Prometheus

Prometheus is a free software application used for event monitoring and alerting.

**Features**:

1. Data Model with Time series data identified by metric name and key/value pairs
2. PromQL
3. Time series collection happens via pull model over HTTP
4. Targets are discovered via service discovery or static configuration

**Components**:

1. Prometheus Server: Server which scrapes and stores data in time series
2. Client Libraries: libraries to sending metrics from application code
3. Push Gateways: gate way for short lived jobs
4. Exporters: applications/sources
5. Alert Manager: to handle alerts

**PromQL**: users query to select and aggregate data based on aggregate functions.

Ex: rate(), instant vector(), range vector()

**Working**:

1. Prometheus collects data in the form of time series. The time series are built through a pull model:
2. The Prometheus server queries(http scrape) a list of data sources (sometimes called exporters) at a specific polling frequency
3. Prometheus data is stored in the form of metrics, with each metric having a name that is used for referencing and querying it
4. Prometheus stores data locally on disk, which helps for fast data storage and fast querying but ability to store metrics in remote storage.
5. Each Prometheus server is standalone, not depending on network storage or other remote services.

**What is the difference between Grafana and Prometheus?**  
Grafana is only a visualization solution. Time series storage is not part of its core functionality. … The way Prometheus stores time series is the best by far (thanks to its dimensional model, which uses key-value tagging along the time series to better organize the data and offer strong query capabilities)

<https://prometheus.io/docs/prometheus/latest/querying/functions/#vector>

Functions:

The "increase" function calculates how much some counter has grown and the "rate" function calculates the amount per second the measure grows.

Analyzing your data I think you used [30s] for the "increase" and [1m] for the "rate" (the correct used values are important to the result).

Basically, for example, in time 2m we have:

increase[30s] = count at 2m - count at 1.5m = 4423 - 4402 = 21

rate[1m] = (count at 2m - count at 1m) / 60 = (4423 - 4381) / 60 = 0.7

Prometheus documentation: [increase](https://prometheus.io/docs/prometheus/latest/querying/functions/#increase) and [rate](https://prometheus.io/docs/prometheus/latest/querying/functions/#rate).

Graphical user interface, application

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated