电商安装部署

一、环境准备

说明：如果已经安装过相关工具就忽略

1 安装 JAVA 运行环境

第一步：上传或下载安装包

cd /usr/local

jdk-8u152-linux-x64.tar.gz

第二步：解压安装包

tar -zxvf jdk-8u152-linux-x64.tar.gz

第三步：建立软连接

ln -s /usr/local/jdk1.8.0_152/ /usr/local/jdk

第四步：修改环境变量

vim /etc/profile

```
export JAVA_HOME=/usr/local/jdk
export JRE_HOME=$JAVA_HOME/jre
export CLASSPATH=.:$CLASSPATH:$JAVA_HOME/lib:$JRE_HOME/lib
export PATH=$PATH:$JAVA_HOME/bin:$JRE_HOME/bin
```

通过命令 source /etc/profile 让 profile 文件立即生效

source /etc/profile

第五步、测试是否安装成功

使用 java -version，出现版本为 java version "1.8.0_152"

2 安装 maven

第一步：上传或下载安装包

cd /usr/local

apache-maven-3.6.1-bin.tar.gz

第二步：解压安装包

tar -zxvf apache-maven-3.6.1-bin.tar.gz

第三步：建立软连接

ln -s /usr/local/apache-maven-3.6.1/ /usr/local/maven

第四步：修改环境变量

vim /etc/profile

export MAVEN_HOME=/usr/local/maven

export PATH=$PATH:$MAVEN_HOME/bin

通过命令 source /etc/profile 让 profile 文件立即生效

source /etc/profile

第五步、测试是否安装成功

mvn –v

3 安装 docker

环境安装：

yum -y install gcc-c++

第一步：安装必要的一些系统工具

yum install -y yum-utils device-mapper-persistent-data lvm2

第二步：添加软件源信息

yum-config-manager --add-repo http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo

第三步：更新并安装 Docker-CE

yum makecache fast

yum -y install docker-ce

第四步：开启 Docker 服务

systemctl start docker

systemctl enable docker

第五步：测试是否安装成功

docker -v

第六步：配置镜像加速器

您可以通过修改 daemon 配置文件/etc/docker/daemon.json 来使用加速器

sudo mkdir -p /etc/docker

sudo tee /etc/docker/daemon.json <<-'EOF'

```
{
  "registry-mirrors": ["https://ldu6wrsf.mirror.aliyuncs.com"]
}
EOF
sudo systemctl daemon-reload
sudo systemctl restart docker
```

4 安装 mysql

已安装或能访问忽略

第一步：拉取镜像

```
docker pull mysql:5.7
```

第二步：启动

```
docker run --name mysql --restart=always -v /home/ljaer/mysql:/var/lib/mysql -p 3306:3306 -e MYSQL_ROOT_PASSWORD=root -d mysql:5.7
```

第三步：测试 mysql

进入容器：

```
docker exec -it mysql /bin/bash
```

登录 mysql：

```
mysql -u root -p
```

如果顺利进入，安装成功

5 安装 rabbitmq

第一步：拉取镜像

```
docker pull rabbitmq:management
```

第二步：启动

```
docker run -d -p 5672:5672 -p 15672:15672 --restart=always --name rabbitmq
```

```
rabbitmq:management
```

第三步：安装延迟队列插件

1. 首先下载 rabbitmq_delayed_message_exchange-3.9.0.ez 文件上传到 RabbitMQ 所在服务器，下载地址：https://www.rabbitmq.com/community-plugins.html

2. 切换到插件所在目录，执行 docker cp rabbitmq_delayed_message_exchange-3.9.0.ez rabbitmq:/plugins 命令，将刚插件拷贝到容器内 plugins 目录下

3. 执行 docker exec -it rabbitmq /bin/bash 命令进入到容器内部，并 cd plugins 进入 plugins 目录

4. 执行 ls -l|grep delay 命令查看插件是否 copy 成功

5. 在容器内 plugins 目录下，执行 rabbitmq-plugins enable rabbitmq_delayed_message_exchange 命令启用插件

6. exit 命令退出 RabbitMQ 容器内部，然后执行 docker restart rabbitmq 命令重启 RabbitMQ 容器

6 安装 redis

已安装或能访问忽略

第一步：拉取镜像

docker pull redis:latest

第二步：启动

```
docker run -d -p 6379:6379  --restart=always redis:latest redis-server
```

7 安装 nacos

已安装或能访问忽略

第一步：拉取镜像

docker pull nacos/nacos-server:1.4.1

第二步：启动

```
docker run --env MODE=standalone --name nacos --restart=always -d -p 8848:8848 -e JVM_XMS=512m -e JVM_XMX=512m nacos/nacos-server:1.4.1
```

8 安装 sentinel

已安装或能访问忽略

第一步：拉取镜像

docker pull bladex/sentinel-dashboard

第二步：启动

```
docker run --name sentinel-dashboard --restart=always -p 8858:8858 -d bladex/sentinel-dashboard:latest
```

9 安装 elasticsearch

已安装或能访问忽略

第一步：拉取镜像

docker pull elasticsearch:7.8.0

第二步：启动

需要建立：两个文件夹

mkdir -p /mydata/elasticsearch/plugins

mkdir -p /mydata/elasticsearch/data

授予权限 chmod 777 /mydata/elasticsearch/data

```
docker run -p 9200:9200 -p 9300:9300 --name elasticsearch --restart=always \
-e "discovery.type=single-node" \
-e ES_JAVA_OPTS="-Xms512m -Xmx512m" \
```

```
-v /mydata/elasticsearch/plugins:/usr/share/elasticsearch/plugins \
-v /mydata/elasticsearch/data:/usr/share/elasticsearch/data \
-d elasticsearch:7.8.0
```

第三步：安装中文分词器

1. 下载 elasticsearch-analysis-ik-7.8.0.zip

2. 上传解压：unzip elasticsearch-analysis-ik-7.8.0.zip -d ik-analyzer

3. 上传到 es 容器：docker cp ./ik-analyzer a24eb9941759:/usr/share/elasticsearch/plugins

4. 重启 es：docker restart a24eb9941759

a24eb9941759：表示容器 ID 运行时，需要改成自己的容器 ID

10 安装 kibana

第一步：拉取镜像

docker pull kibana:7.8.0

第二步：启动

```
docker run --name kibana --restart=always -e ELASTICSEARCH_URL=http://192.168.254.156:9200 -p 5601:5601 -d kibana:7.8.0
```

进入容器修改：docker exec -it 1e12f8dd3efd /bin/bash

cd config

vi kibana.yml

elasticsearch.hosts: ["http://192.168.200.129:9200"]

docker restart 1dc0f78d78ad 重启 kibana ！

测试：安装分词词库是否可以使用！

GET /.kibana/_analyze

{

```
  "text": "我是中国人",
  "analyzer": "ik_max_word"
}
```

11 安装 zipkin

第一步：拉取镜像

docker pull openzipkin/zipkin

第二步：启动

```
docker run --name zipkin --restart=always -d -p 9411:9411 openzipkin/zipkin
```

12 安装 minio

已安装或能访问忽略

第一步：拉取镜像

docker pull minio/minio

第二步：启动

```
docker run \
  -p 9000:9000 \
  -p 9001:9001 \
  --name minio \
  -d --restart=always \
  -e "MINIO_ROOT_USER=admin" \
  -e "MINIO_ROOT_PASSWORD=admin123456" \
  -v /home/data:/data \
  -v /home/config:/root/.minio \
  minio/minio server /data --console-address ":9001"
```

浏览器访问：http://IP:9000/minio/login，

13 安装 logstash

第一步：拉取镜像

docker pull logstash:7.8.0

第二步：启动

```
docker run --name logstash -p 5044:5044 --restart=always --link elasticsearch:es -v /mydata/logstash/logstash.conf:/usr/share/logstash/pipeline/logstash.conf -d logstash:7.8.0
```

--net root_default \

```
# 需要提前在 linux 服务器上环境 /mydata/logstash/logstash.conf

        logstash.conf

        input {
          tcp {
                mode => "server"
                host => "0.0.0.0"
                port => 5044
                codec => json_lines
          }
        }
        filter{


        }
        output {
          elasticsearch {
                hosts => "192.168.200.128:9200"
                index => "gmall-%{+YYYY.MM.dd}"
```

```
        }
    }
```

注意：

停止所有的容器

docker stop $(docker ps -aq)

删除所有的容器

docker rm $(docker ps -aq)

#删除所有的镜像

docker rmi $(docker images -q)

问题：

Docker 容器做端口映射报错

docker: Error response from daemon: driver failed programming external connectivity on endpoint lamp3 (46b7917c940f7358948e55ec2df69a4dec2c6c7071b002bd374e8dbf0d40022c): (iptables failed: iptables --wait -t nat -A DOCKER -p tcp -d 0/0 --dport 86 -j DNAT --to-destination 172.17.0.2:80 ! -i docker0: iptables: No chain/target/match by that name.

解决方法

docker 服务启动时定义的自定义链 DOCKER 被清除

重启即可 systemctl restart docker