# 环境准备

k8s平台集群环境部署完成能正常使用；

镜像仓库正常运行；

# 制作镜像

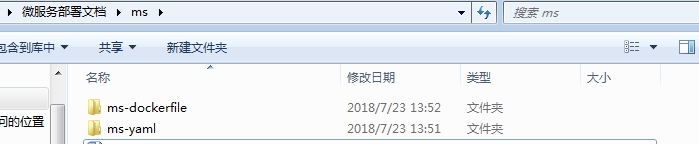
前提条件：基础镜像tomcat7+jdk1.8已经制作完成;

### ms软件包上传服务器

MS\_R1.0.0\_1195.tar.gz

### 创建context.xml、run.sh所需的配置文件

本文所需的dockerfile文件和yaml文件都在ms文件夹中;



### 创建Dockerfile文件，文件内容如下示例：



172.16.18.5:30088/admin/centos7.1-v1-tomcat7\_jdk1.8:v4为tomcat7+jdk1.8的镜像

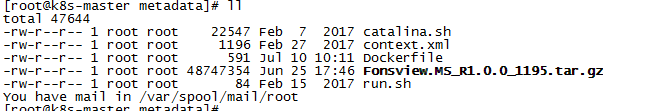
EXPOSE 22 6600其中6600为tomcat端口

ENV VERSION 1195 其中1195为ms的版本号

ADD context.xml /opt/fonsview/3RD/tomcat7.0.63/conf/context.xml为context.xml文件对应的路径；

### 创建镜像

将ms软件包、context.xml、run.sh、Dockerfile放置至服务器的同一目录上，执行以下命令创建镜像：



docker build -t 172.16.18.5:30088/admin/centos7.1-v1-ms:1195 .

其中，172.16.18.5:30088为镜像仓库地址，admin是用户名，1195为ms的版本号“.”表示创建到当前目录；

创建好镜像后，可以用docker images | grep ms命令查看创建的镜像，如下：



### 镜像上传至仓库

docker push 172.16.18.5:30088/admin/centos7.1-v1-ms:1195

# 创建ms的yaml文件

### 创建ms的配置文件mdms.cm.yaml

mdms.cm.yaml文件的类型是ConfigMap，里边定义了ms使用到的相关配置，可根据需要对应的增加、修改、减少配置

用如下命令：

kubectl create -f mdms.cm.yaml

用kubectl get cm | grep mdms可以查看到创建的mdms-config，如下图：



用命令kubectl describe cm mdms-config可以查看mdms-config的内容

后续需要修改mdms.cm.yaml的，修改完成后，

执行如下命令替换生效：

kubectl replace -f mdms.cm.yaml

mdms.cm.yaml（看红色字体部分说明）的内容如下：

apiVersion: v1

kind: ConfigMap

metadata:

name: mdms-config

data:

db-config: | //此处配置的是ms微服务连接的数据库的信息，根据现场实际信息配置

<?xml version='1.0' encoding='utf-8'?>

<Context>

<WatchedResource>WEB-INF/web.xml</WatchedResource>

<Resource name="jdbc/myDataSource" auth="Container"

type="javax.sql.DataSource"

driverClassName="com.mysql.jdbc.Driver"

url="jdbc:mysql://cms-db:3306/cmsdb?useUnicode=true&amp;characterEncoding=utf8"

username="root"

password="admin"

maxActive="100"

maxIdle="1"

maxWait="10000"

/>

</Context>

java\_opts: | //此处配置ms的pod中tomcat运行的内存，可根据实际需要配置调整

"-Djava.awt.headless=true -Dfile.encoding=UTF-8

-server -Xms2048m -Xmx2048m -XX:PermSize=256m -XX:MaxPermSize=256m"

ms-etc: | //此处配置ms的etc下的配置信息

MultiCastProtocol=udp

verifyHMAC=0 //是否要hmac校验，0是不校验，1是校验

checkoutTimeout=5000 //以下5项是连接数据库相关的配置

minPoolSize=20

maxPoolSize=1000

maxStatements=1000

acquireIncrement=2

filter\_area=\u5730\u533a //此处三项，是获取筛选条件相关的配置，和cms中基础数据🡪筛选条件中一致即可

filter\_year=\u5e74\u4efd

filter\_type=\u7c7b\u578b

pictureUrlIP\_old=picture //以下两项是配置元数据的图片地址

pictureUrlIP\_new=http://10.192.116.101:6600/media/picture

### 创建ms的配置文件mdms.svc.yaml

mdms.svc.yaml文件的类型是Service,里边定义了ms的service相关的内容，

具体配置描述的说明请参考“yaml文件说明V1.0.0.doc”中的Service yaml格式说明；mdms.svc.yaml的内容如下：

apiVersion: v1

kind: Service

metadata:

name: metadata

namespace: default

labels:

app: metadata

spec:

selector:

app: metadata

type: NodePort

ports:

- port: 80

targetPort: 6600

mdms.svc.yaml编辑完成保存后，使用如下命令创建ms的service:

kubectl create -f mdms.svc.yaml

用kubectl get service | grep mdms命令查看生成的ms的service，如下图：



后续需要修改mdms.svc.yaml文件，用如下命令替换生效：

kubectl replace -f mdms.svc.yaml

### 创建ms的配置文件mdms.deploy.yaml

mdms.deploy.yaml文件的类型是Deployment，里边定义了ms部署pod所需要的内容，具体配置描述的说明请参考“yaml文件说明V1.0.0.doc”；mdms.deploy.yaml文件内容如下：

apiVersion: extensions/v1beta1

kind: Deployment

metadata:

name: metadata

namespace: default

spec:

replicas: 1 //定义ms的pod的副本数为1

selector:

matchLabels:

app: metadata

template:

metadata:

creationTimestamp: null

labels:

app: metadata

name: metadata

spec:

containers:

- image: harbor.fonsview.com:30080/openplatform/metadata:1195 //ms的镜像名

imagePullPolicy: IfNotPresent

name: metadata

env:

- name: JAVA\_OPTS

valueFrom:

configMapKeyRef:

name: mdms-config

key: java\_opts

volumeMounts:

- name: db-config

mountPath: /opt/fonsview/3RD/tomcat7.0.63/conf/context.xml // context.xml文件路径，里边配置有ms连接的数据库的信息；

subPath: context.xml

- name: sdk-config

mountPath: /opt/fonsview/3RD/tomcat7.0.63/conf/api-gateway.properties // api-gateway.properties文件的路径

subPath: api-gateway.properties

- name: ms-etc

mountPath: /opt/fonsview/NE/metadata/etc/config.properties // config.properties文件的路径

subPath: config.properties

- name: run-config

mountPath: /run.sh

subPath: run.sh

- name: log4j-config

mountPath: /opt/fonsview/NE/metadata/etc/log4j.properties // log4j.properties文件的路径

subPath: log4j.properties

resources:

requests:

cpu: "4"

memory: 4Gi

limits:

cpu: "4"

memory: 8Gi

volumes:

- name: db-config

configMap:

name: mdms-config

items:

- key: db-config

path: context.xml

- name: sdk-config

configMap:

name: sdk-config

items:

- key: apigw\_addr

path: api-gateway.properties

- name: ms-etc

configMap:

name: mdms-config

items:

- key: ms-etc

path: config.properties

- name: run-config

configMap:

name: sdk-config

items:

- key: run\_sh

path: run.sh

- name: log4j-config

configMap:

name: log4j-config

items:

- key: log4j-config

path: log4j.properties

dnsPolicy: ClusterFirst

imagePullSecrets:

- name: openplatform

restartPolicy: Always

mdms.deploy.yaml文件编辑完成保存后，使用如下命定创建deployment

kubectl create -f mdms.deploy.yaml

执行以上命令后用kubectl get pod | grep metadata(mdms.deploy.yaml中用了metadata做标签)命令可以查看到生成一个新的pod，如下图：



后续有需要修改mdms.deploy.yaml后，执行如下命令生效：

kubectl replace -f mdms.deploy.yaml

替换后容器不会重启生效，需要删除之前的容器重新部署

kubectl delete pod podname  (可以通过kubectl get pod查看podname)

可用如下命令进入pod查看配置文件的路径及内容是否正确：

kubectl exec -it metadata-1942294662-wkx96 bash （metadata-1942294662-wkx96为pod名）

# 访问ms微服务

用如下命令分别查看ms的pod所在的节点和NodePort，

查看pod所在的节点在k8s-node1上：

kubectl get pod -o wide| grep metadata 

查看pod的NodePort

kubectl get svc | grep metadata 

通过接口工具或者命令请求ms微服务的接口，ip为pod所在的节点的ip，端口为NodePort，如下示例：

curl -v <http://podip:32353/metadata/v1/channel/all/list?sp>=