

Prometheus

<https://prometheus.io/docs/prometheus/latest/configuration/configuration/>

Local Setup using Vagrant (with Node Exporter):

<https://github.com/raghav1674/prometheus-learning/tree/master/setup>

Local Setup using docker (with cadvisor):

<https://github.com/raghav1674/prometheus-learning/tree/master/docker-prom>

- **Target** : what to be scraped
- **Instance**: Endpoint of the target
- **Job**: Collection of Instances / targets

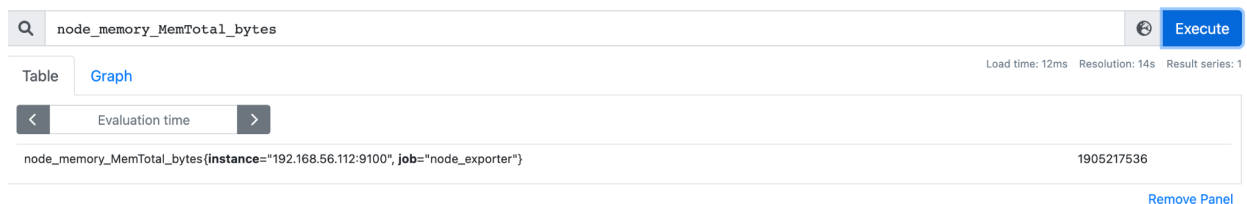
```
scrape_configs:
  # The job name is added as a label to any timeseries scraped from this config.
  - job_name: 'prometheus'

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

  static_configs:
    - targets: ['localhost:9090']
```

Data Types in PromQL:

- **Instant Vector**: for each time series, a single value is return



- **Range Vector**: for each time series, a range of values is returned



- **Scaler:** simple floating value



- **String:** simple string value w/o any label

Selectors/Matchers:

- expression{label='value'}

Operators:

!=

=

=~

And,or,unless

Ignoring: ignore those labels while matching

On: match labels

Eg: Left promql and ignoring(label) right promql

In english: return the left query result which is matching the right query but while matching ignore the label from the right query.

Aggregation Queries:

- Sum based on a label : sum(query) **by** (label)

Functions:

- **Rate:** how fast a count is increasing.

Recording Rules:

- Precomputed queries

Relabel Config: relabelling the source label to some other target labels

<https://grafana.com/blog/2022/03/21/how-relabeling-in-prometheus-works/>

Push Gateway:

<https://prometheus.io/docs/practices/pushing/>

- Do post request on push gateway endpoint

Service Discovery:**ec2_sd_config:**

https://prometheus.io/docs/prometheus/latest/configuration/configuration/#ec2_sd_config

file_sd_config:

https://prometheus.io/docs/prometheus/latest/configuration/configuration/#file_sd_config

Prometheus Agent Installation

[Version >= 2.32.0] (Same setup as prometheus server, just need to set couple of flags)

- Download Prometheus software
- Create Prometheus system user
- Create `/etc/prometheus` -> config dir
 `/var/lib/prometheus` -> data dir
- Copy binaries to `/usr/local/bin` -> (add to Path)
- Copy configuration file with remote write config
- Copy everything to the the `/etc/prometheus`
- Change permissions
- Create Service File with option
 `--enable-feature=agent,--storage.agent.path=/data`

Blog: <https://prometheus.io/blog/2021/11/16/agent/>

Remote Write:

https://prometheus.io/docs/prometheus/latest/configuration/configuration/#remote_write