

Watch:  **Formato de métricas en Prometheus**

09:29 |

Formato de métricas en Prometheus

Prometheus guarda la información en *time series* que no es nada más que una serie de valores con un *timestamp* asociado. Las *time series* se identifican a través de una métrica y sus etiquetas (*labels*)

👁 **OJO:** Una *time series* se genera para cada combinación de nombre de métrica y conjunto de etiquetas *key/value*. Esto quiere decir que si enviamos la misma métrica pero con dos valores diferentes para una misma etiqueta se generarán dos *time series*. Hay que tener esto en cuenta a la hora de diseñar las etiquetas intentando que tengan la menor cardinalidad posible. Por ejemplo una etiqueta “email” tendría una cardinalidad demasiado grande, generaría un *time series* cada vez que se registre un nuevo email. El impacto en disco es muy grande.

Poniendo nombre a nuestras métricas

Las métricas en Prometheus tienen esta forma:

```
api_http_requests_total{method="POST"}
```

El nombre de una métrica especifica una característica de nuestro sistema que queremos medir. Por ejemplo `api_http_requests_total`. Un nombre de métrica en Prometheus tiene que cumplir la siguiente regexp `[a-zA-Z_:][a-zA-Z0-9_:]*`

Las etiquetas son de la forma *key/value*. Por ejemplo `method="POST"`. Las *keys* deben cumplir la siguiente regexp `[a-zA-Z_][a-zA-Z0-9_]*` y los *values* pueden contener un *string* Unicode. Las etiquetas que empiezan por `_` son de uso interno. Una etiqueta con *key* pero *value* vacío no se tendrá en cuenta.



Es importante tener en cuenta estas **best practices**:

- Convención de **prefijos y sufijos**: el prefijo suele ser el namespace o la aplicación monitorizada (en el ejemplo anterior sería `api`) y el sufijo será la unidad de medición (en el ejemplo anterior es `total`, aunque también sería válidas formas tipo `total_bytes`, `total_seconds` o `info`, entre muchas otras)
- No usar el nombre de las etiquetas en el nombre de la métrica. Es redundante y confuso.
- Se deben usar las unidades de medida recomendadas y no sus variaciones: e.j. *seconds*, *bytes*, *meters* y no *milliseconds*, *megabytes*, *kilometers*.

Etiquetas por defecto

Es muy probable que queramos añadir etiquetas por defecto para todas nuestras métricas, por ejemplo el nombre del host o el entorno `env`. Esto es posible gracias a la propiedad `labels` que podremos

configurar para cada target en nuestro archivo de configuración de Prometheus, que veremos en la siguiente lección.

Formato de métricas en Prometheus

- **Counter** (contador): Métrica de acumulación. Representa un incremento. Por ejemplo el número de requests. Su valor nunca decrementa. Se reinicia a 0 si el servicio que genera la métrica es reiniciado.

```
prometheus_counter_example{label="value"} 42
```

- **Gauge** (calibre): Representa un valor numérico simple que puede variar arbitrariamente. Por ejemplo medir el número de requests concurrentes. Soporta valores en *float64*

```
prometheus_gauge_example{label="value"} 522.123
```

- **Histogram** (histograma): Representan observaciones y realiza operaciones sobre las ocurrencias en *buckets* configurables, por ejemplo la duración media de las requests o el tamaño de respuesta en percentiles. También es acumulativo (está compuesto de *counters*). Cada *bucket* contiene el valor acumulado de los anteriores. Exponen recuentos de observación agrupados y el cálculo de los cuantiles de los *buckets* de un histograma se realiza en el lado del servidor usando la función `histogram_quantile` ().

```
prometheus_histogram_example_bucket{label="value", le="0.1"}  
25547  
prometheus_histogram_example_bucket{label="value", le="0.2"}  
26688  
prometheus_histogram_example_bucket{label="value", le="0.4"}  
27760  
prometheus_histogram_example_bucket{label="value", le="1"}  
28641  
prometheus_histogram_example_bucket{label="value", le="3"}  
28782  
prometheus_histogram_example_bucket{label="value", le="8"}  
28844  
prometheus_histogram_example_bucket{label="value", le="20"}  
28855
```

```

prometheus_histogram_example_bucket{label="value", le="60"}
28860
prometheus_histogram_example_bucket{label="value", le="120"}
28860
prometheus_histogram_example_bucket{label="value", le="+Inf"}
28860
prometheus_histogram_example_sum{label="value"}
1863.80491025699
prometheus_histogram_example_count{label="value"} 28860

```

Aclaraciones:

- El anterior ejemplo se lee tal que así: se han hecho 25547 observaciones de esta métrica cuyos valores están comprendidos entre 0 y 0.1. No conocemos los valores exactos de cada observación, así sucesivamente.
- *sum* es la suma de los valores exactos obtenidos, por ejemplo 0.01, 0.25, 10.15... cada valor por separado no se expone, Solo se expone la suma.
- *count* es el número de observaciones que se han hecho. Por eso el *count* total es el máximo valor del último *bucket*
- Por lo tanto podemos decir que la suma de valores observados es 1863.80491025699 durante 28860 observaciones.
- **Summary** : Similar a un histograma, un resumen muestra las observaciones. También proporciona un recuento total de observaciones y una suma de todos los valores observados, calcula cuantiles configurables en una ventana de tiempo. Los *summaries* calculan los cuantiles en el lado del cliente y los exponen directamente, con lo cual los cuantiles no pueden ser agregados en el lado del servidor. Representan un sub-conjunto de la métrica de tipo Histograma y se recomienda utilizar Histogramas siempre que sea posible.

```

prometheus_summary_example{quantile="0.5"} 6.4853e-05
prometheus_summary_example{quantile="0.9"} 0.00010102
prometheus_summary_example{quantile="0.99"} 0.000177367
prometheus_summary_example_sum 1.623860968846092e+06
prometheus_summary_example_count 1.112293682e+09

```

Aclaraciones:

- *sum* es la suma de los valores exactos obtenidos, por ejemplo 0.0001, 0.000025, 1.2,... cada valor por separado no se expone, Solo se expone la suma.
- *count* es el número de observaciones que se han hecho. En este ejemplo se han hecho más de 1 billón de observaciones.
- Por lo tanto podemos decir que la suma de valores observados es 1.623860968846092e+06 tras 1.112293682e+09 observaciones.