

Parsing logs in SUMO Logic is crucial because it modifies the raw log data into structured fields. That makes the search queries more accurate with better results.

Examples

In the first example, we will perform an unparsed log search. As a result, we will receive unparsed logs from the source category starting with “Labs/Apache” within the last 15 minutes.

*Query - _sourcecategory=Labs/Apache/**

#	Time	Message
1	05/23/2025 1:36:05.133 PM	178.233.100.114 - - [2025-05-23 12:36:05.133 +0000] "GET /testimonials/ref=vgfb_ssdsd_4 HTTP/1.1" 401 7036 "http://www.bing.com/search?q=sumo%20logic&src=IE-SearchBox&FORM=IE11SR" "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_6_7; en-us) AppleWebKit/533.21.1 (KHTML, like Gecko) Chrome/19.0.1084.30 Safari/536.5" 6/6310508 Host:apache-prod Name:Http Input Category:Labs/Apache/Access Index:Apache_Access1

Under the message section, we can notice the IP address 178.233.100.114. Being an unparsed log, SUMO will not recognize this IP address as a separate field. Therefore, a result of a search query to count the occurrence of each IP address will bring back a general result.

*Query - _sourcecategory=Labs/Apache/**

| count by ip_address

<<	<	1	of	1	>	>>	Time Compare	v
#	ip_address	_count						
1		3,796						

In the second example, we will use a query parsed with the help of a regular expression (RegEx). A regular expression looks for a given pattern and returns the requested result creating a specific field. Using a query below, we can see the desired outcome.

*Query - _sourceCategory=Labs/Apache/**

| parse regex

"(?<ip_address>\d{1,3}.\d{1,3}.\d{1,3}.\d{1,3})"

| count by ip_address

<<	<	1	of	2	>	>>	Time Compare	v
#	ip_address	_count						
1	78.235.33.64	102						
2	192.168.44.33	32						
3	158.69.196.112	153						
4	70.69.152.165	195						
5	5.35.225.115	89						
6	192.11.22.33	125						
7	161.71.8.142	118						
8	65.98.119.36	300						

For the third example, we can use the unparsed Apache/Error source category first.

Query - `_sourceCategory=Labs/Apache/Error`

46	05/23/2025 2:19:17.839 PM +0100	[2025-05-23 13:19:17.839 +0000] [error] mod_log_sql: database connection error : Can't connect to local MySQL server through socket '/var/lib/mysql/mysql.sock' (13)	Host:apache-prod ▾ Name:Http Input ▾ Category:Labs/Apache/Error ▾ Index:Error ▾
47	05/23/2025 2:19:17.839 PM +0100	[2025-05-23 13:19:17.839 +0000] [crit] [client 138.124.80.163] Invalid method in request XYZ /dksmnvdcss HTTP/1.0, referer: http://www.yoursites.com/	Host:apache-prod ▾ Name:Http Input ▾ Category:Labs/Apache/Error ▾ Index:sumologic_default ▾

From the unparsed search result, we want to individualize two entities: the first one will be the client IP address, and the second will be the mod_log_sql message. To refine our search query, we have to input the text below in the search field.

```
_sourceCategory=Labs/Apache/Error
| parse "[client *]" as client_ip nodrop
| parse "mod_log_sql: *" as message nodrop
```

#	Time	client_ip	message	Message
22	05/23/2025 2:24:17.838 PM +0100		child spawned but unable to open database link	[2025-05-23 13:24:17.838 +0000] [error] mod_link
23	05/23/2025 2:24:17.838 PM +0100		child spawned but unable to open database link	[2025-05-23 13:24:17.838 +0000] [error] mod_link
24	05/23/2025 2:24:17.838 PM +0100	118.116.15.26.22		[2025-05-23 13:24:17.838 +0000] [crit] [client 118.116.15.26.22 /dksmnvdcss HTTP/1.0]

In the final search, we see separate fields for the client_ip and mod_log_sql message. In SUMO Logic, by default, when using the “parse” command, only log lines that match the pattern are kept. “Nodrop” keyword prevents this behaviour by not “dropping” those log lines, but showing every log instead.