

# Dockerfile构建OPC环境及部署

- 主机环境设置
- 构建MySQL数据库
- 部署zookeeper服务
- Dockerfile构建CBS和BDS服务
  - 安装docker环境
  - 准备目录结构
  - 创建容器启动执行的脚本
  - 创建supervisor配置文件
  - 创建Dockerfile文件
  - 构建镜像

软件	版本	来源
centos	7.4.1708	官网镜像
docker	1.13	yum安装
openjdk	1.8	yum安装
supervisor	3.1.4	yum安装
tomcat	7.0.7	内网共享服务器
zookeeper	3.4.10	官网下载
mysql	5.7.24	docker镜像仓库 <a href="https://hub.docker.com">https://hub.docker.com</a>
BDS	2.0	内网POC环境打包下载
CBS	2.0	内网POC环境打包下载

## 主机环境设置

打开主机网络数据转发（调整内存参数）

```
$ systemctl stop firewalld
$ systemctl disable firewalld
$ echo "net.ipv4.ip_forward=1" >> /etc/sysctl.conf
$ sysctl -p
```

## 构建MySQL数据库

```
$ docker pull mysql:5.7.24    ## mysql
$ mkdir /data                ##
$ docker run --name poc_db --restart=always -e MYSQL_ROOT_PASSWORD=123456
-p 3306:3306 -d mysql:5.7.24
## mysqldockerMYSQL_ROOT_PASSWORDmysqlroot3306
```

```
$ mysql -uroot -p -h 127.0.0.1    ## mysql
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.24 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.
```

```
MySQL [(none)]> CREATE DATABASE bds CHARACTER SET utf8 COLLATE
utf8_general_ci;
Query OK, 1 row affected (0.01 sec)
```

```
MySQL [(none)]> CREATE DATABASE cbs CHARACTER SET utf8 COLLATE
utf8_general_ci;
Query OK, 1 row affected (0.01 sec)
```

```
MySQL [(none)]> CREATE USER 'bds'@'%' IDENTIFIED BY 'bds';
Query OK, 0 rows affected (0.02 sec)
```

```
MySQL [(none)]> CREATE USER 'cbs'@'%' IDENTIFIED BY 'cbs';
Query OK, 0 rows affected (0.02 sec)
```

```
MySQL [(none)]> GRANT ALL ON bds.* TO 'cbs'@'%%';
Query OK, 0 rows affected (0.02 sec)
```

```
MySQL [(none)]> CREATE USER 'cbs'@'%' IDENTIFIED BY 'bds';
Query OK, 0 rows affected (0.01 sec)
```

```
MySQL [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)
```

```
MySQL [(none)]> exit
```

Bye

##

```
$ mysql -uroot -p -h 127.0.0.1 bds < bds_db_file.sql
```

```
$ mysql -uroot -p -h 127.0.0.1 cbs < cbs_db_file.sql
```

## harborharbor:

<http://wiki.odc.sunline.cn/pages/viewpage.action?pageId=14189122>

```
$ docker commit odc-opc-db 10.22.60.25/overseas_core/odc_poc_mysql:v2.0
```

```
$ docker push 10.22.60.25/overseas_core/odc_poc_mysql:v2.0
```

## 部署zookeeper服务

```
zookeeper
2181
```

## Dockerfile构建CBS和BDS服务

### 安装docker环境

可以使用源码、二进制或RPM的安装方式，此处为了安装方便快捷使用的yum安装，默认版本为1.13版本

```
$ yum install docker -y
$ docker -version
Client:
Version:      1.13.1
API version:  1.26
Package version: docker-1.13.1-91.git07f3374.el7.centos.x86_64
Go version:   go1.10.3
Git commit:   07f3374/1.13.1
Built:        Wed Feb 13 17:10:12 2019
OS/Arch:      linux/amd64

Server:
Version:      1.13.1
API version:  1.26 (minimum version 1.12)
Package version: docker-1.13.1-91.git07f3374.el7.centos.x86_64
Go version:   go1.10.3
Git commit:   07f3374/1.13.1
Built:        Wed Feb 13 17:10:12 2019
OS/Arch:      linux/amd64
Experimental: false

$ systemctl restart docker
```

### 准备目录结构

注意：所用的目录及文件都放在同一个目录下（目录结构如下）

```
$ tree -L 3
.
bds_pocdb.sql  ##
cbs_pocdb.sql  ##
core-server.ini  ## supervisor
Dockerfile     ## Dockerfile buildDockerfile
entrypoint.sh  ## supervisorCBSBDSsupervisor
sunline        ## zookeeper
  bds
    bin
    conf
    lib
    LICENSE
    logs
    NOTICE
    RELEASE-NOTES
    RUNNING.txt
    temp
    webapps
    work
  cbs
    bin
    etc
    lib
    log
zookeeper-3.4.10  ## zookeeper
  bin
  .....
```

## 创建容器启动执行的脚本

entrypoint.sh

```
entrypoint.sh

#!/bin/sh

# Start supervisor in foreground
exec supervisord -n -c /etc/supervisord.conf  ## supervisor
```

## 创建supervisor配置文件

(安装是在Dockerfile build的时候使用yum安装)

创建core-server.ini文件(这个文件是以后放在docker镜像中的, supervisor的配置文件)

#### core-server.ini

```
[program:cbs-bat]
priority = 1
user = cbs
command = /sunline/cbs/bin/ltts start bat
stdout_logfile = /sunline/cbs/log/onl/console.log
stderr_logfile = /sunline/cbs/log/onl/console.log
autorestart = true

[program:cbs-onl]
priority = 2
user = cbs
command = /sunline/cbs/bin/ltts start onl
stdout_logfile = /sunline/cbs/log/bat/console.log
stderr_logfile = /sunline/cbs/log/bat/console.log
autorestart = true

[program:bds]
priority = 3
user = root
command = su bds /sunline/bds/bin/catalina.sh run ## roottomcat
stdout_logfile = /sunline/bds/logs/catalina.out
stderr_logfile = /sunline/bds/logs/catalina.out
autorestart = true
```

创建Dockerfile文件

## Dockerfile

```
# This dockerfile uses the ubuntu image
FROM docker.io/centos:7.5.1804
ARG Version
MAINTAINER houfei@sunline.cn
ENV LC_ALL en_US.UTF-8
ENV LANG en_US.UTF-8

RUN yum install java-1.8.0-openjdk \
    epel-release -y && \
    yum install supervisor -y && \
    yum clean all && \
    rm -rf /var/cache/yum/*

COPY core-server.ini /etc/supervisord.d/core-server.ini

COPY sunline /sunline
COPY entrypoint.sh /opt/entrypoint.sh
RUN chmod 755 /opt/entrypoint.sh && \
    groupadd cbs -g 2000 && \
    groupadd bds -g 2100 && \
    useradd cbs -u 2000 -g 2000 && \
    useradd bds -u 2100 -g 2100 && \
    chown -R cbs:cbs /sunline/cbs && \
    chown -R bds:bds /sunline/bds
LABEL version=v2.0
LABEL description="ODC_POC"

EXPOSE 8080

CMD /opt/entrypoint.sh
```

## 构建镜像

```
$ docker build -t odc-poc:v2.0 . ## build

$ docker run -it --privileged=true -p 8080:8080 -d odc-poc:v2.0
# ip:8080
```

