K8S 中使用边车容器定时备份数据库并发送邮件

作者：chenteng

原文链接：https://ld246.com/article/1640949648526

来源网站：链滴

K8S中使用边车容器定时备份数据库并发送邮件

@[toc]

一、背景：

一开始的初衷是，想写一个脚本来监控服务器的高占用率进程并通过邮件发送给我，然后突发奇想，以使用这种方式来备份我的数据库，开始动手！

二、设计思路：

通过编写shell脚本，调用linux的mail工具，mysqldump的方式来保存数据库的sql文件，通过mail具添加到附件，最后发送到我的邮箱。

三、编写启动脚本

首先我们来编写一个启动脚本

为了方便以后的个性化配置，我们将脚本中的变量都提取到一个application.yml文件中，文件如下：

```
RUNTIME: 084900 ##启动时间，因为容器时区问题，需要当前时间减去8小时
HOST: 172.17.0.3 ##数据库IP地址
USER: root ##数据库用户
PASSWORD: 123456 ##数据库密码
DATABASE: solo ##数据库名
TARGETMAIL: 1849539179@qq.com ##发送的邮箱地址
```

接下来我们来写一下shell脚本，逻辑也很简单，当前时间与启动时间相同时，则调用sendmail函数送邮件

```
#!/bin/bash
#author: chenteng

RUNTIME=$(cat ./application.yml | grep RUNTIME| awk '{print $2}')
HOST=$(cat ./application.yml | grep HOST| awk '{print $2}')
USER=$(cat ./application.yml | grep USER| awk '{print $2}')
PASSWORD=(cat ./application.yml | grep PASSWORD| awk '{print $2}')
DATABASE=$(cat ./application.yml | grep DATABASE| awk '{print $2}')
TARGETMAIL=$(cat ./application.yml | grep TARGETMAIL| awk '{print $2}')

function  sendmail(){
  mysqldump -h$HOST  -u$USER -p$PASSWORD  --complete-insert --skip-add-drop-table  --
ex-blob $DATABASE  > $DATABASE.sql
  echo -e "mysqlbak_$CURRENT_TIME" |mail -s "mysqlbak_$CURRENT_TIME" -a $DATABASE.s
l $TARGETMAIL
  sleep 1
}
while true
do
  CURRENT_TIME=$(date +%H%M%S)
  if [ $CURRENT_TIME = $RUNTIME ];then
    echo "starting bak mysql database"
    sendmail
    continue
  else
    echo $CURRENT_TIME
    sleep 1
  fi
done
```

四、构建镜像

因为我们最后要放到k8s平台上的，所以我们要构建一个镜像，Dockerfile如下：

```
FROM centos
RUN mkdir /app && yum install -y mysql.x86_64 sendmail  mailx libreport-plugin-mailx
WORKDIR /app
COPY demo.sh .
COPY application.yml .
CMD ["/bin/sh","demo.sh"]
```

使用build构建镜像,要记得加一下最后的点

```
docker build -t mysqlmail-bak:1.0.1 .
```

五、添加边车容器

边车容器(sidecar)：边车容器就是与主容器一起在一个pod中运行的容器，为业务容器赋能，共享一网络空间，所以可以用127.0.0.1:3306连接主容器的数据库。

5.1 创建配置文件

为了方便调试，我把里面的shell脚本也挂载出来。

创建两个configmap，分别对应容器内的配置文件与shell脚本

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: mysqlmail-conf
  namespace: solo
data:
  application.yml: |
    RUNTIME: 105800
    HOST: 127.0.0.1
    USER: root
    PASSWORD: 123456
    DATABASE: solo
    TARGETMAIL: 1849539179@qq.com
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: mysqlmail-shell
  namespace: solo
data:
  demo.sh: |
    #!/bin/bash
    #author: chenteng
    RUNTIME=$(cat ./application.yml | grep RUNTIME| awk '{print $2}')
    HOST=$(cat ./application.yml | grep HOST| awk '{print $2}')
    USER=$(cat ./application.yml | grep USER| awk '{print $2}')
    PASSWORD=$(cat ./application.yml | grep PASSWORD| awk '{print $2}')
    DATABASE=$(cat ./application.yml | grep DATABASE| awk '{print $2}')
    TARGETMAIL=$(cat ./application.yml | grep TARGETMAIL| awk '{print $2}')
    function  sendmail(){
      mysqldump -h$HOST  -u$USER -p$PASSWORD  --complete-insert --skip-add-drop-table
--column-statistics=0 --hex-blob $DATABASE  > $DATABASE.sql
      echo -e "mysqlbak_$CURRENT_TIME" |mail -s "mysqlbak_$CURRENT_TIME" -a $DATABASE
sql $TARGETMAIL
      sleep 1
    }
    while true
    do
      CURRENT_TIME=$(date +%H%M%S)
      if [ $CURRENT_TIME = $RUNTIME ];then
        echo "starting bak mysql database"
        sendmail
        continue
      else
        echo $CURRENT_TIME
        sleep 1
      fi
    done
```

5.2 创建有状态服务部署文件

我们的deploy文件使用的是上篇文章中创建的mysql有状态服务的yaml,有兴趣的可以看下我上篇迁的文章

```yaml
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: mysql
  namespace: solo
spec:
  serviceName: mysql-service
  selector:
    matchLabels:
      app: mysql
  replicas: 1
  template:
    metadata:
      labels:
        app: mysql
    spec:
      containers:
      - name: mysqlmail-bak
        imagePullPolicy: IfNotPresent
        image: mysqlmail-bak:1.0.1
        volumeMounts:
        - name: mysqlmail-conf
          mountPath: /app/application.yml
          subPath: application.yml
        - name: mysqlmail-shell
          mountPath: /app/demo.sh
          subPath: demo.sh
      - name: mysql-pod
        imagePullPolicy: IfNotPresent
        image: mysql:5.7
        env:
        - name: MYSQL_ROOT_PASSWORD
          value: "123456"
        ports:
        - containerPort: 3306
          name: msyql-listin
        volumeMounts:
        - name: mysql-data
          mountPath: /var/lib/mysql
          subPath: mysql-data
        - name: mysql-conf
          mountPath: /etc/mysql/conf.d/my.cnf
          subPath: my.cnf
      volumes:
      - name: mysql-data
        hostPath:
          path: /data/mysql
      - name: mysql-conf
        configMap:
          name: mysql-conf
      - name: mysqlmail-conf
```

```
        configMap:
          name: mysqlmail-conf
      - name: mysqlmail-shell
        configMap:
          name: mysqlmail-shell
---

apiVersion: v1
kind: Service
metadata:
  name: mysql-service
  namespace: solo
  labels:
    app: mysql
spec:
  ports:
  - targetPort: 3306
    port: 3306
  clusterIP: None
  selector:
    app: mysql
```

六、测试

我们上面给他定的时间是RUNTIME: 105800，上海时区也就是18点58分，我们来看一下效果

查看日志，

注意： 当一个pod包含多个容器时，要使用 -c 参数指定查看哪个容器

```
[root@VM-24-15-centos solo]# kubectl logs -n solo mysql-0  -c mysqlmail-bak | grep mysql
C 5
105755
105756
105757
105758
105759
starting bak mysql database
mysqldump: [Warning] Using a password on the command line interface can be insecure.
105801
105802
```

从日志可以看到，邮件已经发送成功了！我们来去邮箱看一下，发现也已经成功了，至此我们的实验美完成！

原文链接：K8S 中使用边车容器定时备份数据库并发送邮件