Notes from the 6/11/2019 Sumo Logic Fundamentals Training

gist.github.com/ToniRib/9f0a731c9d3002531aa9b72ae382197b

262588213843476

Notes from the 6/11/2019 Sumo Logic Fundamentals Training

Sumologic Training

Certification is on home page under certifications (need to be logged in to my own account)

Training Environment

- service.sumologic.com
- training+labs@sumologic.com
- pw: `Sum0Labs!`

Basics

Metadata

- _collector: Name of the collector (defaults to hostname)
- _sourceHost: Hostname of the server (defaults to hostname)
- _sourceName: Name and Path of the log file
- _source: Name of the source this data came through
- **_sourceCategory: Can be freely configured. Main metadata tag (e.g. labs/apache/access)**

Querying

- Implicit AND if you put in multiple keywords

  OR must be specified explicitly
- Nothing in the scoping line is case sensitive (keywords, sourceCategory)

  If you need to enforce case, you have to use the where operator
- single line comment `//` or multi line `/*`
- Time range directly affects query performance
  - `-15m` is the default
  - For debugging, change it from a relative time to an absolute

- Be as specific as possible: the smaller result set, the better
- Double quotes:
    - `*` outside of double quotes is a wildcard matcher
    - `*` inside of double quotes is literal
- `where` can be used with `=`, `in`, or `matches`
    - `matches` accepts wildcards `*` inside double quotes
    - `matches` also accepts regex
- if statements
    - Can use ternary `condition ? result1 : result2 as new_field`
    - Can also use regular `if`

        override a field if null: `if (isNull(new_field), 0, new_field) as new_field`
- Field extraction rules are PREPARSED rules (parsed upon ingestion)
    - Docs: https://help.sumologic.com/Manage/Field-Extractions
    - Allow you to use key value pairs in your scope. Doing so dramatically improves performance.
- Pipe character `|` is the operator delimiter, separating one operator for the next

    Putting new statements on new lines is for human readability

Parse

- 2 main kinds of parsing: `parse regex` and regular `parse`
- Regex
    - Don't debug in Sumo: copy your log message and put it into a regex validator to write the regex, then copy it back into Sumo
    - Uses PHP flavor of regex
    - Can use the keyword `multi` to detect multiple occurrences of the same string. It will show them on multiple lines.

        Docs: https://help.sumologic.com/05Search/Search-Query-Language/01-Parse-Operators/02-Parse-Variable-Patterns-Using-Regex#Parse_multi
- Anchor Parse: combo of plain text parsing and wild cards
    - Uses text within string as an 'anchor' if it doesn't change
    - Highlight text -> right click -> parse selected text -> select what you want and name it
    - Creates the parse statement and adds it to your query
    - Characters scattered throuhgout string are "anchors" and the "*" are greedy wildcards
- You can parse nested fields by doing `parse field=<fieldname> <more parsing stuff>`

- For all parse operators, messages must match at least one segment of the parse expression or they are dropped from the results
  - You can use `nodrop` to keep results that don't meet your criteria
  - Docs: https://help.sumologic.com/05Search/Search-Query-Language/01-Parse-Operators/Parse-nodrop-option

Saving

By default, queries save to your Personal folder unless you select something different

Timeslicing

- Uses the `timeslice` operator and can specify any amount of time you want to slice by
- Docs: https://help.sumologic.com/05Search/Search-Query-Language/Search-Operators/timeslice
- Often used for aggregates (count, avg, etc. by timeslice)

Outlier

- Can do something like `outlier _count` after doing a `count` aggregate to show standard deviations and detect things outside of a normal range
- `threshold` is 3.0 std devs by default
- *Must be used with a timeseries*
- Docs: https://help.sumologic.com/05Search/Search-Query-Language/Search-Operators/outlier
- Alerts
  - Add `where` conditions on `_count_violation`
  - Can see on table when this occurred

Dashboards

- When you're debugging a panel, it will automatically open the search using absolute time
- Can use master timerange to apply all panels to a given timerange
- Click the page looking icon on any panel to see/update the underlying query

Alerts

- 'Alert' and 'Scheduled Search' are synonymous
- Put a query onto a schedule and that gives you the ability to notify on it

Live Search

- Better to use `_collector`

    collectors allow for spaces, so wrap the name in double quote
- Mostly used for data ingest validation
- Not all of the query operators can be used here

App Catalog

- Pre-built dashboards and queries
- Point it to the sourceCategory where the logs are located

Security

https://help.sumologic.com/01Start-Here/Quick-Start-Tutorials/Hands-on_Labs02%3A_Security_Analytics

Advanced Querying

Log Reduce

- Buckets logs together with counts of them (removes variable text and leaves just the structure)
- Called a "signature analysis"
- Docs: https://help.sumologic.com/05Search/LogReduce
- Most effective when something is happened, but you don't know why or what (and there's a whole lot of logs)
- There are actions to help you guys the signature analysis (thumbs up/down)

    Recomputes the relative score trying to determine what is relevant to you

Lookup

- Can reference other same information
- Docs: https://help.sumologic.com/05Search/Search-Query-Language/Search-Operators/lookup
- Threat lookup and geo lookup tables are built into every deployment

- Can use save to create your own lookups: https://help.sumologic.com/01Start-Here/Quick-Start-Tutorials/Hands-on_Labs02%3A_Security_Analytics/11Lab_11_-_Creating_Your_Own_Lookup
    - use `append` with this if you don't want to overwrite the file
    - Don't forget the name of your file. You won't be able to find it again (until some new beta feature is released)
- Kind of performs as a join operator

Surrounding messages

- Allows you to view all of the messages from the metadata tag (hostname, source name, or category) for a given timeframe
- Should highlight and jump you to the original message

LogCompare

- Can run signature comparisons for multiple groups or logs
- Can look at time ranges or against different machines or metadata features
    - Time reference: gives you a +/- percentage for some base (`_deltaPercentage`)
    - `_isNew` shows you only new messages
- Docs: https://help.sumologic.com/05Search/LogCompare

Time Compare

- Can do an aggregate then add a time compare over multiple days
- Something like `<scope> | count user | compare with timeshift 1d 7` which would show user counts compared at the same time each day, for the last 7 days

Search Templates

- Helps non-power users to be able to query in SumoLogic
- Can take any part of the query and create a parameter from it for easy input
- Docs: https://help.sumologic.com/05Search/Get-Started-with-Search/How-to-Build-a-Search/Search-Templates
- Can close the query box and only show the input portion (which is less scary)
- Could use wildcards, but it would have to make sense in the context of the parameter you pulled out
- You can also give the user a list of options to choose from

Correlation Operators

Transaction

- Docs: https://help.sumologic.com/05Search/Search-Query-Language/Transaction-Analytics
- Allows you to analyze related sequences of messages based on a unique transaction identifier such as a SessionID or IP Address
    - Group all of the logs with the same source IP together
    - Map log messages to a state
    - Syntax like `transaction on <field> with states <state1>`
- Field is used for state tracking, timing, and other purposes

    Can track transitions from one state to another `results by flow`

Transactionize

- Docs: https://help.sumologic.com/05Search/Search-Query-Language/Transaction-Analytics/Transactionize-operator
- Groups messages based on a field

Subqueries

- Docs: https://help.sumologic.com/05Search/Subqueries
- Allows you to run an inner query and use that as a match in the other query
- Subquery is defined inside square brackets and with the keyword `subquery:`
- To pass the inner to the outer, you can use a save in the subquery and a lookup in the outer query
- subqueries can be nested (though this will affect performance)
- If you use `compose` without `keywords` then you have to use what field you're passing to the parent query in a `where` clause

    Using `keywords` puts it in the scope and speeds up your query performance

Leave a comment

[Markdown is supported]()