CAdvisor

The docker stats command and the remote API are useful for getting information on the command line, however, if you would like to access the information in a graphical interface you will need a tool such as [CAdvisor](https://github.com/google/cadvisor). CAdvisor provides a visual representation of the data shown by the docker stats command earlier.  Run the docker command below and go to *http://<your-hostname>:8080/*in the browser of your choice to see the CAdvisor interface. You will be shown graphs for overall CPU usage, Memory usage, Network throughput and disk space utilization. You can then drill down into the usage statistics for a specific container by clicking the *Docker Containers* link at the top of the page and then selecting the container of your choice.  In addition to these statistics CAdvisor also shows the limits, if any, that are placed on container, using the Isolation section.

Scout

The next approach for docker monitoring is Scout and it addresses several of limitations of CAdvisor. Scout is a hosted monitoring service which can aggregate metrics from many hosts and containers and present the data over longer time-scales. It can also create alerts based on those metrics. The first step  to getting scout running is to sign up for a Scout account at <https://scoutapp.com/>, the free trial account should be suitable for testing out integration.  Once you have created your account and logged in, click on your account name in the top right corner and then Account Basicsand take note of your Account Key as you will need this to send metrics from our docker server.

## Data Dog

From Scout lets move to another monitoring service, DataDog, which addresses several of the short-comings of Scout as well as all of the limitations of CAdvisor. To get started with DataDog, first sign up for a DataDog account at <https://www.datadoghq.com/>. Once you are signed into your account you will be presented with list of supported integrations with instructions for each type.  Select docker from the list and you will be given a docker run command (show below) to copy into your host. The command will have your API key preconfigured and hence can be run the command as listed. After about 45 seconds your agent will start reporting metrics to the DataDog system.

## Sensu Monitoring Framework

Scout and Datadog provide centralized monitoring and alerting however both are hosted services that can get expensive for large deployments. If you need a self-hosted, centralized metrics service,  you may consider the [sensu open source monitoring framework](http://sensuapp.org/" \t "_blank). To run the Sensu server you can use the [hiroakis/docker-sensu-server](https://registry.hub.docker.com/u/hiroakis/docker-sensu-server/" \t "_blank) container. This container installs sensu-server, the uchiwa web interface, redis, rabbitmq-server, and  the sensu-api. Unfortunately sensu does not have any docker support out of the box. However, using the plugin system you can configure support for both container metrics as well as status checks.

## Prometheus

First lets take a look at Prometheus; it is a self-hosted set of tools which collectively provide metrics storage, aggregation, visualization and alerting. Most of the tools and services we have looked at so far have been push based, i.e. agents on the monitored servers talk to a central server (or set of servers) and send out their metrics. Prometheus on the other hand is a pull based server which expects monitored servers to provide a web interface from which it can scrape data. There are several [exporters available](http://prometheus.io/docs/instrumenting/exporters/) for Prometheus which will capture metrics and then expose them over http for Prometheus to scrape. In addition there are [libraries](http://prometheus.io/docs/instrumenting/clientlibs/) which can be used to create custom exporters. As we are concerned with monitoring docker containers we will use the [container\_exporter](https://github.com/docker-infra/container_exporter) capture metrics. Use the command shown below to bring up the container-exporter docker container and browse to http://MONITORED\_SERVER\_IP:9104/metrics to see the metrics it has collected for you. You should launch exporters on all servers in your deployment. Keep track of the respective MONITORED\_SERVER\_IPs as we will  be using them later in the configuration for Prometheus.

## Sysdig Cloud

Sysdig cloud is a hosted service that provides metrics storage, aggregation, visualization and alerting. To get started with sysdig sign up for a trial account at [https://app.sysdigcloud.com](https://app.sysdigcloud.com/). and complete the registration form. Once you complete the registration form and log in to the account, you will be asked to Setup your Environment and be given a curl command similar to the shown below. Your command will have your own secret key after the -s switch. You can run this command on the host running docker and which you need to monitor. Note that you should replace the [TAGS] place holder with tags to group your metrics. The tags are in the format TAG\_NAME:VALUE so you may want to add a tag role:web or deployment:production. You may also use the containerized sysdig agent.

<http://rancher.com/comparing-monitoring-options-for-docker-deployments/>

## Logentries

Many of the folks getting into the Docker-monitoring game are longtime system monitoring mavens. [Logentries](https://logentries.com/" \t "_blank) is among them, and the company seems to have sensed which way the wind's blowing with containers. It recently rolled out a [free logging service for Docker containers](https://logentries.com/docker/), which hooks into a few common Docker logging APIs (log messages, container metrics, container detection) exposed by way of Docker 1.5 and up. A 30-day trial of Logentries' other (for-pay) services is included, but the container logging service itself remains free. Minor caveat: If you run Docker in a restricted environment, such as Google Compute Engine, you'll need to run the Logentries container in privileged mode.

## Sematext

An in-cloud and on-prem monitoring solution with [several integrations](http://sematext.com/spm/integrations/index.html), [Sematext](http://sematext.com/spm/integrations/docker-monitoring.html" \t "_blank) recently added [Docker monitoring](http://sematext.com/spm/integrations/docker-monitoring.html) to its [SPM](http://www.infoworld.com/cms/article/sematext.com/spm/index.html) solution. Like others in this vein, the monitoring is conducted via a container, with events piped back out to a variety of [sources](http://blog.sematext.com/2015/06/09/docker-monitoring-support/), and one agent can be used to monitor multiple servers. Also included is monitoring for [container-level events](http://blog.sematext.com/2015/06/24/docker-events-and-metrics-monitoring/) (stops, starts, and so on) and [managing the (sometimes copious) logs](http://blog.sematext.com/2015/08/12/docker-log-management/) generated by containers. Plans start at 3.5 cents per server-hour.