Saratoga

Logs Analyzer and Pre-Filters

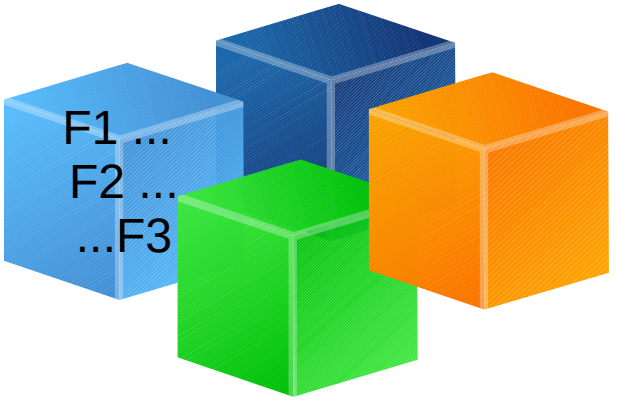
Use Case 1

- JSON Structured Logs AWS Cloud Trail zipped or unzipped Logs
- Logs Analyzer traverses terabytes of logs, with any JSON structure. For all keys, their unique values the analyzer brings back the count. Example for a particular requestID “abcdefgh1234”, it brings back the count of all occurrences of “abcdefgh1234”.

Use Case 2

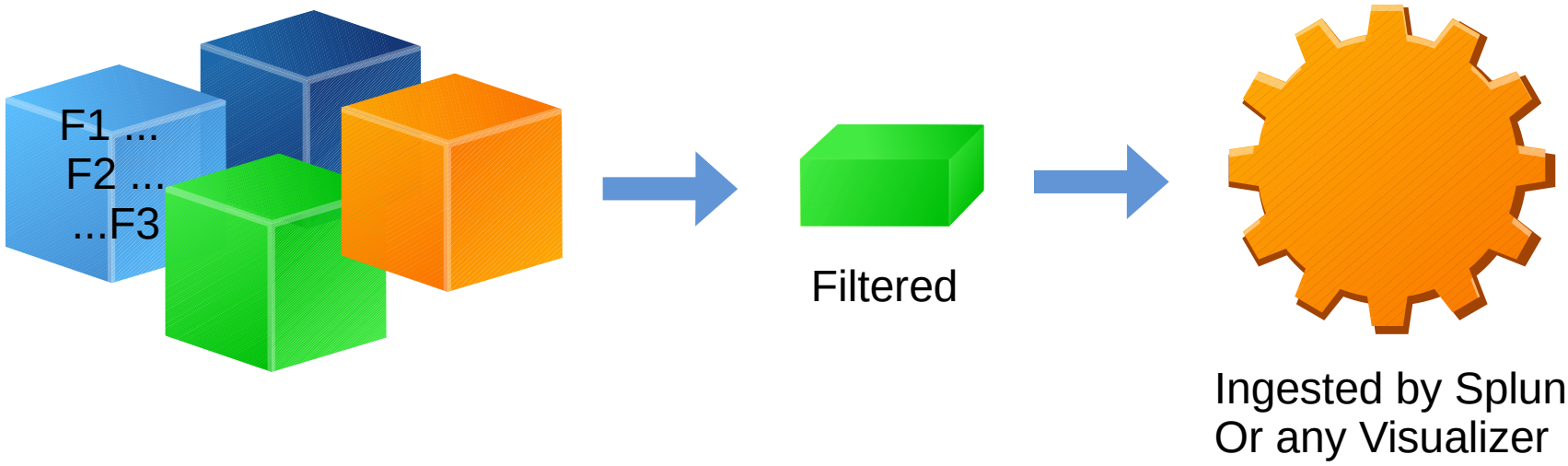
- Unstructured Logs. First Use Case cannot be applied.
- Looks for patterns and returns the Log excerpts. F1 .. F2 .. Fn filters applied to it. Example: F1 can be “requestID”, F2 can be “abcedfgh1234” and F3 can be an ending like “Error”.

F1..F2..F3..Fn can be any sequence, it just has to be in JSON format. Using the sequence, the analyzer looks for in all logs and brings back entire text if it finds the sequence.



Use Case 3

- Use Case 2 + a file output mechanism to be sent to other systems like Logstash, ELK or Splunk for ingestion. The output could be hundredth the size of the original containing relevant filtered data.



Features

- Saratoga Logs Analyzer runs “Headless” (With No GUI)
- It runs on Docker and is built ground up on Node.JS.
- For large log files, minimum 4 core CPU and 8 GB RAM is recommended.
- For APIs, installation notes, readme-s and issues; please check https://github.com/devb-saratoga/saratoga

Open Source MIT License

- Saratoga Logs Analyzer is under Open Source since 2016. It is distributed under MIT License.