sumo logic

METRICS OPERATORS

AGGREGATION

accum

Converts each time-series in the row to a series of running totals. The running total in each series starts from the value of the first data point in the series, then iteratively adds up successive values.

```
RequestCount | accum
```

avg

Calculates the average of all the resulting time series. If grouping is specified, it calculates the average for each group.

```
dep=prod metric=cpu_system | avg
cluster=search metric=cpu_idle | avg by node
```

count

Counts the total number of time series that match the query. If grouping is specified, it counts the total number for each group.

```
dep=prod | count
cluster=search | count by node
```

min

Calculates the minimum value of the time series that match the query. If grouping is specified, it calculates the minimum for each group.

```
dep=prod metric=cpu_system | min
cluster=search metric=cpu_idle | min by node
```

pct

Calculates the specified percentile of the metrics that match the query. If grouping is specified, it calculates the specified percentile for each group.

```
dep=prod metric=cpu_system | pct(95)
cluster=search metric=cpu_idle | pct(99.9) by node
```

max

Calculates the maximum value of the time series that match the query. If grouping is specified, it calculates the maximum for each group.

```
dep=prod metric=cpu_system | max
cluster=search metric=cpu_idle | max by node
```

sum

Calculates the sum of the metrics values that match the query. If grouping is specified, it calculates the sum for each group.

```
dep=prod metric=cpu_system | sum
cluster=search metric=cpu_idle | sum by node
```

RATE / DELTA

delta

Computes the backward difference at each data point in the time series to determine how much the metric has changed from its last value in the series.

```
metric=Net_InBytes Interface=eth0 | delta
```

rate

Computes a rate based on the forward difference at each time in the time series. The difference between the current and the next recorded value in a time series is scaled to a value per second

```
metric=Net_InBytes Interface=eth0 | rate
```

eval

Evaluates a time series based on a user-specified math expression.

```
_sourceCategory=ApacheHttpServer metrics=cpu_idle |
eval _value * 100
_sourceCategory=ApacheHttpServer metrics=re-
quest_per_sec | rate | eval max(_value, 0)
```

Check out our Query Library

http://community.sumologic.com

FILTER

filter

Filters a query to help reduce the number of series returned by applying a boolean test to some aggregate quantity.

```
cpu | filter avg > 80
```

topk

Select the top specified time series sorted by the value of a mathematical expression evaluated over the query time range.

```
metric=cpu_system | topk (10, max)
```

bottomk

Select the bottom specified time series sorted by the value of a mathematical expression evaluated over the query time range.

```
dep=prod metric=cpu_system | bottomk (5, max)
```

SEARCH OPERATORS

parse

Parses the given field to create new fields to use in the metrics query. If no field is specified while parsing Graphite metrics, the metric name is used.

```
dep=prod | parse *-search-* as deployment, instance
cluster=frontend | parse field=user **-* as user_id,
user_type
```

quantize

Segregates time series data by time period. This allows you to create aggregated results in buckets of fixed intervals (for example, 5-minute intervals).

```
_sourceCategory=hostmetrics | quantize to 5m
logins | quantize to 5m using sum
```

timeshift

Shifts the time series from your metrics query by the specified amount of time. This can help when comparing a time series across multiple time periods.

```
cluster=search metric=cpu_idle | timeshift 5h
```

JOIN METRICS QUERIES

You can perform basic math operations (+, -, *, /) on two or more metrics queries.

Measure average CPU usage across cluster by _sourceHost

To measure average CPU usage by _sourceHost in a cluster, add the average user and system CPU utilization, and use along to aggregate the metric by _sourceHost.

1. Query the average value of the CPU_User metric across the cluster.

```
metric=CPU_User cluster=franz | avg by _sourceHost
```

2. Query the average value of the CPU_System metric across the cluster.

```
metric=CPU_Sys cluster=franz | avg by _sourceHost
```

3. Add the two averages.

```
#A + #B along _sourceHost as CPU_by_sourceHost
```