

如何选择Docker监控方案



刘斌

2016/4/23

自我介绍



Cloud Insight, 蓝海讯通

<https://cloud.oneapm.com/>

Agenda

- 什么是监控
- Docker监控原理
- Docker监控方案

为什么监控

- 尽在掌握
- 抱着炸药桶睡觉

监控目的

- 减少宕机时间
- 扩展和性能管理
- 资源计划
- 识别异常事件
- 故障排除、分析

监控层次

- 硬件
- OS、中间件 (MySQL、Tomcat)
- 应用程序
- RUM

Docker监控的挑战

- Docker像host但不是host
- 量大容器
- 生命周期短
- 监控盲点（断层）
- 微服务
- 集群
- 全方位：Host（VM） + Services + Containers + Apps

Docker监控内容

- 配置信息
- Logs
- 主机和Daemon日志
- 容器信息
- Metric (performance)
- Event

内部监控 VS 外部监控

- 在容器内部监控
- 在宿主机上监控

Docker监控指标项目

- CPU
- memory usage
- memory limit
- network IO

Docker监控基础

- docker stats
- Remote API
- 伪文件系统

docker stats

```
$ docker stats redis1 redis2
```

CONTAINER	CPU %	MEM USAGE/LIMIT	MEM %	NET I/O
redis1	0.07%	796 KB/64 MB	1.21%	788 B/648 B
redis2	0.07%	2.746 MB/64 MB	4.29%	1.266 KB/648 B

伪文件系统

- CPU、内存、磁盘
- 网络

Cgroups

- CPU、内存、磁盘
- `/sys/fs/cgroup/{memory,cpuacct,blkio}/system.slice/${docker ps --no-trunc}.scope`
- Standard: `/sys/fs/cgroup/:cgroup/docker/:container_id`
- Systemd: `/sys/fs/cgroup/:cgroup/system.slice/docker-#{id}.scope`

Memory

- memory.stat

```
... ..  
cache 11492564992  
rss 1930993664  
swap 0  
pgfault 728281223  
... ..  
total_cache 11492564992  
total_rss 1930993664  
total_pgpgin 406632648  
total_pgpgout 403355412  
total_swap 0  
total_pgfault 728281223  
... ..
```

Memory指标含义

指标	具体含义
docker.mem.cache	该cgroup中进程使用的块设备的缓存大小
docker.mem.rss	该cgroup中进程使用的和磁盘无关的内存大小，比如堆和栈的内存
docker.mem.swap	该cgroup使用的交换空间的大小
docker.mem.active_anon	该cgroup中进程使用的和磁盘页无关、已被内核标记为 active 状态的内存大小。系统内存不足时，会将标记为 inactive 的页转移到交换区
docker.mem.inactive_anon	该cgroup中进程使用的和磁盘页无关、已被内核标记为 inactive 状态的内存大小。系统内存不足时，会将标记为 inactive 的页转移到交换区
docker.mem.active_file	该cgroup中进程使用的和磁盘页无关、已被内核标记为 active 状态的内存大小
docker.mem.inactive_file	该cgroup中进程使用的和磁盘页无关、已被内核标记为 inactive 状态的内存大小
docker.mem.mapped_file	在控制组中映射到进程的内存量
docker.mem.pgfault	cgroup中进程访问虚拟地址空间中不存在或者受保护内存导致的页面错误（page fault）的次数
docker.mem.pgmajfault	cgroup中进程访问虚拟地址空间中已经被交换出去或者指向映射文件而导致的页面错误（page fault）的次数
docker.mem.pgpgin	该cgroup中内存页被“charged”（添加到记账列表）中的次数
docker.mem.pgpgout	该cgroup中内存页被“uncharged”（添加到记账列表）中的次数
docker.mem.unevictable	该cgroup中进程使用的不可重用的内存大小。一般来说这部分内存被mlock“锁定”

CPU

- `cpuacct.stat`
- `docker.cpu.system`
- `docker.cpu.user`

Blkio

- *_recursive
- blkio.throttle.io_service_bytes
- blkio.throttle.io_serviced

网络数据

- iptables
- 网络设备接口
- 伪文件系统

数据源 (veth device)

```
$ CONTAINER_PID=`docker inspect -f '{{ .State.Pid }}' nginx`
```

```
$ mkdir -p /var/run/netns
```

```
$ ln -sf /proc/$CONTAINER_PID/ns/net /var/run/netns/$CONTAINER_ID
```

```
$ ip netns exec $CONTAINER_ID netstat -i
```

Docker容器网络信息-net namespace

- `$ CONTAINER_PID=`docker inspect -f '{{ .State.Pid }}' nginx``
- `$ cat /proc/$CONTAINER_PID/net/dev`

Docker容器网络信息-文件系统

```
$ pwd
```

```
/sys/class/net/veth559b656/statistics
```

```
$ ls
```

```
collisions  rx_crc_errors  rx_frame_errors  rx_packets      tx_compressed  tx_heartbeat_errors  
multicast   rx_dropped     rx_length_errors tx_aborted_errors tx_dropped     tx_packets  
rx_bytes    rx_errors      rx_missed_errors tx_bytes        tx_errors      tx_window_errors  
rx_compressed rx_fifo_errors rx_over_errors  tx_carrier_errors tx_fifo_errors
```

Docker容器网络接口数据的取得方式

```
package libcontainer

import "github.com/opencontainers/runc/libcontainer/cgroups"

type Stats struct {
    Interfaces []*NetworkInterface
    CgroupStats *cgroups.Stats
}
```

Docker容器网络接口数据的取得方式

```
package libcontainer

type NetworkInterface struct {
    // Name is the name of the network interface.
    Name string

    RxBytes    uint64
    RxPackets  uint64
    RxErrors   uint64
    RxDropped  uint64
    TxBytes    uint64
    TxPackets  uint64
    TxErrors   uint64
    TxDropped  uint64
}
```


Docker容器网络接口数据的取得方式

```
// Reads the specified statistics available under /sys/class/net/<EthInterface>/statistics
func readSysfsNetworkStats(ethInterface, statsFile string) (uint64, error) {
    data, err := ioutil.ReadFile(filepath.Join("/sys/class/net", ethInterface, "statistics", statsFile))
    if err != nil {
        return 0, err
    }
    return strconv.ParseUint(strings.TrimSpace(string(data)), 10, 64)
}
```

Docker监控方案

- 自己动手
- 开源软件
- SaaS

评价标准

- 功能
- 灵活性
- 运维

自己动手

- 灵活性强
- 成本高

自己动手打造监控方案

- 采集
- 存储
- 展示
- 报警（动作）

自己动手

- Docker remote API
- 200L
- Reports the resource usage of Docker containers to InfluxDB (<https://github.com/mustafaakin/docker-resource-reporter>)

性能指标采集

- tcollector
- StatsD
- collectd
- cAdvisor
-

StatsD

- Etsy/Flickr
- UDP/TCP
- 应用和协议

Tcollector

- 来源于OpenTSDB
- 数据采集框架

Collectd

- Statistics collection daemon
- 存储到RRD
- 插件机制 (input/output)
- 简单报警功能

cAdvisor (Container Advisor)

```
sudo docker run \  
  --volume=/:/rootfs:ro \  
  --volume=/var/run:/var/run:rw \  
  --volume=/sys:/sys:ro \  
  --volume=/var/lib/docker/:/var/lib/docker:ro \  
  --publish=8080:8080 \  
  --detach=true \  
  --name=cadvisor \  
  google/cadvisor
```



存储TSDB

- OpenTSDB
- Influxdb
- RRDTool
- Graphite
-

写入方式

- File、TCP/UDP
- HTTP
- JMX/JDBC/SNMP
- AWS/Docker/cAdvisor
- 消息队列（Kafka、ActiveMQ等）
-

数据展示

- Highcharts (Cloud Insight)
- D3 (Datadog)
- echarts
- Google Charts
- Charts.js
- n3-charts

开源可视化工具

- Graphite
- Influxdb + Grafana
- Prometheus



Grafana

开源方案

- cAdvisor (经典) + InfluxDB + Grafana
- Zabbix/Nagios/Hawkular
- Fluentd
- Prometheus
- Riemann
- ATSD (Axibase Time Series Database)
- Hawkular
- ELK

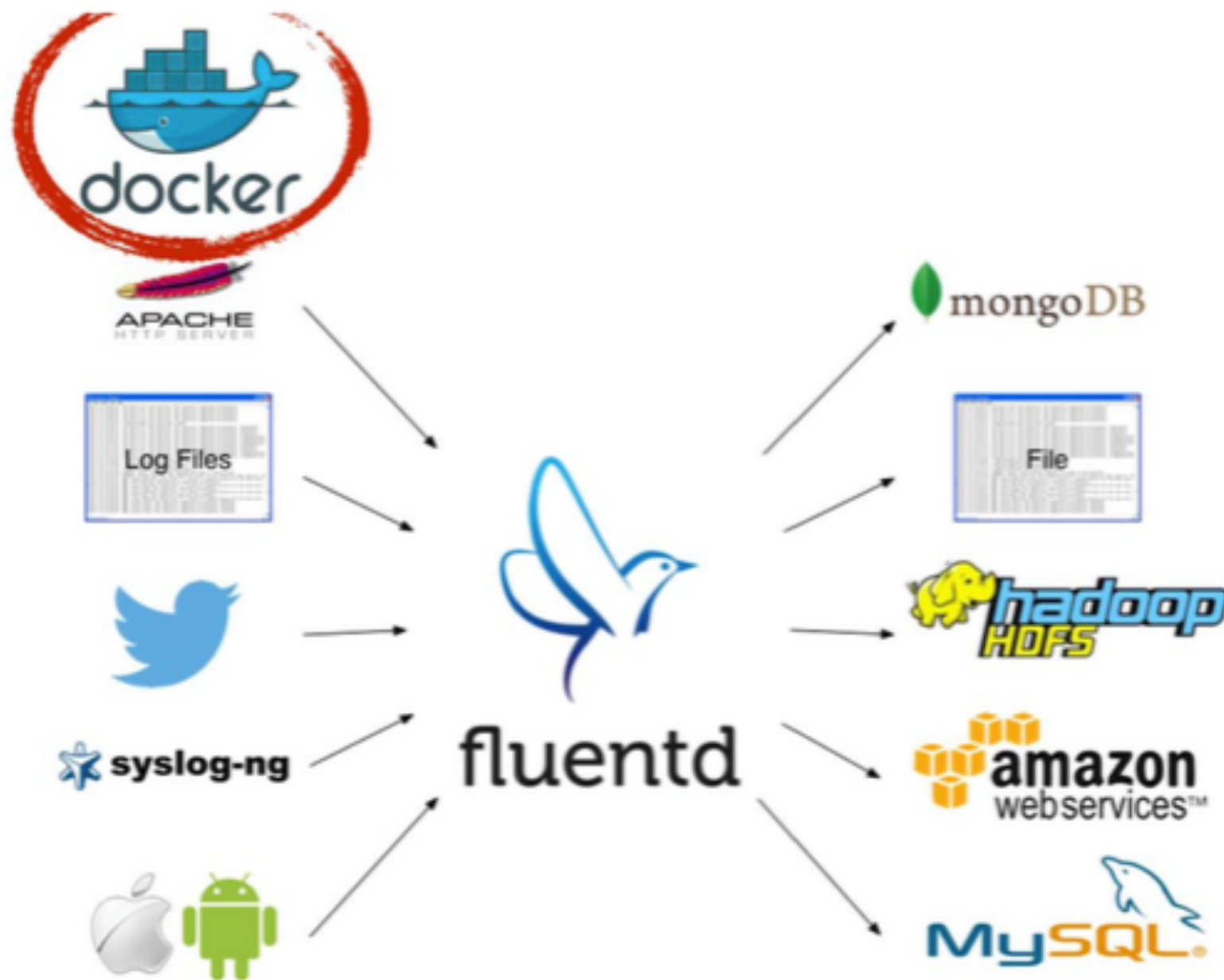
Zabbix

- 最经典（SaaS软件的最大敌人）
- 架构简单、清晰
- 文档丰富
- 包括采集、触发、告警
- agent支持用户自定义监控项
- 通过SNMP、ssh、telnet、IPMI、JMX监控

Zabbix Docker Monitoring

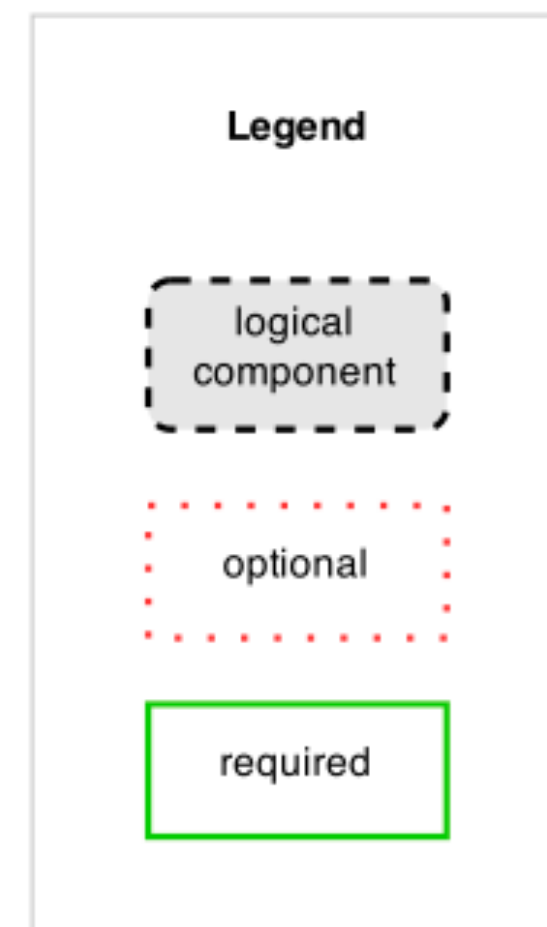
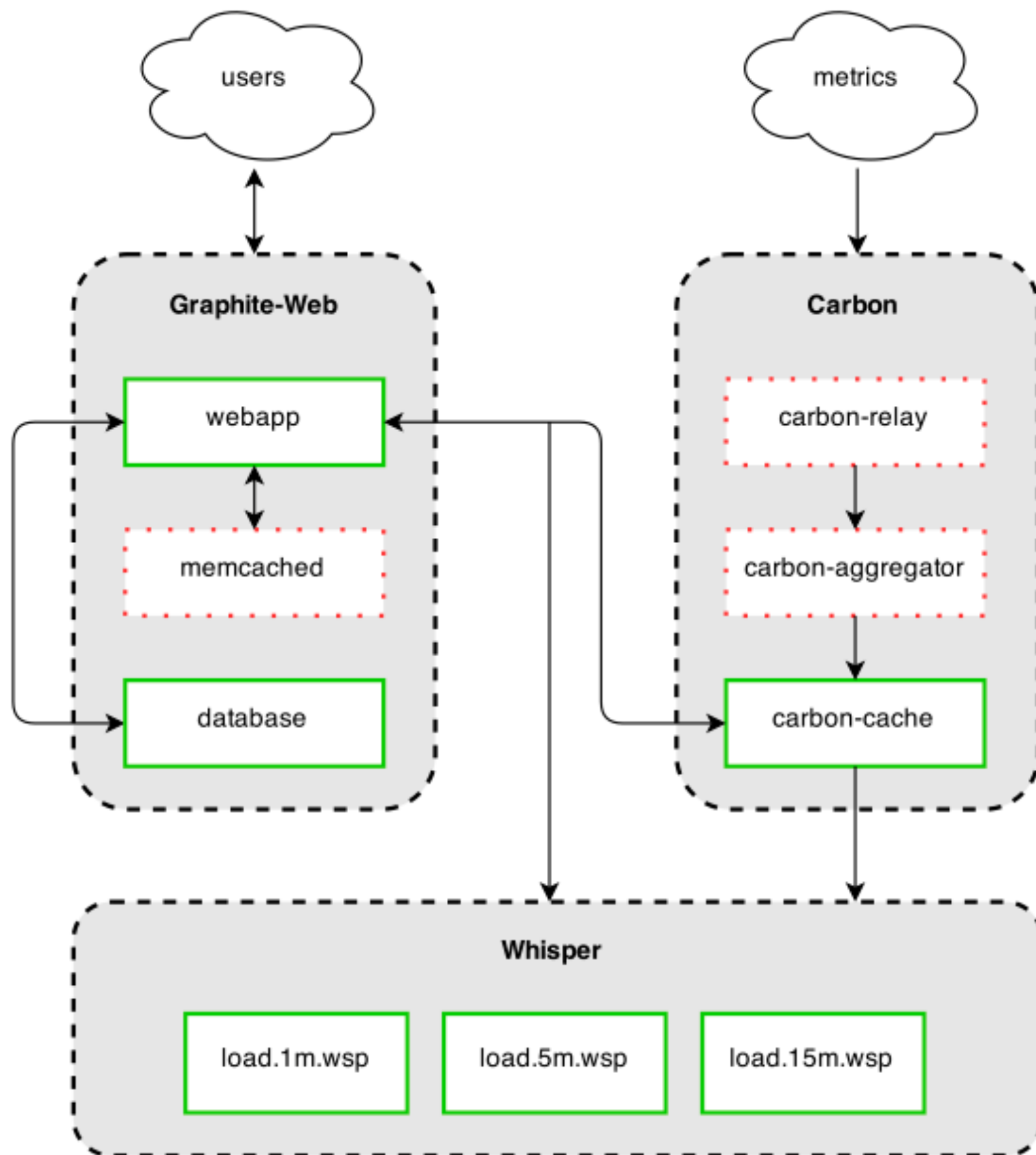
- <https://github.com/monitoringartist/zabbix-docker-monitoring>
- Zabbix Template

Fluentd



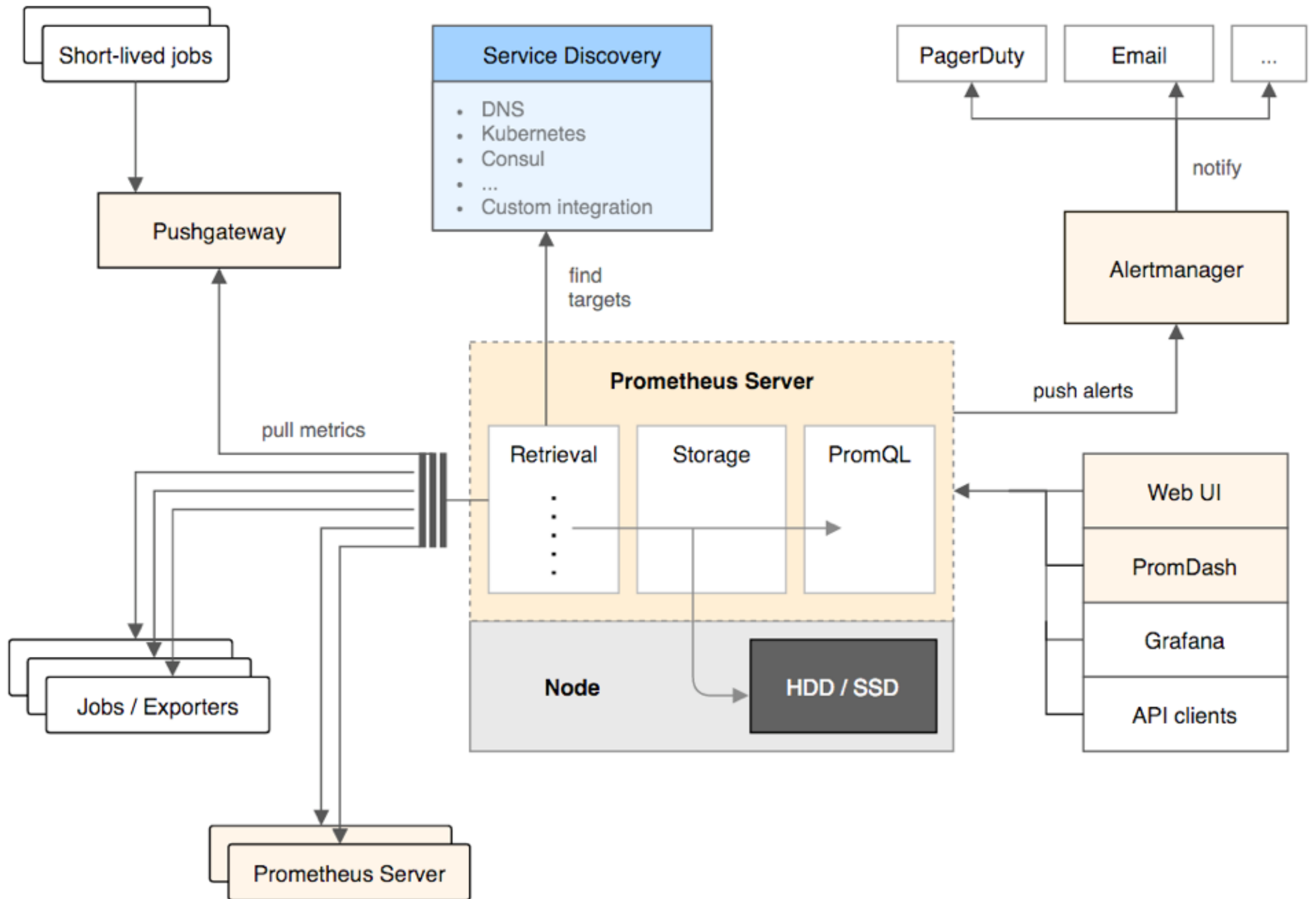
Graphite

- 存储数值型时序列数据
- 根据请求对数据进行可视化（画图）



Prometheus

- 一体化
- 多维度
- 灵活查询语言
- 仪表盘和告警
- LevelDB
- 非分布式、单机自治
- 基于HTTP的pull模式（push需要中间网关）



Riemann

- 事件处理
- Clojure实现
- Protocol Buffer
- 学习曲线？

Heapster

- K8s子项目
- source and sink
- 经典组合: heapster + Influxdb + grafana
- Sink : Kafka、 stdout、 gcm (Google Cloud Monitoring) 、 hawkular、 monasca、 riemann、 opentsdb

开源软件的问题点

- 灵活性受限于upstream
- 维护成本高
- 定制难度大
- 技术栈

SaaS

- turnkey解决方案
- 维护成本 ~ Zero
- 适合中小企业

SaaS

- New Relic
- AppDynamics
- Dynatrace (Ruxit)
- Datadog
- SysDig
- Cloud Insight
- clusterup
- Scout
- Librato

Datadog

- 国外最好
- 功能很强大
- 安装很简单
- 有点贵

Cloud Insight

- 实时数据
- 历史数据（免费版最大保存15天）
- 仪表盘
- 混合监控
- 报警功能

Cloud Insight Docker Overview

docker 仪表盘 By 曾

最新60分钟



添加监控图表

Running / Stopped container



☒ 提高可读性

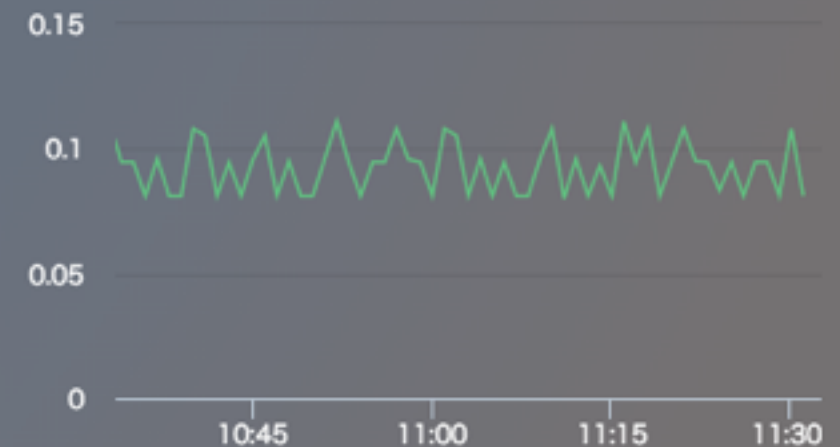
指标	值	标签
sum:docker.containers.running	1.00	*
sum:docker.containers.stopped	5.00	*

Memory by container

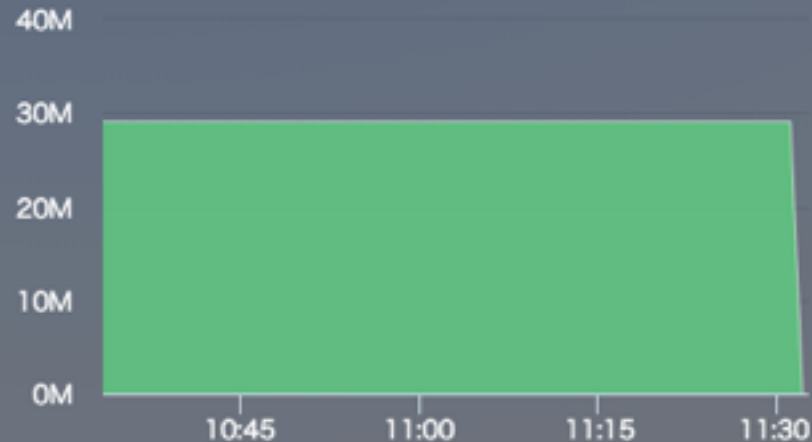


29.08 M

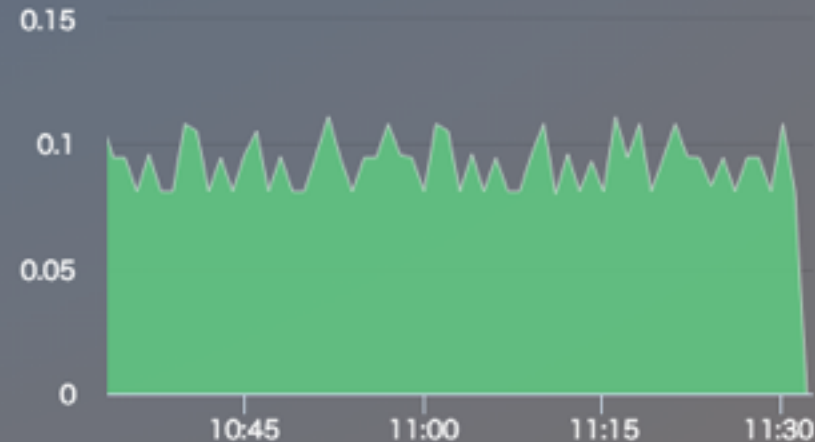
CPU by container



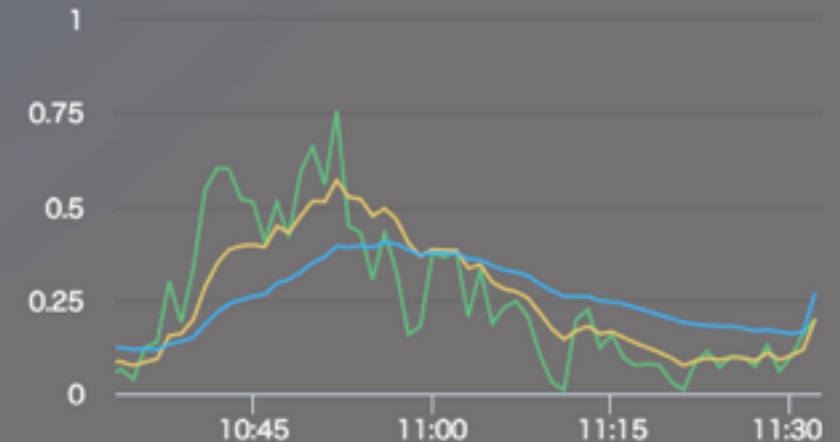
RSS memory by container

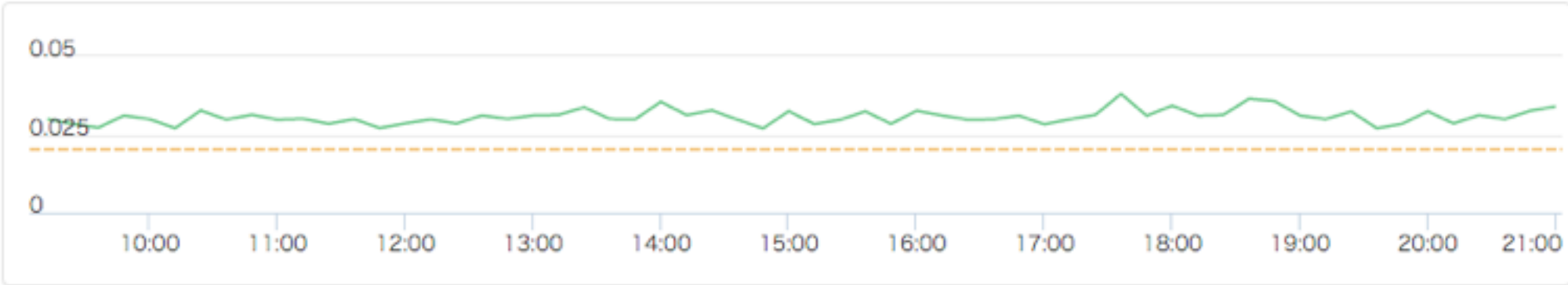


CPU user by Image



System load





1. 选择图表类型

- 时间序列
- 状态值
- 表格
- 饼图

2. 选择和编辑性能监控指标

Get

docker.cpu.system

From

Everywhere

Rate

☐

avg_by

×

docker_image

×

×

显示

曲线图

A

标记线

at

0.020

Show as

warning/orange

bold

×

☐ Label

y = 0.020

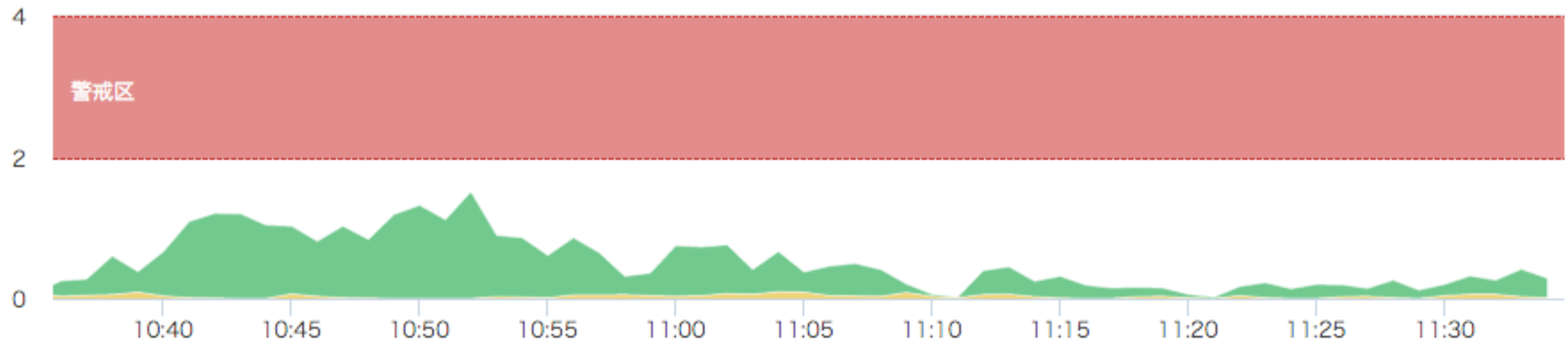
- ➕ 添加指标
- ➕ 添加标记
- ➕ 叠加事件

3. 图表命名

docker.cpu.system

Cloud Insight Docker Share

分享 system.load.1,system.mem.free



分享下面的连接，任何用户则可以不需要用户名密码看到此图表。

<https://cloud.oneapm.com/share/chart?token=418ab180a9c144ba80e729eac4e55658&width=600&height=300>



复制下面的连接，嵌入到自己的代码之中。

```
<iframe src="https://cloud.oneapm.com/share/chart?token=418ab180a9c144ba80e729eac4e55658&width=600&height=300" width="600" height="300" frameborder="0"></iframe>
```



扫描下面的二维码。



Cloud Insight Docker Event Stream

事件流 10 个事件

搜索

最新60分钟

状态

所有

Alert

Info



【告警触发】 0419测试线上报警

`avg(last_5m):avg:system.load.1 {*} >= 0`

最近触发时间: 2016/4/21 星期四 21:08

报警状态 编辑报警

更新于: 2016/4/21 星期四 21:08 创建于: 2016/4/19 星期二 19:16

【告警触发】 0330测试报警事件流-更新再次更新

`avg(last_5m):avg:system.load.1 {host:centos.license} >= 0`

最近触发时间: 2016/4/21 星期四 21:08

报警状态 编辑报警

更新于: 2016/4/21 星期四 21:08 创建于: 2016/4/20 星期三 11:31

Cloud Insight Docker Alert

未命名报警策略#1

最新60分钟 ↓



1. 选择性能指标

avg

system.load.1

over

2. 设置报警条件

该指标在 5分钟 内, 平均值 大于或等于 6 时, 触发报警。

1. 对稀疏的指标设置报警条件时, 建议使用「总计」或「至少一次」来设置; 因为使用「平均值」或「总是」的条件时, 只有当指标在选择的时段内是一条较为完整的曲线时, 才准确; 而针对稀疏的指标时, 触发会不准确。

2. 数值可接 k m M 等, 而 1k = 1000, 1m = 0.001, 1M = 1,000,000。










参考: [Wikipedia](#)

当数据丢失时, 不通知 用户。



















1. 如果所选的性能指标, 在正常情况下, 是需要一直保持有数据的情况, 那么请选择: 「当数据丢失时, 通知用户」。
如: 某个平台需要一直处于运行状态, 那么指标 system.cpu.idle 需要一直有数值与之对应; 此时, 请选择「通知用户」。
反之, 当一组云平台采用了 Auto Scaling 机制, 那么这些平台的指标是稀疏的。此时, 选择: 「当数据丢失时, 不通知用户」比较合理。

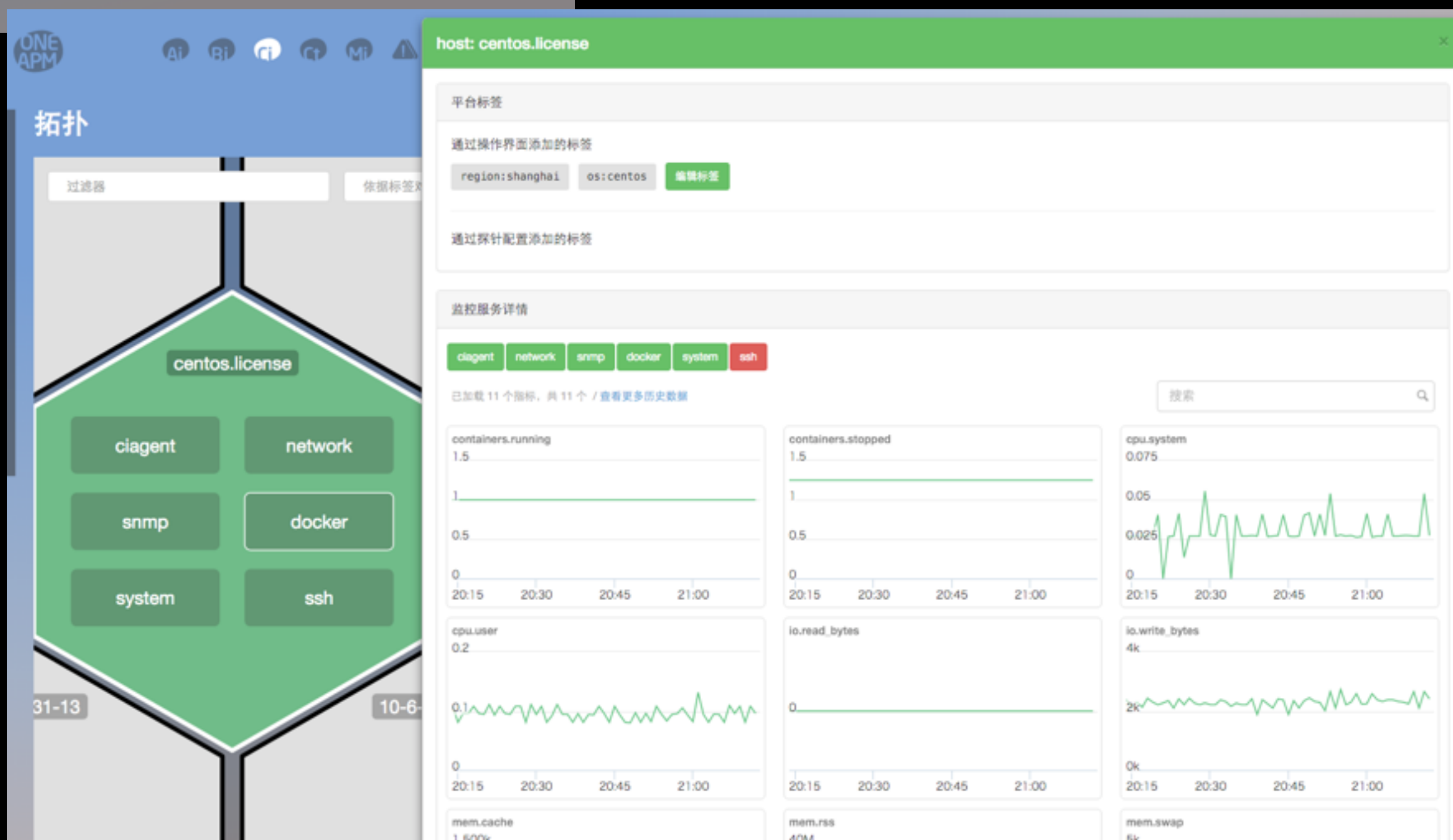
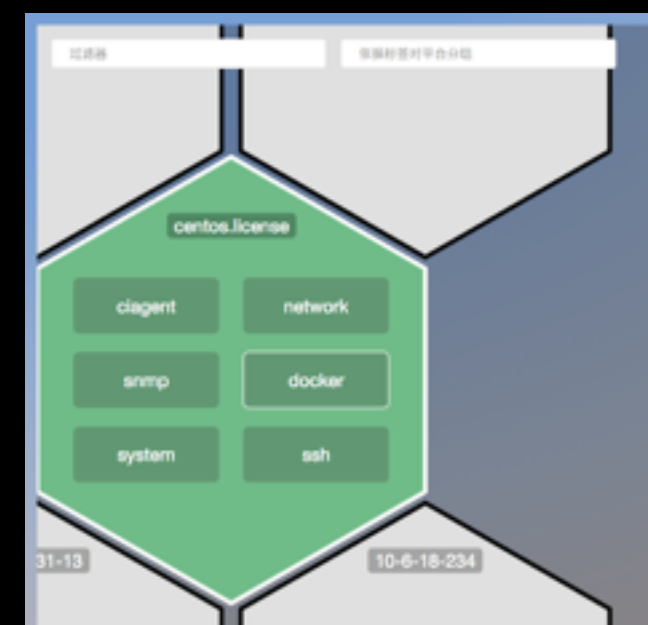
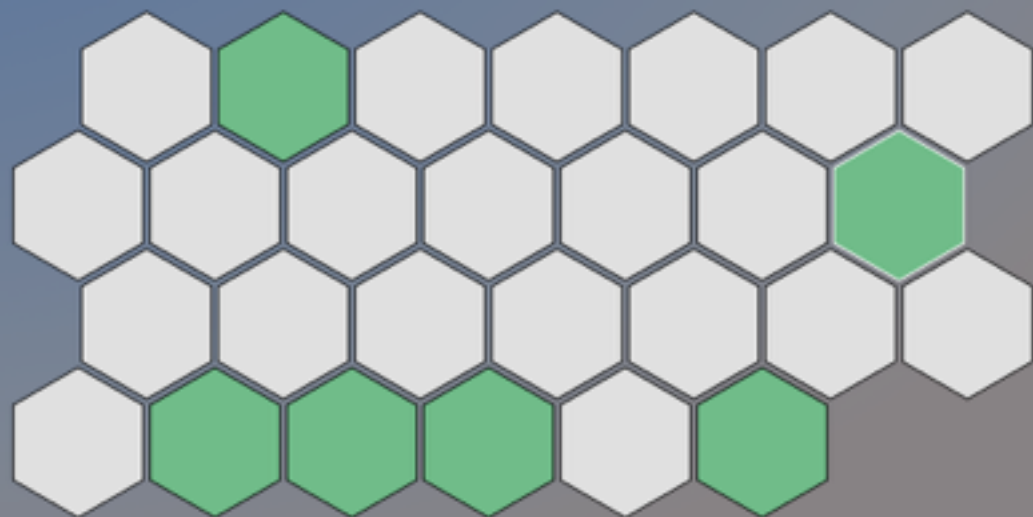
Cloud Insight Docker Integrations

已安装

 Apache	 Apache Tomcat	 Docker	 MongoDB	 MySQL	 Nginx	 PostgreSQL
 PHP FPM	 Redis	 ZooKeeper	 SQL Server	 Microsoft IIS	 BearyChat	 简聊

未安装

 阿里云	 ActiveMQ	 Apache Kafka	 Cassandra	 Couchbase	 CouchDB	 Elastic Search
 Memcached	 RabbitMQ	 Mesos	 Solr	 SNMP	 HAProxy	 Event Viewer
 WMI	 SSH	 HTTP	 TCP			



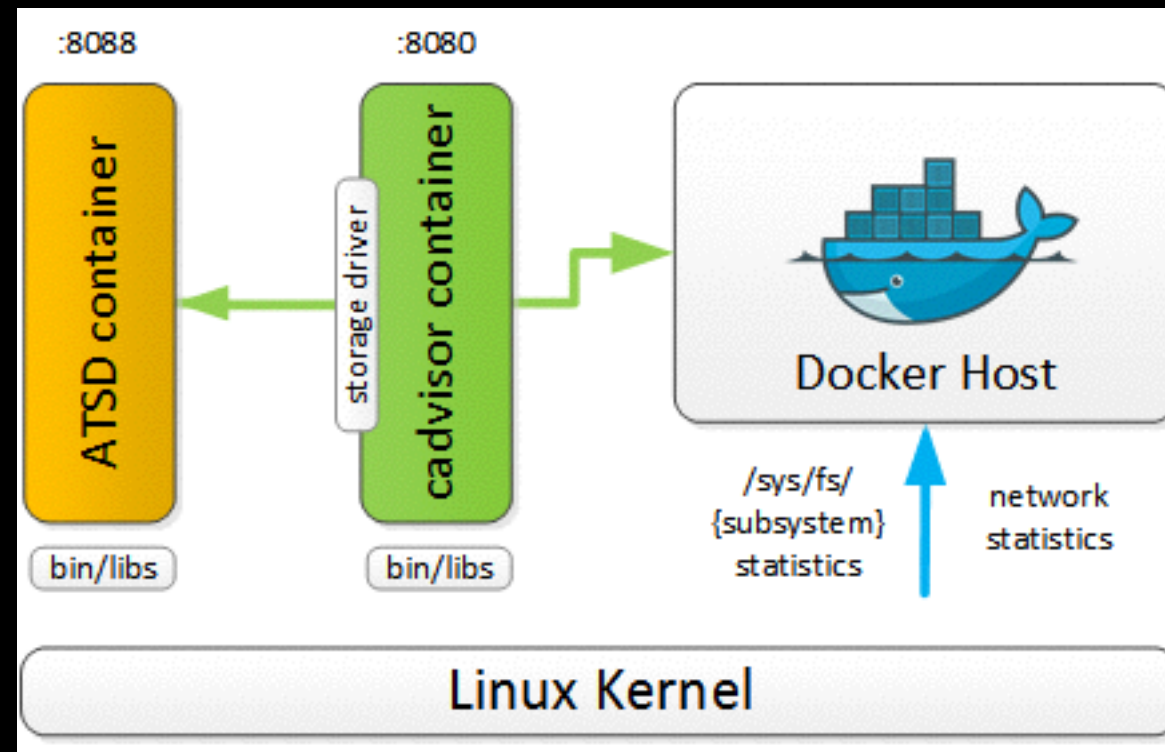
Sysdig

- 免费工具
- SaaS服务 Sysdig Cloud
- 拓扑可视化

Librato

- 数据聚合平台
- 简单探针
- 图表和报警
- 价格不贵

axibase (ATSD)



- 实际上是一个TSDB
- 支持报警
- 预测功能

SaaS的挑战

- 安全性
- 成本（迁移和使用成本）
- 和自有系统的兼容
- 内部抵抗（观念、个人爱好）

趋势

- 拓扑可视化
- 标签机制
- 通过API打通
- Total解决方案

参考资料

- Comparing Seven Monitoring Options for Docker : <http://rancher.com/comparing-monitoring-options-for-docker-deployments/>
- How to collect Docker metrics : <https://www.datadoghq.com/blog/how-to-collect-docker-metrics/>
- 时序列数据库武斗大会: <http://liubin.org/blog/2016/02/18/tsdb-intro/>
- Fluentd Docker Metrics Input Plugin : <https://github.com/kiyoto/fluent-plugin-docker-metrics>
- Docker Runtime metrics : <https://docs.docker.com/v1.8/articles/runmetrics/>

感谢您的倾听

