
🔹 **Query Breakdown**

```
_sourceCategory=Labs/Apache/Access status_code=404
| timeslice 1m
| count(status_code) as error_count by _timeslice
| outlier error_count window=10, consecutive=1, threshold=3, direction=+-
```

---

1. _sourceCategory=Labs/Apache/Access status_code=404

- Pulls logs from the **Apache access logs**.
- Filters only logs with HTTP **status code = 404** (page not found).
- So we're only monitoring failed requests.

---

2. | timeslice 1m

- Breaks the log data into **1-minute time buckets**.
- This lets us analyze trends over time, rather than looking at one big blob of logs.

---

3. | count(status_code) as error_count by _timeslice

- Counts the number of 404s per each 1-minute slice.
- Renames that count as **error_count**.
- Output: a time series showing how many 404s happened every minute.

---

4. | outlier error_count window=10, consecutive=1, threshold=3, direction=+-

This applies **statistical anomaly detection** to the error counts.

Let's break down the parameters:

- **window=10**
    - Uses the last **10 data points (minutes here)** as the baseline to compute a moving average and standard deviation.

- Example: at 12:20, it looks back at counts from 12:10 → 12:19 to determine what is “normal.”

- **consecutive=1**
  - Requires **only 1 consecutive anomaly** to trigger an outlier.
  - If set to consecutive=3, it would need **3 abnormal points in a row** before flagging.
- **threshold=3**
  - Outlier is flagged when the data point is **3 standard deviations away** from the moving average.
  - Standard deviation measures how spread out the numbers are.
  - 3σ threshold is a common choice in anomaly detection (very unlikely to occur randomly).
- **direction=+-**
  - Detects both **spikes up (+)** and **drops down (-)**.
  - If set to +, it would only flag unusually high error counts.
  - If set to -, it would only flag unusually low counts.

🔹 **What the Query Does Overall**

- Monitors **404 error rates** per minute.
- Calculates a baseline of normal behavior (moving average of last 10 minutes).
- Flags any **sudden spike or drop** in 404 errors that is **statistically abnormal (≥3σ away)**.

🔹 **Example of a Log Source Where outlier is Helpful**

👉 **Failed Login Attempts (Authentication Logs)**

Query:

_sourceCategory=Auth/FailedLogins

| timeslice 5m

| count(user) as failed_logins by _timeslice

| outlier failed_logins window=12, consecutive=2, threshold=2, direction=+

- **Use case**: Detect brute force attacks.
- If normally there are 2–3 failed logins every 5 minutes, but suddenly there are 50, the outlier operator will flag it.
- Helps SOC analysts catch **suspicious spikes** without manually setting static thresholds.

✅ **In summary:**

- outlier looks for abnormal patterns based on statistics (not fixed rules).
- **window = history used to learn normal**
- **consecutive = how many anomalies in a row required**
- **threshold = sensitivity (std devs away from mean)**
- **direction = look for spikes, drops, or both**