

Most Used PromQL Expressions

Expression	Purpose	Example
<code>http_requests_total</code>	Basic metric selection	Shows all time series for HTTP requests
<code>http_requests_total{status="500"}</code>	Label filtering	Filters requests with 500 status code
<code>rate(metric_name[5m])</code>	Calculates per-second rate over time	<code>rate(http_requests_total[5m])</code>
<code>sum(rate(metric_name[5m])) by (label)</code>	Aggregates rates by label	<code>sum(rate(http_requests_total[5m])) by (status)</code>
<code>avg_over_time(metric_name[5m])</code>	Average over time range	Useful for smoothing noisy metrics
<code>max_over_time(metric_name[1h])</code>	Maximum value in time range	Detects spikes
<code>node_memory_MemTotal_bytes - node_memory_MemFree_bytes</code>	Arithmetic	Calculates used memory
<code>(A > 100) and (B < 50)</code>	Logical comparison	Combines conditions
<code>up == 0</code>	Service health check	Detects if a target is down

Tips for Using PromQL Effectively

- Use `rate()` for counters that only increase.
- Use `irate()` for short-term spikes.
- Combine `sum()` and `by()` to group metrics meaningfully.
- Use label filters to narrow down noisy metrics.
- Use `offset` to compare current vs. past values