

Datadog - 15/10/2024

Esercizio:

Installare un agent Datadog sulla propria macchina e collegarsi alle metriche di prometheus tramite integrations.

Monitorare poi con Datadog i container presenti sulla propria macchina e inviare una notifica in caso uno di questi non sia più in running.

Infine creare un cluster Kubernetes e collegarlo a Datadog per monitorarlo.

Svolgimento:

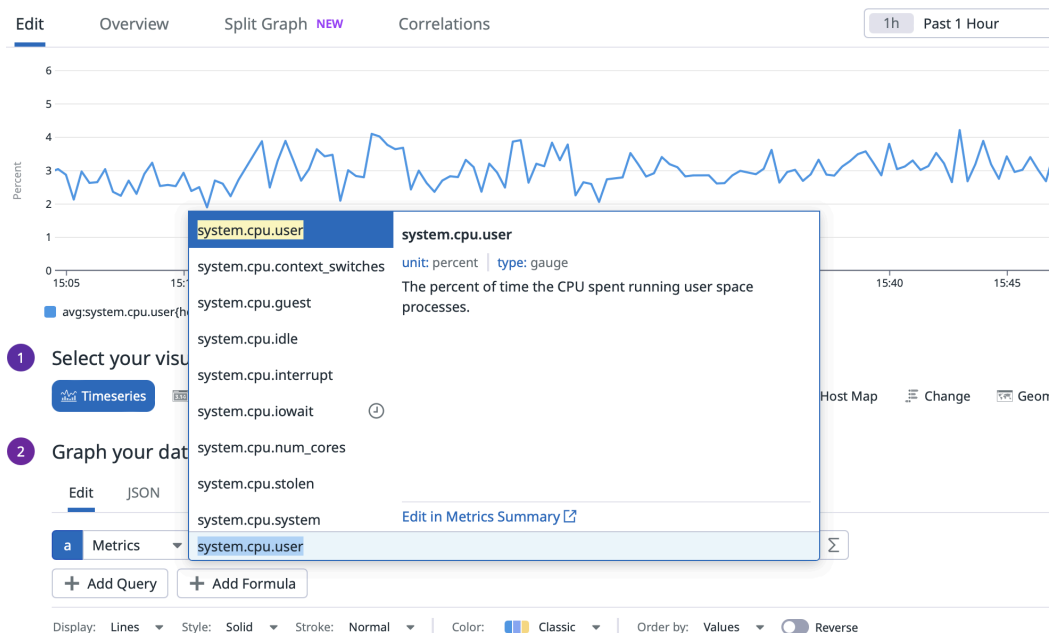
1. Sulla propria macchina con ubuntu utilizzare il seguente comando per installare l'agent:

```
DD_API_KEY=<my_key> DD_SITE="datadoghq.eu" bash -c "$(curl -L https://install.datadoghq.com/scripts/install_script_agent7.sh)"
```

è possibile creare o recuperare la chiave dal proprio account di Datadog al link <https://app.datadoghq.com/organization-settings/api-keys>.

Per verificare se sono presenti errori eseguire il comando: `sudo datadog-agent status`

2. Per testare il corretto funzionamento dell'agent su Datadog creiamo una nuova dashboard (Dashboard>New Dashboard) e un nuovo widget (Dashboard>My_Dashboard>New Widget) e verifichiamo siano disponibili le metriche dell'host dove abbiamo installato l'agent:



3. Per poter verificare che Prometheus sia disponibile tramite le integrazioni di Datadog andare nella sezione "Integrations" di Datadog e nella barra di ricerca scrivere prometheus, infine verificare che sia installata come integrazione.



Prometheus (legacy)

by [Datadog](#)

Prometheus is an open source monitoring system for timeseries metric data

✓ INSTALLED

[Overview](#) [Configure](#) [Support](#) [Release Notes](#)

Configuration

Follow the instructions below to install and configure this check for an Agent running on a host. For containerized environments, see the [Autodiscovery Integration Templates](#) for guidance on applying these instructions.

Installation

The Prometheus check is packaged with the [Datadog Agent](#) starting version 6.1.0.

Configuration

Edit the `prometheus.d/conf.yaml` file to retrieve metrics from applications that expose OpenMetrics / Prometheus end points.

4. Sulla macchina dove abbiamo installato l'agent apriamo il file `conf.yaml` all'indirizzo: `/etc/datadog-agent/conf.d/prometheus.d` e nella sezione `instances` indicare l'indirizzo di prometheus e le metriche che si vogliono riportare su Datadog (in questo caso tutte `"*"`).

Se prometheus è configurato per raccogliere metriche da diversi targets, aggiungiamo questi ultimi come ulteriori istanze:

```
## Every instance is scheduled independently of the others.
#
instances:

  - @param prometheus_url - string - required
    ## The URL where your application metrics are exposed by Prometheus.
    #
    - prometheus_url: http://192.168.56.22:9100/metrics
      namespace: "cadvisor"
      metrics:
        - "*"

  - prometheus_url: http://192.168.56.22:9100/metrics
    namespace: "node"
    metrics:
      - "*"

  - prometheus_url: http://192.168.56.22:9090/metrics


  ## @param namespace - string - required
  ## The namespace to be prepended to all metrics.
  #
  namespace: prometheus

  ## @param metrics - (list of mapping or string) - required
  ## List of `<METRIC_TO_FETCH>: <NEW_METRIC_NAME>` for metrics to be fetched from the Prometheus endpoint.
  ## `<NEW_METRIC_NAME>` is optional. If set, it transforms the name in Datadog.
  ## This list should contain at least one metric.
  #
  metrics:
    - "*"

```

Infine aggiornare il servizio con `sudo systemctl restart datadog-agent` e verificare su dashboard se ora sono disponibili le nuove metriche.

5. Ora per monitorare i container presenti sulla mia macchina verifico che l'integration per docker sia installata su datadog seguo le istruzioni sulla configurazione dell'integration dando i permessi e modificando il file:
`/etc/datadog-agent/conf.d/docker.d/conf.yaml`



Docker Integration

Docker is an open-source project that automates the deployment of applications inside software containers.

✓ INSTALLED

Overview **Configuration** Metrics

This integration is working properly.

Monitoring containers through the Docker integration has the potential to impact your bill. For more information on how this may affect your billing, visit the [Billing FAQ](#) page, or reach out to support@datadoghq.com.

If you want to run the Agent inside a container, follow the [Docker Agent documentation](#).

1. Make sure that Docker is installed and running on your server.
2. Add the user running the Agent to Docker's group.

```
usermod -a -G docker dd-agent
```

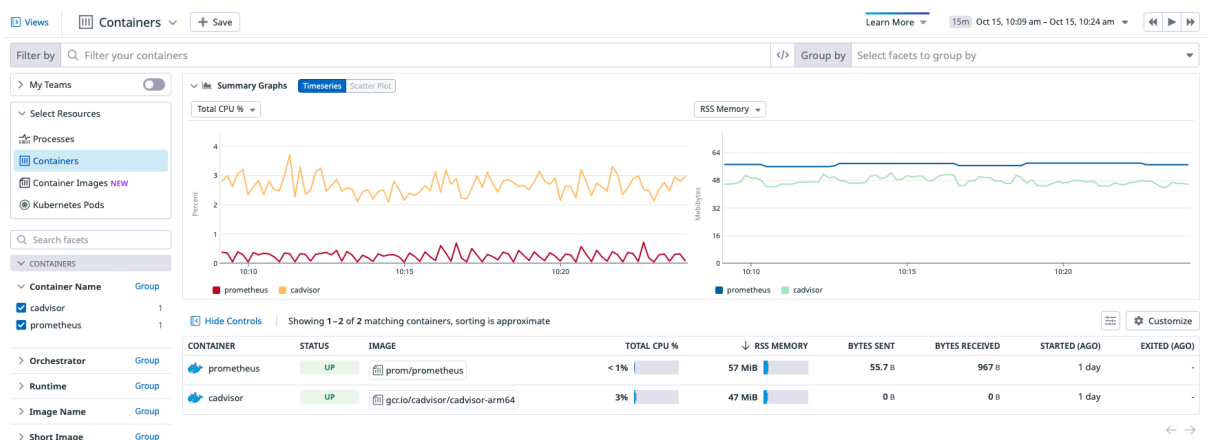
3. Configure the Agent to connect to Docker.
Edit `conf.d/docker.d/docker_daemon.yaml`:

```
init_config:

instances:
  - url: "unix:///var/run/docker.sock"
```

4. [Restart the Agent](#)

6. Per monitorare lo stato dei container è possibile andare in Infrastructure>My_Host>Containers



7. Infine per ricevere un alert tramite notifica nel caso un container non sia più in stato running utilizziamo i Monitors di Datadog che ci permettono di segnalare degli alert in base a una condizione impostata.
Andiamo su Monitors>New Monitor

2 Define the metric

Source Edit

a docker.containers.running.total from (everywhere) avg by (everything) Σ

+ Add Query + Add Formula

Evaluation Details

Evaluate the average Of the change Over 5 minutes Compared to 1 hour

3 Set alert conditions

Trigger when the evaluated value is below the threshold

Alert threshold: < 3 (3 %)

Warning threshold: < Optional

If data is missing for 1 hour Show last known status

> Advanced options

4 Configure notifications & automations

Edit Preview

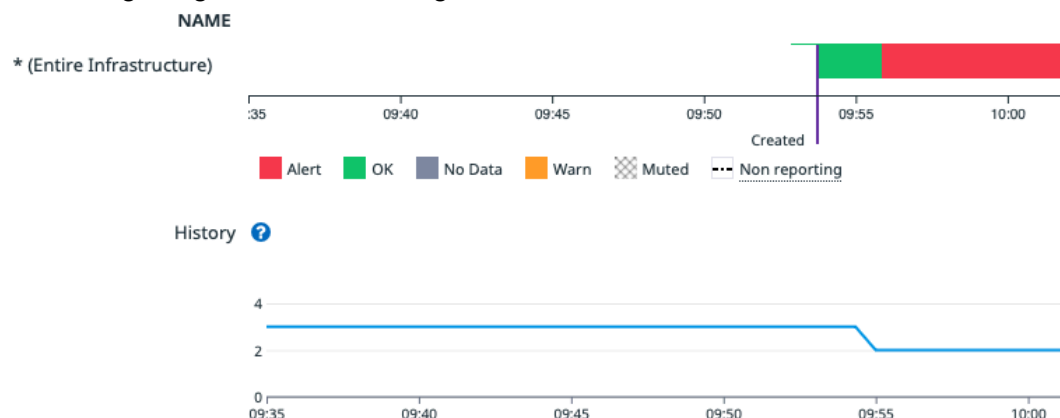
Test datadog

H B I S | ↻ ↺ <> ⌵ ⌶ ☑ | 🖨 📧 — | @ {{

container andato giù @federico.paolucci@sourcesense.com

Nella seconda sezione possiamo selezionare la metrica da monitorare, nella terza sezione invece impostiamo la condizione di alert o di warning, infine settiamo la notifica.

Stopppando un container con il comando `sudo docker stop grafana` è possibile vedere come venga segnalato da Datadog:



Triggered: Test datadog

container andato giù @federico.paolucci@sourcesense.com

The change in **docker.containers.running.total** over * for a 5m avg was < **3.0** compared to 1h ago.



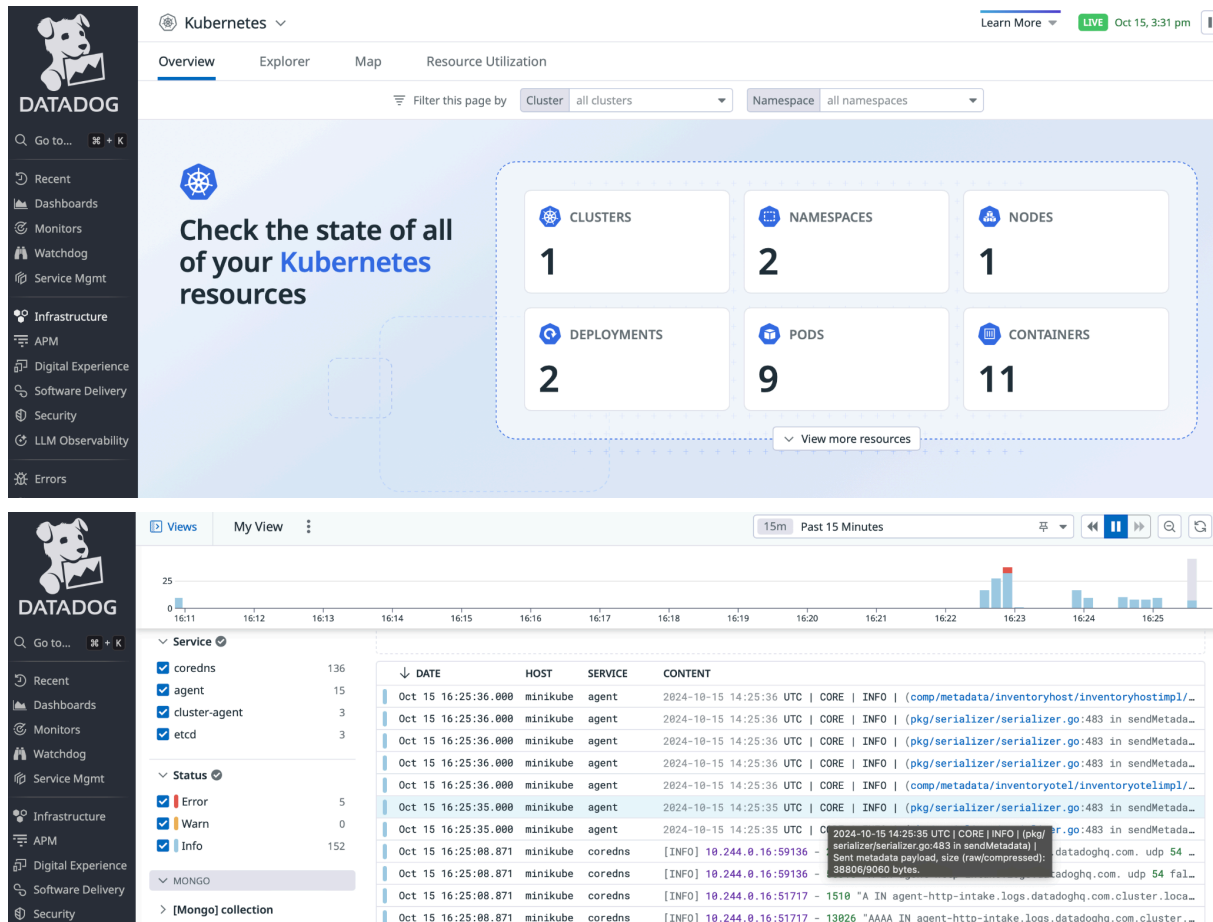
8. Per creare un cluster uso Minikube sul mio pc, con il comando `minikube start` creo il cluster e aggiungo l'agent e l'agent cluster tramite Helm chart. Eseguo i seguenti comandi:

```
helm repo add datadog https://helm.datadoghq.com
```

```
helm repo update
```

```
helm install datadog-agent datadog/datadog \
--set datadog.apiKey=My_Key \
--set datadog.kubelet.enabled=true \
--set datadog.logs.enabled=true \
--set datadog.clusterName=minikube \
--set orchestratorExplorer.enabled=true \
--set datadog.kubelet.tlsVerify=false \
--set clusterAgent.livenessProbe.initialDelaySeconds=120 \
--set clusterAgent.readinessProbe.initialDelaySeconds=120 \
--set clusterAgent.startupProbe.initialDelaySeconds=120 \
--set datadog.logs.containerCollectAll=true \
--namespace=datadog
```

9. Per verificare il corretto funzionamento dell'agent andare su datadog in Infrastructure>Containers>Kubernetes Overview



Per maggiori informazioni:
https://docs.datadoghq.com/getting_started/
<https://docs.datadoghq.com/containers/kubernetes/>