Prometheus PromQL and promtool

CONCEPTS

An instant query returns one value per series at a single time; a range query evaluates at each step. Selectors give instant vectors, a [5m] suffix gives a range vector, and rate() turns a range vector back into an instant vector.

Selector http_requests_total{...}
↓ `[5m]`
Range vector samples per series
↓ `rate()`
Instant vector one value per series
↓ `sum by`
Aggregated fewer series

Counters only go up: always wrap them in rate() or increase() before graphing or summing. Use the plain value only for gauges.

SELECTORS

`<metric>`
Instant vector of all series

`<metric>{<label>="<v>"}`
Exact label match

`<metric>{<label>≠"<v>"}`
Label not equal

`<metric>{<label>=~"<re>"}`
Regex match

`<metric>{<label>!~"<re>"}`
Regex non-match

`<metric>[5m]`
Range vector over 5 minutes

`<metric> offset 1h`
Value from one hour ago

COUNTERS AND GAUGES

`rate(<counter>[5m])`
Per-second average rate

`irate(<counter>[5m])`
Rate from the last two samples

AGGREGATION

`sum(<expr>)`
Total across all series

`sum by (<label>) (<expr>)`
Sum, keeping only these labels

`sum without (<label>) (<expr>)`
Sum, dropping these labels

`avg by (<label>) (<expr>)`
Average per group

OVER TIME AND HISTOGRAMS

`histogram_quantile(0.95, rate(<m>_bucket[5m]))`
Quantile from buckets

OPERATORS AND LABELS

`<a> / ignoring(<label>) <b>`
Match ignoring these labels

`<expr> > 100`
Keep series above a threshold

RUNNING THE SERVER

`prometheus --config.file=<file>`
Start with a config file

`curl -X POST <host>:9090/-/reload`
Reload the config

PROMTOOL CHECK

`promtool check config <file>`
Validate config and referenced rule files

`promtool check rules <rules.yml>`
Validate rule files

PROMTOOL QUERY AND TEST

`promtool query instant <url> '<expr>'`
Run an instant query

`promtool test rules <tests.yml>`
Run rule unit tests