 + 1로 자동 설정됩니다.

1.6 서비스 제거

tkadmin을 시스템에서 완전히 제거하려면 다음 명령어를 실행합니다:

```
/usr/local/tkadmin/bin/tkadmin -u
```

`-u` 옵션은 다음 작업을 수행합니다:

1. systemd 서비스 중지 및 삭제
2. NGINX 리버스 프록시 설정 제거
3. TACHYON 대시보드 인젝터 스크립트 원복 (삽입된 태그 및 관련 파일 제거)



팁

제거 시 `tkadmin.yml` 설정 파일은 삭제되지 않고 보존됩니다. 재설치 시 기존 설정이 그대로 유지됩니다.

1.7 업데이트 절차

tkadmin을 업데이트할 때는 다음 절차를 준수해야 합니다:

1. 서비스 중지

```
systemctl stop tkadmin
```

2. 기존 바이너리를 새 바이너리로 교체

```
# 새 바이너리로 교체
sudo cp tkadmin tkctl /usr/local/tkadmin/bin/
sudo chmod +x /usr/local/tkadmin/bin/tkadmin /usr/local/tkadmin/bin/tkctl
```

3. 새 바이너리로 재설치

```
/usr/local/tkadmin/bin/tkadmin -i
```

주의

업데이트 시

기존 바이너리의 `-u` 옵션을 절대 실행하지 마세요.

버그가 있는 기존 버전이 시스템을 손상시킬 수 있는 Critical Bug가 보고되어 있습니다. 반드시

`systemctl stop` → 바이너리 교체 → 새 바이너리 `-i` 순서를 따르세요.

1.8 기타 CLI 옵션

옵션	설명
<code>-i</code>	systemd 서비스 등록, NGINX 설정 주입, 인젝터 설치
<code>-u</code>	서비스 중지/삭제, NGINX 설정 제거, 인젝터 원복
<code>-k</code>	실행 중인 tkadmin 프로세스 강제 종료
<code>-r</code>	백그라운드 데몬 모드로 실행
<code>-v</code>	버전 정보 출력
<code>service start</code>	시스템 서비스 시작
<code>service stop</code>	시스템 서비스 중지
<code>service restart</code>	시스템 서비스 재시작
<code>service status</code>	시스템 서비스 상태 조회

1.9 다음 단계

설치가 완료되었다면, [최초 접속 가이드](#)로 이동하여 tkadmin에 처음 접속하는 방법을 확인하세요.

2. 배포 가이드 (시스템 관리자)

본 문서는 tkadmin의 신규 설치, 업데이트, 제거 절차를 시스템 관리자 관점에서 상세히 설명합니다.

2.1 배포 사전 요구사항

필수 환경

항목	요구사항
운영체제	Linux (systemd 기반, RHEL/CentOS 7+ 또는 Ubuntu 18.04+)
TACHYON	TTS40 설치 완료 (<code>/usr/local/TACHYON/TTS40/</code>)
권한	root 또는 sudo 권한
NGINX	TACHYON 내장 NGINX (<code>/usr/local/TACHYON/TTS40/nginx/</code>)
Redis	TACHYON 세션 관리용 Redis 서버 (기본: <code>127.0.0.1:6379</code>)

선택 환경

항목	용도
Chromium/Chrome	스크린샷 도구 기능 (선택사항)

2.2 신규 설치 절차

1단계: 바이너리 전송

빌드된 tkadmin 바이너리를 대상 서버의 TACHYON 설치 경로로 전송합니다.

SCP를 통한 전송:


```
scp tkadmin root@서버주소:/usr/local/TACHYON/TTS40/
```

USB를 통한 전송 (폐쇄망 환경):

```
cp /mnt/usb/tkadmin /usr/local/TACHYON/TTS40/
chmod +x /usr/local/TACHYON/TTS40/tkadmin
```

2단계: 서비스 설치

TACHYON 설치 디렉토리에서 설치 명령을 실행합니다:

```
cd /usr/local/TACHYON/TTS40
sudo ./tkadmin -i
```

설치가 정상적으로 완료되면 다음 출력이 표시됩니다:

```
Service tkadmin.service installed and started successfully.
Injected NGINX include into
/usr/local/TACHYON/TTS40/nginx/conf/conf.d/ssl.conf via block parsing
Injected tkadmin script into Tachyon index.html successfully.
```

3단계: 설치 확인

서비스 상태를 확인하여 정상 구동 여부를 점검합니다:

```
# systemd 서비스 상태 확인
systemctl status tkadmin

# 포트 리스닝 확인
netstat -nlpt | grep 13700

# 로그 파일 생성 확인
ls -la /usr/local/TACHYON/TTS40/logs/tkadmin.log

# 웹 접속 확인 (localhost)
curl -s http://127.0.0.1:13700/tkadmin/api/status | python3 -m json.tool
```

2.3 업데이트 절차

⚠ 주의

업데이트는 반드시 `deploy.bat` 을 사용하여 수행합니다. 수동 배포(scp + ssh 조합)는 금지합니다.

deploy.bat을 사용한 자동 업데이트 (권장)

Windows 개발 환경에서 `deploy.bat` 을 실행하면 다음 절차가 자동으로 수행됩니다:

```
deploy.bat
```

`deploy.bat` 내부 동작 순서:

1. SSH 접속
2. `systemctl stop tkadmin` (안전 중지)
3. 기존 바이너리 삭제
4. 새 바이너리 SCP 전송
5. `tkadmin -i` 재설치

수동 업데이트 (deploy.bat 사용이 불가능한 경우)

`deploy.bat` 을 사용할 수 없는 환경에서는 다음 절차를 **정확한 순서**로 수행합니다:

```
# 1. 서비스 안전 중지
systemctl stop tkadmin

# 2. 기존 바이너리 삭제
rm /usr/local/TACHYON/TTS40/tkadmin

# 3. 새 바이너리 복사 (SCP, USB 등)
cp /경로/새_tkadmin /usr/local/TACHYON/TTS40/tkadmin
chmod +x /usr/local/TACHYON/TTS40/tkadmin

# 4. 새 바이너리로 재설치
cd /usr/local/TACHYON/TTS40
sudo ./tkadmin -i
```

**주의 (Critical Bug)**

업데이트 시 기존(구버전) 바이너리의 **-u** 옵션을

절대 실행하지 마십시오

. 구버전 바이너리에 존재하는 버그로 인해 로깅 설정이 빈 문자열이거나 디렉토리 경로만 지정된 경우,
제품 디렉토리 전체가 파일로 덮어써지는

치명적 손상이 발생할 수 있습니다. (2025-12-19 발견)

설정 파일 보존 확인

업데이트 후 **tkadmin.yml** 설정 파일이 보존되었는지 확인합니다:

```
# 설정 파일 존재 확인
ls -la /usr/local/TACHYON/TTS40/tkadmin.yml

# 설정 내용 확인
cat /usr/local/TACHYON/TTS40/tkadmin.yml
```

**주의**

tkadmin.yml 설정 파일은 삭제하지 마십시오. 새 바이너리만 교체하고 설정 파일은 그대로 유지해야 합니다.

tkctl 자동 업데이트 (Self-Healing)

tkadmin 은 서비스 시작 시 내장된 **tkctl** 바이너리의 무결성을 자동으로 검증합니다. **-i** 옵션 없이 바이너리만 교체하고 서비스를 재시작하더라도, **tkadmin** 이 최신 버전의 **tkctl** 을 자동으로 추출하여 **/usr/local/tkadmin/bin/tkctl** 경로에 동기화합니다.

2.4 로그 경로 검증

배포 후 로그 파일 경로가 올바르게 설정되었는지 반드시 확인합니다:

```
# 로그 파일 경로 확인
ls -la /usr/local/TACHYON/TTS40/logs/tkadmin.log
```

```
# 최근 로그 내용 확인
tail -20 /usr/local/TACHYON/TTS40/logs/tkadmin.log
```

`tkadmin.yml` 의 로깅 설정이 올바른지 확인합니다:

```
logging:
  file: "logs/tkadmin.log"      # 반드시 파일명을 포함해야 합니다
  level: "info"
  max_size: 10485760           # 10MB (바이트 단위)
  max_backups: 5
  max_age: 30
  compress: true
```



주의

`logging.file` 값이 빈 문자열(`" "`)이거나 디렉토리 경로(`"logs/"`)만 지정되면 치명적 버그가 발생합니다. 반드시 파일명을 포함한 경로를 설정하세요.

2.5 서비스 제거

tkadmin을 완전히 제거해야 하는 경우 **현재 설치된 최신 바이너리**의 `-u` 옵션을 사용합니다:

```
cd /usr/local/TACHYON/TTS40
sudo ./tkadmin -u
```

제거 시 수행되는 작업:

1. systemd 서비스 중지 및 삭제 (`tkadmin.service`)
2. NGINX 설정에서 tkadmin 관련 include 제거 및 `tkadmin.location` 파일 삭제
3. TACHYON `index.html` 에서 인젝터 스크립트 태그 제거
4. `tkadmin_injector.js` 파일 삭제
5. `tkctl` 바이너리 삭제 (`/usr/local/tkadmin/bin/tkctl`)

2.6 배포 체크리스트

배포 완료 후 다음 항목을 확인합니다:

- ☐ `systemctl status tkadmin` - 서비스 Active(running) 상태 확인

- ☐ `netstat -nlpt | grep 13700` - 포트 리스닝 확인
- ☐ 로그 파일 경로: `/usr/local/TACHYON/TTS40/logs/tkadmin.log` 정상 생성 확인
- ☐ `tkadmin.yml` 설정 파일 보존 여부 확인
- ☐ 웹 UI 접속 확인: `https://서버주소/tkadmin/`
- ☐ TACHYON 대시보드에서 tkadmin 메뉴 표시 확인 (show_link 활성화 시)
- ☐ Recovery 포트 접근 확인: `http://127.0.0.1:13701/recovery/`

2.7 배포 수칙 요약

규칙	설명
deploy.bat 사용 필수	수동 배포(scp + ssh)는 금지, 반드시 deploy.bat을 통해 배포
기존 바이너리 <code>-u</code> 실행 금지	구버전 바이너리의 언인스톨은 시스템 손상 위험
안전 중지 절차	<code>systemctl stop</code> → 바이너리 삭제 → 새 바이너리 복사 → <code>-i</code> 설치
설정 파일 보존	<code>tkadmin.yml</code> 은 삭제하지 않고 바이너리만 교체
로그 경로 검증	배포 후 <code>/usr/local/TACHYON/TTS40/logs/tkadmin.log</code> 경로 확인
시스템 리미트	서비스 유닛 파일에 <code>LimitNOFILE=65535</code> 설정 포함 확인

3. 최초 접속 가이드

3.1 TACHYON 대시보드에서 접근하기

tkadmin은 TACHYON 대시보드와 긴밀하게 통합되어 있습니다. 가장 자연스러운 접근 방법은 TACHYON 대시보드를 통한 진입입니다.

방법 1: 사이드바 메뉴

1. TACHYON 대시보드(`https://[서버IP]/`)에 관리자 계정으로 로그인합니다.
2. 좌측 사이드바에서 '**Admin**' 메뉴(방패 아이콘)를 클릭합니다.
3. Auth Bridge가 자동으로 인증을 처리하고 tkadmin 메인 화면으로 이동합니다.

방법 2: 시스템 설정 페이지

1. TACHYON 대시보드에 로그인합니다.
2. 시스템 설정 페이지로 이동합니다.
3. 우측 상단의 '**tkadmin 이동**' 버튼을 클릭합니다.

Auth Bridge 인증 흐름

TACHYON 대시보드에서 tkadmin에 접근하면, 다음과 같은 자동 인증 과정이 진행됩니다:

1. **토큰 스캔**: Auth Bridge 페이지(`/tkadmin/bridge`)가 로드되며, 브라우저의 `localStorage` 및 `sessionStorage` 에서 TACHYON 인증 토큰(JWT)을 자동으로 탐색합니다.
2. **서버 검증**: 발견된 토큰의 JWT 서명을 검증하고, Redis에 해당 세션이 실제로 존재하는지 확인합니다.
3. **세션 발급**: 검증 성공 시 `tk_session` 쿠키가 발급되고, tkadmin 메인 화면(`/tkadmin/`)으로 자동 리다이렉트됩니다.



팁

이 모든 과정은 1~2초 내에 자동으로 완료됩니다. 별도의 ID/비밀번호 입력이 필요하지 않습니다.



주의

TACHYON에서 로그아웃하면 Redis 세션이 삭제되므로, tkadmin 세션도 즉시 무효화됩니다. 이는 보안을 위한 실시간 동기화 정책입니다.

3.2 직접 URL 접근

TACHYON 대시보드를 거치지 않고 직접 URL로 접근할 수도 있습니다:

```
https://[서버IP]/tkadmin/
```

이 경우에도 Auth Bridge가 동작하여 TACHYON 인증 토큰이 브라우저에 존재하면 자동 로그인됩니다. 토큰이 없거나 만료된 경우에는 TACHYON 로그인 페이지(</user/login>)로 자동 리다이렉트됩니다.

3.3 Recovery Mode 접근 (긴급 시)

TACHYON 인증 서버에 장애가 발생하여 정상적인 로그인이 불가능한 경우, Recovery Mode를 통해 tkadmin에 접근할 수 있습니다.

접근 방법

1. 브라우저에서 Recovery 포트로 직접 접속합니다:

```
http://[서버IP]:13701/tkadmin/
```

2. Recovery 로그인 화면이 표시됩니다.
3. **운영체제(OS) 계정**의 사용자명과 비밀번호를 입력합니다.
 - TACHYON 계정이 아닌 Linux OS 계정(PAM 인증)을 사용합니다.
4. 인증 성공 시 **tk_recovery** 쿠키가 발급되고 tkadmin 화면으로 이동합니다.
 - Recovery Mode에서는 제한된 기능만 제공됩니다 (대시보드, 설정 변경, 알림 조회, 감사 로그).

IP 기반 접근 제어 (ACL)

Recovery 포트는 보안을 위해 IP 기반 접근 제어가 적용됩니다:

허용 대상	설명
127.0.0.1 (localhost)	서버 자체에서의 접근 항상 허용
::1 (IPv6 localhost)	IPv6 로컬 접근 허용
emergency_ips 목록	tkadmin.yaml에 등록된 비상 접근 IP 허용

허용되지 않은 IP에서 접근 시, **403 Forbidden** 페이지가 표시되며 차단 사유가 안내됩니다.



주의

Recovery Mode는 긴급 상황에서만 사용하세요. 평상시에는 TACHYON 대시보드를 통한 정상 접근을 권장합니다.



팁

비상 접근 IP는 tkadmin 내/외부 설정 메뉴에서 관리할 수 있습니다. 자세한 내용은 [내/외부 설정 편집기](#)를 참조하세요.

3.4 화면 구성 소개

tkadmin에 접속하면 다음과 같은 화면 구성을 확인할 수 있습니다:

1. 상단 헤더 바

화면 최상단에 위치한 헤더 영역입니다.

요소	설명
로고	tkadmin 제품 로고. 클릭 시 대시보드로 이동
알림 버튼	종 모양 아이콘. 미읽음 알림이 있으면 빨간색 배지에 개수가 표시됨
장애 경고 배너	장애 발생 시 상단에 빨간색 배너가 표시됨. 복구 시 자동으로 숨겨짐
나가기 버튼	클릭 시 TACHYON 운영 화면(/)으로 복귀

2. 좌측 사이드바

메뉴 네비게이션 영역입니다. 접힘(아이콘 모드)/펼침 상태를 전환할 수 있으며, 브라우저 `localStorage` 에 상태가 저장되어 새로고침 후에도 유지됩니다.

주요 메뉴 항목:

메뉴	설명
대시보드	시스템 상태 모니터링 및 서비스 리소스 점검
내/외부 설정	tkadmin 및 TACHYON 주요 설정 편집
전문가 편집기	Monaco Editor 기반 설정 파일 직접 편집
서비스 관리	systemd 기반 서비스 제어 (시작/중지/재시작)
환경 체크	OS 설정, SELinux, 방화벽 등 시스템 환경 점검
시스템 로그	시스템 로그 실시간 조회
관리자 보고	Watchdog 감지 이벤트 및 CLI 보고 이력



팁

브라우저 너비가 768px 이하일 경우 사이드바가 자동으로 아이콘 모드(Collapsed)로 전환됩니다.

3. 메인 콘텐츠 영역

사이드바에서 선택한 메뉴에 따라 해당 기능의 화면이 표시되는 영역입니다. 기본적으로 **대시보드** 화면이 표시됩니다.

3.5 나가기 버튼

헤더 우측의 **나가기** 버튼을 클릭하면 TACHYON 운영 화면(루트 경로 `/`)으로 복귀합니다. tkadmin의 세션은 유지되므로, 다시 접근하면 별도 인증 없이 바로 사용할 수 있습니다.

3.6 다음 단계

- [대시보드](#): 시스템 상태 모니터링 화면을 확인하세요.
- [내/외부 설정 편집기](#): 주요 설정 항목을 관리하는 방법을 배우세요.

4. 대시보드

4.1 개요

대시보드는 tkadmin의 메인 화면으로, TACHYON 솔루션의 운영 환경 및 설정 상태를 한눈에 확인할 수 있는 통합 모니터링 화면입니다. 접속 시 기본으로 표시되며, 사이드바의 '대시보드' 메뉴를 클릭하여 언제든지 돌아올 수 있습니다.

v0.2.0 — SSE 실시간 데이터 푸시 및 인프라 로드 최적화

대시보드의 모든 데이터는 ****SSE(Server-Sent Events)****를 통해 실시간으로 갱신됩니다. 서버에서 상태가 변경될 때만 데이터가 실시간으로 Push되어 네트워크 오버헤드가 기존 폴링 방식 대비 **99% 감소**했습니다. 또한 리눅스 **cgroup v2**를 직접 읽어 프로세스 리소스를 측정함으로써 CPU 부하를 **79% 절감**하여 저사양 환경에서도 안정적으로 동작합니다.

4.2 상단 장애 알림 배너

대시보드 상단(헤더 바 내부)에는 장애 경고 배너가 위치합니다.

상태	동작
장애 발생	빨간색 배너가 나타나며, 장애가 발생한 서비스명과 메시지가 표시됩니다. 배너를 클릭하면 해당 장애의 상세 리포트로 이동합니다.
복구 완료	동일 서비스에 대해 복구 성공(RECOVERY_SUCCESS) 알림이 수신되면, 배너가 자동으로 숨겨 집니다.
정상	활성 장애가 없으면 배너가 표시되지 않습니다.

**주의**

배너가 사라지더라도 알림 버튼의 빨간 बै지와 관리자 보고 목록의 미읽음 상태는 유지됩니다. 관리자가 직접 "읽음" 처리해야 बै지가 사라집니다. 이는 장애 이력의 감사 추적성을 보장하기 위한 정책입니다.

4.3 알림 버튼 및 बै지

헤더 우측의 종 모양 알림 버튼에는 미읽음 알림 개수를 표시하는 빨간색 बै지가 표시됩니다.

1. 미읽음 알림이 있으면 बै지에 숫자가 표시됩니다 (최대 **99+**).
2. 알림 버튼을 클릭하면 팝오버 형태의 알림 목록이 표시됩니다.
3. 목록에서 '**모두 읽음**' 버튼을 클릭하면 모든 알림이 읽음 처리되고 बै지가 사라집니다.

알림 시스템은 **10초 주기**로 서버에 새로운 알림이 있는지 자동으로 확인합니다.

4.4 카드 섹션

대시보드는 여러 개의 정보 카드로 구성되어 있습니다. 각 카드의 역할을 상세히 설명합니다.

1. 라이선스 / 기관 정보

TACHYON 솔루션의 라이선스 및 기관 정보를 표시합니다.

항목	설명
회사명	설치된 솔루션의 회사(기관)명
기관 코드	라이선스에 등록된 고유 기관 코드
에이전트 수	라이선스에서 허용된 에이전트 설치 가능 수량

이 정보는 TACHYON의 `app_info.properties_dev` 파일에서 자동으로 크롤링됩니다. 정보를 습득하지 못한 경우 **N/A** 로 표시됩니다.

2. 시스템 가동 시간

서버의 가동 시간(Uptime)을 표시합니다.

항목	설명
Uptime	서버가 마지막으로 부팅된 이후 경과한 시간 (예: 5d 12h 30m)
설정 요약	현재 적용 중인 솔루션 이름 (예: TACHYON Admin)


팁

Uptime은 `/proc/uptime` 을 직접 읽어 표시하며, 외부 명령어(`uptime`)에 의존하지 않습니다.

3. CPU 사용률

서버의 실시간 CPU 사용률을 프로그레스 바와 함께 표시합니다.

요소	설명
사용률 수치	현재 CPU 사용률 (% , 소수점 1자리)
프로그레스 바	사용률에 따라 너비가 변하는 시각적 표시
색상 변화	정상(초록/파랑) → 주의(노랑) → 경고(빨강)으로 단계별 색상 전환

프로그레스 바는 부드러운 애니메이션(0.5초 전환)으로 갱신되어 시각적 안정감을 제공합니다.

4. 메모리 상태

서버의 메모리 사용 현황을 상세하게 표시합니다.

항목	설명	예시
사용량 / 전체	현재 사용 중인 메모리와 전체 메모리	12.5 / 32.0 GB (39%)
Free	여유(사용 가능) 메모리	19.5 GB
Total	전체 물리 메모리	32.0 GB
프로그레스 바	사용률에 따른 시각적 표시	-

5. 디스크 사용량

서버의 디스크 사용 현황을 루트 디스크와 제품 디렉토리별로 구분하여 표시합니다.

상단 요약 영역:

항목	설명
Root	루트(/) 파티션의 사용량 / 전체 / 사용률 (예: 45.2 GB / 100 GB (45.2%))

하위 디렉토리별 용량:

TACHYON 설치 경로(/usr/local/TACHYON/TTS40/) 아래의 주요 디렉토리별 사용량이 목록으로 표시됩니다:

디렉토리	설명
TACHYON	솔루션 메인 디렉토리
MariaDB	데이터베이스 데이터 디렉토리
OpenSearch	검색 엔진 데이터 디렉토리
Kafka	메시지 브로커 데이터 디렉토리
Redis	인메모리 캐시 데이터 디렉토리

각 항목은 **디렉토리 이름**, **전체 경로**(마우스 오버 시 툴팁), **사용 용량**이 표시됩니다.



팁

디스크 사용량은 `unix.Statfs` 와 `filepath.WalkDir` 을 통해 직접 계산되며, `df` 나 `du` 같은 외부 명령어에 의존하지 않습니다.

6. 서비스 리소스 상태 점검

TACHYON 솔루션에 포함된 모든 서비스의 실시간 리소스 상태를 테이블 형태로 표시합니다.

테이블 컬럼

컬럼	설명
SERVICE	서비스 유닛 이름 (예: <code>tachyon-shield</code> , <code>mariadb</code> , <code>nginx</code> 등)
STATUS	서비스 활성 상태 (<code>Active</code> = 초록색 배지, <code>Exited</code> = 주황색 배지 + ? 힌트, <code>Inactive</code> = 빨간색 배지 + ? 힌트)
PID	프로세스 ID (비활성 시 -)
CPU(%)	해당 서비스의 CPU 사용률
MEMORY	해당 서비스의 메모리 사용량 (예: <code>256.4 MB</code>)

테이블 기능

- 컬럼 정렬:** 각 컬럼 헤더를 클릭하면 오름차순/내림차순으로 정렬됩니다. 현재 정렬 상태는 화살표 아이콘으로 표시됩니다.
- 컬럼 너비 조절:** 컬럼 경계를 드래그하여 너비를 자유롭게 조절할 수 있습니다.
- 정렬 초기화:** 좌측 상단의 회전 화살표 버튼(↻)을 클릭하면 정렬이 기본 상태(이름 오름차순)로 초기화됩니다.
- 수동 새로고침:** 우측 상단의 새로고침 버튼을 클릭하면 즉시 데이터를 갱신합니다.
- Total Services:** 테이블 우측 상단에 전체 서비스 수가 표시됩니다.

4.5 자동 갱신

대시보드의 데이터 갱신 방식은 다음과 같습니다:

항목	갱신 방식	비고
대시보드 전체 데이터	SSE (실시간 Push)	변경 사항 발생 시 즉시 전송. 가장 효율적인 통신 방식.
시스템 리소스 (CPU/MEM)	SSE (실시간 Push)	cgroup v2 직접 읽기로 초정밀 실시간 업데이트.
알림 및 배지	10초 폴링	시스템 안정성을 위해 별도 주기로 동작.

탭 비활성 시 동작

브라우저 탭이 백그라운드로 전환되면:

- SSE 연결이 자동으로 **해제**됩니다 (서버 리소스 절약).
- 모든 폴링 타이머가 **정지**됩니다.

3. 탭으로 복귀하면 SSE가 자동 **재연결**되고 즉시 최신 데이터를 수신합니다.

대시보드를 벗어나 다른 메뉴로 이동하면 SSE 연결 및 자동 갱신 타이머가 정리되어 불필요한 API 호출이 발생하지 않습니다.

4.6 조작 순서 요약

1. TACHYON 대시보드 또는 직접 URL을 통해 tkadmin에 접속합니다.
 2. 대시보드가 기본 화면으로 자동 표시됩니다.
 3. 상단 카드에서 라이선스, 시스템 가동 시간, CPU, 메모리, 디스크 상태를 확인합니다.
 4. 하단 서비스 테이블에서 각 서비스의 상태와 리소스 사용량을 확인합니다.
 5. 장애가 감지되면 상단 빨간색 배너와 알림 배지를 통해 즉시 인지할 수 있습니다.
 6. 알림 버튼을 클릭하여 상세 알림 내역을 확인하고, 필요 시 '모두 읽음' 처리합니다.
-

4.7 다음 단계

- [내/외부 설정 편집기](#): tkadmin의 주요 설정을 관리하는 방법을 확인하세요.

5. 내/외부 설정 편집기

5.1 개요

내/외부 설정 메뉴는 tkadmin의 핵심 설정 항목을 웹 UI에서 직접 편집할 수 있는 폼 기반 편집기입니다. 사이드바에서 '**내/외부 설정**' 메뉴를 클릭하면 접근할 수 있습니다.

이 화면에서 편집하는 설정은 `tkadmin.yml` 파일에 저장되며, 솔루션 이름, 네트워크 포트, 비상 접근 IP, 관리콘솔 통합 설정, 로깅 설정 등을 관리합니다.

5.2 설정 항목 상세

솔루션 이름

항목	내용
필드명	솔루션 이름
설명	tkadmin이 관리하는 TACHYON 솔루션의 표시 이름
기본값	TACHYON Admin
영향 범위	대시보드 설정 요약 영역에 표시됨

네트워크 및 보안 설정

Listen 주소

항목	내용
필드명	Listen 주소
설명	tkadmin 서버가 바인딩할 네트워크 주소
기본값	0.0.0.0 (모든 인터페이스에서 수신)

포트

항목	내용
필드명	포트
설명	tkadmin 메인 서비스의 리스닝 포트
기본값	13700
특이사항	변경 시 서비스 재시작이 필요합니다

포트 필드 옆에는 '재시작 필요' 태그가 노란색 뱃지로 표시되어, 이 값을 변경하면 서비스 재시작이 필요하다는 점을 시각적으로 안내합니다.



주의

포트를 변경하면 NGINX 리버스 프록시 설정도 자동으로 동기화됩니다. 변경 후 서비스가 재시작되면 새 포트로 접속해야 합니다.

긴급 접속 허용 IP (Emergency IPs)

항목	내용
필드명	긴급 접속 허용 IP (Emergency IPs)
설명	장애 시 인증을 우회하여 접근할 수 있는 IP 주소 목록
입력 형식	쉼표(,)로 구분하여 복수 IP 입력
예시	192.168.0.50, 10.0.0.99

이 IP 목록에 등록된 주소에서는 다음 권한이 부여됩니다:

- **메인 서비스:** TACHYON 인증 없이 emergency-admin 권한으로 자동 접근
- **Recovery 서비스:** Recovery 포트(13701) 접근 허용

현재 IP 자동 추가 기능

필드 우측에 현재 접속 중인 클라이언트 IP가 표시되며, '+' 버튼을 클릭하면 해당 IP가 목록에 자동으로 추가됩니다.

1. 현재 IP가 우측에 파란색으로 표시됩니다 (예: **현재 IP: 192.168.1.100**).
2. '+' 아이콘 버튼을 클릭합니다.
3. IP가 입력 필드에 자동으로 추가되고, 성공 토스트 메시지가 표시됩니다.
4. 이미 목록에 포함된 IP인 경우, 중복 안내 메시지가 표시됩니다.


팁

IP 추가 시 클립보드에도 자동으로 복사되어, 다른 곳에 붙여넣기할 수 있습니다.

설치 경로 (Target Dir)

항목	내용
필드명	설치 경로 (Target Dir)
설명	TACHYON 솔루션이 설치된 루트 디렉토리 경로
기본값	<code>/usr/local/TACHYON/TTS40/</code>
영향 범위	환경 정보 크롤링, 디스크 사용량 계산, 전문가 편집기 파일 탐색 범위

관리콘솔 통합 설정

TACHYON 대시보드와의 통합 연동을 제어하는 설정입니다.

링크 노출 여부 (show_link)

항목	내용
필드명	TACHYON 대시보드에 tkadmin 링크 노출
타입	체크박스 (켜짐/꺼짐)
기본값	꺼짐 (<code>false</code>)
설명	활성화하면 TACHYON 대시보드의 사이드바와 시스템 설정 페이지에 tkadmin 접근 링크가 노출됩니다


팁

이 설정은 인젝터 스크립트(`tkadmin_injector.js`)의 동작을 제어합니다. 비활성화해도 직접 URL 접근(`/tkadmin/`)은 가능합니다.

접속 허용 관리자 ID (allowed_ids)

항목	내용
필드명	접속 허용 관리자 ID (Allowed IDs)
설명	tkadmin에 접근할 수 있는 TACHYON 통합 UI 계정 ID 목록
입력 형식	쉼표(,)로 구분하여 복수 ID 입력
기본값	<code>tsadmin</code>
예시	<code>inca, admin, tsadmin</code>

비워두면 모든 TACHYON 관리자가 tkadmin에 접속할 수 있습니다.

시스템 로깅 및 감사 로그 설정

tkadmin 자체의 로깅 정책을 설정합니다.

항목	설명	기본값
로그 파일 경로	시스템 로그가 저장될 파일 경로	<code>logs/tkadmin.log</code>
로그 레벨	로깅 레벨 (Debug / Info / Warn / Error)	<code>Info</code>
최대 파일 크기 (MB)	로그 파일 하나의 최대 크기, 초과 시 로테이션	<code>10</code> MB
보관 파일 개수	로테이션된 백업 파일의 최대 보관 개수	<code>5</code> 개
보관 일수	로그 파일의 최대 보존 기간	<code>30</code> 일

하단의 **감사 로그 (Audit DB)** 안내 영역에는 관리자 보고 및 장애 이력이 `app.db` 파일에 영구 보관된다는 안내가 표시됩니다.

5.3 설정 파일 위치

모든 설정은 다음 파일에 YAML 형식으로 저장됩니다:

```
/usr/local/TACHYON/TTS40/tkadmin.yml
```

설정 파일의 경로는 바이너리 실행 파일의 이름을 기반으로 자동 결정됩니다. 예를 들어 바이너리 이름이 `tkadmin` 이면 같은 디렉토리의 `tkadmin.yml` 이 설정 파일이 됩니다.

5.4 클라이언트 IP 표시 기능

내/외부 설정 화면에서는 현재 접속 중인 클라이언트의 IP 주소가 '**현재 IP**' 레이블과 함께 표시됩니다. 이 정보는 서버의 API 응답(`client_ip` 필드)에서 제공되며, 비상 접근 IP를 등록할 때 참고 용도로 활용됩니다.

5.5 저장 플로우

일반 저장 (포트 변경 없음)

1. 설정 항목을 수정합니다.
2. 하단의 '**설정 저장**' 버튼을 클릭합니다.
3. 서버에 설정이 저장되고, '**설정이 저장되었습니다!**' 토스트 메시지가 화면 상단에 표시됩니다.
4. 토스트 메시지는 3초 후 자동으로 사라집니다.
5. 설정 품이 저장된 최신 값으로 자동 갱신됩니다.

포트 변경 시 저장

포트 값을 변경한 후 저장하면, 서비스 재시작이 필요하므로 추가 확인 절차가 진행됩니다.

1. 포트 값을 변경합니다 (예: `13700` → `13800`).
2. '**설정 저장**' 버튼을 클릭합니다.
3. **확인 모달**이 표시됩니다:
 - 제목: '**서비스 재시작 필요**'
 - 메시지: '**포트 설정을 변경하면 서비스가 재시작됩니다. 저장하시겠습니까?**'
 - 버튼: '**취소**' / '**확인**'
4. *****확인*****을 클릭하면:

- 설정이 저장됩니다.
- NGINX 프록시 설정이 새 포트로 자동 동기화됩니다.
- '서비스 재시작 중...' 토스트가 표시됩니다.
- 서비스가 재시작되며, 약 3초 후 페이지가 자동으로 새로고침됩니다.

5. '**취소'**를 클릭하면:

- '저장이 취소되었습니다.' 토스트가 표시됩니다.
- 설정 품이 원래 값으로 복원됩니다 (변경 전 상태로 되돌림).

저장 실패 시

서버에서 설정 저장에 실패하면, 에러 모달이 표시되며 실패 원인이 안내됩니다.

5.6 설정 취소

하단의 '취소' 버튼을 클릭하면, 수정한 내용이 모두 폐기되고 설정 품이 서버에 저장된 원래 값으로 다시 로드됩니다.

5.7 조작 순서 요약

일반 설정 변경

1. 사이드바에서 '내/외부 설정' 메뉴를 클릭합니다.
2. 변경하고자 하는 설정 항목을 수정합니다.
3. '설정 저장' 버튼을 클릭합니다.
4. 성공 토스트 메시지를 확인합니다.

비상 접근 IP 추가

1. '내/외부 설정' 메뉴로 이동합니다.
2. '긴급 접속 허용 IP' 필드에서 현재 IP 옆의 '+' 버튼을 클릭합니다.
3. 또는 직접 IP를 심표로 구분하여 입력합니다.
4. '설정 저장' 버튼을 클릭합니다.

포트 변경

1. '내/외부 설정' 메뉴로 이동합니다.
2. '포트' 필드의 값을 변경합니다.
3. '설정 저장' 버튼을 클릭합니다.
4. 확인 모달에서 '**확인**'을 클릭합니다.
5. 서비스 재시작이 완료될 때까지 약 3초 대기합니다.
6. 페이지가 자동으로 새로고침되면, 새 포트로 정상 접속되었는지 확인합니다.

관리콘솔 통합 설정

1. '내/외부 설정' 메뉴로 이동합니다.
 2. 'TACHYON 대시보드에 tkadmin 링크 노출' 체크박스를 켜거나 끕니다.
 3. 필요 시 '접속 허용 관리자 ID' 목록을 수정합니다.
 4. '설정 저장' 버튼을 클릭합니다.
-

5.8 다음 단계

- [대시보드](#)로 돌아가 시스템 상태를 확인하세요.

6. 전문가 편집기

6.1 개요

전문가 편집기는 TACHYON 솔루션 전반의 설정 파일을 웹 브라우저에서 직접 편집할 수 있는 고급 기능입니다. Microsoft의 오픈소스 코드 편집기인 **Monaco Editor**를 내장하여, VS Code에 준하는 편집 경험을 제공합니다.

tkadmin의 **자동 파일 탐색 엔진(Crawler)**이 **TargetDir** 하위 디렉토리를 스캔하여 발견한 설정 파일들을 좌측 파일 트리에 자동으로 표시하며, 운영자는 클릭 한 번으로 해당 파일을 열고 편집할 수 있습니다.

6.2 화면 구성

전문가 편집기는 크게 두 영역으로 나뉩니다:

영역	설명
좌측 - 파일 탐색기	Crawler가 발견한 설정 파일 목록을 카테고리별로 표시
우측 - Monaco Editor	선택된 파일의 내용을 구문 하이라이팅과 함께 편집

6.3 파일 트리 탐색

자동 파일 탐색 엔진 (Crawler)

tkadmin은 **tkadmin.yml**에 정의된 **target_dir** (기본값: **/usr/local/TACHYON/TTS40/**) 하위를 재귀적으로 탐색하여 설정 파일을 자동으로 발견합니다.

지원 파일 형식

확장자	설명
<code>.yaml</code>	YAML 설정 파일
<code>.yaml_dev</code>	YAML 개발/운영 오버라이드 파일
<code>.properties</code>	Java Properties 형식 설정
<code>.properties_dev</code>	Properties 개발/운영 오버라이드 파일
<code>.json</code>	JSON 형식 설정
<code>.conf</code>	NGINX 등 시스템 설정 파일

제외 폴더

다음 디렉토리는 탐색에서 자동 제외됩니다:

- `.git` - Git 저장소 메타데이터
- `node_modules` - Node.js 의존성
- `logs` - 로그 파일 디렉토리
- `temp` - 임시 파일 디렉토리
- `backup` - 백업 파일 디렉토리
- `dist` - 빌드 결과물 디렉토리

카테고리 분류

발견된 파일은 다음 기준에 따라 자동으로 분류됩니다:

카테고리	조건	예시
GLOBAL	<code>conf</code> 폴더 내부에 위치한 전역 설정 파일	<code>conf/app_info.properties_dev</code>
SERVICE	서비스 명칭이 포함된 개별 서비스 설정	<code>dist/tachyon-server/application.yaml_dev</code>
SYSTEM	<code>nginx</code> 등 시스템 관련 설정	<code>nginx/tkadmin.conf</code>

6.4 파일 열기 및 편집

파일 열기 순서

1. 좌측 **파일 탐색기**에서 편집할 파일을 클릭합니다.
2. 선택된 파일이 강조 표시되며, 우측 **Monaco Editor**에 파일 내용이 로드됩니다.
3. 상단 **에디터 툴바**에 현재 편집 중인 파일 경로와 **저장 (Ctrl+S)** 버튼이 표시됩니다.

편집 기능

Monaco Editor는 다음과 같은 풍부한 편집 기능을 제공합니다:

기능	설명
구문 하이라이팅	YAML, Properties, JSON, NGINX 설정 등 파일 확장자에 따라 자동 적용
코드 접기/펼치기	블록 단위로 코드를 접거나 펼쳐 가독성 향상
자동 완성	편집 중인 언어의 키워드 자동 제안
공백 문자 표시	들여쓰기 오류를 방지하기 위해 공백/탭 문자 시각적 표시

언어 자동 감지

파일 확장자에 따라 편집기 언어 모드가 자동 설정됩니다:

확장자	언어 모드
<code>.yaml</code> , <code>.yaml_dev</code>	YAML
<code>.properties</code> , <code>.properties_dev</code>	Properties
<code>.json</code>	JSON
<code>.conf</code>	NGINX
기타	Plain Text

단축키

단축키	동작
<code>Ctrl+S</code> (Windows/Linux) / <code>Cmd+S</code> (macOS)	파일 저장

6.5 저장 프로세스

파일 저장 시 다음 단계가 자동으로 수행됩니다:

저장 순서

1. **저장 (Ctrl+S)** 버튼을 클릭하거나, **Ctrl+S** 단축키를 누릅니다.
2. (YAML 파일인 경우) 서버에서 **YAML 구문 유효성 검사**가 수행됩니다.
 - 문법 오류가 있으면 저장이 중단되고 오류 메시지가 모달로 표시됩니다.
3. 기존 파일이 **자동 백업**됩니다.
 - 백업 파일명 형식: **원본파일명.YYYYMMDDHHMMSS.bak**
 - 예시: **application.yml** → **application.yml.20250615143022.bak**
4. 새 내용이 파일에 저장됩니다.
5. **감사 로그**에 변경 이력이 기록됩니다.
 - 기록 형식: **사용자 [GUID]가 설정 파일 '경로'을(를) 수정했습니다.**
6. 성공 알림 토스트가 표시됩니다: **설정이 저장되었습니다. (백업 생성됨)**

YAML 구문 검증 실패 시

YAML 파일 저장 시 구문 오류가 발견되면:

1. 저장이 즉시 **중단**됩니다 (파일이 변경되지 않음).
2. 오류 모달에 상세 오류 메시지가 표시됩니다.
 - 예시: **YAML 문법 오류: yaml: line 15: did not find expected key**
3. 오류를 수정한 후 다시 저장을 시도합니다.

6.6 보안

Path Traversal 방지

전문가 편집기는 **TargetDir** 외부의 파일에 접근할 수 없도록 강력한 경로 보안 검증을 수행합니다:

- 상대 경로 탐색(**..**) 사용 차단
- 절대 경로(**/** 또는 **** 로 시작) 접근 차단
- 위반 시 HTTP 403 Forbidden 응답 반환

이를 통해 운영자가 실수로 또는 의도적으로 시스템 파일에 접근하는 것을 원천 차단합니다.

6.7 주의사항

**주의**

설정 파일을 수정한 후에는 해당 서비스를
재시작
해야 변경 내용이 적용됩니다. 서비스 재시작은
서비스 관리
메뉴에서 수행할 수 있습니다.

**주의**

YAML 파일의 들여쓰기(Indentation)는 반드시
스페이스
를 사용해야 합니다. 탭 문자를 사용하면 구문 오류가 발생합니다. Monaco Editor에서 공백 문자가 점
(.)으로 표시되므로 이를 참고하세요.

**팁**

저장 시 자동 생성되는 **.bak** 백업 파일을 활용하면, 설정 변경 후 문제가 발생했을 때 이전 상태로 빠르게 원복할 수 있습니다.

**팁**

tkadmin은 폐쇄망(Air-gap) 환경을 지원합니다. Monaco Editor를 포함한 모든 외부 라이브러리가 바
이너리에 내장되어 있어, 인터넷 연결 없이도 정상 동작합니다.

6.8 관련 API

API	메서드	설명
<code>/tkadmin/api/config/external/files</code>	GET	Crawler가 발견한 설정 파일 목록 조회
<code>/tkadmin/api/config/external/file?path=...</code>	GET	특정 설정 파일 내용 조회
<code>/tkadmin/api/config/external/file</code>	POST	설정 파일 저장 (백업 + 검증 + 감사 로그 포함)

| 6.9 다음 단계

- [서비스 관리](#)에서 설정 변경 후 서비스를 재시작하세요.
- [시스템 로그](#)에서 변경 이력을 확인할 수 있습니다.

7. 설정 파일 레퍼런스

tkadmin의 모든 설정은 `tkadmin.yml` 파일을 통해 관리됩니다. 본 문서는 설정 파일의 전체 옵션과 환경 탐색(Discovery) 전략을 상세히 설명합니다.

7.1 설정 파일 경로

설정 파일은 표준 설치 시 `conf/` 서브디렉토리에 위치합니다.

- 표준 설치 경로: `/usr/local/tkadmin/conf/tkadmin.yml`
- 탐색 우선순위:
 - `conf/tkadmin.yml` (표준 설치 경로)
 - 실행 파일(`.exe`)과 동일한 경로의 `.yml` 파일 (개발/포터블 모드)
 - 현재 실행 디렉토리의 `tkadmin.yml` (Fallback)

설정 파일이 존재하지 않으면 내장된 기본값이 사용되며, 웹 UI를 통해 저장하면 파일이 자동 생성됩니다.

7.2 전체 설정 옵션

기본 설정

키	설명	기본값	허용 범위	재시작 필요
<code>target_dir</code>	TACHYON 설치 경로	<code>/usr/local/TACHYON/TTS40/</code>	유효한 디렉토리 경로	예
<code>listen_addr</code>	서버 바인딩 주소	<code>0.0.0.0</code>	유효한 IP 주소	예
<code>port</code>	메인 서버 포트	<code>13700</code>	1024 ~ 65535	예
<code>recovery_port</code>	Recovery Mode 서버 포트	<code>port + 1</code> (13701)	1024 ~ 65535	예
<code>solution_name</code>	솔루션 표시 이름	<code>TACHYON Admin</code>	임의 문자열	아니오
<code>debug</code>	디버그 모드 활성화	<code>false</code>	<code>true</code> / <code>false</code>	아니오

보안 설정

키	설명	기본값	허용 범위	재시작 필요
<code>emergency_ips</code>	Recovery Mode 접근 허용 IP 목록	<code>[]</code> (빈 배열)	IP 주소 문자열 배열	아니오
<code>show_link</code>	TACHYON 대시보드 tkadmin 메뉴 표시 여부	<code>false</code>	<code>true</code> / <code>false</code>	아니오
<code>allowed_ids</code>	tkadmin 메뉴가 표시될 TACHYON 사용자 ID 목록	<code>["tsadmin"]</code>	문자열 배열	아니오

로깅 설정

키	설명	기본값	허용 범위	재시작 필요
<code>log_path</code>	로그 파일 경로 (레거시)	<code>logs/tkadmin.log</code>	유효한 파일 경로	예
<code>logging.level</code>	로그 출력 레벨	<code>info</code>	<code>debug</code> , <code>info</code> , <code>warn</code> , <code>error</code>	아니오
<code>logging.file</code>	로그 파일 경로	<code>logs/tkadmin.log</code>	유효한 파일 경로 (파일명 포함 필수)	예
<code>logging.max_size</code>	로그 파일 최대 크기 (바이트)	<code>10485760</code> (10MB)	양의 정수	아니오
<code>logging.max_backups</code>	보관할 로그 백업 파일 수	<code>5</code>	0 이상 정수	아니오
<code>logging.max_age</code>	로그 보존 기간 (일)	<code>30</code>	0 이상 정수	아니오
<code>logging.compress</code>	로그 백업 파일 GZIP 압축	<code>true</code>	<code>true</code> / <code>false</code>	아니오

감사 로그 설정

키	설명	기본값	허용 범위	재시작 필요
<code>audit_log_retention_days</code>	감사 로그 보존 기간 (일)	<code>0</code> (영구 보존)	0 이상 정수	아니오

7.3 설정 파일 전체 예시

```
# tkadmin 설정 파일
target_dir: "/usr/local/TACHYON/TTS40/"
listen_addr: "0.0.0.0"
port: 13700
recovery_port: 13701
solution_name: "TACHYON Admin"
```

```

debug: false

# 보안 설정
emergency_ips:
  - "10.10.1.100"
  - "192.168.1.50"
show_link: true
allowed_ids:
  - "tsadmin"
  - "operator01"

# 감사 로그 보존 기간 (0 = 영구 보존)
audit_log_retention_days: 90

# 로깅 설정
logging:
  level: "info"
  file: "logs/tkadmin.log"
  max_size: 10485760      # 10MB (바이트 단위)
  max_backups: 5
  max_age: 30            # 30일
  compress: true

```

7.4 주요 설정 항목 상세

port

메인 HTTP 서버가 바인딩하는 포트 번호입니다.

- 기본값: 13700
- 포트를 변경하면 NGINX proxy_pass 설정도 자동으로 동기화됩니다.
- 포트 변경 시 반드시 서비스를 재시작해야 합니다.



팁

웹 UI에서 포트를 변경하고 저장하면, NGINX 설정이 자동으로 업데이트되고 재시작 확인 대화상자가 표시됩니다.

recovery_port

Recovery Mode 서버가 바인딩하는 포트 번호입니다.

- 기본값: port + 1 (메인 포트가 13700이면 13701)

- Recovery 서버는 메인 서버와 독립적으로 동작하는 별도의 Gin 인스턴스입니다.
- Linux 환경에서만 구동됩니다.

emergency_ips

Recovery Mode 접근이 허용되는 IP 주소 목록입니다.

- **기본값:** 빈 배열 `[]`
- `localhost` (127.0.0.1, ::1)는 항상 자동 허용됩니다.
- 이 목록에 없는 IP에서 Recovery 포트에 접근하면 403 Forbidden 페이지가 표시됩니다.
- 메인 서버의 인증 우회(Emergency Bypass)에도 동일한 IP 목록이 사용됩니다.

show_link / allowed_ids

TACHYON 대시보드에서 tkadmin 메뉴 표시를 제어합니다.

- `show_link: true` 설정 시 인젝터 스크립트를 통해 TACHYON 대시보드에 tkadmin 메뉴가 표시됩니다.
- `allowed_ids`에 포함된 사용자 ID로 로그인한 경우에만 메뉴가 보입니다.
- 이 설정들은 인증 없이 접근 가능한 `/tkadmin/api/nav-config` API를 통해 인젝터 스크립트에 전달됩니다.

logging

구조화된 로깅 시스템(Zap + Lumberjack)의 동작을 제어합니다.

- **로그 엔진:** go.uber.org/zap (고성능 구조화 JSON 로깅)
- **로테이션 엔진:** gopkg.in/natefinch/lumberjack.v2
- 콘솔(컬러 인쇄)과 파일에 동시 출력됩니다.

⚠ 주의

`logging.file` 값이 빈 문자열이거나 디렉토리 경로만 지정되면 치명적 버그가 발생할 수 있습니다.
반드시 파일명을 포함한 경로를 설정하세요.

7.5 환경 탐색(Discovery) 우선순위

tkadmin은 고정 설정에 의존하지 않고, TACHYON 서버의 운영 환경 정보를 동적으로 탐색하여 연결 정보를 결정합니다.

우선순위 체계

우선 순위	레벨	소스	설명
1	Level 1 (Direct)	CLI 인자 (Flags)	실행 시 명령줄로 명시된 설정값
2	Level 2 (Discovery)	운영 파일 크롤링	<code>app_info.properties_dev</code> , <code>*.yml_dev</code> 등에서 습득한 최신 정보
3	Level 3 (Internal Default)	소스 코드 내 상수	바이너리 내 하드코딩된 기본값 (최후의 Fallback)

탐색 대상 파일

- `conf/app_info.properties_dev` : 라이선스, DB 접속 정보, Redis 설정 등 핵심 메타데이터
- `*.yml_dev` : 각 마이크로서비스별 세부 연결 및 포트 정보

탐색 경로

TACHYON 설치 루트(`target_dir`) 하위의 다음 디렉토리를 탐색합니다:

- `conf/` : 공통 설정 파일 (Global 분류)
- `dist/` : 서비스 배포 디렉토리
- 기타 서비스 하위 디렉토리 (Service 분류)

설계 기초

원칙	설명
운영 환경 인지(Env-Aware)	Redis 비밀번호 변경 등 환경 변화를 사용자 개입 없이 자동 감지
안정한 실행 보장	설정 파일이 훼손되거나 없어도 내장 Fallback으로 최소 관리 기능 유지
자동 동기화	환경 정보를 메모리에 캐싱하되, 주기적으로 재스캔하여 변경 사항 반영



팁

`tkadmin.yml` 설정 파일이 삭제되거나 손상되더라도 tkadmin은 내장 기본값과 운영 환경 크롤링을 통해 정상적으로 구동됩니다. 이는 극한의 장애 상황에서도 관리 도구의 가용성을 보장하기 위한 설계입니다.

8. 서비스 관리

8.1 개요

서비스 관리 화면은 TACHYON 솔루션과 관련된 모든 systemd 서비스를 한눈에 확인하고, 시작/중지/재시작 등의 제어를 수행할 수 있는 기능입니다. 백그라운드에서는 **Watchdog 엔진**이 30초 주기로 서비스 상태를 감시하며, 장애 발생 시 의존성 순서에 따라 자동 복구를 시도합니다.

8.2 서비스 목록 표시

자동 탐색 대상

tkadmin은 다음 서비스들을 자동으로 탐색하여 목록에 표시합니다:

미들웨어 서비스 (Infrastructure)

서비스명	역할
<code>mariadb</code>	관계형 데이터베이스
<code>redis</code>	인메모리 캐시/세션 저장소
<code>zookeeper</code>	분산 코디네이션
<code>kafka</code>	메시지 브로커
<code>opensearch</code>	검색/로그 엔진
<code>logstash-kafka-os</code>	로그 파이프라인
<code>nginx</code>	웹 서버/리버스 프록시
<code>opensearch-dashboards</code>	로그 시각화 대시보드

애플리케이션 서비스 (TACHYON)

- `TACHYON-Auth1` , `TACHYON-Api1` 등 `/usr/lib/systemd/system/` 및 `/etc/systemd/system/` 경로에서 `TACHYON*.service` 패턴으로 자동 탐색

관리 서비스

- `tkadmin` (본 도구 자체 서비스)

상태 표시

각 서비스의 상태는 다음과 같이 시각적으로 구분됩니다:

상태	표시 색상	설명
Active (Running)	초록색	서비스가 정상 실행 중
Exited (완료)	주황색 + ? 힌트	초기 설정 등 1회성 작업을 마친 서비스
Inactive	회색	서비스가 중지된 상태
Failed	빨간색 + ? 힌트	서비스 실행 실패

리소스 정보

각 서비스에 대해 다음 리소스 정보가 실시간으로 표시됩니다:

항목	설명	데이터 출처
PID	프로세스 ID	<code>systemctl show -p MainPID</code>
CPU%	CPU 사용률 (코어 수 대비 정규화)	cgroup v2 직접 읽기 (<code>cpu.stat</code>) 또는 <code>ps</code> Fallback
Memory	메모리 사용량 (사람이 읽기 쉬운 형식)	cgroup v2 직접 읽기 (<code>memory.current</code>) 또는 <code>ps</code> Fallback

8.3 서비스 제어

제어 버튼

서비스 관리 화면에서 다음 세 가지 제어 버튼을 사용할 수 있습니다:

버튼	색상	동작	systemctl 명령
시작	초록색	중지된 서비스를 시작	<code>systemctl start <서비스명></code>
중지	빨간색	실행 중인 서비스를 중지	<code>systemctl stop <서비스명></code>
재시작	파란색 (Primary)	서비스를 재시작	<code>systemctl restart <서비스명></code>

서비스 제어 순서

1. 제어할 서비스를 확인합니다.
2. 원하는 제어 버튼(**시작**, **중지**, **재시작**)을 클릭합니다.
3. 확인 팝업이 표시됩니다: **서비스를 [동작] 하시겠습니까?**
4. **확인**을 클릭하면 서버에 제어 명령이 전송됩니다.
5. 실행 결과 메시지가 표시됩니다.

8.4 Watchdog 자동 복구

동작 원리

Watchdog 엔진은 tkadmin이 실행되는 동안 백그라운드에서 지속적으로 동작합니다:

1. **30초** 주기로 모든 TACHYON 관련 서비스의 상태를 점검합니다.
2. 서비스가 중단(**inactive**) 또는 실패(**failed**) 상태로 감지되면:
 - **FAILURE** 알림을 생성합니다.
 - 해당 서비스에 대해 **자동 재시작**을 시도합니다.
3. 재시작 후 **5초** 대기 후 상태를 재확인합니다.
 - 정상화 시: **RECOVERY_SUCCESS** 알림 생성
 - 여전히 비정상 시: **RECOVERY_FAILED** 알림 생성

의존성 기반 복구 순서

Watchdog은 서비스 간 의존성을 고려하여 **우선순위 순서**대로 복구를 시도합니다. 인프라 서비스가 먼저 복구되어야 애플리케이션 서비스가 정상 동작할 수 있기 때문입니다:

우선순위	서비스	역할
1	<code>mariadb</code>	데이터베이스
2	<code>redis</code>	캐시/세션
3	<code>zookeeper</code>	분산 코디네이션
4	<code>kafka</code>	메시지 브로커
5	<code>opensearch</code>	검색 엔진
6	<code>logstash-kafka-os</code>	로그 파이프라인
7	<code>nginx</code>	웹 서버
8	<code>opensearch-dashboards</code>	시각화
9+	<code>TACHYON-Auth1</code> , <code>TACHYON-Api1</code> ...	TACHYON 앱 서비스



팁

인프라 서비스(우선순위 1~8)가 복구된 후에는

10초간 안정화 대기

를 거친 뒤 다음 서비스의 상태를 점검합니다. 이를 통해 인프라가 완전히 기동된 상태에서 상위 애플리케이션이 복구되도록 보장합니다.

복구 결과 알림

Watchdog의 감지 및 복구 결과는 다음 방식으로 관리자에게 통보됩니다:

- **상단 빨간 배너:** Active 상태의 FAILURE 알림 시 표시
- **알림 배지:** 헤더 영역의 종 아이콘에 미읽음 건수 표시
- **관리자 보고:** 모든 이벤트가 [관리자 보고](#) 화면에 기록

8.5 Watchdog 오탐 방지

의도적 서비스 제어와 Watchdog 간의 충돌 방지

운영자가 웹 UI 또는 `tkcli` 를 통해 의도적으로 서비스를 중지했을 때, Watchdog이 이를 장애로 오인하여 자동 재시작하는 것을 방지하는 메커니즘이 내장되어 있습니다.

동작 방식

1. 웹 UI에서 서비스 제어 시:

- `ControlService` API 호출 후 `RecordIntentionalAction` 이 자동 실행됩니다.
- Watchdog이 해당 서비스의 종단을 감지하더라도, 의도적 조치로 기록되어 있으므로 알림 및 자동 재시작을 억제합니다.

2. tkcli에서 서비스 제어 시:

- tkcli는 먼저 `pending` 상태를 보고하여 의도적 조치를 **선등록**합니다.
- 이후 실제 서비스 제어를 수행하고, 완료 결과를 보고합니다.
- Watchdog은 pending 등록을 확인하고 알림을 억제합니다.

3. 의도적 플래그 만료:

- 등록된 의도적 조치 플래그는 **2분** 후 자동 만료됩니다.
- 만료 후에도 서비스가 중단 상태이면 Watchdog이 정상적으로 장애 알림 및 복구를 수행합니다.

자동 업데이트/재시작 시 Grace Period (Lazy Loading)



서비스가 자체적으로 설정 적용이나 업데이트를 위해 짧은 시간 동안 재시작될 수 있습니다. 이 경우 운영자의 명시적 조치 없이도 서비스가 잠시 중단되는데, Watchdog이 이를 장애로 오인하지 않도록 **Grace Period(유예 시간) 메커니즘**이 적용됩니다.

전환 상태 자동 감지

Watchdog은 systemd의 **SubState** 값을 분석하여 서비스가 자체 전환 중인지 판별합니다:

SubState	의미	Grace Period
auto-restart	systemd가 자동 재시작 중	15초
start	서비스 시작 진행 중	15초
stop	서비스 종료 진행 중 (재시작 전 단계)	15초
activating	서비스 활성화 전환 중	15초
deactivating	서비스 비활성화 전환 중	15초
reloading	설정 리로드 중	15초
dead	서비스가 완전히 종료됨	5초
failed	서비스 실행 실패	5초

동작 흐름

1. Watchdog이 서비스 종단을 감지합니다.
2. SubState가 전환 상태(`auto-restart` , `start` 등)이면 → **15초** 유예 시간을 부여합니다.
3. SubState가 `dead` / `failed` 이면 → 축소된 **5초** 유예 시간을 부여합니다.
4. 유예 시간 후 서비스 상태를 **재확인**합니다:
 -  복구됨 → 알림 없이 정상 통과 (Info 레벨 로그만 기록)
 -  여전히 비정상 → 기존 장애 처리 흐름 진입 (FAILURE 알림 + 자동 복구 시도)



팁

Grace Period 덕분에 서비스가 자체 업데이트로 인해 잠시(수 초간) 중단되더라도 불필요한 장애 알림이 발생하지 않습니다. 이는 운영자의 알림 피로도(Alert Fatigue)를 줄여줍니다.

8.6 주의사항

주의

`tkadmin` 자체 서비스를 중지하면 웹 인터페이스에 접속할 수 없게 됩니다. 자체 서비스 재시작 시에는 약 1~2초간 접속이 중단될 수 있습니다.

**주의**

인프라 서비스(MariaDB, Redis 등)를 중지하면, 이에 의존하는 TACHYON 애플리케이션 서비스도 함께 영향을 받을 수 있습니다. 인프라 서비스 중지 전 영향 범위를 반드시 확인하세요.

**팁**

서비스를 의도적으로 중지한 경우, 2분 이내에 다시 시작하면 Watchdog의 불필요한 알림을 피할 수 있습니다. 장시간 중지가 필요한 경우, 2분 후 Watchdog이 자동 재시작을 시도할 수 있으므로 유의하세요.

8.7 관련 API

API	메서드	설명
<code>/tkadmin/api/service/:action</code>	POST	서비스 제어 (start, stop, restart)
<code>/tkadmin/api/service/resources</code>	GET	서비스별 리소스(CPU, 메모리, PID) 조회
<code>/tkadmin/api/monitor/report</code>	POST	tkcli 외부 보고 수신

8.8 다음 단계

- [환경 체크](#)에서 OS 설정 및 방화벽 상태를 점검하세요.
- [관리자 보고](#)에서 Watchdog 장애/복구 이력을 확인할 수 있습니다.

9. 환경 체크

9.1 개요

환경 체크 화면은 TACHYON 솔루션이 운영되는 리눅스 서버의 핵심 시스템 설정을 점검하고 조정할 수 있는 기능입니다. OS Limits, SELinux, 방화벽(Firewalld) 설정을 웹 인터페이스에서 직접 확인하고 변경할 수 있어, 터미널 접속 없이도 운영 환경을 관리할 수 있습니다.



주의

환경 체크 기능은

Linux 전용

기능입니다. Windows 개발 환경에서는 더미(Mock) 데이터가 표시되며, 실제 시스템 변경은 수행되지 않습니다.

9.2 화면 구성

환경 체크 화면은 세 개의 카드로 구성됩니다:

영역	설명
OS Limits (limits.conf)	시스템 리소스 제한값 확인 및 설정 (화면 상단, 2열 너비)
SELinux 설정	SELinux 보안 정책 상태 확인 및 변경
방화벽 허용 정책	Firewalld 포트/서비스 허용 관리

우측 상단의 **데이터 새로고침** 버튼을 클릭하면 모든 영역의 데이터를 최신 상태로 갱신할 수 있습니다.

9.3 OS Limits 확인 및 설정

현재 설정 확인

`/etc/security/limits.conf` 파일에 정의된 현재 시스템 리소스 제한값이 테이블 형태로 표시됩니다.

컬럼	설명	예시
Domain	적용 대상 (사용자/그룹)	<code>*</code> (모든 사용자), <code>root</code>
Type	제한 유형	<code>soft</code> (경고 한계), <code>hard</code> (절대 한계)
Item	제한 항목	<code>nofile</code> (파일 열기 수), <code>nproc</code> (프로세스 수)
Value	제한값	<code>65535</code>

새 리미트 추가

테이블 하단의 입력 폼을 사용하여 새로운 리미트를 추가할 수 있습니다:

1. **Domain** 필드에 적용 대상을 입력합니다 (기본값: `*`).
2. **Type** 드롭다운에서 `soft` 또는 `hard` 를 선택합니다.
3. **Item** 필드에 제한 항목을 입력합니다 (기본값: `nofile`).
4. **Value** 필드에 제한값을 입력합니다 (기본값: `65535`).
5. 항목 추가 버튼을 클릭합니다.
6. 성공 시 토스트 메시지가 표시되고, 테이블이 자동 갱신됩니다.

자주 사용되는 설정 예시:

```
*      soft    nofile    65535
*      hard    nofile    65535
*      soft    nproc     65535
*      hard    nproc     65535
```



팁

TACHYON 솔루션은 대량의 파일 디스크립터를 사용하므로, `nofile` 값을 `65535` 이상으로 설정하는 것을 권장합니다. tkadmin의 systemd 서비스 유닛에는 `LimitNOFILE=65535` 가 기본 포함되어 있습니다.

9.4 SELinux 상태 제어

현재 상태 확인

SELinux의 현재 실행 상태와 부팅 설정이 표시됩니다:

- **현재 상태:** 실시간 적용 중인 SELinux 모드
- **부팅 설정:** `/etc/selinux/config`에 설정된 부팅 시 적용 모드

상태	표시 색상	설명
Enforcing	초록색	SELinux 정책을 강제 적용
Permissive	노란색	정책 위반을 로그에만 기록하고 차단하지 않음
Disabled	빨간색	SELinux가 비활성화된 상태

모드 변경

세 개의 버튼 중 원하는 모드를 클릭하여 SELinux 상태를 변경할 수 있습니다:

1. 변경할 모드 버튼(**Enforcing**, **Permissive**, **Disabled**)을 클릭합니다.
2. 확인 팝업이 표시됩니다: **SELinux 모드를 [모드명] (으)로 변경하시겠습니까?**
3. **확인**을 클릭하면 변경이 수행됩니다.
4. 변경 성공 시 토스트 메시지가 표시되고 화면이 갱신됩니다.

적용 방식:

- **즉시 적용:** `setenforce` 명령으로 현재 세션에 즉시 반영
- **영구 반영:** `/etc/selinux/config` 파일을 수정하여 재부팅 후에도 유지

⚠ 주의

Disabled 모드로 변경하면 전역 설정 파일(`/etc/selinux/config`)이 수정됩니다. 일부 환경에서는 완전한 비활성화를 위해 **시스템 재부팅**이 필요할 수 있습니다.

💡 팁

TACHYON 솔루션 운영 시 SELinux로 인한 권한 문제가 발생하면, 먼저 **Permissive** 모드로 변경하여 로그를 확인한 후 적절한 정책을 추가하는 것을 권장합니다.

9.5 방화벽(Firewalld) 포트 관리

현재 상태 확인

Firewalld의 활성화 여부와 현재 허용된 포트/서비스 목록이 표시됩니다:

- **활성화 상태:** Firewalld 구동 여부 (초록색 체크 아이콘 또는 비활성 안내)
- **허용 포트 목록:** 태그(Tag) 형태로 현재 열린 포트/서비스 표시

포트 추가

1. **유형 선택:** **Port** 또는 **Service** 중 선택합니다.
 - **Port:** 특정 포트/프로토콜 (예: **8080/tcp** , **443/tcp**)
 - **Service:** 사전 정의된 서비스명 (예: **http** , **https**)
2. 입력 필드에 포트 번호(프로토콜 포함) 또는 서비스명을 입력합니다.
3. **정책 허용 추가** 버튼을 클릭합니다.
4. 성공 시 토스트 메시지가 표시되고 포트 목록이 갱신됩니다.

포트 제거

1. 허용 포트 목록에서 제거할 포트 태그 옆의 **X** 아이콘을 클릭합니다.
2. 확인 팝업이 표시됩니다: **포트 [포트명] 허용 정책을 삭제하시겠습니까?**
3. **확인**을 클릭하면 해당 포트 허용 정책이 제거됩니다.
4. 성공 시 토스트 메시지가 표시되고 목록이 갱신됩니다.

TACHYON 솔루션 필수 포트 예시:

포트	용도
443/tcp	HTTPS (NGINX)
13700/tcp	tkadmin 메인 서비스
13701/tcp	tkadmin Recovery 서비스
3306/tcp	MariaDB
6379/tcp	Redis
9092/tcp	Kafka
9200/tcp	OpenSearch

9.6 주의사항

**주의**

OS Limits, SELinux, 방화벽 설정 변경은 시스템 전체에 영향을 미칩니다. 변경 전 현재 설정을 반드시 확인하고, 운영 환경에서는 유지보수 시간에 수행할 것을 권장합니다.

**주의**

방화벽에서 tkadmin 포트(**13700/tcp**)를 제거하면, 외부에서 웹 인터페이스에 접근할 수 없게 됩니다. 포트 제거 시 영향을 반드시 사전에 검토하세요.

**팁**

환경 체크 화면의 데이터가 최신이 아닌 것 같다면, 우측 상단의 **데이터 새로고침** 버튼을 클릭하여 서버에서 최신 정보를 다시 조회하세요.

9.7 관련 API

API	메서드	설명
<code>/tkadmin/api/system/os-checks</code>	GET	OS Limits, SELinux, 방화벽 통합 상태 조회
<code>/tkadmin/api/system/os-checks/limits</code>	POST	OS Limit 항목 추가
<code>/tkadmin/api/system/os-checks/selinux</code>	POST	SELinux 모드 변경
<code>/tkadmin/api/system/os-checks/firewall</code>	POST	방화벽 포트 추가/제거

9.8 다음 단계

- [시스템 로그](#)에서 시스템 로그를 실시간으로 모니터링하세요.
- [서비스 관리](#)에서 서비스 상태를 확인하세요.

10. 시스템 로그

10.1 개요

시스템 로그 화면은 tkadmin 애플리케이션의 시스템 로그를 웹 브라우저에서 실시간으로 확인할 수 있는 뷰어입니다. Zap + Lumberjack 기반의 구조화된 JSON 로그를 파싱하여 사람이 읽기 쉬운 테이블 형태로 제공하며, 강력한 필터링과 무한 스크롤을 통해 과거 로그까지 빠르게 탐색할 수 있습니다.

10.2 화면 구성

영역	설명
상단 툴바	레벨 필터, 검색어 입력, 검색 버튼, 실시간 상태 표시
로그 테이블	시간, 레벨, 메시지, 호출자(Caller) 컬럼으로 구성된 로그 데이터

테이블 컬럼

컬럼	설명	기본 너비
확장	구조화된 데이터(Fields)가 있는 경우 토글 버튼 표시	40px
시간	로그 기록 시각 (HH:MM:SS.mmm 형식)	200px
레벨	로그 레벨 배지 (INFO, WARN, ERROR, DEBUG)	80px
메시지	로그 본문 내용	500px
호출자	로그를 기록한 소스 코드 위치	160px



팁

컬럼 헤더의 경계선을

마우스로 드래그

하면 컬럼 너비를 자유롭게 조절할 수 있습니다.

10.3 실시간 로그 추적

자동 폴링

시스템 로그 화면에 진입하면 **3초 주기**로 새로운 로그를 자동으로 감지하여 표시합니다.

항목	값
폴링 주기	3초
표시 위치	테이블 최상단에 새 로그 삽입
상태 표시	툴바 우측의 초록색 점 + "실시간 업데이트 중" 텍스트

동작 방식

1. 화면 진입 시 최근 로그 100건을 최초 로드합니다.
2. 이후 3초마다 마지막으로 수신한 로그의 타임스탬프(**lastTs**) 이후의 새 로그만 조회합니다.
3. 새 로그가 발견되면 테이블 최상단에 자동으로 삽입합니다.
4. 다른 메뉴로 이동하면 폴링 타이머가 자동 해제됩니다.

10.4 필터링 기능

레벨별 필터

상단 툴바의 드롭다운에서 원하는 로그 레벨을 선택하면, 해당 레벨의 로그만 필터링하여 표시합니다.

레벨	색상	설명
모든 레벨	-	필터 없이 전체 로그 표시 (기본값)
INFO	파란색	일반적인 정보 로그
WARN	노란색	경고 수준의 로그
ERROR	빨간색	오류 발생 로그
DEBUG	회색	디버그 모드 활성화 시의 상세 로그

**팁**

프로덕션 환경에서는 기본 로그 레벨이 **INFO** 로 설정되어 있습니다. **DEBUG** 로그를 보려면 `tkadmin.yml` 의 `debug` 옵션을 `true` 로 변경해야 합니다.

시간대 필터

API 수준에서 `since` 와 `until` 파라미터를 지원하여 특정 시간 범위의 로그만 조회할 수 있습니다:

파라미터	설명
<code>since</code>	이 시각 이후의 로그만 조회 (실시간 폴링에 자동 사용)
<code>until</code>	이 시각 이전의 로그만 조회 (과거 로그 로드에서 자동 사용)

검색어 필터

1. 상단 툴바의 **검색 입력 필드**에 검색어를 입력합니다.
2. **검색** 버튼을 클릭하거나 **Enter** 키를 누릅니다.
3. 대소문자를 **구분하지 않고** 로그 전체 내용에서 매칭되는 항목만 표시합니다.

검색어 하이라이팅

검색어와 일치하는 텍스트에는 **시각적 강조 효과**가 자동 적용됩니다. 이를 통해 방대한 로그 메시지에서 원하는 키워드를 빠르게 식별할 수 있습니다.

필터 조합

레벨 필터와 검색어 필터는 **동시에 적용**할 수 있습니다. 예를 들어:

- 레벨: **ERROR** + 검색어: **mariadb** → MariaDB 관련 오류 로그만 표시

10.5 과거 로그 조회 (무한 스크롤)

동작 방식

1. 로그 테이블을 아래로 스크롤합니다.
2. **스크롤이 최하단에 도달**하면 (50px 이내) 자동으로 과거 로그를 요청합니다.
3. 현재 표시된 로그 중 가장 오래된 타임스탬프(**oldestTs**) 이전의 로그 100건을 추가 로드합니다.
4. 로드된 과거 로그는 테이블 하단에 추가됩니다.
5. 더 이상 로드할 로그가 없을 때까지 반복할 수 있습니다.

회전된 압축 로그 파일 지원

tkadmin의 로그 시스템(Lumberjack)은 로그 파일을 자동 회전(Rotate)하며, 오래된 파일은 GZIP으로 압축합니다:

항목	설정값
파일당 최대 크기	10MB
보관 백업 수	5개
보존 기간	30일
압축 형식	GZIP (.gz)

무한 스크롤 시 현재 활성 로그 파일의 데이터가 소진되면, **회전된 .gz 압축 파일**도 자동으로 스트림 해제하여 조회합니다. 이를 통해 과거 수일 ~ 수주 전의 로그까지 끊임 없이 탐색할 수 있습니다.

10.6 로그 상세 데이터 확장

각 로그 항목에 JSON 형태의 구조화된 추가 데이터(Fields)가 포함된 경우, 해당 행의 좌측에 **확장 토글 버튼** (▶)이 표시됩니다.

조회 방법

1. 확장 가능한 로그 행의 좌측 ▶ 버튼을 클릭합니다.
2. 행 아래에 Fields 데이터가 JSON 포맷으로 펼쳐집니다.
3. 다시 클릭하면 접힙니다.

Fields에는 다음과 같은 추가 정보가 포함될 수 있습니다:

- 오류 스택트레이스
- 요청 파라미터
- 서비스/모듈 식별 정보
- 성능 측정 데이터

10.7 성능

시스템 로그 뷰어는 대용량 로그 파일에서도 빠른 응답 속도를 보장하기 위해 다음과 같은 최적화가 적용되어 있습니다:

기법	설명
Seek + Tail	파일 끝에서부터 역방향으로 읽어 최신 로그를 우선 로드 (64KB 버퍼 단위)
스트리밍 처리	전체 파일을 메모리에 로드하지 않고 청크 단위로 처리
조건부 조기 종료	시간대 필터(<code>since</code>) 조건 충족 시 나머지 파일 스캔을 즉시 중단
건수 제한	한 번의 요청당 최대 1,000건으로 제한하여 과도한 응답 방지

10.8 주의사항

⚠ 주의

로그 레벨을 `DEBUG` 로 변경하면 로그 양이 대폭 증가하여 디스크 사용량이 빠르게 증가할 수 있습니다. 디버깅 완료 후 반드시 `INFO` 레벨로 복원하세요.

💡 팁

특정 시간대의 문제를 조사할 때는 레벨 필터(`ERROR`)와 검색어 필터를 조합하면 효율적입니다. 예를 들어, `ERROR` 레벨 + `timeout` 검색어로 타임아웃 관련 오류만 빠르게 추출할 수 있습니다.

10.9 관련 API

API	메서드	설명
<code>/tkadmin/api/logs</code>	GET	로그 조회 (필터링, 페이징 지원)

쿼리 파라미터

파라미터	기본값	설명
<code>limit</code>	100	조회할 최대 로그 건수 (최대 1000)
<code>level</code>	(전체)	로그 레벨 필터 (INFO, WARN, ERROR, DEBUG)
<code>search</code>	(없음)	검색어 (대소문자 구분 없음)
<code>since</code>	(없음)	이 시각 이후 로그만 조회
<code>until</code>	(없음)	이 시각 이전 로그만 조회

10.10 다음 단계

- [관리자 보고](#)에서 Watchdog 장애/복구 알림 이력을 확인하세요.
- [전문가 편집기](#)에서 설정 변경 후 발생하는 로그를 모니터링하세요.

11. 서비스 로그 뷰어

v0.2.3 — 멀티패널 서비스 실시간 파일 로그 모니터링 시스템

11.1 개요

서비스 로그 뷰어는 TACHYON 서비스별 파일 로그를 **멀티패널 UI**로 동시에 실시간 모니터링하는 기능입니다. 최대 **8개** 서비스를 동시에 열어 각각의 로그를 실시간으로 추적할 수 있습니다.

i 시스템 로그가 tkadmin 자체 로그(journalctl 기반)를 표시하는 반면, **서비스 로그**는 TACHYON 각 서비스의 파일 로그를 직접 읽어 표시합니다.

11.2 화면 구성

도구 모음 (Toolbar)

화면 상단의 도구 모음에서 다음 작업을 수행할 수 있습니다:

항목	설명
서비스 추가	체크박스 팝오버에서 모니터링할 서비스를 다중 선택합니다
레이아웃 전환	그리드 / 수직 분할 / 수평 분할 모드를 전환합니다
전체 삭제	열린 모든 패널을 한 번에 닫습니다

서비스 추가 (체크박스 멀티선택)

"서비스 추가" 버튼을 클릭하면 카테고리별 체크박스 팝오버가 나타납니다:

- **TACHYON 서비스:** Api, Auth, Manager, Report, Stat, Batch, Watchdog 등 동적 발견된 서비스
- **미들웨어:** Redis, Kafka, Zookeeper, OpenSearch, Logstash, Nginx

사용 방법:

1. "서비스 추가" 버튼 클릭 → 팝오버 열기
2. 원하는 서비스 체크박스 선택 (카테고리 전체 선택 가능)
3. "패널 추가" 버튼 클릭 → 선택한 서비스들의 패널이 일괄 생성

💡 이미 열려있는 서비스는 "열림" 배지가 표시되며 선택할 수 없습니다.

로그 패널

각 패널은 독립적으로 동작하며 다음 요소로 구성됩니다:

- **헤더 (Header):** 드래그 핸들, 서비스 이름, 상태 표시(스트리밍/일시정지), 액션 버튼들
- **로그 영역 (Body):** 로그 내용 표시 (레벨별 색상 구분)
- **상태 바 (Footer):** 로그 줄 수, 버퍼 사용률, 스트리밍 상태, 파일 크기

11.3 주요 기능

1. 실시간 로그 스트리밍

각 패널은 **2초 간격**으로 새로운 로그를 자동으로 가져옵니다.

- 새 로그가 추가되면 자동으로 맨 아래로 스크롤됩니다
- 각 패널은 **최대 500줄**까지 캐시합니다 (초과 시 오래된 줄 제거)

2. 로그 레벨 표시

로그 레벨에 따라 자동으로 색상이 적용됩니다:

레벨	색상	설명
ERROR	빨간색	오류 메시지
WARN	노란색	경고 메시지
INFO	초록색	일반 정보
DEBUG	회색	디버깅 정보
FATAL	밝은 빨간색	치명적 오류

3. 로그 레벨 필터

패널 헤더의 **E / W / I / D** 토글 버튼으로 특정 레벨만 표시할 수 있습니다.

- 활성화된 레벨만 필터링되어 표시됩니다
- 필터 적용 시 상태 바에 **150/300줄** 형태로 표시(보이는 줄 / 전체 줄)
- 레벨이 감지되지 않는 줄(일반 텍스트)은 항상 표시됩니다

4. 레이아웃 모드 전환

도구 모음의 레이아웃 스위처로 3가지 배치 모드를 선택할 수 있습니다:

아이콘	레이아웃	설명
그리드	Grid	기본값. 패널 수에 따라 자동 격자 배치
수직	Vertical	세로 분할 (각 패널이 전체 높이, 좌우 배치)
수평	Horizontal	가로 분할 (각 패널이 전체 너비, 상하 배치)

자동 그리드 레이아웃 (Grid 모드):

패널 수	레이아웃
1개	1열 (전체 폭)
2개	2열
3~4개	2열 × 2행
5~8개	3열

5. 패널 드래그앤드롭 순서 변경

패널 헤더 좌측의 ****드래그 핸들 아이콘 (:)****을 잡고 드래그하여 패널 순서를 변경할 수 있습니다.

- 드래그 시 드롭 위치에 파란색 인디케이터 라인이 표시됩니다
- 드롭 시 부드러운 착지 애니메이션이 적용됩니다
- 최대화 상태에서는 드래그가 차단됩니다

6. 패널 최대화

패널 헤더의 **최대화 아이콘**을 클릭하면:

- 해당 패널이 **전체 화면**으로 확장됩니다
- 다른 패널과 도구 모음이 숨겨집니다
- **ESC** 키 또는 최소화 아이콘으로 원래 크기로 복원됩니다

7. 로그 다운로드

패널 헤더의 **다운로드 아이콘**을 클릭하면:

- 현재 패널에 캐시된 로그가 텍스트 파일로 저장됩니다
- 파일명 형식: **{서비스명}_{날짜시간}.log**

8. 검색 기능

패널 헤더의 **돋보기 아이콘**을 클릭하면:

- 검색 입력창이 나타납니다
- 입력한 키워드를 포함하는 줄만 필터링되어 표시됩니다
- 대소문자를 구분하지 않습니다

9. 일시 정지 / 재개

패널 헤더의 **일시정지 아이콘**을 클릭하면:

- 해당 패널의 실시간 스트리밍이 일시 정지됩니다
- 상태가 "일시정지"로 변경됩니다
- 다시 클릭하면 스트리밍이 재개됩니다

10. 메모리 관리

각 패널은 **최대 500줄**의 로그를 버퍼에 유지합니다.

- 상태 바에 버퍼 사용률 인디케이터가 표시됩니다
- 버퍼가 **80%** 이상 차면 경고 색상으로 변경됩니다
- 버퍼가 가득 차면 오래된 줄부터 자동으로 제거됩니다

11. 팝아웃 (Pop-out)

패널 헤더의 **팝아웃 아이콘**을 클릭하면 해당 서비스의 로그가 **독립 브라우저 창**으로 분리됩니다.

- 멀티 모니터 환경에서 별도 화면에 특정 로그를 상시 배치할 수 있습니다.
- 팝아웃된 창도 원본 패널과 동일한 실시간 업데이트가 전송됩니다.

12. 세션 상태 유지 (State Persistence)

서비스 로그 뷰어의 모든 상태는 **sessionStorage**를 기반으로 브라우저 세션 동안 안전하게 유지됩니다.

- **유지 대상:** 열려있는 패널 목록, 패널별 순서, 선택된 레이아웃, 검색 필터링 상태.

- 브라우저를 새로고침하거나 다른 메뉴로 이동 후 복귀해도 이전의 모니터링 환경이 즉시 복원됩니다.

11.4 대시보드에서 바로 가기


대시보드 → 서비스 로그 섹션의 각 서비스 카드에 있는 **터미널 아이콘**을 클릭하면:

1. 자동으로 서비스 로그 화면으로 전환됩니다
2. 해당 서비스의 패널이 자동으로 생성되어 로그 스트리밍을 시작합니다

11.5 지원 서비스

서비스 로그 뷰어는 다음 카테고리의 서비스 로그를 지원합니다:

카테고리	서비스 예시
TACHYON 서비스	Api, Auth, Manager, Report, Stat, Batch, Watchdog 등
미들웨어	Redis, Kafka, Zookeeper, OpenSearch, Logstash, Nginx

 실제 서버에 로그 파일이 존재하는 서비스만 목록에 표시됩니다.

 tkadmin 자체 로그는 **시스템 로그** 메뉴에서 확인하세요.

11.6 비활성 탭 최적화

브라우저 탭이 숨겨지면 자동으로 모든 패널의 폴링이 중지됩니다:

- 탭 복귀 시 자동으로 스트리밍이 재개됩니다
- 오랜 시간 비활성 후 복귀 시 자동 일시정지 여부를 묻는 안내가 표시될 수 있습니다

11.7 팁

- 자주 확인하는 서비스는 패널을 열어둔 채로 최대화하면 편리합니다
- 예러만 모니터링하려면 **E** 버튼만 활성화하세요
- 체크박스 팝오버에서 카테고리 전체 선택으로 관련 서비스를 한 번에 추가하세요
- 드래그 핸들로 중요한 서비스 패널을 상단에 배치하세요

- 멀티 모니터 환경에서는 팝아웃으로 서비스별 로그를 별도 화면에 배치하세요

12. 관리자 보고

12.1 개요

관리자 보고 화면은 Watchdog 엔진의 자동 장애 감지 이벤트와 `tkcli` 외부 CLI 도구를 통한 수동 작업 보고를 단일 테이블에 통합 표시하는 기능입니다. 운영자는 이 화면을 통해 서비스의 장애 발생, 자동 복구, 수동 제어 등 모든 운영 이벤트를 시간순으로 파악하고, 읽음/미읽음 상태를 관리할 수 있습니다.

12.2 화면 구성

영역	설명
상단 툴바	유형 필터, 검색어 입력, 검색/모두 읽음 버튼, 데이터 소스 선택
보고 테이블	시간, 유형, 서비스, 메시지, 읽음 상태 컬럼

테이블 컬럼

컬럼	설명	기본 너비
시간	이벤트 발생 시각 (한국어 날짜/시간 형식)	220px
유형	이벤트 유형 배지	120px
서비스	대상 서비스명	150px
메시지	이벤트 상세 내용	400px
상태	읽음 여부 표시	80px

12.3 보고 유형

관리자 보고에 기록되는 이벤트 유형은 다음과 같습니다:

유형	배지 색상	발생 조건	예시 메시지
FAILURE	빨간색 (ERROR)	Watchdog이 서비스 중단/실패 감지	장애 감지: 서비스 'redis'가 중단된 상태입니다.
RECOVERY_SUCCESS	파란색 (INFO)	Watchdog 자동 재시작 성공 또는 tkcli 작업 성공	복구 성공: 'redis' 서비스가 정상화되었습니다.
RECOVERY_FAILED	노란색 (WARN)	Watchdog 자동 재시작 실패 또는 tkcli 작업 실패	복구 실패: 'kafka' 재시작 후에도 비정상 상태입니다.
INFO	파란색 (INFO)	tkcli 외부 보고 (수동 작업 결과)	[tkcli] 서비스 'nginx'에 대해 'restart' 명령이 수동 수행되었습니다.
CONFIG_UPDATE	파란색 (INFO)	전문가 편집기를 통한 설정 파일 변경	사용자 [GUID]가 설정 파일 'application.yml'을(를) 수정했습니다.

12.4 필터링 기능

유형별 필터

상단 툴바의 드롭다운에서 원하는 이벤트 유형을 선택하면, 해당 유형의 보고만 필터링합니다.

옵션	설명
모든 유형	필터 없이 전체 보고 표시 (기본값)
장애 감지	FAILURE 유형만 표시
복구 성공	RECOVERY_SUCCESS 유형만 표시
복구 실패	RECOVERY_FAILED 유형만 표시

서비스명/메시지 검색

1. 상단 툴바의 검색 입력 필드에 검색어를 입력합니다.
2. 검색 버튼을 클릭하거나 Enter 키를 누릅니다.
3. 서비스명 또는 메시지 내용에서 검색어가 포함된 항목만 표시합니다.
4. 검색은 대소문자를 구분하지 않습니다.

데이터 소스 선택

상단 톨바 우측에서 데이터 소스를 전환할 수 있습니다:

소스	API	설명
전체 감사 로그 (History DB)	<code>/tkadmin/api/audit-logs</code>	SQLite에 영구 보관된 전체 이력 (기본값)
실시간 알림 (Latest 100)	<code>/tkadmin/api/alerts</code>	메모리 캐시의 최근 100건 알림

12.5 읽음 관리

미확인 항목 시각적 강조

읽지 않은(미확인) 보고 항목은 다음과 같이 시각적으로 구분됩니다:

요소	미읽음	읽음
행 배경색	연한 빨간색 배경 (<code>rgba(239, 68, 68, 0.03)</code>)	투명
상태 아이콘	빨간색 ● (볼드)	회색 체크마크

개별 읽음 처리

1. 보고 테이블에서 특정 항목의 **행을** 클릭합니다.
2. 보고 상세 모달이 표시됩니다.
 - 점검 시점
 - 대상 서비스
 - 이벤트 유형
 - 상세 내용
 - 조치 권고 사항
3. 모달이 표시되면 해당 항목은 자동으로 **읽음** 처리됩니다.
4. 닫기 버튼으로 모달을 닫습니다.

일괄 읽음 처리

1. 상단 톨바의 **모두 읽음** 버튼을 클릭합니다.
2. 모든 미확인 보고 항목이 즉시 읽음 상태로 전환됩니다.
3. 헤더의 알림 배지와 사이드바 배지도 함께 업데이트됩니다.

12.6 실시간 갱신

관리자 보고 화면에 진입하면 **10초 주기**로 보고 데이터를 자동 폴링합니다:

항목	값
폴링 주기	10초
동작	선택된 데이터 소스에서 최신 데이터를 조회하여 테이블 갱신
종료 조건	다른 메뉴로 이동하면 타이머 자동 해제

12.7 알림 배너 정책

상단 빨간 배너

- **Active FAILURE만 표시**: 현재 장애가 발생했으나 아직 복구되지 않은(동일 서비스에 대한 **RECOVERY_SUCCESS** 가 없는) 최신 **FAILURE** 알림만 빨간색 상단 배너로 표시합니다.
- **복구 시 자동 숨김**: 장애가 발생한 서비스에 대해 **RECOVERY_SUCCESS** 알림이 수신되면, 해당 장애 배너는 즉시 숨김 처리됩니다.
- **클릭 동작**: 배너를 클릭하면 해당 장애의 상세 보고 모달이 열립니다.

알림 뱃지

- **헤더 영역**: 종 아이콘 옆에 미읽음 건수가 뱃지로 표시됩니다 (최대 **99+**).
- **사이드바**: 관리자 보고 메뉴 항목에 미읽음 건수가 뱃지로 표시됩니다 (최대 **9+**).
- **갱신 주기**: 10초 주기로 정확한 미읽음 건수를 DB에서 조회하여 업데이트합니다.

감사 추적성 준수

⚠ 주의

배너가 사라지더라도(복구 성공 시), 관리자가 직접 읽음 처리

하기 전까지는 알림 뱃지 및 보고 목록의 미읽음 상태가 절대 자동으로 해제되지 않습니다

. 이는 사후 장애 이력 확인을 보장하기 위한

감사 추적성(Audit Integrity)

원칙에 따른 설계입니다.

즉, 복구 성공으로 배너는 사라지지만:

- 헤더 알림 배지의 미읽음 건수는 유지됩니다.
- 관리자 보고 테이블에서 해당 항목은 미읽음 상태로 남아 있습니다.
- 관리자가 직접 항목을 열어 확인하거나, **모두 읽음** 버튼을 클릭해야만 읽음 처리됩니다.

12.8 알림 데이터 영속성

SQLite 기반 영구 보관

모든 장애 및 복구 알림은 SQLite 데이터베이스(`audit_logs` 테이블)에 기록됩니다.

필드	설명
<code>id</code>	고유 식별자 (자동 증가)
<code>time</code>	이벤트 발생 시각
<code>service</code>	대상 서비스명
<code>type</code>	이벤트 유형 (FAILURE, RECOVERY_SUCCESS 등)
<code>message</code>	상세 메시지
<code>is_healing</code>	Watchdog 자동 복구 중 여부
<code>read</code>	읽음 여부 (0/1)

재시작 시 데이터 복원

tkadmin이 재시작되면 SQLite에서 최근 알림 내역(최대 100건)을 메모리로 로드하여 대시보드 상태를 복원합니다. 따라서 서비스 재시작 후에도 이전의 장애/복구 이력이 보존됩니다.

12.9 주의사항

**주의****모두 읽음**

버튼을 클릭하면 모든 미확인 보고가 일괄 읽음 처리됩니다. 중요한 장애 보고를 놓치지 않도록, 가급적 개별 항목을 확인한 후 읽음 처리하는 것을 권장합니다.

**팁**

반복적인 **RECOVERY_FAILED** 알림이 발생하면, 해당 서비스의 로그를 [시스템 로그](#) 화면에서 확인하고, 근본 원인(MariaDB 연결 실패, 포트 충돌 등)을 해결해야 합니다.

**팁**

tkcli 를 통한 원격 작업 이력도 이 화면에 통합 표시되므로, 여러 운영자가 수행한 작업의 이력을 한 곳에서 추적할 수 있습니다.

12.10 관련 API

API	메서드	설명
<code>/tkadmin/api/alerts</code>	GET	메모리 캐시의 최근 알림 조회 (최대 100건)
<code>/tkadmin/api/audit-logs</code>	GET	SQLite 감사 로그 전체 조회
<code>/tkadmin/api/alerts/:id</code>	GET	특정 알림 상세 조회
<code>/tkadmin/api/alerts/read</code>	POST	모든 알림 일괄 읽음 처리
<code>/tkadmin/api/alerts/unread</code>	GET	미읽음 건수 조회
<code>/tkadmin/api/monitor/report</code>	POST	tkcli 외부 보고 수신

tkcli 보고 API 페이로드

```
{
  "service": "서비스명",
  "action": "start|stop|restart",
  "status": "pending|success|fail",
```



```
"source": "tkcli"
}
```

필드	설명
<code>service</code>	제어 대상 서비스명
<code>action</code>	수행한 동작
<code>status</code>	<code>pending</code> (사전 등록), <code>success</code> (성공), <code>fail</code> (실패)
<code>source</code>	보고 출처 (예: <code>tkcli</code>)

12.11 다음 단계

- [서비스 관리](#)에서 장애가 발생한 서비스를 직접 제어하세요.
- [시스템 로그](#)에서 서비스 오류의 근본 원인을 추적하세요.
- [전문가 편집기](#)에서 설정 변경 이력을 확인하세요.

13. Recovery Mode (긴급 복구 모드)

tkadmin은 TACHYON 인증 서버 장애 시에도 핵심 관리 기능에 접근할 수 있도록, 메인 서버와 독립된 포트로 동작하는 긴급 관리용 Recovery 서버를 제공합니다.

13.1 개요

항목	설명
독립 포트	<code>recovery_port</code> (기본값: 메인 <code>port</code> + 1, 즉 <code>13701</code>)
목적	TACHYON 인증 서버(Auth) 장애 시 핵심 설정 관리 기능 접근 보장
플랫폼 제한	Linux 전용 (<code>runtime.GOOS == "linux"</code>)
인증 방식	OS 계정 인증 (PAM)
세션 관리	<code>tk_recovery</code> 쿠키 (1시간 유효)

Recovery 서버는 메인 서버 시작 시 별도의 고루틴(Goroutine)에서 독립적으로 구동됩니다.

`recovery_port` 가 0보다 큰 값으로 설정된 경우에만 시작됩니다.

13.2 접근 방법

1단계: Recovery 로그인 페이지 접속

브라우저에서 Recovery 포트로 직접 접근합니다:

```
http://서버주소:13701/recovery/
```

또는 서버 로컬에서 접근:

```
curl http://127.0.0.1:13701/recovery/
```

2단계: OS 계정으로 로그인

Recovery 로그인 페이지에서 Linux 시스템의 OS 계정으로 인증합니다.

- **사용자명:** Linux 시스템 계정 (예: `root`, 또는 로그인 가능한 일반 계정)
- **비밀번호:** 해당 OS 계정의 비밀번호

3단계: 관리 대시보드 진입

인증에 성공하면 tkadmin 대시보드에 **(RECOVERY MODE)** 표시와 함께 진입합니다.

13.3 인증 방식

OS 계정 인증 (PAM)

Recovery Mode는 TACHYON JWT 인증 대신 운영체제의 PAM(Pluggable Authentication Modules) 인증을 사용합니다.

인증 처리 과정:

1. 사용자가 `username` 과 `password` 를 입력합니다.
2. `/etc/passwd` 에서 해당 사용자의 존재 여부를 확인합니다.
3. 사용자의 로그인 셸이 유효한지 검증합니다.
 - 차단되는 셸: `/sbin/nologin`, `/usr/sbin/nologin`, `/bin/false`
4. `su` 명령을 통해 비밀번호를 검증합니다.
5. 인증 성공 시 `tk_recovery` 쿠키를 설정합니다.

로그인 가능 조건:

- `/etc/passwd` 에 등록된 계정이어야 합니다.
- 로그인 셸이 `nologin/false`가 아닌 유효한 셸이어야 합니다.
- 올바른 비밀번호를 입력해야 합니다.



팁

대부분의 환경에서 `root` 계정으로 로그인하는 것이 가장 확실합니다. tkadmin 프로세스 자체가 root 권한으로 실행되기 때문입니다.

세션 관리

- 쿠키 이름: `tk_recovery`
- 유효 시간: 3600초 (1시간)
- 쿠키 옵션: `HttpOnly` 활성화
- 세션 검증: 모든 API 요청 시 `tk_recovery` 쿠키의 `RECOVERY_SESSION_` 접두사를 검증합니다.

- 인증되지 않은 요청은 `/recovery/` 로그인 페이지로 리다이렉트됩니다.

13.4 IP 기반 접근 제어 (ACL)

Recovery 서버는 보안을 위해 IP 기반 접근 제어를 적용합니다.

허용 대상

IP	허용 여부	설명
<code>127.0.0.1</code>	자동 허용	IPv4 localhost
<code>::1</code>	자동 허용	IPv6 localhost
<code>localhost</code>	자동 허용	localhost 문자열
<code>emergency_ips</code> 목록	설정 시 허용	<code>tkadmin.yml</code> 에 등록된 IP

차단 시 동작

허용되지 않은 IP에서 접근하면 `error_403.html` 페이지가 표시됩니다:

- 표시 내용:** 클라이언트 IP, 차단 사유, `emergency_ips` 설정 안내
- 로그 기록:** `[RECOVERY-ACL] Denied access` 경고 로그가 기록됩니다.

emergency_ips 설정 방법

`tkadmin.yml` 파일에서 허용할 IP를 설정합니다:

```
emergency_ips:
  - "10.10.1.100"      # 관리자 PC
  - "192.168.1.50"    # 예비 관리자 PC
  - "172.16.0.10"     # 원격 관리 서버
```



주의

`emergency_ips`에 등록된 IP는 Recovery 포트뿐만 아니라 메인 서버의 인증 우회(Emergency Bypass)에도 사용됩니다. 신뢰할 수 있는 관리자 IP만 등록하세요.

13.5 제공 API (제한적)

Recovery Mode에서는 핵심 관리 기능에 해당하는 제한된 API만 제공됩니다.

메서드	경로	설명
GET	/tkadmin/api/status	서비스 상태 조회 (서비스 목록, 시스템 리소스, 라이선스 정보)
GET	/tkadmin/api/config	현재 설정 조회
POST	/tkadmin/api/config	설정 변경 (emergency_ips, port 등)
POST	/tkadmin/api/config/restart	서비스 재시작
GET	/tkadmin/api/alerts	알림 목록 조회
GET	/tkadmin/api/audit-logs	감사 로그 조회
GET	/tkadmin/api/alerts/:id	알림 상세 조회

메인 서버에서 제공되는 다음 API는 Recovery Mode에서 **제공되지 않습니다**:

- 시스템 통계 (/system/stats)
- 서비스 리소스 (/services)
- 서비스 제어 (/service/:action)
- 외부 파일 편집 (/config/external/*)
- 로그 조회 (/logs)
- OS 환경 체크 (/system/os-checks)
- 모니터링 보고 (/monitor/report)

13.6 사용 시나리오

시나리오 1: TACHYON Auth 서비스 장애

1. TACHYON Auth 서비스가 중단되어 웹 UI 로그인이 불가능합니다.
2. 서버에 SSH로 접속합니다.
3. `http://127.0.0.1:13701/recovery/` 에 접근합니다.
4. `root` 계정으로 로그인합니다.
5. 대시보드에서 서비스 상태를 확인합니다.
6. 필요 시 설정을 변경하고 재시작합니다.

시나리오 2: 원격 긴급 접근

1. 관리자 PC의 IP가 **emergency_ips** 에 등록되어 있습니다.
2. 브라우저에서 **http://서버주소:13701/recovery/** 에 접근합니다.
3. OS 계정으로 로그인하여 긴급 관리를 수행합니다.

시나리오 3: emergency_ips 미등록 시

1. 허용되지 않은 IP에서 Recovery 포트 접근 시 403 에러가 발생합니다.
2. 서버에 SSH로 접속하여 localhost에서 Recovery에 진입합니다.
3. 설정 변경 API를 통해 본인의 IP를 **emergency_ips** 에 추가합니다.
4. 이후 원격에서도 Recovery 포트에 접근할 수 있습니다.

```
# SSH 접속 후 localhost에서 emergency_ips 추가
curl -b "tk_recovery=RECOVERY_SESSION_root" \
  -X POST http://127.0.0.1:13701/tkadmin/api/config \
  -H "Content-Type: application/json" \
  -d '{"emergency_ips": ["10.10.1.100"], "port": 13700, "target_dir": "/usr/local/TACHYON/TTS40/"}'
```

13.7 보안 고려사항

항목	설명
IP 제한	localhost 및 emergency_ips 만 접근 가능하여 외부 공격 노출을 최소화
OS 인증	TACHYON 토큰이 아닌 실제 운영체제 계정으로 인증하므로 별도 취약점에 의존하지 않음
제한된 API	전체 API의 일부만 노출하여 공격 표면을 최소화
세션 만료	1시간으로 세션 유효 시간을 제한하여 장기 세션 탈취 위험 감소
HttpOnly 쿠키	JavaScript를 통한 쿠키 접근을 차단하여 XSS 공격 방어

⚠ 주의

Recovery Mode는 긴급 상황에서의 관리 접근을 보장하기 위한 기능입니다. 일상적인 관리 작업에는 메인 서버를 통한 정상적인 TACHYON 인증을 사용하세요.

14. API 레퍼런스

tkadmin이 제공하는 REST API의 전체 목록과 사용 방법을 설명합니다.

14.1 공통 사항

항목	설명
Base Path	/tkadmin/api/
인증 (메인)	tk_session 쿠키 (TACHYON Auth Bridge)
인증 (Recovery)	tk_recovery 쿠키 (OS 계정 인증)
인증 우회	localhost 및 emergency_ips 목록의 IP는 인증 없이 접근 가능
응답 형식	JSON (Content-Type: application/json)
에러 응답	{"error": "에러 메시지"} 형식

인증 흐름

1. 쿠키 확인: tk_session 쿠키가 없으면 /tkadmin/bridge 로 리다이렉트
2. JWT 검증: 쿠키의 JWT 토큰 유효성 검증
3. Redis 세션 확인: Redis에서 TOKEN:GUID 키로 실제 세션 존재 여부 확인
4. 세션 만료 시: 쿠키 삭제 후 /user/login?reason=expired 로 리다이렉트

GET /tkadmin/api/stream (SSE)

서비스 상태, 시스템 리소스, 라이선스 정보를 실시간으로 Push합니다. SSE(Server-Sent Events) 프로토콜을 사용합니다.

요청:

```
curl -b "tk_session=<TOKEN>" -N https://서버주소/tkadmin/api/stream
```

응답 형식: `text/event-stream`

이벤트 데이터: `/tkadmin/api/status` 와 동일한 JSON 구조가 `data:` 필드로 전송됩니다.

동작 상세:

항목	설명
연결 시	즉시 현재 상태 데이터를 1회 전송
변경 감지	5초 주기로 서버 상태를 체크하여 변경 시에만 Push
Heartbeat	30초마다 SSE 주석(<code>: heartbeat</code>)을 전송하여 연결 유지
HTTP 헤더	<code>Content-Type: text/event-stream</code> , <code>Cache-Control: no-cache</code> , <code>X-Accel-Buffering: no</code>

이벤트 예시:

```
data: {"services":[{"name":"TACHYON-Api1","status":"active","sub_state":"running","pid":"12345","cpu":"2.3","memory":"156.7"}],"system":{"cpu_usage":15.2,"version":"0.2.0","last_checked":"2026-02-18 17:50:00"}}

: heartbeat

data: {"services":[...],"system":{"cpu_usage":16.8},...}
```



팁

SSE 연결은 브라우저의 `EventSource` API를 사용하여 자동으로 재연결됩니다. 네트워크 문제 시 3회 실패 후 5초 폴링으로 자동 전환됩니다.

GET /tkadmin/api/status

서비스 상태, 설정 요약, 라이선스 정보, 시스템 리소스를 통합 조회합니다.

요청:


```
curl -b "tk_session=<TOKEN>" https://서버주소/tkadmin/api/status
```

응답 필드:

필드	타입	설명
<code>services</code>	Array	TACHYON 서비스 상태 목록 (이름, 상태, PID, CPU, 메모리)
<code>config_summary</code>	String	솔루션 이름
<code>license</code>	Object	라이선스 정보 (company, code, agents)
<code>system</code>	Object	시스템 리소스 (CPU, 메모리, 디스크)
<code>version</code>	String	tkadmin 버전
<code>last_checked</code>	String	마지막 체크 시간 (YYYY-MM-DD HH:MM:SS)

응답 예시:

```
{
  "services": [
    {
      "name": "TACHYON-Api1",
      "status": "active",
      "sub_state": "running",
      "pid": "12345",
      "cpu": "2.3",
      "memory": "156.7"
    }
  ],
  "config_summary": "TACHYON Admin",
  "license": {
    "company": "주식회사 예시",
    "code": "COMP001",
    "agents": "5000"
  },
  "system": {
    "cpu_usage": 15.2,
    "memory_total": 16384,
    "memory_used": 8192,
    "disk_total": 500000,
    "disk_used": 250000
  }
}
```

```
},
"version": "0.1.5.142",
"last_checked": "2025-12-20 14:30:00"
}
```

GET /tkadmin/api/system/stats

시스템 리소스 통계(CPU, 메모리, 디스크)를 조회합니다.

요청:

```
curl -b "tk_session=<TOKEN>" https://서버주소/tkadmin/api/system/stats
```

응답: 시스템 리소스 객체 (CPU 사용률, 메모리 총량/사용량, 디스크 총량/사용량, 업타임 등)

GET /tkadmin/api/services

TACHYON 서비스 목록 및 리소스 사용 현황을 조회합니다.

요청:

```
curl -b "tk_session=<TOKEN>" https://서버주소/tkadmin/api/services
```

응답: 서비스 상태 배열 (각 서비스의 이름, 활성 상태, PID, CPU/메모리 사용량)

14.2 설정 관리

GET /tkadmin/api/config

현재 tkadmin 설정을 조회합니다.

요청:

```
curl -b "tk_session=<TOKEN>" https://서버주소/tkadmin/api/config
```

응답 필드:

필드	타입	설명
<code>target_dir</code>	String	TACHYON 설치 경로
<code>listen_addr</code>	String	서버 바인딩 주소
<code>port</code>	Integer	메인 서버 포트
<code>solution_name</code>	String	솔루션 표시 이름
<code>emergency_ips</code>	Array	Recovery 접근 허용 IP 목록
<code>log_path</code>	String	로그 파일 경로
<code>debug</code>	Boolean	디버그 모드 여부
<code>show_link</code>	Boolean	TACHYON 대시보드 메뉴 표시 여부
<code>allowed_ids</code>	Array	메뉴 표시 허용 사용자 ID 목록
<code>client_ip</code>	String	요청자의 현재 IP (편의 제공)

응답 예시:

```
{
  "target_dir": "/usr/local/TACHYON/TTS40/",
  "listen_addr": "0.0.0.0",
  "port": 13700,
  "solution_name": "TACHYON Admin",
  "emergency_ips": ["10.10.1.100"],
  "log_path": "logs/tkadmin.log",
  "debug": false,
  "show_link": true,
  "allowed_ids": ["tsadmin"],
  "client_ip": "10.10.1.100"
}
```

POST /tkadmin/api/config

tkadmin 설정을 변경합니다. 설정은 `tkadmin.yml` 파일에 저장됩니다.

요청:

```
curl -b "tk_session=<TOKEN>" \
-X POST https://서버주소/tkadmin/api/config \
-H "Content-Type: application/json" \
-d '{
  "target_dir": "/usr/local/TACHYON/TTS40/",
  "port": 13700,
  "solution_name": "TACHYON Admin",
  "emergency_ips": ["10.10.1.100"],
  "show_link": true,
  "allowed_ids": ["tsadmin", "operator01"]
}'
```

응답 필드:

필드	타입	설명
message	String	처리 결과 메시지
will_restart	Boolean	포트 변경으로 인해 재시작이 필요한지 여부

응답 예시 (포트 변경 시):

```
{
  "message": "설정이 저장되었습니다. 포트 변경을 적용하려면 서비스를 재시작해주세요.",
  "will_restart": true
}
```

응답 예시 (포트 미변경 시):

```
{
  "message": "설정이 저장되었습니다.",
  "will_restart": false
}
```

**팁**

설정 저장 시 NGINX `proxy_pass` 포트가 자동으로 동기화됩니다.

POST /tkadmin/api/config/restart

tkadmin 서비스를 재시작합니다. 내부적으로 `systemctl restart tkadmin` 명령을 실행합니다.

요청:

```
curl -b "tk_session=<TOKEN>" \
-X POST https://서버주소/tkadmin/api/config/restart
```

응답:

```
{
  "message": "서비스가 재시작됩니다."
}
```



주의

재시작 요청 후 1초의 지연을 두고 서비스가 재시작됩니다. 재시작 완료까지 일시적으로 서비스에 접근할 수 없습니다.

GET /tkadmin/api/nav-config

TACHYON 대시보드 인젝터용 네비게이션 설정을 조회합니다. 이 API는 **인증 없이** 접근 가능합니다.

요청:

```
curl https://서버주소/tkadmin/api/nav-config
```

응답 필드:

필드	타입	설명
<code>show_link</code>	Boolean	메뉴 표시 여부
<code>allowed_ids</code>	Array	메뉴 표시 허용 사용자 ID 목록

응답 예시:

```
{
  "show_link": true,
  "allowed_ids": ["tsadmin"]
}
```

14.3 외부 파일 편집

GET /tkadmin/api/config/external/files

TACHYON 설치 경로 하위의 편집 가능한 설정 파일 목록을 조회합니다.

요청:

```
curl -b "tk_session=<TOKEN>" https://서버주소/tkadmin/api/config/external/files
```

응답: 설정 파일 배열 (각 파일의 상대 경로, 카테고리(global/service/system), 최종 수정 시간)

탐색 대상 파일 패턴: `.yaml`, `.yaml_dev`, `.properties`, `.properties_dev`, `.json`, `.conf`
(Nginx)

제외 디렉토리: `.git`, `node_modules`, `logs`, `temp`, `backup`, `dist`

GET /tkadmin/api/config/external/file

특정 설정 파일의 내용을 조회합니다.

요청 파라미터:

파라미터	위치	필수	설명
<code>path</code>	Query	예	파일의 상대 경로 (target_dir 기준)

요청:

```
curl -b "tk_session=<TOKEN>" \
  "https://서버주소/tkadmin/api/config/external/file?
  path=conf/app_info.properties_dev"
```

응답 필드:

필드	타입	설명
path	String	파일 상대 경로
content	String	파일 내용

보안 제한:

- `..` 이 포함된 경로는 차단됩니다 (Path Traversal 방지).
- `/` 또는 `\` 로 시작하는 절대 경로는 차단됩니다.

POST /tkadmin/api/config/external/file

설정 파일의 내용을 저장합니다.

요청 본문:

필드	타입	필수	설명
path	String	예	파일의 상대 경로
content	String	예	저장할 파일 내용

요청:

```
curl -b "tk_session=<TOKEN>" \
  -X POST https://서버주소/tkadmin/api/config/external/file \
  -H "Content-Type: application/json" \
  -d '{
    "path": "conf/app_info.properties_dev",
    "content": "# 수정된 설정 내용\nkey=value"
  }'
```

응답:

```
{
  "message": "설정이 성공적으로 저장되었습니다."
}
```

동작 상세:

1. 보안 검사: Path Traversal 차단 (`..` , `/` , `\` 시작 경로)
2. 구문 검증: `.yaml` / `.yaml_dev` 파일은 YAML 문법 유효성을 검사한 후 저장
3. 자동 백업: 기존 파일을 `*.YYYYMMDDHHMMSS.bak` 형식으로 백업
4. 감사 로그: 변경 이력이 감사 로그에 기록됨

에러 응답 예시 (YAML 문법 오류):

```
{
  "error": "YAML 문법 오류: yaml: line 5: did not find expected key"
}
```

14.4 서비스 제어

POST /tkadmin/api/service/:action

TACHYON 서비스의 시작/중지/재시작을 제어합니다.

경로 파라미터:

파라미터	설명
<code>:action</code>	수행할 작업: <code>start</code> , <code>stop</code> , <code>restart</code>

요청 파라미터:

파라미터	위치	필수	설명
<code>service</code>	Query	아니오	서비스 이름 (기본값: <code>tachyon-shield</code>)

요청:


```
# TACHYON-Api1 서비스 재시작
curl -b "tk_session=<TOKEN>" \
  -X POST "https://서버주소/tkadmin/api/service/restart?service=TACHYON-Api1"

# 기본 서비스 중지
curl -b "tk_session=<TOKEN>" \
  -X POST https://서버주소/tkadmin/api/service/stop
```

응답:

```
{
  "message": "Service restarted successfully"
}
```

내부적으로 `systemctl <action> <service>` 명령을 실행하며, Watchdog이 의도적인 동작으로 인식하도록 기록합니다.

POST /tkadmin/api/monitor/report

외부 CLI 도구(tkcli 등)에서 수행한 서비스 제어 결과를 보고합니다.

요청 본문:

필드	타입	필수	설명
<code>service</code>	String	예	서비스 이름
<code>action</code>	String	예	수행된 작업 (<code>start</code> , <code>stop</code> , <code>restart</code>)
<code>status</code>	String	예	결과 (<code>success</code> , <code>fail</code> , <code>pending</code>)
<code>source</code>	String	예	보고 주체 (예: <code>tkcli</code>)

요청:

```
curl -b "tk_session=<TOKEN>" \
  -X POST https://서버주소/tkadmin/api/monitor/report \
  -H "Content-Type: application/json" \
  -d '{
```

```
"service": "TACHYON-Api1",
"action": "restart",
"status": "success",
"source": "tkcli"
}'
```

응답:

```
{
  "message": "External action reported successfully"
}
```

동작 상세:

- **status** 가 **pending** 인 경우: 의도적 동작만 기록하고 반환 (Watchdog 알림 억제용)
- **status** 가 **success** 인 경우: **RECOVERY_SUCCESS** 타입 알림으로 등록
- **status** 가 **fail** 인 경우: **RECOVERY_FAILED** 타입 알림으로 등록

14.5 알림 / 감사 로그

GET /tkadmin/api/alerts

Watchdog 감지 이벤트 및 외부 보고 알림 목록을 조회합니다.

요청:

```
curl -b "tk_session=<TOKEN>" https://서버주소/tkadmin/api/alerts
```

응답: 알림 배열 (각 알림의 ID, 시간, 서비스명, 유형, 메시지, 복구 여부, 읽음 상태)

알림 유형:

유형	설명
FAILURE	서비스 장애 감지
RECOVERY_SUCCESS	서비스 복구 성공
RECOVERY_FAILED	서비스 복구 실패
CONFIG_UPDATE	설정 파일 변경

GET /tkadmin/api/alerts/unread

미읽음 알림 수를 조회합니다.

요청:

```
curl -b "tk_session=<TOKEN>" https://서버주소/tkadmin/api/alerts/unread
```

응답:

```
{
  "unread_count": 3
}
```

POST /tkadmin/api/alerts/read

모든 미읽음 알림을 읽음으로 처리합니다.

요청:

```
curl -b "tk_session=<TOKEN>" \
-X POST https://서버주소/tkadmin/api/alerts/read
```

응답:

```
{
  "message": "All alerts marked as read"
}
```

}

GET /tkadmin/api/alerts/:id

특정 알림의 상세 정보를 조회합니다.

경로 파라미터:

파라미터	설명
:id	알림 ID (정수)

요청:

```
curl -b "tk_session=<TOKEN>" https://서버주소/tkadmin/api/alerts/42
```

응답:

```
{
  "id": 42,
  "time": "2025-12-20T14:30:00Z",
  "service": "TACHYON-Api1",
  "type": "FAILURE",
  "message": "서비스 TACHYON-Api1이 비정상 종료되었습니다.",
  "is_healing": true,
  "read": false
}
```

GET /tkadmin/api/audit-logs

감사 로그를 조회합니다. SQLite DB에 저장된 이력을 최신순으로 반환합니다.

요청 파라미터:

파라미터	위치	필수	기본값	설명
limit	Query	아니오	100	반환할 최대 레코드 수

요청:

```
curl -b "tk_session=<TOKEN>" "https://서버주소/tkadmin/api/audit-logs?limit=50"
```

응답: 감사 로그 배열 (각 로그의 ID, 시간, 서비스명, 유형, 메시지, 복구 시도 여부, 읽음 상태)

14.6 환경 체크

GET /tkadmin/api/system/os-checks

운영체제 환경 설정 상태를 점검합니다. SELinux, 방화벽, OS 리미트 등을 확인합니다.

요청:

```
curl -b "tk_session=<TOKEN>" https://서버주소/tkadmin/api/system/os-checks
```

응답: OS 환경 체크 결과 (SELinux 모드, 방화벽 상태, 허용된 포트/서비스, OS 리미트 설정)

POST /tkadmin/api/system/os-checks/selinux

SELinux 모드를 변경합니다.

요청 본문:

필드	타입	필수	설명
mode	String	예	SELinux 모드 (enforcing , permissive , disabled)

요청:

```
curl -b "tk_session=<TOKEN>" \
-X POST https://서버주소/tkadmin/api/system/os-checks/selinux \
-H "Content-Type: application/json" \
-d '{"mode": "permissive"}'
```

응답:

```
{
  "message": "SELinux 설정이 변경되었습니다."
}
```

POST /tkadmin/api/system/os-checks/firewall

방화벽 포트 또는 서비스를 추가/제거합니다 (firewalld 기반).

요청 본문:

필드	타입	필수	설명
port	String	예	포트 번호 또는 서비스 이름 (예: 13700/tcp , http)
action	String	예	수행할 작업 (add , remove)

요청:

```
curl -b "tk_session=<TOKEN>" \
  -X POST https://서버주소/tkadmin/api/system/os-checks/firewall \
  -H "Content-Type: application/json" \
  -d '{"port": "13700/tcp", "action": "add"}'
```

응답:

```
{
  "message": "방화벽 정책이 반영되었습니다."
}
```

POST /tkadmin/api/system/os-checks/limits

OS 리미트 설정(/etc/security/limits.conf)을 추가합니다.

요청 본문:

필드	타입	필수	설명
domain	String	예	도메인 (예: *, root)
type	String	예	타입 (soft , hard)
item	String	예	항목 (예: nofile , nproc)
value	String	예	설정값 (예: 65535)

요청:

```
curl -b "tk_session=<TOKEN>" \
-X POST https://서버주소/tkadmin/api/system/os-checks/limits \
-H "Content-Type: application/json" \
-d '{"domain": "*", "type": "hard", "item": "nofile", "value": "65535"}'
```

응답:

```
{
  "message": "OS Limit 설정이 추가되었습니다."
}
```

14.7 로그 조회

GET /tkadmin/api/logs

tkadmin 시스템 로그를 구조화된 JSON 형태로 조회합니다. 최신 로그가 먼저 반환되며, 로테이션된 .gz 압축 파일도 지원합니다.

요청 파라미터:

파라미터	위치	필수	기본값	설명
<code>limit</code>	Query	아니오	<code>100</code>	반환할 최대 로그 수 (최대 1000)
<code>search</code>	Query	아니오	-	대소문자 무시 키워드 검색
<code>level</code>	Query	아니오	-	로그 레벨 필터 (<code>DEBUG</code> , <code>INFO</code> , <code>WARN</code> , <code>ERROR</code>)
<code>since</code>	Query	아니오	-	시작 시간 필터 (이후 로그만 반환)
<code>until</code>	Query	아니오	-	종료 시간 필터 (이전 로그만 반환)

요청:

```
# 최근 50건 조회
curl -b "tk_session=<TOKEN>" "https://서버주소/tkadmin/api/logs?limit=50"

# ERROR 레벨만 필터링
curl -b "tk_session=<TOKEN>" "https://서버주소/tkadmin/api/logs?level=ERROR&limit=100"

# 키워드 검색
curl -b "tk_session=<TOKEN>" "https://서버주소/tkadmin/api/logs?search=redis&limit=50"

# 시간대 필터링
curl -b "tk_session=<TOKEN>" \
  "https://서버주소/tkadmin/api/logs?since=2025-12-20T10:00:00&until=2025-12-20T12:00:00"
```

응답 필드:

각 로그 항목의 필드:

필드	타입	설명
<code>level</code>	String	로그 레벨 (DEBUG, INFO, WARN, ERROR)
<code>ts</code>	String	타임스탬프
<code>caller</code>	String	호출 위치 (소스 파일:라인)
<code>msg</code>	String	로그 메시지
<code>fields</code>	Object	추가 구조화 데이터 (선택)

응답 예시:

```
[
  {
    "level": "INFO",
    "ts": "2025-12-20T14:30:00.123Z",
    "caller": "main.go:142",
    "msg": "[SERVER] Starting tkadmin",
    "fields": {
      "addr": ":13700",
      "version": "0.1.5.142"
    }
  },
  {
    "level": "WARN",
    "ts": "2025-12-20T14:29:58.456Z",
    "caller": "main.go:152",
    "msg": "[REDIS] Connection warning",
    "fields": {
      "error": "dial tcp 127.0.0.1:6379: connect: connection refused",
      "addr": "127.0.0.1:6379"
    }
  }
]
```

14.8 서비스 파일 로그 (v0.2.2)

GET /tkadmin/api/service/logs/file/list

TACHYON 서비스별 로그 파일 목록을 조회합니다. 실제 파일이 존재하는 서비스만 반환됩니다.

요청:

```
curl -b "tk_session=<TOKEN>" https://서버주소/tkadmin/api/service/logs/file/list
```

응답 필드:

각 서비스 항목:

필드	타입	설명
service	String	서비스 이름
category	String	카테고리 (TACHYON Core, Communication, 미들웨어, 보안, 관리도구)
log_path	String	로그 파일 절대 경로

응답 예시:

```
[
  {"service": "TACHYON-Package", "category": "TACHYON Core", "log_path":
"/usr/local/TACHYON/TTS40/dist/.../package.log"},
  {"service": "kafka", "category": "미들웨어", "log_path":
"/usr/local/TACHYON/TTS40/dist/.../server.log"},
  {"service": "tkadmin", "category": "관리도구", "log_path":
"/usr/local/tkadmin/logs/tkadmin.log"}
]
```

GET /tkadmin/api/service/logs/file

서비스 파일 로그의 내용을 offset 기반으로 읽습니다. 증분 읽기를 지원하여 실시간 폴링에 적합합니다.

요청 파라미터:

파라미터	위치	필수	기본값	설명
<code>service</code>	Query	예	-	서비스 이름
<code>lines</code>	Query	아니오	<code>100</code>	반환할 최대 줄 수
<code>offset</code>	Query	아니오	<code>0</code>	파일 읽기 시작 위치 (바이트)

요청:

```
# 초기 로드 (최근 100줄)
curl -b "tk_session=<TOKEN>" \
  "https://서버주소/tkadmin/api/service/logs/file?service=TACHYON-
  Package&lines=100"

# 증분 읽기 (offset 이후의 새 줄만)
curl -b "tk_session=<TOKEN>" \
  "https://서버주소/tkadmin/api/service/logs/file?service=TACHYON-
  Package&lines=100&offset=45678"
```

응답 필드:

필드	타입	설명
<code>lines</code>	Array	로그 줄 배열 (각 줄: <code>{line, level}</code>)
<code>offset</code>	Integer	다음 읽기 시작 위치 (바이트)
<code>fileSize</code>	Integer	현재 파일 크기 (바이트)
<code>rotated</code>	Boolean	로그 로테이션 감지 여부

응답 예시:

```
{
  "lines": [
    {"line": "2026-02-19 08:10:00 [INFO] Service started", "level": "info"},
    {"line": "2026-02-19 08:10:01 [WARN] Connection retry", "level": "warn"}
  ],
  "offset": 46200,
  "fileSize": 46200,
```

```
"rotated": false  
}
```

**팁**

`offset` 을 이전 응답의 `offset` 값으로 지정하면 새로 추가된 줄만 반환됩니다. `rotated` 가 `true` 이면 로그 파일이 로테이션되어 `offset`이 0으로 리셋된 것을 의미합니다.

메인 서버 API

메서드	경로	설명	인증
GET	/tkadmin/api/status	통합 상태 조회	필요
GET	/tkadmin/api/stream	SSE 실시간 상태 Push	필요
GET	/tkadmin/api/system/stats	시스템 리소스 통계	필요
GET	/tkadmin/api/services	서비스 목록 및 리소스	필요
GET	/tkadmin/api/config	설정 조회	필요
POST	/tkadmin/api/config	설정 변경	필요
POST	/tkadmin/api/config/restart	서비스 재시작	필요
GET	/tkadmin/api/nav-config	네비게이션 설정	불필요
GET	/tkadmin/api/config/external/files	외부 설정 파일 목록	필요
GET	/tkadmin/api/config/external/file	외부 설정 파일 내용	필요
POST	/tkadmin/api/config/external/file	외부 설정 파일 저장	필요
POST	/tkadmin/api/service/:action	서비스 제어	필요
POST	/tkadmin/api/monitor/report	외부 보고 수신	필요
GET	/tkadmin/api/alerts	알림 목록	필요
GET	/tkadmin/api/alerts/unread	미읽음 알림 수	필요
POST	/tkadmin/api/alerts/read	전체 읽음 처리	필요
GET	/tkadmin/api/alerts/:id	알림 상세	필요
GET	/tkadmin/api/audit-logs	감사 로그	필요
GET	/tkadmin/api/system/os-checks	OS 환경 체크	필요
POST	/tkadmin/api/system/os-checks/selinux	SELinux 설정	필요
POST	/tkadmin/api/system/os-checks/firewall	방화벽 설정	필요

메서드	경로	설명	인증
POST	/tkadmin/api/system/os-checks/limits	OS 리미트 설정	필요
GET	/tkadmin/api/logs	로그 조회	필요
GET	/tkadmin/api/service/logs/file/list	서비스 파일 로그 목록	필요
GET	/tkadmin/api/service/logs/file	서비스 파일 로그 읽기	필요

Recovery 서버 API


메서드	경로	설명
GET	/recovery/	Recovery 로그인 페이지
POST	/recovery/login	OS 계정 인증
GET	/tkadmin/api/status	상태 조회
GET	/tkadmin/api/config	설정 조회
POST	/tkadmin/api/config	설정 변경
POST	/tkadmin/api/config/restart	서비스 재시작
GET	/tkadmin/api/alerts	알림 조회
GET	/tkadmin/api/audit-logs	감사 로그
GET	/tkadmin/api/alerts/:id	알림 상세

15. 웹 터미널

v0.6.2 — 브라우저에서 tkctl CLI를 직접 사용하는 웹 터미널

15.1 개요

웹 터미널은 브라우저에서 직접 `tkctl` 대화형 CLI 셸을 사용할 수 있는 기능입니다. 별도의 SSH 접속이나 터미널 프로그램 없이도 tkctl의 모든 명령어와 자동완성 기능을 웹 UI에서 바로 사용할 수 있습니다.

 웹 터미널은 서버에서 실행되는 `tkctl --shell` 프로세스에 PTY(가상 터미널)를 할당하여 완전한 터미널 경험을 제공합니다. go-prompt 기반의 **Tab 자동완성**, **명령어 히스토리** 등이 모두 지원됩니다.

15.2 접근 방법

1. 사이드바 하단의 "**도구**" 섹션을 찾습니다.
2. "**터미널**" 메뉴를 클릭합니다.
3. 화면이 터미널 모드로 전환되며, 자동으로 서버에 연결됩니다.

15.3 화면 구성

상태 표시줄

터미널 상단에 연결 상태를 표시하는 바가 있습니다:

상태	표시	설명
연결됨	 연결됨	서버와 정상 연결된 상태
연결 중	 연결 중...	서버에 연결을 시도하는 중
연결 끊김	 연결 끊김	서버와의 연결이 끊어진 상태
재연결 중	 재연결 중 (N/5)	자동 재연결을 시도하는 중

상태 표시줄 우측의 **새로고침 버튼**을 클릭하면 수동으로 재연결할 수 있습니다.

터미널 영역

- `tkctl>` 프롬프트가 표시되면 명령어를 입력할 수 있습니다.
- 화면 크기에 맞게 자동으로 리사이즈됩니다.
- 다크/라이트 테마에 따라 터미널 색상이 자동으로 변경됩니다.

15.4 사용 방법

기본 명령어

터미널에서 다음과 같은 tkctl 명령어를 사용할 수 있습니다:

```
tkctl> help                # 사용 가능한 명령어 목록 표시
tkctl> service check       # 전체 서비스 상태 확인
tkctl> service start <서비스명> # 서비스 시작
tkctl> service stop <서비스명>  # 서비스 중지
tkctl> service restart <서비스명> # 서비스 재시작
tkctl> analyze             # 시스템 분석
tkctl> env check           # 환경 점검
tkctl> exit                # 터미널 종료
```

Tab 자동완성

명령어 입력 중 **Tab** 키를 누르면 사용 가능한 명령어와 옵션이 자동완성 목록으로 표시됩니다:

```
tkctl> ser[Tab]
> service      서비스 관리 명령
```



```
tkctl> service [Tab]
> check      서비스 상태 확인
> start      서비스 시작
> stop       서비스 중지
> restart    서비스 재시작
```

특수 키

키	동작
Tab	자동완성
Ctrl+C	현재 입력 취소
Ctrl+D	터미널 종료 (= <code>exit</code>)
↑ / ↓	명령어 히스토리 탐색
clear	화면 지우기

15.5 연결 관리

자동 재연결

네트워크 불안정 등으로 연결이 끊어지면 터미널이 자동으로 재연결을 시도합니다:

- 재시도 간격: 3초
- 최대 재시도: 5회
- 5회 시도 후에도 연결되지 않으면 상태 표시줄에 "연결 끊김"이 표시됩니다.
- 상태 표시줄의 **새로고침 버튼**을 클릭하면 재시도 횟수가 초기화되고 다시 연결을 시도합니다.

세션 관리

- 다른 메뉴로 이동하면 현재 터미널 세션이 자동으로 종료됩니다.
- 터미널 메뉴로 다시 돌아오면 새 세션이 시작됩니다.
- 서버당 최대 **5개**의 동시 터미널 세션이 허용됩니다.
- **30분** 동안 키 입력이 없으면 세션이 자동으로 종료됩니다.

15.6 테마 지원

터미널은 tkadmin의 다크/라이트 테마에 자동으로 대응합니다:

- **다크 모드:** Tokyo Night 스타일 컬러 스킴 (어두운 배경 + 밝은 텍스트)
- **라이트 모드:** 밝은 배경 + 어두운 텍스트

상단 헤더의 테마 전환 버튼으로 테마를 변경하면 터미널 색상이 즉시 반영됩니다.

15.7 제한사항

항목	설명
실행 환경	Linux 서버에서만 동작 (PTY 필요)
권한	tkadmin 서버 프로세스와 동일한 OS 권한으로 실행
동시 세션	서버당 최대 5개
유휴 타임아웃	30분 무입력 시 자동 종료
브라우저 지원	WebSocket을 지원하는 모든 최신 브라우저

15.8 트러블슈팅

연결이 되지 않을 때

1. **서버 상태 확인:** tkadmin 서비스가 정상 실행 중인지 확인합니다.
2. **tkctl 바이너리 확인:** 서버에 tkctl 바이너리가 설치되어 있는지 확인합니다.

```
ls -la /usr/local/tkadmin/bin/tkctl
```

3. **방화벽 확인:** WebSocket 연결(포트 13700)이 방화벽에 의해 차단되지 않는지 확인합니다.
4. **세션 한도 확인:** 최대 동시 세션(5개)에 도달하지 않았는지 확인합니다.

입력이 반영되지 않을 때

- 상태 표시줄이 "**연결됨**"(녹색)인지 확인합니다.
- 연결이 끊어졌다면 새로고침 버튼을 클릭하여 재연결합니다.
- 브라우저 콘솔(F12)에서 WebSocket 관련 오류 메시지를 확인합니다.

99. 자주 묻는 질문 (FAQ)

tkadmin 운영 중 자주 발생하는 문제와 해결 방법을 안내합니다.

99.1 Q1. 로그인에 안 됩니다

Auth Bridge 동작 원리

tkadmin은 TACHYON 인증 시스템과 연동하는 **Auth Bridge** 방식으로 로그인을 처리합니다. 동작 흐름은 다음과 같습니다:

1. 사용자가 `/tkadmin/` 에 접속하면 `tk_session` 쿠키 존재 여부를 확인합니다.
2. 쿠키가 없으면 `/tkadmin/bridge` 페이지로 리다이렉트합니다.
3. Bridge 페이지에서 브라우저의 `localStorage` 및 `sessionStorage` 를 스캔하여 TACHYON JWT 토큰을 탐색합니다.
4. 토큰을 발견하면 서버로 전송하여 JWT 유효성을 검증하고, Redis에서 실제 세션 존재 여부를 확인합니다.
5. 검증에 성공하면 `tk_session` 쿠키를 설정하고 대시보드로 진입합니다.

확인 사항

TACHYON 인증 서버 상태 확인

TACHYON Auth 서비스가 정상 구동 중인지 확인합니다:

```
systemctl status TACHYON-Auth1
```

서비스가 중지된 경우 시작합니다:

```
systemctl start TACHYON-Auth1
```

Redis 연결 상태 확인

tkadmin은 Redis에 저장된 TACHYON 세션 데이터(`TOKEN:GUID`)를 실시간으로 검증합니다. Redis 연결에 문제가 있으면 로그인이 실패합니다.

```
# Redis 서비스 상태 확인
systemctl status redis

# Redis 포트 확인
netstat -nlpt | grep redis

# Redis 연결 테스트
redis-cli -h 127.0.0.1 -p 6379 ping
```

tkadmin 로그에서 Redis 관련 경고를 확인할 수 있습니다:

```
grep "REDIS" /usr/local/TACHYON/TTS40/logs/tkadmin.log
```

Recovery Mode 접근

TACHYON 인증 서버 자체에 장애가 있는 경우, Recovery Mode를 통해 긴급 접근할 수 있습니다. 자세한 내용은 [Recovery Mode 가이드](#)를 참조하세요.

기본 Recovery 포트: **13701** (메인 포트 + 1)

```
http://서버주소:13701/recovery/
```

99.2 Q2. 서비스가 시작되지 않습니다

서비스 상태 확인

```
systemctl status tkadmin
```

상태가 **failed** 인 경우 상세 로그를 확인합니다:

```
journalctl -u tkadmin -n 50 --no-pager
```

포트 충돌 확인

tkadmin의 기본 포트는 **13700**입니다. 해당 포트가 이미 사용 중인지 확인합니다:

```
netstat -nlpt | grep 13700
# 또는
ss -nlpt | grep 13700
```

다른 프로세스가 포트를 점유하고 있다면, 해당 프로세스를 종료하거나 `tkadmin.yml` 에서 포트를 변경합니다.

로그 파일 확인

```
tail -50 /usr/local/TACHYON/TTS40/logs/tkadmin.log
```

주요 오류 패턴:

- `ListenAndServe failed` : 포트 바인딩 실패 (포트 충돌 또는 권한 부족)
- `Database initialization failed` : SQLite DB 초기화 실패
- `Logger not initialized` : 로그 경로 설정 오류

PID 싱글톤 체크

tkadmin은 동시에 하나의 인스턴스만 실행되도록 `tkadmin.pid` 파일을 사용합니다. 비정상 종료 후 PID 파일이 남아 있으면 새로운 인스턴스가 시작되지 않을 수 있습니다.

```
# PID 파일 확인
cat /usr/local/TACHYON/TTS40/tkadmin.pid

# 해당 PID의 프로세스가 실제로 실행 중인지 확인
ps -p $(cat /usr/local/TACHYON/TTS40/tkadmin.pid)

# 프로세스가 없으면 PID 파일 삭제 후 재시작
rm /usr/local/TACHYON/TTS40/tkadmin.pid
systemctl start tkadmin
```

99.3 Q3. 설정 변경 후 반영되지 않습니다

포트 변경 시 재시작 필요

`port` 값을 변경한 경우 서비스를 재시작해야 적용됩니다. 웹 UI에서 설정을 저장하면 포트 변경이 감지될 때 재시작 확인 대화상자가 표시됩니다.

```
systemctl restart tkadmin
```

NGINX 자동 동기화 확인

tkadmin은 설정 저장 시 NGINX `proxy_pass` 포트를 자동으로 동기화합니다. NGINX 설정이 올바르게 반영되었는지 확인합니다:

```
cat /usr/local/TACHYON/TTS40/nginx/conf/conf.d/tkadmin.location
```

`proxy_pass` 뒤의 포트 번호가 `tkadmin.yml` 의 `port` 값과 일치하는지 확인하세요.

tkadmin.yml 파일 직접 확인

설정 파일의 실제 내용을 확인합니다:

```
cat /usr/local/TACHYON/TTS40/tkadmin.yml
```



팁

YAML 문법 오류가 있으면 설정 파일 전체가 무시되고 기본값이 적용됩니다. YAML 들여쓰기에 탭 (Tab) 대신 공백(Space)을 사용하고 있는지 확인하세요.

99.4 Q4. 로그가 표시되지 않습니다

로그 파일 경로 확인

기본 로그 파일 경로는 `/usr/local/TACHYON/TTS40/logs/tkadmin.log` 입니다. 파일이 존재하는지 확인합니다:

```
ls -la /usr/local/TACHYON/TTS40/logs/tkadmin.log
```

파일이 없는 경우 `tkadmin.yml` 의 `logging.file` 설정을 확인합니다:

```
logging:
  file: "logs/tkadmin.log"
```

⚠ 주의

`logging.file` 값이 빈 문자열이거나 디렉토리 경로만 지정되면 치명적인 버그가 발생할 수 있습니다. 반드시 파일명을 포함한 전체 경로를 입력하세요.

로그 레벨 설정 확인

`tkadmin.yml` 에서 로그 레벨을 확인합니다:

```
logging:
  level: "info"    # debug, info, warn, error
```

상세한 로그를 보려면 `debug` 로 변경합니다. `debug` 레벨은 `tkadmin.yml` 의 `debug: true` 설정이 활성화된 경우에도 자동으로 적용됩니다.

브라우저 캐시 초기화

웹 UI에서 로그가 표시되지 않는 경우 브라우저 캐시를 초기화합니다:

- **Chrome/Edge:** `Ctrl + Shift + Delete` > 캐시된 이미지 및 파일 삭제
- 또는 `Ctrl + Shift + R` 로 강제 새로고침

99.5 Q5. 알림 배너가 사라지지 않습니다

Active FAILURE vs 읽음 처리 차이

tkadmin의 알림 시스템은 두 가지 독립적인 상태를 관리합니다:

구분	설명
Active FAILURE 배너	현재 장애가 발생했으나 아직 복구되지 않은 서비스에 대한 빨간색 상단 배너
읽음/미읽음 상태	관리자가 해당 알림을 확인(읽음 처리)했는지 여부

배너 자동 숨김 정책

- 장애가 발생한 서비스에 대해 **복구 성공(RECOVERY_SUCCESS)** 알림이 수신되면, 해당 장애 배너는 **즉시 자동으로 숨김** 처리됩니다.
- 그러나 배너가 사라지더라도 **알림 뱃지(빨간 점)**와 **관리자 보고 목록의 미읽음 상태는 유지**됩니다.

읽음 처리 방법

관리자 보고 페이지에서 읽음 처리를 수행합니다:

1. 사이드바에서 **관리자 보고** 메뉴를 클릭합니다.
2. 미확인(미읽음) 항목이 시각적으로 강조 표시됩니다.
3. **전체 읽음** 버튼을 클릭하면 모든 미읽음 알림이 읽음 처리됩니다.



팁

이 설계는 감사 추적성을 보장하기 위한 것입니다. 배너가 자동으로 사라지더라도, 관리자가 직접 읽음 처리하기 전까지는 사후 장애 이력을 확인할 수 있도록 미읽음 상태가 유지됩니다.

99.6 Q5-1. 서비스가 자체 업데이트 중인데 장애 알림이 발생합니다

Grace Period (Lazy Loading) 메커니즘

tkadmin의 Watchdog 엔진은 서비스가 자체적으로 설정 적용이나 업데이트를 위해 짧게 재시작되는 경우를 자동으로 감지하여, 불필요한 장애 알림을 억제하는 **Grace Period(유예 시간)** 메커니즘을 제공합니다.

정상 동작인 경우

다음과 같은 상황에서는 Watchdog이 자동으로 유예 시간을 부여합니다:

- 서비스가 **auto-restart**, **start**, **stop** 등 **전환 상태**인 경우 → **15초** 유예
- 서비스가 **dead**, **failed** 상태이지만 직후 재시작되는 경우 → **5초** 유예

유예 시간 내에 서비스가 자동 복구되면, 장애 알림 없이 정상 처리됩니다.

알림이 계속 발생하는 경우

Grace Period(15초) 이후에도 서비스가 복구되지 않으면 실제 장애로 판단하여 알림이 발생합니다. 이 경우:

1. 해당 서비스의 로그를 **시스템 로그** 화면에서 확인합니다.
2. **journalctl -u <서비스명> -n 50 --no-pager** 명령으로 systemd 로그를 확인합니다.
3. 서비스의 설정 파일에 문법 오류가 없는지 점검합니다.

**팁**

Grace Period 기능은 자동으로 동작하며, 별도 설정 없이 활성화됩니다. 서비스 자체 업데이트로 인한 짧은 재시작(수 초~15초 이내)에서는 장애 알림이 발생하지 않습니다.

99.7 Q6. 긴급 복구 모드로 접근하려면?

Recovery 포트 접근 방법

TACHYON 인증 서버에 장애가 발생하여 정상적인 로그인에 불가능할 때, Recovery Mode를 통해 핵심 관리 기능에 접근할 수 있습니다.

기본 Recovery 포트는 **메인 포트 + 1** (기본값: **13701**)입니다.

```
http://서버주소:13701/recovery/
```

**주의**

Recovery Mode는
Linux 환경에서만
동작합니다.

OS 계정(root) 인증

Recovery Mode는 TACHYON JWT 인증 대신 **운영체제 PAM 인증**을 사용합니다. Linux 시스템의 실제 계정(예: **root**)으로 로그인합니다.

- 로그인 가능 조건: **/etc/passwd** 에서 해당 사용자의 셸이 **/sbin/nologin** 또는 **/bin/false** 가 아닌 경우
- 인증 성공 시 **tk_recovery** 쿠키로 세션이 유지됩니다 (1시간 유효)

emergency_ips 설정 확인

Recovery 포트 접근은 IP 기반 ACL(접근 제어 목록)로 제한됩니다:

- 자동 허용: **127.0.0.1**, **::1** (localhost)
- 추가 허용: **tkadmin.yml** 의 **emergency_ips** 목록에 등록된 IP

```
# tkadmin.yml 설정 예시
emergency_ips:
  - "10.10.1.100"
  - "192.168.1.50"
```

허용되지 않은 IP에서 접근하면 **403 Forbidden** 페이지가 표시되며, 클라이언트 IP와 차단 사유가 안내됩니다.



팁

Recovery Mode에서 제공되는 설정 변경 API를 통해 **emergency_ips** 목록에 본인의 IP를 추가할 수 있습니다. 다만 이를 위해서는 먼저 서버에 SSH로 접속하여 localhost에서 Recovery 포트에 접근해야 합니다.

99.8 Q7. TACHYON 대시보드에 Admin 메뉴가 보이지 않습니다

인젝터 스크립트 설치 여부 확인

tkadmin 관리 메뉴가 TACHYON 대시보드 사이드바에 표시되려면 인젝터 스크립트가 설치되어 있어야 합니다.

```
# 인젝터 스크립트 파일 확인
ls -la /usr/local/TACHYON/TTS40/front/html/tkadmin_injector.js

# index.html에 스크립트 태그가 삽입되었는지 확인
grep "tkadmin_injector" /usr/local/TACHYON/TTS40/front/html/index.html
```

인젝터가 설치되어 있지 않다면 재설치합니다:

```
sudo ./tkadmin -i
```

allowed_ids 설정 확인

tkadmin.yml의 **allowed_ids** 설정을 통해 tkadmin 메뉴가 표시될 TACHYON 사용자 ID를 제어할 수 있습니다.

```
allowed_ids:
  - "tsadmin"
  - "operator01"
```

현재 로그인한 TACHYON 계정 ID가 `allowed_ids` 목록에 포함되어 있는지 확인하세요.

show_link 활성화 여부 확인

`tkadmin.yml` 에서 `show_link` 설정이 활성화되어 있는지 확인합니다:

```
show_link: true
```

`show_link: false` (기본값)인 경우 인젝터 메뉴가 표시되지 않습니다.



팁

`show_link` 와 `allowed_ids` 설정은 `/tkadmin/api/nav-config` API를 통해 인젝터 스크립트에 전달됩니다. 이 API는 인증 없이 접근 가능하므로 TACHYON 대시보드에서 바로 참조할 수 있습니다.

99.9 Q8. 전문가 편집기에서 저장이 실패합니다

YAML 문법 오류 확인

tkadmin의 전문가 편집기는 `.yml` 및 `.yml_dev` 파일 저장 시 YAML 구문 유효성을 자동으로 검사합니다. 문법 오류가 있으면 저장이 거부되며 오류 메시지가 표시됩니다.

주요 YAML 문법 오류 원인:

- **탭(Tab) 사용:** YAML은 들여쓰기에 탭을 허용하지 않습니다. 공백(Space)만 사용하세요.
- **콜론 뒤 공백 누락:** `key:value` 가 아닌 `key: value` 형태여야 합니다.
- **따옴표 미닫힘:** 문자열에 특수문자(`#` , `:` , `{` , `}` 등)가 포함된 경우 따옴표로 감싸야 합니다.

파일 권한 확인

대상 파일에 대한 쓰기 권한이 있는지 확인합니다:

```
ls -la /usr/local/TACHYON/TTS40/conf/파일명
```

tkadmin은 **root** 권한으로 실행되므로 일반적으로 권한 문제가 발생하지 않지만, SELinux가 활성화된 환경에서는 추가 설정이 필요할 수 있습니다.

Path Traversal 차단 안내

보안을 위해 다음과 같은 경로 접근은 차단됩니다:

- **..** 을 포함하는 상대 경로 (상위 디렉토리 탐색)
- **/** 로 시작하는 절대 경로
- **** 로 시작하는 Windows 스타일 경로

편집 가능한 파일은 **target_dir** (기본: **/usr/local/TACHYON/TTS40/**) 하위의 설정 파일로 제한됩니다.

주의

전문가 편집기로 저장할 때마다 기존 파일이 ***.YYYYMMDDHHMMSS.bak** 형식으로 자동 백업됩니다. 잘못된 수정이 있더라도 백업 파일을 통해 원복할 수 있습니다.

