세션 2

# 로그 수집·인덱싱

## 인덱스·역색인 · 로그 스키마 · 정규화 · _bulk 적재

01
### 인덱스·역색인
문서·인덱스·샤드와 검색을 가속하는 역색인

02
### 로그 스키마
실습에서 쓸 필드 레퍼런스와 타입

03
### 정규화
이질적 로그를 공통 필드명으로 매핑

04
### _bulk 적재
문서를 직접 넣고 인덱스·매핑 확인

S2

로그 수집·인덱싱

# 세션 개요와 학습 목표

> 로그가 어떻게 문서로 저장되고 검색 가능한 형태가 되는지 익힌다.

## 이 세션에서 얻는 것

- **인덱스·역색인**: 문서가 인덱스에 담기고 역색인이 만들어져 검색이 빨라지는 원리를 설명할 수 있다.
- **로그 스키마**: 실습에서 쓸 필드 레퍼런스와 text·keyword 타입의 차이를 구분해 이해한다.
- **정규화와 적재**: 원본 로그를 공통 필드로 정규화하고 _bulk API로 직접 적재해 본다.

## 실습 준비

- **공용 서버 접속**: 개인 PC 설치 없이 브라우저로 공용 대시보드에 접속하고 Dev Tools Console을 연다.
- **기본 인덱스**: loggen이 채우는 logs-demo-* 인덱스가 이미 있어 바로 매핑·문서를 확인할 수 있다.

S2

로그 수집·인덱싱

# 인덱스·문서·샤드

> 검색 엔진의 저장 단위를 문서·인덱스·샤드·매핑으로 나눠 이해한다.

## 핵심 단위

- **문서**: 로그 한 건이 하나의 JSON 문서이며 @timestamp·event·client_ip 같은 필드로 구성된다.
- **인덱스**: 유사한 문서를 모아 담는 컨테이너로, 이 랩은 logs-demo-YYYY.MM.DD를 쓴다.
- **샤드**: 인덱스를 물리적으로 나눈 조각으로, 데이터를 분산 저장해 대용량과 병렬 검색을 지원한다.
- **매핑**: 각 필드의 타입(date·keyword·ip·number·text)을 정의해 검색·집계 동작을 결정한다.

## 인덱스 수명주기

- **일자별 분할**: 인덱스명이 날짜로 끝나 하루치씩 쌓이므로 오래된 인덱스를 통째로 관리·삭제하기 쉽다.
- **패턴으로 통합**: logs-demo-* 와일드카드로 여러 날 인덱스를 한 번에 검색·집계한다.

S2
로그 수집·인덱싱

# 역색인과 분석기(토큰화)

분석기가 text를 토큰으로 쪼개 역색인에 담아야 검색이 가능하다.

```
[원본 문서]
doc1: url = "/search?id=1"
doc2: url = "/login"

[분석기(analyzer) – text 토큰화]
"/search?id=1"
  -> ["search", "id", "1"]
"/login"
  -> ["login"]

[역색인 – 토큰 → 문서 목록]
"search" -> [doc1]
"id"     -> [doc1]
"login"  -> [doc2]

질의 url:search
  → 곧바로 [doc1] 반환
```

## 동작 원리

- **분석기**: text 필드 값을 소문자·토큰 단위로 쪼개 부분 검색과 전문 검색을 가능하게 한다.
- **keyword**: 분석하지 않고 원문을 통짜로 저장해 정확 일치와 집계에 쓴다.
- **역색인**: 토큰에서 문서 목록으로 거꾸로 가리켜 전체 스캔 없이 검색을 끝낸다.

S2
로그 수집·인덱싱

# text 필드 vs keyword 필드

> 같은 값이라도 타입에 따라 검색·집계 동작이 완전히 달라진다.

| 항목 | text vs keyword |
| --- | --- |
| 저장방식 | text=토큰화 / keyword=원문 |
| 용도 | text=전문검색 / keyword=정확일치·집계 |
| 집계 | text=불가 / keyword=가능 |
| 예시필드 | url·user_agent=text / event·geo.country=keyword |
| 멀티필드 | url.keyword 서브필드로 둘 다 |

## 집계 규칙

- **.keyword 필수**: 정렬·집계엔 원문을 통짜로 가진 keyword 필드가 필요하다.
- **Terms 오류**: text 필드를 Terms 집계에 쓰면 "invalid for Terms aggregation" 오류가 난다.
- **멀티필드**: url은 검색용 text, url.keyword는 집계용으로 함께 매핑된다.

S2

로그 수집·인덱싱

# 로그 스키마 필드 레퍼런스

> 모든 쿼리는 아래 필드명을 정확히 써야 검색이 된다.

| 필드 | 타입 | 예시 | 의미 |
|---|---|---|---|
| @timestamp | date | 2026-07-08T04:12:33Z | 이벤트 발생 시각(UTC) |
| event | keyword | access · attack · auth_fail | 이벤트 분류(핵심) |
| client_ip | ip | 203.0.113.7 | 요청 출발 IP |
| geo.country | keyword | KR · US · CN · RU | 접속 국가 코드 |
| http.method | keyword | GET · POST · PUT | HTTP 메서드 |
| http.status | number | 200 · 401 · 403 · 404 · 500 | 응답 상태코드 |
| http.bytes | number | 200 ~ 40000 | 응답 크기(byte) |
| url | text | /login · /.env | 요청 경로(+payload) |
| response_ms | number | 2 ~ 900 | 응답 시간(ms) |
| user_agent | text | Chrome · curl · sqlmap | User-Agent 문자열 |
| attack_type | keyword | sqli · xss · path_traversal | 공격 유형 |

S2
로그 수집·인덱싱

# 핵심 필드 심화

> 필드 간 관계와 조건부 존재를 알아야 쿼리가 정확해진다.

## 조건부 필드

- **event**: access·attack·auth_fail로 나뉘는 최상위 분류 필드로 대부분의 쿼리가 여기서 출발한다.
- **attack_type·severity**: event가 attack인 문서에만 존재하며 공격 유형과 심각도를 담는다.
- **user·geo.country**: user는 auth_fail 시 로그인 시도 계정, geo.country는 ISO2 국가코드를 담는다.

## 필드명 정확성

- **오타 주의**: attack_type을 attacktype으로 잘못 쓰면 결과가 0건이 되며 오류 없이 조용히 빈 화면만 나온다.
- **중첩 표기**: geo.country·http.status처럼 점(.)으로 중첩 구조 하위 필드를 지정한다.

S2
로그 수집·인덱싱

# 실제 문서 예시(JSON)

> 이벤트 유형에 따라 존재하는 필드가 달라진다.

```
# 인증 실패 문서 — user 필드 존재
{
  "@timestamp": "2026-07-08T04:12:33.482Z",
  "event": "auth_fail",
  "client_ip": "203.0.113.7",
  "geo": { "country": "RU" },
  "http": { "method": "POST", "status": 401, "bytes": 512 },
  "url": "/login",
  "response_ms": 88,
  "user_agent": "python-requests/2.31",
  "user": "admin"
}
# 공격 문서 — attack_type·severity 존재, url에 payload
{
  "@timestamp": "2026-07-08T04:13:01.107Z",
  "event": "attack", "attack_type": "sqli", "severity": "high",
  "client_ip": "10.0.0.9",
  "url": "/search?id=1' OR '1'='1",
  "http": { "method": "GET", "status": 500 }
}
```

S2
로그 수집·인덱싱

# 로그 정규화

> 서로 다른 원본 로그를 하나의 공통 스키마로 맞추는 과정이다.

## 정규화란

- **필드 매핑**: 원본 로그 라인을 파싱해 client_ip·http.status 같은 공통 필드명으로 옮기는 것이 정규화다.
- **타입 강제**: 파싱 시 status는 숫자, @timestamp는 날짜로 타입을 맞춰야 범위·정렬 쿼리가 동작한다.
- **중첩 표현**: http.method·geo.country처럼 관련 필드를 객체로 묶어 구조를 명확히 한다.

## 정규화의 이점

- **공통 스키마**: ECS 유사 스키마를 쓰면 방화벽·웹서버·인증 등 이질적 소스를 한 스키마로 검색한다.
- **재사용성**: 쿼리·대시보드·알럿 규칙을 소스가 달라도 그대로 재사용할 수 있다.

S2
로그 수집·인덱싱

# 수집 파이프라인

수집기가 원본 로그를 파싱·정규화해 엔진으로 전송한다.

| 수집기 | 특징 |
| --- | --- |
| Logstash | ELK · 풍부한 필터·변환 |
| Beats | ELK · 경량 전송 에이전트 |
| Data Prepper | OpenSearch 수집 파이프라인 |
| Fluent Bit | 경량 · 컨테이너 환경 |
| Ingest Pipeline | 엔진 내부 전처리 |

## 선택 관점

- **역할**: 수집기는 파싱·정규화·전송을 담당해 원본 로그를 문서로 바꾼다.
- **조합**: 경량 수집기로 모으고 엔진 Ingest Pipeline으로 전처리하는 조합이 일반적이다.

S2

로그 수집·인덱싱

# _bulk 로 문서 적재

Dev Tools Console에서 여러 문서를 한 번에 넣는다.

```
# 문서 여러 개 한 번에 적재 (_bulk)
# NDJSON: (액션 줄) + (소스 줄) 을 쌍으로 반복
POST logs-demo-2026.07.09/_bulk
# 액션 줄 – {"index":{}} 는 새 문서 색인
{ "index": {} }
# 소스 줄 – 인증 실패 문서
{ "@timestamp":"2026-07-09T04:00:00Z","event":"auth_fail","client_ip":"10.0.0.9","user":"admin","url":"/login","http":{"method":"POST","status":401} }
# 액션 줄
{ "index": {} }
# 소스 줄 – SQLi 공격 문서
{ "@timestamp":"2026-07-09T04:00:01Z","event":"attack","attack_type":"sqli","severity":"high","client_ip":"10.0.0.9","url":"/search?id=1' OR '1'='1","http":{"method":"GET","status":500} }
# 각 줄은 한 줄로 작성, 마지막 줄 끝에 개행 필요
```

S2
로그 수집·인덱싱

# 인덱스·매핑 확인

적재 후 필드 타입과 문서 수가 의도대로인지 확인한다.

```
# 매핑 확인 — 각 필드의 타입 점검
GET logs-demo-*/_mapping
#   event·geo.country 는 keyword, url·user_agent 는 text 인지 확인

# 문서 수 확인 — 적재가 반영됐는지
GET logs-demo-*/_count

# 문서 1건 조회 — 실제 필드 구조 확인
GET logs-demo-*/_search
{ "size": 1 }

# 인덱스 목록 — 일자별 인덱스와 문서 수·용량
GET _cat/indices/logs-demo-*?v
#   docs.count 가 늘었으면 적재 성공
```

S2
로그 수집·인덱싱

# 인덱스 패턴 logs-demo-*

> 여러 일자 인덱스를 하나로 묶어 Discover에서 검색하게 한다.

## 인덱스 패턴이란

- **와일드카드 통합**: logs-demo-* 패턴이 여러 일자 인덱스를 하나로 묶어 한 번에 검색·집계한다.
- **기본값 지정**: 이 랩은 logs-demo-*가 기본 인덱스 패턴으로 지정돼 있어 바로 Discover에서 시작한다.
- **시간 필드**: 패턴 생성 시 시간 필드를 @timestamp로 지정해 시간 범위 검색을 활성화한다.

## 문제 해결

- **UTC 주의**: 로그는 UTC로 저장되므로 데이터가 안 보이면 시간 범위를 넓혀 Refresh한다.
- **자동 복구**: 패턴이 없다는 안내가 뜨면 대개 10분 내 복구되며, 급하면 Dashboards Management에서 생성한다.

S2
로그 수집·인덱싱

# 세션 정리와 다음 예고

> 로그가 문서로 저장되고 검색 가능해지는 전 과정을 묶어 정리한다.

## 기억할 것

- **역색인**: 분석기가 text를 토큰으로 쪼개 역색인에 담아야 대량 로그도 즉시 검색·집계된다.
- **스키마**: event가 핵심 분류 필드이며 집계엔 keyword(또는 .keyword) 타입이 필요하다.
- **정규화·적재**: 원본을 공통 필드로 정규화하고 _bulk로 적재한 뒤 매핑·문서 수를 확인한다.

## 다음 예고 — 세션 3

- **KQL 검색**: Discover 검색창에 KQL을 입력해 이벤트·상태코드·공격 유형을 필터링한다.
- **연산자**: AND·OR·범위·와일드카드로 조건을 조합해 의심 트래픽을 추려낸다.