Creating meaningful alerts in a SIEM (Security Information and Event Management) system is critical because it directly affects how well a security team can detect, understand, and respond to real threats, without getting overwhelmed.

We can use the following query to find spikes in client errors within last 60 minutes.

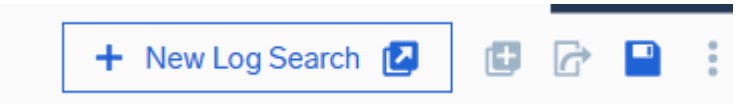
Query - _sourceCategory=Labs/Apache/Access (status_code=200 or status_code=404)
| timeslice 1m
| if (status_code = "200", 1, 0) as successes
| if (status_code = "404", 1, 0) as fails
| sum(successes) as success_cnt, sum(fails) as fail_cnt by _timeslice
*| (fail_cnt/(success_cnt+fail_cnt)) * 100 as failure_rate_pct*
| outlier failure_rate_pct window=5, threshold=3, consecutive=1, direction=+ 6.
| where failure_rate_pct_violation > 0

This query is designed to monitor HTTP 404 errors in your Apache access logs and compare them against successful 200 OK responses, checking every minute. It calculates the percentage of failed requests — essentially, how many of the total requests are ending in a "Not Found" error. By tracking this failure rate over time, the query builds a picture of what "normal" looks like. Then, it applies statistical analysis using the outlier operator to detect when that failure rate suddenly increases beyond expected levels. If such a spike occurs, it flags the event as an anomaly. This kind of monitoring is useful for quickly identifying broken links, misconfigured routes, unexpected bugs in your web application, or even suspicious traffic patterns that might indicate probing or scanning. By catching these issues early, you can respond before users are significantly affected.

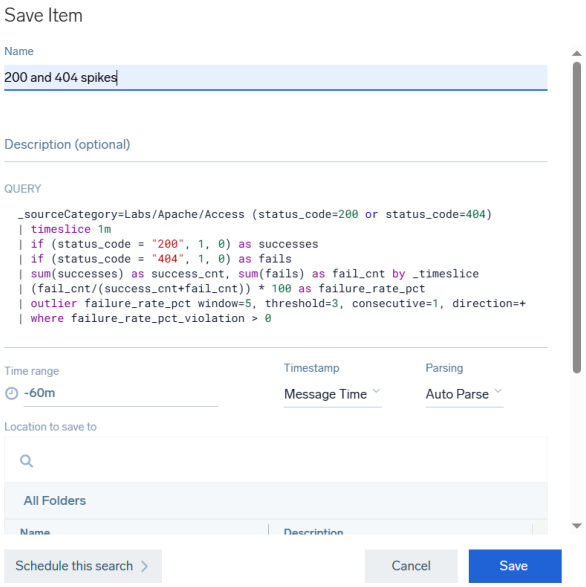
Here is the result of this query.

#	Time	succ...	fail_...	failur...	failur...	failure...	failure...	failure...	failure_rate_pct_vi...	failure...
1	06/19/2025 3:10:00.000 PM	162	14	7.95455	6.43547	2.04284	3.71539	1	1	4.23915

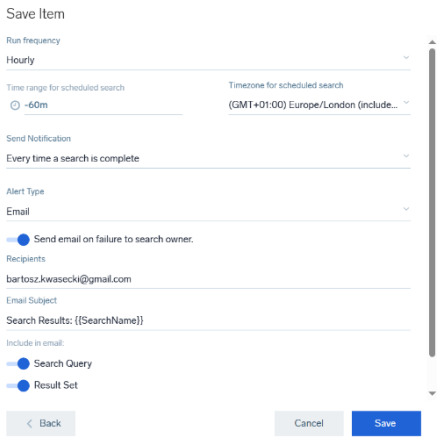
We can see that within the last 60 minutes, there was only 1 one minute period that had a sudden increase in the expected levels of client errors. To create an alert, we should click on the save as icon.



Next, we should set the name of the search and click ‘Schedule this search’.



Then, just set up a frequency of your alerts and the email address on which you want to receive them. Click save, and you have just set up a new alert.



Alerts can be very helpful for plenty of reasons. They can help with detecting:

Failed Login Spikes:

Leverage the outlier operator to identify sudden surges in failed login attempts. These anomalies may signal brute-force attacks, credential stuffing, or automated login abuse targeting your authentication systems.

Unusual Administrative Activity:

Continuously monitor AWS IAM or other admin-level operations for unexpected bursts of activity. A spike in privileged actions can indicate insider threats, compromised credentials, or unauthorized changes to critical infrastructure.

Access to Sensitive URLs:

Track traffic patterns to high-risk endpoints, such as /admin, /wp-login.php, or custom backend panels. Sharp increases in access attempts may reveal reconnaissance, vulnerability scanning, or early stages of an exploitation attempt.

Denied Firewall Traffic:

Alert on significant upticks in denied firewall connections. These could reflect DDoS attempts, port scanning behavior, misconfigured network rules, or unauthorized access attempts probing your perimeter defenses.