链滴

Docker 实践安装软件案例集合

作者：someone33881

原文链接：https://ld246.com/article/1546526404227

来源网站：链滴

  本文主要介绍应用docker工具，安装一些软件环境诸如nginx、tomcat、mysql、redis、activemq、mongodb、kafka、zookeeper等

Docker环境安装

CentOS下安装docker

```
yum install -y yum-utils device-mapper-persistent-data  lvm2  #安装依赖包
yum-config-manager --add-repo https://mirrors.ustc.edu.cn/docker-ce/linux/centos/docker-ce.repo #配置yum软件源
yum makecache fast #更新yum软件源
yum install docker-ce #安装docker-ce

systemctl enable docker
systemctl start docker #启动docker ce

docker run hello-world #测试docker是否安装成功
```

MacOS下安装docker

```
brew cask install docker
#或者docker官网下载stable/edge版本，按照一般软件安装即可
docker --version

#更改镜像地址
#Docker for mac 应用图标 -> Perferences... -> Daemon -> Registry mirrors，在其中添加加速址如http://hub-mirror.c.163.com之后重启docker应用即可
```

Docker命令

docker build -t mytagname .

docker run ...

docker rm

docker stop

docker start

docker-machine(docker主机相关命令：如 ps、create等)

Dockerfile

docker-compose

docker-compose up

https://yeasy.gitbooks.io/docker_practice/compose/commands.html

Docker日志

1、向Rancher注册Docker所在的当前mac主机命令

```
sudo mkdir /var/lib/rancher
sudo chmod -R 777 /var/lib/rancher
sudo docker run --rm --privileged -v /var/run/docker.sock:/var/run/docker.sock -v /var/lib/ra
cher:/var/lib/rancher rancher/agent:v1.2.10 http://192.168.211.31:8080/v1/scripts/B2303546C
D40CB9666B:1514678400000:JKonbRx51onkA0Bwb312LTZ9Nk
```

docker安装其他软件

Nginx

```
mkdir -p ~/docker/nginx/www ~/docker/nginx/logs ~/docker/nginx/conf
docker pull nginx
docker run -p 80:80 --name mynginx -d nginx
docker run -p 80:80 --name mynginx -v /Users/zorke/docker/nginx/www:/www -v /Users/zor
e/docker/nginx/conf/nginx.conf:/etc/nginx/nginx.conf -v /Users/zorke/docker/nginx/logs:/w
wlogs -v /Users/zorke/docker/nginx/conf/sites-available:/etc/nginx/conf/sites-available -d ng
nx
```

Tomcat

```
mkdir -p ~/docker/tomcat/webapps ~/docker/tomcat/logs ~/docker/tomcat/conf
```

```
docker pull tomcat
docker run --name mytomcat -p 8080:8080 -d tomcat
docker start mytomcat
```

MySQL

```
mkdir -p ~/docker/mysql/data ~/docker/mysql/logs ~/docker/mysql/conf
docker pull mysql:5.7.20
docker run -p 33060:3306 --name mymysql -e MYSQL_ROOT_PASSWORD=root123 -d mysql:
.7.20
docker start mymysql
```

Redis

```
mkdir -p ~/docker/redis ~/docker/redis/data
docker pull redis
docker run -p 6380:6379 -d redis redis-server --appendonly yes

docker run --name myredis -p 6379:6379 -v /Users/zorke/docker/redis/data:/data -d redis:4.0
8 redis-server --appendonly yes
docker start myredis
```

ActiveMQ

```
docker run --name myactivemq -it  -d -v /Users/zorke/docker/activemq/data:/data/activemq
v /Users/zorke/docker/activemq/log:/var/log/activemq  -p 8161:8161 -p 61616:61616 -p 616
3:61613  webcenter/activemq:5.13.2
docker start myactivemq
```

Mongodb

```
mkdir -p ~/docker/mongo  ~/docker/mongo/db
docker pull mongo:3.2
docker run --name mymongodb -p 27017:27017 -v ~/docker/mongo/db:/data/db -d mongo:
.2
docker start mymongodb
```

Kafka(+Zookeeper)

```
#1、镜像下载
docker pull wurstmeister/zookeeper   #下载zk镜像
docker pull wurstmeister/kafka       #下载kafka镜像

#2、docker-compose文件编写
#在自己存放docker-compose.yml文件的目录下创建一个docker-compose.yml文件，内容如下
```

```
version: '2'
services:
  zookeeper:
    image: wurstmeister/zookeeper
    ports:
      - "2181:2181"
  kafka:
    image: wurstmeister/kafka
    ports:
      - "9092"
    environment:
      KAFKA_ADVERTISED_HOST_NAME: 192.168.1.9
      #KAFKA_ADVERTISED_LISTENERS: "192.168.1.9"
      KAFKA_ZOOKEEPER_CONNECT: zookeeper:2181
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
#其中ip地址为宿主机上的 docker-machine ip 地址

#3、启动docker-compose(zookeeper+一个kafka节点)【docker-compose.yml文件所在目录下执
】
docker-compose up -d

#4、若需要启动多个kafka 节点，比如3个，则在第3条的基础之上执行下述命令【docker-compose.
ml文件所在目录下执行】
docker-compose scale kafka=3   #docker ps 应该可以看到已经成功启动了一个zookeeper容器
三个Kafka容器

#5、可用性测试
docker exec -it kafka_kafka_1 /bin/bash    #进入其中一种kafka容器,假设容器名为kafka_kafka_1

$KAFKA_HOME/bin/kafka-topics.sh --create --topic test --zookeeper kafka_zookeeper_1:2181
--replication-factor 1 --partitions 1  #创建一个topic（其中假设zookeeper容器名为 kafka_zooke
per_1，topic名为test）

$KAFKA_HOME/bin/kafka-topics.sh --zookeeper kafka_zookeeper_1:2181 --describe --topic t
st #查看新创建的topic

$KAFKA_HOME/bin/kafka-console-producer.sh --topic=test --broker-list kafka_kafka_1:9092
发布消息: （输入若干条消息后 按^C 退出发布）

$KAFKA_HOME/bin/kafka-console-consumer.sh --bootstrap-server kafka_kafka_1:9092 --from
beginning --topic test  #接收消息，接收到了发布的消息，证明整个部署正常，就可以正式开始开
工作了
```

MAC-brew安装kafka

```
#1、安装（也会安装依赖zookeeper）
brew install kafka     #注意：安装目录：/usr/local/Cellar/kafka/1.1.0

#2、配置文件的位置
/usr/local/etc/kafka/server.properties
/usr/local/etc/kafka/zookeeper.properties
```

#3、启动zookeeper
zookeeper-server-start /usr/local/etc/kafka/zookeeper.properties &

#4、启动kafka
kafka-server-start /usr/local/etc/kafka/server.properties &

#5、测试kafka
kafka-topics --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic t
st #创建topic，使用单个分区和只有一个副本创建一个名为“test”的主题
kafka-topics --list --zookeeper localhost:2181 #查看创建的topic，运行list topic命令可以看到
主题
kafka-console-producer.sh --broker-list localhost:9092 --topic test #发送一些消息，Kafka提
了一个命令行客户端，它将从文件或标准输入接收输入，并将其作为消息发送到Kafka集群。默认情
下，每行都将作为单独的消息发送。运行生产者，然后在控制台中键入一些消息发送到服务器。
kafka-console-consumer --bootstrap-server localhost:9092 --topic test --from-beginning #
费消息，Kafka还有一个命令行消费者，将消息转储到标准输出