Kubernetes 日志实践

陈全照 | 云澔
阿里云容器服务

阿里云容器服务


DevOps
微服务
DUBBO
企业应用
.NET
ORACLE
智能创新
IoT
容器服务
多集群管理
安全合规
混合云
日志、监控
应用仓库
镜像/解决方案
阿里云
docker
Linux
Windows
专有云
物理机
虚拟机

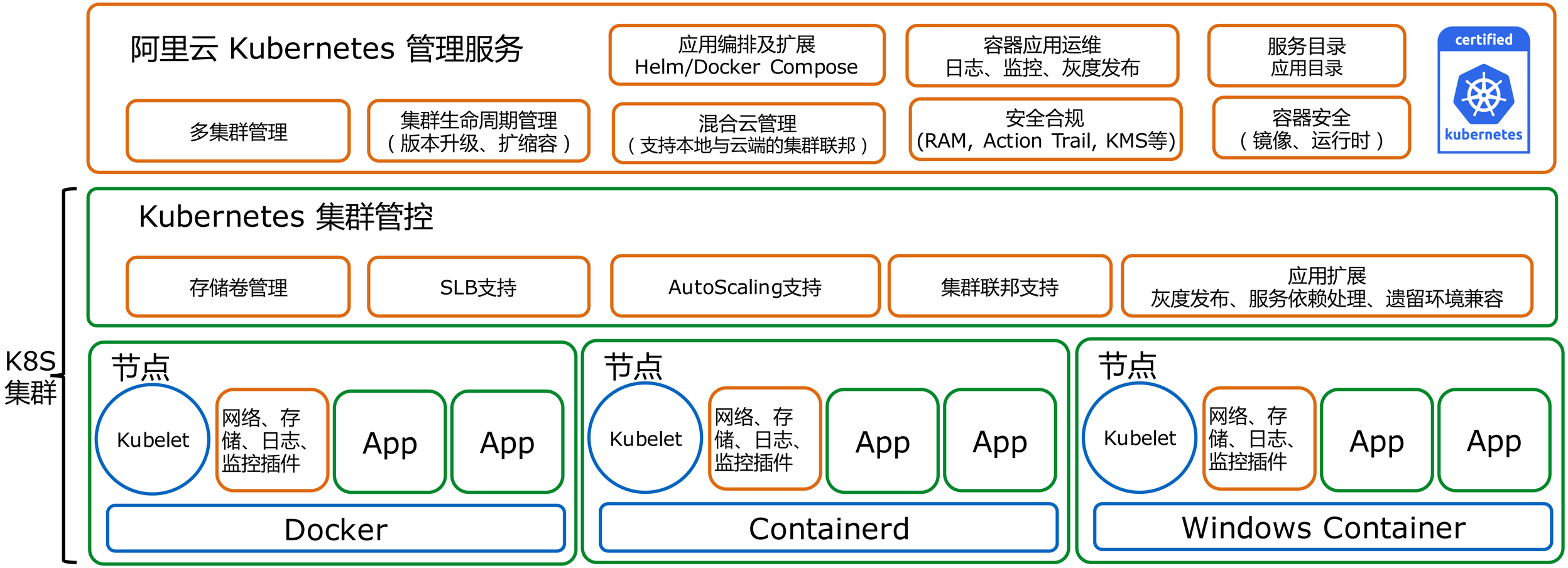
阿里云容器服务Kubernetes

- 完全兼容原生Kubernetes，支持扩展
- 深度整合阿里云公有云资源
- 提供安全、高可用保障和升级服务

全球首批通过k8s一致性认证

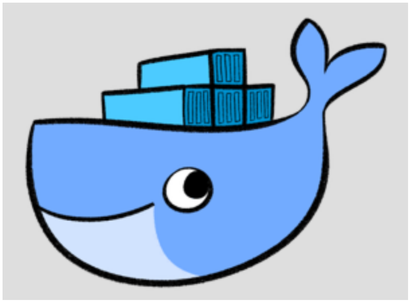
白金会员

阿里云容器服务Kubernetes



容器日志采集难点

- **容器本身特性**
  采集目标多
  弹性伸缩性

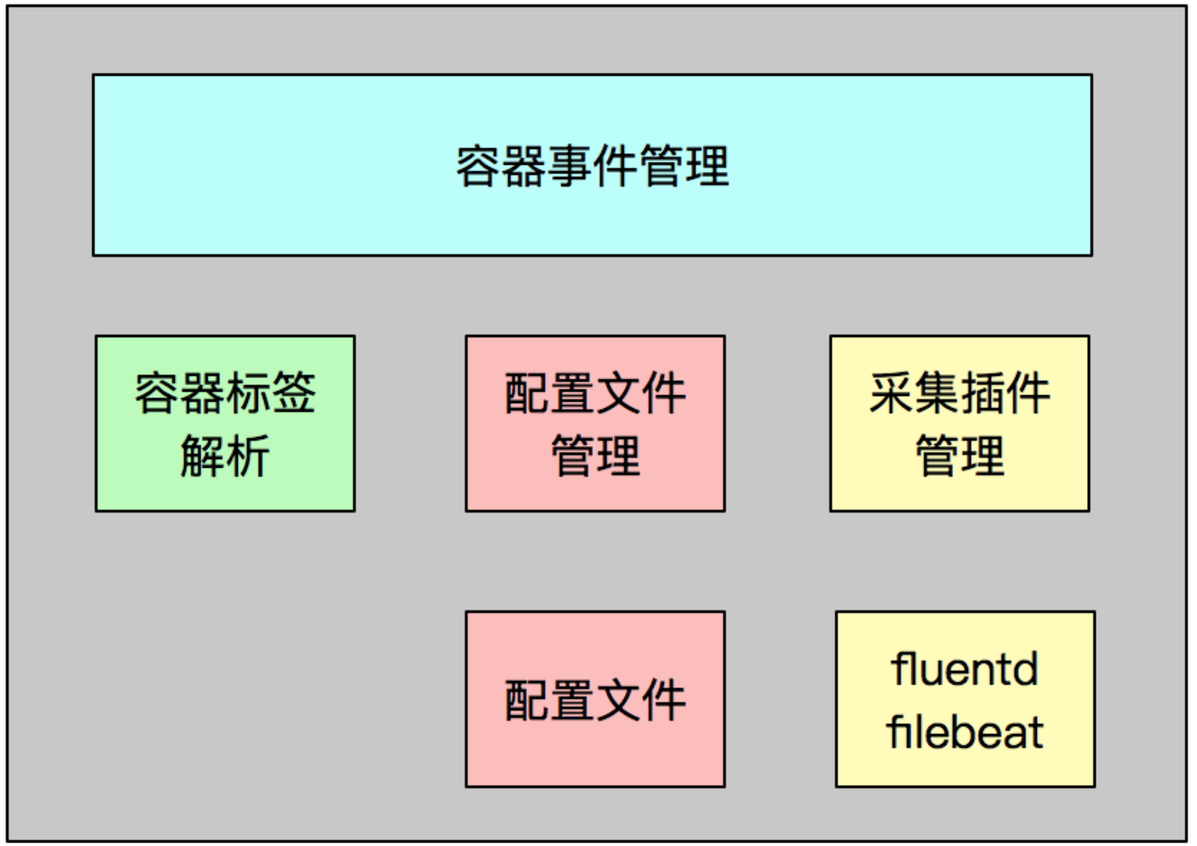


- **现有采集工具缺陷**
  缺乏动态配置能力
  日志采集重复或丢失
  未明确标记日志源

容器日志采集难点破解利器 Log-Pilot

Log-Pilot是一个智能容器日志采集工具，能够高效便捷地将容器日志采集输出到多种后台日志存储系统中，不仅能够采集容器标准输出日志，同时能够动态发现配置采集容器内文件日志

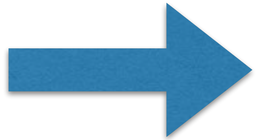
log-pilot



容器日志采集难点破解利器 Log-Pilot

- **容器本身特性**
  采集目标多
  弹性伸缩性

- **现有采集工具缺陷**
  缺乏动态配置能力
  日志采集重复或丢失
  未明确标记日志源



Log-Pilot 破解之道

同时支持stdout和容器内文件日志采集
支持声明式日志配置

自动感知（发现）机制
自动checkpoint及句柄保持机制
自动日志数据打标

Log-Pilot 声明式日志配置

- **支持声明式日志配置**

  容器标签：aliyun.logs.$name=$path

  环境变量：aliyun_logs_$name=$path

  示例：

  aliyun.logs.catalina=stdout

  aliyun.logs.access=/usr/local/tomcat/logs/*.log

  自定义标签前缀：

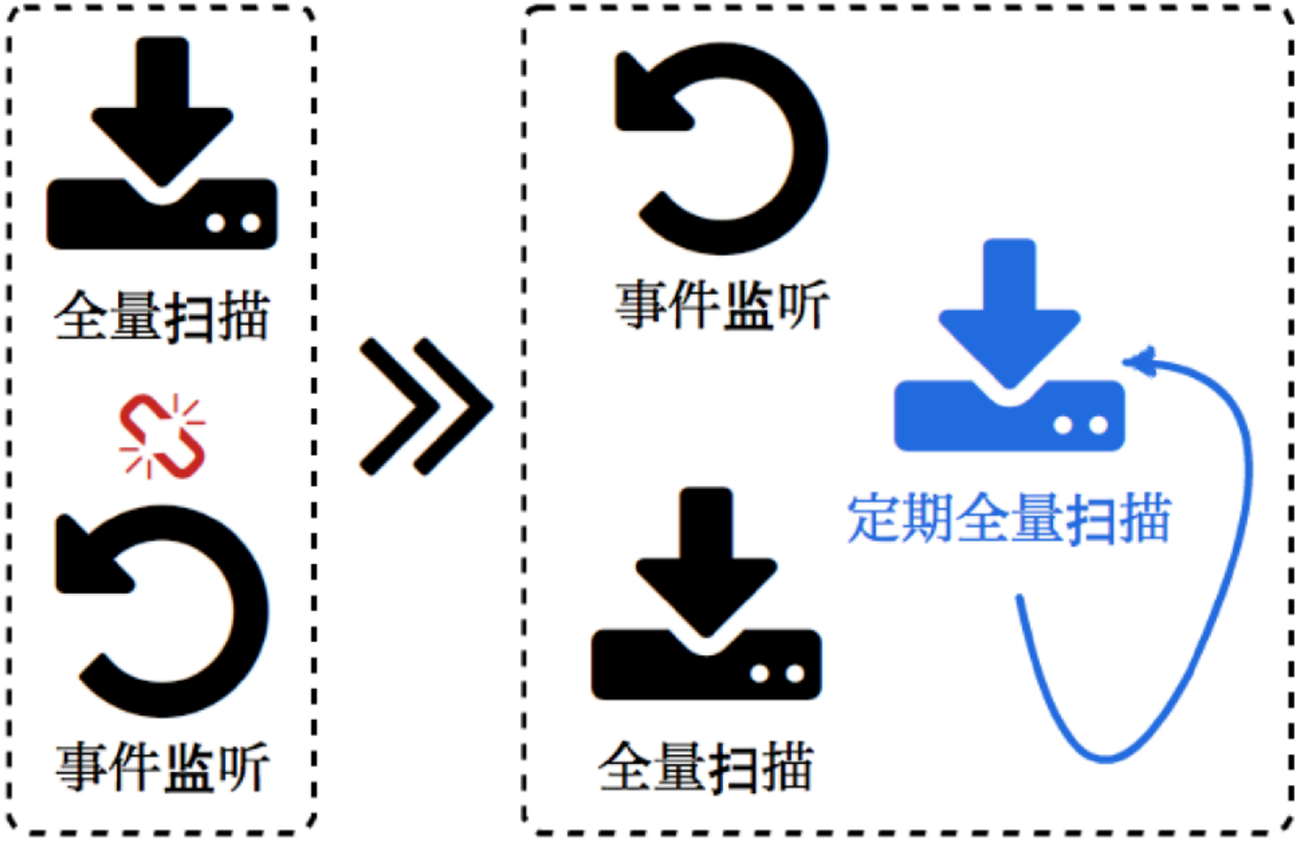
  PILOT_LOG_PREFIX: "aliyun,custom"

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: tomcat
spec:
  containers:
  - name: tomcat
    image: tomcat
    env:
    - name: aliyun_logs_catalina
      value: "stdout"
    - name: aliyun_logs_access
      value: "/usr/local/tomcat/logs/catalina.*.log"
    volumeMounts:
    - name: accesslogs
      mountPath: /usr/local/tomcat/logs
  volumes:
  - name: accesslogs
    emptyDir: {}
```

Log-Pilot 自动发现机制

Log-Pilot能够自动感知宿主机上容器的创建删除进而动态调整容器日志采集配置

一次全量+事件监听
事件监听+定期全量



Log-Pilot Checkpoint及句柄保持特性

CheckPoint机制

1. 实时跟踪日志采集偏移量
2. 定期持久化
3. 维持日志文件信息与偏移量映射关系

句柄保持

1. 监测到新日志文件产生时立即打开文件句柄并维持打开状态
2. 当日志文件删除且日志采集完成才会释放文件句柄

Log-Pilot 自动日志数据打标

docker_app：容器所属应用名称
docker_service：容器所属服务名称
docker_container：容器名称

Swarm

k8s_pod：容器所属pod名称
k8s_pod_namespace：容器所属namespace
k8s_node_name：容器所在node名称
k8s_container_name：自定义k8s容器名称

K8S

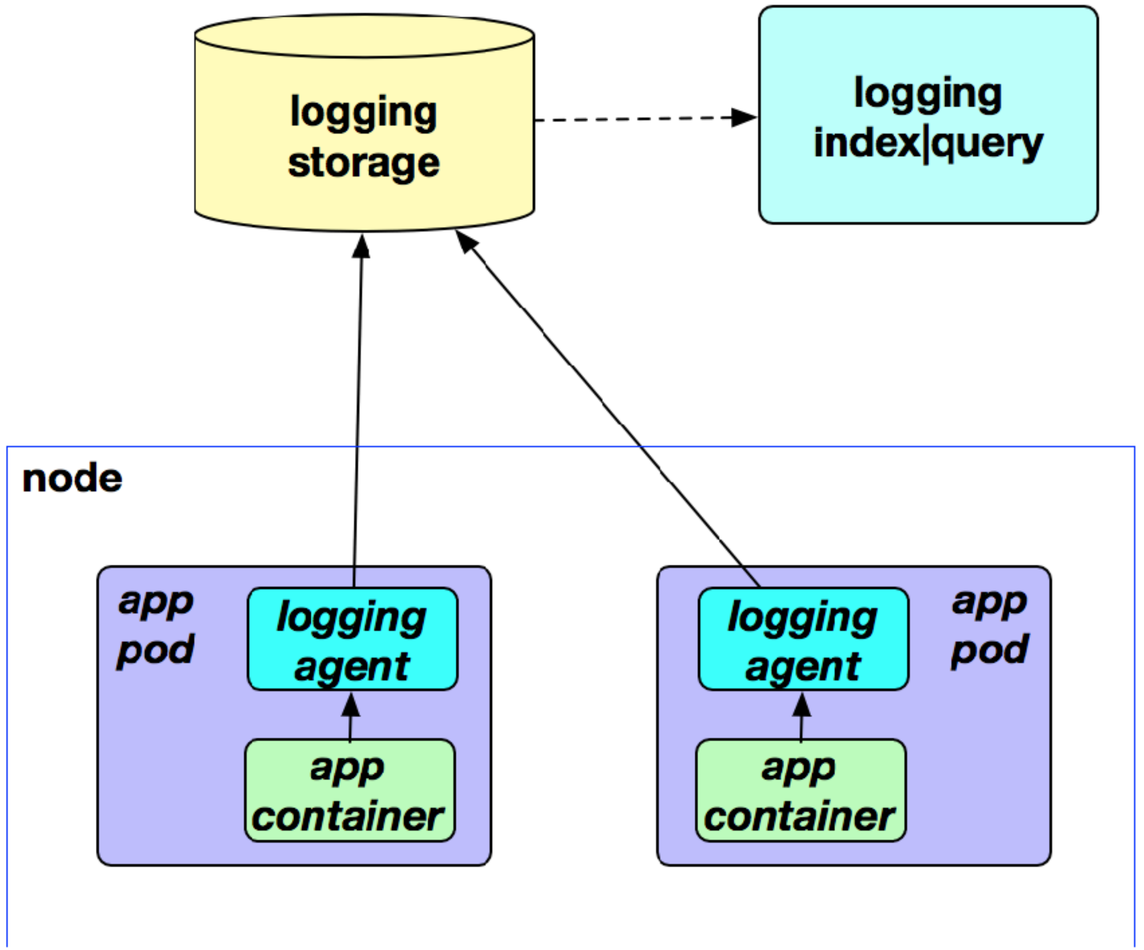
Log-Pilot 高级特性

- 低资源消耗
- 支持自定义tag
- 支持多种解析格式
- 支持自定义日志输出target
- 支持fluentd和filebeat插件
- 支持多种日志存储后端

容器日志采集模式一

- **SideCar模式**

  每个Pod中都附带一个专用logging容器用于本Pod内容器的日志收集



劣势：

1. 资源占用多，无论是CPU还是MEM
2. 占用后端过多连接数，集群规模越大引起潜在问题越大

容器日志采集模式二

- **Node模式**

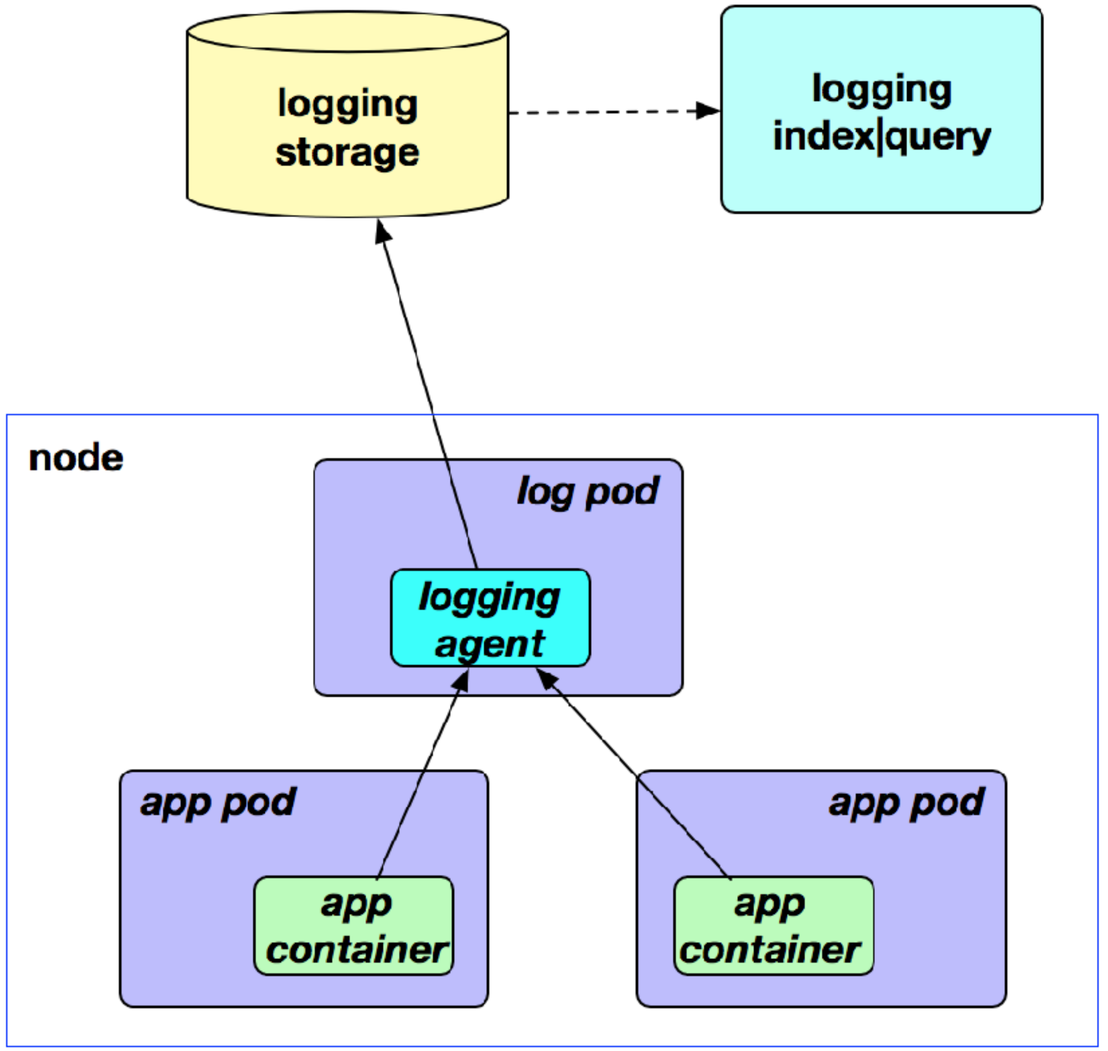
  每个Node上仅会部署一个logging容器用于本Node所有容器的日志收集

  优势：

  1. 资源占用少，集群规模越大优势越明显
  2. 社区推荐模式

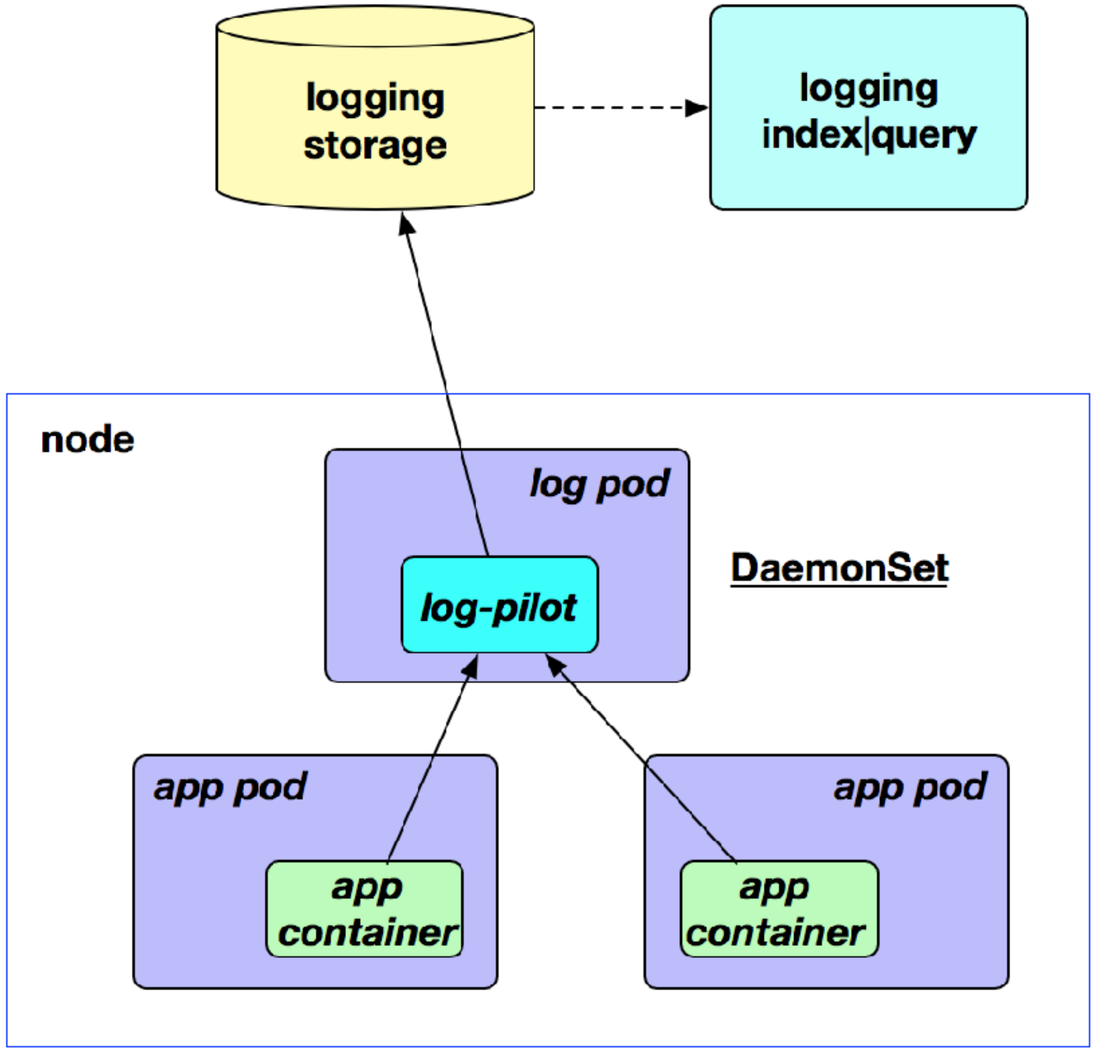
  缺陷：

  需要更智能的logging agent配合



Log-Pilot 高级特性 - 低资源消耗

1. 每台节点上仅部署一个log-pilot容器
2. 每个容器内部仅起一个主进程进行
   日志监控采集

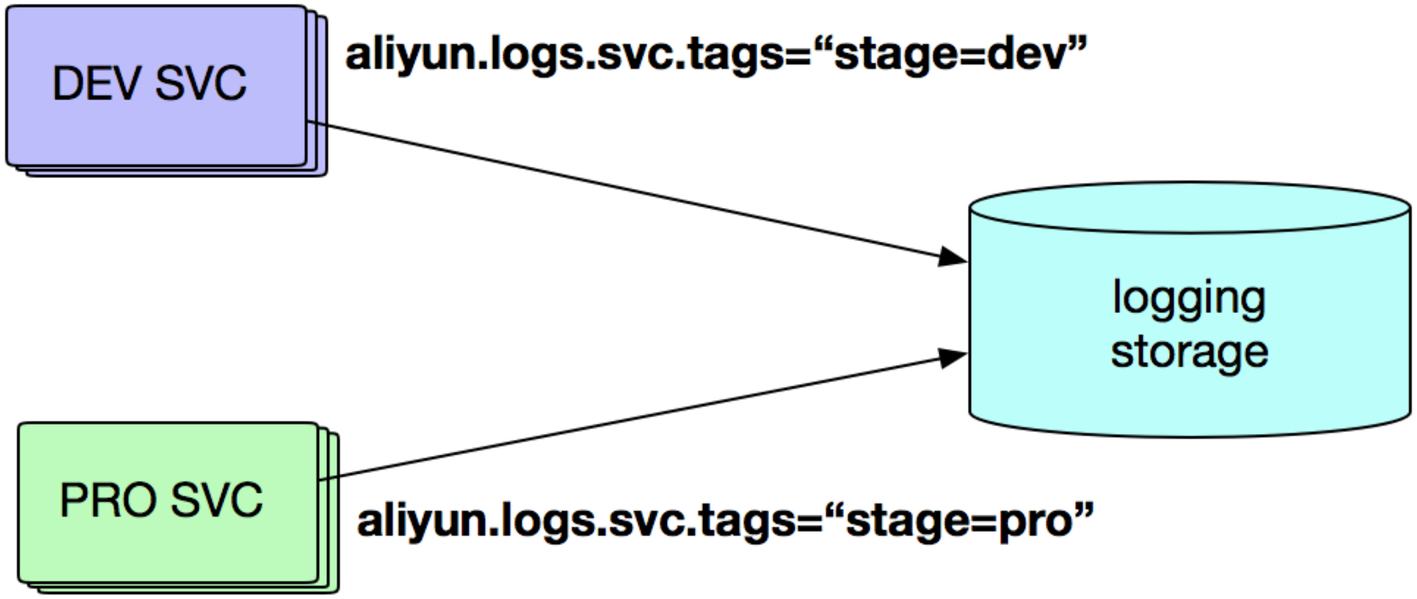


Log-Pilot 高级特性 - 自定义tag

容器标签：aliyun.logs.$name.tags=“k1=v1,k2=v2”
环境变量：aliyun_logs_$name_tags=“k1=v1,k2=v2”
k1=v1,k2=v2将会自动被采集到容器日志输出中

用途：

1. 日志标记
2. 日志路由
3. 日志过滤
4. 日志统计



Log-Pilot 高级特性 - 多种解析格式

容器标签：aliyun.logs.$name.format=<format>

环境变量：aliyun_logs_$name_format=<format>

format:

1. none
2. json
3. csv
4. nginx
5. apache2
6. regexp

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: tomcat
spec:
  containers:
  - name: tomcat
    image: tomcat
    env:
    - name: aliyun_logs_catalina
      value: "stdout"
    - name: aliyun_logs_catalina_format
      value: "json"
    - name: aliyun_logs_access
      value: "/usr/local/tomcat/logs/catalina.*.log"
    volumeMounts:
    - name: accesslogs
      mountPath: /usr/local/tomcat/logs
  volumes:
  - name: accesslogs
    emptyDir: {}
```

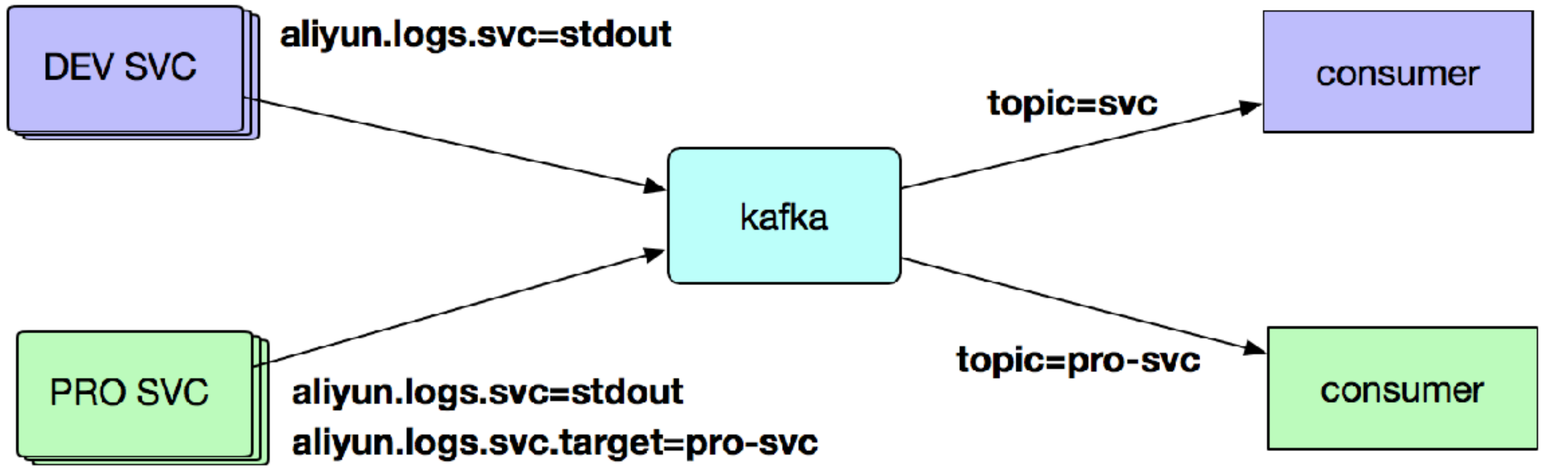
Log-Pilot 高级特性 - 自定义输出target

容器标签：aliyun.logs.$name.target=<target>

环境变量：aliyun_logs_$name_target=<target>

<target>：自定义字符串，分别指代：

1. elasticsearch -> index
2. kafka - > topic
3. aliyun sls -> logstore name



Log-Pilot 高级特性 - 支持多插件多后端

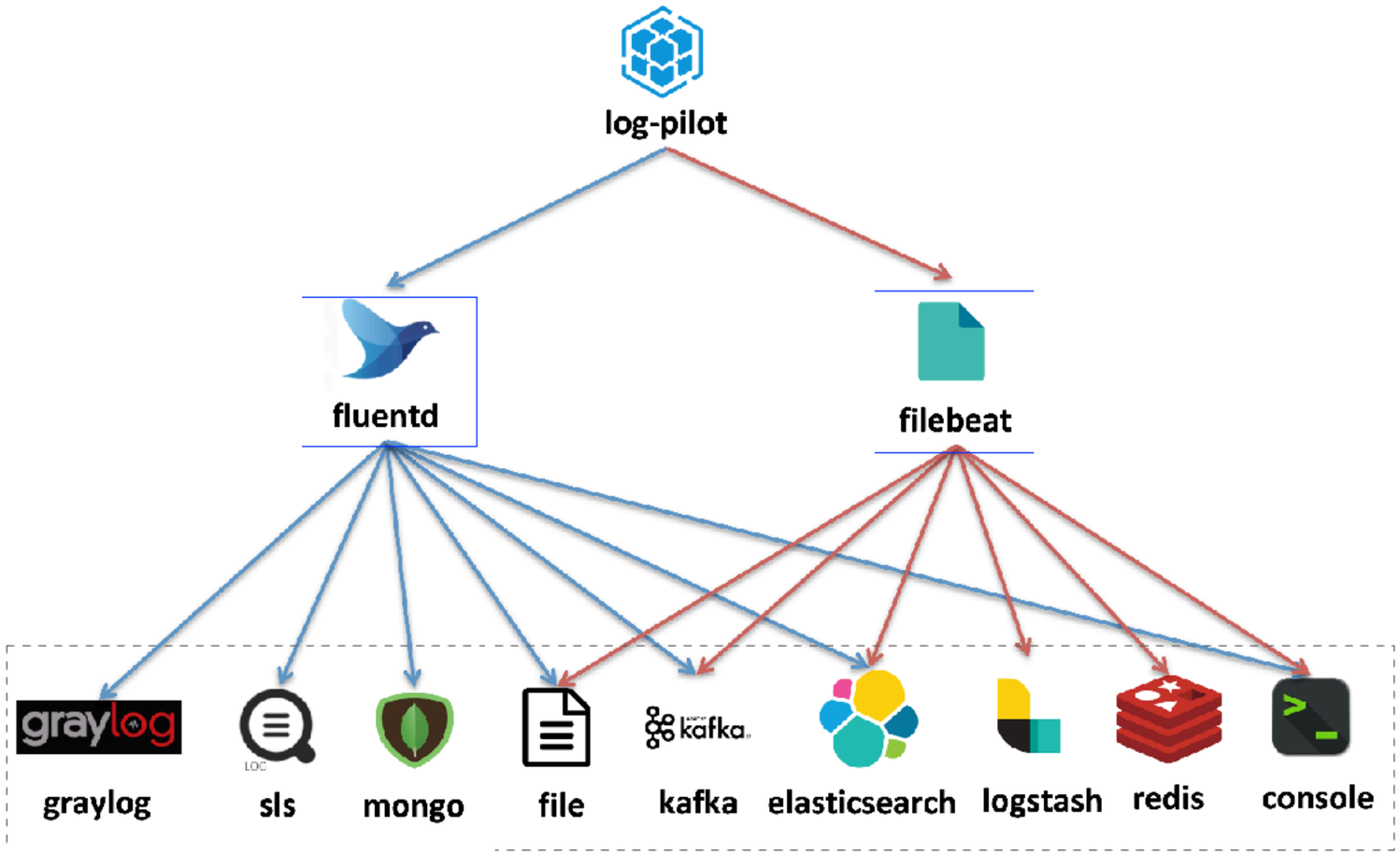
- **多插件支持**

  PILOT_TYPE: fluentd | filebeat

  fluentd： 对接后端多

  filebeat： 高性能

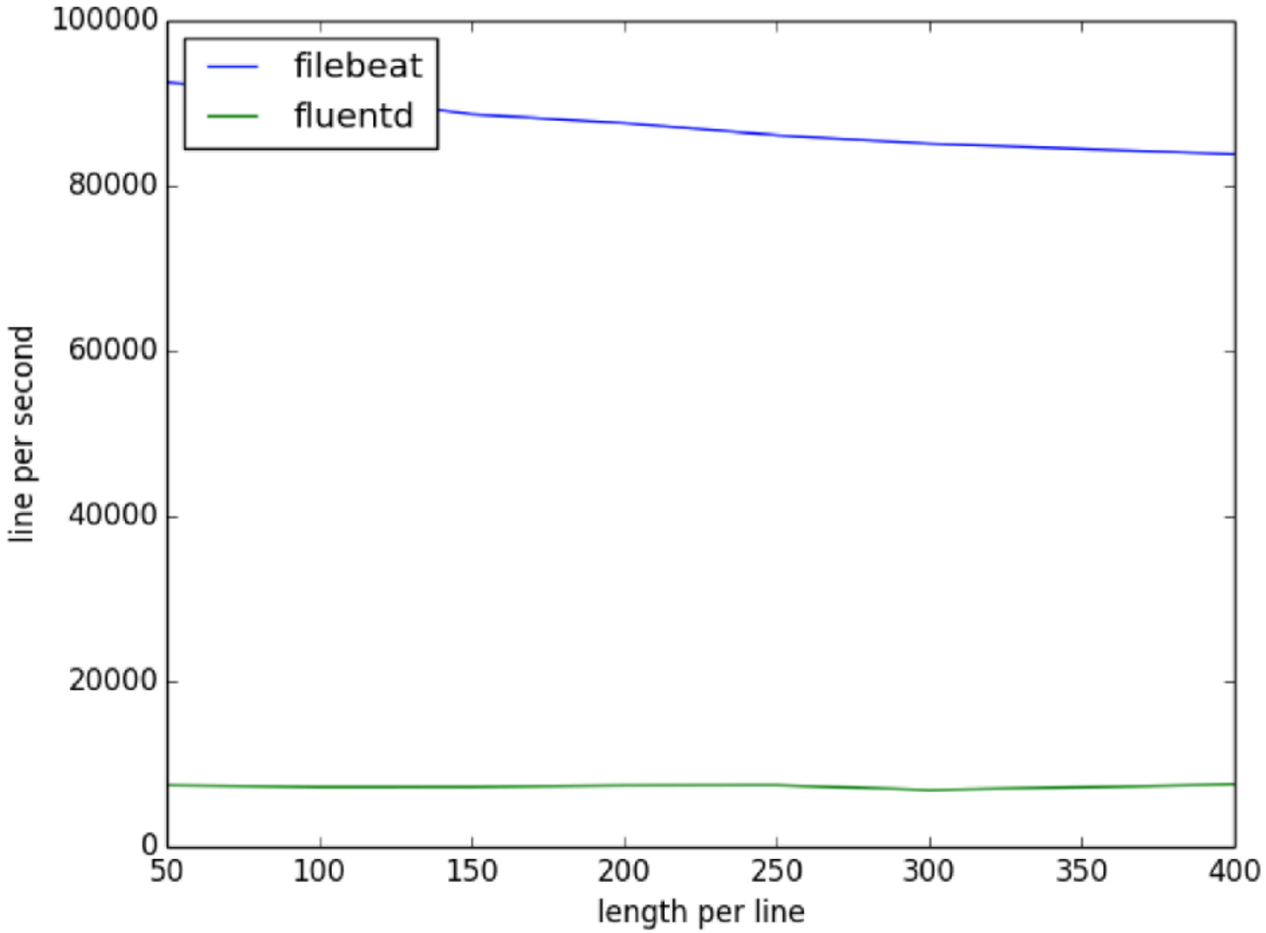
- **多后端支持**

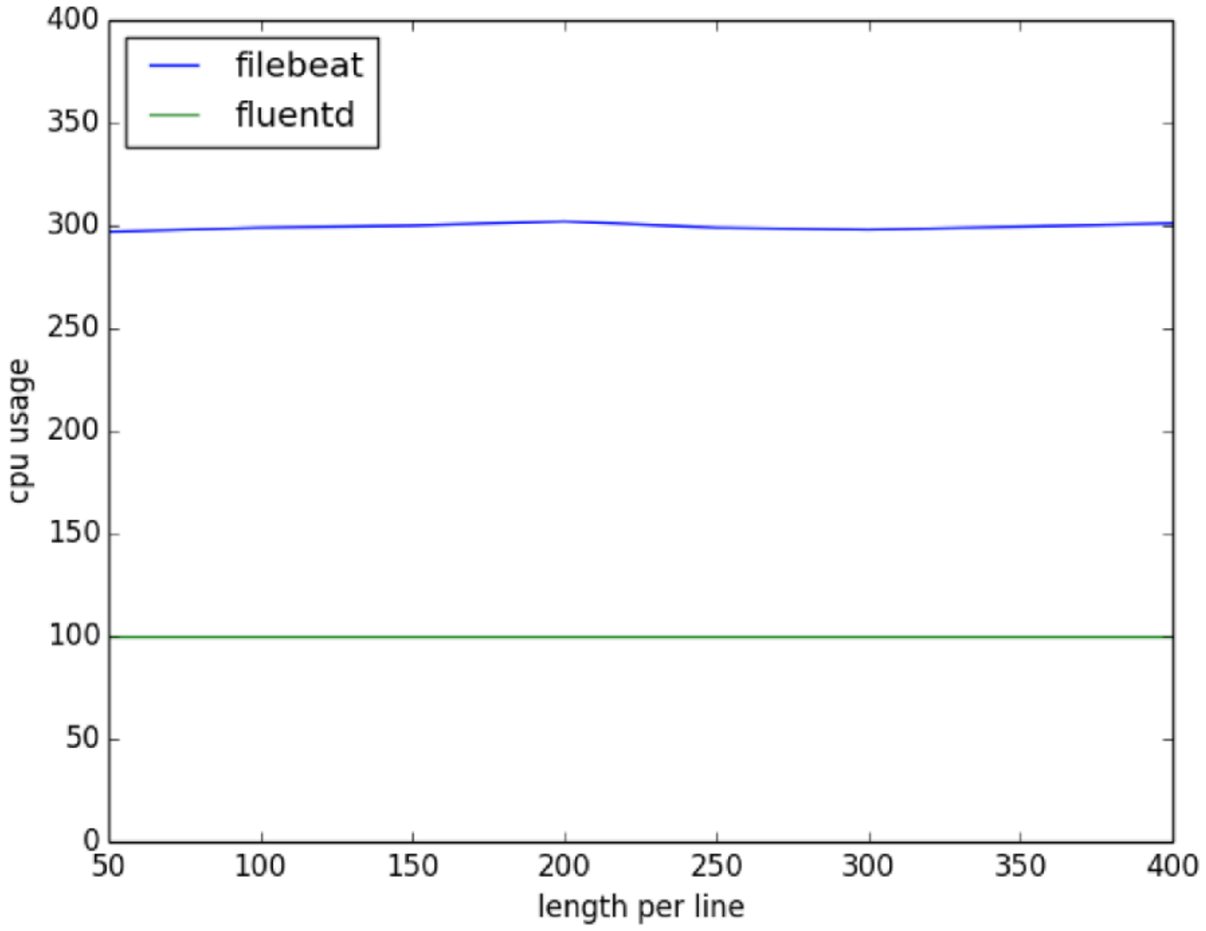


Log-Pilot 性能压测一

场景一： 积压大量日志采集速率

fluentd: 7500line/s
filebeat: 85000line/s -> 100000line/s

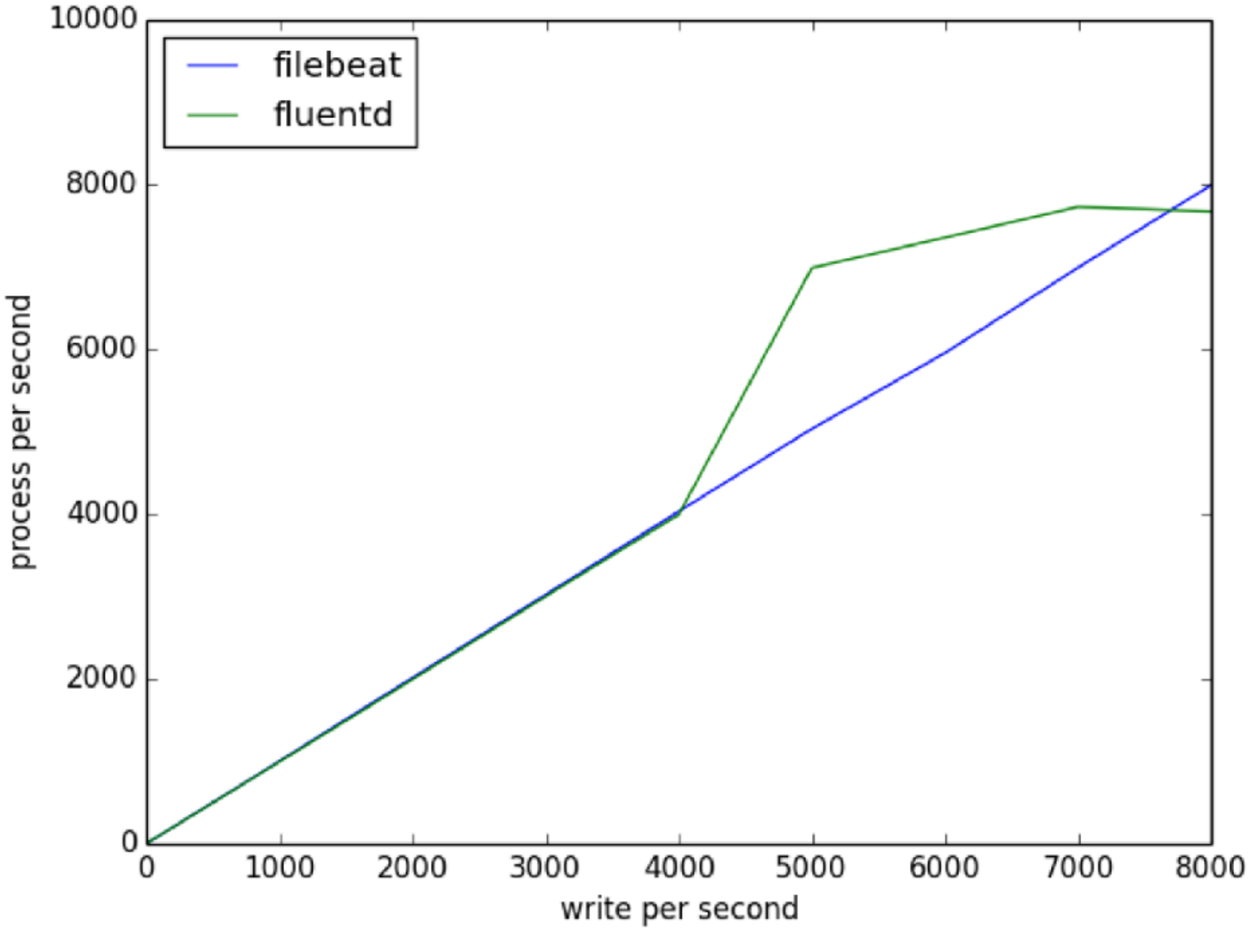


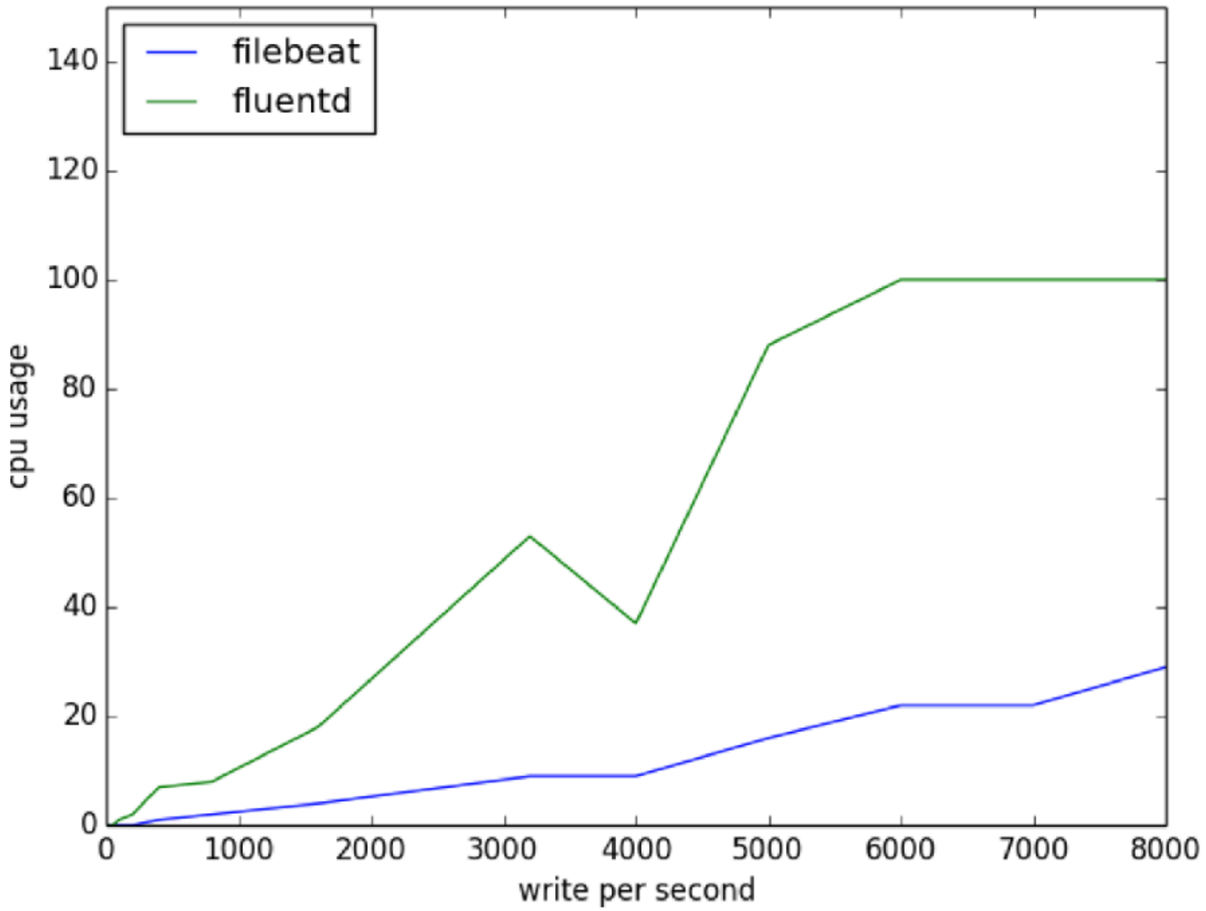


Log-Pilot 性能压测二

场景二： 边产生日志边采集速率

fluentd: 7500line/s
filebeat: 85000line/s -> 100000line/s





阿里云容器日志采集 Log-Pilot

- 开源开放

  https://github.com/AliyunContainerService/log-pilot

- 增强扩展

谢谢！

Q & A

容器服务团队博客

MORE THAN JUST CLOUD | Alibaba Cloud