
📝 Working with Apache Access Logs in Sumo Logic

🔹 Step 1: Basic Search with _sourceCategory

Query:

_sourceCategory=Labs/Apache/Access

✅ Explanation

- _sourceCategory is a **metadata tag** that classifies logs into categories (in this case, Labs/Apache/Access for Apache access logs).
- Running this simple search will return all raw Apache access logs in that category.
- Example log entry:
- 192.168.1.5 - - [19/Aug/2025:10:45:12 +0000] "GET /index.html HTTP/1.1" 200 2326
- This log contains important fields such as IP address, request method, URL, HTTP version, and **status code (200 in this case)**.

🔹 Step 2: Parsing Status Codes

Query:

_sourceCategory=Labs/Apache/Access

| parse "HTTP/1.1\" * " as status_code

✅ Explanation

- The parse operator is used to **extract specific values** from raw logs.
- In Apache logs, the HTTP status code appears right after "HTTP/1.1".
- The statement "HTTP/1.1\" * " tells Sumo Logic to:
  - Look for the text HTTP/1.1" in the log.
  - Extract the next value (represented by *).
  - Store it as a new field called status_code.

Example Output:

status_code

200

404

500

Now instead of just raw logs, you have a structured field (status_code) that can be counted, grouped, or filtered.

🔹 **Step 3: Identifying Client vs Server Errors**

Query:

```
_sourceCategory=Labs/Apache/Access
| parse "HTTP/1.1\" * " as status_code
| if(status_code matches "4*", 1, 0) as client_err
| if(status_code matches "5*", 1, 0) as server_err
| sum(server_err) as server_errors_cnt, sum(client_err) as client_errors_cnt
```

✅ **Explanation of Operators**

1. **if(condition, value_if_true, value_if_false)**
   - A conditional operator that assigns values based on logic.
   - Example:
     - if(status_code matches "4*", 1, 0) → If the status code starts with 4 (client error), mark it as 1, else 0.
     - if(status_code matches "5*", 1, 0) → If the status code starts with 5 (server error), mark it as 1, else 0.
2. **matches**
   - Used inside the if condition to check patterns.
   - "4*" matches any status code starting with 4 (like 400, 403, 404).
   - "5*" matches any status code starting with 5 (like 500, 502, 503).
3. **sum(field)**
   - Adds up all the values in a field.

    - Since client_err and server_err are either 1 or 0, summing them gives the **total number of errors**.

Final Output Example:

```
server_errors_cnt   client_errors_cnt
-----------------   -----------------
12                  85
```

📌 **Why This Matters**

- **Client Errors (4xx):** Show when users make invalid requests (e.g., 404 Not Found, 403 Forbidden). High counts may indicate broken links or probing activity.
- **Server Errors (5xx):** Show when the server fails to handle a request. High counts may indicate server issues or overload.
- By parsing and aggregating with conditions, you transform **raw logs into actionable insights** that can highlight problems in real-time.
