SPL to ES|QL Equivalency Workaround

Overview

This document outlines workarounds for transitioning Splunk Processing Language (SPL) commands to Elasticsearch Query Language (ES|QL), focusing on commands with no direct ES|QL equivalents. Where possible, alternative approaches using ES|QL functions or APIs are provided.

Command Workarounds

inputlookup

- **SPL Command:** inputlookup
  Description (SPL): Loads data from a saved lookup file.
- **ES|QL Equivalent:** ENRICH
  Description (ES|QL): Reads data from a stored dataset or file.

Examples:
SPL Example: `| inputlookup csoc_aws_account_id_lookup.csv cloud.account.id OUTPUT account_name`
ES|QL Equivalent:
FROM logs

| ENRICH enrich-csoc_aws_account_id_lookup-policy ON cloud.account.id WITH account_name

-

multisearch

- **SPL Command:** multisearch
  Description (SPL): Combines results from multiple searches.

- **ES|QL Equivalent:** UNION (via Multi-search API)
  Description (ES|QL): Merges results from multiple queries.

- This can be accomplished with the Multi-search API
  https://www.elastic.co/guide/en/elasticsearch/reference/8.8/search-multi-search.html

Examples:
SPL Example:

| multisearch

[search index=aws_logs eventType=login]

[search index=firewall_logs action=blocked]

ES|QL Equivalent:

POST _msearch

{}

{ "query": { "bool": { "must": [{ "match": { "event.type": "login" } }] } }, "index": "aws_logs" }

{}

{ "query": { "bool": { "must": [{ "match": { "action": "blocked" } }] } }, "index": "firewall_logs" }

fillnull

- **SPL Command:** fillnull
  Description (SPL): Replaces NULL values with a default.

- **ES|QL Equivalent:** EVAL with COALESCE
  Description (ES|QL): Uses COALESCE to replace NULL values with a default.

- Closest command is "IS_NULL" - the use case may be able to be fulfilled with "| WHERE IS_NULL" and an EVAL command

Examples:
SPL Example: `| fillnull value="N/A" field_name`
ES|QL Equivalent:

FROM logs

| EVAL field_name = COALESCE(field_name, "N/A")

- Alternative Using IS_NULL + EVAL

  FROM logs

| EVAL field_name = CASE(IS_NULL(field_name), "N/A", field_name)

join

- **SPL Command:** join
  Description (SPL): Merges records from different datasets based on a key.

- **ES|QL Equivalent:** No direct equivalent
  Description (ES|QL): Combines datasets using a common field.

- Couldn't find any specific workaround, I think maybe one can use enrich command in certain ways to achieve this.

Examples:
SPL Example:

index=aws_logs eventType=login

| join cloud.account.id [search index=account_info]

ES|QL Equivalent:

FROM aws_logs

| ENRICH enrich-csoc_aws_account_id_lookup-policy ON cloud.account.id WITH account_name

outputlookup

- **SPL Command:** outputlookup
  Description (SPL): Saves query results to a new dataset or file.

- **ES|QL Equivalent:** No direct equivalent
  Description (ES|QL): Saves processed data to an external location or dataset.

- Couldn't find any specific workaround.

dedup

- **SPL Command:** dedup
  Description (SPL): Removes duplicate values while keeping the first occurrence.

- **ES|QL Equivalent:** STATS FIRST(field) BY field
  Description (ES|QL): Keeps only the first occurrence of a field within grouped data.

- Can be accomplished using the filtering available in Kibana or in the Elastic query language(s)

Examples:
SPL Example: | `dedup user_id`
ES|QL Equivalent:

FROM logs

| STATS FIRST(user_id) BY user_id

append

- **SPL Command:** append
  Description (SPL): Appends results from another query.

- **ES|QL Equivalent:** No direct equivalent
  Description (ES|QL): Combines query results sequentially.

- No direct equivalent or workaround. // This could potentially be approximated using the Multi-search API: https://www.elastic.co/guide/en/elasticsearch/reference/8.8/search-multi-search.html // Generally, Elastic does not have a notion of subsearches, but this might be possible with nested aggregations, multi-searches, or with more complex visualizations.

iplocation

- **SPL Command:** iplocation
  Description (SPL): Adds geographic location data for IPs.

- **ES|QL Equivalent:** ENRICH GEOLOCATION
  Description (ES|QL): Enriches data with geographic information for IPs.

- There is no direct equivalent but one can use external enrich policy to add fields related to IP location.

Examples:
ES|QL Equivalent:

FROM logs

| ENRICH geoip-enrichment ON ip_field WITH country, city, latitude, longitude

collect

- **SPL Command:** collect
  Description (SPL): Groups and collects field values.

- **ES|QL Equivalent:** BUCKET
  Description (ES|QL): Aggregates and collects field values.

- No direct equivalent, a potential workaround is to use bucketing. // Elastic does not have summary indices, but this could potentially be approximated by bucketing results.

latest

- **SPL Command:** latest
  Description (SPL): Gets the most recent value of a field.

- **ES|QL Equivalent:** No direct equivalent
  Description (ES|QL): Retrieves the latest value within each group.

- Could potentially be achieved using sort function.

Conclusion

This document provides practical workarounds for SPL commands lacking direct ES|QL equivalents. Advanced queries and optimizations can be further explored in the Elasticsearch documentation.