



Checkpoint Monitoring

User Guide v2.1 50.3

Содержание

Введение	2
Установка	2
Описание структуры	2
graphite	2
grafana	3
tracing-monitor	4
docker-monitor	4
collectd	5
archiver	5

Авторские права компании «Простые решения» защищены законом.

Несанкционированное копирование, демонстрация, распространение или другое использование разрабатываемых компанией продуктов и документации к ним в какой бы то ни было форме и любыми средствами являются противозаконными.

Введение

Система мониторинга(в дальнейшем, именуемая "монитор" или "система") предназначена для сборки и визуализации данных работы сервера, помогающая проанализировать работоспособность системы. Монитор имеет веб-интерфейс, доступ к которому осуществляется через порт сервера 30000, где отображаются следующие данные:

1. Загрузка процессора
2. Размер свободной и занимаемой оперативной памяти
3. Размер свободного пространства на диске
4. Производительность сетевого интерфейса
5. Загрузка процессора с группировкой по контейнерам докера
6. Загрузка оперативной памяти с группировкой по контейнерам докера
7. Информация об обработанных данных детектированием
8. Количество кадров пришедших с каждой камеры, детектов и идентификаций
9. И другое

Установка

Изначально настраивается актуальная тайм-зона. Это важно для отображения данных. Далее система устанавливается система с помощью команды: `sam in monitor_version`. На контейнеры, по которым будет собираться статистика, устанавливается метка `com.smilart.monitoring=on`.

Описание структуры

Система мониторинга(в дальнейшем, именуемая "монитор" и "система") состоит из группы компонент:

1. graphite
2. grafana
3. tracing-monitor
4. docker-monitor
5. collectd
6. archiver

graphite

Компонент, являющийся центральным звеном монитора. Он пропускает через себя все события и хранит их до определенного времени. Для получения данных от системы мониторинга Graphite нужно настроить на чтение обменника "graphite" сервера Rabbitmq и

порт 2003, для строковых данных.

grafana

Принимает от graphite все данные и отображает согласно настроенным схемам Dashboards. Схемы расположены в каталоге /opt/grafana/public/dashboards.

1. Smilart.Checkpoint.Daemons

Информация о загрузке демонов:

Графики с количеством запросов и ответов по каждому демону.
Сводный график ответов от демонов и общем количестве кадров.

2. Smilart.Checkpoint.Docker

Информация в разрезе контейнеров докера, отмеченных меткой:

Общий график загрузки памяти по каждому контейнеру докера, отмеченному заданной меткой.

Общий график загрузки процессора по каждому контейнеру докера, отмеченному заданной меткой.

Отдельные графики по загрузке памяти и процессора по контейнерам: `osgi`, `rabbitmq`, `mongodb` и `zookeeper`.

3. Smilart.Checkpoint.FPS

Информация в разрезе каждой камеры:

Графики по каждой камере с информацией о количестве кадров, найденных лиц и идентифицированных персон.

4. Smilart.Checkpoint.SystemOperational

Информация о работоспособности на системном уровне:

Панель с информацией об общей загрузке памяти контейнерами отмеченными меткой.

Панели с информацией о загрузке памяти по контейнерам: `rabbitmq`, `mongodb`, `zookeeper`, `osgi`.

Панель общая загрузка процессора контейнерами отмеченными меткой.

Панели загрузка процессора по контейнерам: `rabbitmq`, `mongodb`, `zookeeper`, `osgi`.

Панели частота кадров с каждой камеры в секунду.

Панели частота ответов от демонов FD и FR в секунду.

5. Smilart.Checkpoint

Сводная информация о работе системы:

График загрузки процессора.

График использования памяти.

График свободного места на диске.

Общий график загрузки процессора по контейнерам докера, отмеченных меткой.

Общий график использования памяти по контейнерам докера, отмеченных меткой.

График загрузки сети.

График количества ряда событий.

График ошибок детекта.

График процента протектированных кадров.

6. Smilart.Checkpoint.BusinessOperational(home)

Текущие данные(среднее за 5 последних секунд):

Панели для каждой камеры с текущими данными по количеству кадров, найденных лиц и идентифицированных персон.

tracing-monitor

Собирает и группирует события от КПП.

Подключается к обменнику системы трейсинга "platform.trace" и, согласно настройкам, вылавливает необходимые события, группирует их по свойству события, считает их количество за секунду и отправляет данные в обменник "graphite". Для работы компонента необходимо настроить трассировку событий КПП в rabbitmq. Для выбора отлавливаемых событий и свойства, по которому идет группировка, в папке /data/share/docker.monitor/meta находится файл настройки событий: metrics_description.csv В файле описана таблица, где первая колонка - это имя отслеживаемого события, вторая - свойство события, по которому идет группировка, третье - шаблон с _ для подстановки значения из свойства события, по которому идет группировка, в имя метрики. То есть автоматически генерируется метрика, в которой знак нижнего подчеркивания заменяется значением из свойства события из второй колонки данного файла. Имена, полученных метрик, начинаются trace.\$host.monitor.

docker-monitor

Собирает статистику с использованием Docker API для локального сервера.

Информация с разбиением по контейнеру, отмеченному меткой com.smilart.monitoring=on:

1. загрузка процессора
2. потребление памяти
3. использование сетевого интерфейса

Полученные данные транслируются на 2003 порт graphite. Имена, полученных метрик, начинаются с `docker.$host`.

collectd

Собирает аппаратные данные об использовании процессора, памяти, сети и жесткого диска и передает в graphite. Имена, полученных метрик, начинаются с `collectd.$host`.

archiver

Подсчитывает количество событий трейсинга с обменника "platform.trace" за секунду и передает в обменник "graphite". Имена, полученных метрик, начинается с `trace.$host`.