

Prometheus - 14/10/2024

Task:

Installare Prometheus su una macchina virtuale tramite docker, visualizzare lo stato delle risorse della macchina virtuale e del container con prometheus.

Fare due widget in Grafana, uno che mostri in timeseries un throttling di cpu sulla vm e un altro di tipo "stats" che mi indichi la percentuale di memoria utilizzata dai container e che faccia salire su widget solo quello con la memoria più bassa.

Svolgimento:

Per poter ricevere le metriche relative allo stato delle risorse della macchina virtuale che contiene il container che ospita Prometheus abbiamo bisogno di un node exporter.

Per le metriche relative al container abbiamo bisogno di cadvisor che ci permette di visualizzare le risorse e le informazioni dei nostri container.

Useremo le metriche prodotte dal node exporter e dal container con cadvisor per leggerle tramite prometheus.

1. Attraverso un vagrant e un playbook ansible tiriamo su la macchina virtuale, eseguendo quindi il playbook che installa e configura docker, il container con prometheus, il node exporter e il container con il Vagrantfile e il playbook sono disponibili al link:
<https://github.com/federicopao/sou-lab-cni>
2. Per aggiungere le metriche del node exporter e del cadvisor a prometheus abbiamo configurato il prometheus.yml nel seguente modo:

```
global:
  scrape_interval: 15s

scrape_configs:
  - job_name: 'prometheus'
    static_configs:
      - targets: ['localhost:9090']
  - job_name: 'node_exporter'
    static_configs:
      - targets: ['192.168.56.22:9100']
  - job_name: 'cadvisor'
    static_configs:
      - targets: ['192.168.56.22:8080']
```

3. Avviamo l'infrastruttura con il comando "vagrant up". Dobbiamo avviare manualmente il node exporter. Entriamo in ssh nella macchina con "vagrant ssh node2" e usiamo i comandi:

```
cd node_exporter-0.18.1.linux-arm64
./node_exporter
```

e su un'altra scheda del terminale sempre in ssh con la macchina virtuale usiamo il comando `curl http://localhost:9100/metrics | grep "node_"` per vedere se le metriche del node exporter sono attive.

Sul nostro browser possiamo verificare se il container con il nostro cadvisor funziona correttamente all'indirizzo <http://192.168.56.22:8080> dove possiamo vedere lo stato delle risorse dei nostri container:



- riavviamo il container con prometheus usando il comando `sudo docker restart prometheus` e nel browser del nostro pc verifichiamo che il node exporter sia riconosciuto da prometheus all'indirizzo <http://192.168.56.22:9090/targets>. Il risultato atteso dovrebbe essere il seguente:

Prometheus Alerts Graph Status Help

Targets

All scrape pools All Unhealthy Collapse All Filter by endpoint or labels

cadvisor (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://192.168.56.22:8080/metrics	UP	instances="192.168.56.22:8080" job="cadvisor"	17.121s ago	102.612ms	

node_exporter (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://192.168.56.22:9100/metrics	UP	instances="192.168.56.22:9100" job="node_exporter"	6.728s ago	34.650ms	

prometheus (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instances="localhost:9090" job="prometheus"	12.634s ago	19.098ms	

- Dalla propria macchina su browser entrare su prometheus e testare le metriche fornite dal node exporter:



- Apriamo ora Grafana all'indirizzo <http://192.168.56.22:3000> (Grafana è stato avviato in precedenza con l'esecuzione del playbook), andiamo ad aggiungere i dati di Prometheus a Grafana nella sezione "Data Sources".
- creo una dashboard che mostri in timeseries un throttling di cpu sulla virtual machine utilizzando la seguente query:

```
3. sum(1 - rate(node_cpu_seconds_total{job="node_exporter",
mode="idle"}[5m]))
```

la metrica `rate(node_cpu_seconds_total in mode=idle)` mi mostra il tempo medio di inattività della cpu, il `1- rate(..)` serve per sottrarre dal totale di utilizzo, l'utilizzo medio. il risultato è il seguente:



8. Creo una dashboard di tipo “stats” che mi indichi la percentuale di memoria utilizzata dai container e che faccia salire su widget solo quello con la memoria più bassa:

Come prima cosa identifico le metriche che andrò a utilizzare:

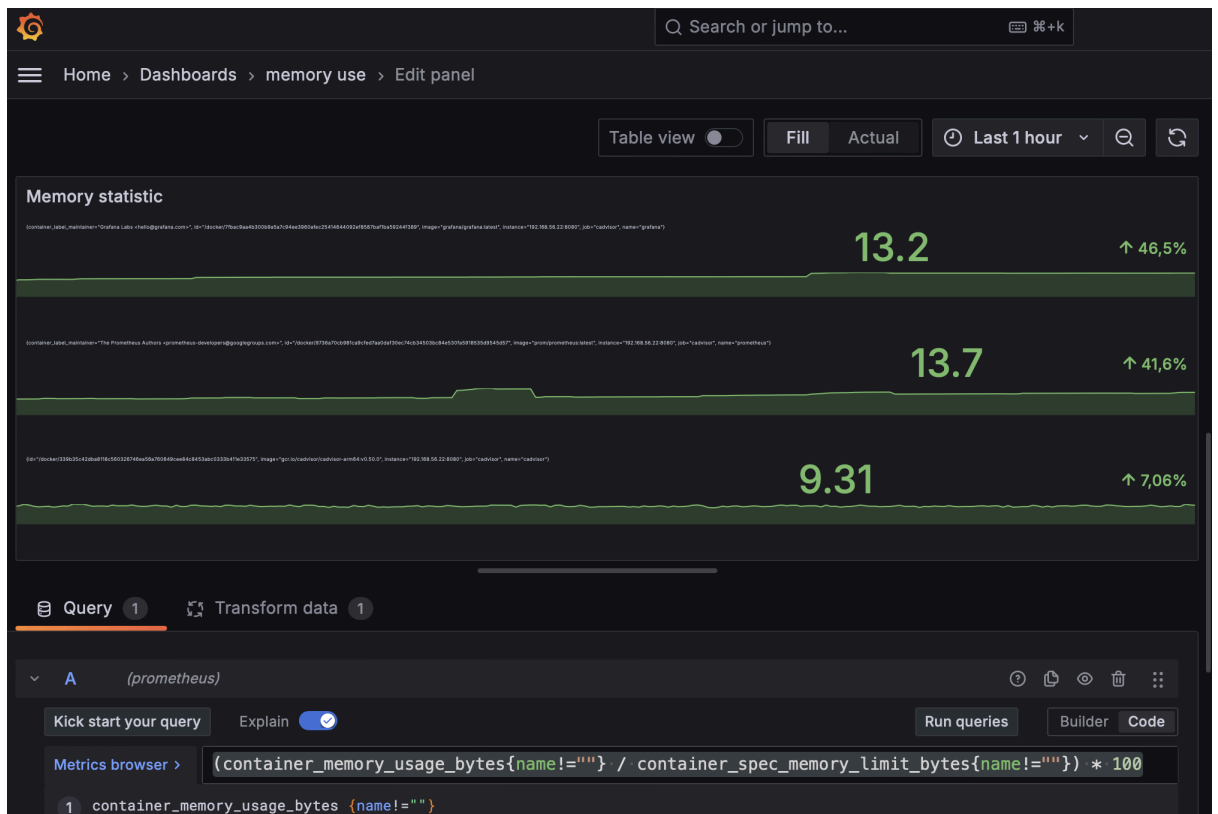
`container_memory_usage_byte` mi mostra i byte di memoria usati da un container.

`container_spec_memory_limit_byte` mostra i byte di memoria massimi assegnati a quel container in fase di creazione (se in fase di creazione del container non viene definito un limite di memoria, la metrica restituirà come valore 0).

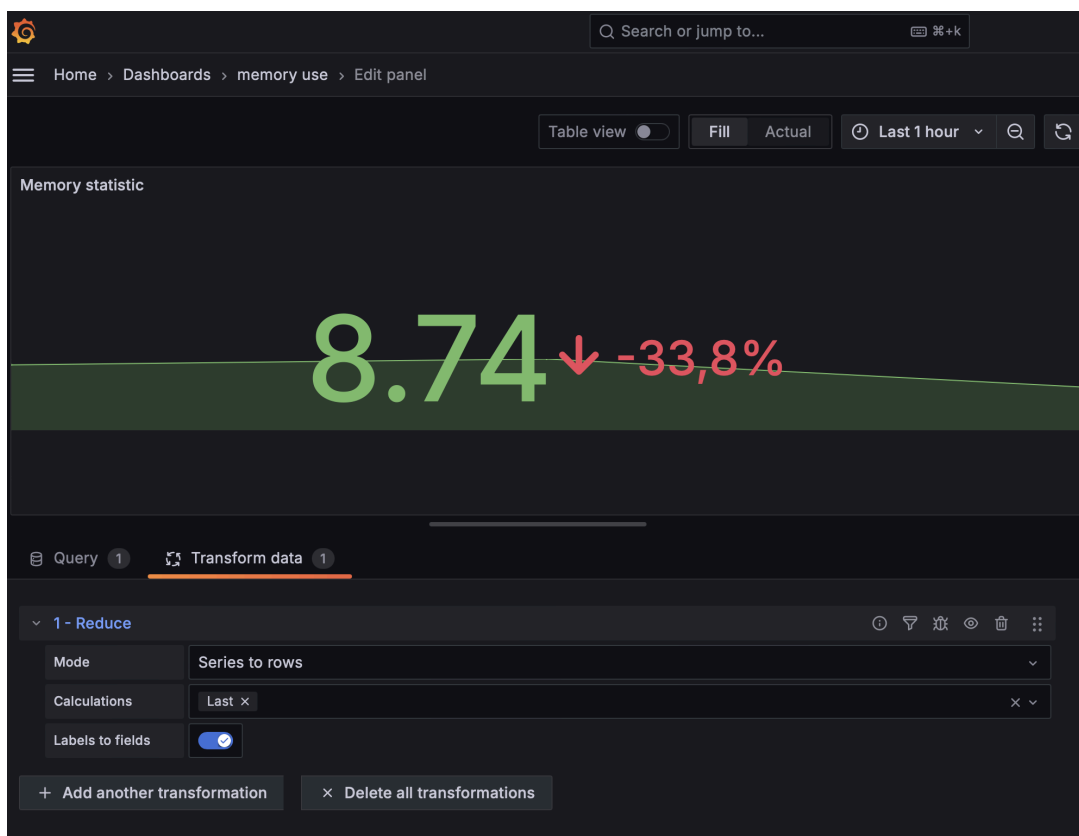
utilizzo la seguente query e seleziono il tipo di dashboard “stat”:

```
(container_memory_usage_bytes{name!=""}) /
container_spec_memory_limit_bytes{name!=""}) * 100
```

Il risultato è il seguente:



Per mostrare quale container utilizza la percentuale minima su dashboard utilizziamo l'opzione "reduce" della sezione transform data:



Per maggiori informazioni:

<https://prometheus.io/docs/guides/node-exporter/#tarball-installation>

<https://prometheus.io/docs/guides/cadvisor/>

<https://github.com/google/cadvisor>

<https://stackoverflow.com/questions/71126401/find-exact-cpu-percentage-from-the-metrics-exported-by-prometheus-node-exporter>