Tip Sheet: Cribl Search™

Basic Concepts

DATASETS AND DATASET PROVIDERS

Datasets
A *dataset* organizes and references data. It describes *what* to query, acting as a "container" for data from sources like filesystems, S3 buckets, or Edge workers. For example, a dataset named `myVPCFlowlogs` contains Amazon VPC Flow Logs and is referenced as `dataset=myVPCFlowlogs` in queries.

Each dataset includes a *Processing* section, where you can assign *Datatypes* to detect format, extract timestamps and other fields, and manipulate the parsed events.

Dataset Providers
A *dataset provider* categorizes datasets by defining *where* to send queries. Cribl Search's dataset providers contain connection information, such as API keys. Examples include object stores (e.g., S3) or APIs (e.g., AWS or Okta).

Supported Providers:

- Object Stores and Data Lakes:
  Amazon S3, Azure Blob Storage, Google Cloud Storage, MinIO, Amazon Security Lake, Cribl Data Lake (S3-based)
- APIs:
  AWS, Azure, Google Cloud Platform, Okta, Zoom, Tailscale, Google Workspace, Microsoft Graph, Generic HTTP
- Analytics Services:
  Azure Data Explorer, Log Analytics, Elasticsearch, OpenSearch
- Metrics Services:
  Prometheus
- Data Warehouses:
  ClickHouse, Snowflake

DATATYPES AND PARSING

Datatyping helps Cribl Search break data from datasets into discrete events, apply timestamps, and parse fields. Cribl Search supports default and custom datatypes, which you can define and test through an intuitive UI.

SEARCH QUERY

A *search* is a query expression that processes data and returns results. You build queries using *operators*, separated by one or more pipe | symbols. A basic Example:

```
dataset=myVPCFlowlogs | limit 1000
```

This retrieves data from `myVPCFlowlogs` and returns up to 1,000 results.

BUCKET PATHING

The *Bucket Path* defines a dataset's scope, using tokens and key-value pairs in a JavaScript expression. For example:

- `my-bucket/${data}/` extracts the data field for all events.
- `my-bucket/${data}/${*}` extracts data and the wildcarded path as fields.

List of Supported Data Formats:

- Gzip: `.gz, .gzip, .tgz, .tar.gz, application/x-gzip`
- Journal: `.journal, .journal~`
- LZ4 `.lz4`
- Parquet: `.parquet, .pqt, .parq`
- Snappy: `.snappy`
- Splunk Rawdata
- Tar: `.tar, .tgz, .tar.gz`
- Text: `.log, .csv, .json, .ndjson, .txt`
- ZIP: `.zip`
- Zstd: `.zst`

Operators and Functions

Operators process data in a search query. Here's an example with one operator:

```
dataset=myVPCFlowlogs | summarize count() by srcport
```

This aggregates all events in myVPCFlowlogs, counts events, and groups by srcport.

Commonly Used Operators

CATEGORY	OPERATOR	DESCRIPTION	EXPRESSION
Searching / Filtering	`limit`	Retrieves up to a specified number of events from the dataset. Controls access costs and data volume.	`| limit 1000`
	`where`	Filters events based on a boolean expression.	`| where field has "Cribl"`

	`cribl`	Finds specific events using string or comparison expressions. (This is Cribl Search's implicit operator, so you don't need to specify it in the query.)	`“goats”`
			`“goats” and (“climb” or “rock climb”)`
			`network in (“sector7”)`
			`earliest=-2h@h latest=-1h@min`

	`project`	Keeps only the fields specified, renames fields, or inserts new computed fields.	`| project cost=price*quantity, price`
	`project-away`	Excludes specific fields from the results, optionally using patterns or wildcards.	`| project-away price, quantity, zz*`
	`render`	Enforces a specific visualization of the search results, either event or table, overriding the default display format.	`| render table`
Sorting / Manipulation	`sort`	Arranges events in order by one or more fields.	`| sort by Timestamp asc`
	`extract`	Extracts information from a field via a parser or regular expression.	`| extract source=foobar type=csv "field1,field2,field3"`
	`extend`	Calculates one or more expressions, and assigns the results to fields.	`| extend Duration = CreatedOn - CompletedOn , IsSevere = Level == “Critical” or Level == “Error”`
Lookups / Joins	`lookup`	Retrieves fields from a lookup table through a first-match process.	`| lookup lookupTable on commonField`

	`join`	Merges events from two different data scopes, allowing for complex queries across datasets.	`let RightScopeName = RightScope;`
			`LeftScope`
			`| join [ JoinOptions ] RightScopeName on JoinConditions`

Aggregation	`summarize`	Aggregates the content of the input, allowing operations like min, max, or count over a dataset.	`| summarize count() by price`
Summarization	`timestats`	Aggregates events by time periods or bins. Useful for time-series analysis.	`| timestats span=1m count()`
	`count`	Returns the total number of input events.	`| count`

Search Optimization

OPTIMIZATION	INSTEAD OF THIS	DO THIS
Push “filter” expressions to the left-most side of the query.	`dataset="my-datasource" | where tenantId == "foo-bar-12345" | where proc == "bash" | where data_source == "stdout"`	`dataset="my-datasource" tenantId="foo-bar-12345" proc="bash" data_source="stdout"`
Use comma-separated functions in operators.	`… | extend field1="foo" | extend field2="bar" | extend field3="pike"`	`… | extend field1="foo", field2="bar", field3="pike"`
Specify fields in Parquet searches with project.	`dataset="a-parquet-datasource" | summarize sum(bytes) by customer, account`	`dataset="a-parquet-datasource" | project bytes, customer, account | summarize sum(bytes) by customer, account`
Move coordinator functions (e.g., lookup, sort) to the end.	`dataset="my-datasource" dataSource="VPC Flow Logs" | lookup service_names on dst_port | summarize count() by service_name`	`dataset="my-datasource" dataSource="VPC Flow Logs" | summarize count() by dst_port | lookup service_names on dst_port`

Search Path Optimization

CONCEPT	EXPLANATION
S3 Longest Static Prefix	If specifying field1 and field2 at a certain time (e.g., /bucket/inputID/foo/bar/2023/11/01/12/00/), this uses the longest static prefix to fetch data efficiently for S3 objects.
Reverse-Order Criteria	Use bucket/inputId/year/month/day/hour/minute/field1/field2/ to reduce unnecessary object search overhead, improving wall clock time and reducing costs.
Cribl Stream Destinations	Cribl Stream automatically partitions time on top, and allows adding fields below time boundaries for further efficiency.

ABOUT CRIBL

Cribl, the Data Engine for IT and Security, empowers organizations to transform their data strategy. Customers use Cribl’s vendor-agnostic solutions to analyze, collect, process, and route all IT and security data from any source or in any destination, delivering the choice, control, and flexibility required to adapt to their ever-changing needs. Cribl’s product suite, which is used by Fortune 1000 companies globally, is purpose-built for IT and Security, including Cribl Stream, the industry’s leading observability pipeline, Cribl Edge, an intelligent vendor-neutral agent, Cribl Search, the industry’s first search-in-place solution, and Cribl Lake, a turnkey data lake. Founded in 2018, Cribl is a remote-first workforce with an office in San Francisco, CA.

Learn more: www.cribl.io | Try now: Cribl sandboxes | Join us: Slack community | Follow us: LinkedIn and Twitter