Overview

The following information is collected and transmitted to SPM (Cloud or On-Premises version):SPM 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
  - RancherOS
  - Amazon ECS
  - 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:/var/r
```

Environment Variables (passed by "-e" in the command line):

Environment Variable

Description

SPM_TOKEN

The identifier of your SPM App for Docker. SPM Overview of your Apps: https://apps.sematext.com/users-web/services.do#spm

HOSTNAME_LOOKUP_URL

On Amazon ECS the "HOSTNAME" variable is not avialable for container tasks. Please point HOSTNAME_LOOKUP_URL to "169.254.169.254/latest/meta-data/local-hostname"

HTTPS_PROXY

URL for a proxy server

Parameters for logging

LOGSENE_TOKEN

The identifier for a Logsene App

When this parameter is present all outputs of containers are shipped to Logsene by default

In addition a log forwarding service is started on exposed port 9000, to expose the service please use "-p YOUR_LOGGING_PORT:9000" The service accepts line delimited JSON (like logstash, bunyan format) or text input (like syslog lines) and forwards it to the specified Logsene Application Example:

or

Supported input formats:

journalctl -o short | short-iso | json

Messages from dockerd are parsed from MESSAGE field using time, level and message fields

We recommend to use the json option to make all fields available in Logsene

Bunyan JSON format (node.js)

Logstash JSON format

Syslog lines

Filter logs from specific containers

MATCH_BY_NAME

a regular expression to whitelist containers by name

MATCH_BY_IMAGE

a regular expression to whitelist contianers by image names

SKIP_BY_NAME

a regular expression to ignore containers by name

SKIP_BY_IMAGE

a regular expression to ignore containers by image names

DISABLE_DOCKER_LOGS

disables the collection of logs via Docker API (e.g. if you use the TCP service to provide logs)

REMOVE_ANSI_ESCAPE_SEQ

removes ANSI terminal escape sequences (e.g. colored logs) before the parsing starts. Default value: "enabled"

Custom pattern definitions for the log parser

Custom pattern definitions in a Docker volume

To use a custom pattern definition for the integrated log parser simply mount a file to '/etc/logagent/patterns.yml':

PATTERNS_URL

To use a custom pattern definition for the integrated log parser from the given URL to a patterns.yml file.

LOGAGENT_PATTERNS

To use a custom pattern definition for the integrated log parser via env. variable e.g. -e LOGAGENT_PATTERNS="$(cat ./patters.yml)"

PATTERN_MATCHING_ENABLED

Activate 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. Please note, logs are always tagged with container meta data (name, image, compose project/service etc.).

Volume mount to access Docker Unix Socket

-v PATH_TO_UNIX_DOCKER_SOCKET:/var/run/docker.sock - The monitoring container needs access to the hosts Docker socket to make be able to get the stats. The location in the container is /var/run/docker.sock.

The parameter –privileged might be helpful when Sematext Docker 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/

Settings for On Premises

LOGSENE_RECEIVER_URL

URL for bulk inserts into Logsene. Required only for Logsene On-Premises only.

SPM_RECEIVER_URL

URL for bulk inserts into SPM. Required only for SPM On-Premises.

EVENTS_RECEIVER_URL

URL for SPM events receiver. Required for SPM On-Premises only.

Geo-IP Configuration

GEOIP_ENABLED

true enables GeoIP lookups in the log parser, default value: false

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).

Docker Swarm, Docker UCP and Docker Datacenter

Connect your Docker client to Swarm or UCP remote API endpoint and deploy following docker-compose.yml file with your SPM and Logsene token:

```
sematext-agent-docker:
image: 'sematext/sematext-agent-docker:latest'
environment:
 - LOGSENE_TOKEN="REPLACE THIS WITH YOUR LOGSENE TOKEN"
 - SPM_TOKEN="REPLACE THIS WITH YOUR SPM TOKEN"
 - affinity:container!=*sematext-agent-docker*
  restart: always
  volumes:
    - /var/run/docker.sock:/var/run/docker.sock
```

Then use docker-compose scale to deploy the agent to each node.

```
docker-compose scale sematext-agent-docker=$(docker info | grep Nodes | awk '{ print $2 }')
```

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/ubuntulinux/#create-a-docker-group

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_TOK
```

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_PAT
sematext/sematext-agent-docker
```

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=tr
docker logs sematext-agent
```

Parameters for debug output:

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

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