

Simple Docker Monitor

Progetto Gestione di Reti 2017/2018

Autore: **Sturba Gionatha**

Matricola: **531274**

Descrizione

Una delle recenti tecnologie sviluppatasi nell'ambito IT sono i Docker Container.

Un container è un'applicazione isolata (con tutte le sue dipendenze) eseguita in namespace separati (network namespace, file system, users, IPC).

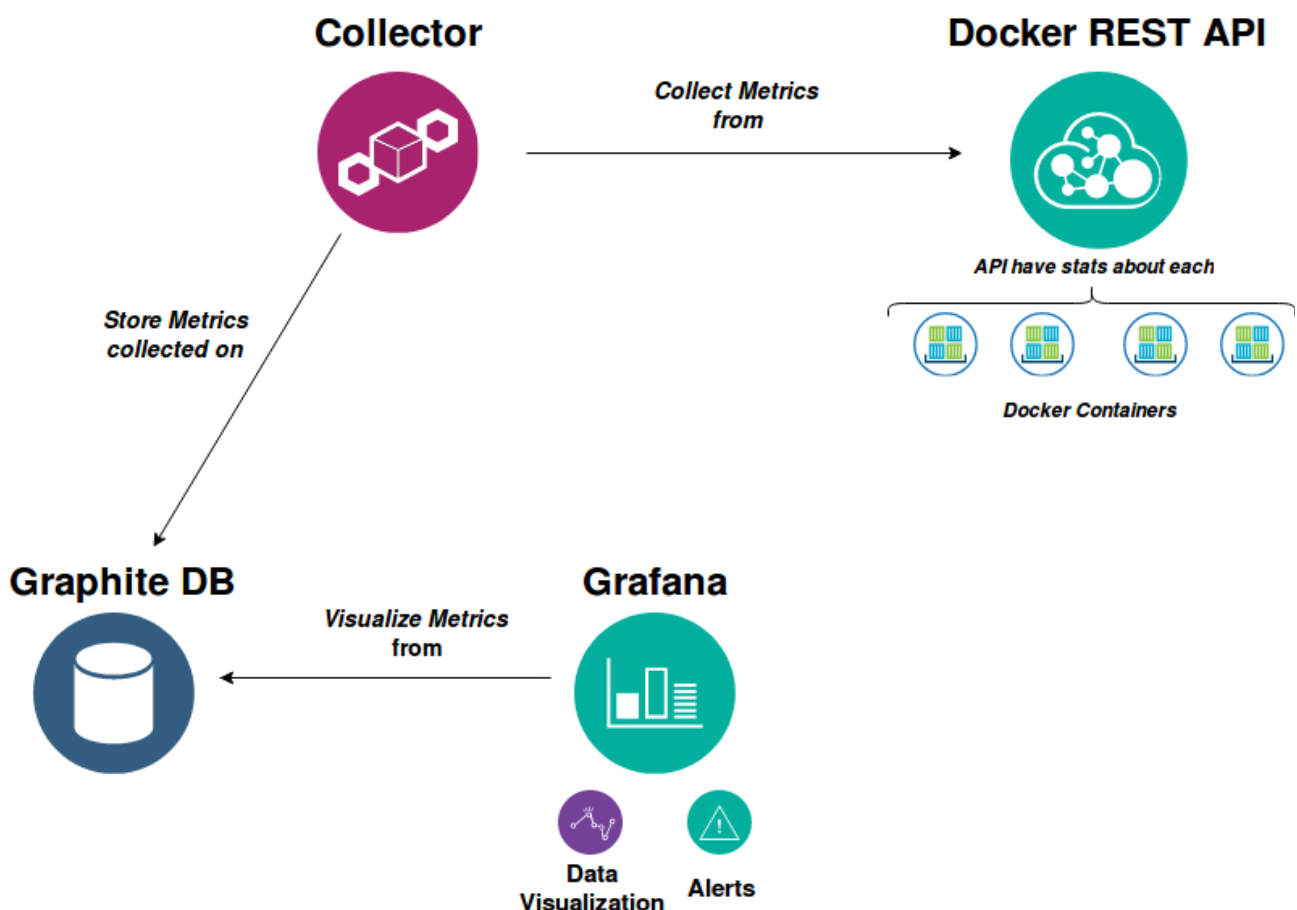
I container condividono il kernel della macchina host (ovvero la macchina dove vi è allocato) e non hanno uno strato di virtualizzazione tramite hypervisor.

Docker , invece, è uno strumento che ha, come scopo principale, quello di rendere più facile la creazione, il deploy e l'esecuzione di applicazioni utilizzando i container.

Essendo, quindi, delle applicazioni vere e proprio che girano all'interno del sistema operativo, nasce la necessita' di dover monitorare questi container, soprattutto sull'impatto che hanno sulla macchina in cui girano (consumo Cpu,Ram,Disk,Network ecc...).

Il progetto consiste nello sviluppo di un sistema in grado di collezionare, processare ed aggregare una serie di metriche riguardanti i Docker container in esecuzione sulla macchina.

Architettura del sistema



E' stato sviluppato un collezionatore, in Java, che ha il compito di ricavare le metriche sui container in esecuzione, tramite la Docker Remote REST API.

Quest' API, di tipo REST, dopo essere stata attivata sulla macchina in cui girerà il sistema ([guida attivazione API](#)), conterra' una serie di informazioni riguardanti ogni container in esecuzione sulla macchina.

Queste informazioni o statistiche, vengono collezionate dal Collector, tramite il protocollo REST, per poi essere processate in delle metriche piu' affine (es: consumo in percentuale cpu, network speed on default interface, ecc..).

Una volta processate le metriche, quest'ultime vengono inviate e salvate all'interno di un Time Series Database, in questo caso Graphite.

Vi e' poi un altro servizio, Grafana, in grado di visualizzare le metriche raccolte all'interno di Graphite, con opportuni grafici e con la possibilita' di settare anche degli alerts.

IL COLLECTOR

Il Collezionatore di metriche sui container, e' stato sviluppato per il progetto, mentre Graphite e Grafana vengono utilizzati come servizi di supporto al collezionatore.

E' stato scritto in Java, ed ha un architettura multithread.

In questo caso abbiamo un thread (*collector.java*) che ha il compito di interrogare l'API per vedere quali container sono attivi in un determinato istante, per poterne poi ricavare delle metriche; operazione che si ripete ad intervalli ogni tot secondi (*settando un timeout all'interno del file di configurazione*).

L'operazione di recupero informazioni e invio metriche al DB, riguardanti un particolare container, viene svolta però da un altro thread (*MetricContainerCollector.java*) all'interno di un pool, che è gestito sempre dal collector(*collector.java*).

Una volta recuperate le informazioni su un particolare container, vengono processate delle metriche(*consumo CPU, consumo RAM, operazioni IO, network speed (tx,rx)*), che verranno poi inviate al DB gestito da Graphite, tramite una connessione di rete.

Graphite e Grafana

Come già accennato, sono stati usati 2 servizi di supporto al collector, Graphite e Grafana.

Ovviamente sono propedeutici al funzionamento del collector, quindi devono essere installati sulla macchina host (**nel caso del progetto, vengono fatti partire come 2 docker container separati, all'interno dello script *init_all.sh***).

Graphite viene utilizzato come DB per collezionare le metriche inviategli dal collector.

In genere utilizza la porta 2003 per ricevere le metriche, grazie al Carbon Metric daemon.

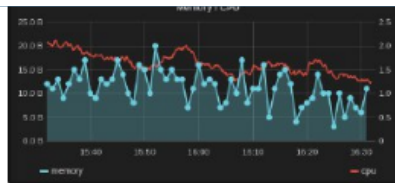
Grafana viene utilizzato per Data Visualization delle metriche, ed anche per settare degli alert al superamento di valori di soglia limite.

In genere è raggiungibile sulla porta 3000.

Esempi di DashBoard di Grafana:



Esempio di Allert:



Unread messages



10:37:22 AM

[Alerting] Containers Cpu Usage alert
 Message: Containers Cpu Usage is above 60%
 URL: <http://localhost:3000/d/yYg5-bSmz/containers?fullscreen=true&edit=true&tab=alert&panelId=2&orgId=1>



10:37:29 AM

[OK] Containers Cpu Usage alert
 Message: Containers Cpu Usage is above 60%
 URL: <http://localhost:3000/d/yYg5-bSmz/containers?fullscreen=true&edit=true&tab=alert&panelId=2&orgId=1>

GI

Write a message...

TE

SEND

Una volta che il sistema funziona correttamente, e' possibile accedere alla Web Interface di Grafana (su porta 3000) per visualizzare e creare i propri grafici, e settare gli alert per particolari situazioni critiche.