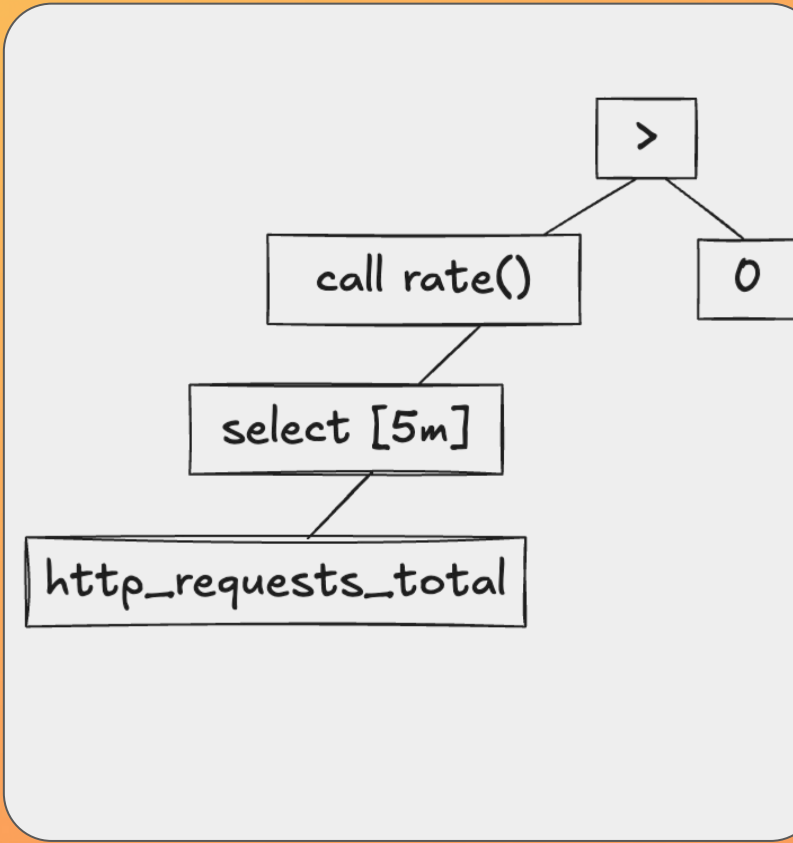
Grafana Labs

# Inside a PromQL Query

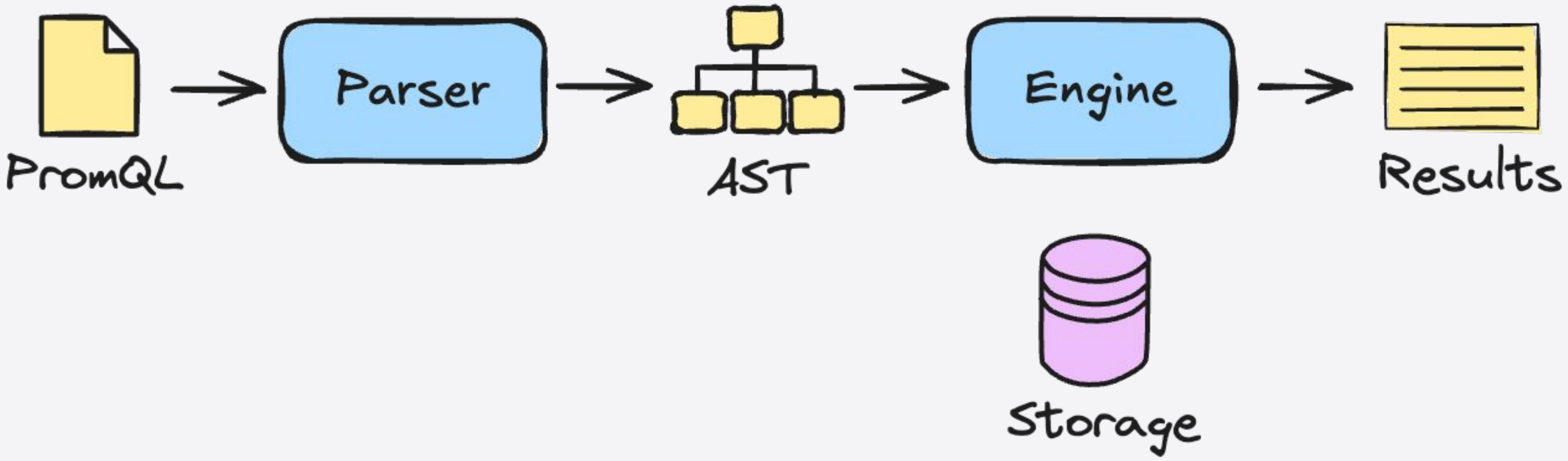


**Bryan Boreham**
Grafana Labs
@bboreham
@bboreham@grafana.social

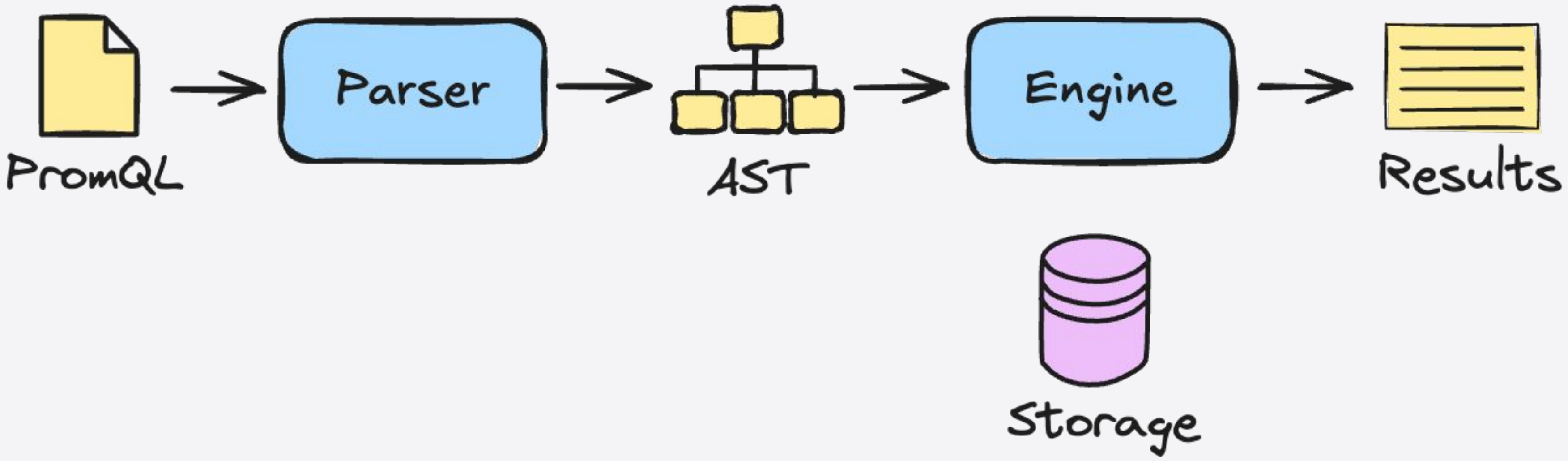
## Outline

- Overall Flow
- Selectors
- Instant and range queries
- Functions
- Aggregations (three styles)
- Operators
- Final sorting and output

## Overall flow



## Overall flow



## PromQL

A query is built up from:

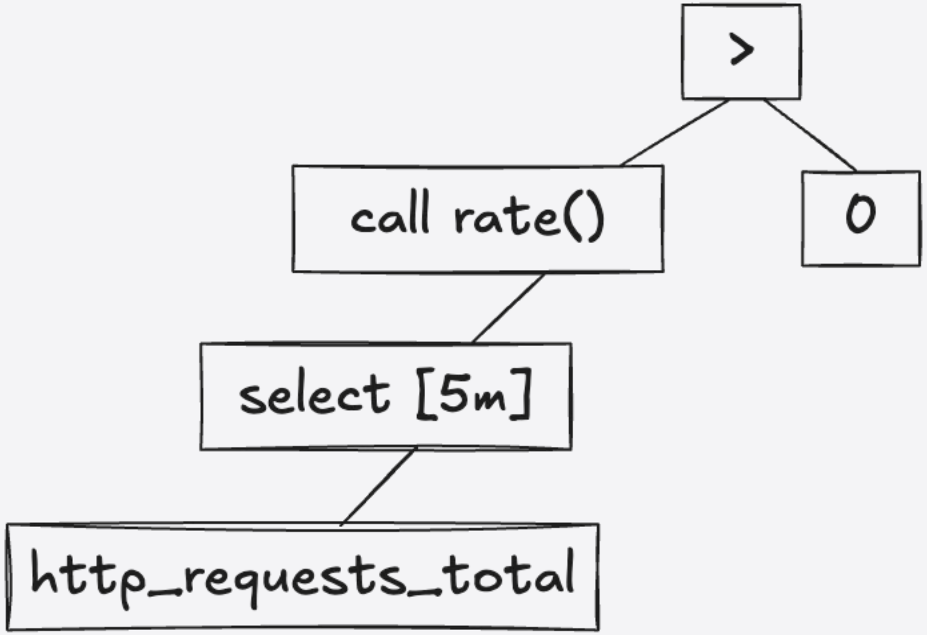
- Selectors: `http_requests_total{status="200"}`.
- Functions: `abs`, `rate`.
- Aggregations: `sum by (status)`.
- Operators: `>`, `+`.

## PromQL Example

```
rate(http_requests_total[5m]) > 0
```

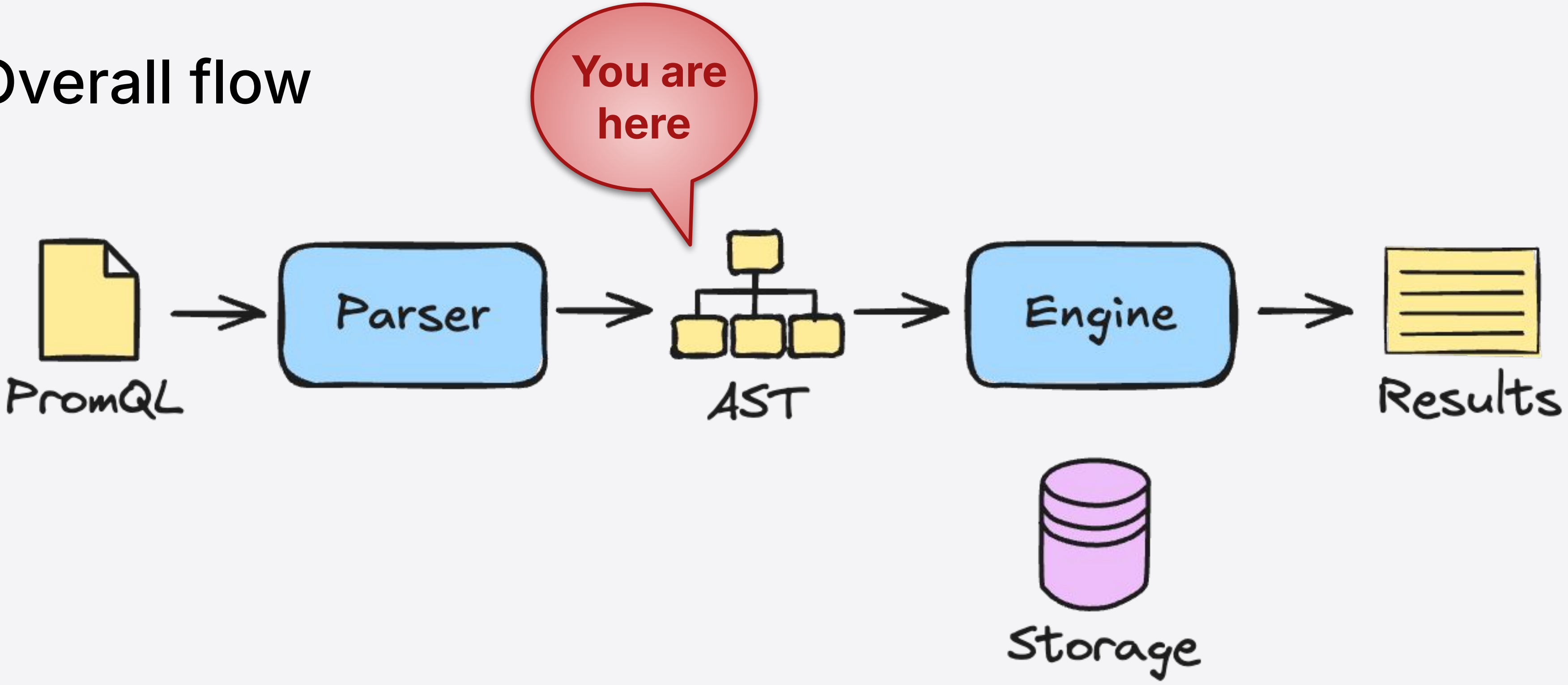
## Abstract Syntax Tree

```
rate(http_requests_total[5m]) > 0
```

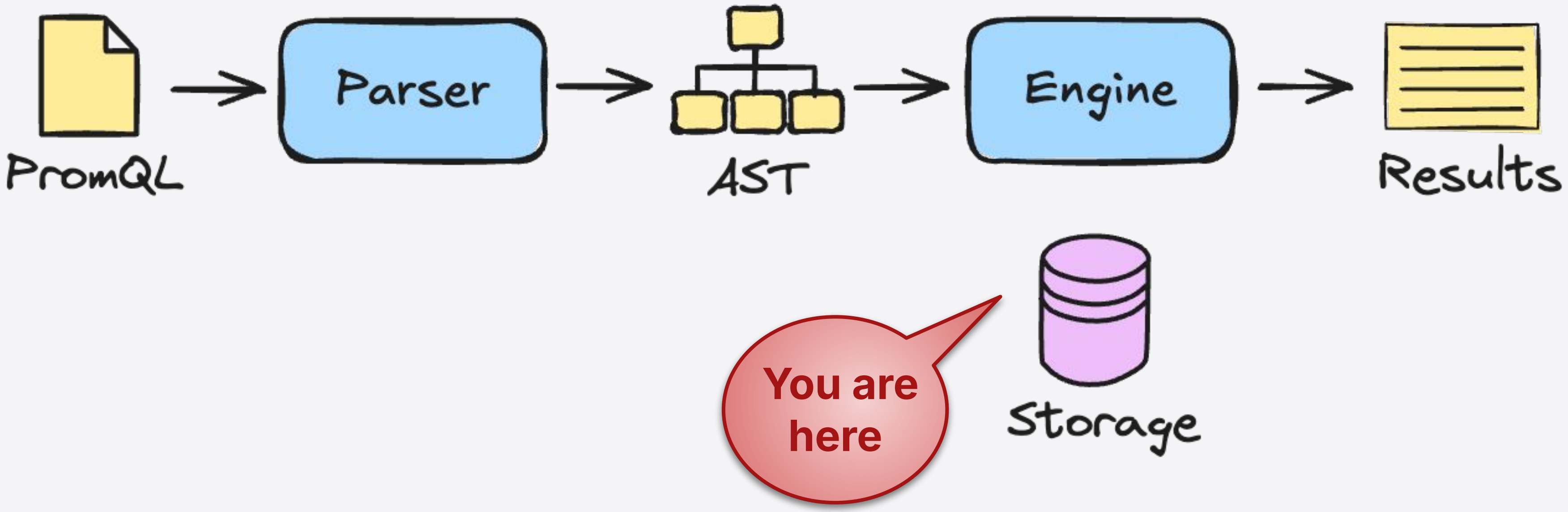


## There. Is. No. Query. Planner.

## Overall flow



## Overall flow

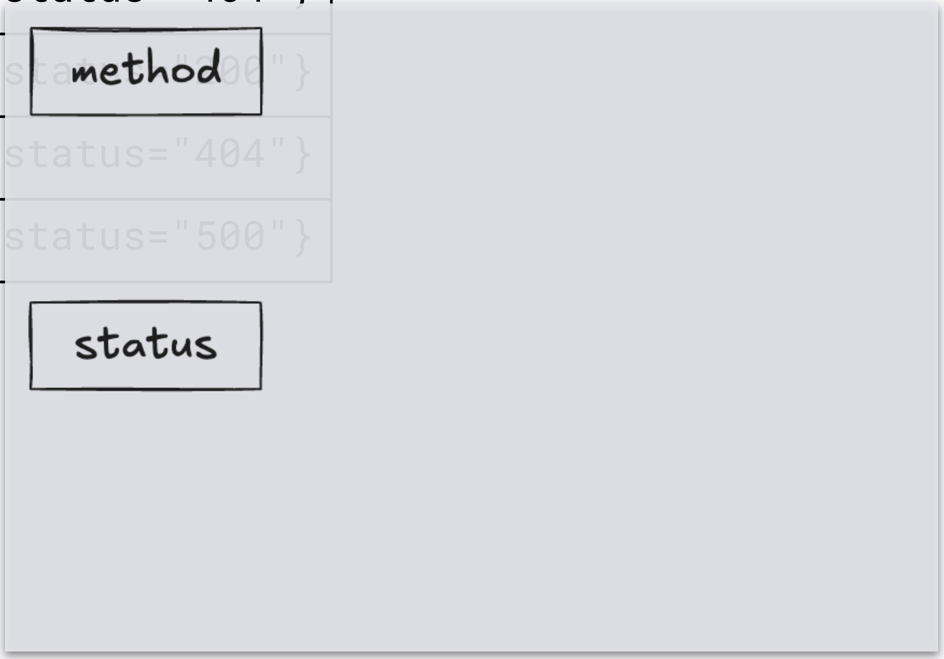


## Example data

| ID | Series |
|---|---|
| 13 | `http_requests_total{method="GET",status="200"}` |
| 42 | `http_requests_total{method="GET",status="404"}` |
| 23 | `http_requests_total{method="PUT",status="200"}` |
| 21 | `http_requests_total{method="PUT",status="404"}` |
| 73 | `http_requests_total{method="PUT",status="500"}` |

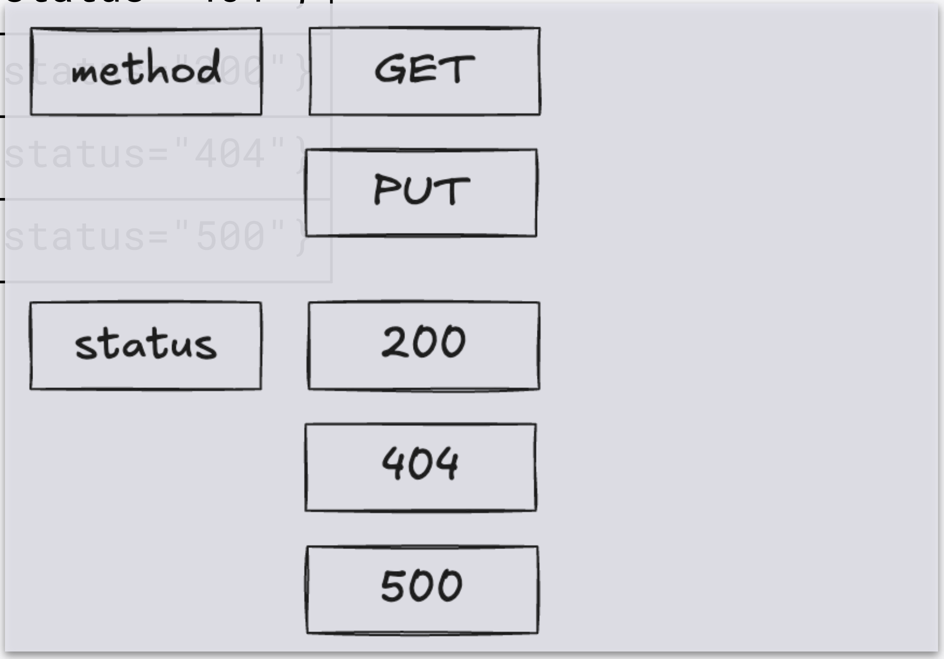
# Index the label names

| ID | Series |
|---|---|
| 13 | http_requests_total{method="GET",status="200"} |
| 42 | http_requests_total{method="GET",status="404"} |
| 23 | http_requests_total{method="PUT",status="200"} |
| 21 | http_requests_total{method="PUT",status="404"} |
| 73 | http_requests_total{method="PUT",status="500"} |



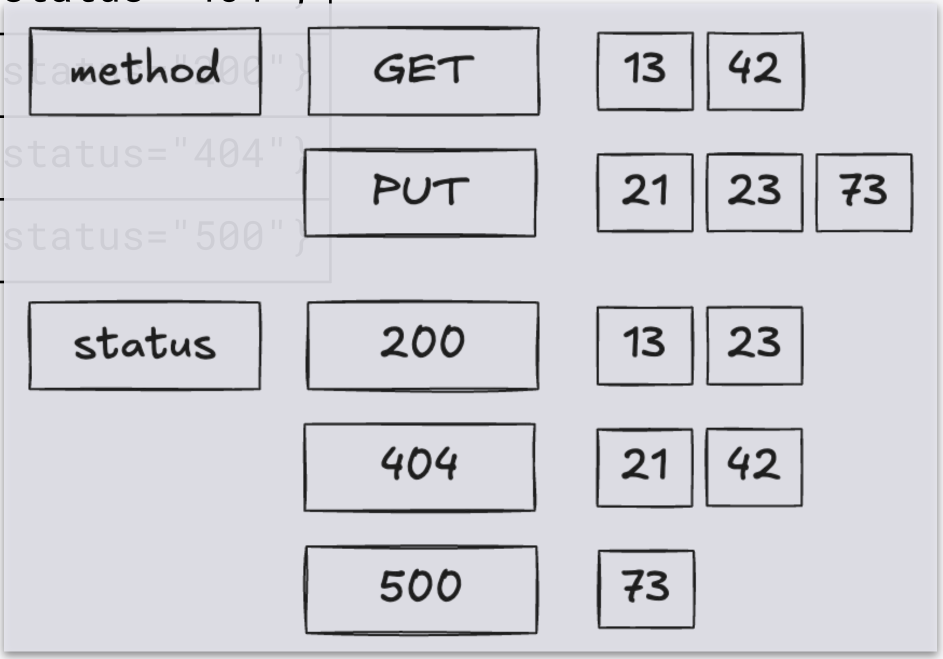
# Index the label values

| ID | Series |
|---|---|
| 13 | http_requests_total{method="GET",status="200"} |
| 42 | http_requests_total{method="GET",status="404"} |
| 23 | http_requests_total{method="PUT",status="200"} |
| 21 | http_requests_total{method="PUT",status="404"} |
| 73 | http_requests_total{method="PUT",status="500"} |



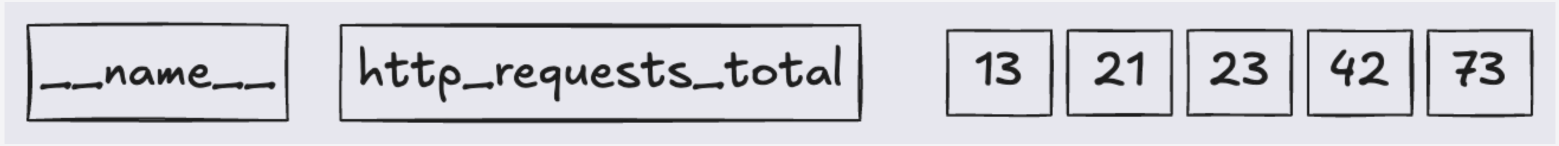
# Index the series

| ID | Series |
|---|---|
| 13 | `http_requests_total{method="GET",status="200"}` |
| 42 | `http_requests_total{method="GET",status="404"}` |
| 23 | `http_requests_total{method="PUT",status="200"}` |
| 21 | `http_requests_total{method="PUT",status="404"}` |
| 73 | `http_requests_total{method="PUT",status="500"}` |

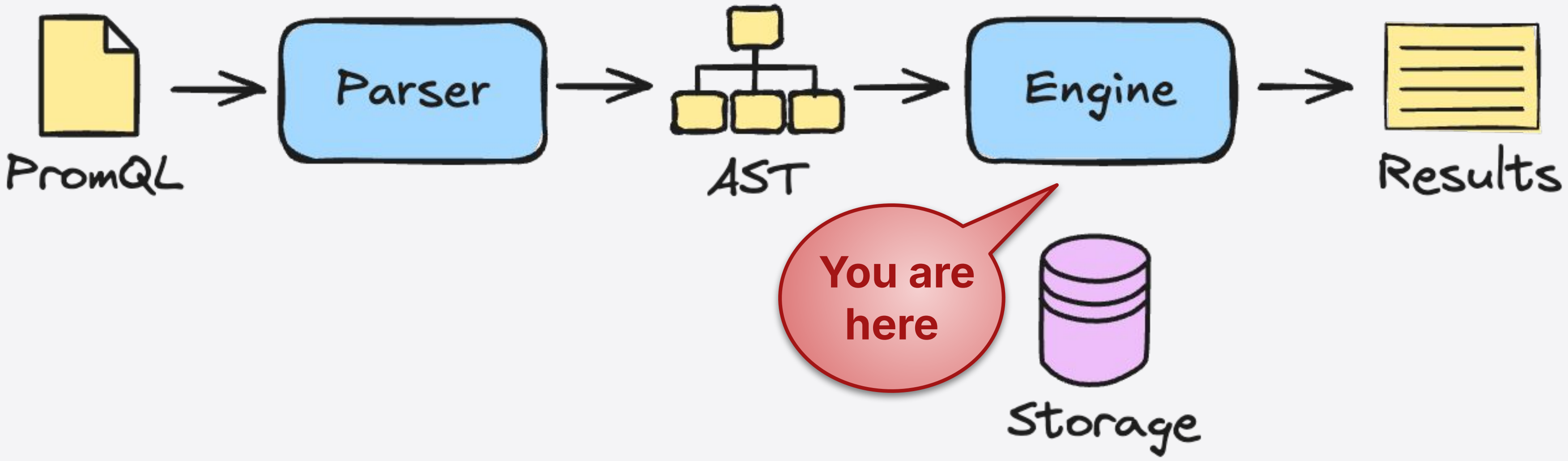


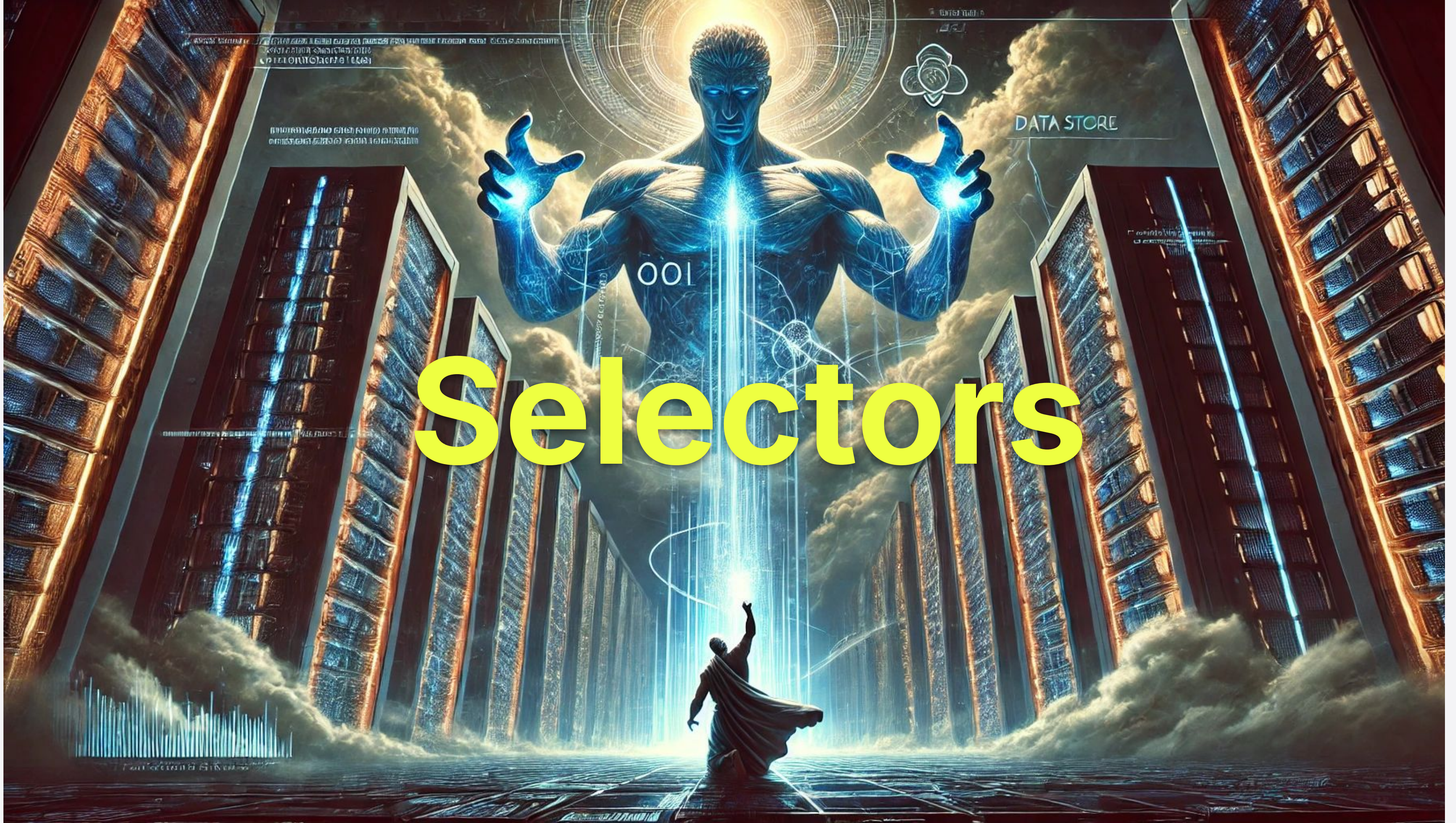
# Index series name

| ID | Series |
|---|---|
| 13 | `http_requests_total{method="GET",status="200"}` |
| 42 | `http_requests_total{method="GET",status="404"}` |
| 23 | `http_requests_total{method="PUT",status="200"}` |
| 21 | `http_requests_total{method="PUT",status="404"}` |
| 73 | `http_requests_total{method="PUT",status="500"}` |



## Overall flow

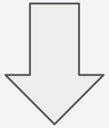




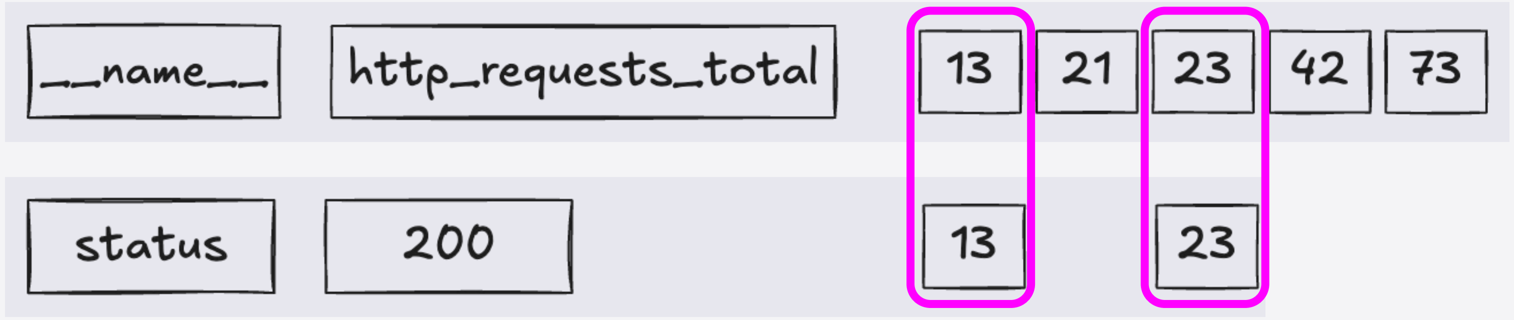
## Selectors

## Selector Example

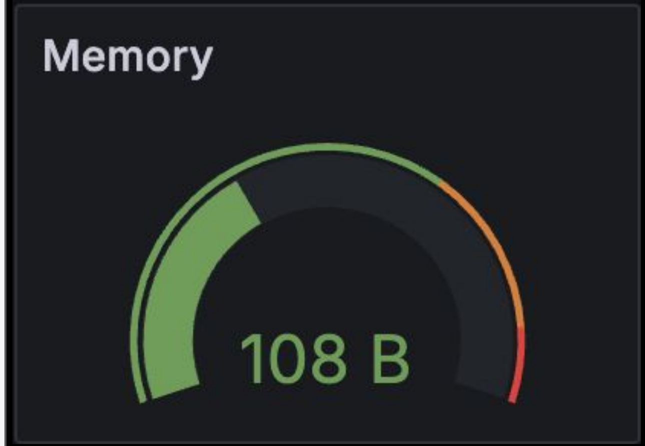
```
http_requests_total{status="200"}
```

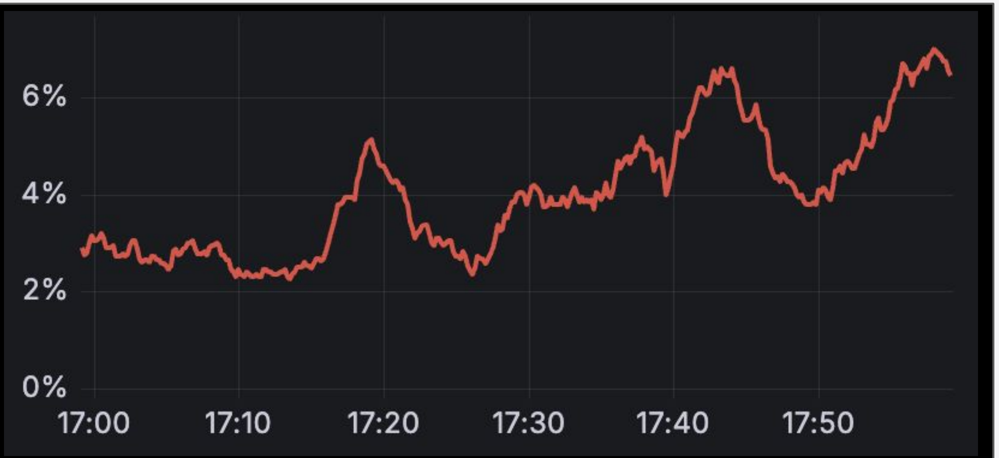


```
{__name__="http_requests_total",status="200"}
```

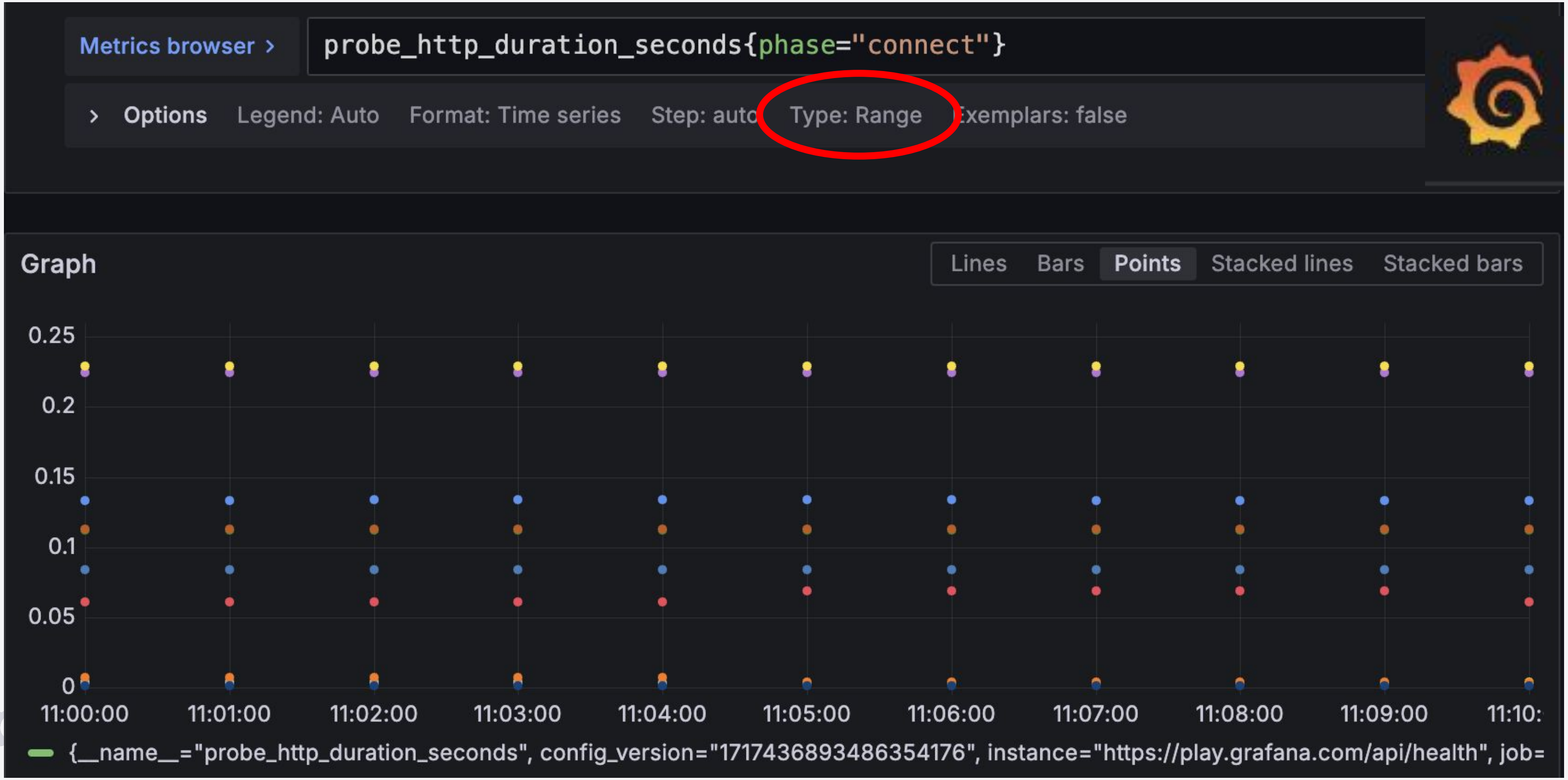


# Instant vs Range Query





# Range Query



Metrics browser >

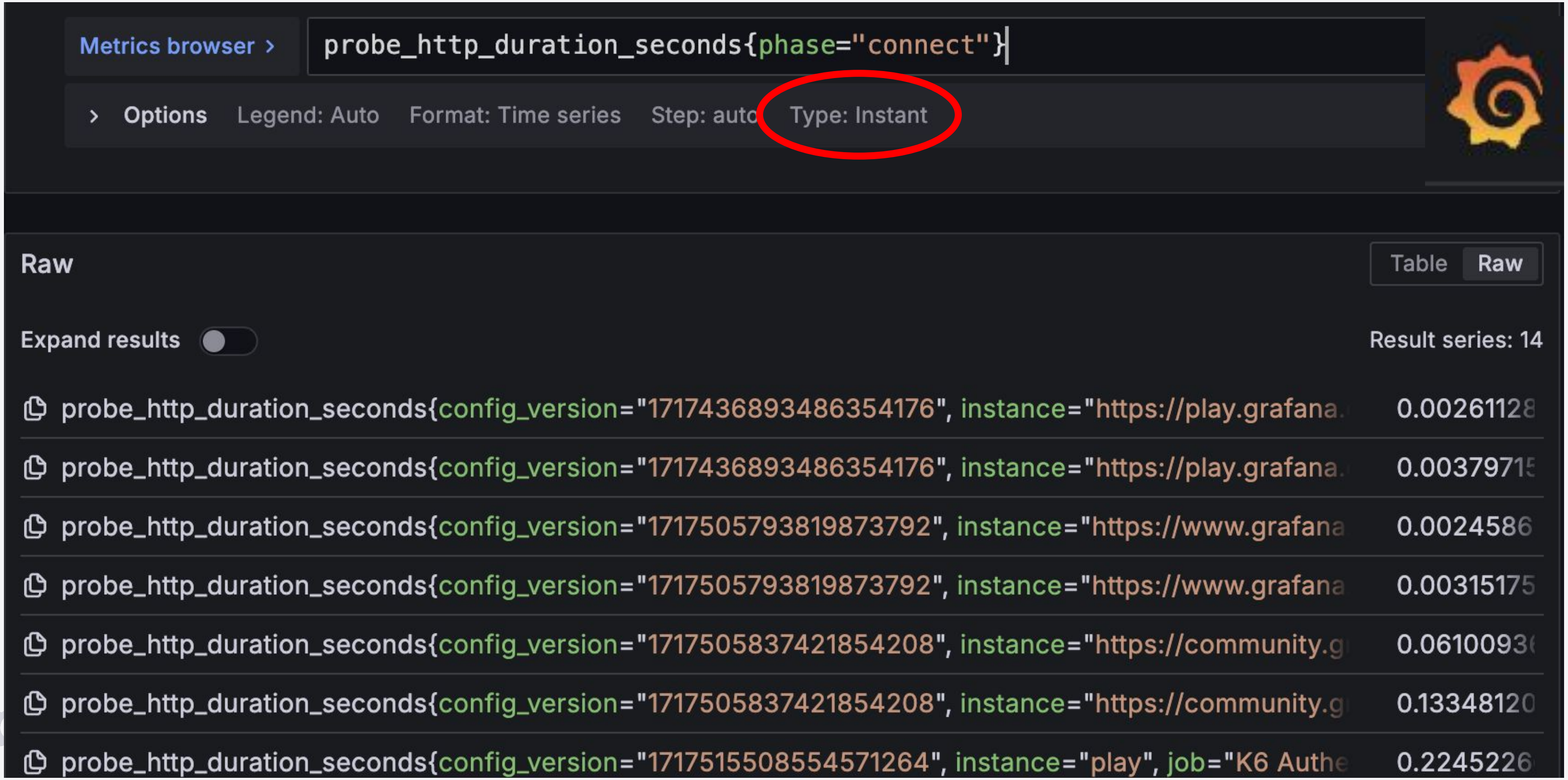
```
probe_http_duration_seconds{phase="connect"}
```

Options  Legend: Auto  Format: Time series  Step: auto  Type: Range  Exemplars: false

Graph

🟩 {__name__="probe_http_duration_seconds", config_version="1717436893486354176", instance="https://play.grafana.com/api/health", job=

## Instant Query



```
probe_http_duration_seconds{phase="connect"}
```

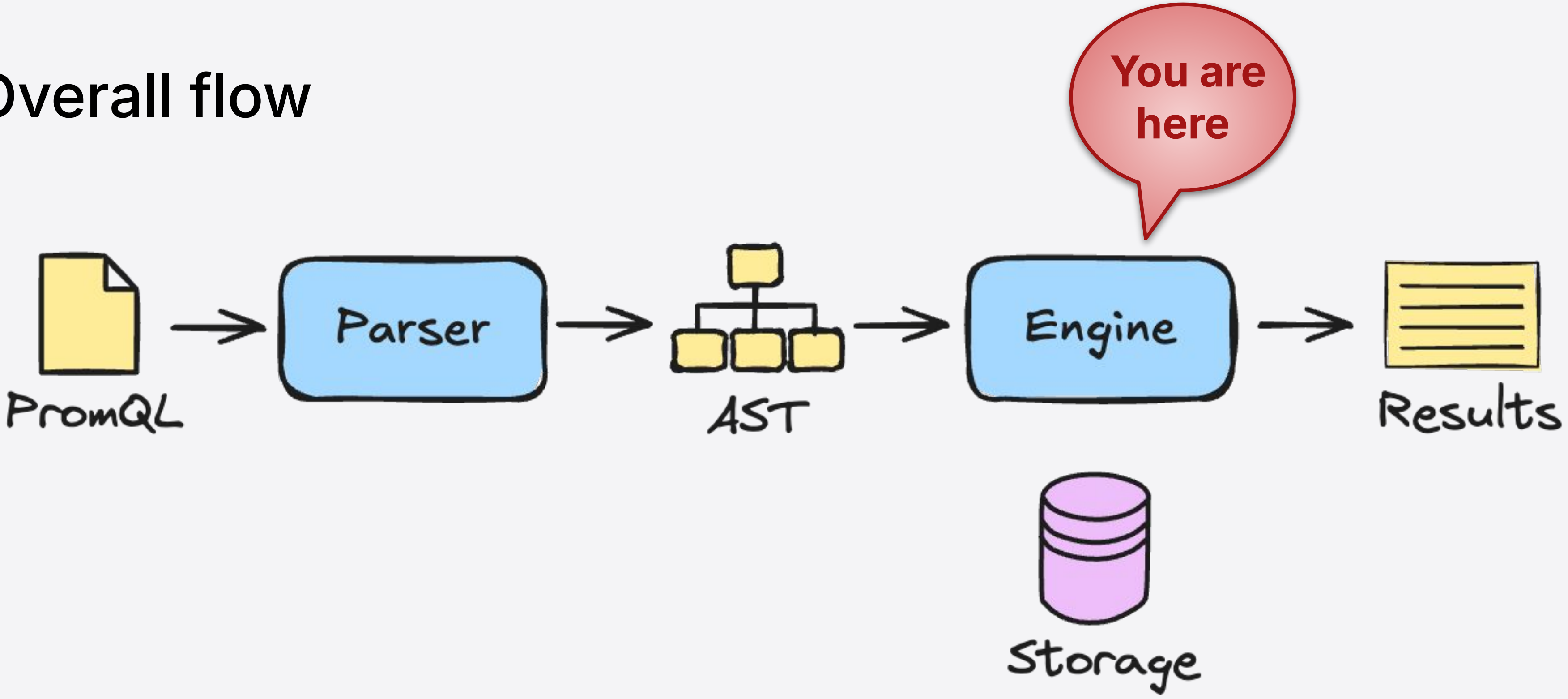
Metrics browser › | Options | Legend: Auto | Format: Time series | Step: auto | Type: Instant

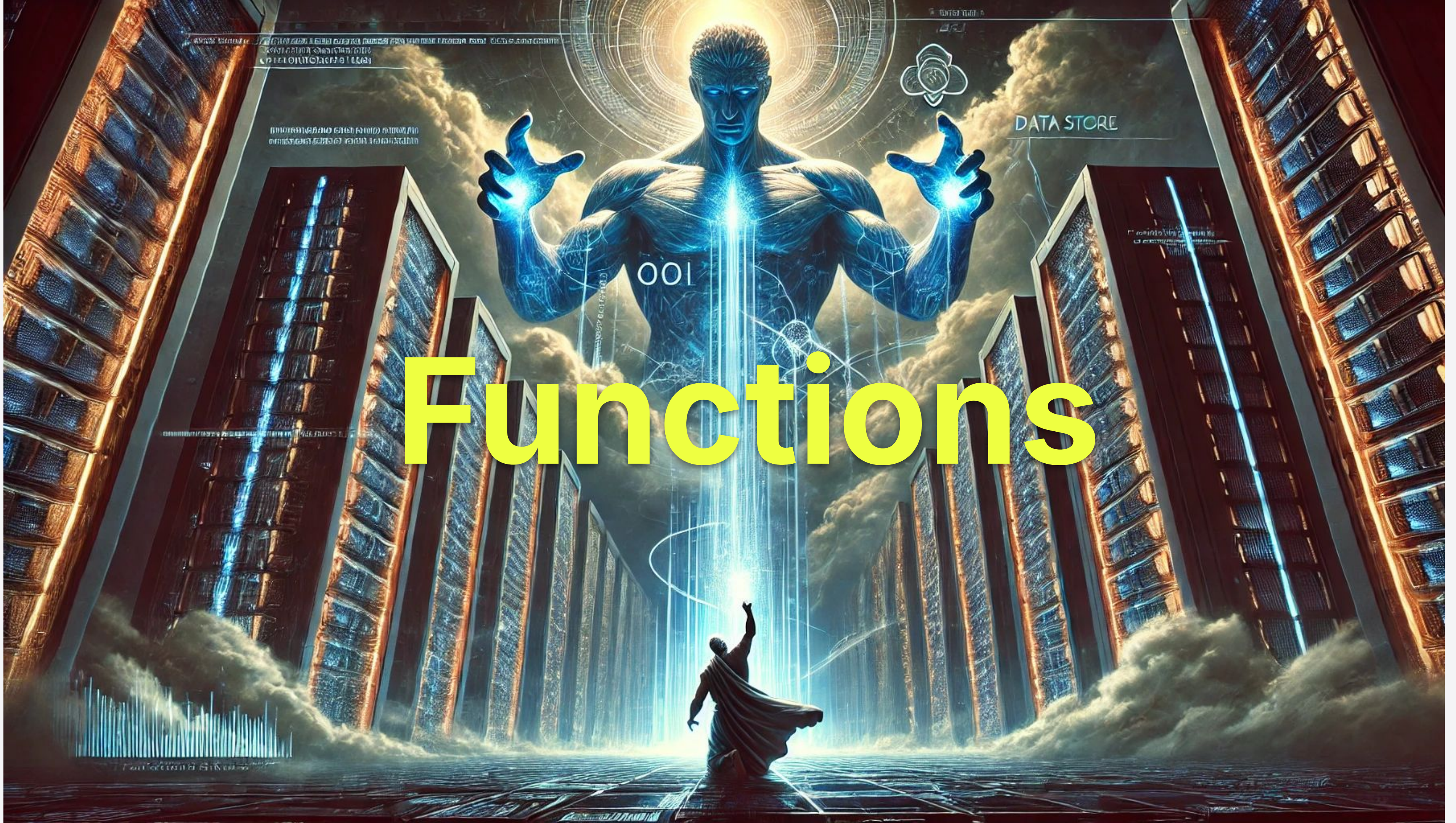
Raw | Table | Raw

Expand results | Result series: 14

```
probe_http_duration_seconds{config_version="1717436893486354176", instance="https://play.grafana    0.00261128
probe_http_duration_seconds{config_version="1717436893486354176", instance="https://play.grafana    0.00379715
probe_http_duration_seconds{config_version="1717505793819873792", instance="https://www.grafana     0.0024586
probe_http_duration_seconds{config_version="1717505793819873792", instance="https://www.grafana     0.00315175
probe_http_duration_seconds{config_version="1717505837421854208", instance="https://community.g     0.0610093
probe_http_duration_seconds{config_version="1717505837421854208", instance="https://community.g     0.13348120
probe_http_duration_seconds{config_version="1717515508554571264", instance="play", job="K6 Authe    0.2245226
```

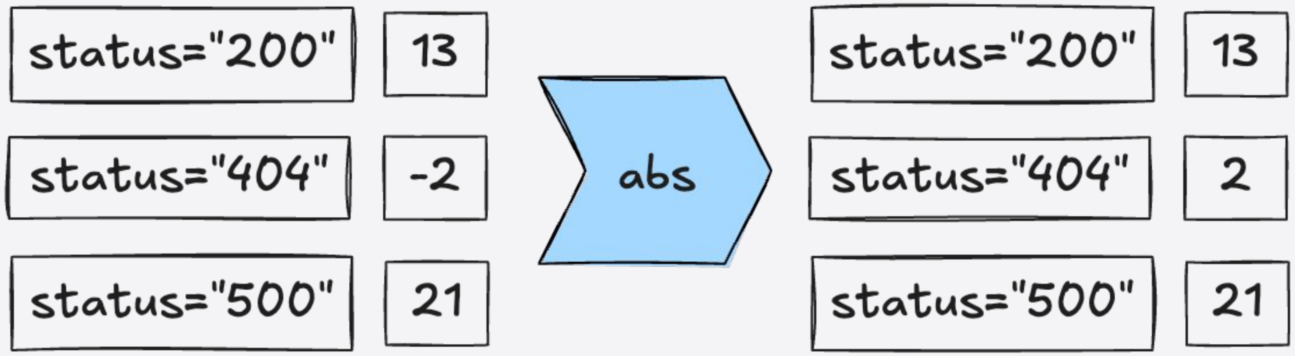
## Overall flow



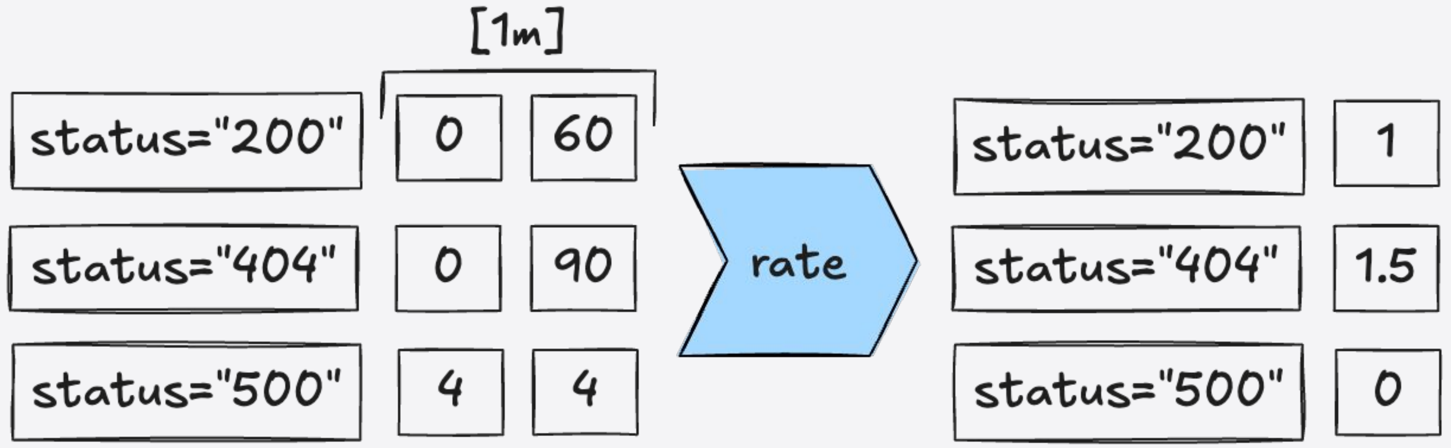


# Functions

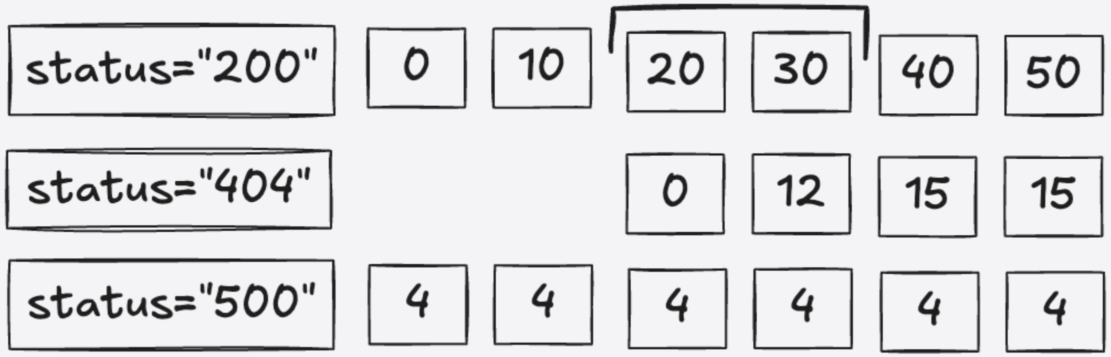
## Calling a function with an instant vector



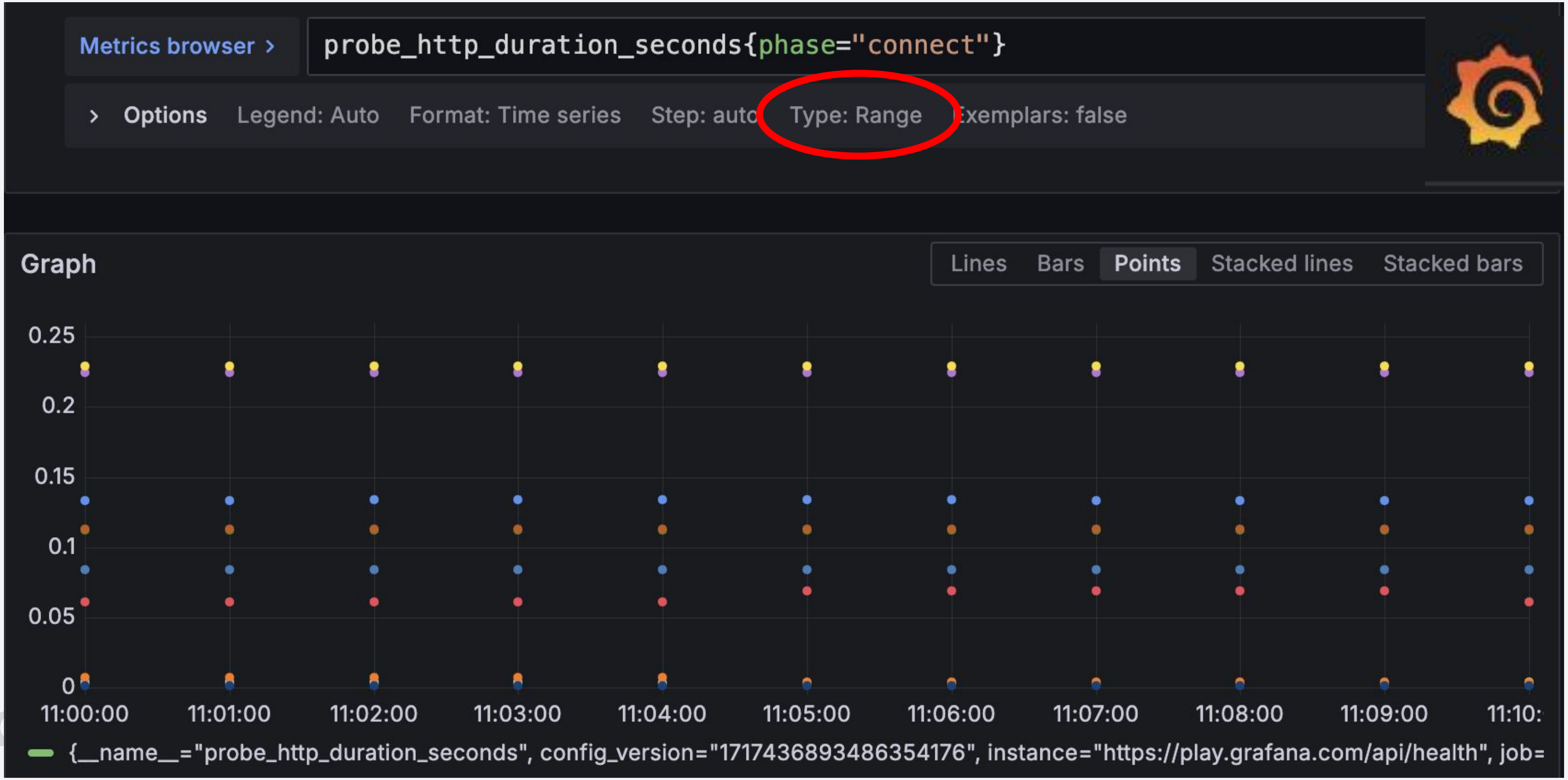
# Calling a function with a range vector



# Range query over a range vector



# Range Query

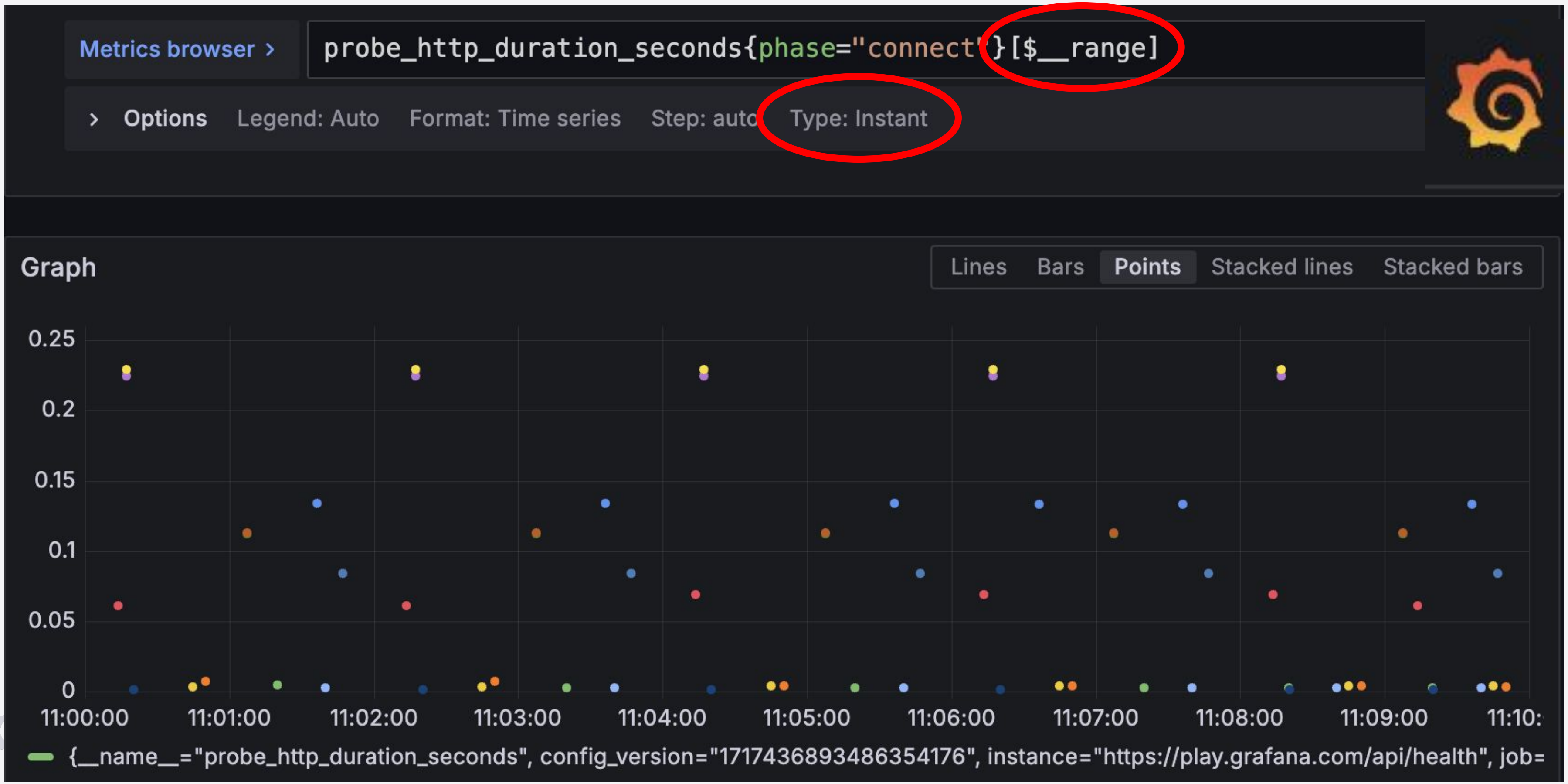


```
probe_http_duration_seconds{phase="connect"}
```

Options Legend: Auto Format: Time series Step: auto Type: Range Exemplars: false

🟩 {__name__="probe_http_duration_seconds", config_version="1717436893486354176", instance="https://play.grafana.com/api/health", job=

# Instant Query with Range



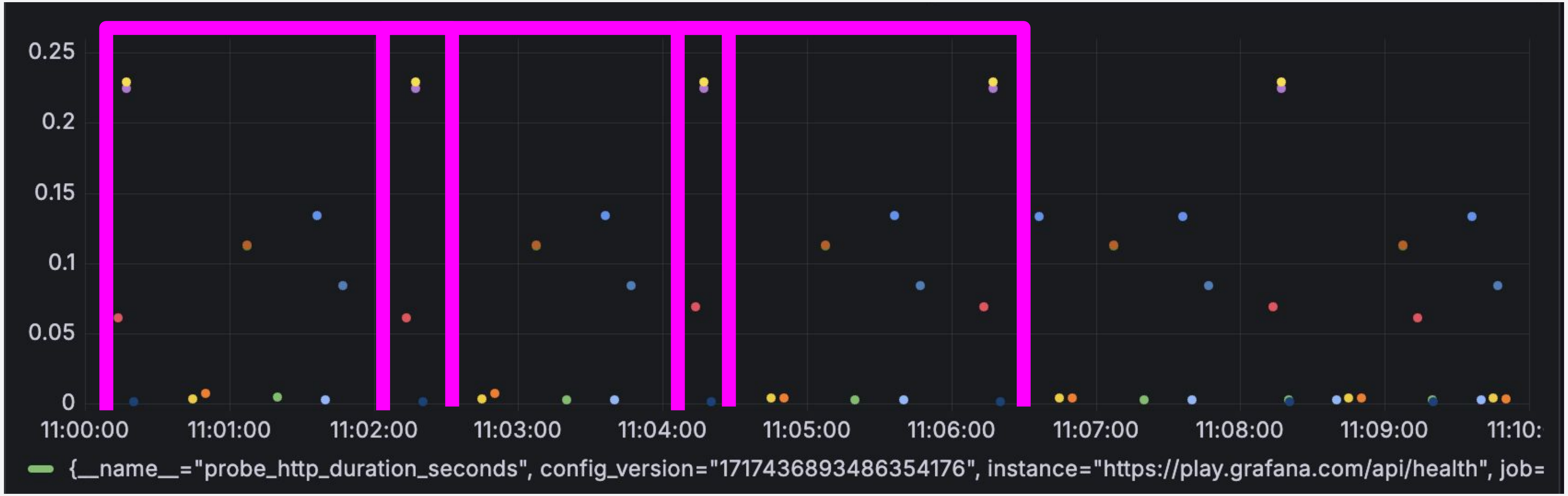
```
probe_http_duration_seconds{phase="connect"}[$__range]
```

› Options Legend: Auto Format: Time series Step: auto Type: Instant

🟩 {__name__="probe_http_duration_seconds", config_version="1717436893486354176", instance="https://play.grafana.com/api/health", job=

## Use an Instant Query (with range) to see underlying data points

# Range query over a range vector

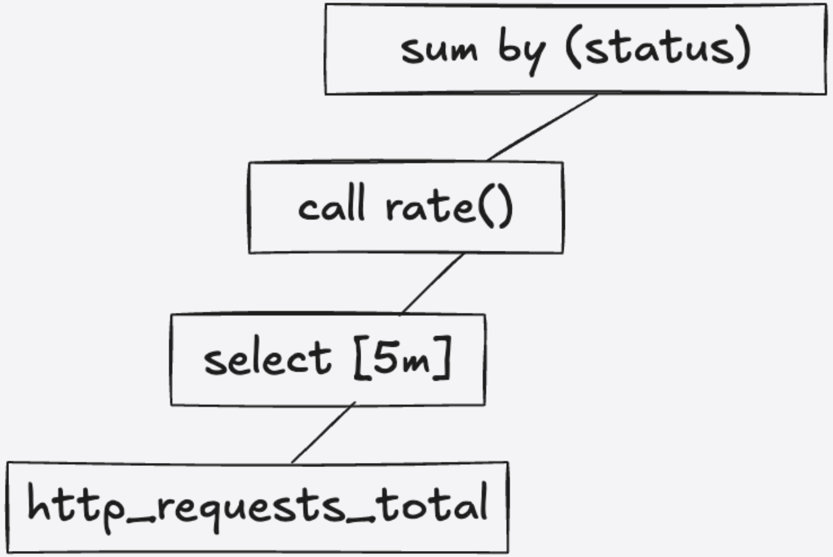


{__name__="probe_http_duration_seconds", config_version="1717436893486354176", instance="https://play.grafana.com/api/health", job=

## Aggregations

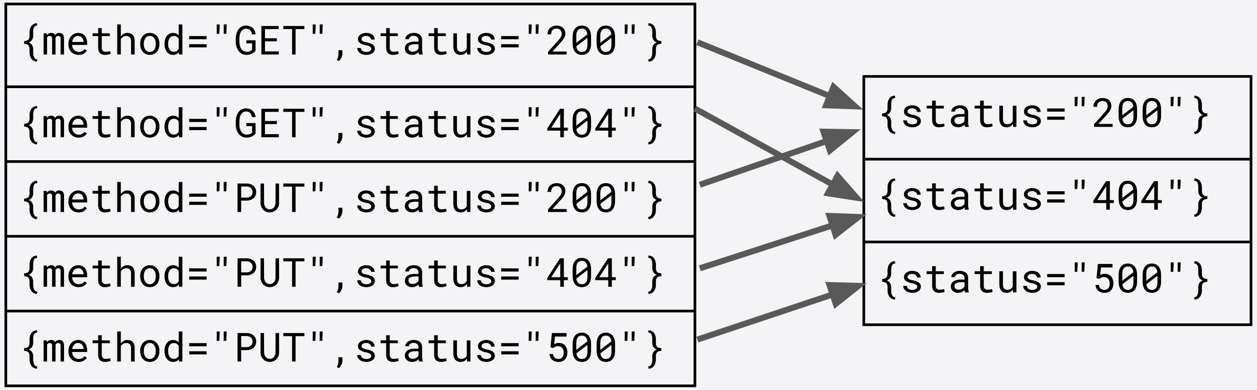
## Aggregations: sum

```
sum by (status) (rate(http_requests_total[5m]))
```



## Aggregations: sum

```
sum by (status) (rate(http_requests_total[5m]))
```



{method="GET",status="200"}
{method="GET",status="404"}
{method="PUT",status="200"}
{method="PUT",status="404"}
{method="PUT",status="500"}
{status="200"}
{status="404"}
{status="500"}

## Aggregations: topk

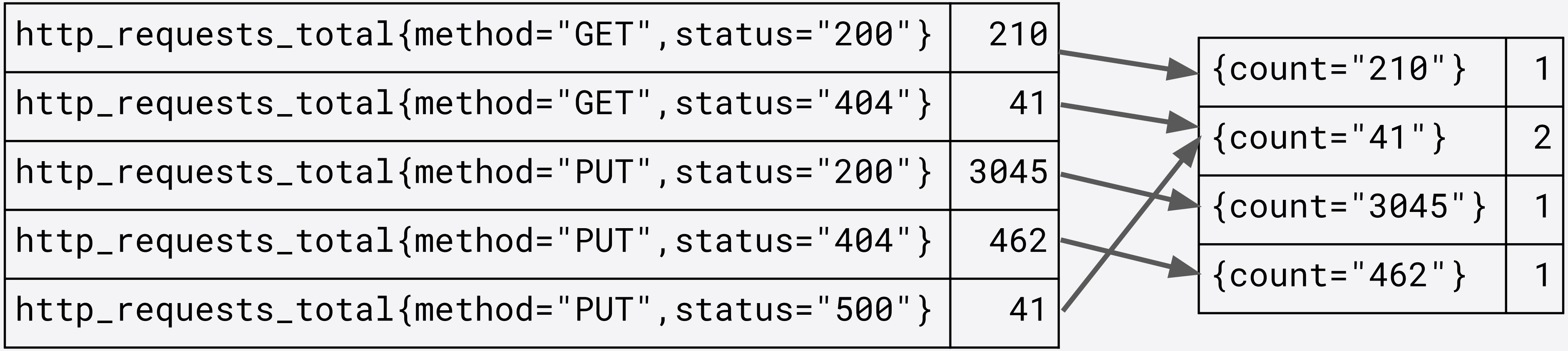
```
topk(2, http_requests_total)
```

| | | |
|---|---|---|
| | http_requests_total{method="GET",status="200"} | 210 |
| | http_requests_total{method="GET",status="404"} | 41 |
| → | http_requests_total{method="PUT",status="200"} | 3045 |
| → | http_requests_total{method="PUT",status="404"} | 462 |
| | http_requests_total{method="PUT",status="500"} | 41 |

| |
|---|
| 462 |
| 3045 |

## Aggregations: count_values

```
count_values("count",http_requests_total)
```



| | |
|---|---|
| `http_requests_total{method="GET",status="200"}` | 210 |
| `http_requests_total{method="GET",status="404"}` | 41 |
| `http_requests_total{method="PUT",status="200"}` | 3045 |
| `http_requests_total{method="PUT",status="404"}` | 462 |
| `http_requests_total{method="PUT",status="500"}` | 41 |

| | |
|---|---|
| `{count="210"}` | 1 |
| `{count="41"}` | 2 |
| `{count="3045"}` | 1 |
| `{count="462"}` | 1 |

# Operators

## Operators - matching all labels

`mem_total_mb - mem_free_mb`

| | |
|---|---|
| mem_total_mb{host="a1"} | 1024 |
| mem_total_mb{host="a2"} | 1024 |
| mem_total_mb{host="b3"} | 2048 |

| | |
|---|---|
| mem_free_mb{host="a1"} | 640 |
| mem_free_mb{host="a2"} | 0 |
| mem_free_mb{host="b3"} | 80 |

| | |
|---|---|
| {host="a1"} | 384 |
| {host="a2"} | 1024 |
| {host="b3"} | 1968 |

## Operators - 'info' series

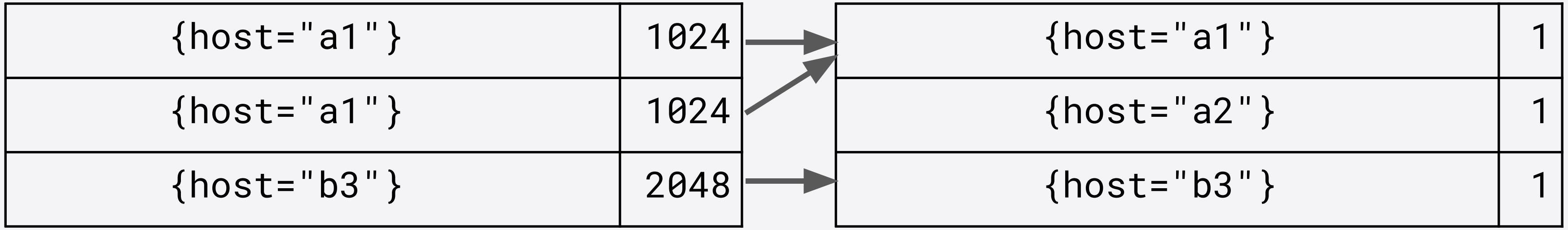
```
disk_mb * on (host) group_left(team) host_info
```

| | |
|---|---|
| disk_mb{host="a1",disk="x"} | 1024 |
| disk_mb{host="a1",disk="y"} | 1024 |
| disk_mb{host="b3",disk="x"} | 2048 |

| | |
|---|---|
| host_info{host="a1",team="a"} | 1 |
| host_info{host="a2",team="a"} | 1 |
| host_info{host="b3",team="b"} | 1 |

## Operator matching via 'signatures'

```
disk_mb * on (host) group_left(team) host_info
```



| | |
|---|---|
| {host="a1"} | 1024 |
| {host="a1"} | 1024 |
| {host="b3"} | 2048 |

| | |
|---|---|
| {host="a1"} | 1 |
| {host="a2"} | 1 |
| {host="b3"} | 1 |

## Operators - 'info' series

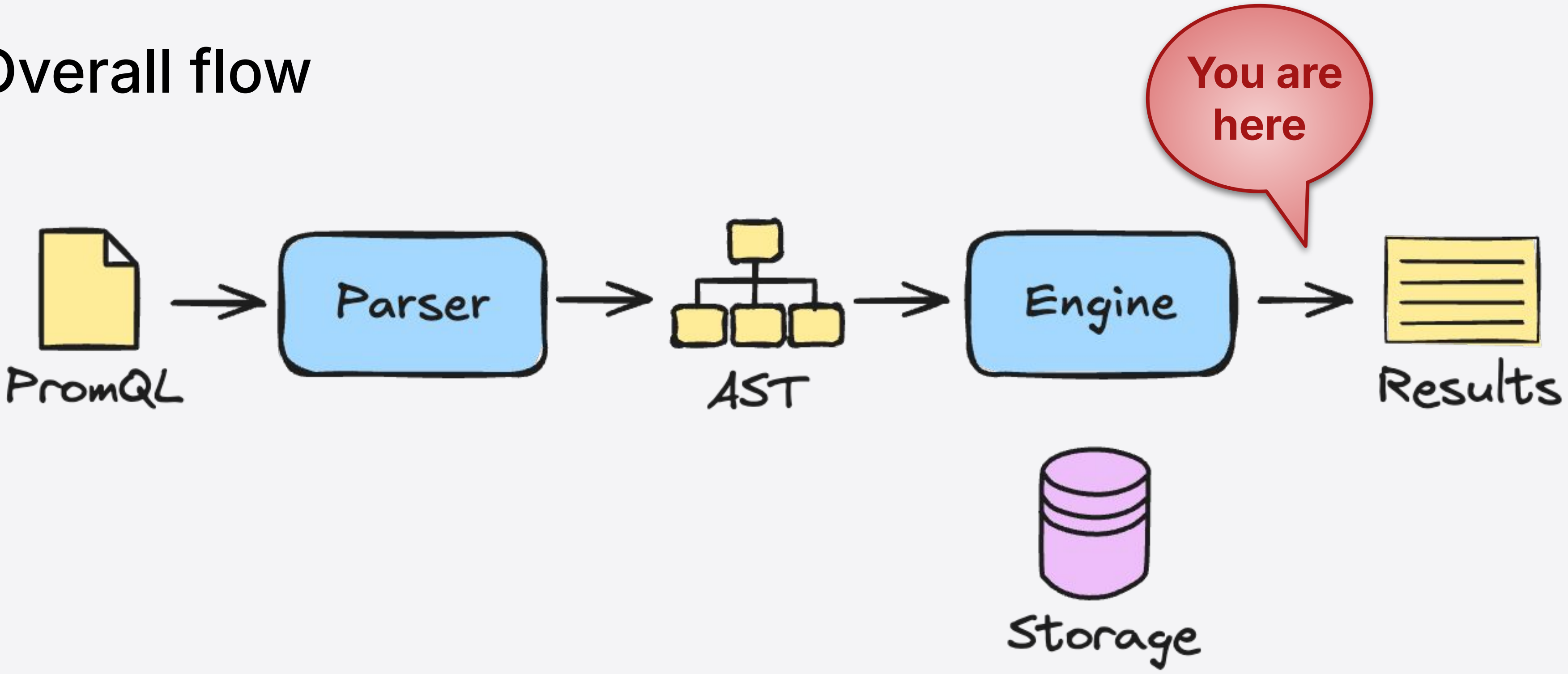
`disk_mb * on (host) group_left(team) host_info`

| | |
|---|---|
| disk_mb{host="a1",disk="x"} | 1024 |
| disk_mb{host="a1",disk="y"} | 1024 |
| disk_mb{host="b3",disk="x"} | 2048 |

| | |
|---|---|
| host_info{host="a1",team="a"} | 1 |
| host_info{host="a2",team="a"} | 1 |
| host_info{host="b3",team="b"} | 1 |

| | |
|---|---|
| {disk="x",host="a1",team="a"} | 1024 |
| {disk="y",host="a1",team="a"} | 1024 |
| {disk="x",host="b3",team="b"} | 2048 |

## Overall flow



## Final results

| | | | |
|---|---|---|---|
| {disk="x",host="a1",team="a"} | 1024 | 906 | 878 |
| {disk="y",host="a1",team="a"} | 1024 | 1102 | 1184 |
| {disk="x",host="b3",team="b"} | 2048 | 2048 | 200 |

## Final results, sorted

| | | | |
|---|---|---|---|
| {disk="x",host="a1",team="a"} | 1024 | 906 | 878 |
| {disk="x",host="b3",team="b"} | 2048 | 2048 | 200 |
| {disk="y",host="a1",team="a"} | 1024 | 1102 | 1184 |

Grafana Labs

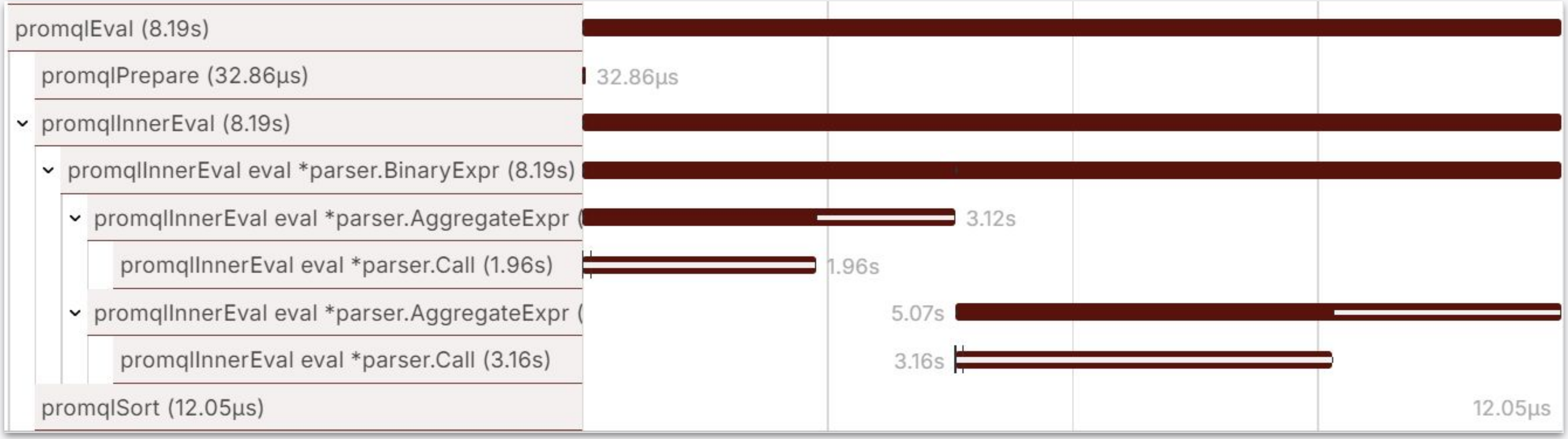
# Thank you

@bboreham

## How do we find out what the PromQL Engine is doing?

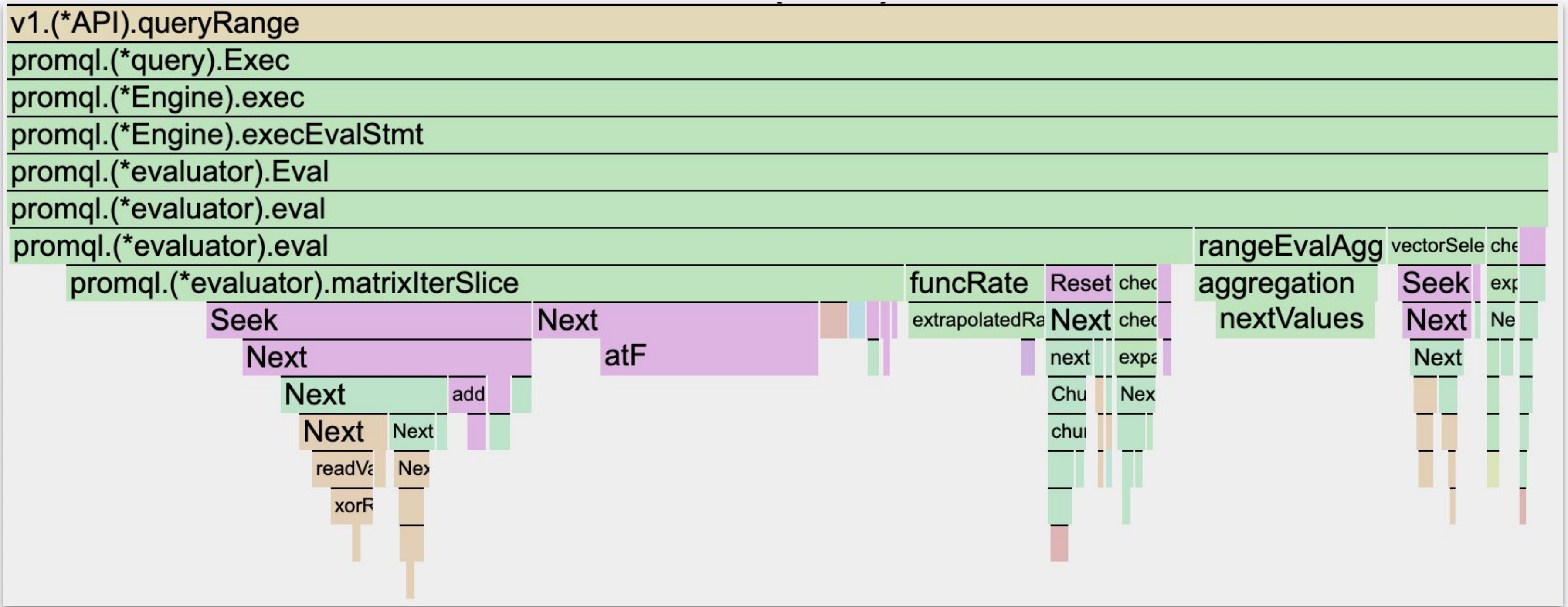
# Distributed Tracing

```
sum by(job, mode) (rate(node_cpu_seconds_total[1m]))
 / on(job) group_left sum by(job)(rate(node_cpu_seconds_total[1m]))
```



```
promqlEval (8.19s)
  promqlPrepare (32.86µs)                                32.86µs
⌄ promqlInnerEval (8.19s)
  ⌄ promqlInnerEval eval *parser.BinaryExpr (8.19s)
    ⌄ promqlInnerEval eval *parser.AggregateExpr (        3.12s
        promqlInnerEval eval *parser.Call (1.96s)         1.96s
    ⌄ promqlInnerEval eval *parser.AggregateExpr (        5.07s
        promqlInnerEval eval *parser.Call (3.16s)         3.16s
  promqlSort (12.05µs)                                    12.05µs
```

# Profile


v1.(*API).queryRange
promql.(*query).Exec
promql.(*Engine).exec
promql.(*Engine).execEvalStmt
promql.(*evaluator).Eval
promql.(*evaluator).eval
promql.(*evaluator).eval
rangeEvalAgg
vectorSele
che
promql.(*evaluator).matrixIterSlice
funcRate
Reset
chec
aggregation
Seek
exp
Seek
Next
extrapolatedRa
Next
chec
nextValues
Next
Ne
Next
atF
next
expa
Next
Next
add
Chu
Nex
Next
Next
chui
readVa
Nex
xorR

# Code scale

```
  3660 promql/engine.go
  1869 promql/functions.go
   464 promql/quantile.go
   535 promql/value.go
   993 promql/parser/parser.y
  1046 promql/parser/parse.go
  1071 promql/parser/lex.go
 14992 total
```

```
  3480 promql/engine_test.go
  4497 promql/parser/parse_test.go
  1469 promql/promqltest/test.go
   489 promql/operators.test
   510 promql/histograms.test
   574 promql/aggregators.test
   970 promql/native_histograms.test
  1236 promql/functions.test
 17015 total
```

## There are other PromQL Engines

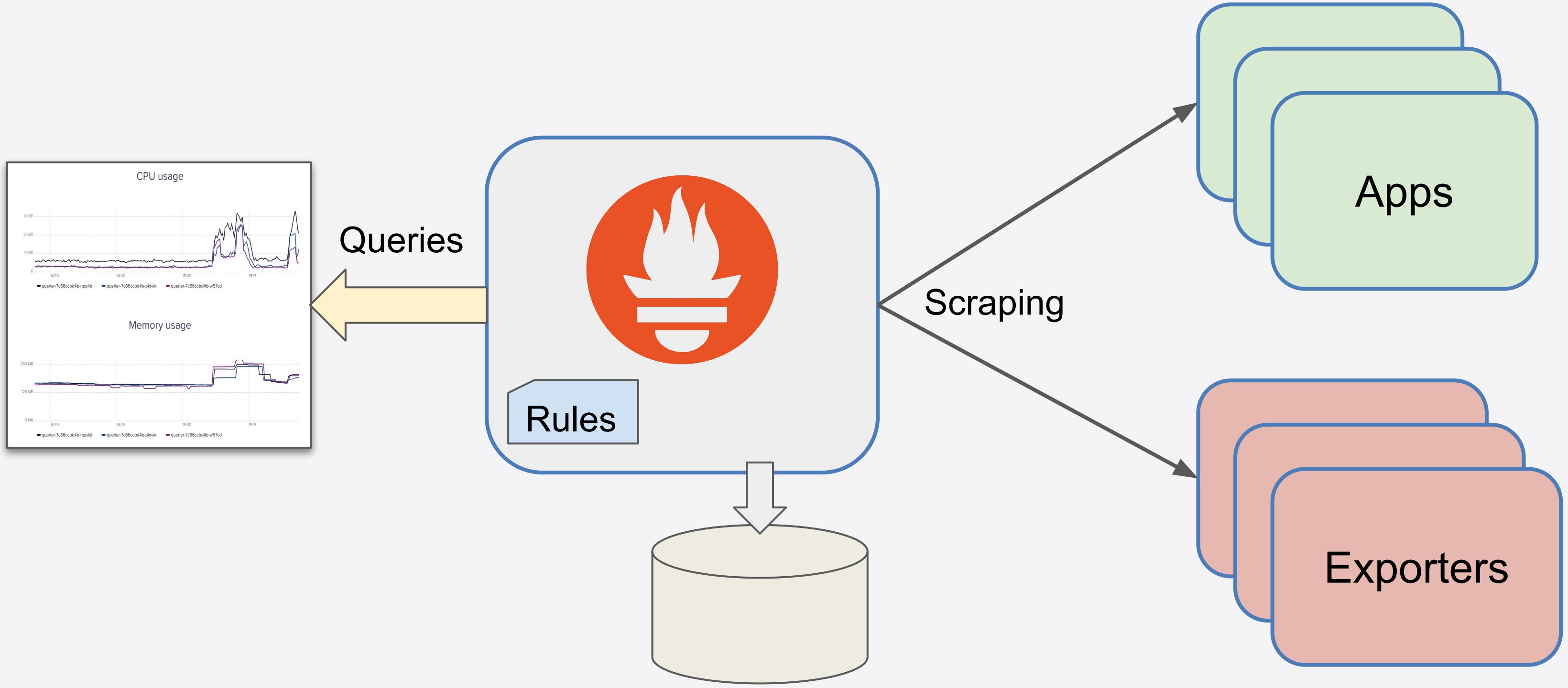
In Thanos

In Mimir

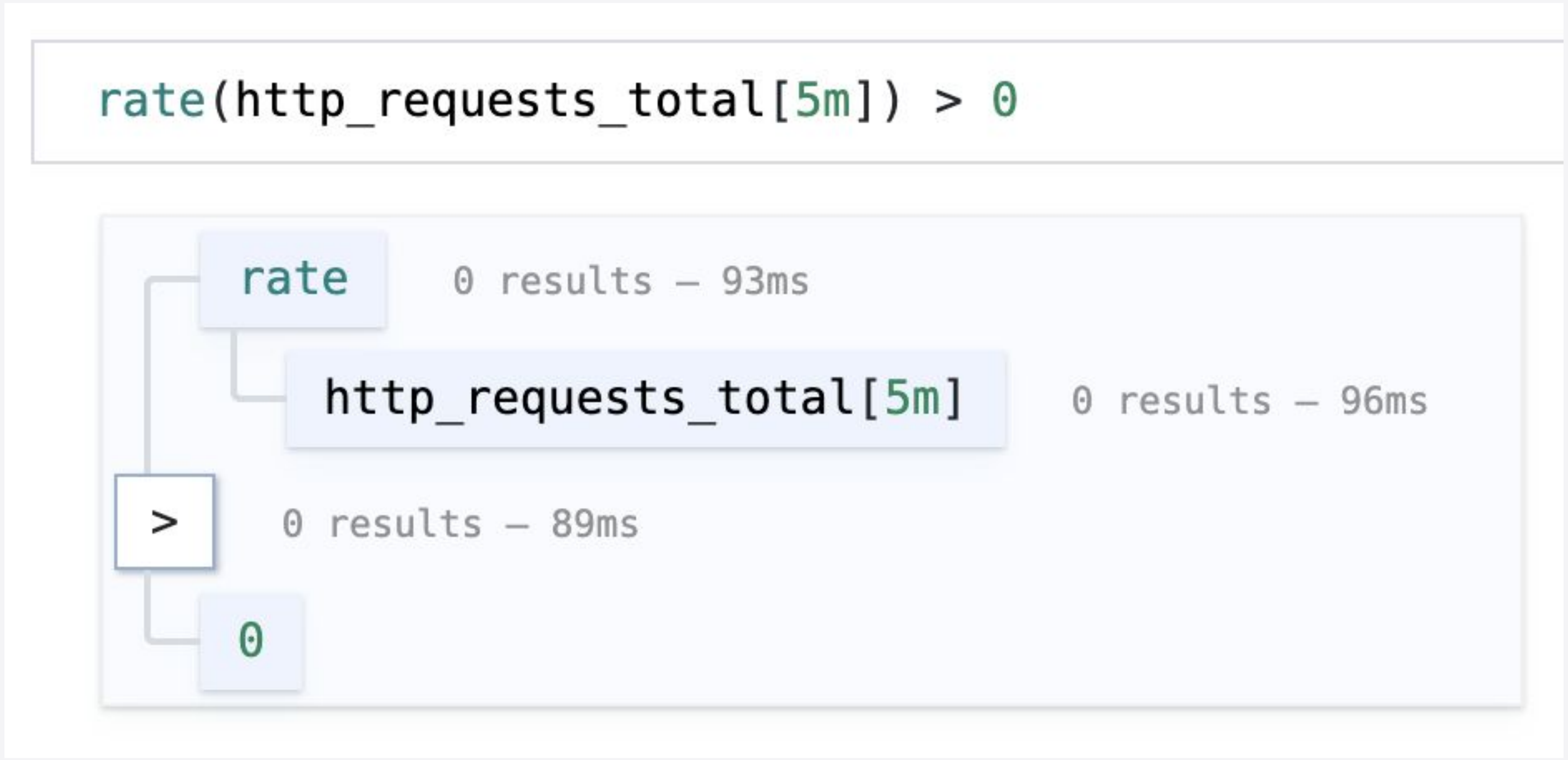
Maybe other projects

But I am going to talk about OG Prometheus

## Big picture again



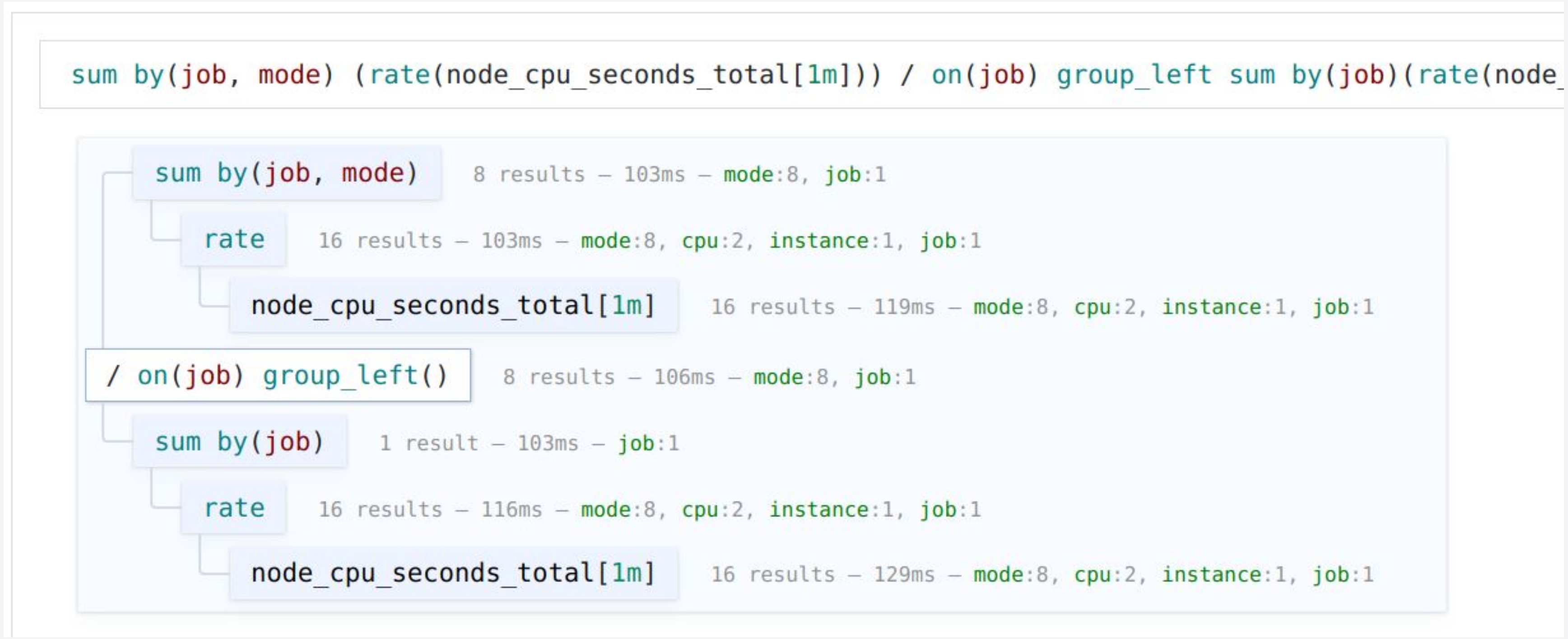
## PromLens



```
rate(http_requests_total[5m]) > 0
```

```
rate    0 results – 93ms
  http_requests_total[5m]    0 results – 96ms
>    0 results – 89ms
0
```

# PromLens



```
sum by(job, mode) (rate(node_cpu_seconds_total[1m])) / on(job) group_left sum by(job)(rate(node_
```

```
sum by(job, mode)                  8 results – 103ms – mode:8, job:1
    rate                           16 results – 103ms – mode:8, cpu:2, instance:1, job:1
        node_cpu_seconds_total[1m]   16 results – 119ms – mode:8, cpu:2, instance:1, job:1
/ on(job) group_left()             8 results – 106ms – mode:8, job:1
sum by(job)                        1 result – 103ms – job:1
    rate                           16 results – 116ms – mode:8, cpu:2, instance:1, job:1
        node_cpu_seconds_total[1m]   16 results – 129ms – mode:8, cpu:2, instance:1, job:1
```