

Getting Started with Prometheus

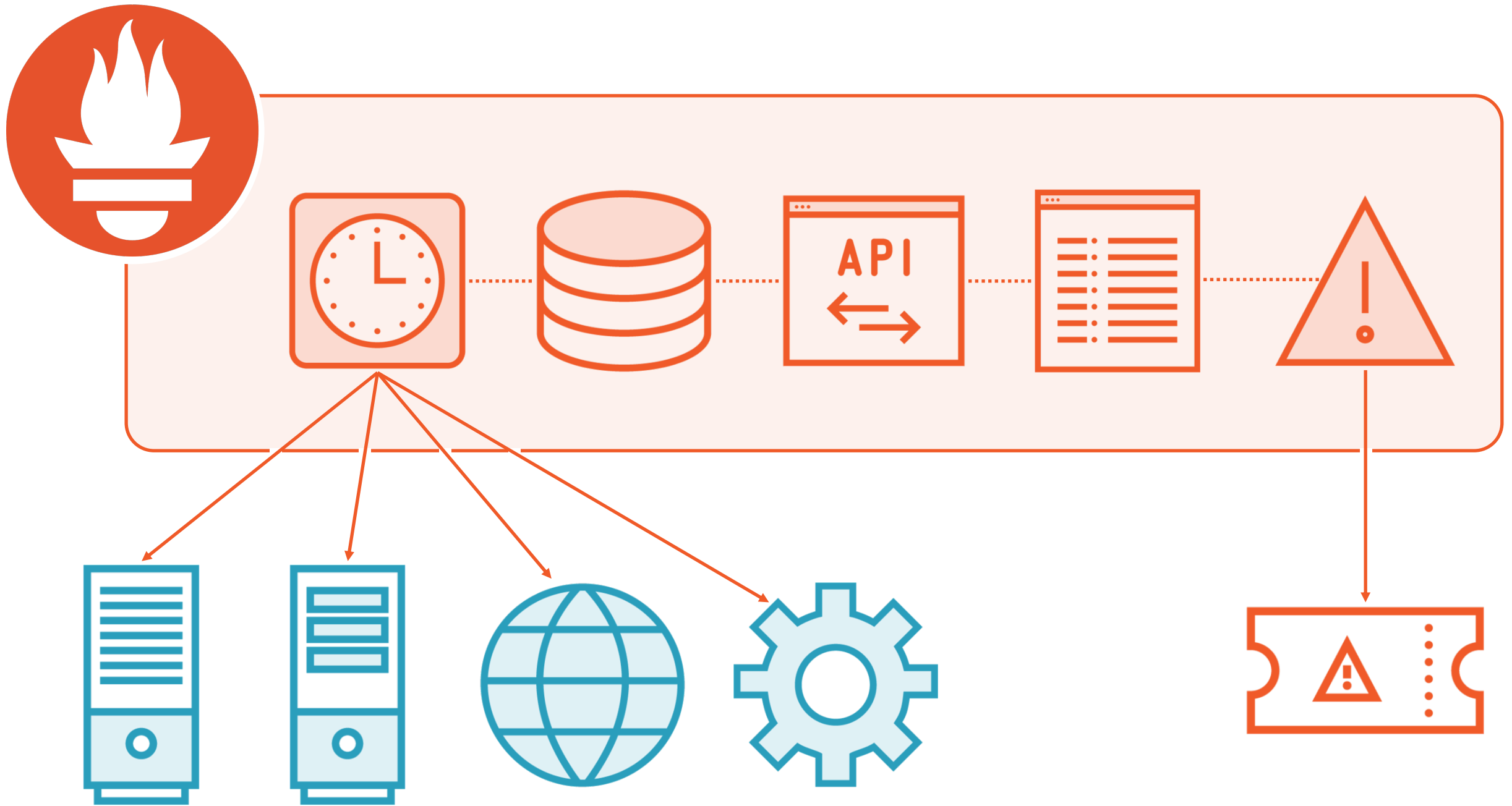
UNDERSTANDING HOW PROMETHEUS WORKS



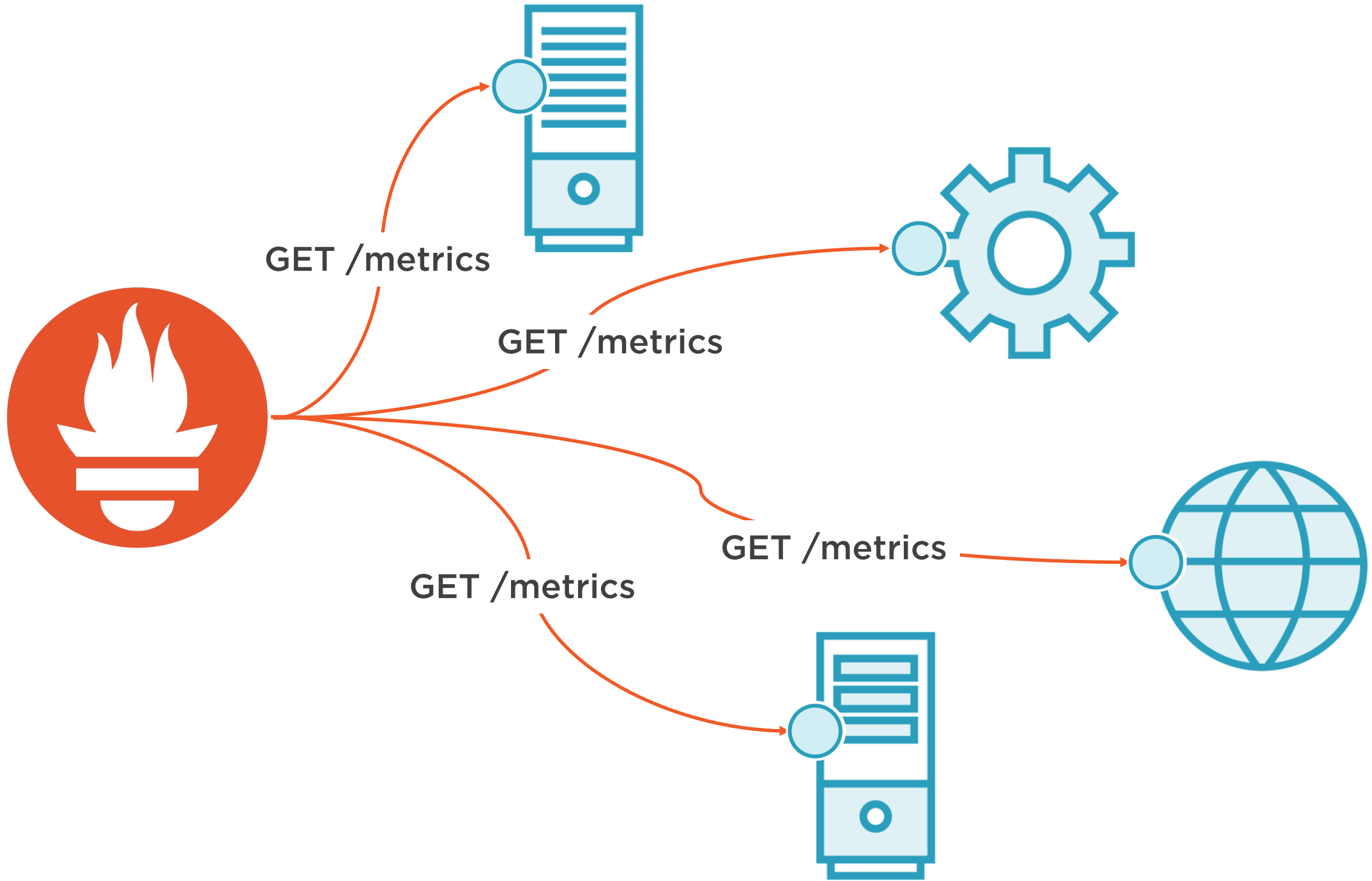
Elton Stoneman

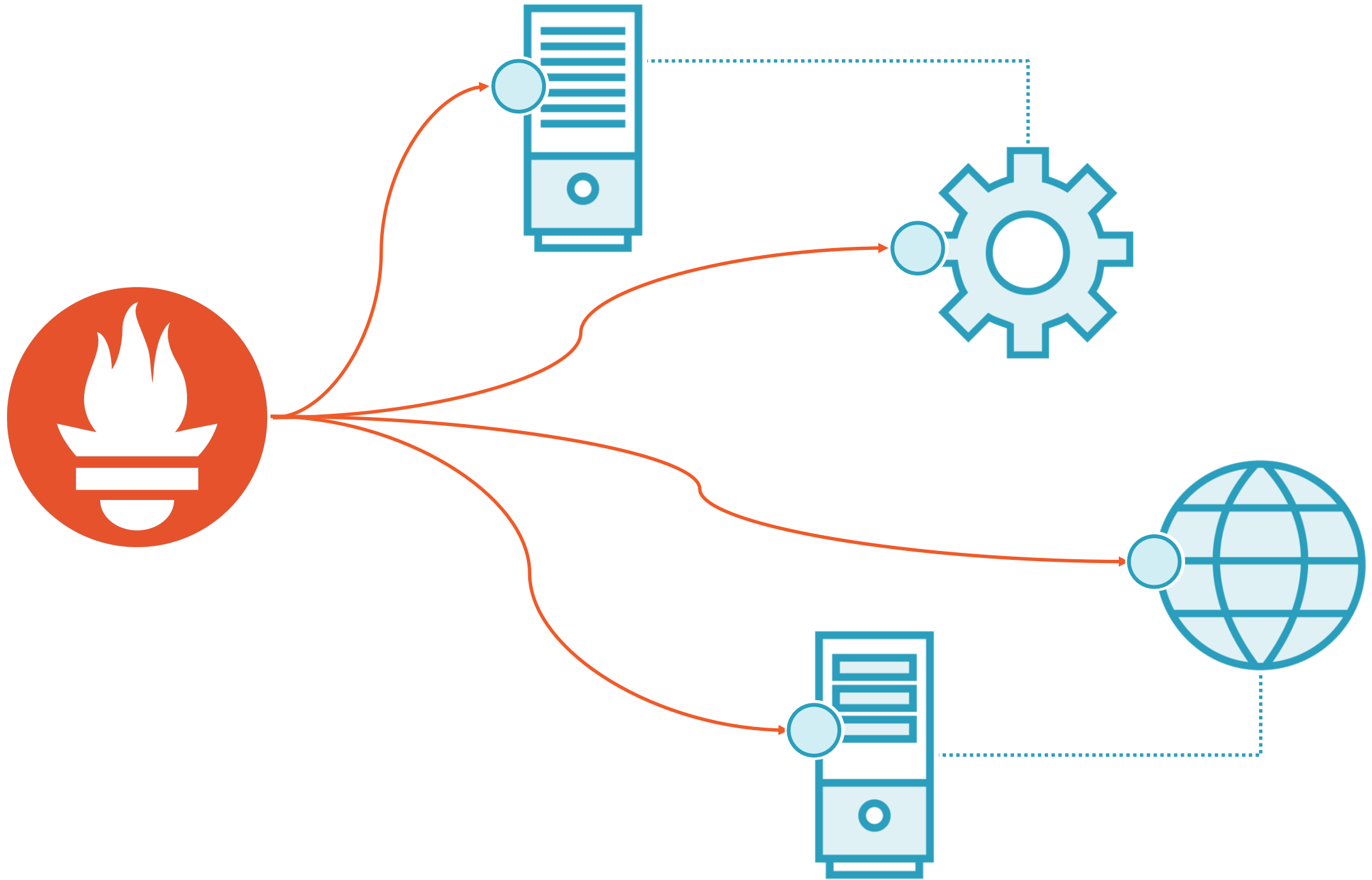
CONSULTANT & TRAINER

@EltonStoneman | blog.sixeyed.com

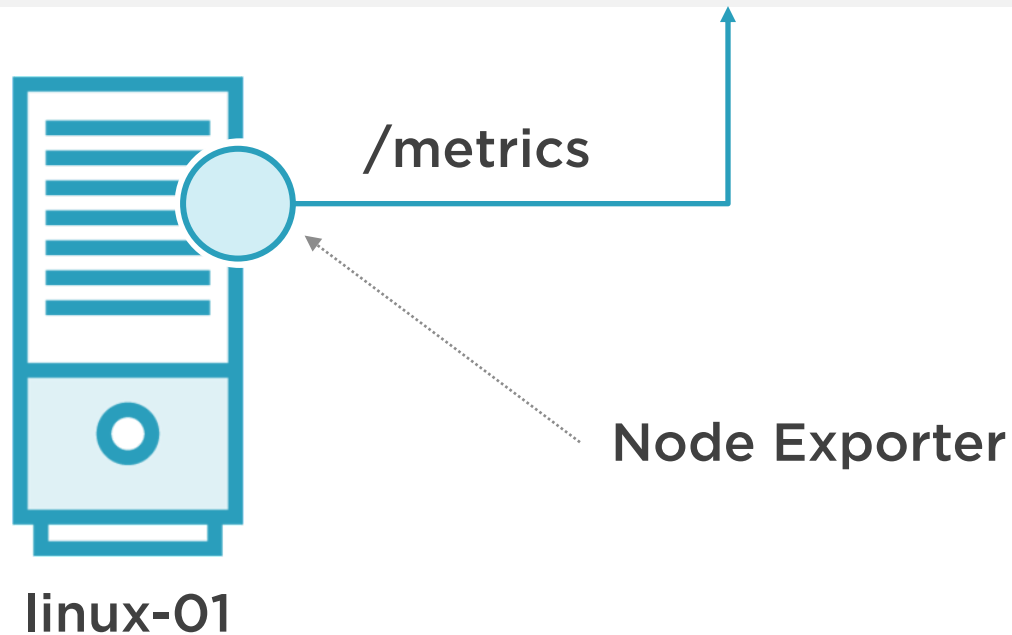


What Makes Prometheus so Awesome

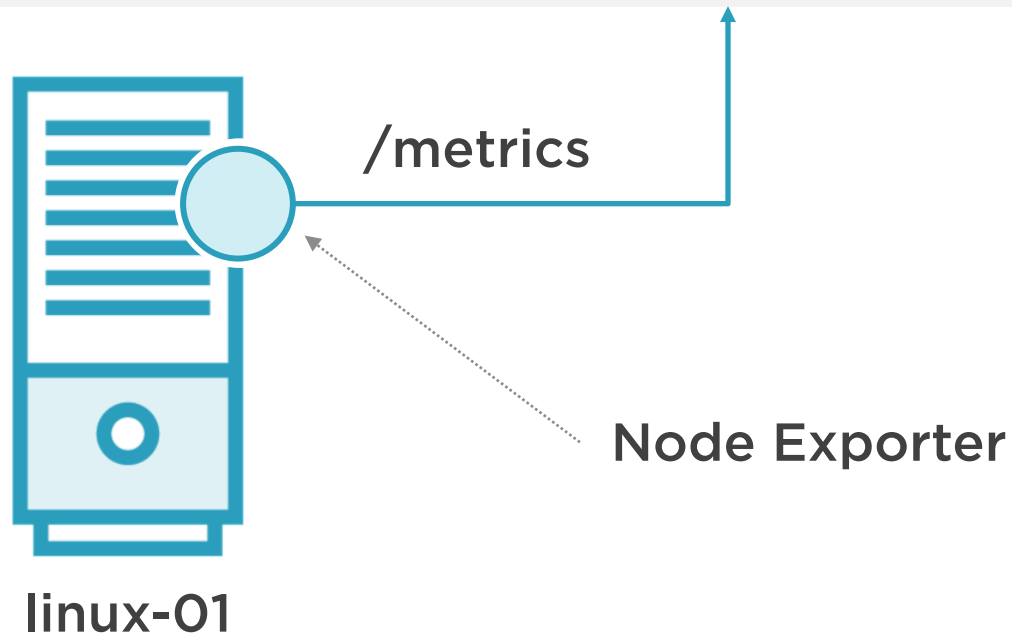




```
# HELP node_filefd_allocated File descriptor statistics: allocated.  
# TYPE node_filefd_allocated gauge  
node_filefd_allocated 1184  
  
# HELP node_disk_io_time_seconds_total Total seconds spent doing I/Os.  
# TYPE node_disk_io_time_seconds_total counter  
node_disk_io_time_seconds_total{device="sda"} 104.296
```

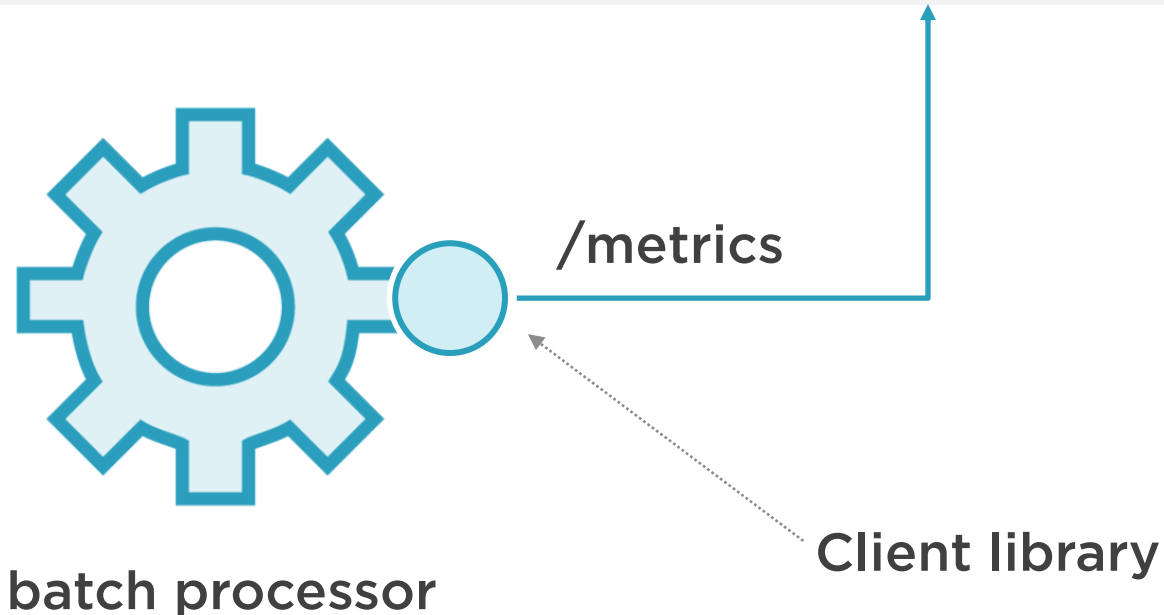


```
# HELP node_filefd_allocated File descriptor statistics: allocated.  
# TYPE node_filefd_allocated gauge  
node_filefd_allocated 1184  
  
# HELP node_disk_io_time_seconds_total Total seconds spent doing I/Os.  
# TYPE node_disk_io_time_seconds_total counter  
node_disk_io_time_seconds_total{device="sda"} 104.296
```



```
# HELP process_cpu_seconds_total Total user and system CPU time.  
# TYPE process_cpu_seconds_total counter  
process_cpu_seconds_total 5.23
```

```
# HELP worker_jobs_total Worker jobs handled  
# TYPE worker_jobs_total counter  
worker_jobs_total{status="processed"} 1570222  
worker_jobs_total{status="failed"} 159665
```



HELP process_cpu_seconds_total Total user and system CPU time.

TYPE process_cpu_seconds_total counter

process_cpu_seconds_total 5.23

HELP worker_jobs_total Worker jobs handled

TYPE worker_jobs_total counter

worker_jobs_total{status="processed"} 1570222

worker_jobs_total{status="failed"} 159665

HELP node_filefd_allocated File descriptor statistics: allocated.

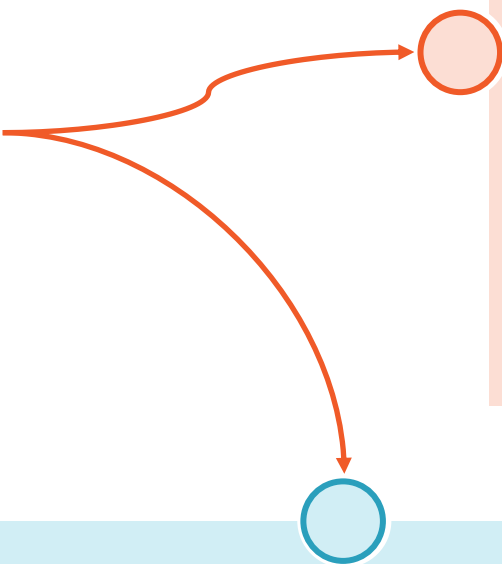
TYPE node_filefd_allocated gauge

node_filefd_allocated 1184

HELP node_disk_io_time_seconds_total Total seconds spent doing I/Os.

TYPE node_disk_io_time_seconds_total counter

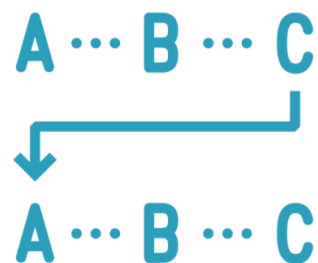
node_disk_io_time_seconds_total{device="sda"} 104.296

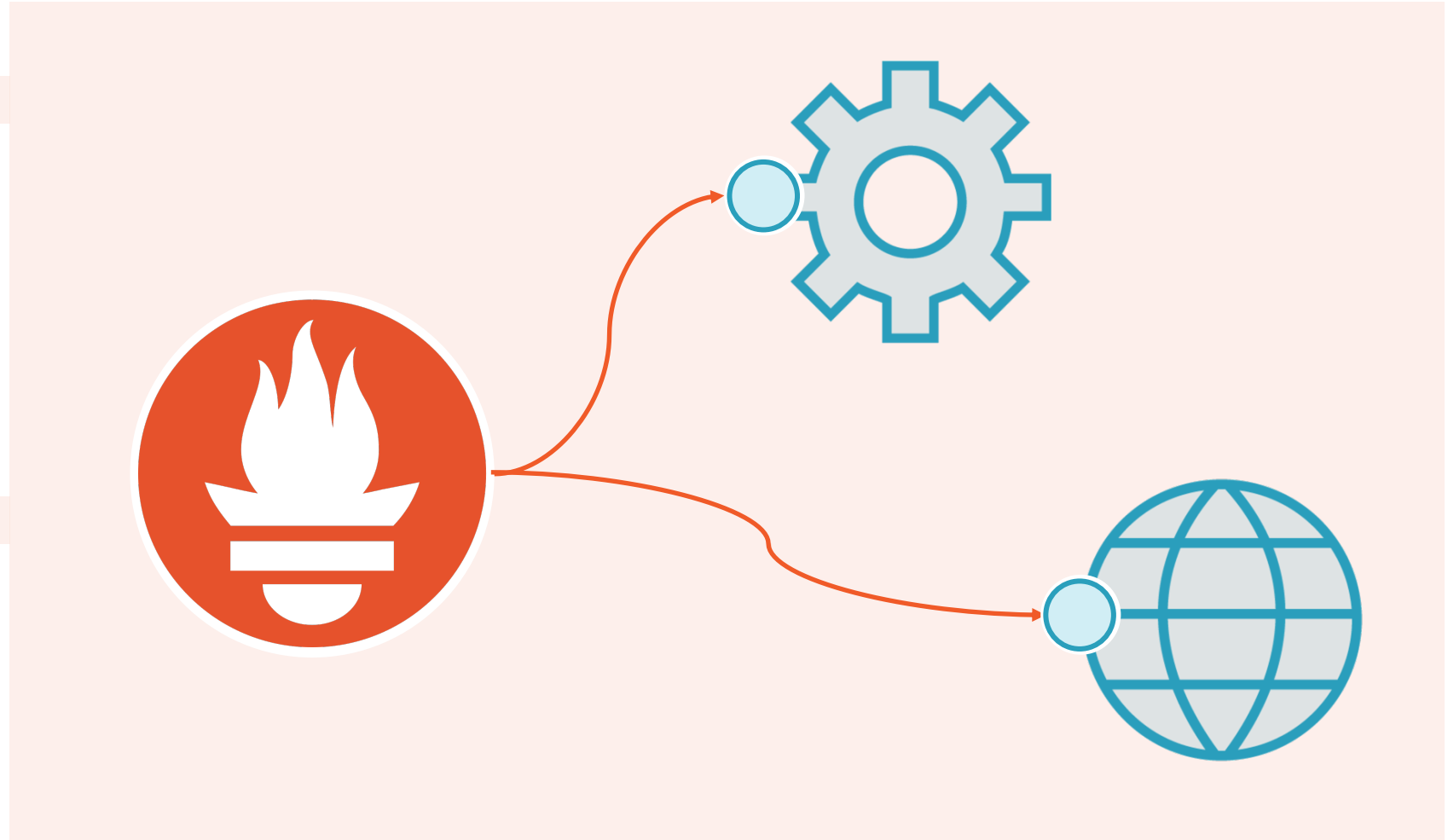
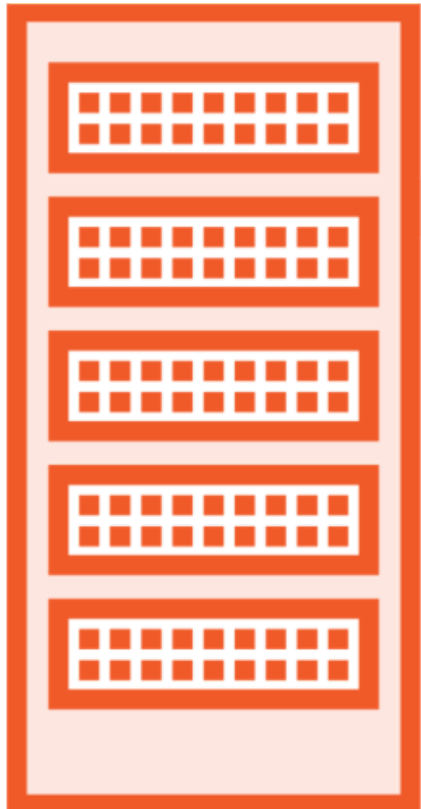


Client libraries



Exporters



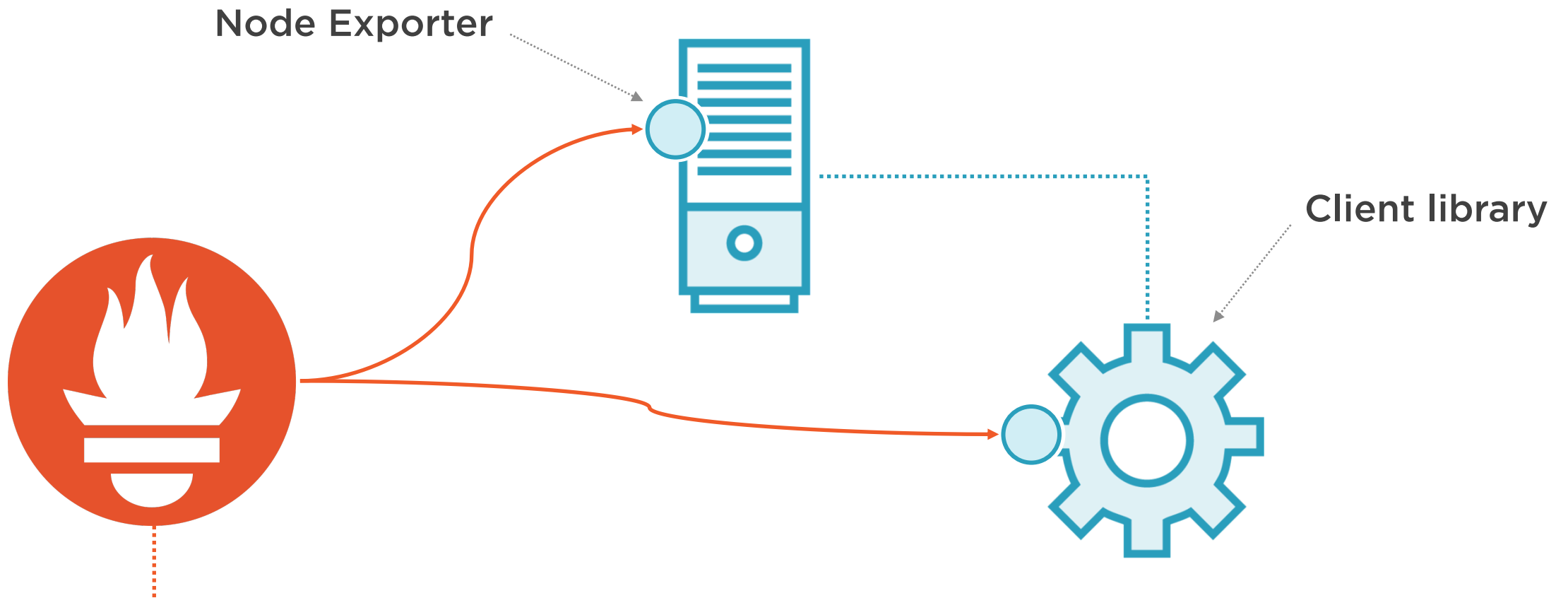


Demo



Metrics: counters and gauges

- Batch application metrics
- Linux Node Exporter
- Metric types and labels



```
process_cpu_seconds_total 5.23
worker_jobs_total{status="processed"} 1570222
worker_jobs_total{status="failed"} 159665
node_filefd_allocated 1184
node_disk_io_time_seconds_total{device="sda"} 104.296
```

Metric type: Counter



`http_requests_total 10000000`

`cpu_seconds_total 3000`

@ 22:00

`http_requests_total 970000`

`cpu_seconds_total 3000`

@ 21:00

`http_requests_total - http_requests_total offset 1h`
`= 300000`

Metric type: Gauge



`http_requests_active` **2000**

`memory_allocated_bytes` **4.832e+09**

@ 22:00

`http_requests_active` **900**

`memory_allocated_bytes` **3.642e+09**

@ 21:00

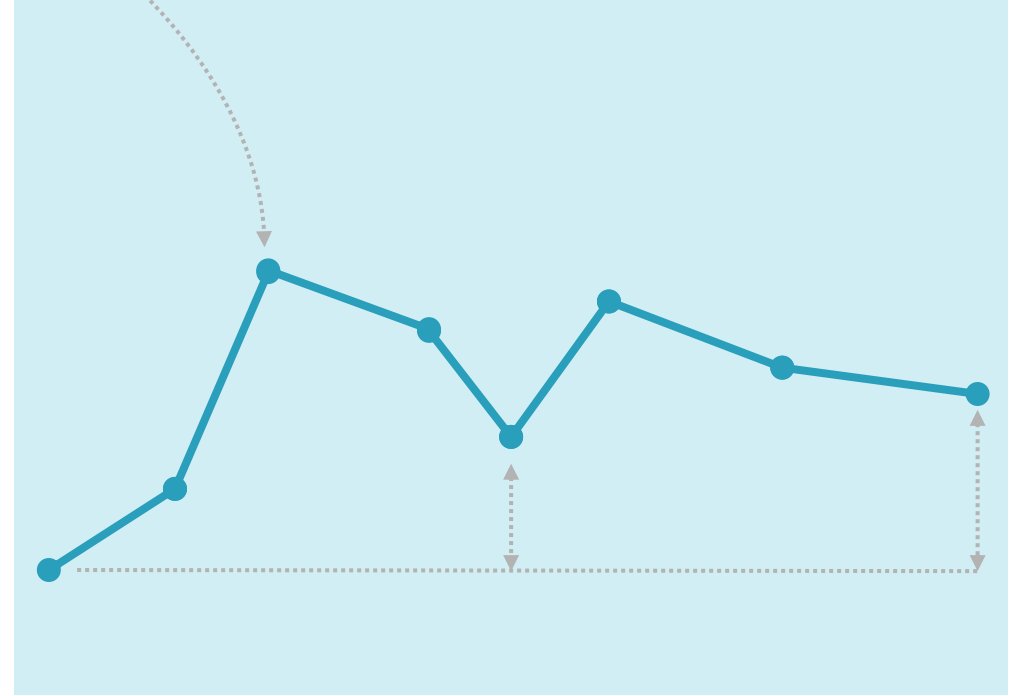


`memory_allocated_bytes` / (1024*1024*1024)

= 4.5 # *gigabytes*



http_requests_total - Rate of change



memory_allocated_bytes - Gauge value

Metric type: Summary

{y,x}

calculation_seconds_count **3**

calculation_seconds_sum **15**

@ 21:00

calculation_seconds_count **10**

calculation_seconds_sum **113**

@ 21:01



rate(calculation_seconds_sum[5m]) /

rate(calculation_seconds_count[5m])

average

Metric type: Histogram



```
calculation_seconds_bucket{le="1"}    0  
calculation_seconds_bucket{le="5"}    3  
calculation_seconds_bucket{le="10"}   6  
calculation_seconds_bucket{le="20"}   9  
calculation_seconds_bucket{le="60"}  10
```

@ 21:00



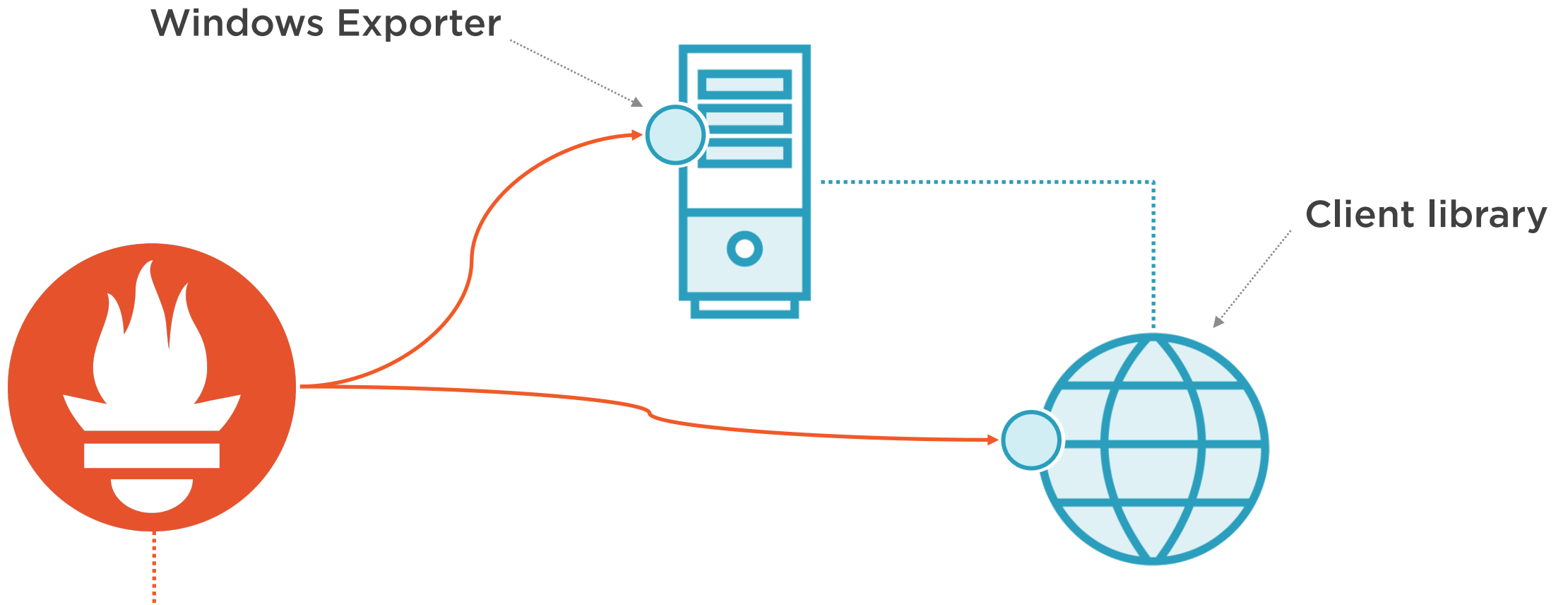
```
calculation_seconds_bucket{le="20"} /  
calculation_seconds_bucket{le="+Inf"} # SLA
```

Demo



Metrics: summaries and histograms

- Windows Exporter
- Web application metrics
- Labels and granularity



```
web_delay_seconds_sum 14
web_delay_seconds_count 2
http_request_duration_seconds_bucket{code="200",method="GET",le="0.002"} 13 #...
windows_cpu_time_total{core="0,0",mode="user"} 36.21875
windows_logical_disk_free_bytes{volume="C:"} 1.24613820416e+11
```

Labels



http_requests_total

code: 200

path: /

→ 800

code: 500

path: /p1

→ 12980

```
http_requests_total{code="200",path="/" } 800
```

```
http_requests_total{code="500",path="/p1" } 12980
```

```
http_requests_total{code="500",path="/p2" } 1064
```

```
http_requests_total{code="404",path="/p3" } 36
```

Incrementing Counters with Labels

index.page

```
if (ok)
{
    requestCounter.Labels("200", "/").Inc();
}
else
{
    requestCounter.Labels("500", "/").Inc();
}
```

Labels

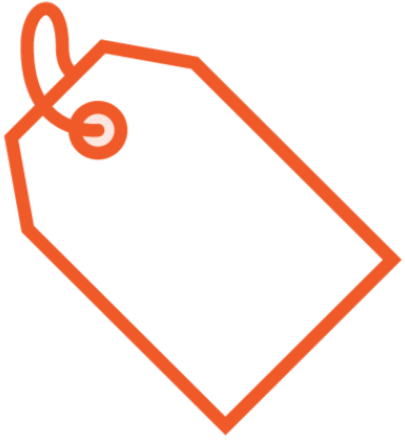


```
http_requests_total{code="200",path="/" } 800
http_requests_total{code="500",path="/p1" } 12980
http_requests_total{code="500",path="/p2" } 1064
http_requests_total{code="404",path="/p3" } 36
```

```
sum without(code, path) (http_requests_total)
= 14880      # all requests
```

```
sum without(path) (http_requests_total{code="500"})
= 14044      # all errors
```

Labels



http_requests_total

code: 200

path: /

host: w01

region: dc1

os: win

gen: 4

Labels



http_requests_total



1x time series

Labels



http_requests_total

host: w01..w19

20x time series

Labels



http_requests_total

host: w01..w19

host: dc1..dc4

80x time series

Labels



http_request_durations_bucket

le: 0.1..10

method: GET, POST

code: 200, 400, 500

60x time series

Labels



`http_request_durations_bucket`

`le: 0.1..10`

`method: GET, POST, HEAD, PUT`

`code: 200, 400, 500, 401, 403, 503`

240x time series

Labels



http_request_durations_bucket

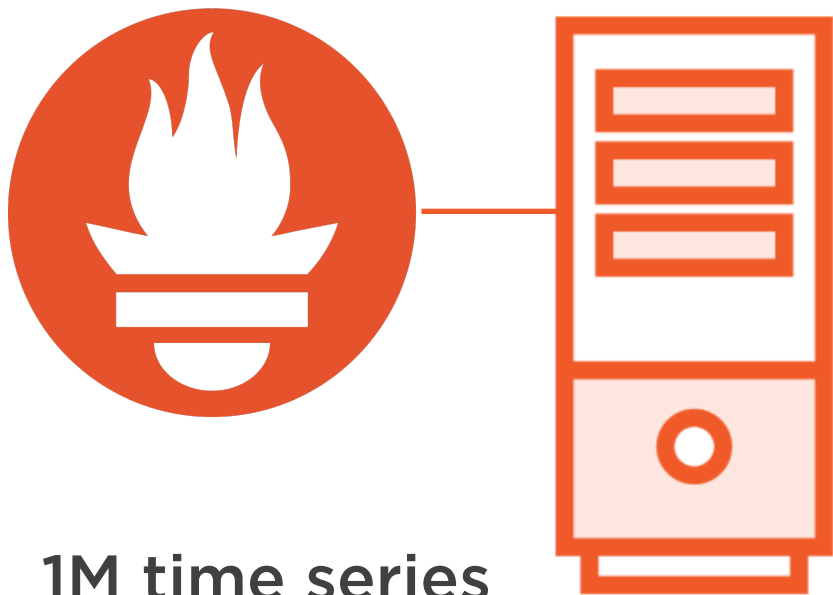
le: 0.1..10

method: GET, POST, HEAD, PUT

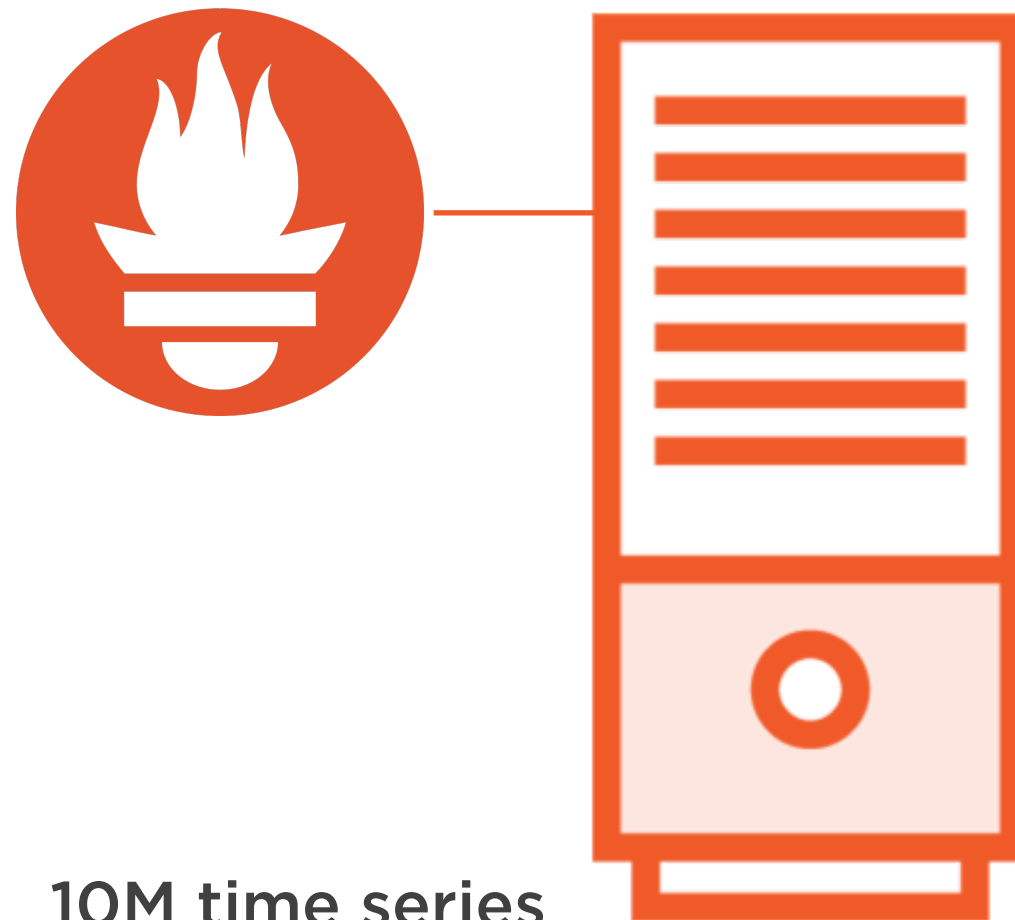
code: 200, 400, 500, 401, 403, 503

path: /, /p1, /p2, /p3...

? time series



1M time series



10M time series

Labels



http_request_durations_bucket

le: 0.1..10

method: GET, POST, HEAD, PUT

code: 200, 400, 500, 401, 403, 503

path: /, /p1, /p2, /p3...

userId: !

∞ time series

Summary



Prometheus architecture

- Pull model to fetch metrics
- Time-series database
- Query API and UI
- Alerting

Monitoring systems

- HTTP metrics endpoint
- Exporters & client libraries

Metric format

- Counters & gauges
- Summaries & histograms
- Labels for granularity

Up Next:

Running and Configuring Prometheus
