



链滴

kubernetes 日志管理分析

作者: [Leif160519](#)

原文链接: <https://ld246.com/article/1591763841647>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



我们了解到k8s主要分为两种部署方式，一种是kubeadm，另外一种是二进制，而kubeadm方式部署的k8s集群，除了kubelet之外是使用systemd之外，其他组件都是以容器化的方式部署的，而二进制部署的集群，都是用systemd方式管理组件服务的，因此这两种部署方法查看日志的方式也不一样。

一、使用systemd管理(适用于kubeadm)

指定kubelet日志：

```
journalctl -u kubelet
```

提示：日志过多建议重定向到文件中方便查看。

而journal这个命令实际查看的日志来源于：`/var/log/messages`

补充：主配置文件地址：`/var/lib/kubelet/config.yaml`

注意：二进制部署方式一般会指定日志的路径，故上述方法不适用二进制部署方式

二、使用容器管理(输出控制台)

```
kubectl logs <pod名称> -n <命名空间>
```

可搭配`--tail` 和 `-f` 使用

该日志被docker容器接管，实际目录为：`/var/lib/docker/containers/<container-id>/<container-id>-json.log`

查看方式：查询该pod所在的节点-》查询对应的docker容器id-》进入对应目录查看

三、查看容器标准输出日志(日志在本地文件中)

查看日志方式：进入pod中-》查看对应目录下的日志

改进办法：将日志文件持续化到宿主机上

扩展：

docker中涉及的两个卷volume和bind mount，其中volume是卷目录由docker管理，bind mount挂载宿主机上文件或者目录到容器中，而k8s与docker相对应的，也有两种卷，分别为emptydir和hostath,emptydir由k8s管理，文件存放在k8s自己设定的目录中，hostpath可以由用户控制，可以放到主机上任意目录中。

1.hostPath

示例：

```
apiVersion: v1
kind: Pod
metadata:
  name: my-pod
spec:
  containers:
  - name: web
    image: lizhenliang/nginx-php
    volumeMounts:
    - name: logs
      mountPath: /usr/local/nginx/logs
  volumes:
  - name: logs
    hostPath:
      path: /tmp/logs
      type: Directory
```

apply之前先在每个节点上创建、tmp/logs目录，否则pod会创建失败

参数解释：

- volumes：设置数据卷来源
- type：挂载的类型，文件还是目录
- path：挂载的文件或者目录的路径
- volumeMounts.name与volumes.name：卷名称需要对应
- mountPath：容器中目录挂载的路径

pod创建成功后，会在对应节点的/tmp/logs下发现有两个nginx的日志文件

```
[root@k8s-master k8s]# kubectl get pod -o wide
NAME      READY   STATUS    RESTARTS   AGE   IP           NODE      NOMINATED NODE   READINESS GATES
my-pod    1/1     Running   0           2m48s  10.244.36.92 k8s-node1 <none>         <none>

[root@k8s-node1 ~]# ls /tmp/logs/
access.log  error.log
```

curl 10.244.36.92 后会在access.log下实时刷出日志：

```
[root@k8s-master k8s]# curl 10.244.36.92
<html>
<head><title>403 Forbidden</title></head>
<body>
<center><h1>403 Forbidden</h1></center>
<hr><center>nginx</center>
</body>
</html>
```

```
[root@k8s-node1 logs]# tail -f access.log
10.244.235.192 - - [10/Jun/2020:12:26:12 +0800] "GET / HTTP/1.1" 403 146 "-" "curl/7.29.0"
```

将日志文件挂载到宿主机目录，这样一来，就可以进容器查看或者在宿主机查看日志了。

2.emptyDir

示例:

```
apiVersion: v1
kind: Pod
metadata:
  name: my-pod2
spec:
  containers:
  - name: web
    image: lizhenliang/nginx-php
    volumeMounts:
    - name: logs
      mountPath: /usr/local/nginx/logs
  volumes:
  - name: logs
    emptyDir: {}
```

```
[root@k8s-master k8s]# kubectl get pod -o wide
NAME      READY   STATUS             RESTARTS   AGE   IP           NODE           NOMINATED NODE   READINESS GATES
my-pod    1/1     Running            0           9m8s  10.244.36.92 k8s-node1     <none>            <none>
my-pod2   0/1     ContainerCreating  0           22s   <none>       k8s-node2     <none>            <none>
```

emptyDir日志目录查看方法:

- 1.在分配的节点上使用docker ps 命令查找my-pod2分配的pod-id

```
[root@k8s-node2 logs]# docker ps | grep my-pod2
e16183cbeed    lizhenliang/nginx-php    "docker-entrypoint.s..."    2 minutes ago    Up 2 minutes
1998f87abd49   k8s_web_my-pod2_default_d9e46257-e369-483b-a127-fb90f7ce17d8_0    registry.aliyuncs.com/google_containers/pause:3.2    "/pause"    3 minutes ago    Up 3 minutes
k8s_POD_my-pod2_default_d9e46257-e369-483b-a127-fb90f7ce17d8_0
```

- 2.进入目录 `/var/lib/kubelet/pods/<pod-id>/volumes/kubernetes.io~empty-dir/:`

```
[root@k8s-node2 logs]# cd /var/lib/kubelet/pods/
25cc8b05-7775-4457-beec-2866548d51e9/ dif4f026-23c8-4bd1-a024-8d96148fd536/ ed77a0e3-4d5a-4cee-81bf-5b75a0a4337f/
62e79533-4822-45ab-9583-5f10c64785ee/ d9e46257-e369-483b-a127-fb90f7ce17d8/ ef983d52-ba33-42ef-8192-fcef69234dd9/
[root@k8s-node2 logs]# cd /var/lib/kubelet/pods/d9e46257-e369-483b-a127-fb90f7ce17d8/volumes/kubernetes.io~
kubernetes.io~empty-dir/ kubernetes.io~secret/
[root@k8s-node2 logs]# cd /var/lib/kubelet/pods/d9e46257-e369-483b-a127-fb90f7ce17d8/volumes/kubernetes.io~empty-dir/
[root@k8s-node2 kubernetes.io~empty-dir]# ls
logs
[root@k8s-node2 kubernetes.io~empty-dir]# cd logs/
[root@k8s-node2 logs]# ll
总用量 4
-rw-r--r-- 1 root root 0 6月 10 12:31 access.log
-rw-r--r-- 1 root root 636 6月 10 12:31 error.log
```

curl 同样达到上述效果。