

K8S微讲堂 日志和监控

杨博华 Advisory Software Engineer

11/30/2017

“Kubernetes”系列公开课

- 每周四晚8点档

1. Kubernetes 初探
2. 上手 Kubernetes
3. Kubernetes 的资源调度
4. Kubernetes 的运行时
5. Kubernetes 的网络管理
6. Kubernetes 的存储管理
7. **Kubernetes 的日志与监控**
8. Kubernetes 的应用部署
9. 扩展 Kubernetes 生态
10. Kubernetes 的企业实践

课程Wiki: <http://ibm.biz/opentech-ma>

日志收集概况

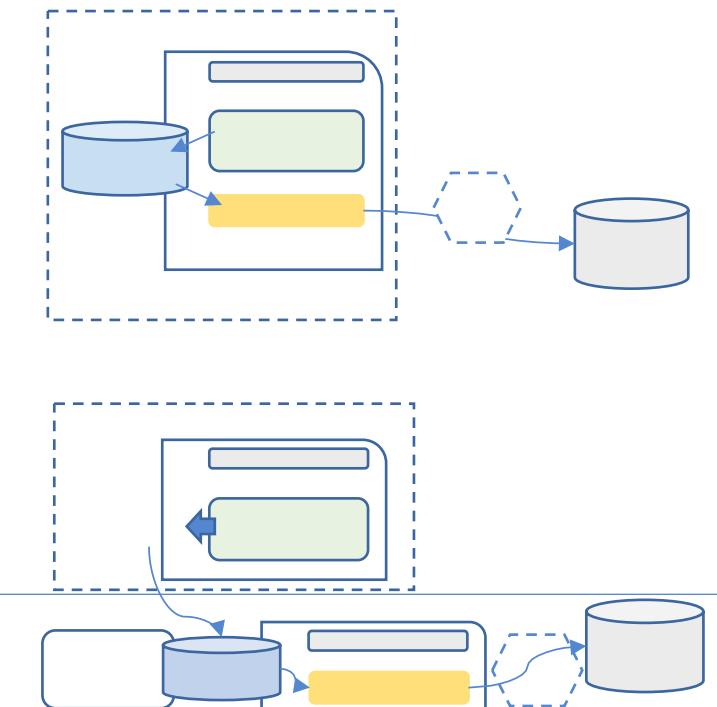
- 日志的分类
 - K8S 的日志
 - K8S Cluster 里面部署的应用程序的日志
- 分析处理
 - 常见的方案比如 ELK
- 收集日志是个挑战
- 不同的部署方案带来的日志收集挑战
 - K8S 的组件部署成 systemd service (二进制日志)
 - 需要对 systemd service 的日志进行单独收集
 - 增加额外的管理和维护成本
 - K8S 的组件部署成 Docker 容器 (使用 hyperkube 容器)
 - 可以和负载应用程序使用统一的收集方案
 - Docker 引擎支持 *-live-restore* 可以避免成为单点故障源

日志收集概况

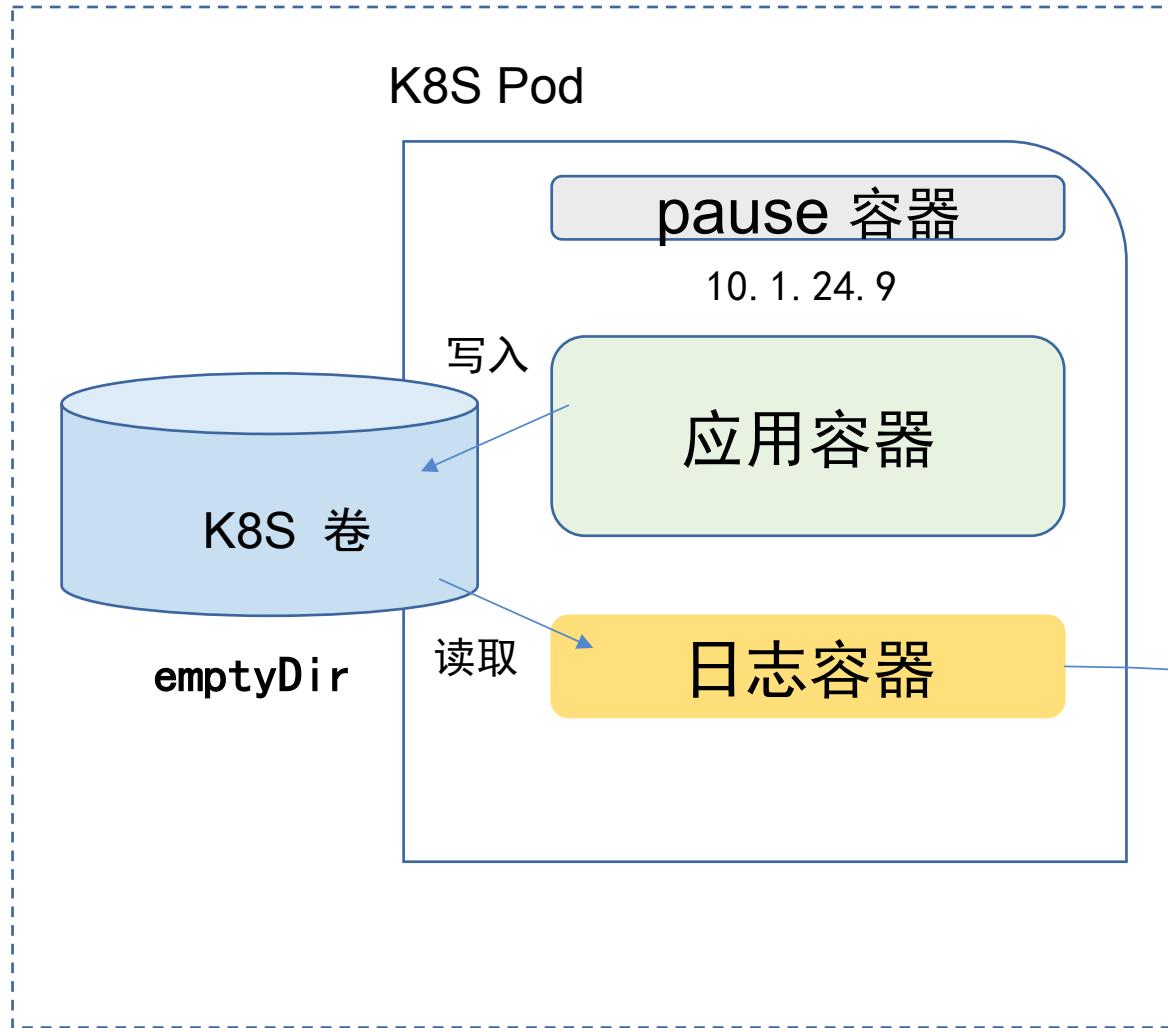
- 日志的分类
 - K8S 的日志
 - K8S Cluster 里面部署的应用程序的日志
- 分析处理
 - 常见的方案比如 ElasticSearch + LogStash + Kibana 的方案
- 收集日志是个挑战
 - 数量多 / 规模大 - 微服务/云化的应用数量动辄成千上万
 - 需要考虑性能 / 稳定性 / 扩展性 等等一系列问题
 - 容器化 / 隔离化 - 容器的进程和文件系统和宿主机是隔离的
 - 需要考虑如何收集容器化的应用产生的日志的问题
- 收集日志的目的
 - 日志不仅仅用于诊断，日志也是数据
 - 数据分析产生价值，集中分析产生更大价值

容器当中的日志怎么收集

- 让每个应用自行上传自己的日志
 - 增加应用复杂度 / 增加管理复杂度
- 附加专用日志上传容器（side-car模式）
 - 在每一个Pod中包含一个日志上传容器，
 - 应用容器和日志容器通过共享卷交换日志数据
- 使用Docker引擎的日志收集功能
 - 利用Docker Log Driver 收集每个容器的标准输出
 - 容器的标准输出会被写到宿主机的日志目录
 - 日志上传容器从宿主机的日志目录上传日志
 - 将整套日志收集系统从用户应用中分离出来



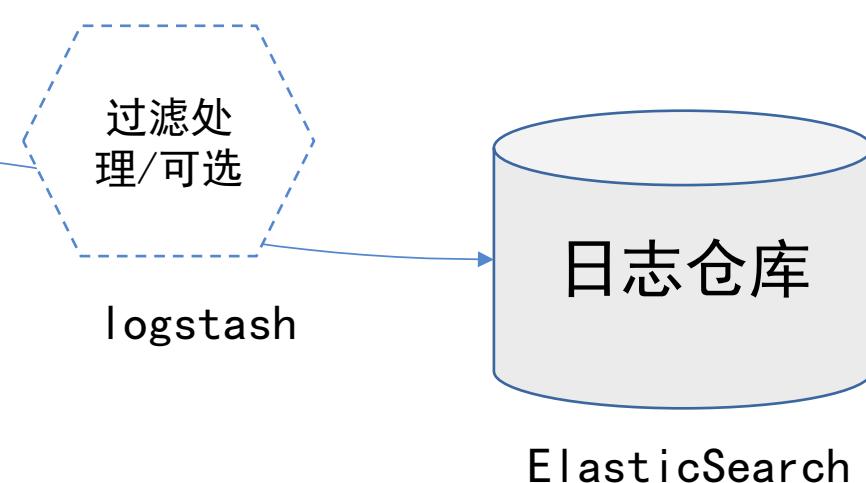
Sider-car 方案



虚线方框里的东西会作为一个整体被K8S调度
虚线方框里的东西共享相同的生命周期

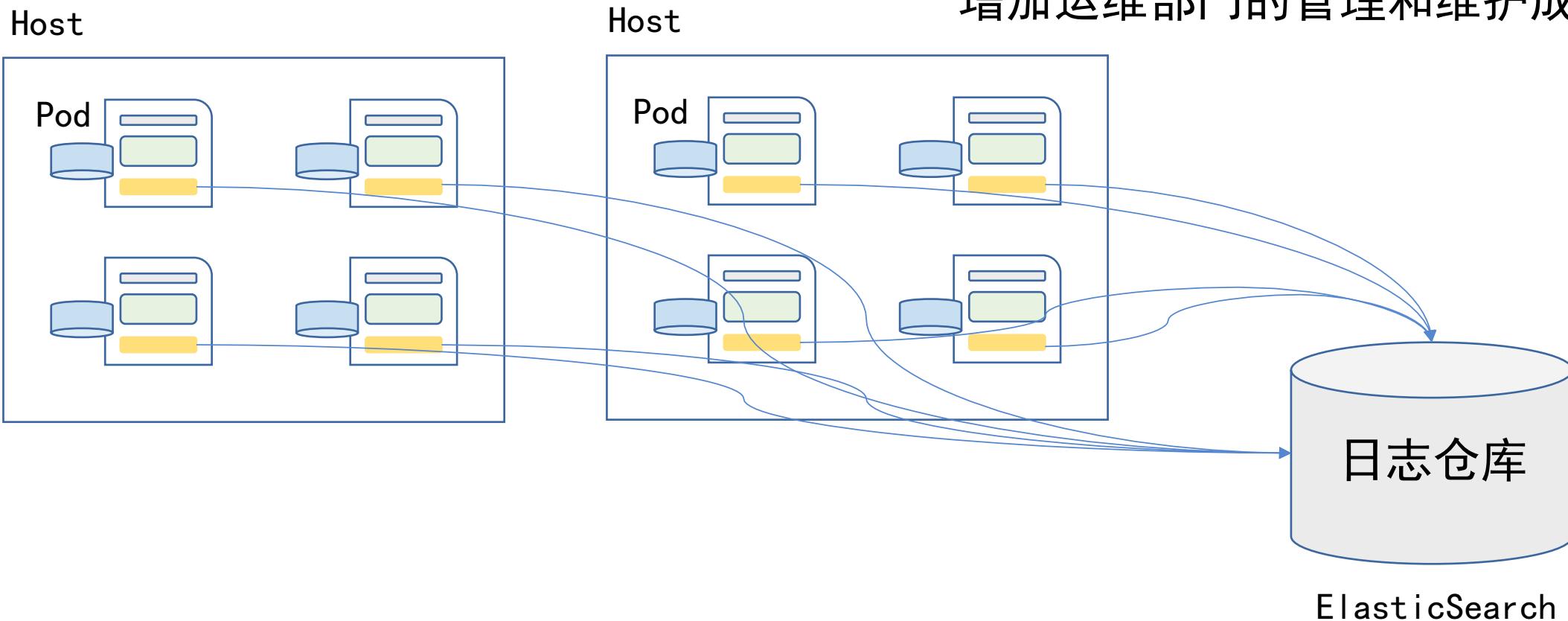
日志输出在卷上可以避免
容器内部文件系统无限增大的问题

需要针对不同的应用设置日志收集路径

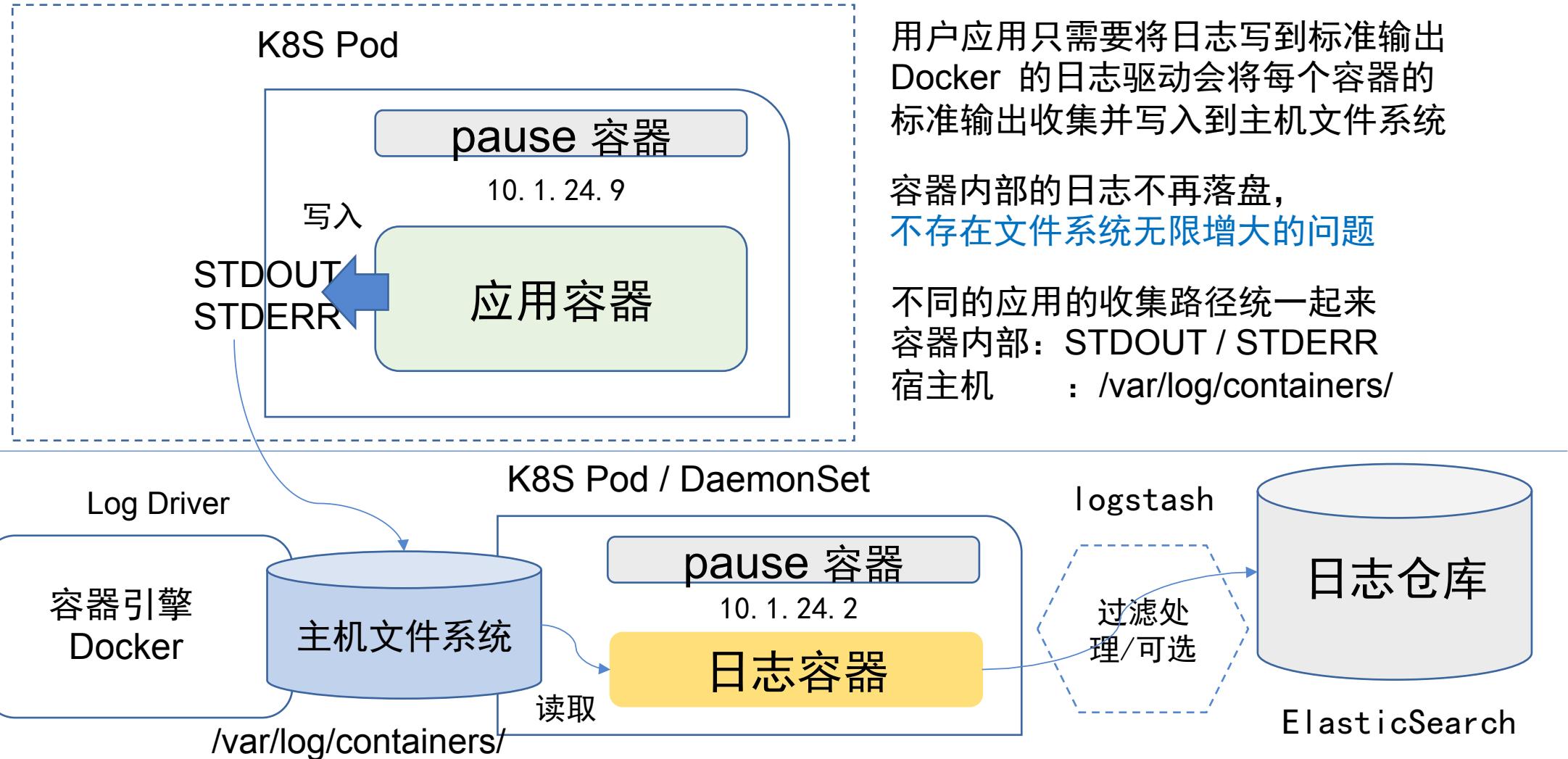


Sider-car 方案

每个应用都需要植入一个日志容器
针对不同应用的日志路径不同
需要设置日志收集配置
增加运维部门的管理和维护成本

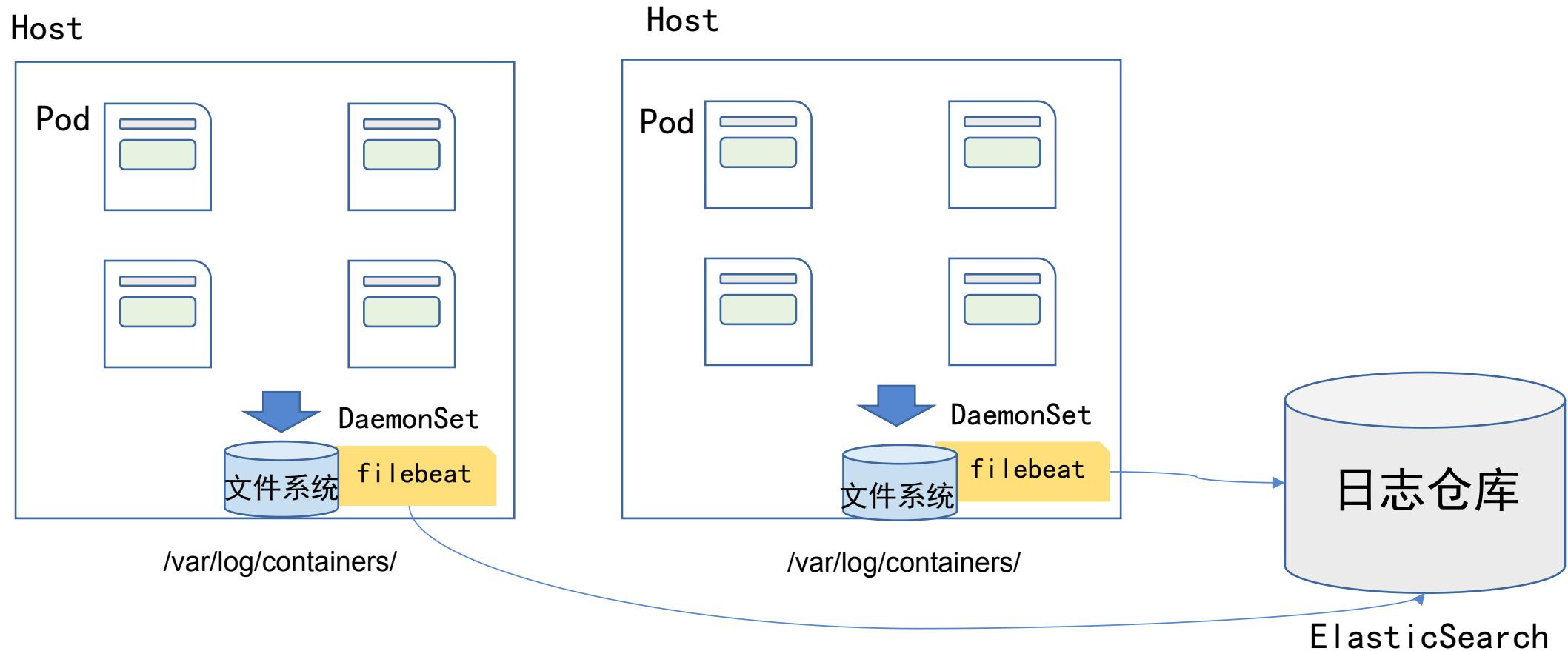


Docker 引擎 Log Driver 的收集方案

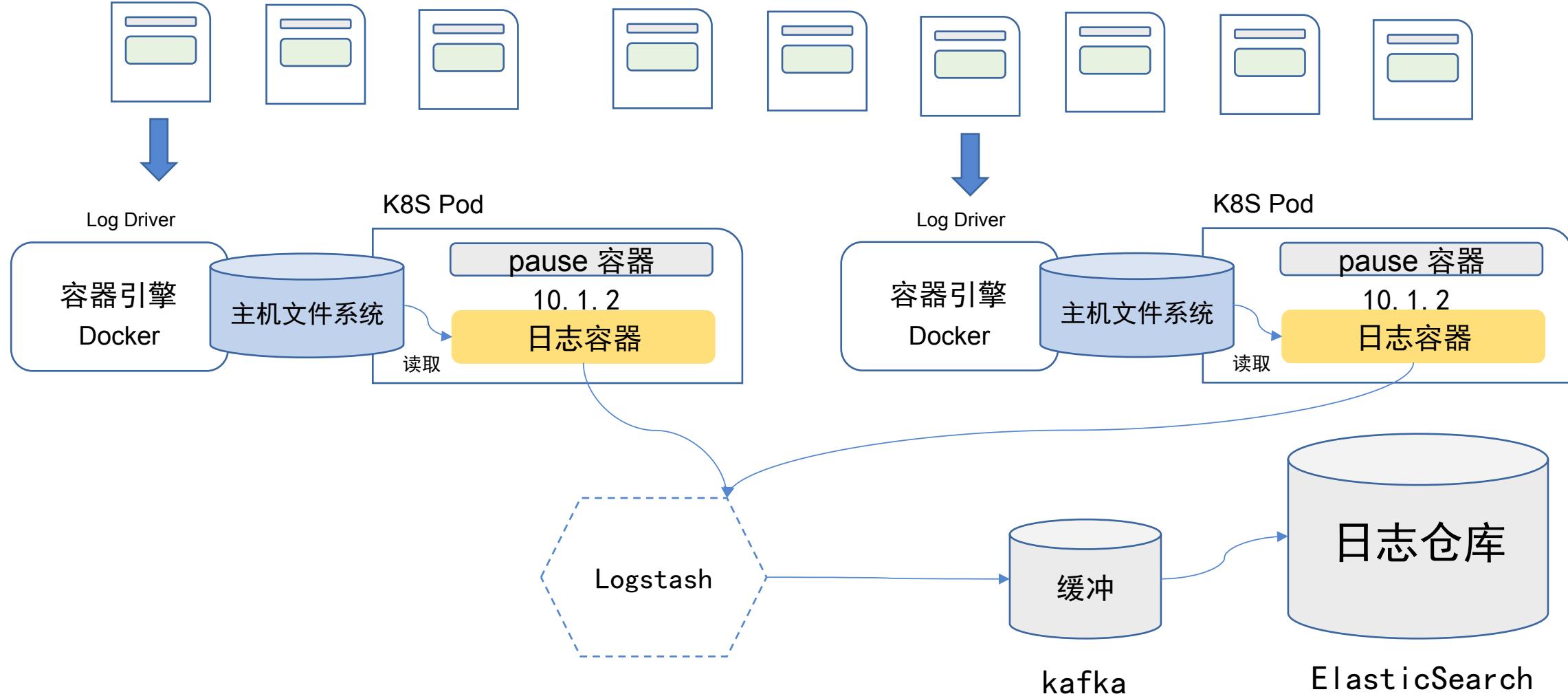


Docker 引擎 Log Driver 方案

日志收集容器完全从应用中独立出来
每台主机仅需部署一个日志容器



Docker 引擎的收集方案



Docker Supported Log Drivers

Driver	Description
none	No logs will be available for the container and <code>docker logs</code> will not return any output.
json-file	The logs are formatted as JSON. The default logging driver for Docker.
syslog	Writes logging messages to the <code>syslog</code> facility. The <code>syslog</code> daemon must be running on the host machine.
journald	Writes log messages to <code>journald</code> . The <code>journald</code> daemon must be running on the host machine.
gelf	Writes log messages to a Graylog Extended Log Format (GELF) endpoint such as Graylog or Logstash.
fluentd	Writes log messages to <code>fluentd</code> (forward input). The <code>fluentd</code> daemon must be running on the host machine.
awslogs	Writes log messages to Amazon CloudWatch Logs.
splunk	Writes log messages to <code>splunk</code> using the HTTP Event Collector.
etwlogs	Writes log messages as Event Tracing for Windows (ETW) events. Only available on Windows platforms.
gcplogs	Writes log messages to Google Cloud Platform (GCP) Logging.

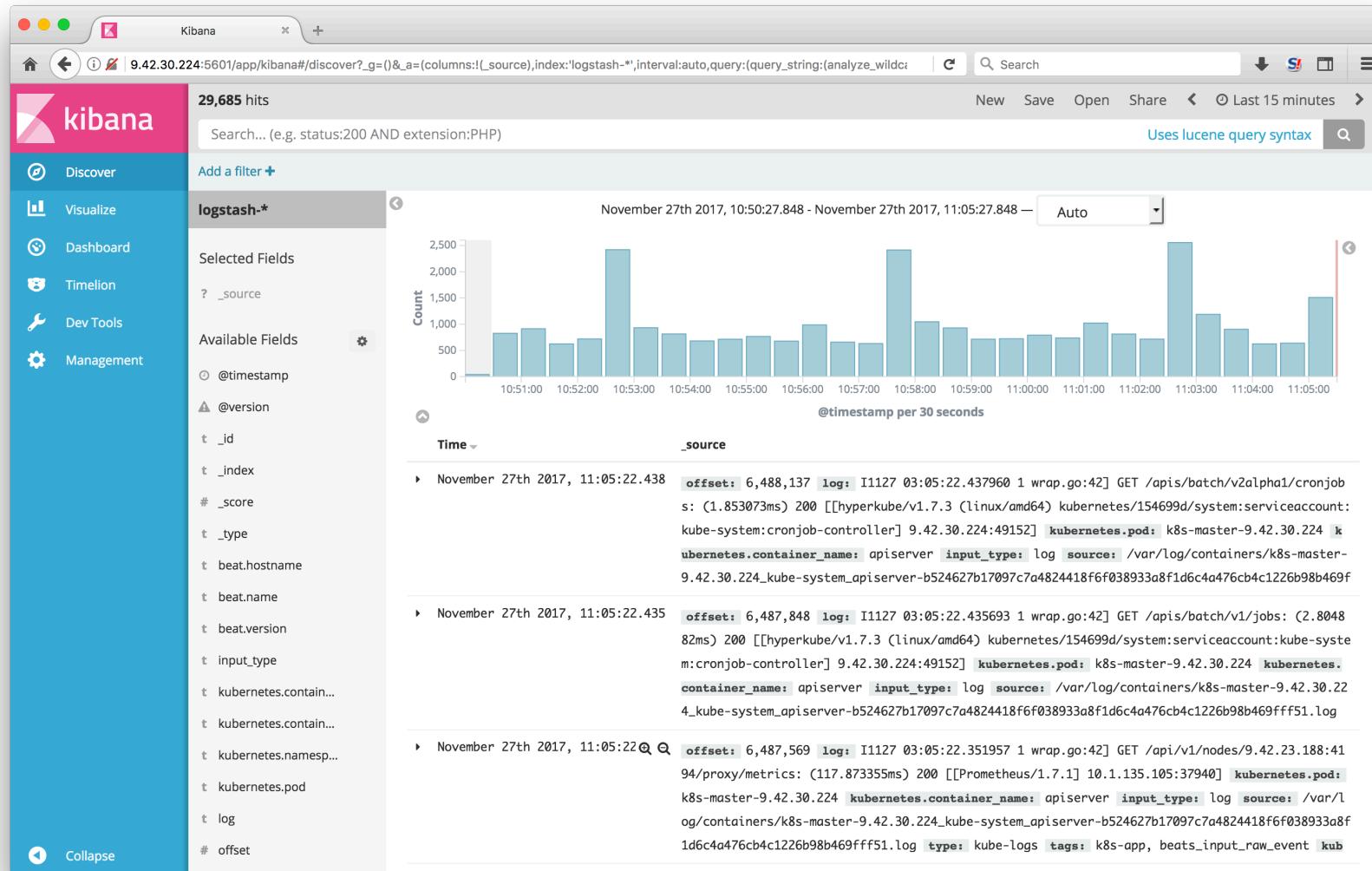
Limitations of logging drivers

The `docker logs` command is not available for drivers other than `json-file` and `journald`.

docker, logging, driver

<https://docs.docker.com/engine/admin/logging/overview/#supported-logging-drivers>

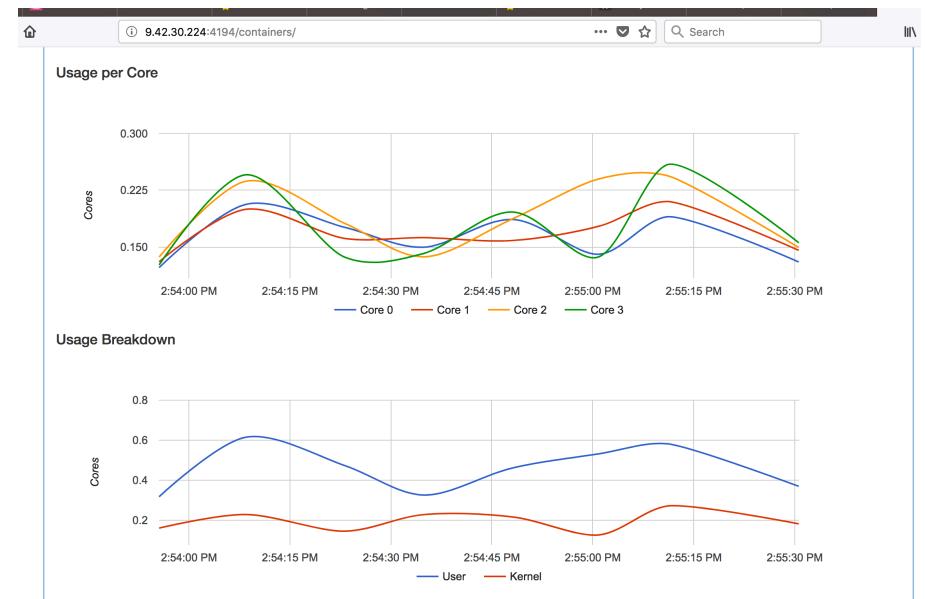
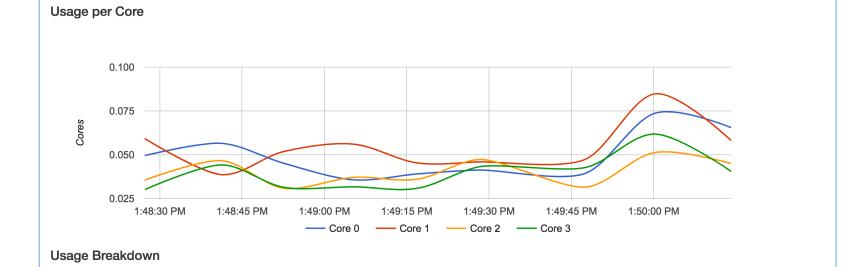
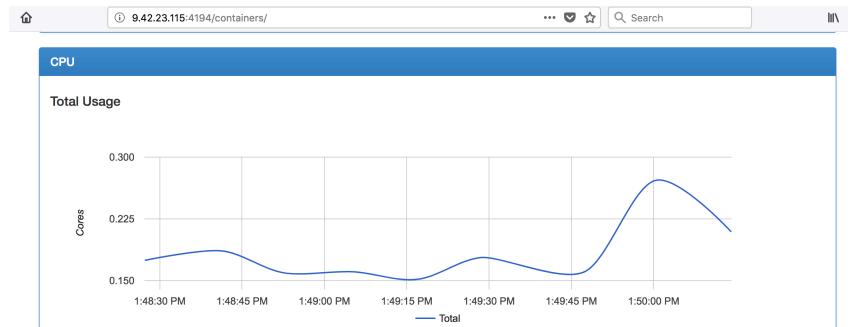
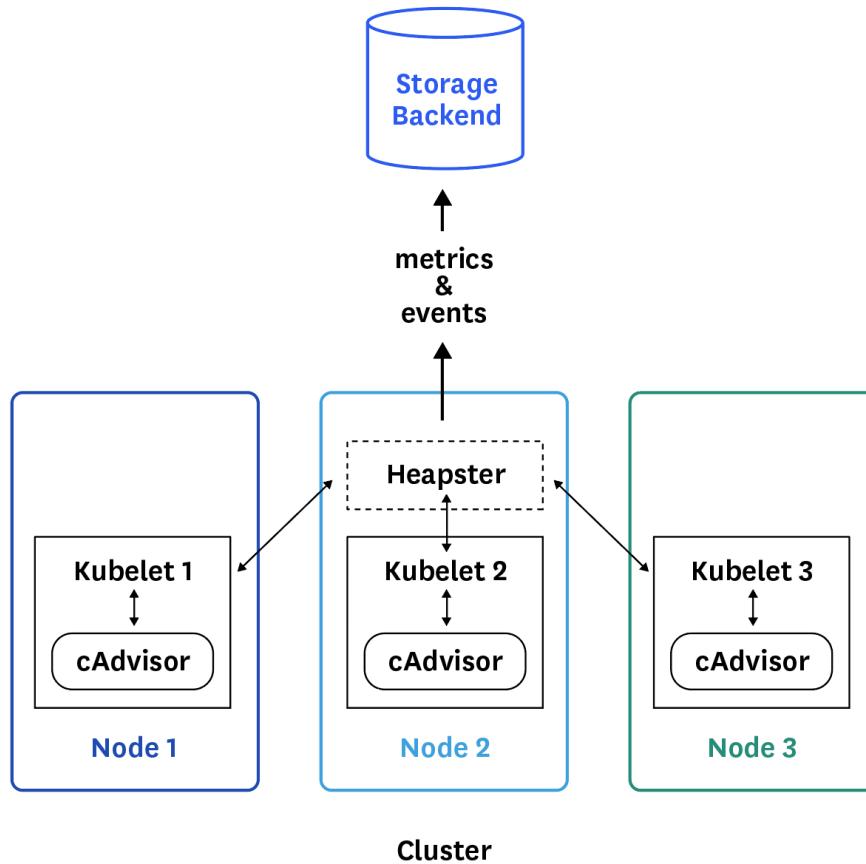
日志分析 ELK



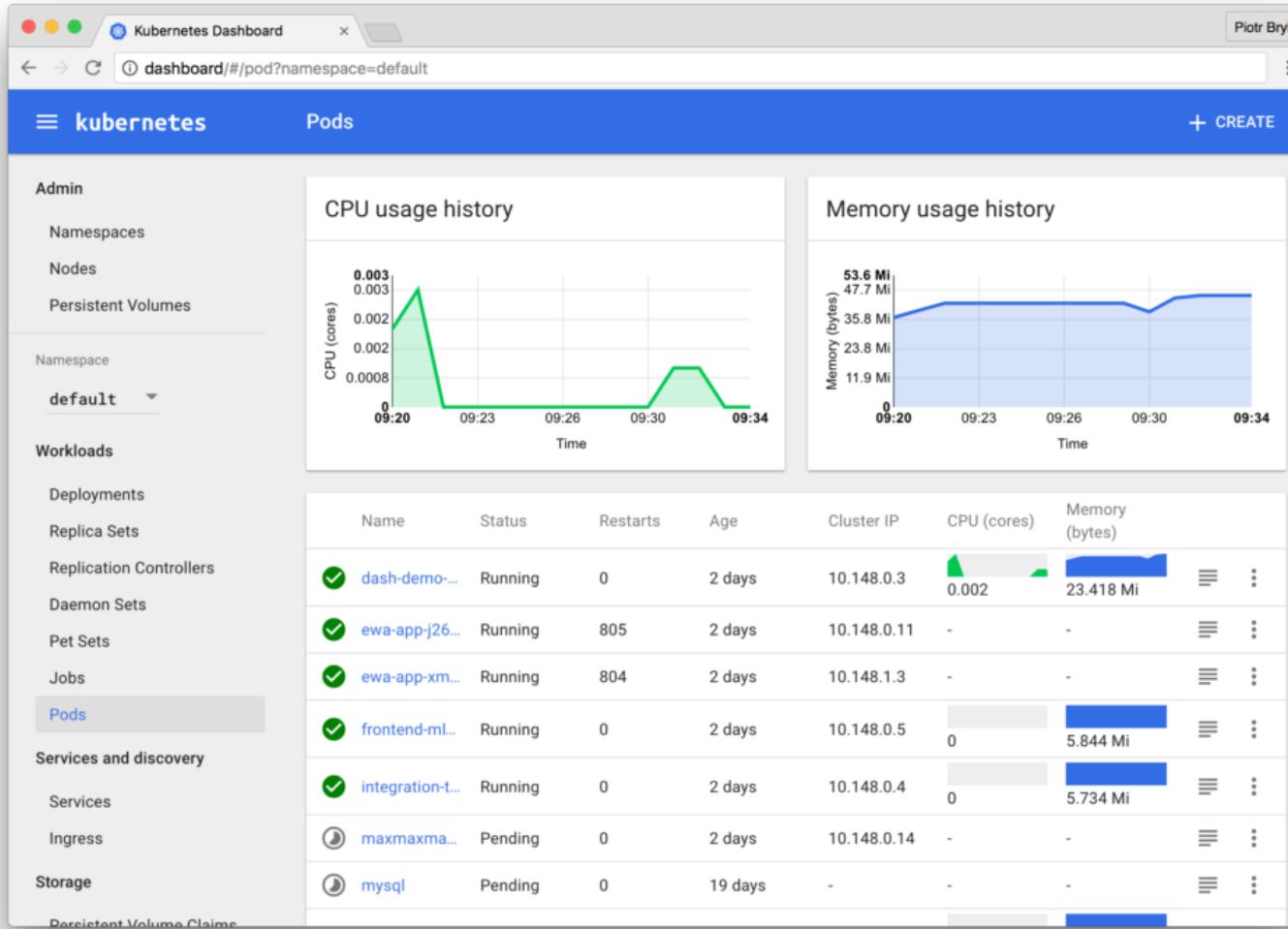
Kubernetes 的监控

- 云监控和传统监控的不同
 - 容器的封闭性，隔离性
 - 传统监控方案无法进入
 - 需要专用的容器级别的监控方案
 - 比如 cAdvisor / Heapster
 - 容器的动态调度
 - 资源占用动态变化
 - 资源不足不一定出发报警/扩容请求
 - 更多的是触发调度请求
 - 网络的虚拟化 / 软件定义网络
 - 传统网络监控方案只能覆盖底层网络
 - 网络虚拟化软件本身成为了主要的网络故障点
- 云上的应用程序的监控
 - APM
 - 应用性能管理
 - 不仅仅是监控，拿到数据以后要做出响应
 - Deep Dive Monitor
 - Instrument 怎样注入容器运行时
 - Side Car 方案
- 常见的方案
 - Heapster + InfluxDB + Grafana
 - Google 有现成的部署示例
 - Heapster + Prometheus + Grafana

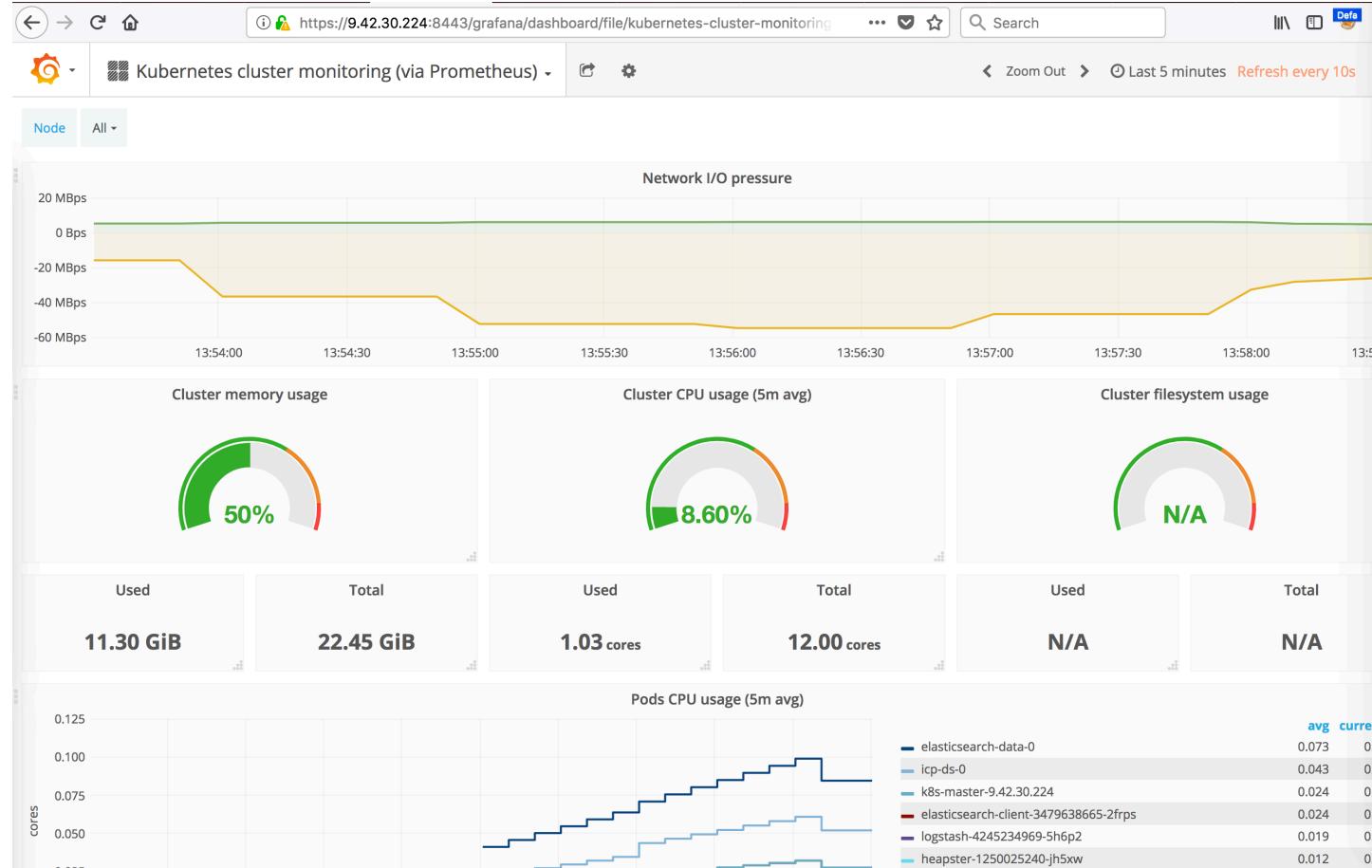
cAdvisor / Heapster



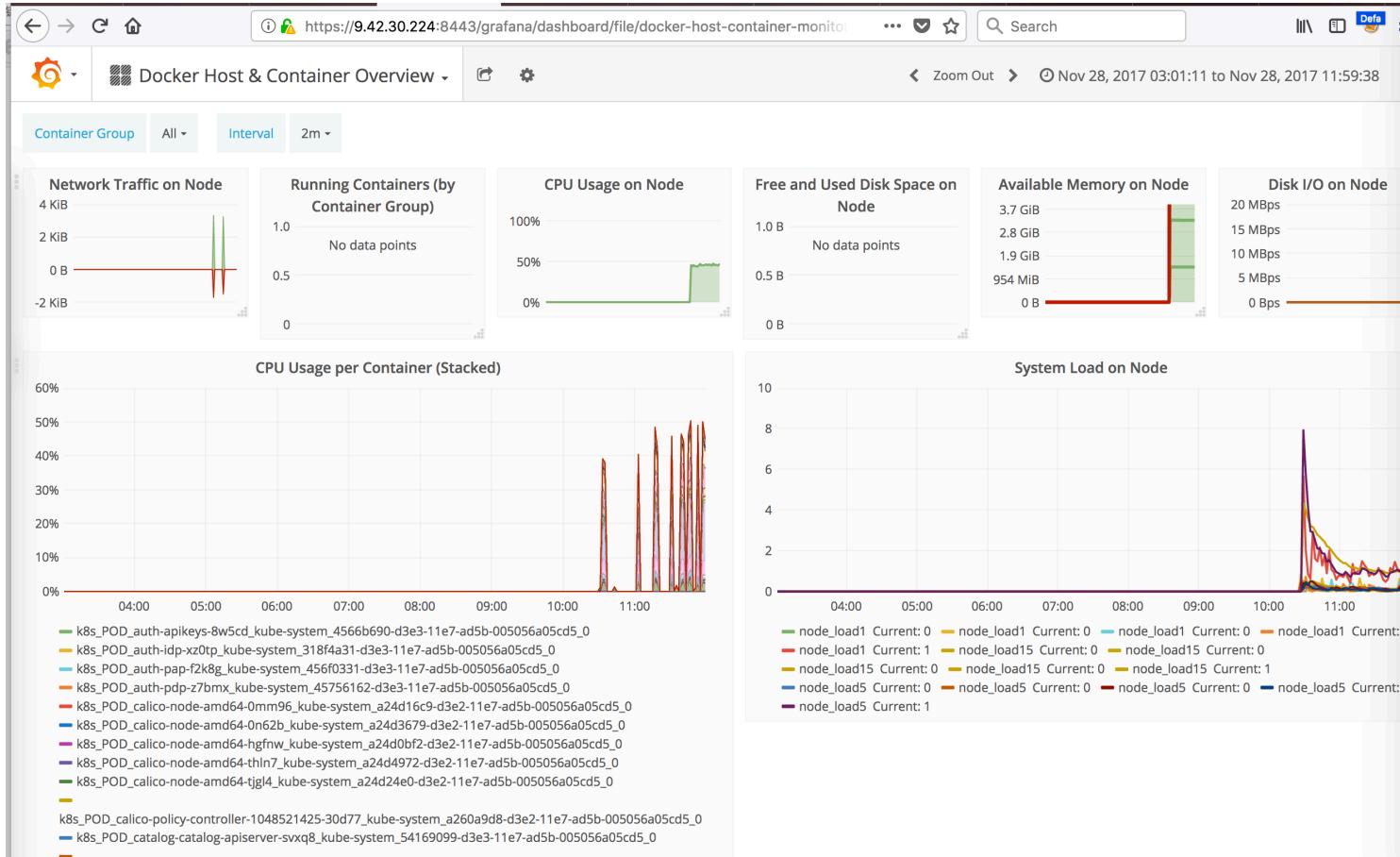
Kubernetes Dashboard



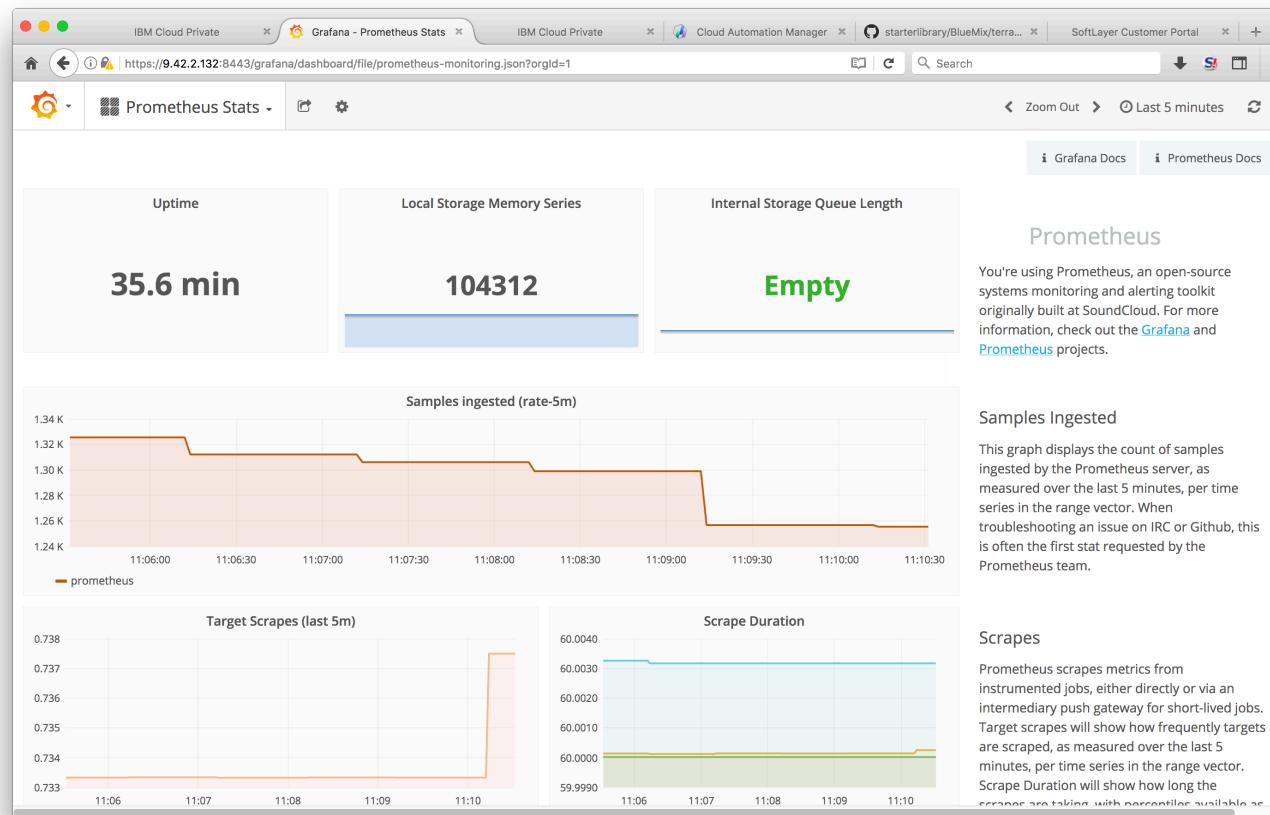
Heapster + Prometheus + Grafana



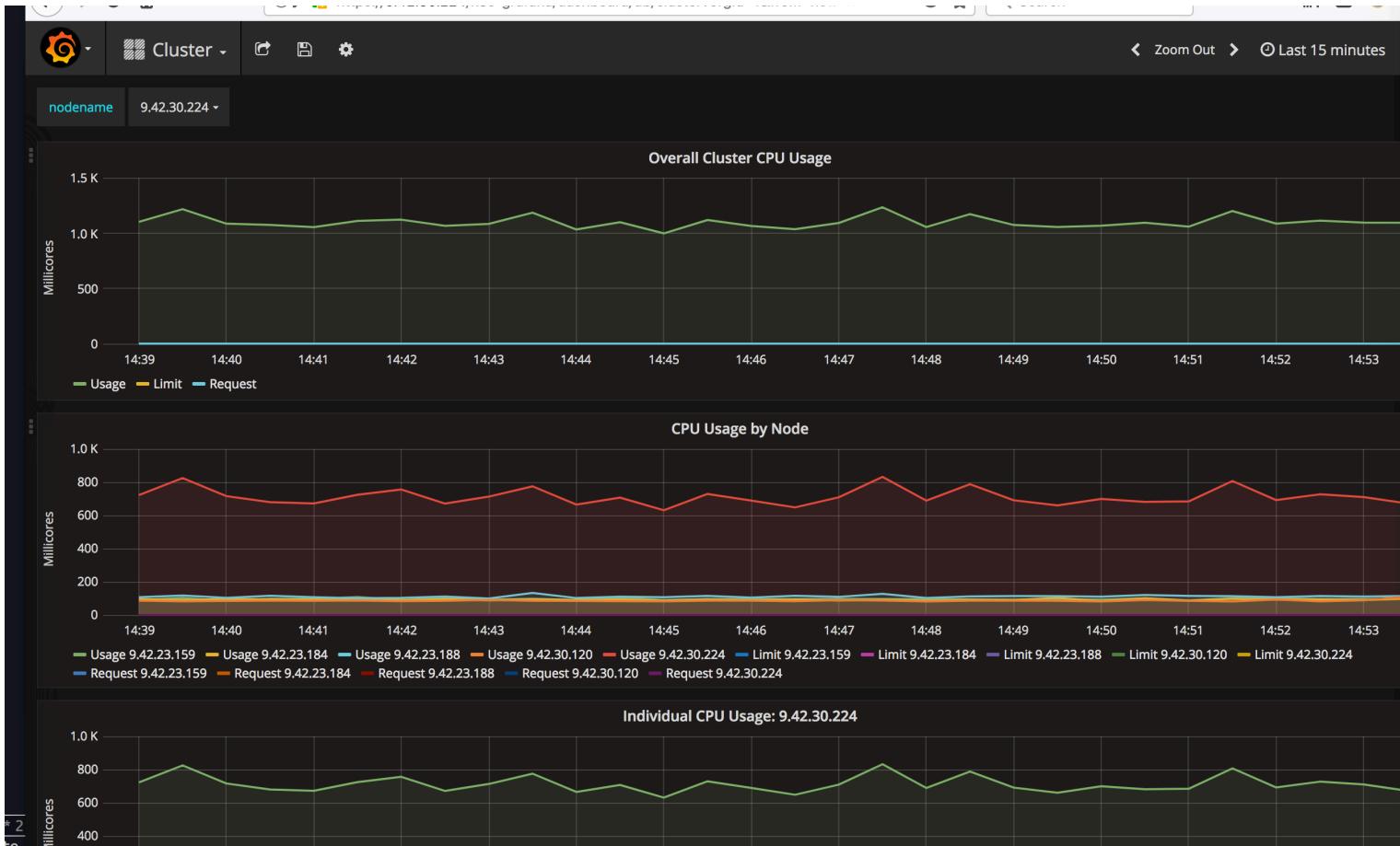
Heapster + Prometheus + Grafana



Prometheus



Heapster + InfluxDB + Grafana



APM for DevOps

