Docker部署

##### 安装docker

* 1. docker设置yum源

yum-config-manager \

--add-repo \

https://download.docker.com/linux/centos/docker-ce.repo

* 1. 安装指定版本

#查询版本列表

yum list docker-ce --showduplicates | sort -r

#指定版本安装

yum install -y docker-ce-19.03.9

* 1. 启动docker

systemctl start docker.service

* 1. 验证docker安装是否成功（有client和service两部分表示成功）

docker version

* 1. 拉取镜像

docker pull mysql:5.7.20

* 1. 使用镜像生成容器

docker run --name mynginx -d mysql:5.7.20

1. **Docker生成镜像**
   1. 生成镜像

docker commit eafd9111ada6[容器id] docker/lua

* 1. 保存镜像

docker save -o rocketmq.tar rocketmq ##-o：指定保存的镜像的名字；rocketmq.tar：保存到本地的镜像名称；rocketmq：镜像名字，通过"docker images"查看

1. **Docker加载镜像**
   1. 导入镜像

docker load < 文件名

* 1. 查看images

docker images

* 1. 加载镜像

docker run -d -it -v /var/run/dbus:/var/run/dbus -v /dev/disk/by-uuid:/dev/disk/by-uuid -v /opt/yufei.tan/docker\_test/data:/mnt/data 52020da95c36(镜像id或者名称) /bin/bash

* 1. 查看docker 镜像

docker ps

* 1. 运行对应docker镜像

docker attach 186faabdc9e9（id值）

1. **安装mysql**

3.1 拉取mysql镜像

docker pull mysql:5.7.20

3.2 mysql服务挂载到宿主机上

创建文件夹存放mysql数据及配置文件

cd /opt

mkdir mysql\_docker

cd mysql\_docker

mkdir datadir #把mysql服务器挂载在目录中

mkdir conf.d #用来存放配置文件

mysql服务挂载到宿主机

docker run --name=mysql5.7 -p 3307:3306 -v/opt/mysql\_docker/datadir:/var/lib/mysql -v /opt/mysql\_docker/conf.d:/etc/mysql/conf.d -eMYSQL\_ROOT\_PASSWORD=baifendian -d mysql:5.7.20

简单解释一下 **-v挂载**的意思：就是将/var/lib/mysql目录下的文件转移到/ opt/mysql\_docker/datadir中。好处是：当容器被销毁时，容器内数据不会随着容器消失而消失。

mysql容器容器命名为 mysql5.7 密码为baifendian -d 表示后台运行容器

3.3 运行容器

docker exec -it [容器的id] bash

docker exec -it mysql5.7 bash

docker container ls -a #查看所有container

docker container stop [容器ID] #停止container

docker rm 1c4586fe95ee #通过container id 删除container

##创建镜像

docker commit eafd9111ada6 docker/lua

## 镜像保存

docker save -o rocketmq.tar rocketmq ##-o：指定保存的镜像的名字；rocketmq.tar：保存到本地的镜像名称；rocketmq：镜像名字，通过"docker images"查看

curl -H "Content-type: application/json" -X POST -d ' {"text":"农民占道晒玉米,车辆易侧翻,漯河交警劝离又“赠”反光锥。"}' <http://127.0.0.1:10291/extract_entity>

1. **安装rabbitmq**
2. 安装erlang

a.新建文件 内容如下vim /etc/yum.repos.d/rabbitmq-erlang.repo

[rabbitmq-erlang]

name=rabbitmq-erlang

baseurl=https://dl.bintray.com/rabbitmq-erlang/rpm/erlang/21/el/7

gpgcheck=1

gpgkey=https://dl.bintray.com/rabbitmq/Keys/rabbitmq-release-signing-key.asc

repo\_gpgcheck=0

enabled=1

b.yum install erlang

1. Rabbitmq安装

*#新建一个repo文件，内容如下*

vim /etc/yum.repos.d/rabbitMQ.repo

[bintray-rabbitmq-server]

name=bintray-rabbitmq-rpm

baseurl=https://dl.bintray.com/rabbitmq/rpm/rabbitmq-server/v3.7.x/el/7/

gpgcheck=0

repo\_gpgcheck=0

enabled=1

yum install rabbitmq-server

1. 添加启动项

chkconfig rabbitmq-server on

1. 启动服务

systemctl start rabbitmq-server

1. 新增一个用户

rabbitmqctl add\_user admin bfd123

1. 设置用户角色超级管理员(administrator)

rabbitmqctl set\_user\_tags admin administrator

1. 查看当前用户列表

rabbitmqctl list\_users

1. **启动配置自动启动**

rabbitmq-plugins enable rabbitmq\_management

1. 问题解决：  
   pika.exceptions.ProbableAccessDeniedError: ConnectionClosedByBroker: (530) "NOT\_ALLOWED - access to vhost '/' refused for user 'admin'"

执行命令：

rabbitmqctl delete\_vhost next

rabbitmqctl add\_vhost /

rabbitmqctl set\_permissions -p / admin '.\*' '.\*' '.\*'

curl -H "Content-type: application/json" -X POST -d '{"text\_id": "754b8e83-921d-47d3-81f0-db39d321b6e5","title":"强降雨致四川部分地方受灾 雅安6人失联2人被埋","type":0,"post\_time":"2020-11-27 12:00:00","content": "强降雨致四川部分地方受灾雅安6人失联2人被埋。记者从四川省防汛抗旱指挥部了解到，8月10日20时至11日8时，盆地西部、川西高原降大到暴雨，雅安、眉山、成都等地降大暴雨，雅安局部降特大暴雨，降雨量超过50毫米的有19县200站，超过100毫米的有10县84站，超过250米的有2县9站，最大为芦山县玉溪站447.5毫米。受强降雨影响，青衣江干流及支流芦山河、名山河出现超保证水位洪水。据雅安市和相关区县防办电话报告，暴雨造成雅安市因灾6人失联（宝兴县2人接到电话通知转移后，因居住地偏远和交通、通信中断失联；天全县1人在开车途中失联；雨城区3人因涵管被土石等杂物堵塞致河流改道冲毁房屋，造成失联），另有2人被埋（雨城区2人因涵被土石等杂物堵塞致河流改道冲毁房屋，其中发现1人有生命体征）。目前，当地正在全力搜救失联和被埋人员。据初步统计，强降雨致雅安等地部分场镇低洼处进水，部分道路、房屋、农田、水、电力设施等不同程度受损，详细灾情正在进一步统计核实中。水利部、省防指已派出工作组赴雅安指导当地防汛救灾工作。省防指已于11日8时启动防汛Ⅳ级应急响应。（消息来源：央视新闻客端中央广播电视总台央视记者杨妮黄鹂）"}' http://127.0.0.1:9099/extract\_mark\_graph

1 ls

2 ll

3 exit

4 fdisk -l

5 fdisk /dev/vdc

6 fdisk -l

7 fdisk -u /dev/vdb

8 fdisk -lu /dev/vdb

9 mkfs -t ext4 /dev/vdb1

10 cp /etc/fstab /etc/fstab.bak

11 echo `blkid /dev/vdb1 | awk '{print $2}' | sed 's/\"//g'` /mnt ext4 defaults 0 0 >> /etc/fstab

12 cat /etc/fstab

13 mount /dev/vdb1 /mnt

14 df -h

15 fdisk -l

16 fdisk -u /dev/vdc

17 df -h

18 fdisk -u /dev/vdc

19 fdisk -l

20 mkfs -t ext4 /dev/vdc1

21 cp /etc/fstab /etc/fstab.bak2

22 echo `blkid /dev/vdc1 | awk '{print $2}' | sed 's/\"//g'` /mnt ext4 defaults 0 0 >> /etc/fstab

23 cat /etc/fstab

24 mount /dev/vdc1 /mnt

25 df -h

26 cat /etc/fstab

27 vi /etc/fstab

28 cat /etc/fstab

29 df -h

30 echo `blkid /dev/vdb1 | awk '{print $2}' | sed 's/\"//g'` /mnt1 ext4 defaults 0 0 >> /etc/fstab

31 mount /dev/vdb1 /mnt1

32 ls

33 cd ../

34 ls

35 mkdir mnt1

36 ls

37 mount /dev/vdb1 /mnt1

38 df –h

(yjgl)@ZJZX&zj189

mv /boot/initramfs-$(uname -r).img /boot/initramfs-$(uname -r).img.bak

dracut /boot/initramfs-$(uname -r).img $(uname -r)

Init 3

rpm -qa | grep kernel