# Docker安装大数据组件手册

## 需要下载的东西

1. 创建共享文件夹:

mkdir -p /home/docker/share/

cd /home/docker/share/

1. 下载镜像包

wget <http://mirror.nsc.liu.se/centos-store/altarch/7.6.1810/isos/aarch64/CentOS-7-aarch64-Everything-1810.iso>

1. 下载JDK安装包

wget <https://github.com/AdoptOpenJDK/openjdk8-binaries/releases/download/jdk8u191-b12/OpenJDK8U-jdk_aarch64_linux_hotspot_8u191b12.tar.gz>

1. redis需要下载东西：

wget <https://download.redis.io/releases/redis-5.0.9.tar.gz>

1. es需要下载的东西

wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-5.6.12.tar.gz

1. zk需要下载的东西

wget https://mirror.bit.edu.cn/apache/zookeeper/zookeeper-3.6.1/apache-zookeeper-3.6.1-bin.tar.gz

wget https://mirrors.huaweicloud.com/kunpeng/maven/jline/jline/2.11/jline-2.11.jar

## 搭建docker环境

1. 安装docker：

yum install docker

1. 启动docker:

systemctl daemon-reload

systemctl restart docker.service

## Redis安装

### 方法一：使用git的项目直接构建redis镜像，但是需要外网（建议）

虽然docker上已经有了redis5.0.9镜像，但是经过验证，运行后会出现错误，需要用github上的源码直接进行构建。

clone github上的redis项目:

git clone <https://github.com/docker-library/redis.git>

cd redis/5.0

构建redis5.0.9镜像（需要外网环境）：

docker build -t arm64v8/cetos\_redis:5.0.9 .

docker run -p 6379:6379 -v $PWD/data:data -d redis:5.0.9 reids-server --appendonly yes

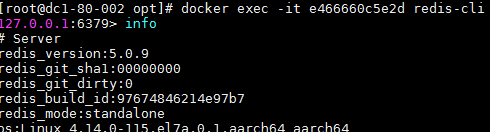
用docker ps –a可查看容器已经正常运行：



验证：

docker exec -it e466660c5e2d redis-cli

127.0.0.1:6379> info



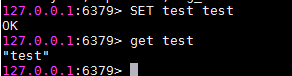
127.0.0.1:6379> SET test test

OK

127.0.0.1:6379> get test

"test"

127.0.0.1:6379>

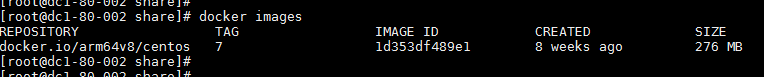


### 方法二：使用源码安装，使用过程，需要自己配置较多参数

#### 创建容器

拉取centos7镜像：docker pull centos:centos7

查看自己镜像ID：docker images



1. 使用该centos镜像运行一个容器，将共享目录挂载到容器的/home目录

docker run -it --privileged=true -v /home/docker/share/:/home 1d353df489e1 bash

1. 配置本地yum源：

挂载镜像：

mount -o loop /home/CentOS-7-aarch64-Everything-1810.iso /media

配置repo:

mv /etc/yum.repos.d/ /etc/yum.repos.d-bak

mkdir /etc/yum.repos.d

vi /etc/yum.repos.d/local.repo

增加以下内容：

[Local]

name=CentOS-7.6 Local

baseurl=file:///media/

enabled=1

gpgcheck=0

使yum源生效：

yum clean all

yum makecache

##### 安装依赖环境

1. 安装相关工具

yum install gcc.aarch64 gcc-c++.aarch64 gcc-gfortran.aarch64 libgcc.aarch64

yum install make

yum install net-tools

yum install tcl

yum install which

安装过程中可能会提示依赖库版本不对，请依照FAQ解决。

1. 安装libatomic.so

先用find / -name libatomic.so\*找一下是否有相关so文件，如果有加上软链接

ln -s /usr/lib64/libatomic.so.1 /usr/lib64/libatomic.so

如果没有,yum下载一下

yum install libatomic

ln -s /usr/lib64/libatomic.so.1 /usr/lib64/libatomic.so

### 安装redis：

cd /opt

cp /home/redis-5.0.9.tar.gz /opt

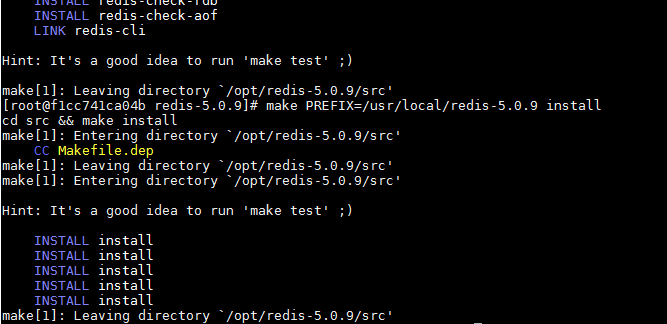
tar -zxvf redis-5.0.9.tar.gz

cd redis-5.0.9

make

make PREFIX=/usr/local/redis-5.0.9 install

编译成功：



### 启动reds

进入“/usr/local”目录。

cd /usr/local

建立软链接，便于后期版本更换。

ln -s redis-3.0.5 redis

打开配置文件。

vi /etc/profile

添加Redis到环境变量：

export REDIS\_HOME=/usr/local/redis

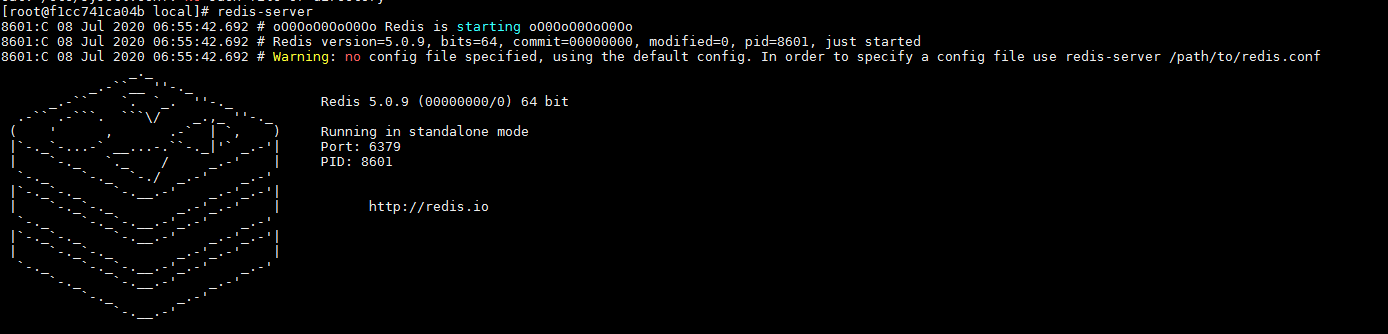
export PATH=$REDIS\_HOME/bin:$PATH

环境变量生效。

source /etc/profile

启动redis，直接运行：

redis-server



### 验证

在物理机上，直接连接上述容器：

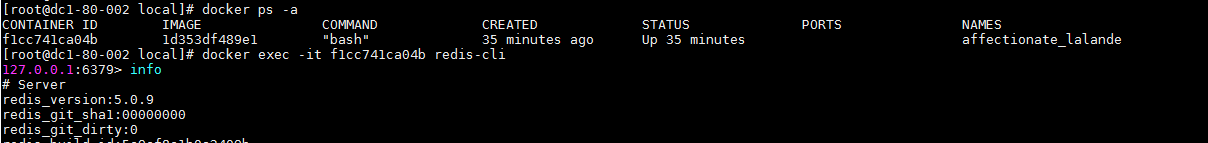
查询容器ID:

docker ps -a

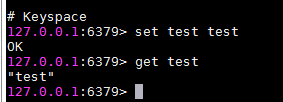


连接redis:

docker exec -it f1cc741ca04b redis-cli



测试：

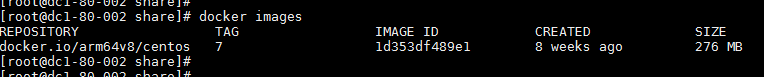


## ElasticSearch

### 创建容器

拉取centos7镜像：docker pull centos:centos7

查看自己镜像ID：docker images



1. 使用该centos镜像运行一个容器，将共享目录挂载到容器的/home目录

docker run -it --privileged=true -v /home/docker/share/:/home 1d353df489e1 bash

1. 配置本地yum源：

挂载镜像：

mount -o loop /home/CentOS-7-aarch64-Everything-1810.iso /media

配置repo:

mv /etc/yum.repos.d/ /etc/yum.repos.d-bak

mkdir /etc/yum.repos.d

vi /etc/yum.repos.d/local.repo

增加以下内容：

[Local]

name=CentOS-7.6 Local

baseurl=file:///media/

enabled=1

gpgcheck=0

使yum源生效：

yum clean all

yum makecache

### 安装jdk：

mkdir -p /opt/tools/installed/

cp /home/OpenJDK8U-jdk\_aarch64\_linux\_hotspot\_8u191b12.tar.gz /opt/tools/installed/

cd /opt/tools/installed/

tar -zxf OpenJDK8U-jdk\_aarch64\_linux\_hotspot\_8u191b12.tar.gz

vi /etc/profile

在“/etc/profile”文件末尾处增加下面的代码:

export JAVA\_HOME=/opt/tools/installed/jdk8u191-b12

export PATH=$JAVA\_HOME/bin:$PATH

按ESC键，输入:wq保存并退出。

运行下面命令，使修改的环境变量生效。

source /etc/profile

### 单机部署elasticsearch

#### 解压缩安装包

cp /home/elasticsearch-5.6.12.tar.gz /usr/local/

cd /usr/local/

tar -zxvf elasticsearch-5.6.12.tar.gz

ln -s elasticsearch-5.6.12 elasticsearch

vi /etc/profile

添加如下内容：

export ES\_HOME=/usr/local/elasticsearch

export PATH=$ES\_HOME/bin:$PATH

保存后执行：

source /etc/profile

#### 创建elasticsearch用户：

useradd elasticsearch

chown -R elasticsearch:elasticsearch /usr/local/elasticsearch

chown -R elasticsearch:elasticsearch /usr/local/elasticsearch-5.6.12

mkdir /home/elasticsearch/data

mkdir /home/elasticsearch/logs

chown -R elasticsearch:elasticsearch /home/elasticsearch/data

chown -R elasticsearch:elasticsearch /home/elasticsearch/logs

#### 修改配置(按照自己需求修改)

##### 修改elasticsearch.yml

cd /usr/local/elasticsearch

vim config/elasticsearch.yml

修改配置为 network.host: 0.0.0.0

##### 修改系统环境变量

查看系统环境变量vm.max\_map\_count：

sysctl -a|grep vm.max\_map\_count

如果不是262144，则修改：

vim/etc/sysctl.conf

末尾添加 vm.max\_map\_count=262144

修改完后需要重启才能生效，使用下面命令使得本次启动有效：

sysctl -w vm.max\_map\_count=262144

##### 修改JVM分配大小

根据情况设置，我这里是VM虚拟机，给的大小是512m，即：

-Xms2g

-Xms2g

##### 修改最大文件描述符数量和用户最大线程数

vi /etc/security/limits.conf

添加下面代码至文件末尾：

\* soft nofile 65536

\* hard nofile 65536

\* soft nproc 4096

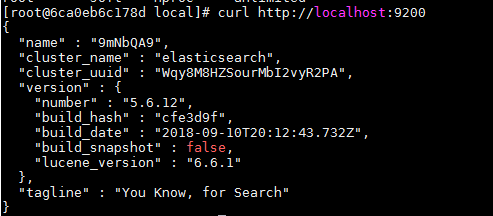
\* hard nproc 4096

### 启动验证

su elasticsearch

./bin/elasticsearch –d

curl <http://localhost:9200>

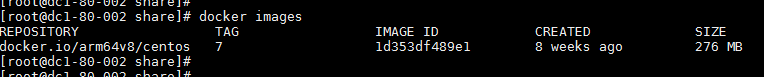


## Zookeeper

### 创建容器

拉取centos7镜像：docker pull centos:centos7

查看自己镜像ID：docker images



1. 使用该centos镜像运行一个容器，将共享目录挂载到容器的/home目录

docker run -it --privileged=true -v /home/docker/share/:/home 1d353df489e1 bash

1. 配置本地yum源：

挂载镜像：

mount -o loop /home/CentOS-7-aarch64-Everything-1810.iso /media

配置repo:

mv /etc/yum.repos.d/ /etc/yum.repos.d-bak

mkdir /etc/yum.repos.d

vi /etc/yum.repos.d/local.repo

增加以下内容：

[Local]

name=CentOS-7.6 Local

baseurl=file:///media/

enabled=1

gpgcheck=0

使yum源生效

yum clean all

yum makecache

### 安装jdk

mkdir -p /opt/tools/installed/

cp /home/OpenJDK8U-jdk\_aarch64\_linux\_hotspot\_8u191b12.tar.gz /opt/tools/installed/

cd /opt/tools/installed/

tar -zxf OpenJDK8U-jdk\_aarch64\_linux\_hotspot\_8u191b12.tar.gz

vi /etc/profile

在“/etc/profile”文件末尾处增加下面的代码:

export JAVA\_HOME=/opt/tools/installed/jdk8u191-b12

export PATH=$JAVA\_HOME/bin:$PATH

按ESC键，输入:wq保存并退出。

运行下面命令，使修改的环境变量生效。

source /etc/profile

### 用下载的jline-2.11.jar替换x86的jline-2.11.jar包

cp /home/apache-zookeeper-3.6.1-bin.tar.gz /usr/local/

cd /usr/local/

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

cp /home/jline-2.11.jar /usr/local/zookeeper/lib/jline-2.11.jar

按y回车覆盖



使用md5sum验证是否替换成功：

md5sum /home/jline-2.11.jar /usr/local/zookeeper/lib/jline-2.11.jar



### 单机部署zookeeper

ln -s apache-zookeeper-3.6.1-bin zookeeper

vi /etc/profile

添加：

export ZOOKEEPER\_HOME=/usr/local/zookeeper

export PATH=$ZOOKEEPER\_HOME/bin:$PATH

按ESC键，输入:wq保存并退出。

运行下面命令，使修改的环境变量生效。

source /etc/profile

cd /usr/local/zookeeper/conf

cp zoo\_sample.cfg zoo.cfg

vi zoo.cfg

修改：

dataDir=/usr/local/zookeeper/tmp

添加：

server.1=Datanode1:2888:3888

mkdir /usr/local/zookeeper/tmp

touch /usr/local/zookeeper/tmp/myid

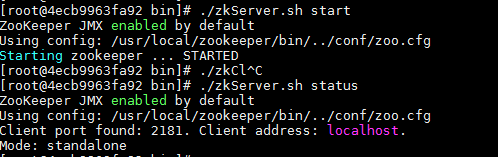
echo 1 > /usr/local/zookeeper/tmp/myid

#### 运行验证

cd /usr/local/zookeeper/bin

./zkServer.sh start

./zkServer.sh status



### 集群部署

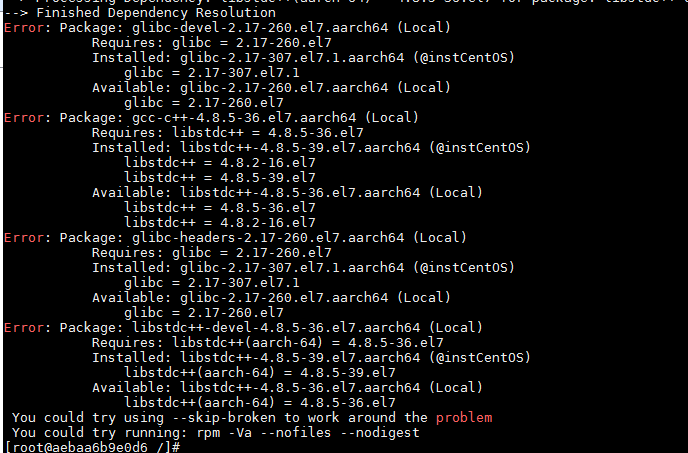
配置参照：

<https://support.huaweicloud.com/dpmg-apache-kunpengbds/kunpengzookeeper_04_0001.html>

## FAQ

### 安装报错：某些依赖包的版本过高

yum install gcc.aarch64 gcc-c++.aarch64 gcc-gfortran.aarch64 libgcc.aarch64



因为系统是centos7.8的，类库版本比较高，需要回退一下

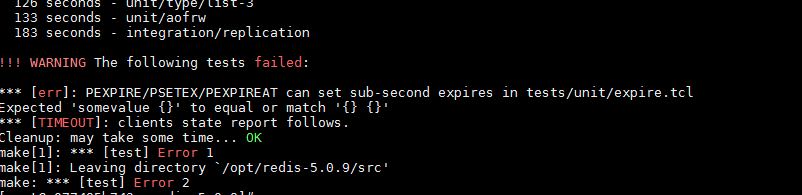
yum -y downgrade glibc glibc-common libstdc++

重新运行：

yum install gcc.aarch64 gcc-c++.aarch64 gcc-gfortran.aarch64 libgcc.aarch64

### 如果make test报错PEXPIRE/PSETEX/PEXPIREAT can set sub-second expires in tests/unit/expire.tcl Expected 'somevalue {}' to equal or match '{} {}'

如图



vi修改tests/unit/expire.tcl文件：

