Overview

The following information is collected and transmitted to Sematext Cloud or Sematext Enterprise version. Sematext Cloud integration for Docker uses the open-source Docker monitoring agent available on Docker Registry as a ready-to-go sematext-agent-docker image.

Type

Description

Operating System Metrics

Host machine metrics

CPU Usage

Memory Usage

Network Stats

Disk I/O Stats

Docker Container Metrics/Stats

CPU Usage / limits

Memory Usage / Limits / Fail Counters

Network Stats

Disk I/O Stats

Events

Agent Startup Event

server-info – created by spm-agent framework with node.js and OS version info on startup. Please note the agent is implemented in node.js.

Docker-info – Docker Version, API Version, Kernel Version on startup

Docker Events

Container Lifecycle Events| create, exec_create, destroy, export, …

Container Runtime Events

die, exec_start, kill, pause, restart, start, stop, unpause, …

Docker Logs

Default Fields

hostname / IP address

container id

container name

image name

message

Log formats

(detection and log parsers)

NGINX

APACHE httpd, Kafka, Solr, HBase, Zookeeper, Cassandra

MySQL

MongoDB

Redis

Elasticsearch

NSQ / Nsq.io

patterns are maintained here:

https://github.com/sematext/logagent-js

JSON, Plain Text

Supported Platforms

- Docker >= 1.6
- Platforms using Docker:
  - Docker Cloud
  - Docker Data Center
  - Kubernetes
  - Mesos
  - CoreOS
  - Rancher
  - Amazon ECS
  - Red Hat OpenShift
  - DEIS PaaS

Installation and Configuration

1. Create an SPM App of type "Docker" in SPM
2. Click the "**Install Monitor**" button and follow the customized instructions for the created SPM App

Step 2) provides customized instructions (including the SPM App Token) for this general procedure:

Installation of the Docker Image of the monitoring agent:

```
docker pull sematext/sematext-agent-docker
```

Configuration during start of sematext-agent-docker:

- Set the SPM_TOKEN
- Pass the Docker UNIX domain socket to the container

```
docker run -d --name sematext-agent -e SPM_TOKEN=YOUR-SPM-TOKEN -v /var/run/docker.sock:/va
```

Configuration Parameters

Parameter / Environment variable	Description
Required Parameters	
SPM_TOKEN	SPM Application Token enables metric and event collection

Parameter / Environment variable	Description
LOGSENE_TOKEN	Logsene Application Token enables logging to Logsene, see logging specific parameters for filter options and Log Routing section to route logs from different containers to separate Logsene applications
`-v /var/run/docker.sock`	Path to the docker.sock socket (optional, if dockerd provides TCP on 2375, see also DOCKER_PORT and DOCKER_HOST parameter)

Parameter / Environment variable	Description
TCP and TLS connection	If the Unix socket is not available Sematext Agent assumes the Container Gateway Address (autodetect) and port 2375 as default (no TLS) - this needs no configuration. In case the Docker Daemon TCP settings are different, you have to configure the TCP settings. The TCP settings can be modified with the following parameters

Parameter / Environment variable	Description
DOCKER_HOST	E.g. tcp://ip-reachable-from-container:2375/ - default value ‘unix:///var/run/docker.sock’. When the Unix socket is not available the agent tries to connect to tcp://gateway:2375. In case a TCP socket is used there is no need to mount the Docker Unix socket as volume
DOCKER_PORT	Rancher Agent will use its gateway address (autodetect) with the given DOCKER_PORT
DOCKER_TLS_VERIFY	0 or 1
DOCKER_CERT_PATH	Path to your certificate files, mount the path to the container with “-v $DOCKER_CERT_PATH$:DOCKER_CERT_PATH”

Parameter / Environment variable	Description
Configuration via docker swarm secrets:	
CONFIG_FILE	Path to the configuration file, containing environment variables `key=value`. Default value: `/run/secrets/sematext-agent`. Create a secret with `docker secret create sematext-agent ./sematext-agent.cfg`. Start Sematext Docker agent with 'docker service create –mode global –secret sematext-agent –mount type=bind,src=/var/run/docker.sock,dst=/var/run/docker.sock sematext/sematext-agent-docker
Optional Parameters:	

Parameter / Environment variable	Description
–privileged	The parameter might be helpful when Sematext Agent could not start because of limited permission to connect and write to the Docker socket /var/run/docker.sock. The privileged mode is a potential security risk, we recommend to enable the appropriate security. Please read about Docker security: https://docs.docker.com/engine/security/security/
HOSTNAME_LOOKUP_URL	On Amazon ECS, a metadata query must be used to get the instance hostname (e.g. “169.254.169.254/latest/meta-data/local-hostname”)

Parameter / Environment variable	Description
HTTPS_PROXY	URL for a proxy server (behind firewalls)
LOGSENE_RECEIVER_URL	URL for bulk inserts into Logsene. Required for Sematext Enterprise (local IP:PORT) or Sematext Cloud Europe: https://logsene-receiver.eu.sematext.com
SPM_RECEIVER_URL	URL for bulk inserts into SPM. Required for Sematext Enterprise (local IP:PORT) or Sematext Cloud Europe: https://spm-receiver.eu.sematext.com/receiver/v1.

Parameter / Environment variable	Description
EVENTS_RECEIVER_URL	URL for SPM events receiver. Required for Sematext Enterprise (local IP:PORT) or Sematext Cloud Europe: https://event-receiver.eu.sematext.com
Docker Logs Parameters	
TAGGING_LABELS	A list of docker label names or environment variable names to tag container logs. Supporting wildcards e.g. TAGGING_LABELS='com.docker.swarm,*com.myapp.*'
Whitelist containers for logging	
MATCH_BY_NAME	Regular expression to white list container names

Parameter / Environment variable	Description
MATCH_BY_IMAGE	Regular expression to white list image names
Blacklist containers	
SKIP_BY_NAME	Regular expression to black list container names
SKIP_BY_IMAGE	Regular expression to black list image names for logging
PATTERNS_URL	Load pattern.yml via HTTP e.g. `-e PATTERNS_URL=https://raw.githubusercontent.com/sematext/lo`
LOGAGENT_PATTERNS	Pass patterns.yml via env. variable e.g. `-e LOGAGENT_PATTERNS="$(cat ./patters.yml)"`

Parameter / Environment variable	Description
PATTERN_MATCHING_ENABLED	Activates logagent-js parser, default value is `true`. To disable the log parser set the value to `false`. This could increase the throughput of log processing for nodes with a very high log volume.
-v /yourpatterns/patterns.yml:/etc/logagent/patterns.yml	to provide custom patterns for log parsing, see logagent-js
-v /tmp:/logsene-log-buffer	Directory to store logs, in a case of a network or service outage. Docker Agent deletes these files after successful transmission.
GEOIP_ENABLED	Enables GeoIP lookups in the log parser, default value: `false`

Parameter / Environment variable	Description
MAXMIND_DB_DIR	Directory for the Geo-IP lite database, must end with `/`. Storing the DB in a volume could save downloads for updates after restarts. Using `/tmp/` (ramdisk) could speed up Geo-IP lookups (consumes add. ~30 MB main memory).

Parameter / Environment variable	Description
ENABLE_LOGSENE_STATS	Enables logging of transmission stats to Logsene. Default value 'false'. Provides a number of logs received, a number of logs shipped, number of failed/successful HTTP transmissions (bulk requests to Logsene) and retransmissions of failed requests.

Docker Swarm and Docker Enterprise

Connect your Docker client to Swarm or UCP remote API endpoint and deploy Sematext Docker Agent with following docker command with your SPM and Logsene token:

```
docker service create -mode global -name sematext-agent-docker \
-mount type=bind,src=/var/run/docker.sock,dst=/var/run/docker.sock \
-e SPM_TOKEN="REPLACE THIS WITH YOUR SPM TOKEN" \
-e LOGSENE_TOKEN="REPLACE THIS WITH YOUR LOGSENE TOKEN" \
sematext/sematext-agent-docker
```

Please refer to Monitoring and Logging for Docker Enterprise Edition for further information.

Kubernetes Support

Run Sematext Agent as Kubernetes DaemonSet.

1. Get a free account at sematext.com/spm
2. Create an SPM App of type "Docker" and copy the SPM Application Token
- For logs (optional) create a Logsene App to obtain an App Token for Logsene
3. Create sematext-agent.yml - and set your SPM and Logsene App Token in the section spec.env.
4. Run the DaemonSet

```
kubectl create -f sematext-agent.yml
```

CoreOS Support

To install SPM for Docker including log forwarding from journald execute these commands:

```
export $SPM_TOKEN=YOUR-SPM-TOKEN
export $LOGSENE_TOKEN=YOUR-SPM-TOKEN
etcdctl set /sematext.com/myapp/spm/token $SPM_TOKEN
etcdctl set /sematext.com/myapp/logsene/token $LOGSENE_TOKEN
wget https://raw.githubusercontent.com/sematext/sematext-agent-docker/master/coreos/sematext
fleetctl load sematext-agent.service; fleetctl start sematext-agent.service
wget https://raw.githubusercontent.com/sematext/sematext-agent-docker/master/coreos/logsene
fleetctl load logsene.service; fleetctl start logsene.service;
```

Please note the provided .service scripts use port 9000 for the logging service. The provided service templates could be changed after the download.

An alternative way to install the services is to include the content of the unit files in the cloud-init config file.

The latest documentation, install script, and service files are available in the Github repository

Access to the Docker Socket / Docker API

Please note the Docker Daemon can be configured to use Unix sockets (default), TCP sockets (default port 2375) and TLS sockets (authentication with certificates). Depending on your Docker setup, Sematext Agent needs to be configured to access the Docker Socket (API access).

Docker Unix Socket

Make sure that you have the permissions to access /var/run/docker.sock (or the actual location of the docker unix socket). E.g. use 'sudo' to run the "docker run" command.

Check your permissions first:

```
ls -la /var/run/docker.sock
srw-rw---- 1 root docker 0 Dec  3 07:52 /var/run/docker.sock
```

If you like to create a docker group, to access docker without super user permissions, see https://docs.docker.com/engine/installation/linux/docker-ee/ubuntu/

How to activate the Unix Socket in parallel to a TCP socket?
Check the configuration of the Docker Daemon in /etc/defaults/docker - it is possible to activate TCP and the Unix socket in parallel, simply add "-H unix:///var/run/docker.sock" and restart dockerd.

```
## /etc/defaults/docker
DOCKER_OPTS="-H tcp://0.0.0.0:2375 -H unix:///var/run/docker.sock"
```

Run Sematext Agent with access to the Unix socket:

```
docker run --name sematext-agent --restart=always \
-v /var/run/docker.sock:/var/run/docker.sock \
-e SPM_TOKEN=YOUR_SPM_TOKEN -e LOGSENE_TOKEN=YOUR_LOGSENE_TOKEN \
sematext/sematextagent-docker
```

Docker TCP Socket

When Sematext Agent can't find the Unix socket it tries to connect to Docker Daemon via TCP on port 2375. The parameter DOCKER_PORT specifies the TCP port of the local Docker Daemon (set in /etc/default/docker in DOCKER_OPTS). This setup is typically used in Docker Swarm Nodes (TCP port 2375).

Run Sematext Agent with Access to Docker TCP socket:

```
docker run --name sematext-agent -e DOCKER_PORT=2375 -e SPM_TOKEN=YOUR_SPM_TOKEN -e LOGSENE_
```

Relevant Parameters:

-e DOCKER_PORT - Sematext Agent will use the container gateway address (autodetect) with the given DOCKER_PORT

-e DOCKER_HOST - e.g. tcp://ip-of-docker-host-reachable-from-container-network:2375/

Docker TLS Socket

To access the Docker TLS socket (on port 2376 or 3376 for Docker Swarm Master), Sematext Agent needs access to the certifcates. Please use the following parameters to configure TLS access:

- -e DOCKER_HOST - e.g. tcp://ip-reachable-from-container:2375/
- -e DOCKER_TLS_VERIFY - 0 or 1
- -e DOCKER_CERT_PATH - path to your certificate files, mount the path to the countainer with "-v *DOCKER_CERT_PATH* :DOCKER_CERT_PATH"

Run Sematext Agent with access to Docker TLS socket:

```
# Example with docker-machine
docker-machine env --swarm swarm-master
# export DOCKER_TLS_VERIFY="1"
# export DOCKER_HOST="tcp://192.168.99.101:3376"
# export DOCKER_CERT_PATH="/Users/stefan/.docker/machine/machines/swarm-master"
# export DOCKER_MACHINE_NAME="swarm-master"
eval "$(docker-machine env swarm-master)"
docker run -d --name sematext-agent --restart=always
 -e SPM_TOKEN=YOUR_SPM_TOKEN  -e LOGSENE_TOKEN=YOUR_LOGSENE_TOKEN \
 -e DOCKER_TLS_VERIFY -e DOCKER_CERT_PATH -e DOCKER_HOST -v $DOCKER_CERT_PATH:$DOCKER_CERT_P
sematext/sematext-agent-docker
```

Log Routing

Routing logs from different containers to separate Logsene Apps can be configured via docker labels (or environment variables e.g. on Kubernetes). Simply tag a container with the label (or environment variable) `LOGSENE_TOKEN=YOUR_LOGSENE_TOKEN`. Sematext Agent inspects the containers for this Label and ships the logs to the defined Logsene App.

To disable logging to Logsene/Elasticsearch label the application container with `LOGSENE_ENABLED=false`. `LOGSENE_ENABLED=true` enables logging for the container again.

Example: The following command will start nginx webserver and logs for this container will be shipped to the related Logsene App.

```
docker run --label LOGSENE_TOKEN=REPLACE_WITH_YOUR_LOGSENE_TOKEN -p 80:80 nginx
# or use environment variable on Kubernetes (no support for Docker labels)
# docker run -e LOGSENE_TOKEN=REPLACE_WITH_YOUR_LOGSENE_TOKEN -p 80:80 nginx
```

All other container logs will be shipped to the Logsene App specified in the docker run command for `sematext/sematext-agent-docker` with the environment variable `LOGSENE_TOKEN`.

Integrated Log Parser

SPM for Docker recognizes log formats - so your logs arrive in a structured format in Logsene! The format recognition, data extractions, date parsing etc. is provided by logagent-js and covers:

- Format detection e.g. for
- Mongo DB
  - Nginx
  - Apache httpd, Kafka, Cassandra, HBase, Solr, Zookeeper
  - MySQL
  - Redis

- plain text log messages
- line delimited JSON logs
- GeoIP enrichment for webserver logs or any other field defined in the pattern definitions

To use a custom pattern definition simply mount a volume to '/etc/logagent/patterns.yml':

```
-v /mydir/patterns.yml:/etc/logagent/patterns.yml
```

Feel free to contribute to logagent-js to enrich the default pattern set.

Known Issues

Conflict with Docker logging-drivers. Sematext Docker Agent is running with a valid Logsene Token, but Logsene does not show container logs.

Please note that Sematext Docker Agent collects logs via Docker Remote API. If you use a Docker logging-driver other than the default json-file driver, logs will not be available via the Docker Remote API. Please make sure that your container or docker daemon uses json-file logging driver. This ensures that logs are exposed via Docker Remote API. To check, run the "docker logs" command. If "docker logs CID" is shows container logs then Sematext Docker Agent should be able to collect the logs as well.

Troubleshooting and How-To

The following command enables **debug** information to stdout - to be displayed with "docker logs container_id_of_sematext-agent-docker":

```
docker run -d --name sematext-agent -e SPM_TOKEN=YOUR-SPM_TOKEN -e spmagent_logger__console=
docker logs sematext-agent
```

Parameters for debug output:

```
-e SPM_LOG_TO_CONSOLE=true - enables internal log messages to the console. Normally only met
-e SPM_LOG_LEVEL=debug - "info|warn|error|debug" - set this to "debug" to see all messages o
-e DEBUG_SPM_LOGGING=enabled - very detailed logging before parsing, after parsing, inserts
```

If running Sematext Docker Agent in debug mode doesn't help you spot and solve the problem please send us the diagnostics package as described below.

Run the following to collect basic information for our support, such as environment variables, and configuration:

```
$ docker exec -it sematext-agent spm-client-diagnostics
...
SPM diagnostics info is in /tmp/spm-diagnose.zip
Please e-mail the file to support@sematext.com
```

Please contact us via chat or email us the output of that command and the generated ZIP file (to support@sematext.com). You can copy the ZIP file to your host using "docker cp":

```
docker cp sematext-agent:/tmp/spm-diagnose.zip .
```

Github Repository

Latest information for sematext-agent-docker and open issues

Integration

- Instructions: https://apps.sematext.com/ui/howto/Docker/overview

Metrics

Metric Name	Key	Agg	Type	Description
container count	docker.containers	Avg	Long	
write wait time	docker.io.write.wait.time	Sum	Double	
write time	docker.io.write.time	Sum	Double	
read	docker.io.read	Sum	Long	
read wait time	docker.io.read.wait.time	Sum	Double	
read time	docker.io.read.time	Sum	Double	
write	docker.io.write	Sum	Long	
mem fail counter	docker.memory.fail.count	Sum	Long	
pages in	docker.swap.pages.in	Sum	Long	
mem usage	docker.memory.usage	Avg	Long	
pages fault	docker.swap.pages.fault	Sum	Long	
pages out	docker.swap.pages.out	Sum	Long	
mem limit	docker.memory.limit	Avg	Long	
cpu throttled time	docker.cpu.throttled.time	Sum	Double	

Metric Name	Key	Agg	Type	Description
cpu usage	docker.cpu.percent	Avg	Double	
tx dropped	docker.network.tx.dropped	Sum	Long	
received	docker.network.rx.bytes	Sum	Long	
transmitted	docker.network.tx.bytes	Sum	Long	
received	docker.network.rx.packets	Sum	Long	
rx errors	docker.network.rx.errors	Sum	Long	
transmitted	docker.network.tx.packets	Sum	Long	
rx dropped	docker.network.rx.dropped	Sum	Long	
tx errors	docker.network.tx.errors	Sum	Long	