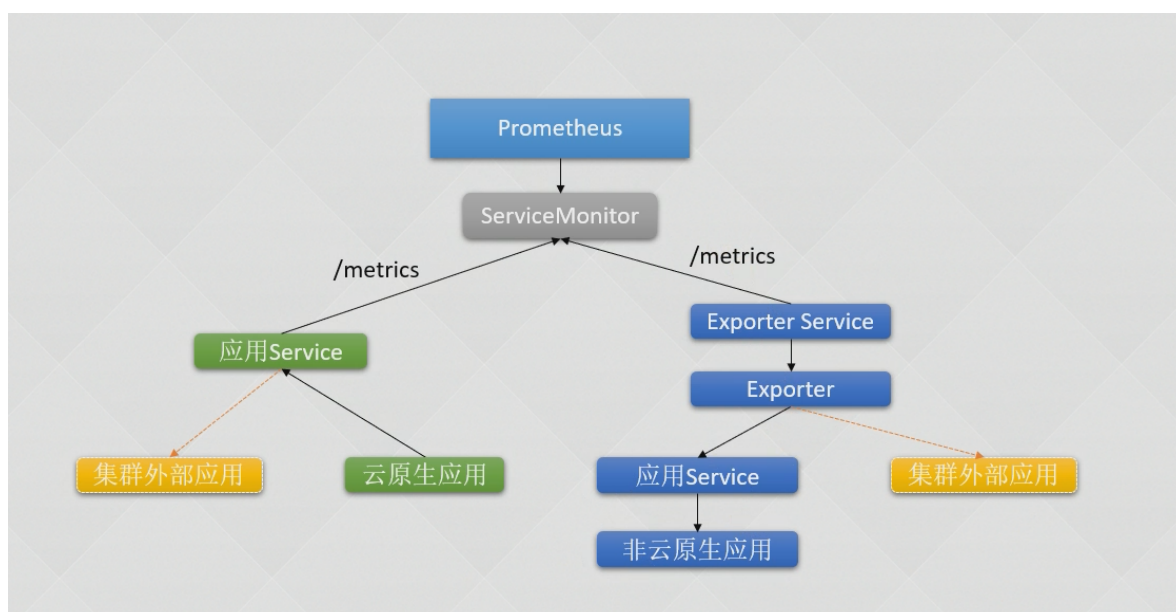


# Prometheus非云原生应用之监控MySQL集群

## 一、首先看下监控流程图

- 从图中看看到，云原生应用是自己提供了 `/metrics` 接口 的，可以直接使用ServiceMonitor的形式去监控服务
- 但是，非云原生应用没 `/metrics` 接口，所以得部署一个Exporter去采集数据，然后Exporter Service代理个 `/metrics` 接口，再使用ServiceMonitor的形式去监控服务【注意需要有应用的svc提供给Exporter去采集数据】



## 二、监控MySQL实战【其他的部署其他的Exporter 采集数据即可】

### 2.1、部署MySQL

```
# 简单部署一个
[root@k8s-master01 ~]# kubectl create deployment mysql --image=registry.cn-beijing.aliyuncs.com/dotbalo/mysql:5.7.23
[root@k8s-master01 ~]# kubectl set env deploy/mysql MYSQL_ROOT_PASSWORD=mysql
```

### 2.2、暴露一个MySQL的svc，提供给Exporter去采集数据

```
[root@k8s-master01 ~]# kubectl expose deployment mysql --port 3306
```

### 2.3、配置Exporter所需的用户、权限

```
[root@k8s-master01 ~]# kubectl exec -it po mysql-69d6f69557-bxkrt -- bash

root@mysql-69d6f69557-bxkrt:/# mysql -uroot -pmysql

mysql> CREATE USER 'exporter'@'%' IDENTIFIED BY 'exporter' WITH
MAX_USER_CONNECTIONS 3;

mysql> GRANT PROCESS,REPLICATION CLIENT, SELECT ON *.* TO 'exporter'@'%';

mysql> quit
```

## 2.4、配置MySQL Exporter 采集Mysql监控数据

- 官网: [https://github.com/prometheus/mysqld\\_exporter](https://github.com/prometheus/mysqld_exporter) 【其他也是去github找就行】
- <https://github.com/orgs/prometheus/repositories?type=all> 这

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mysql-exporter
  namespace: monitoring
spec:
  replicas: 1
  selector:
    matchLabels:
      k8s-app: mysql-exporter
  template:
    metadata:
      labels:
        k8s-app: mysql-exporter
    spec:
      containers:
        - name: mysql-exporter
          image: registry.cn-beijing.aliyuncs.com/dotballo/mysqld-exporter
          env:
            - name: DATA_SOURCE_NAME
              # mysql数据源地址,如果是外部的MySQL域名解析直接写实际IP即可
              # "username:password@(IP:PORT)/"
              value: "exporter:exporter@(mysql.default:3306)/"
          imagePullPolicy: IfNotPresent
          ports:
            - containerPort: 9104
---
apiVersion: v1
kind: Service
metadata:
  name: mysql-exporter
  namespace: monitoring
  labels:
    k8s-app: mysql-exporter
spec:
  type: ClusterIP
  selector:
    k8s-app: mysql-exporter
  ports:
    - name: api
```

```
port: 9104
protocol: TCP
```

# 查看部署结果

```
[root@k8s-master01 ~]# kubectl get po -n monitoring -l k8s-app=mysql-exporter
NAME                                READY   STATUS    RESTARTS   AGE
mysql-exporter-74978c4cc4-87882    1/1     Running   0           28s

[root@k8s-master01 ~]# kubectl get svc -n monitoring -l k8s-app=mysql-exporter
NAME            TYPE        CLUSTER-IP    EXTERNAL-IP  PORT(S)    AGE
mysql-exporter  ClusterIP   10.97.240.30  <none>       9104/TCP   56s

# 测试是否能够获取到数据  svcIP:port/metrics
[root@k8s-master01 ~]# curl 10.97.240.30:9104/metrics
```

## 2.5、创建ServiceMonitor

```
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  name: mysql-exporter
  namespace: monitoring
  labels:
    k8s-app: mysql-exporter
    namespace: monitoring
spec:
  jobLabel: k8s-app # jobLabel指定的标签的值将会作为prometheus配置文件中
  scrape_config下job_name的值, 也就是Target, 如果不写, 默认为service的名称
  endpoints:
    - port: api # mysql-exporter的svc port name, 也可以写端口号, 但是不方便维护,
      # 后期改端口号 此处不需要修改
      interval: 30s # 检测时间间隔
      scheme: http # 协议
  selector:
    matchLabels:
      k8s-app: mysql-exporter
  namespaceSelector:
    matchNames:
      - monitoring

# 查看创建结果
[root@k8s-master01 ~]# kubectl get servicemonitors -n monitoring -l k8s-app=mysql-exporter
NAME            AGE
mysql-exporter  73s
```

## 2.6、判断Prometheus是否成功监控上

- 登录Prometheus页面查看即可

```
[root@k8s-master01 ~]# kubectl get svc -n monitoring prometheus-k8s
```

PrometheusAlertsGraphStatus▼Help

metrics\_path="/metrics/probes"

namespace="k8s-system"node="k8s-node02"

service="kubernetes"

monitoring/mysql-exporter/0 (1/1 up)show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://10.244.122.155:9104/metrics	UP	<div>container="mysql-exporter"endpoint="api"instance="10.244.122.155:9104"job="mysql-exporter"namespace="monitoring"pod="mysql-exporter-74978c4cc4-87882"service="mysql-exporter"</div>	20.439s ago	28.43ms	

monitoring/node-exporter/0 (5/5 up)show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
https://192.168.1.110:9100/metrics	UP	<div>container="kube-rbac-proxy"endpoint="https"instance="k8s-master01"job="node-exporter"namespace="monitoring"pod="node-exporter-6jr4m"service="node-exporter"</div>	8.993s ago	155.6ms	
https://192.168.1.111:9100/metrics	UP	<div>container="kube-rbac-proxy"endpoint="https"instance="k8s-master02"job="node-exporter"namespace="monitoring"pod="node-exporter-nfkn9"service="node-exporter"</div>	1.472s ago	12.08ms	

2.7、grafana页面配置

# 官网找

https://grafana.com/grafana/dashboards/6239-mysql/

目前比较常用的 Exporter 工具如下：↵

类型↵	Exporter↵
数据库↵	MySQL Exporter, Redis Exporter, MongoDB Exporter, MSSQL Exporter↵
硬件↵	Apcupsd Exporter, IoT Edison Exporter, IPMI Exporter, Node Exporter↵
消息队列↵	Beanstalkd Exporter, Kafka Exporter, NSQ Exporter, RabbitMQ Exporter↵
存储↵	Ceph Exporter, Gluster Exporter, HDFS Exporter, ScaleIO Exporter↵
HTTP 服务↵	Apache Exporter, HAProxy Exporter, Nginx Exporter↵
API 服务↵	AWS ECS Exporter, Docker Cloud Exporter, Docker Hub Exporter, GitHub Exporter↵
日志↵	Fluentd Exporter, Grok Exporter↵
监控系统↵	Collectd Exporter, Graphite Exporter, InfluxDB Exporter, Nagios Exporter, SNMP Exporter↵
其它↵	Blackbox Exporter, JIRA Exporter, Jenkins Exporter, Confluence Exporter↵