M05-03 · GIBALJA VISUAL MANUAL

데이터 보존·백업·파기 설계하기

한 문장 목표: 데이터의 `목적 → 복사본 → 보존 시계 → 복구 계약 → 예외 → 파기 → 증거`를 한 생애주기로 연결하고, “백업이 있다”를 “정해진 시점까지 실제로 복원했고 다시 삭제할 수 있다”로 바꿉니다.

난이도	개념	실습	셀프 테스트	최종 산출물
Level 1	60분	60분	15분	생애주기 표, 보존·파기 매트릭스, 복원 훈련 기록, 삭제·hold 판정서

운영 DB의 row를 지웠다고 데이터가 사라진 것은 아닙니다. 같은 값이 replica, 검색 색인, cache, 분석 warehouse, log, 개발용 CSV, 외부 SaaS, snapshot, base backup, WAL에 남을 수 있습니다. 반대로 backup 파일이 존재한다고 복구할 수 있는 것도 아닙니다. 필요한 parent·WAL·암호화 key·configuration이 있고 실제로 기동·검증해 본 경우에만 복구 계약이 증명됩니다.

그림 1. 생애주기는 저장 기간 표 하나가 아니라 여덟 판단의 연결입니다. 앞 gate가 비어 있으면 뒤의 자동 삭제·복원·감사 증거도 흔들립니다.


1
목적
왜 필요한가
2
분류
민감도·법적 근거
5
복구
RPO·RTO
6
예외
hold·분쟁

ckup expiry + restore 재삼제 + 검증 가능한 기록

0. 이 PDF를 공부하는 방법

1회차 · 그림만 읽기 · 20분

그림 1부터 14까지 제목과 맨 아래 결론 띠만 읽습니다. 다음 여덟 문장을 소리 내어 말합니다.

```
한 row가 아니라 그 row의 모든 copy 위치를 찾는다.
보존 규칙은 trigger + duration + exception + action + owner다.
replica, backup, archive는 목적과 실패 방식이 다르다.
RPO는 복구 지점, RTO는 복구 시간의 목표다.
backup 무결성 검사와 실제 restore 훈련은 다르다.
삭제 요청은 primary DELETE가 아니라 copy sweep이다.
legal hold는 범위와 종료 조건이 있는 예외다.
파기 완료는 작업 로그가 아니라 verification + validation + 승인 증거다.
```

2회차 · 한 데이터의 복사본 지도 그리기 · 25분

내 서비스에서 개인정보 하나를 고릅니다. 예: 이메일, 전화번호, 학습 답안. 운영 DB 외에 존재할 수 있는 위치를 적습니다.

위치	실제 존재 여부	생성 주체	owner	삭제 방법	복원 시 재생성
primary DB					
read replica					
search·cache					
analytics·AI feature					
logs					
CSV·external vendor					
backup					

3회차 · 20개 시나리오 판정 · 50분

데이터 생애주기 스튜디오를 열어 시나리오를 순서대로 실행합니다. `RPO·RTO`, `legal hold`, `backup chain`, `복사본 상태`를 하나씩 바꾸고 outcome이 왜 달라지는지 설명합니다.

4회차 · 내 서비스의 증거서 작성 · 40분

데이터 보존·백업·파기 증거서를 채웁니다. 최소 데이터 class 3개, copy 위치 7개, restore drill 1회, 삭제 요청 1건의 모의 기록을 만듭니다.

1. 생애주기는 목적과 증거 사이의 계약입니다

데이터 설계는 table을 만든 순간 끝나지 않습니다. 데이터가 필요한 동안에는 정확성·가용성·기밀성을 지키고, 필요가 끝나면 다른 의무를 확인한 뒤 사용을 멈추고 안전하게 파기해야 합니다. 그 전 과정을 다음 질문으로 연결합니다.

gate	핵심 질문	최소 증거
1 목적	왜 이 데이터를 처리합니까	purpose·lawful basis·업무 owner
2 분류	유출·손실·변조 시 영향은 무엇입니까	민감도·개인정보 여부·복구 중요도
3 복사본	어디에 원본·파생본이 있습니까	copy map·vendor·data flow
4 보존	언제부터 언제까지 남깁니까	trigger·duration·exception·action
5 복구	얼마나 과거로, 얼마나 빨리 복구합니까	RPO·RTO·backup chain
6 예외	삭제를 멈출 유효한 의무가 있습니까	법적 근거·범위·owner·review date
7 파기	어떤 copy를 어떤 방법으로 닫습니까	삭제·비식별·접근 금지·만료 결과
8 증거	누가 결과가 충분하다고 승인했습니까	verification·validation·실패·승인

이 교재의 사례

`yeoncore 학습 서비스`에는 다음 데이터가 있다고 가정합니다.

data class	한 record의 뜻	사용 목적	대표 copy
회원 프로필	한 회원의 연락·계정 정보	로그인·안내	DB, replica, search, support SaaS, backup
학습 답안	한 회원의 한 문제 응답	채점·피드백	DB, analytics, AI feature, log, backup
결제 기록	한 결제의 금액·상태	결제·정산·분쟁 대응	DB, payment vendor, finance export, backup
보안 audit	한 보안 사건·접근 기록	탐지·조사·입증	log store, archive, backup

같은 회원과 연결되어도 목적·법적 근거·보존 trigger가 다를 수 있습니다. 그러므로 “회원 데이터 3년”처럼 table 단위 하나의 규칙으로 뭉개지 않습니다.

30초 확인 1
회원 탈퇴 시 결제 기록과 학습 답안을 무조건 같은 날 같은 방법으로 지워야 합니까? 서로 다른 판단에 필요한 열을 세 가지 말해 보세요.

2. 먼저 한 row의 모든 copy를 지도에 올립니다

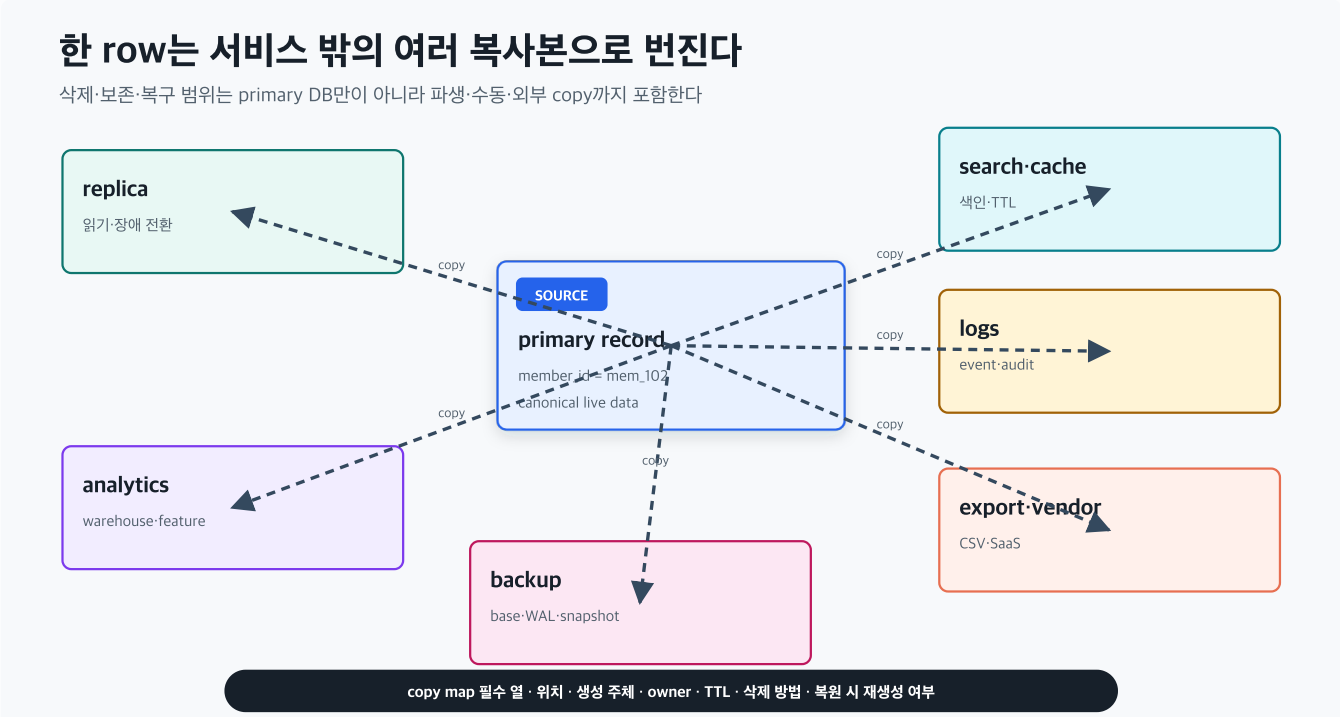
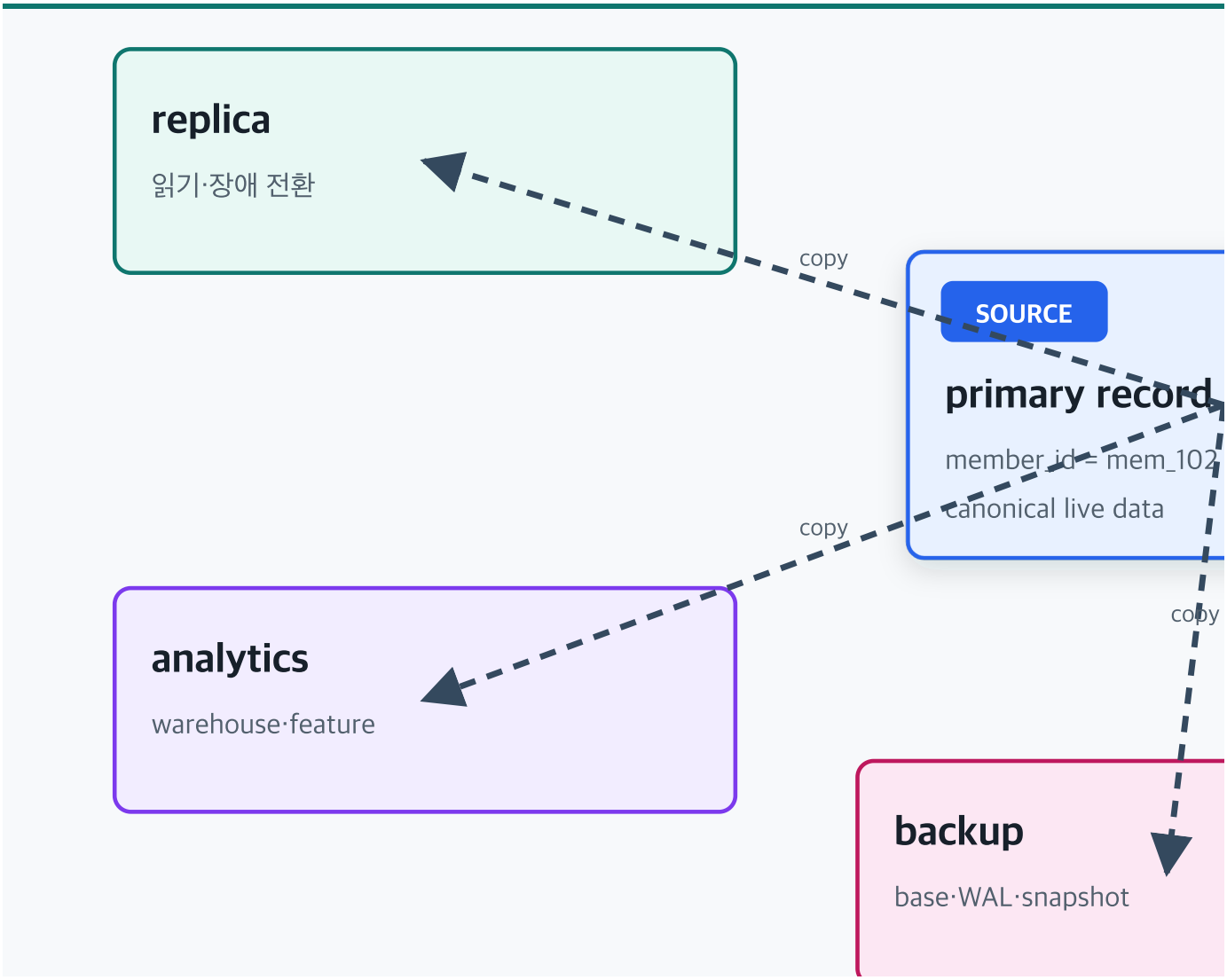
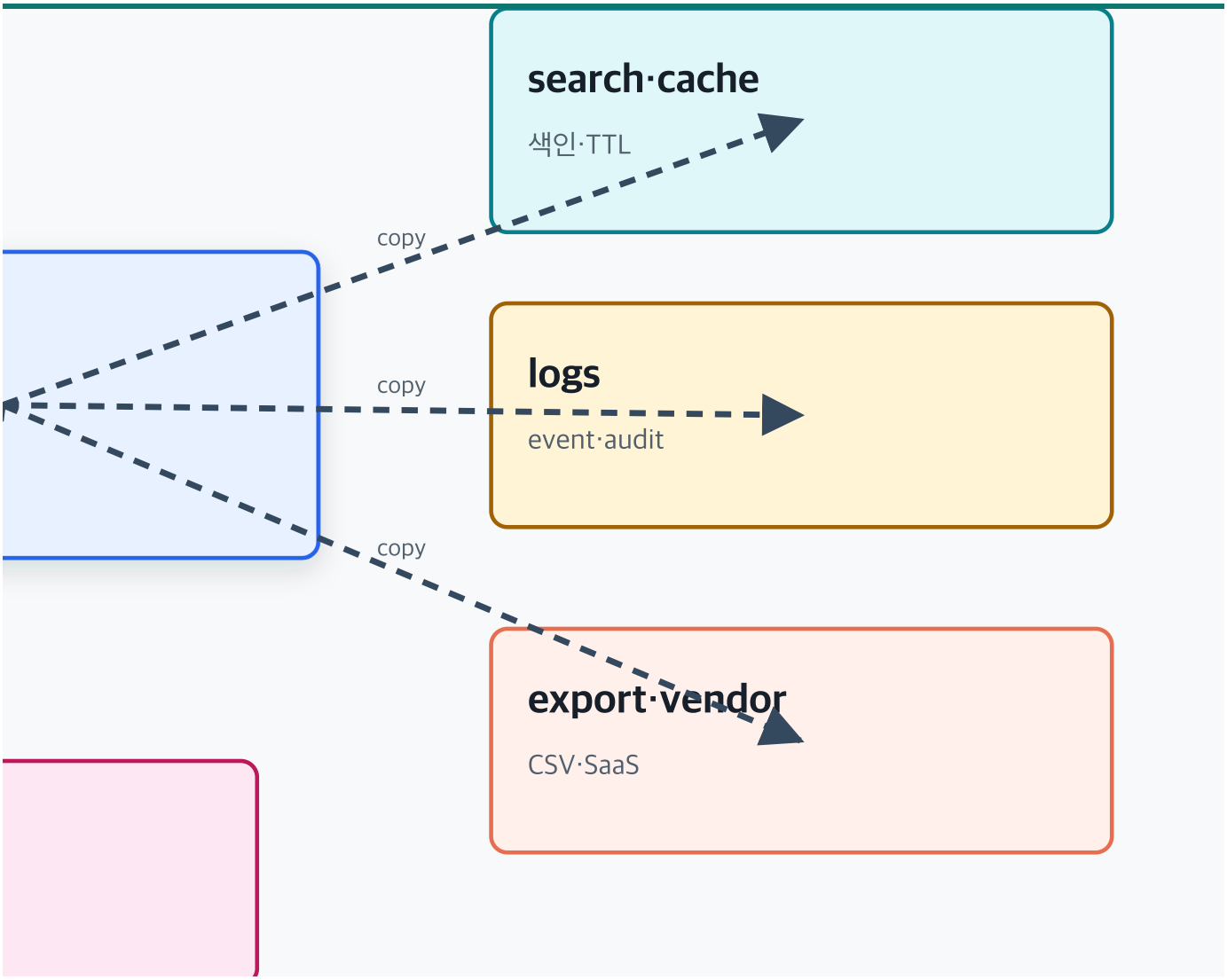


그림 2. 데이터는 code가 만든 자동 copy뿐 아니라 사람이 내려받은 CSV, 외부 처리자, 개발·시험 환경에도 남습니다. 찾지 못한 copy는 삭제할 수도 복구할 수도 없습니다.


replica
읽기·장애 전환
copy
SOURCE
primary record
member_id = mem_102
canonical live data
copy
copy
analytics
warehouse·feature
backup
base·WAL·snapshot


search·cache
색인·TTL
copy
logs
event·audit
copy
copy
export·vendor
CSV·SaaS

2.1. copy는 값이 완전히 같지 않아도 됩니다

다음은 모두 원본에서 파생된 copy입니다.

- 이메일 원문을 가진 회원 row
- 이메일을 lower-case로 바꾼 검색 색인
- 회원 ID와 행동을 결합한 analytics event
- 이메일 일부를 가린 support ticket
- 회원 ID로 만든 AI feature
- 삭제된 row가 아직 들어 있는 과거 backup

삭제·정정·접근 요청의 범위를 정할 때는 직접 식별자뿐 아니라 같은 사람·record로 다시 연결할 수 있는 파생값과 reference도 확인합니다.

2.2. copy register의 필수 열

열	이유	예시
system·location	대상을 찾는 주소	`member-db`, `search/member-v3`
copy type	역할·실패 모드를 구분	primary, replica, derived, export, backup
fields·subject key	어떤 값을 어떻게 찾는지	`member_id`, email hash
creator·frequency	copy가 다시 생기는 조건	CDC 5초, nightly ETL
owner·operator	결정·실행 책임	data owner, platform owner
access·credential	함께 손상될 가능성	production role, separate backup role
retention·TTL	자동 만료 여부	cache 24h, backup 35d
deletion method	row·partition·object 단위 행동	DELETE, purge index, expire object
restore behavior	과거 copy가 재등장하는지	restore 후 suppression 필요
evidence	완료를 무엇으로 확인하는지	count, job ID, vendor confirmation

2.3. data map을 찾는 순서

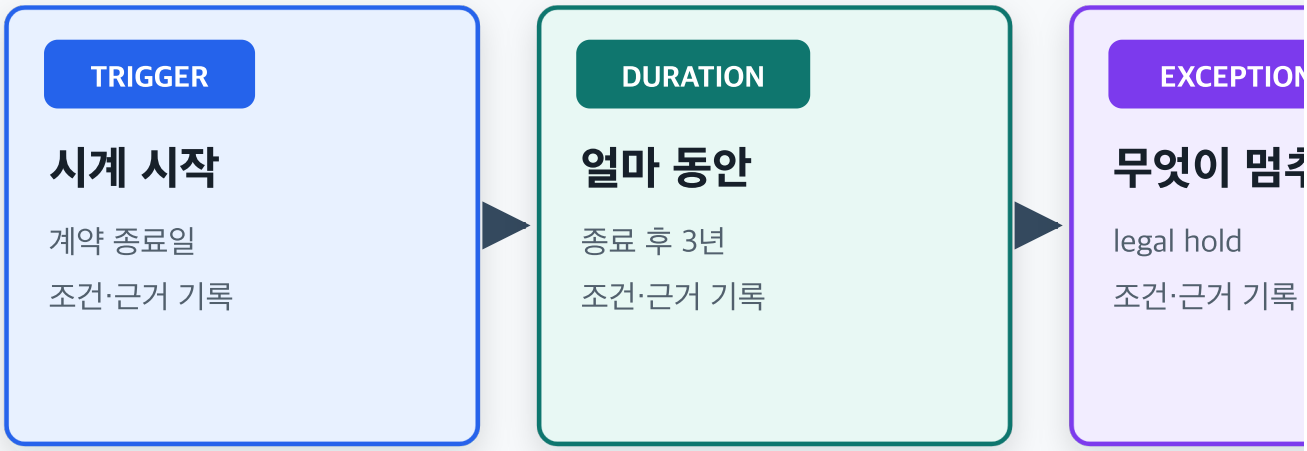
1. 데이터 모델과 API response에서 canonical field를 찾습니다.
2. CDC·queue·ETL·search indexing·analytics event를 따라갑니다.

3. log·error report·observability label을 확인합니다.
4. 관리자 export·정기 report·support attachment를 확인합니다.
5. 외부 vendor와 processor의 저장·재위탁 위치를 확인합니다.
6. dev·test·local dump·notebook·AI training·feature store를 확인합니다.
7. snapshot·base backup·WAL·object version·offline copy를 확인합니다.

"DB에서 찾을 수 없음"은 파기 증거가 아닙니다.
검색 색인, log, object storage, 외부 vendor, backup에서 같은 subject를 찾는 방법이 없다는 뜻일 수 있습니다.
찾기 어려운 copy를 만들었다면 생성 시점부터 subject reference·만료·삭제 절차를 설계합니다.

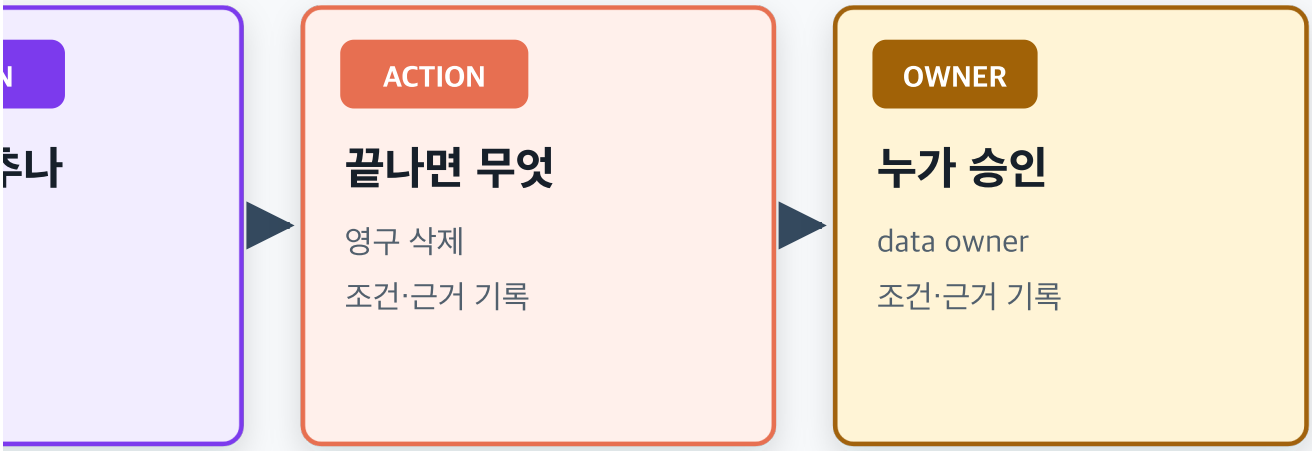
3. 보존 규칙은 다섯 요소의 문장입니다

그림 3. `3년 보관`만으로는 시작일·예외·종료 행동·승인자를 알 수 없습니다. 다섯 요소를 한 문장으로 만들어야 자동화와 검토가 가능합니다.



예: 계약 종료일로부터 3년, 분장

hold 해제 후 privacy owner 승인으로 live·deri



보존 명령이 있으면 일시 정지

ved copy를 파기하고 backup 만료를 추적한다.

3.1. trigger · 시계는 언제 시작합니까

가능한 trigger는 수집일, 생성일, 마지막 거래일, 계약 종료일, 회원 탈퇴일, 분쟁 종료일, 회계연도 종료일처럼 서로 다릅니다. `created_at`과 `contract_ended_at`을 바꿔 쓰면 만료일이 몇 년씩 달라질 수 있습니다.

3.2. duration · 얼마 동안입니까

기간에는 단위와 계산 규칙이 필요합니다.

```
나쁜 규칙: 오래 보관한다.
불완전한 규칙: 3년 보관한다.
좋은 규칙: 계약 종료일 00:00 UTC를 trigger로 3년 뒤 월말까지 보존한다.
```

3.3. exception · 무엇이 예정 파기를 멈춥니까

다른 법률에 따른 보존, 유효한 분쟁·조사 보존, 이용자와의 계약상 의무처럼 근거가 있는 예외를 정의합니다. 예외에는 범위·시작·owner·검토일·해제 조건이 있어야 합니다.

3.4. action · 끝나면 무엇을 합니까

가능한 action은 영구 삭제, 안전한 비식별 처리, 접근 제한 후 backup 만료, 저장매체 Clear·Purge·Destroy, 다른 법률 보존을 위한 분리 저장 등입니다. “archive로 이동”은 최종 행동이 아니라 다음 상태일 수 있습니다.

3.5. owner · 누가 근거와 결과를 승인합니까

시스템 운영자가 버튼을 누를 수 있어도 보존 근거를 혼자 결정하지는 않습니다. 최소한 업무 owner, data·privacy owner, platform operator의 책임을 나눕니다. 법적 충돌은 해당 관할과 조직 상황을 아는 법률·개인정보 담당자가 검토합니다.

보존 규칙 예시

data class	trigger	duration	exception	action	owner
학습 답안 원본	강좌 종료	1년	성적 이의 처리 중	live·analytics 삭제, backup 만료	learning owner
회원 프로필	탈퇴 완료	지체 없이 처리	다른 법률 보존 대상 분리	live·search·vendor 삭제	privacy owner
결제 거래	거래 종료	해당 근거로 정한 기간	분쟁 hold	분리 보존 후 만료 파기	finance owner

data class	trigger	duration	exception	action	owner
보안 audit	event 발생	risk 기준 기간	사고 조사 hold	archive 만료·media sanitization	security owner

30초 확인 2

`마지막 활동 후 365일` 규칙에서 마지막 활동이 새로 발생하면 기존 만료일을 바꿉니까? 바꾼다면 누가 어떤 event로 계산하고 이전 결정을 어떻게 기록합니까?

4. active·archive·hold·destroy는 다른 상태입니다

그림 4. 상태는 저장 위치보다 허용된 행동으로 정의합니다. archive와 hold는 계속 마음대로 사용할 수 있다는 뜻이 아닙니다.


수집
active
archive
목적·동의
서비스 사용
제한 조회
업무 사용 허용
접근 제한
상태 전환 증거 · trigger 시각 · 정책 vers

sion · 승인자 · 대상 count · 실패·재시도

상태	허용할 수 있는 행동	제한해야 할 행동	exit condition
active	원래 목적의 읽기·변경	목적 밖 재사용	목적 달성·trigger 발생
archive	정해진 증빙·조회	일반 서비스·마케팅 사용	보존 만료
hold	보존·무결성 보호	예정 파기·목적 밖 이용	hold 해제·재평가
destroy pending	파기 job·검증	업무 접근·새 파생 copy	모든 대상 처리·검증
destroyed	최소 증거 조회	원데이터 복원·사용	증거 자체 만료 정책

archive는 보존 기간을 무한 연장하지 않습니다

archive로 옮기면 비용과 접근 경로가 달라질 뿐 데이터 필요성이 자동으로 생기지 않습니다. archive에도 trigger·duration·access role·검색 사유·파기 방법이 필요합니다.

anonymization은 삭제와 같은가

재식별이 합리적으로 불가능하도록 처리되었는지는 데이터, 다른 결합 가능 정보, 조직의 접근 수단에 따라 달라집니다. 단순 hash·ID 치환·열 삭제를 자동으로 익명화라고 부르지 않습니다. 이 교재의 시뮬레이터에서는 `anonymize`를 무조건 완료로 판정하지 않고 조직의 별도 검증·승인이 있는 action으로 기록하도록 합니다.

5. replica·backup·archive를 구분합니다

그림 5. replica는 현재 서비스 가용성, backup은 과거 상태 복구, archive는 장기 보존·제한 조회에 중심을 둡니다.

REPLICA

replica

지금 읽을 수 있나

변경 빠르게 복제
오삭제도 전파 가능
가용성 목적

BACKUP

backup

과거로 복원할 수 있나

복구 지점 보존
독립성·무결성 필요
복원 시험 필수

replica ≠ backup · backup ≠ archive

각 copy의 목적·쓰기 가능성·접근 주

ARCHIVE

archive

업무상 오래 남겨야 하나

접근·사용 제한
검색성·법적 근거
보존 종료 뒤 파기

· archive ≠ 계속 사용 가능한 운영 DB

주체·만료·복원 절차를 따로 적는다.

비교	replica	backup	archive
대표 질문	지금 읽고 장애 전환할 수 있나	사고 전 상태로 돌아갈 수 있나	정해진 근거로 오래 보존할 수 있나
변화 속도	빠르게 동기·비동기 복제	주기·연속 기록	정책에 따라 batch 이동
오삭제 영향	그대로 전파될 수 있음	이전 지점이 남을 수 있음	쓰기 제한 시 별도 영향
쓰기 가능성	보통 시스템이 계속 갱신	일반 업무 쓰기 금지	제한·append-only 등 정책
완료 증거	lag·failover	integrity + restore drill	접근·보존·파기 audit

replica가 backup이 아닌 이유

사용자 실수로 `DELETE`를 실행하거나 application bug가 잘못된 값을 쓰면 변경이 replica에도 복제될 수 있습니다. 같은 관리자 credential, 같은 cloud account, 같은 network 경로를 공유하면 공격·오작동의 실패 영역도 공유합니다.

backup이 archive가 아닌 이유

backup은 전체 복구를 위해 형식과 chain을 유지하므로 개별 record를 검색·제출하기 어려울 수 있습니다. 반대로 archive는 업무·법적 조회를 위해 특정 record를 찾을 수 있지만, database 전체와 configuration을 장애 전 상태로 복원하지 못할 수 있습니다.

6. RPO·RTO를 서비스 언어로 바꿉니다

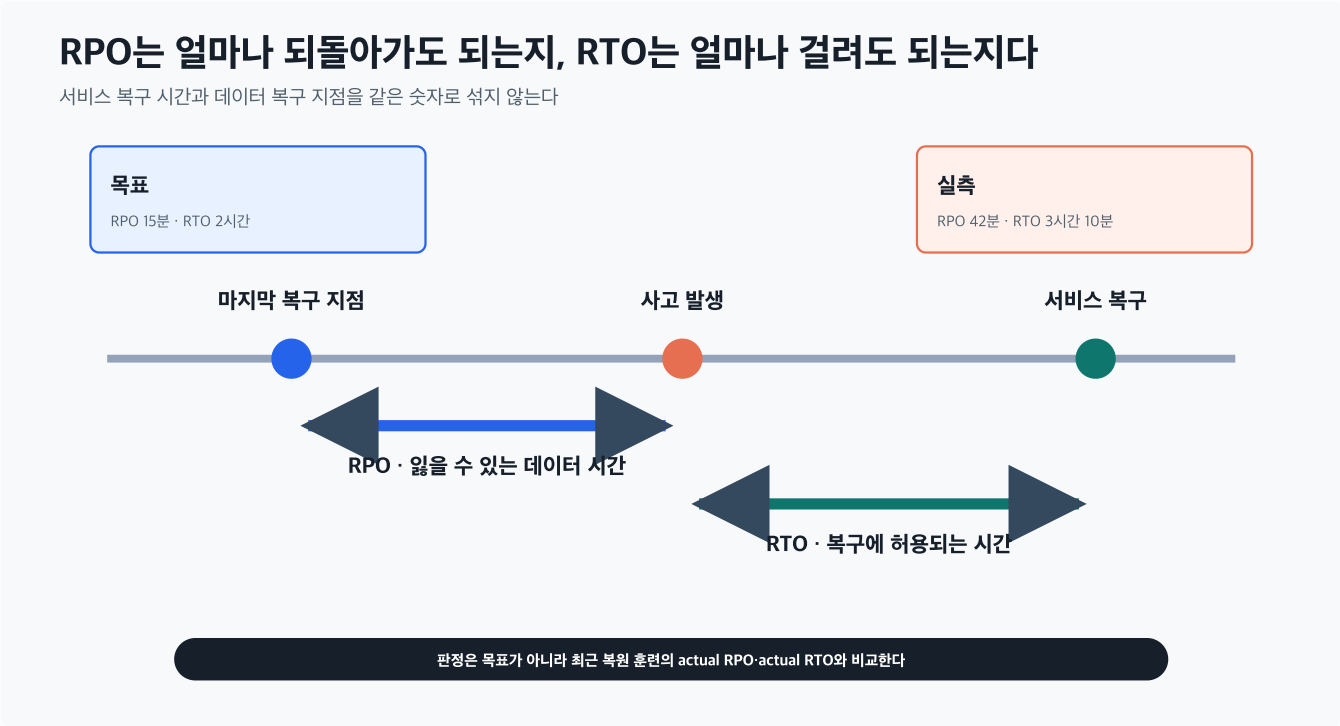


그림 6. RPO는 사고 전 어느 시점까지의 데이터 손실을 허용하는지, RTO는 사고 뒤 서비스 복구까지 허용하는 시간을 나타냅니다.

목표

RPO 15분 · RTO 2시간

실측

RPO 42분 · RTO 3시간 10분

actual RPO·actual RTO와 비교한다

NIST RPO 용어 정의는 중단 후 데이터를 복구해야 하는 과거 시점을 가리킵니다. NIST RTO 용어 정의는 업무·임무에 받아들이기 어려운 영향이 생기기 전의 전체 복구 시간을 설명합니다.

6.1. 목표를 업무 영향으로 정합니다

```
RPO 15분
= 사고 직전 최대 15분의 제출 답안이 사라질 수 있음을 업무가 수용한다.

RTO 2시간
= 사고 인지 뒤 2시간 이내 핵심 학습 서비스를 합의한 수준으로 열어야 한다.
```

숫자는 기술팀이 임의로 정하지 않습니다. 잃은 주문·답안을 재입력할 수 있는지, 규제·계약 영향, 사용자 피해, 수동 대체 절차, peak 시간대를 함께 평가합니다.

6.2. 목표와 실측을 분리합니다

항목	target	drill actual	gap	판정
RPO	15분	42분	+27분	실패
data restore	60분	75분	+15분	개선 필요
application ready	120분	190분	+70분	RTO 실패
security validation	RTO 안	25분	포함 여부 확인	계약 보완

RTO 안에 database 파일 복사만 포함할지, DNS·secret·권한·queue·search rebuild·application smoke·보안 승인까지 포함할지 경계를 명시합니다.

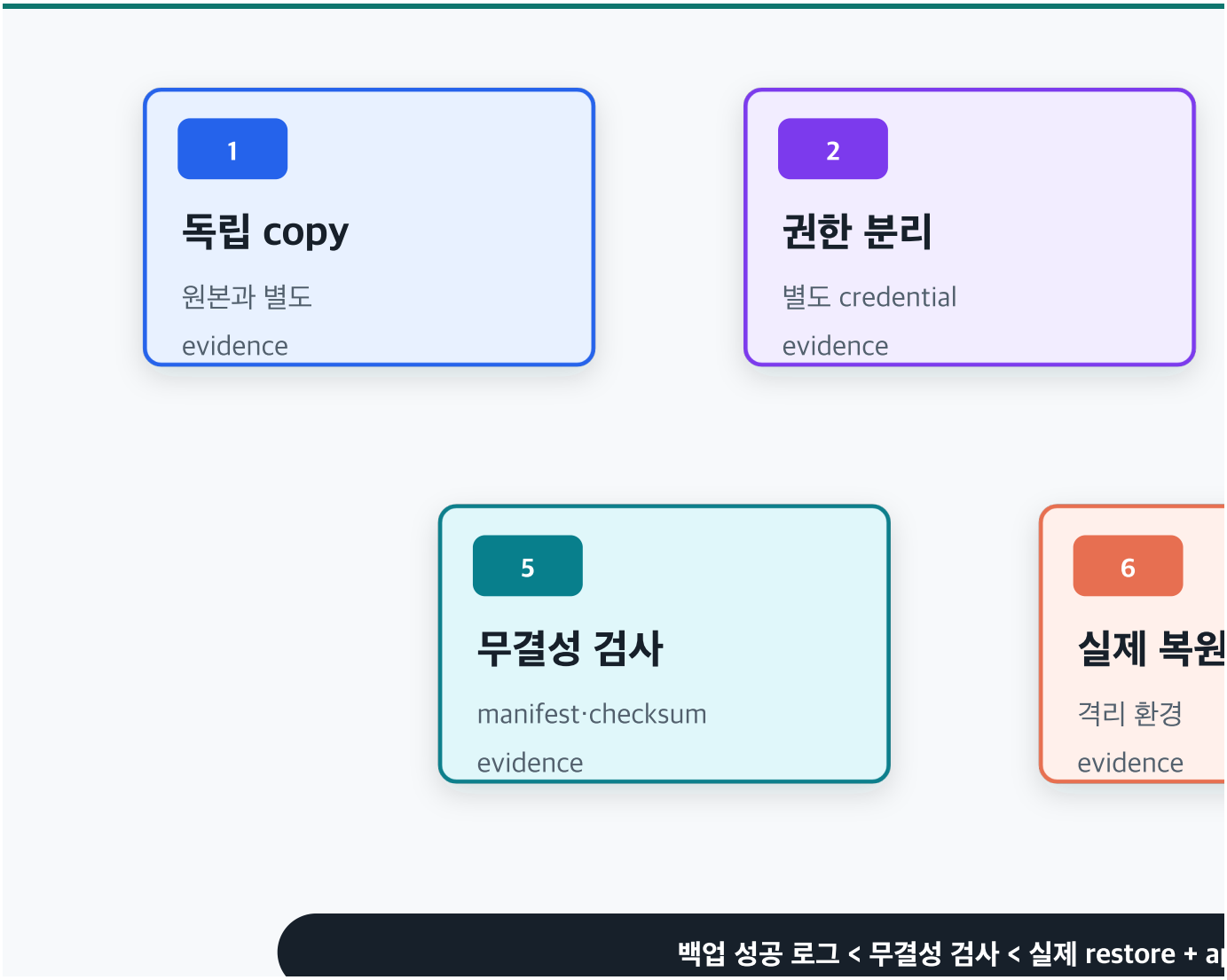
6.3. 데이터 class마다 다를 수 있습니다

회원 인증은 RTO가 짧고, 오래된 학습 report archive는 더 길 수 있습니다. 동일 DB에 같이 있어도 중요도별 복구 순서와 partial service 전략을 세울 수 있습니다. 다만 PostgreSQL PITR은 기본적으로 cluster 전체를 특정 시점으로 복구하는 방식이므로, record나 table 하나만 선택적으로 되돌리는 것과 구분합니다.

30초 확인 3
백업이 매일 00시에 한 번 성공하고 사고가 23:50에 발생했습니다. 다른 연속 기록이 없다면 최악의 실제 RPO는 몇 분에 가깝습니까?

7. 좋은 backup은 독립성과 복원 가능성으로 평가합니다

그림 7. copy 수만 세지 말고 같은 실패에 함께 무너질 수 있는지와 실제 복원했는지를 확인합니다.


1
독립 copy
원본과 별도
evidence
2
권한 분리
별도 credential
evidence
5
무결성 검사
manifest·checksum
evidence
6
실제 복원
격리 환경
evidence
백업 성공 로그 < 무결성 검사 < 실제 restore + a

3

offline·immutable

위험에 맞게

evidence

4

암호화·key

복구 가능

evidence

7

의존성 보존

base·WAL·parent

evidence

pplication smoke test + measured RPO/RTO

일곱 평가 축

1. **독립 copy:** 원본 손상과 동시에 사라지지 않는 위치입니까?
2. **권한 분리:** production admin이 backup도 즉시 지울 수 있습니까?
3. **offline·immutable:** threat model에 맞는 접근 불가능·변경 방지 copy가 있습니까?
4. **암호화·key:** 저장·전송 중 보호되고 key·권한도 복구할 수 있습니까?
5. **무결성 검사:** manifest·file list·size·checksum·필요 WAL을 확인합니까?
6. **실제 restore:** 격리 환경에서 database를 열고 업무 데이터를 검사합니까?
7. **의존성 보존:** base·incremental parent·WAL·configuration·extension·key가 같이 남습니까?

CISA #StopRansomware Guide는 중요한 데이터의 offline·encrypted backup과 정기적인 가용성·무결성 시험을 권고합니다. 연결 가능한 backup도 ransomware의 표적이 될 수 있습니다. immutable copy는 도움이 될 수 있지만 잘못된 권한·보존·비용·법적 요구를 자동 해결하지는 않으므로 threat model과 조직 환경에 맞춰 선택합니다.

backup job 성공의 증거 수준

수준	증거	아직 모르는 것
1	scheduler가 exit 0	파일 내용·누락·key
2	object·size 존재	corruption·필요 chain
3	manifest·checksum 통과	server 기동·업무 의미
4	isolated restore 성공	application·권한·사용자 흐름
5	app smoke·invariant·RPO/RTO 통과	production 전환·사람·절차
6	전환·rollback·재삭제까지 훈련	다음 version·새 dependency

핵심 판정 backup success log는 출발점입니다.

복구 가능성의 가장 가까운 증거는 실제 restore와 업무 검증입니다.

8. PostgreSQL의 세 복구 방법을 구분합니다

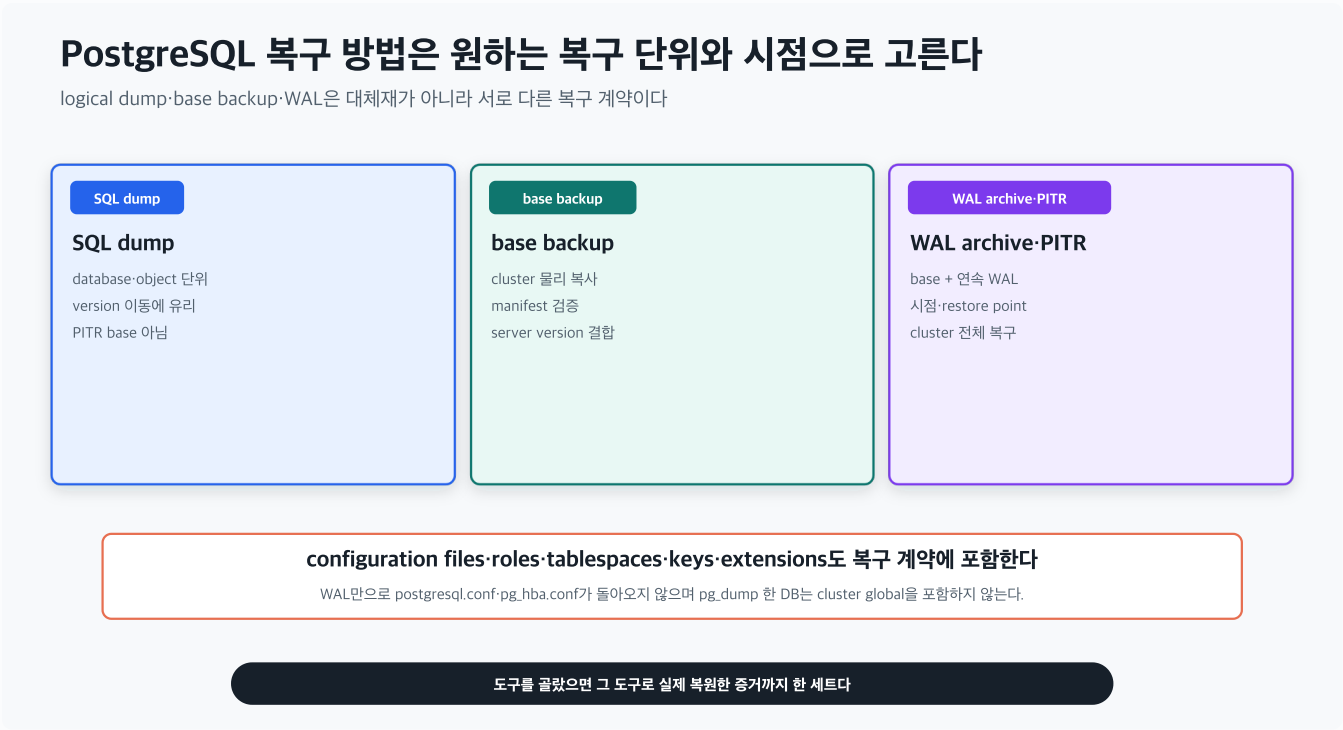


그림 8. logical dump, physical base backup, continuous WAL archive는 복구 단위·호환성·복구 시점이 다릅니다.

SQL dump

SQL dump

database·object 단위

version 이동에 유리

PITR base 아님

base backup

base backup

cluster 물리 복사

manifest 검증

server version 결합

configuration files·roles·tablespaces·l

WAL만으로 postgresql.conf·pg_hba.conf가 돌아오지 않

WAL archive·PITR

WAL archive·PITR

base + 연속 WAL

시점·restore point

cluster 전체 복구

keys·extensions도 복구 계약에 포함한다

으며 pg_dump 한 DB는 cluster global을 포함하지 않는다.

PostgreSQL 18 Backup and Restore는 SQL dump, file-system level backup, continuous archiving을 설명합니다.

8.1. SQL dump · 논리적 백업

`pg_dump`는 SQL 명령 또는 archive 형식으로 database를 재구성합니다. dump는 시작 시점의 일관된 snapshot을 만들고, 일반적으로 새 PostgreSQL version이나 다른 machine architecture로 옮길 때 유리합니다. custom·directory 형식은 `pg_restore`로 선택·병렬 복원할 수 있습니다.

```
pg_dump -Fc appdb > appdb.dump
createdb -T template0 restore_lab
pg_restore --exit-on-error -d restore_lab appdb.dump
```

주의할 점:

- `pg_dump`는 한 database 단위이며 cluster-wide role·tablespace 정보를 포함하지 않습니다.
- `pg_dumpall --globals-only` 같은 별도 global 백업 계약이 필요할 수 있습니다.
- 여러 database를 `pg_dumpall`로 처리해도 database별 snapshot 시점은 동기화되지 않습니다.
- logical dump는 WAL replay의 base backup으로 사용할 수 없습니다.
- text dump는 오류 뒤 일부만 복원될 수 있으므로 `ON_ERROR_STOP`·single transaction 정책을 검토합니다.

8.2. base backup · 물리적 cluster 복사

`pg_basebackup`은 running PostgreSQL cluster의 base backup을 만듭니다. `backup_manifest`를 생성해 file·checksum·필요 WAL 검증에 사용할 수 있습니다. 물리 backup은 server version·architecture·configuration 결합을 확인합니다.

```
pg_basebackup -h db.example -D /isolated/backup -Fp -X stream
pg_verifybackup /isolated/backup
```

pg_verifybackup 공식 문서는 manifest, file 존재·size, checksum, 필요한 WAL의 읽기·parse를 확인하지만 running server가 수행할 모든 검사를 대신하지 못한다고 명시합니다. 도구가 통과해도 test restore와 실제 database 내용 검증을 수행해야 합니다.

8.3. WAL archive·PITR · 시점 복구

Continuous Archiving and Point-in-Time Recovery는 base backup과 이후 WAL sequence를 보존해 base 시점부터 연속적인 시점으로 복구하는 방식을 설명합니다.

```
restore target = 2026-07-15 14:31:00+09
필요 조건 = 적절한 base backup + 그 이후 target까지 끊김 없는 WAL
```

PITR은 timestamp, named restore point 등으로 목표를 정할 수 있고 recovery timeline을 통해 복구 분기를 관리합니다. 그러나 다음을 놓치지 않습니다.

- 원하는 일부 table만이 아니라 PostgreSQL cluster 전체 시점 복구가 중심입니다.
- WAL archive만으로 `postgresql.conf`, `pg_hba.conf`, `pg_ident.conf`가 복원되지 않습니다.
- configuration, extension, OS dependency, secret, certificate, DNS, application version을 별도 계약으로 관리합니다.
- 너무 짧은 `archive_timeout`은 archive 파일 수와 저장 비용을 크게 늘릴 수 있습니다.
- 목표 시점에서 사용자 연결을 열기 전에 데이터·권한·삭제 suppression을 검사합니다.

9. incremental backup은 의존성 chain입니다

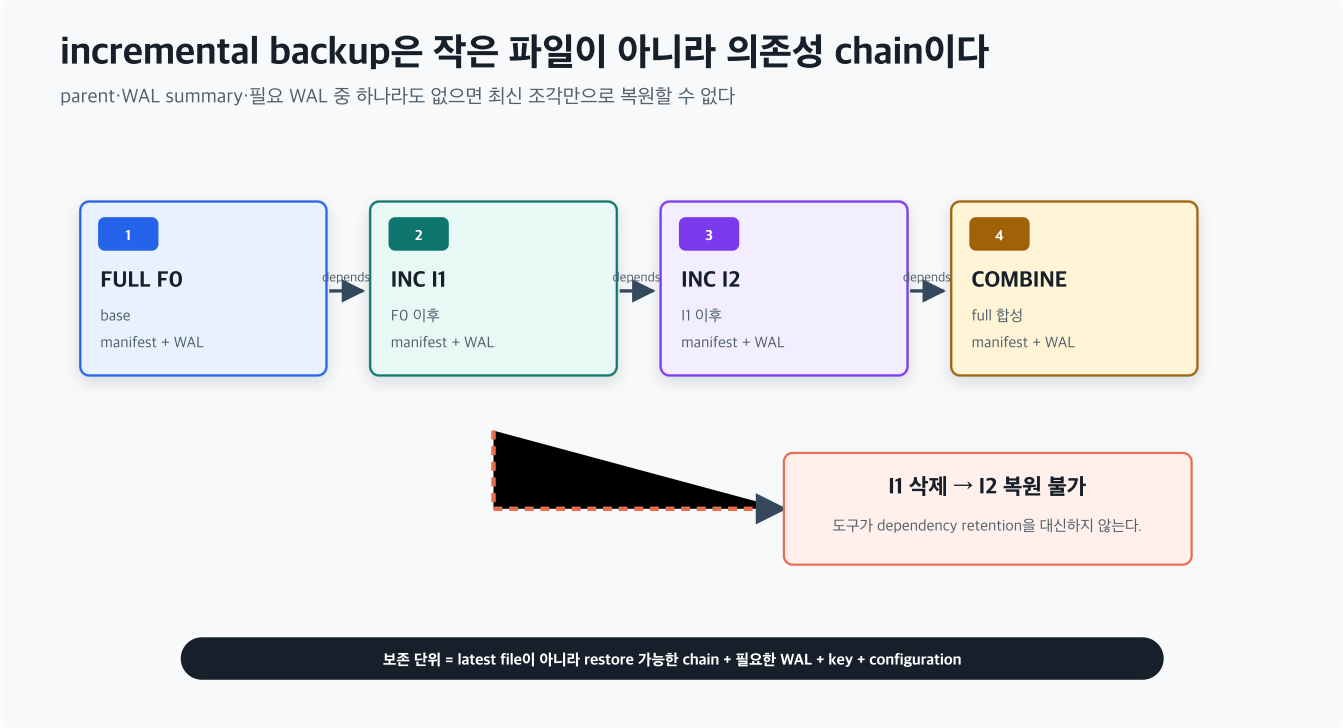
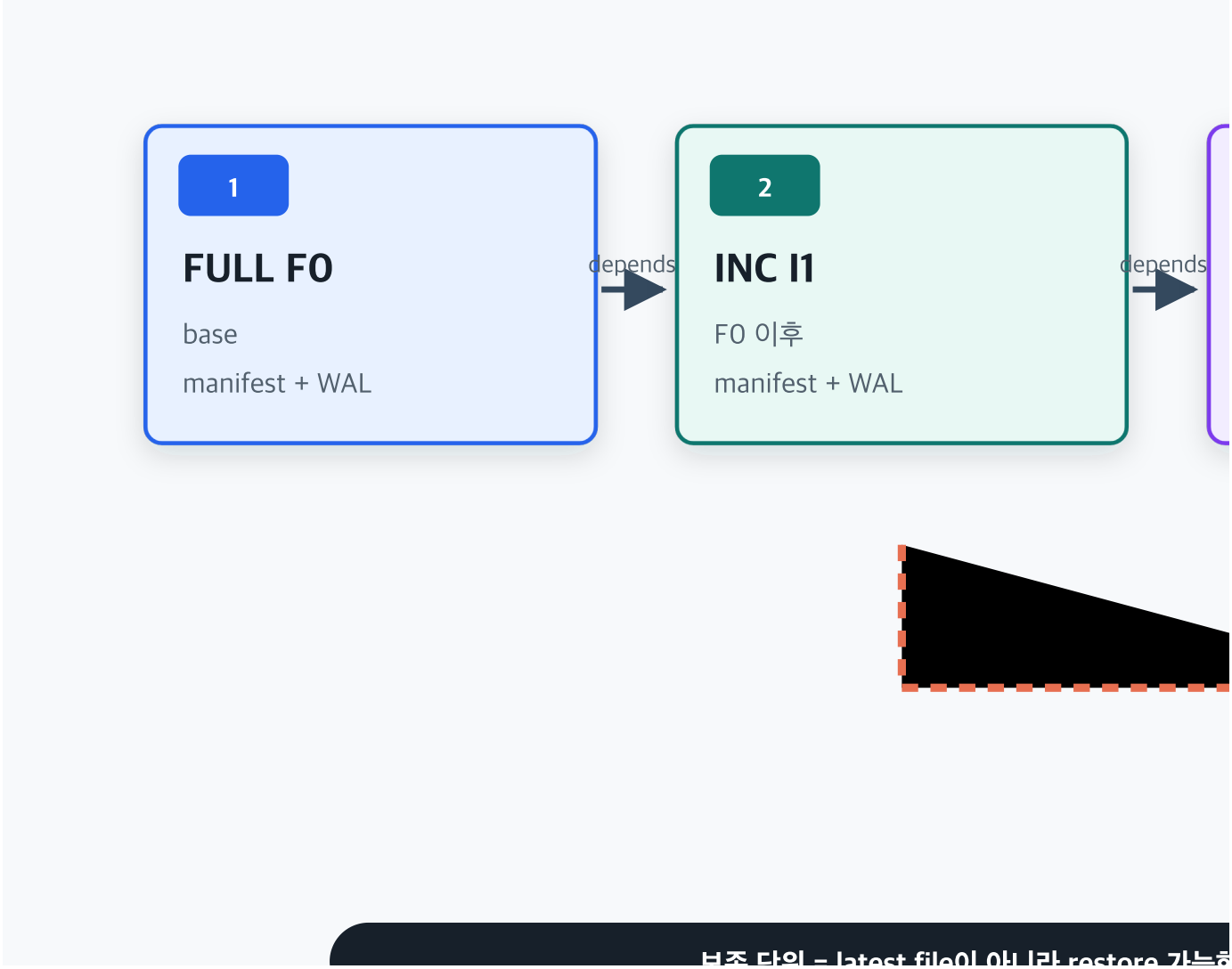
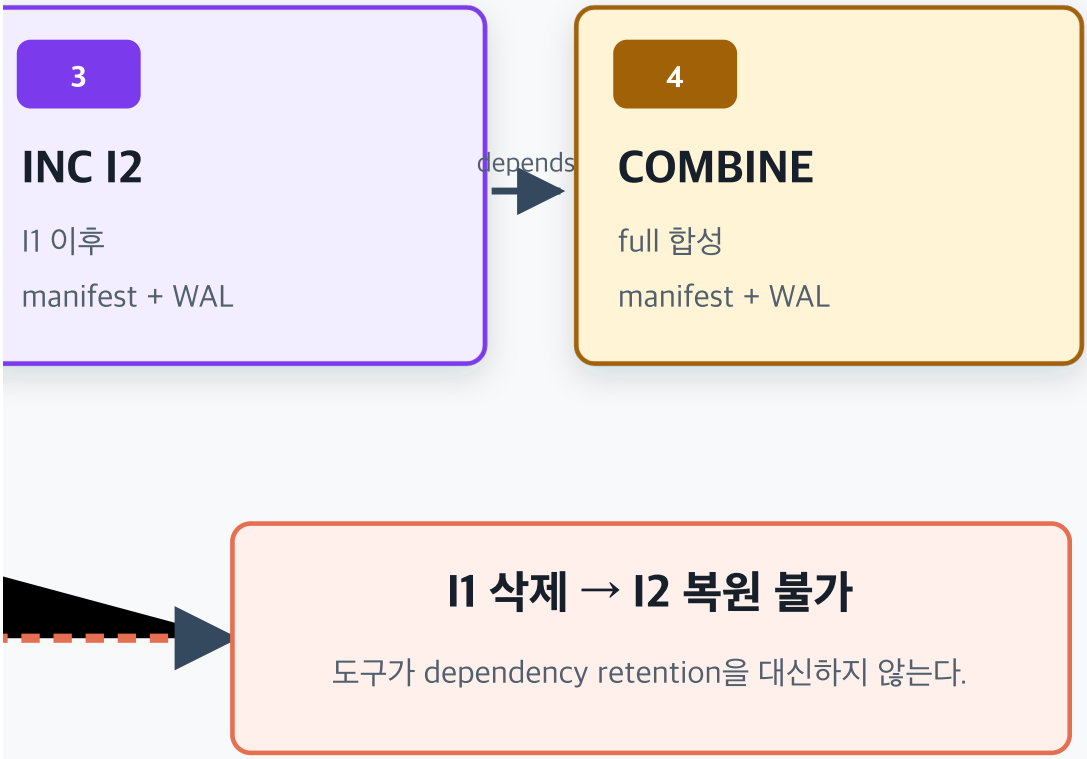


그림 9. 최신 incremental 파일 하나가 아니라 parent chain·WAL summary·필요 WAL·manifest가 함께 있어야 복원 단위가 됩니다.


1
FULL F0
base
manifest + WAL
depends
2
INC I1
F0 이후
manifest + WAL
depends


3
INC I2
I1 이후
manifest + WAL
depends
4
COMBINE
full 합성
manifest + WAL
I1 삭제 → I2 복원 불가
도구가 dependency retention을 대신하지 않는다.

PostgreSQL 18의 `pg_basebackup --incremental`은 선행 backup의 manifest와 WAL summary를 이용합니다. 복원 시에는 선행 full·incremental chain과 관련 WAL을 보존하고 `pg_combinebackup`으로 full backup을 합성할 수 있습니다.

운영자가 직접 관리해야 하는 것

dependency	누락 시 영향	보존 증거
full parent F0	모든 child의 기준 소실	immutable ID·manifest
incremental I1·I2	이후 change 재구성 불가	parent pointer·checksum
WAL summary	incremental 생성 조건 실패	server version·range
recovery WAL	목표 시점 재생 불가	first·last segment·gap check
encryption key	파일은 있어도 해독 불가	key ID·recovery test
configuration	server·접근 정책 불일치	versioned config backup

PostgreSQL은 운영자가 어떤 backup이 다른 incremental backup의 parent인지에 맞춰 retention을 자동 관리해 주지 않습니다. “30일보다 오래된 파일 삭제” 같은 object age 규칙이 아직 필요한 parent를 지우지 않는지 dependency graph로 검증합니다.

restore set을 하나의 object처럼 다룹니다

```
restore_set_id = rs_2026_07_15_1430
contains = F0 + I1 + I2 + required WAL + config + key reference
target = 2026-07-15 14:30 KST
verified = manifest PASS
restored = drill_2026_07_15 PASS
expires = 마지막 child가 의존하지 않는 시점 + policy
```

10. 복원 훈련은 격리·검증·폐기까지 이어집니다

그림 10. restore 명령이 끝난 시각과 서비스가 안전하게 준비된 시각은 다릅니다. 훈련 환경 자체도 민감한 운영 copy이므로 마지막에 안전하게 파기합니다.

복원 완료 조건

manifest 통과 + 목표 시점 확인 + 핵심 invariant + 권한·secret 교체 + app

+ 삭제된 개인정보의 suppression·재삭제 + 훈련 환경의 안전한 파기

› smoke + actual RPO/RTO

10.1. drill 시작 전

항목	결정
incident	실수로 회원 1,000명을 삭제한 14:31 사건
target	삭제 직전 14:30:59
scope	PostgreSQL cluster + object attachments + search rebuild
target RPO·RTO	15분 · 120분
isolated environment	production route·queue·email 차단
test data handling	실데이터 접근 승인·격리·drill 후 파기
stop rule	WAL gap, key mismatch, unexpected outbound traffic

10.2. restore 뒤 데이터 검증

```
-- schema version
SELECT version FROM schema_migrations ORDER BY version DESC LIMIT 1;

-- 핵심 row count와 기간
SELECT COUNT(*), MIN(created_at), MAX(created_at) FROM member;

-- 업무 invariant 예시
SELECT COUNT(*) AS orphan_count
FROM enrollment e
LEFT JOIN member m ON m.member_id = e.member_id
WHERE m.member_id IS NULL;
```

검증은 row count 하나로 끝내지 않습니다.

- schema·extension·collation·timezone
- 핵심 table count·min/max timestamp·sample
- PK·FK·UNIQUE·업무 invariant
- role·grant·row security·secret rotation
- queue·scheduler·webhook·email이 격리되어 있는지
- application login·조회·쓰기의 제한된 smoke test
- 목표 시점과 actual recovery point

- incident start부터 application-ready까지 actual RTO

10.3. production 연결 전 삭제 suppression

과거 backup에는 이미 삭제된 개인정보가 들어 있을 수 있습니다. restore 직후 외부 서비스·사용자 연결을 열기 전에 삭제 ledger를 적용하고 search·analytics·vendor로 다시 퍼지지 않게 합니다.

10.4. drill 환경 파기

훈련 database, log, screenshot, export, temporary key, operator local file을 inventory에 넣고 파기합니다. 훈련을 위해 만든 copy가 장기 방치되면 복구 연습이 새로운 위험을 만듭니다.

30초 확인 4
`pg_verifybackup PASS`, PostgreSQL 기동 성공, application 로그인 성공 가운데 어느 하나가 나머지 둘을 자동으로 증명합니까? 왜 세 단계가 모두 필요합니까?

11. 삭제 요청은 copy sweep입니다

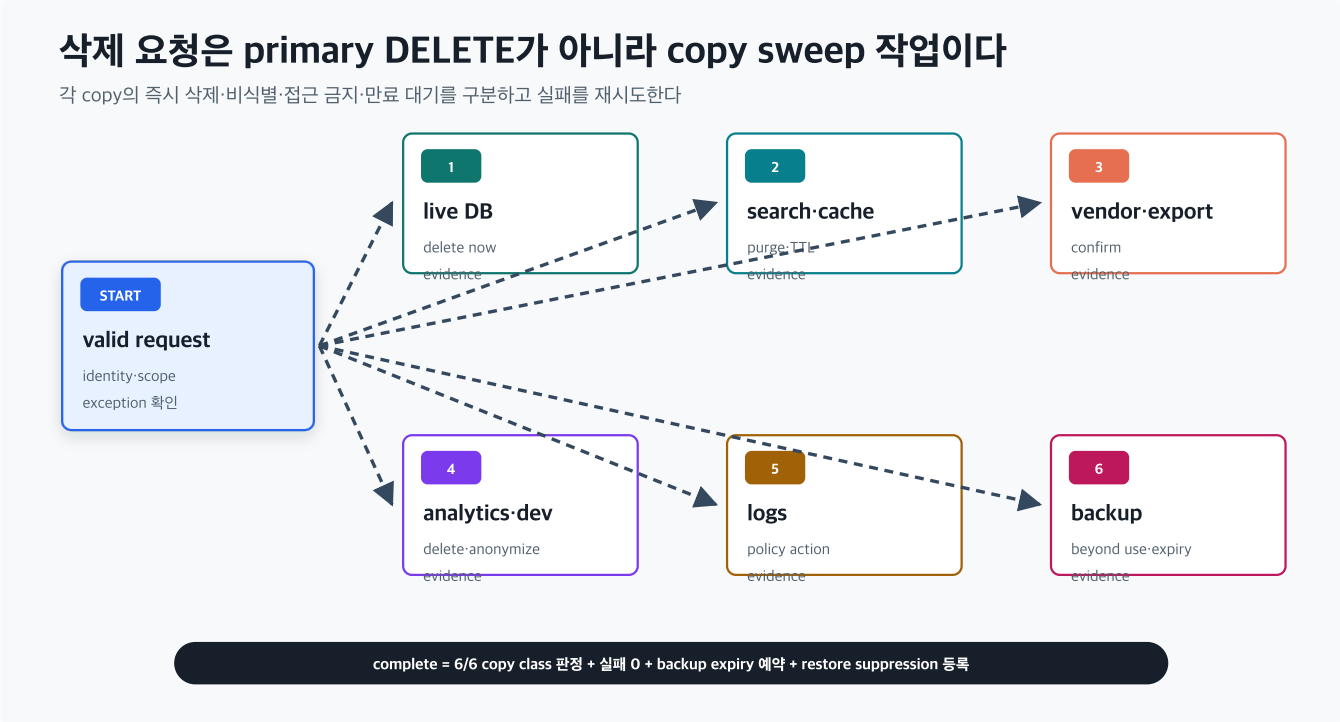
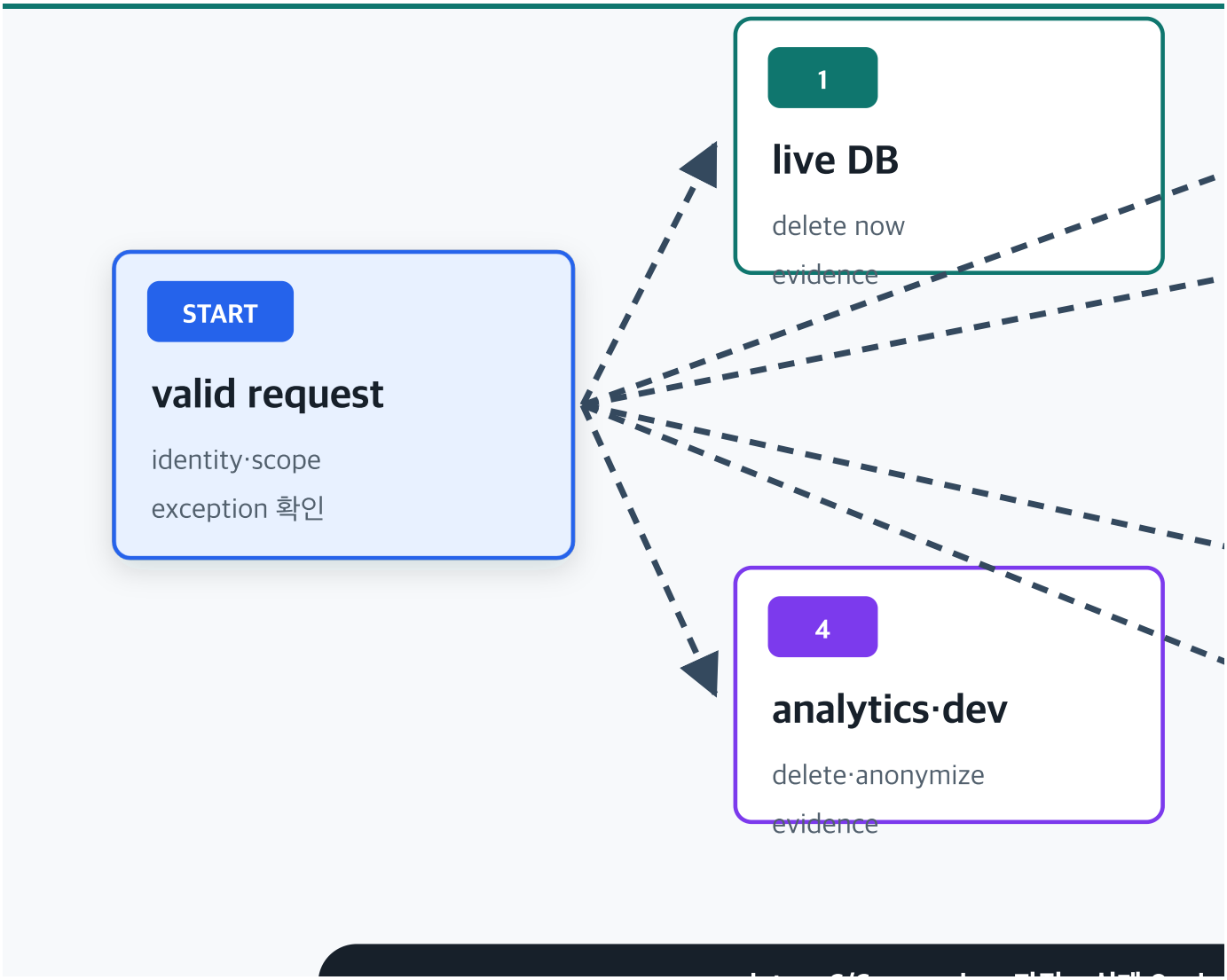
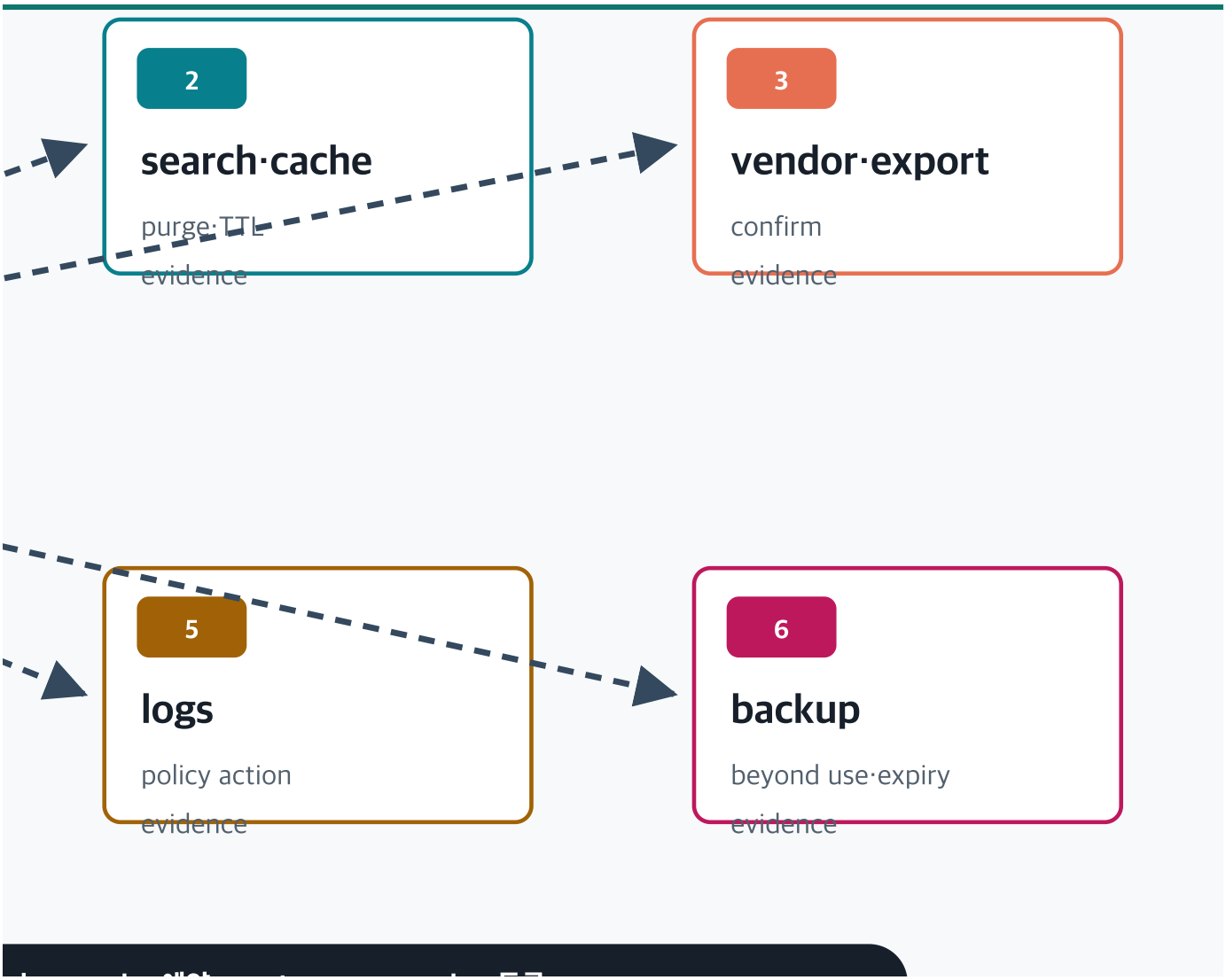


그림 11. 유효한 요청·근거를 확인한 뒤 copy class마다 `삭제·비식별·제한·만료·예외`를 판정하고 실패를 재시도합니다.


START
valid request
identity·scope
exception 확인
1
live DB
delete now
evidence
4
analytics·dev
delete·anonymize
evidence


2
search·cache
purge·TTL
evidence
3
vendor·export
confirm
evidence
5
logs
policy action
evidence
6
backup
beyond use·expiry
evidence

11.1. 요청 접수와 identity·scope

1. 요청자의 identity와 권한을 확인합니다.
2. 대상 subject·account·tenant·기간을 명확히 합니다.
3. 삭제·처리정지·정정 등 요청 유형을 구분합니다.
4. 유효한 보존 의무·분쟁 hold의 범위와 충돌을 확인합니다.
5. 대상 copy map과 processor를 펼칩니다.

11.2. copy별 결과 상태

상태	의미	다음 행동
deleted	target data가 제거됨	count·verification 저장
anonymized	승인된 방법으로 재식별 위험을 낮춤	validation·재평가 근거
restricted	사용·접근 금지	만료·hold review 추적
expiry scheduled	granular deletion 곤란한 backup	만료 job·restore 재삭제
exception hold	유효한 근거로 파기 일시정지	범위·owner·review date
failed	기술·vendor 오류	재시도·escalation
missing	위치·subject mapping 불명	inventory 개선·조사

11.3. 삭제 job의 idempotency

같은 삭제 요청이 재시도되어도 이미 삭제된 상태를 오류로 만들지 않고 남은 copy만 닫을 수 있어야 합니다.

```
request_id = erasure_2841
subject_ref = protected(mem_102)
copy = search/member-v3
expected = 1 document
actual = 1 deleted
retry = safe
outcome = CLOSED
```

11.4. 로그에 원데이터를 다시 남기지 않습니다

삭제 증거에 전체 이메일·전화번호·답안을 복사하면 증거 저장소가 새 개인정보 원장이 됩니다. 요청 ID, 보호된 subject reference, 대상 위치, count, method, timestamp, policy version, 오류, 승인처럼 검증에 필요한 최소 정보를 기록합니다.

12. backup 만료와 restore 후 재삭제를 설계합니다

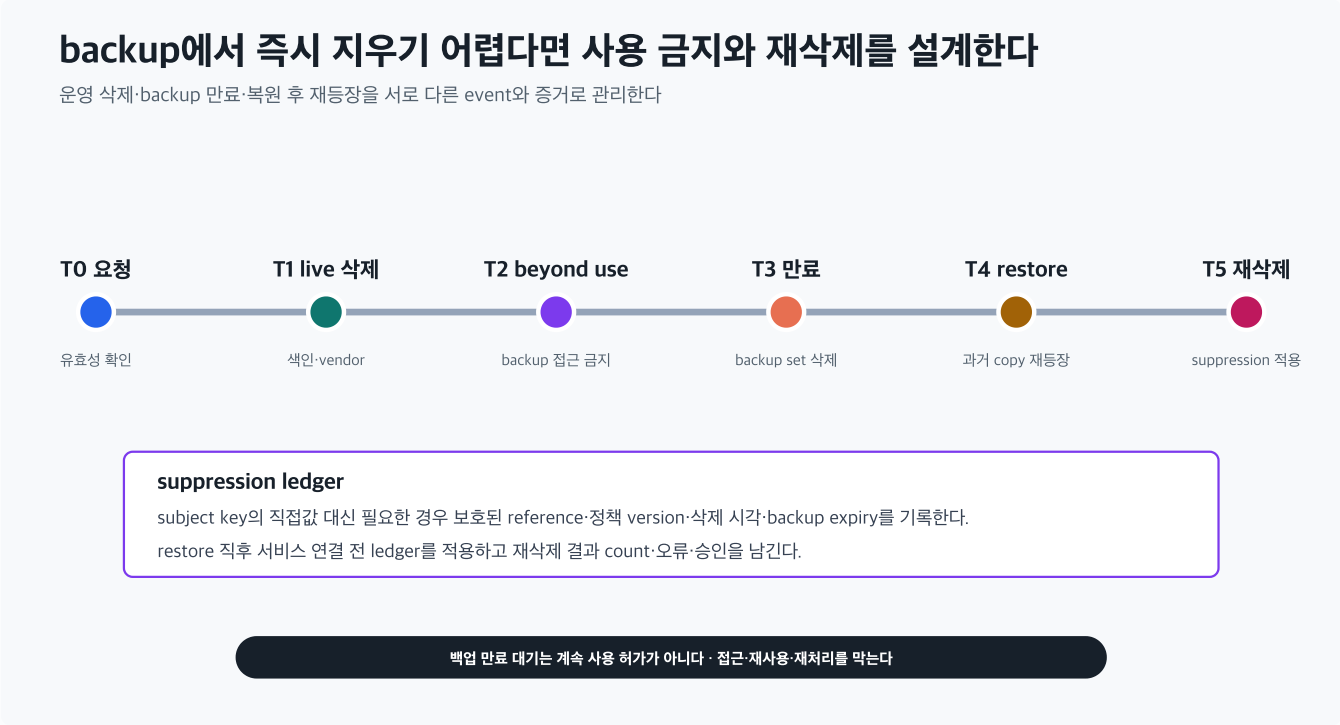
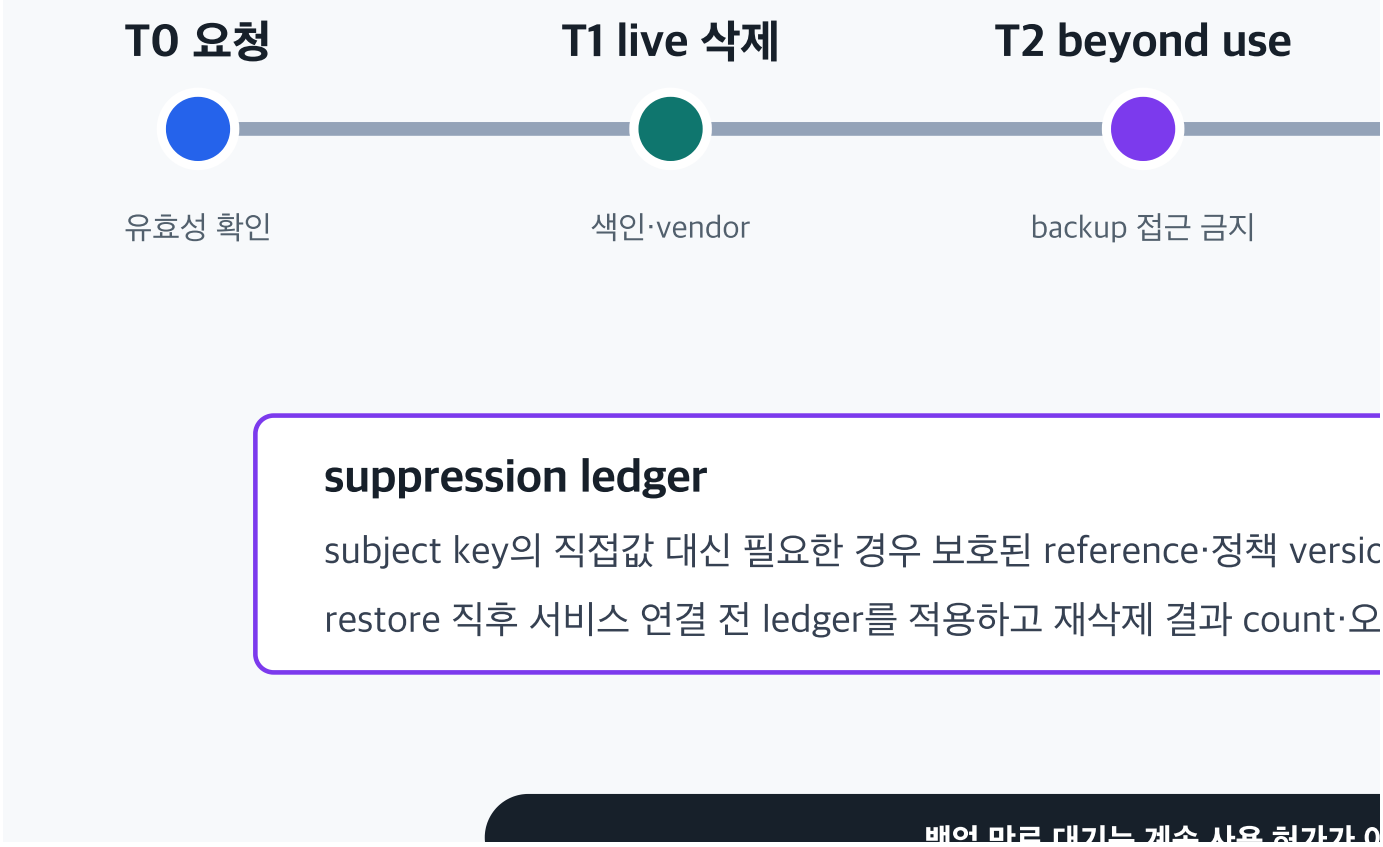


그림 12. 운영 삭제와 backup 만료는 다른 event입니다. backup에서 즉시 개별 삭제가 어렵다면 사용 금지·정해진 만료·restore 시 재삭제를 한 정책으로 묶습니다.


T0 요청
T1 live 삭제
T2 beyond use
유효성 확인
색인·vendor
backup 접근 금지
suppression ledger
subject key의 직접값 대신 필요한 경우 보호된 reference·정책 versic
restore 직후 서비스 연결 전 ledger를 적용하고 재삭제 결과 count·오

on·삭제 시각·backup expiry를 기록한다.

류·승인을 남긴다.

대한민국 개인정보 보호법 제21조는 보유기간 경과·목적 달성 등 개인정보가 불필요해졌을 때 지체 없이 파기하고, 다른 법령에 따라 보존해야 하면 다른 개인정보와 분리해 저장·관리하도록 규정합니다. 개인정보 보호법 시행령 제16조는 전자적 파일을 복원이 불가능한 방법으로 영구 삭제하고 기술적 특성상 현저히 곤란한 경우 법에서 정한 방식으로 복원이 불가능하도록 조치하는 기준을 둡니다. 실제 적용은 최신 법령·고시와 조직 상황을 개인정보·법률 담당자가 검토해야 합니다.

backup의 개별 삭제가 곤란할 때 운영 설계

이 교재는 특정 backup 제품·관할에서 항상 허용된다고 단정하지 않습니다. 적용 법률과 기술을 검토한 뒤 다음 통제를 문서화합니다.

1. live·search·analytics·vendor 등 일상 사용 copy는 가능한 범위에서 즉시 처리합니다.
2. backup 속 대상은 업무·분석·마케팅 등 다른 목적으로 사용하지 못하게 합니다.
3. backup set은 정해진 schedule로 만료시키고 임의 연장을 막습니다.
4. exception·hold가 있으면 범위·근거·review date를 기록합니다.
5. restore 시 suppression ledger를 적용해 삭제 대상이 서비스로 재등장하지 않게 합니다.
6. restore 후 재삭제 count·오류·승인과 원래 backup 만료를 증거로 남깁니다.

영국 규제기관의 ICO Right to Erasure guidance는 backup에서 즉시 덮어쓰지 못하는 경우 데이터를 사용 범위 밖에 두고 다른 목적으로 사용하지 않으며 정해진 schedule에 따라 만료시키는 운영 접근을 설명합니다. 이는 영국 GDPR 관할의 참고 예시이며 한국 법률의 대체 근거가 아닙니다.

suppression ledger의 역설

삭제를 기억하려면 최소 reference가 필요할 수 있습니다. 원 개인정보를 그대로 보존하지 않고 다음을 검토합니다.

항목	설계 질문
subject reference	재삭제에 충분하면서 원값 노출을 줄였는가
key·salt	누가 접근하고 rotation·복구는 어떻게 하는가
purpose	restore 재삭제 외 사용을 막는가
retention	모든 관련 backup 만료 뒤 언제 ledger를 지우는가
evidence	어떤 restore set에 적용했는지 증명 가능한가

13. legal hold는 범위가 있는 예외입니다

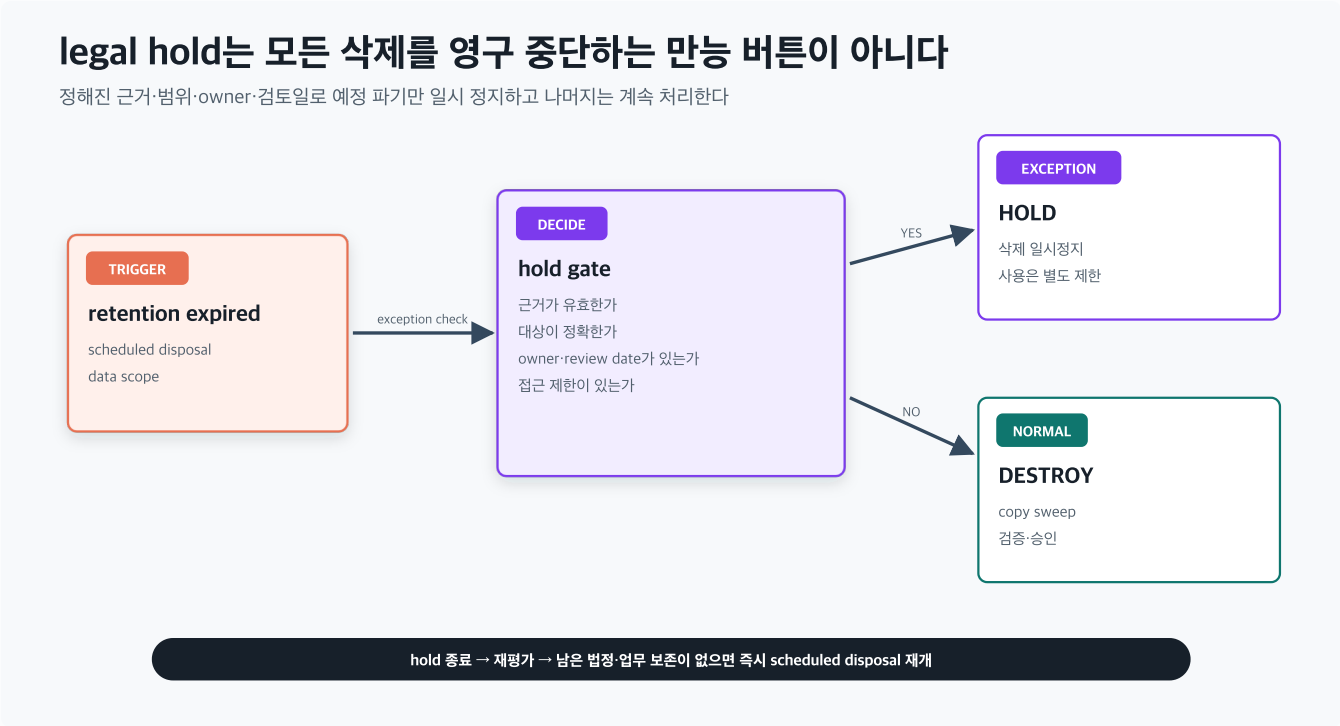
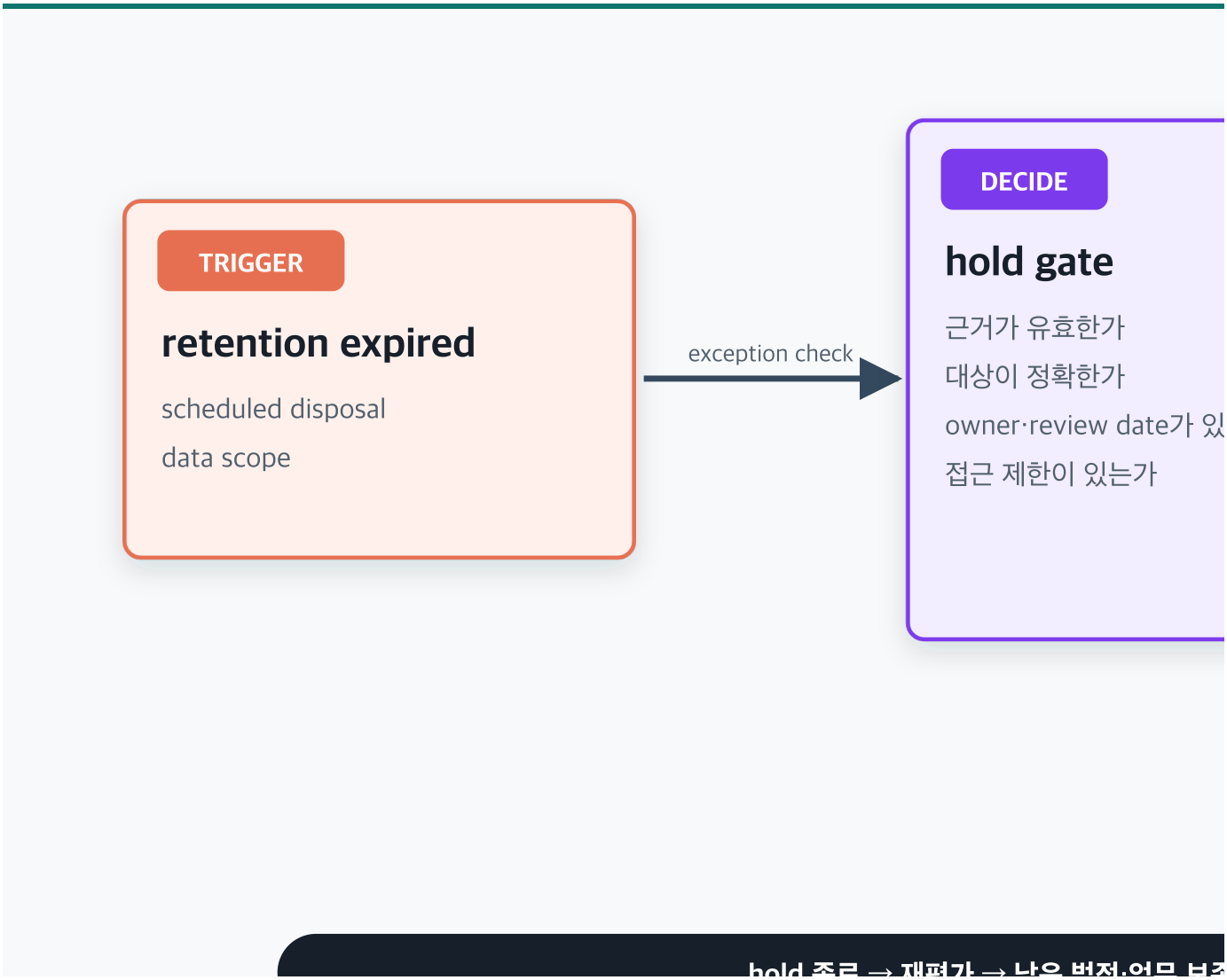
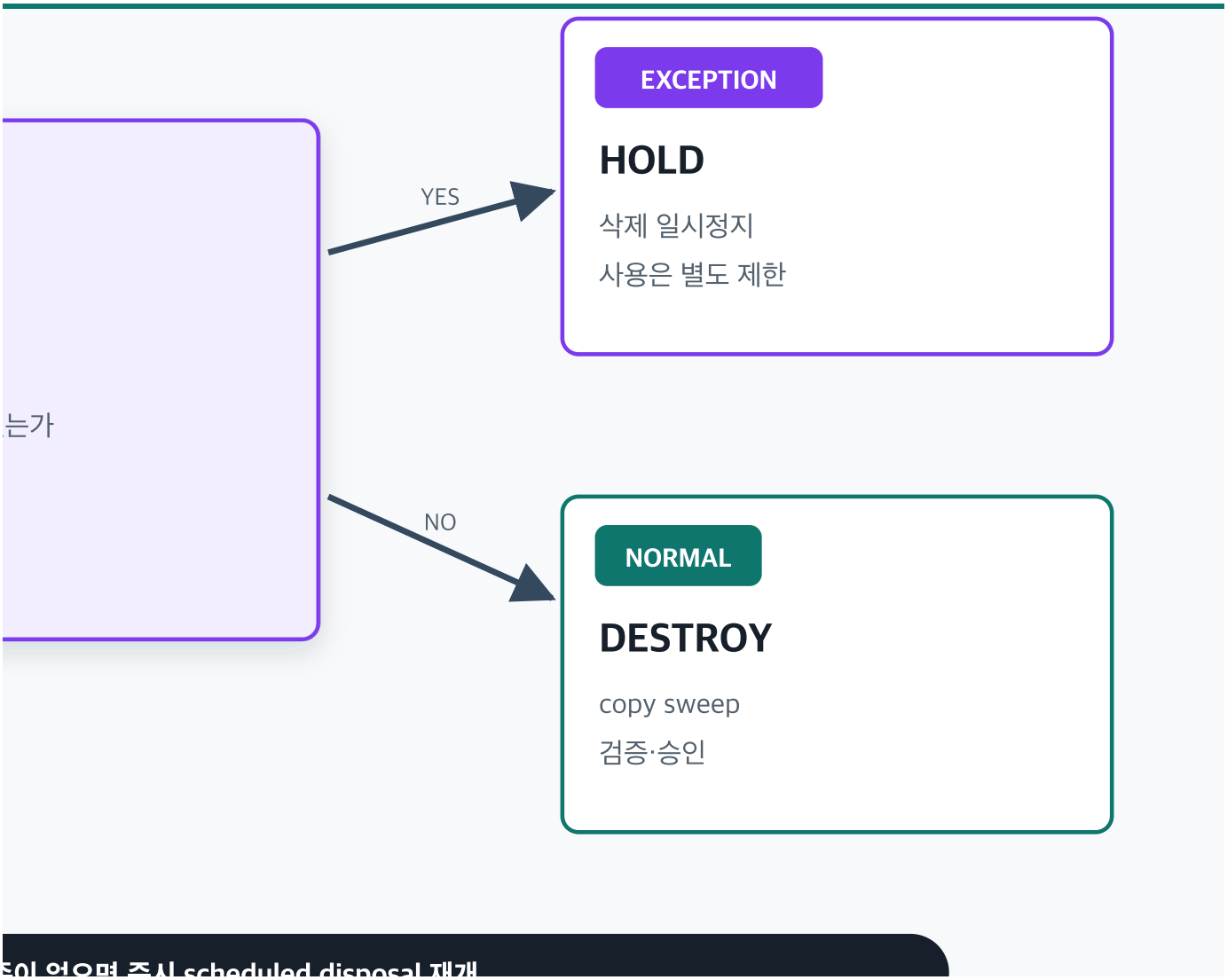


그림 13. hold는 예정 파기를 일시 정지하지만 원래 목적 밖 사용을 자동 허용하지 않습니다. 대상 밖 데이터의 정상 파기도 계속되어야 합니다.


TRIGGER
retention expired
scheduled disposal
data scope
exception check
DECIDE
hold gate
근거가 유효한가
대상이 정확한가
owner·review date가 있
접근 제한이 있는가


EXCEPTION
HOLD
삭제 일시정지
사용은 별도 제한
YES
는가
NO
NORMAL
DESTROY
copy sweep
검증·승인

hold register

열	예시
hold ID	`hold_dispute_2026_17`
authority·reason	분쟁 대응 요청·내부 승인 문서
scope	특정 member·payment·기간
systems	payment DB, finance export, vendor case
started_at	2026-06-10
owner·approver	legal owner·privacy owner
access restriction	named review role only
review_at	2026-08-10
release condition	분쟁 종결·승인
release action	남은 보존 근거 재평가 후 파기 재개

범위를 넓혀 잡지 않습니다

한 회원의 결제 분쟁 때문에 모든 회원의 모든 학습 기록을 hold하지 않습니다. subject, record type, 기간, system을 가능한 범위에서 구체화합니다. 기술적으로 세밀한 분리가 어렵다면 제한 사유·보완 통제·분리 계획을 기록합니다.

hold 해제는 삭제 완료가 아닙니다

hold가 끝나면 원래 retention clock과 다른 남은 의무를 다시 평가합니다. 이미 기간이 끝났고 다른 근거가 없다면 scheduled disposal을 재개합니다. 해제 event가 삭제 queue로 연결되지 않으면 hold copy가 영구 보존될 수 있습니다.

법률 판단 경계
이 교재는 제품·프로세스 설계를 위한 학습 자료이며 개별 사안의 법률 자문이 아닙니다. 어떤 의무가 우선하는지, 보존 범위·기간·파기 방식이 적절한지는 최신 법령, 관할, 계약, 사건 사실을 확인할 수 있는 담당자가 승인합니다.

14. 저장매체 파기는 방법·검증·승인을 나눕니다

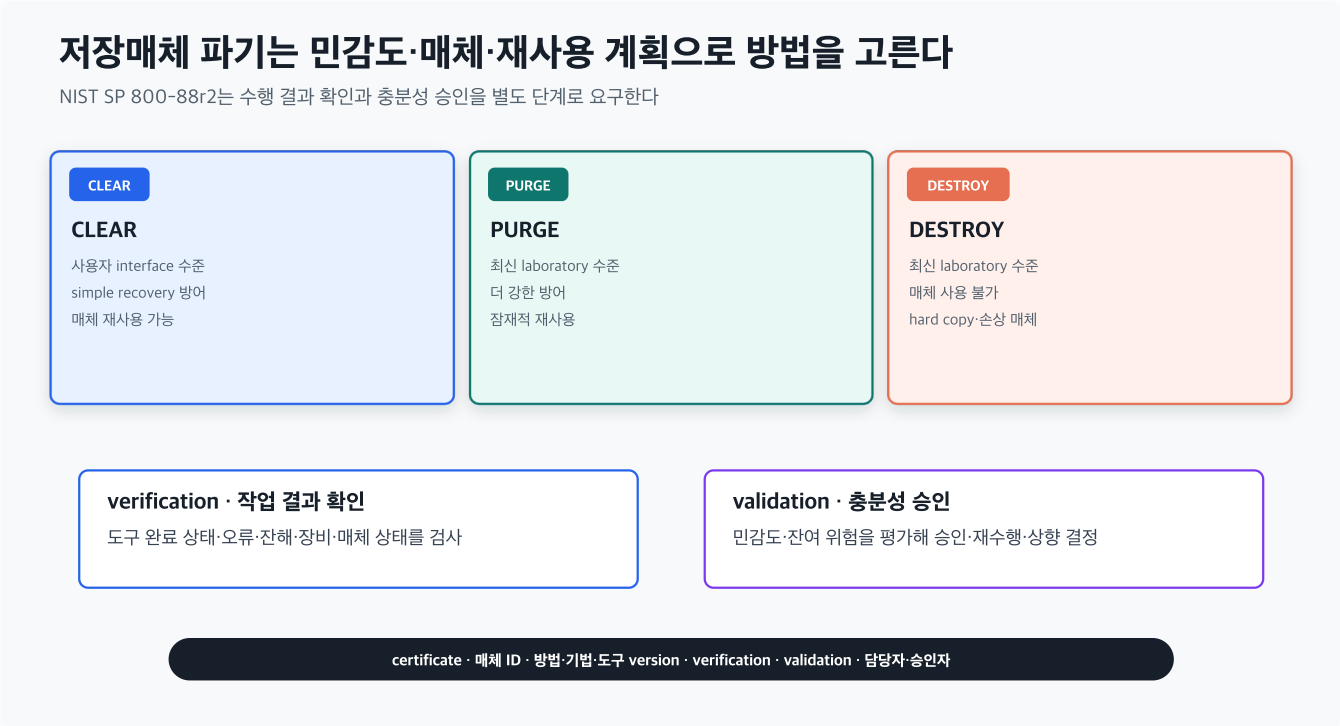


그림 14. NIST SP 800-88 Rev.2의 Clear·Purge·Destroy는 방어 수준과 재사용 가능성이 다릅니다. 수행 성공 확인과 충분성 승인을 별도로 기록합니다.

CLEAR

CLEAR

사용자 interface 수준

simple recovery 방어

매체 재사용 가능

PURGE

PURGE

최신 laboratory 수준

더 강한 방어

잠재적 재사용

verification · 작업 결과 확인

도구 완료 상태·오류·잔해·장비·매체 상태를 검사

certificate · 매체 ID · 방법·기법·도구 versio

DESTROY

DESTROY

최신 laboratory 수준

매체 사용 불가

hard copy·손상 매체

validation · 충분성 승인

민감도·잔여 위험을 평가해 승인·재수행·상향 결정

n · verification · validation · 담당자·승인자

NIST SP 800-88 Rev.2는 2025년 9월 발행되어 Rev.1을 대체했습니다. media sanitization을 정해진 노력 수준에서 target data 접근을 실행 불가능하게 만드는 과정으로 설명하고, 정보 민감도에 맞는 program·technique·control을 설계하도록 안내합니다.

14.1. 세 sanitization method

method	목표	media 재사용	학습자 해석
Clear	사용자에게 제공되는 같은 interface의 단순·비침습 복구를 방어	보통 가능	user-addressable 영역의 논리적 처리
Purge	최신 laboratory technique으로 target recovery를 실행 불가능하게 함	잠재적으로 가능	더 강한 logical·physical technique
Destroy	최신 laboratory technique의 복구를 방어하고 저장매체를 다시 쓰지 못하게 함	불가	매체 파괴·폐기

NIST Rev.2는 가능한 경우 Clear보다 Purge를 사용하라고 설명하지만, 실제 technique은 media type·vendor 구현·민감도·재사용 계획·승인 표준에 따라 선택합니다. cloud·virtual storage에서는 물리 media에 직접 접근할 수 없으므로 cryptographic erase가 가능한 purge 선택지일 수 있지만, 암호화 구현·key 계층·외부 관리 key·traceability의 전제 조건을 검증해야 합니다.

14.2. verification과 validation

Verification은 사용한 technique이 성공적으로 끝났는지 확인합니다.

- 도구 completion status
- error·anomaly
- media health
- 파괴된 잔해와 사용 장비
- physical operation 완료 상태

Validation은 verification 결과, 민감도, 잔여 위험을 평가해 target data가 충분히 sanitization되었다고 승인하거나 거부합니다. 거부하면 다른 technique으로 다시 수행하거나 더 강한 method로 상향합니다.

14.3. certificate of sanitization

증거	내용
media identity	vendor·model·serial·property ID·source

증거	내용
data classification	민감도·운영·손상 상태
method·technique	Clear·Purge·Destroy와 실제 technique
tool	제품·장비·version
verification	completion·오류·잔해·상태
validation	승인·거부·추가 action
performer·approver	이름·직책·일시·위치·서명
destination	내부 재사용·외부 재사용·recycle·제조사·기타

NIST Rev.2는 모든 media를 무조건 full sample로 읽는 것을 기본 요구로 두지 않고, 조직 정책에 명시된 경우를 제외하면 elaborate sampling이 필요하지 않다고 설명합니다. 대신 error·anomaly·도구 결과와 민감도 관점의 validation을 강조합니다.

15. 한 사건을 처음부터 끝까지 판정합니다

사건 · 14:31 잘못된 삭제

운영자가 tenant 조건을 빠뜨린 SQL로 회원 1,000명을 삭제했습니다. replica에도 5초 안에 반영됐고, 14:30까지의 WAL archive가 있습니다. 마지막 base backup은 전날 02:00, target RPO는 15분, RTO는 120분입니다.

15.1. 첫 15분

판단	행동	증거
추가 손상 방지	write path 차단·incident 선언	incident ID·시각
current state 보존	logs·WAL·SQL·actor 보존	hash·access restriction
복구 target	14:30:59 restore point	timezone·timeline
사용자 영향	삭제 대상·재입력 가능성	expected·actual count
복구 전략	isolated PITR 후 검증	owner 승인

15.2. 격리 복원

1. 전날 base backup과 target까지 WAL gap을 확인합니다.
2. 별도 network·credential의 isolated cluster에 복원합니다.
3. recovery target과 timeline을 확인합니다.
4. 14:31 삭제 전 회원 count·sample·invariant를 검사합니다.
5. schema·role·extension·configuration을 검사합니다.
6. application smoke와 outbound 차단을 확인합니다.

15.3. actual RPO·RTO

```
incident detected        14:33
last restored commit     14:30:52
target recovery point    14:30:59
application ready        16:18

actual data loss window  8 seconds from target
actual RTO               105 minutes from detection
```

목표를 통과해도 끝이 아닙니다. 잘못된 SQL의 원인, tenant guard, 변경 전 target SELECT, row count stop rule, production role, restore 속도, communication을 개선합니다.

사건 · 탈퇴 회원이 과거 restore에서 재등장

1. restore set이 만들어진 뒤 탈퇴한 회원 목록을 suppression ledger에서 불러옵니다.
2. production route를 열기 전에 restored DB에서 해당 subject를 재삭제합니다.
3. search·analytics·vendor로 재발행되는 job을 차단하거나 tombstone을 우선 적용합니다.
4. expected·actual count, 실패, retry, 승인 결과를 남깁니다.
5. 새 복구 지점이 이후 backup에 다시 반영되는지 확인합니다.

16. 데이터 생애주기 스튜디오 실습

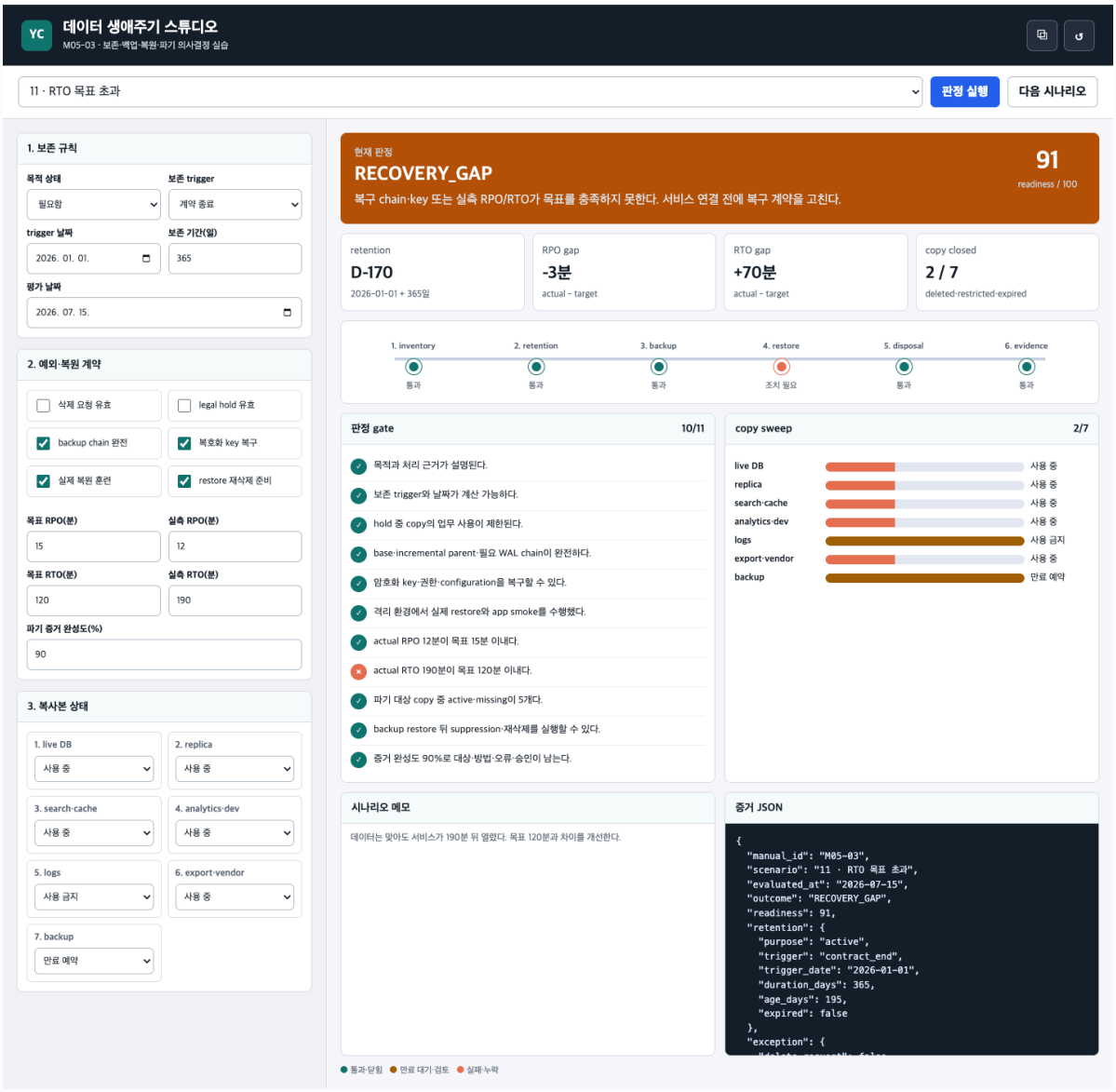


그림 15. 20개 시나리오에서 실제 날짜·RPO·RTO gap을 계산하고 RETAIN, HOLD, DELETE_NOW, RECOVERY_GAP, RESTORE_UNPROVEN, RESTRICT_UNTIL_EXPIRY, EVIDENCE_INCOMPLETE를 판정합니다.

16.1. 반드시 실행할 8개 시나리오

순서	시나리오	바꿔 볼 값	설명할 것
1	정상 보유 중	trigger date	만료까지 D-day
2	만료됐지만 live 잔존	live → deleted	DELETE_NOW가 닫히는 조건
3	유효한 legal hold	hold off	파기 재개 여부

순서	시나리오	바꿔 볼 값	설명할 것
4	검색 색인 누락	search → deleted	copy sweep count
5	RPO 목표 초과	actual 42 → 12	gap과 outcome
6	incremental parent 삭제	chain off → on	dependency 영향
7	검증만 하고 복원 안 함	drill off → on	verification과 restore 차이
8	파기 증거 부족	evidence 35 → 90	작업과 승인 증거 차이

16.2. 증거 JSON 읽기

```
{
  "outcome": "RECOVERY_GAP",
  "recovery": {
    "rpo": {"target": 15, "actual": 42, "gap": 27},
    "chain_complete": true,
    "restore_drill": true
  },
  "failed_gates": ["actual RPO 42분이 목표 15분 이내다."]
}
```

JSON은 법률 판단 엔진이 아니라 학습용 의사결정 기록입니다. 실제 조직에서는 policy version, authority, approver, system evidence link, ticket, vendor response를 붙입니다.

17. 60분 실전 과제

Mission A · 15분 · copy map

내 서비스의 가장 민감하거나 가장 중요한 data class 하나를 선택합니다.

- canonical system 1개
- 자동 파생 copy 3개
- 사람이 만든 export 1개
- 외부 vendor 1개
- backup 종류 2개

각 위치의 owner, subject 찾기 방법, TTL, 삭제 방법, restore 시 재생성 여부를 적습니다.

Mission B · 10분 · retention rule

다음 문장을 완성합니다.

```
[data class]는 [trigger]부터 [duration] 동안 보존한다.
[exception]이면 [scope]의 예정 파기를 [review date]까지 정지한다.
종료되면 [copy list]에 [action]을 실행하고 [owner]가 승인한다.
```

Mission C · 20분 · restore drill 카드

항목	내 결정
incident·target point	
target RPO·RTO	
restore set·dependency	
isolated environment	
data invariant 3개	
application smoke 3개	
security stop rule	
suppression·re-delete	
drill environment disposal	

Mission D · 15분 · 삭제·hold 판정

탈퇴 회원 1명과 결제 분쟁 회원 1명을 가정합니다. 같은 copy map에 서로 다른 action을 적고, 왜 다른지 `목적·근거·trigger·exception·owner`로 설명합니다.

18. 자주 실패하는 설계 12가지

실패	왜 위험한가	바꿀 증거
보유 기간만 적음	시작·예외·종료 행동 불명	5요소 규칙

실패	왜 위험한가	바꿀 증거
DB table만 inventory	search·CSV·vendor 누락	copy map
replica를 backup이라 부름	오삭제·공격이 함께 전파	독립 restore point
backup job 성공만 봄	corruption·기동·업무 오류 미확인	restore drill
RPO·RTO를 같은 숫자로 씀	손실 지점과 복구 시간 혼동	target·actual timeline
latest incremental만 보존	parent·WAL 누락	dependency graph
key를 backup하지 않음	암호화 파일을 해독 못함	key recovery test
PITR로 한 table만 복구한다고 가정	cluster 시점 복구와 불일치	isolated cluster·selective export
DELETE 후 backup을 잊음	restore에서 대상 재등장	expiry·suppression
legal hold를 전체 영구 보존으로 사용	범위 밖 데이터 과잉 보존	scope·review·release
파기 도구 exit 0만 기록	결과 충분성 미승인	verification·validation
drill 환경을 방치	새 민감 copy 생성	cleanup·sanitization certificate

19. 셀프 테스트 · 먼저 답을 적으세요

문제 1 · copy map

primary DB에서 회원 이메일을 삭제했습니다. 삭제 완료를 말하기 전에 확인할 copy class를 여섯 가지 쓰세요.

답: _____

문제 2 · retention rule

`결제 기록을 5년 보관` 문장에서 빠진 네 요소를 쓰세요.

답: _____

문제 3 · archive

archive로 옮겼다는 사실이 보존 근거와 파기 완료를 모두 증명하지 못하는 이유를 쓰세요.

답: _____

문제 4 · replica

read replica가 backup을 대신하지 못하는 대표 실패 상황을 두 가지 쓰세요.

답: _____

문제 5 · RPO·RTO

사고가 15:00, 마지막 복구 가능한 commit이 14:38, application ready가 17:10입니다. 사고 시각 기준 실제 data gap과 RTO는 각각 얼마입니까?

답: _____

문제 6 · backup evidence

manifest·checksum이 통과했어도 실제 restore drill이 필요한 이유를 세 가지 쓰세요.

답: _____

문제 7 · PostgreSQL dump

`pg_dump appdb` 하나만으로 cluster 전체를 복구할 때 빠질 수 있는 정보를 두 가지 쓰세요.

답: _____

문제 8 · PITR

PostgreSQL PITR에 필요한 핵심 두 구성과 WAL만으로 돌아오지 않는 configuration 예시 하나를 쓰세요.

답: _____

문제 9 · incremental

최신 incremental backup I2가 있는데도 복원할 수 없는 이유를 세 가지 쓰세요.

답: _____

문제 10 · backup 속 삭제

backup에서 개별 row 즉시 삭제가 기술적으로 어렵다고 판단한 경우의 운영 통제를 네 가지 쓰세요.

답: _____

문제 11 · legal hold

유효한 hold에 반드시 기록할 정보를 네 가지 쓰고, hold 해제 후 할 일을 한 가지 쓰세요.

답: _____

문제 12 · sanitization assurance

NIST SP 800-88 Rev.2 관점에서 verification과 validation의 차이를 쓰세요.

답: _____

20. 셀프 테스트 정답·해설

정답 1

예: replica, search·cache, analytics·AI feature, logs, export·dev copy, external vendor, backup 가운데 여섯 가지입니다. 실제 완료는 조직의 copy map에 존재하는 모든 위치를 대상으로 합니다.

정답 2

기간 외에 trigger, exception, action, owner가 필요합니다. 기간도 단위·끝 경계 계산 규칙을 명시합니다.

정답 3

archive는 저장·접근 상태의 변화일 뿐 목적·법적 근거·보존 종료일을 자동 생성하지 않습니다. archive 자체에도 접근 제한·trigger·duration·파기 방법과 증거가 필요합니다.

정답 4

오삭제·잘못된 UPDATE가 replica에도 전파되는 경우, 같은 관리자 계정·cloud account·network 실패로 원본과 replica가 함께 손상되는 경우가 대표적입니다.

정답 5

사고 시각과 마지막 commit의 차이는 22분입니다. 사고 15:00부터 application ready 17:10까지 RTO는 130분입니다. 조직이 RPO를 사고 시각 외 다른 기준으로 측정한다면 그 경계를 계약에 명시합니다.

정답 6

검사 도구가 running server의 모든 동작을 검증하지 못하고, 업무상 올바른 row·invariant를 알지 못하며, role·secret·extension·application smoke·RPO/RTO를 대신 측정하지 못하기 때문입니다.

정답 7

cluster-wide role과 tablespace 정의가 대표적입니다. 또한 다른 database, configuration, extension·OS dependency·secret을 별도 계약으로 확인합니다.

정답 8

적절한 base backup과 목표 시점까지 끊김 없는 WAL sequence가 핵심입니다. 예: `postgresql.conf`, `pg_hba.conf`, `pg_ident.conf`는 WAL만으로 복원되지 않습니다.

정답 9

full parent 또는 중간 incremental parent가 없거나, 필요한 WAL·WAL summary가 없거나, manifest·암호화 key·호환 server가 없으면 복원하지 못할 수 있습니다.

정답 10

live·파생 copy 즉시 처리, backup data의 업무 사용·접근 금지, 정해진 backup 만료 schedule, 임의 연장 차단, restore suppression·재삭제, expiry·재삭제 증거 가운데 네 가지입니다. 실제 허용 방식은 적용 법률·기술·조직 정책을 검토합니다.

정답 11

근거·reason, 정확한 scope, owner·approver, started_at, review_at, release condition, access restriction 가운데 네 가지입니다. 해제 후 남은 보존 근거를 재평가하고 없으면 scheduled disposal을 재개합니다.

정답 12

verification은 사용한 파기 technique이 완료되었는지 도구 결과·오류·잔해·media 상태를 확인합니다. validation은 그 결과와 민감도·잔여 위험을 평가해 충분성을 승인하거나 거부하고 재수행·상향을 결정합니다.

21. 완료 체크리스트

inventory·retention

- [ ] data class마다 목적·근거·owner가 있다.
- [ ] primary, replica, derived, log, export, vendor, backup을 지도에 올렸다.
- [ ] subject를 각 copy에서 찾는 key가 있다.
- [ ] 보존 규칙에 trigger·duration·exception·action·owner가 있다.
- [ ] archive의 접근·보존·파기 규칙이 별도다.

- [ ] legal hold에 scope·review·release가 있다.

backup·restore

- [ ] RPO와 RTO를 업무 영향으로 승인했다.
- [ ] target과 latest drill actual을 분리했다.
- [ ] backup copy·credential·failure domain의 독립성을 검토했다.
- [ ] encryption key·configuration·extension도 복구 계약에 있다.
- [ ] full·incremental·WAL dependency graph가 있다.
- [ ] manifest·checksum과 실제 restore를 모두 수행한다.
- [ ] 격리 환경에서 data invariant·app smoke·보안 stop rule을 검증한다.
- [ ] drill 환경을 파기하고 증거를 남긴다.

deletion·evidence

- [ ] 삭제 요청의 identity·scope·exception을 확인한다.
- [ ] copy별 expected·actual·failure·retry를 기록한다.
- [ ] backup beyond-use·expiry schedule을 기록한다.
- [ ] restore 전 suppression·재삭제 gate가 있다.
- [ ] 증거에 원 개인정보를 불필요하게 복제하지 않는다.
- [ ] media 파기 method·technique·tool version을 기록한다.
- [ ] verification과 validation을 분리한다.
- [ ] operator와 approver의 책임이 분리되어 있다.

22. 공식 참고 자료

PostgreSQL 18

- Backup and Restore
- SQL Dump
- File System Level Backup
- Continuous Archiving and Point-in-Time Recovery
- pg_basebackup
- pg_verifybackup

- pg_combinebackup
- pg_dump
- pg_restore

복구·보안·저장매체 파기

- NIST Recovery Point Objective glossary
- NIST Recovery Time Objective glossary
- NIST SP 800-34 Rev.1 Contingency Planning Guide
- NIST SP 800-88 Rev.2 Guidelines for Media Sanitization
- NIST SP 800-88 Rev.2 official PDF
- CISA #StopRansomware Guide

개인정보 보존·파기

- 대한민국 개인정보 보호법 제21조
- 개인정보 보호법 시행령 제16조 · 국가법령정보센터
- 개인정보보호위원회 개인정보 처리방침 작성지침 2025.4
- EU GDPR official text · Article 5 storage limitation
- ICO Storage limitation guidance
- ICO Right to erasure guidance

버전 기준일: 2026-07-15. PostgreSQL current 문서는 지원 중인 18 계열을 확인했습니다. NIST SP 800-88 Rev.2는 2025년 9월 최종본이며 Rev.1을 대체합니다. 대한민국 법령은 국가법령정보센터의 현재 조문을 우선 확인해야 합니다. EU·영국 자료는 운영 비교 예시이며 한국 법률의 대체 근거가 아닙니다. 이 교재는 기술·프로세스 학습 자료로 개별 사안의 법률 자문을 제공하지 않습니다.