



Monitoring System

User Guide v3.2 66

Table of Contents

Overview.....	1
Installation.....	1
Project structure.....	1
influxdb.....	1
grafana.....	2
docker-monitor.....	2
collectd.....	2
Monitoring plugins.....	3
How to create a plugin.....	3
Description of storing metrics in InfluxDB.....	4
Example 1. Influx structure for docker metrics is organized as.....	4
Example 2. Collectd metrics:.....	4
Description of Grafana settings.....	4
dashboard.json must be prepared as described on the official Grafana website:.....	4

Overview

Monitoring system is designed to collect and visualize data, helps to analyze the OS performance and performance of Smilart products. It has a web interface on the port 30000, which displays the following information by default:

1. CPU load
2. used and free memory size
3. free disk space
4. network performance
5. CPU load with a grouping of docker containers
6. used memory with a grouping of docker containers

Installation

Before installation, you need to check whether the timezone is configured using the utility:

```
date-time-config
```

This is important for displaying data on the web page.

Next, the system is installed using the command:

```
sam in monitor:3.2_66
```



The label 'com.smilart.monitoring=on' is set on the containers for which statistics will be collected.

Project structure

Monitoring system consists of the following components:

1. grafana
2. influxdb
3. docker-monitor
4. collectd

influxdb

A component that is the central element of Monitoring System. This is a database that aggregates events and stores them for a specific time (2 month). InfluxDB has the port 8083 for detailed

viewing of metrics. Users work with a more user-friendly interface on the port 30000 It is called Grafana.

grafana

Grafana accepts all data from the database and displays it according to the configured Dashboard diagrams. Diagrams are located in `/etc/monitor/dashboards/`.

1. Smilart.Docker

```
Information by docker containers:  
total memory usage graph for each container docker;  
total CPU usage graph for each container docker;  
individual graphs for loading memory and CPU.
```

2. Smilart.Hardware (default Dashboard)

```
Information by OS:  
CPU usage graph;  
memory usage graph;  
disk space graph;  
network load graph
```

INFO: Authorization is not required to view the metrics. But for their editing it is necessary to log in Grafana as `admin/admin` (by default).

docker-monitor

Collects statistics using the Docker API for a local server.

Information split into each container that is marked as `'com.smilart.monitoring=on'`:

1. CPU usage.
2. memory usage.
3. network load.

The resulting data is broadcast on the UDP 8086 port of InfluxDB.

collectd

Collectd collects hardware data on a processor, memory, network, and hard disk usage and transfers to InfluxDB.

Monitoring plugins

To write a plugin, user must know the monitoring structure. It consists of the following parts:

- grafana settings;
- format for storing metrics in InfluxDB;
- monitor container settings.

How to create a plugin

The purpose of installing the plugin is to automatically add Dashboards and DataSource to Grafana.

If you want to configure the monitor container to work with your Dashboards, you need:

- 1). Create a directory with the path `/etc/monitor/dashboards/<new directory>`
- 2). Create a file with DataSources settings from Grafana `/etc/monitor/dashboards/<new directory>/smilart.ds`

DataSource option in Grafana determines from where metrics for graphs will be taken.

The file `smilart.ds` is a special file. Its name may change but extension will always be `.ds`. This is how monitor container determines at the startup that this file is special. DataSource file has JSON format and it can be retrieved by request `"GET /api/datasources"` from Grafana REST API. DataSource file example:

```
{
  "name": "DataSource name from Grafana",
  "type": "influxdb - storage type",
  "url": "http://localhost:8086 - Grafana local address",
  "access": "proxy - not recommended to change this option",
  "isDefault": false - not recommended to change this option,
  "database": "sml_apps - DB name in InfluxDB",
  "user": "root - user of InfluxDB",
  "password": "root"
}
```

- 3). Create a file with Dashboard settings from Grafana at `/etc/monitor/dashboards/<new directory>/<new file>.json`

A file with Dashboards settings is created by standard Grafana tools. To do this you need to prepare Dashboards with metrics graphs in advance. Next, export it to a json format.

- 4). Apply settings

After creating files with Grafana settings, you should restart the monitor with the command:

```
systemctl restart monitor
```

Description of storing metrics in InfluxDB

The storage format is determined by the convenience of Dashboards creation in Grafana.

Influx web-interface is open on 8083 port and 8086 port is used for connecting with client applications.

Example 1. Influx structure for docker metrics is organized as

- DB name: docker;
- Tables name corresponds to a measurement name (cpu, memory);
- Columns name corresponds to a containers' name;
- Fields as metric values.

Example 2. Collectd metrics:

- DB name: collectd;
- Tables name corresponds to a measurement name (cpu, memory);
- Attributes represent columns' metric (host, type_instance, value);
- fields corresponds to attribute values and metric value.

Description of Grafana settings

dashboard.json must be prepared as described on the official Grafana website:

- Login to Grafana with admin on 30000 port. Default login is "admin" and password is "admin".
- Create DataSource for InfluxDB. DB should already has new metrics.
- Create Dashboard and graphics;
- Export to JSON file.