Substrate Prometheus Node Exporter

Introduction

Prometheus is one of the most widely used monitoring tool for managing high availability services supported by Cloud Native Computing Foundation. By providing Prometheus metrics in Substrate, node operators can easily adopt widely used display/alert tools such as Grafana and Alertmanager without setting-up/operating external Prometheus push gateways (which is an antipattern in the first place) through RPC connections. Easy access to such monitoring tools will benefit parachain developers/operators and validators to have much higher availability of their services.

Table of Contents

Metrics

substrate can report and serve the Prometheus metrics, which in their turn can be consumed by Prometheus collector(s).

This functionality is disabled by default.

To enable the Prometheus metrics, set in your cli command (--prometheus-addr,--prometheus-port). Metrics will be served under /metrics on 33333 port by default.

List of available metrics

Consensus metrics, namespace: `substrate`

Name	Type	Tags	Description
consensus*finality*block*height*number	IntGauge		finality Height of the chain
consensus*best*block*height*number	IntGauge		best Height of the chain
consensus*target*syn_number	IntGauge		syning Height target number
consensus*num*txs	Gauge		Number of transactions
consensus*node*memory	IntGauge		Node's primary memory
consensus*node*cpu	IntGauge		Node's cpu load
consensus*state*cache_size	IntGauge		used state cache size

p2p*peers*number	IntGauge		Number of peers node's connected to
p2p*peer*receive*bytes*per_sec	IntGauge		number of bytes received from a given peer
p2p*peer*send*bytes*per_sec	IntGauge		number of bytes sent to a given peer
Resource*receive*bytes*per*sec(Future)	IntGauge		Operating System of bytes received
Resource*send*bytes*per*sec(Future)	IntGauge		Operating System of bytes sent
Resource*cpu*use(Future)	IntGauge		Operating System cpu load
Resource*disk*use(Future)	IntGauge		Operating System disk use
validators*sign*prevote(Future)	IntGauge	validator addr	validator sign vote list
validators*sign*precommit(Future)	IntGauge	validator addr	validator sign commit list

Start Prometheus

Install prometheus

https://prometheus.io/download/

```
wget <download URL>
tar -zxvf <prometheus tar file>
```

Edit Prometheus config file

You can visit prometheus.yml to download default `prometheus.yml`.

Then edit `prometheus.yml` and add `jobs` :

```
- job_name: kusama
    static_configs:
    - targets: ['localhost:33333']
      labels:
        instance: local-validator
```

Note： value of targets is ip:port which used by substrate monitor

Start Prometheus

```
cd <prometheus file>
./prometheus
```

The above example, you can save `prometheus.yml` at `~/volumes/prometheus` on your host machine

You can visit `http://localhost:9090` to see prometheus data.

Start Grafana

Install Grafana

https://grafana.com/docs/installation/debian/

```
apt-get install -y software-properties-common
sudo add-apt-repository "deb https://packages.grafana.com/oss/deb stable main"
wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -
```

```
sudo apt-get update
sudo apt-get install grafana
sudo service grafana-server start
./prometheus
```

You can visit `http://localhost:3000/` to open grafana and create your own dashboard.

> Tips: The default username and password are both admin. We strongly recommend immediately changing your username & password after login

Seting Grafana

Default ID:PW is admin.