# Centos 安装docker

|  |
| --- |
| 1、安装依赖  docker依赖于系统的一些必要的工具，可以提前安装。  yum install -y yum-utils device-mapper-persistent-data lvm2  2、添加软件源  yum-config-manager --add-repo http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo  3、安装docker-ce  yum clean all yum makecache fastyum -y install docker-ce  4、启动服务  通过systemctl启动服务  systemctl start docker  重启服务  Systemctl restart docker.service  5、查看安装版本  这样子就安装成功了，启动服务以后可以使用docker version查看一下当前的版本。  docker version   1. 运行hello-world   docker run hello-world |

Dubbo支持很多注册中心，官方推荐使用zookeeper

<http://dubbo.apache.org/zh-cn/docs/user/references/registry/zookeeper.html>

Zookeeper 官网

<https://zookeeper.apache.org/>

Zookeeper下载地址

<https://archive.apache.org/dist/zookeeper/>

# 2 docker安装镜像失败，使用国内数据源

问题：docker: Error response from daemon: Get https://registry-1.docker.io/v2/: net/http: request canceled

解决：

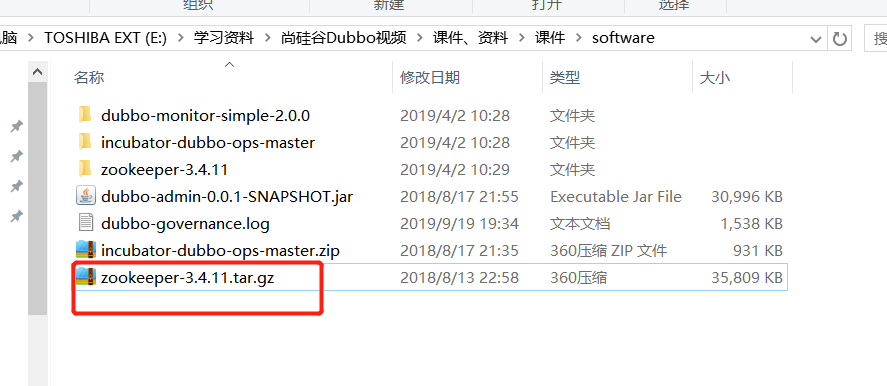
Docker默认使用国外官方源，下载速度较慢，可更换为国内的数据源

方案：

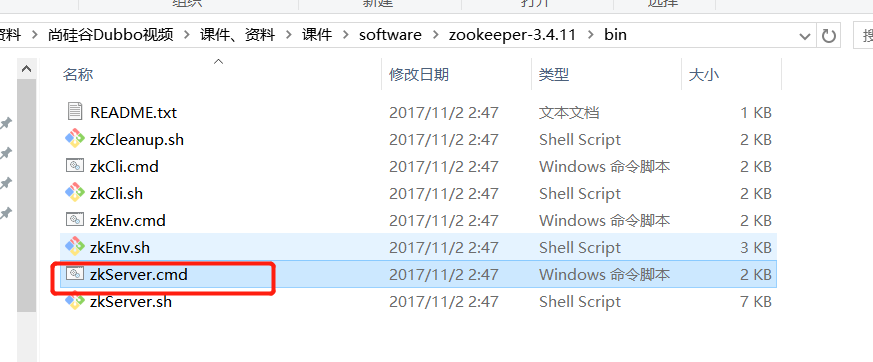
|  |
| --- |
| 修改或新增 /etc/docker/daemon.json  # vi /etc/docker/daemon.json  {  "registry-mirrors": ["http://hub-mirror.c.163.com"]  } |

示例

zookeeper-3.4.11.tar.gz



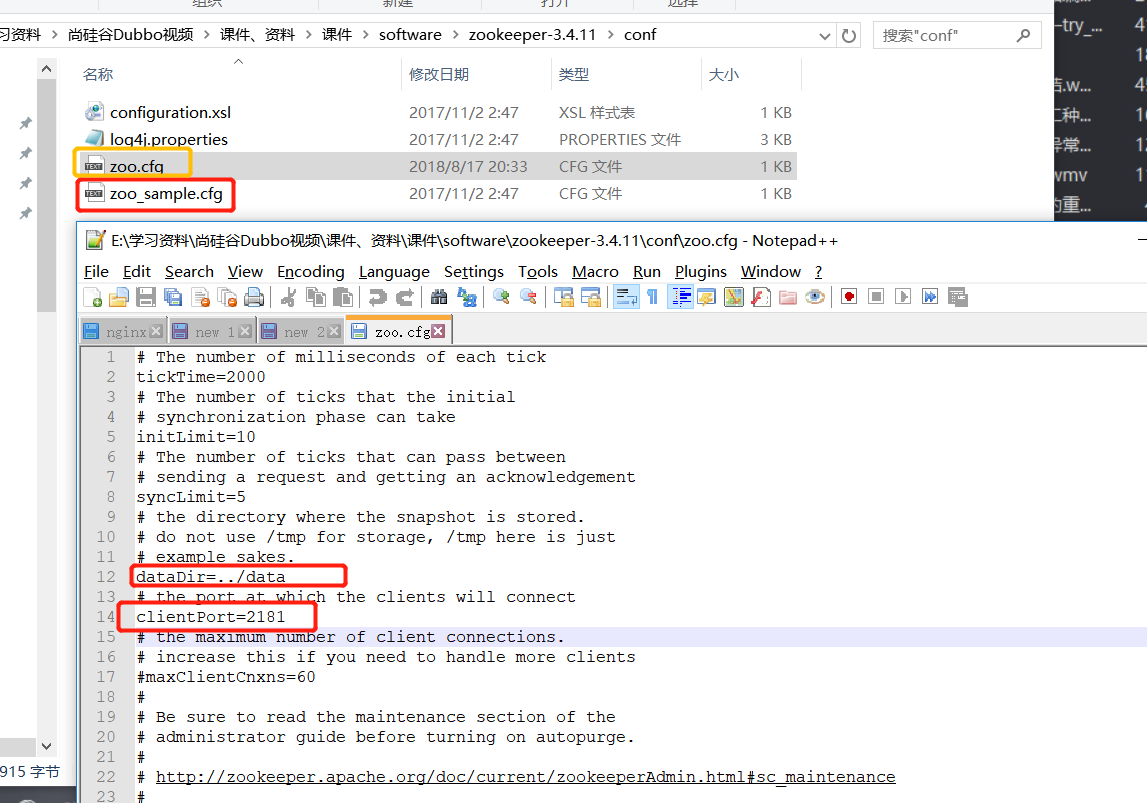
2.1 第一次启动bin目录下的zkServer.cmd会有问题，因为zookeeper的配置文件missing



解决方法：

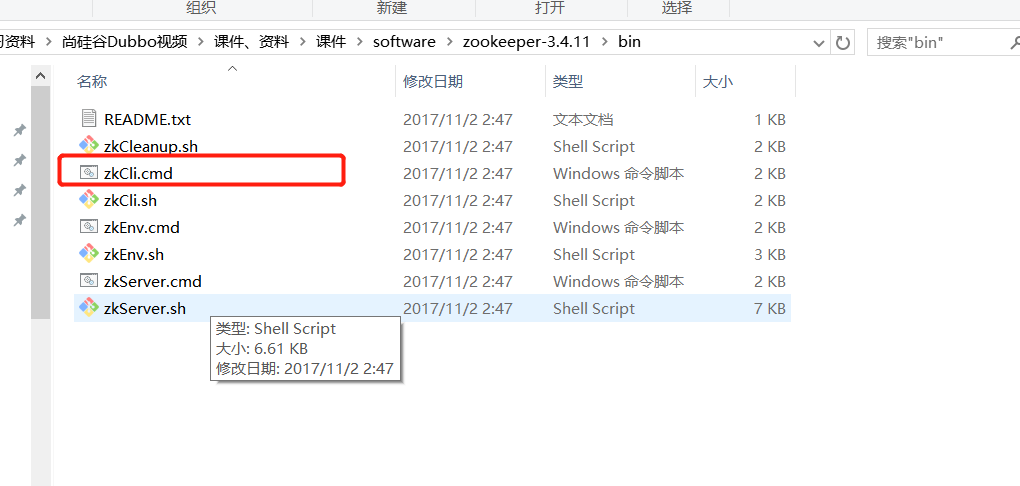
在conf目录下拷贝zoo\_sample.cfg 命名为 zoo.cfg

这个文件中主要有临时数据配置放在那里，以及zookeeper的端口号



2.2 保存以上配置后就可以进入bin目录启动zookeeper，即可启动成功

运行zkCli.cmd可以测试与zookeeper的连接



# Docker 运行zookeeper

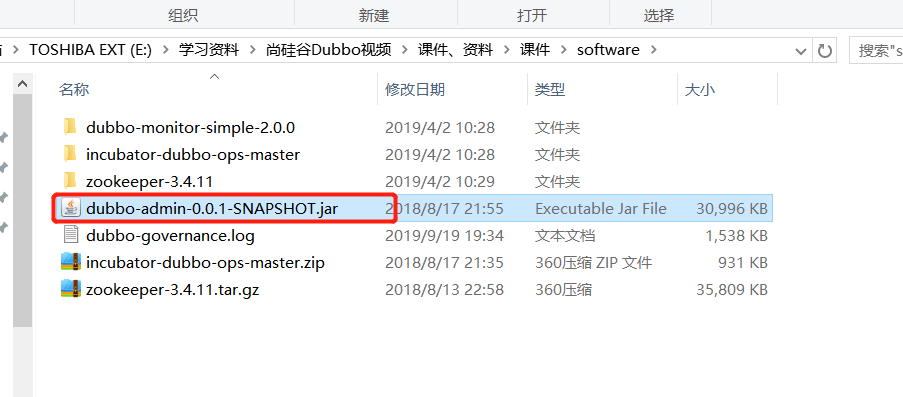
|  |
| --- |
| docker run -d --name zookeeper -p 2181:2181 -v /etc/localtime:/etc/localtime wurstmeister/zookeeper  查看docker日志  Docker logs -f zookeeper |

进入dubbo的github地址，下载dubbo-admin 工程，使用maven打包，dubbo2.6版本以后会打包成一个jar包执行jar包就行

Dubbo-admin地址

<https://github.com/apache/dubbo-admin>

打包后示例



Java -jar jar包 执行启动

默认配置监控端口号 7001

默认账号是root密码也是root

登录成功示例



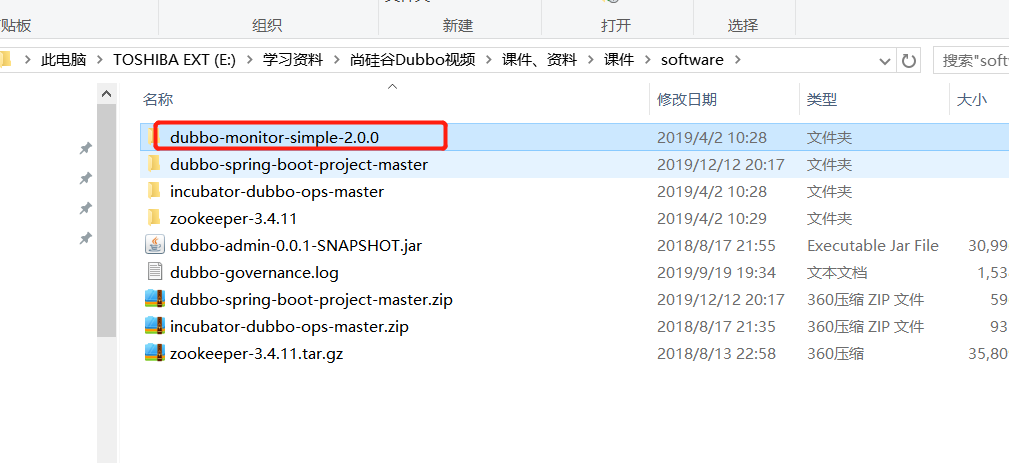
# 3 docker 安装kafak

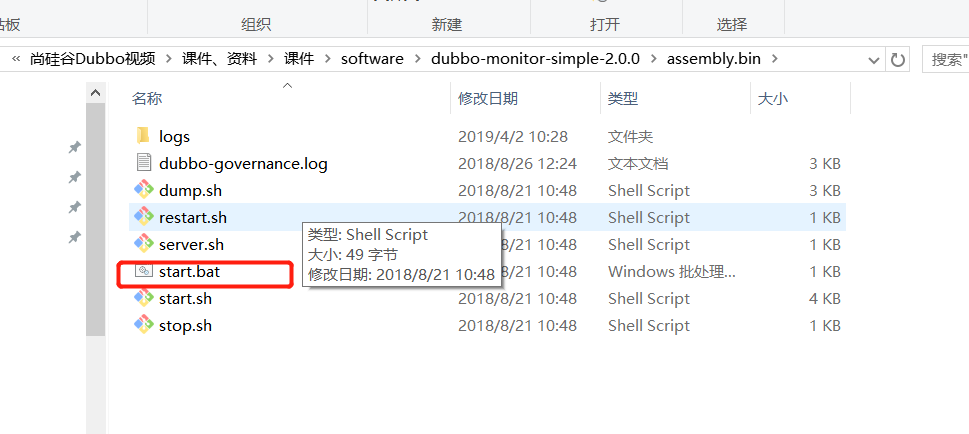
|  |
| --- |
| 安装镜像  docker pull wurstmeister/kafka  生成容器  docker run -d --name kafka -p 9092:9092 \  -e KAFKA\_BROKER\_ID=0 \  -e KAFKA\_ZOOKEEPER\_CONNECT=192.168.1.6:2181/kafka \  -e KAFKA\_ADVERTISED\_LISTENERS=PLAINTEXT://192.168.1.6:9092 \  -e KAFKA\_LISTENERS=PLAINTEXT://0.0.0.0:9092 \  -v /etc/localtime:/etc/localtime wurstmeister/kafka |

<http://dubbo.apache.org/zh-cn/docs/user/best-practice.html>

# 4 使用kafka

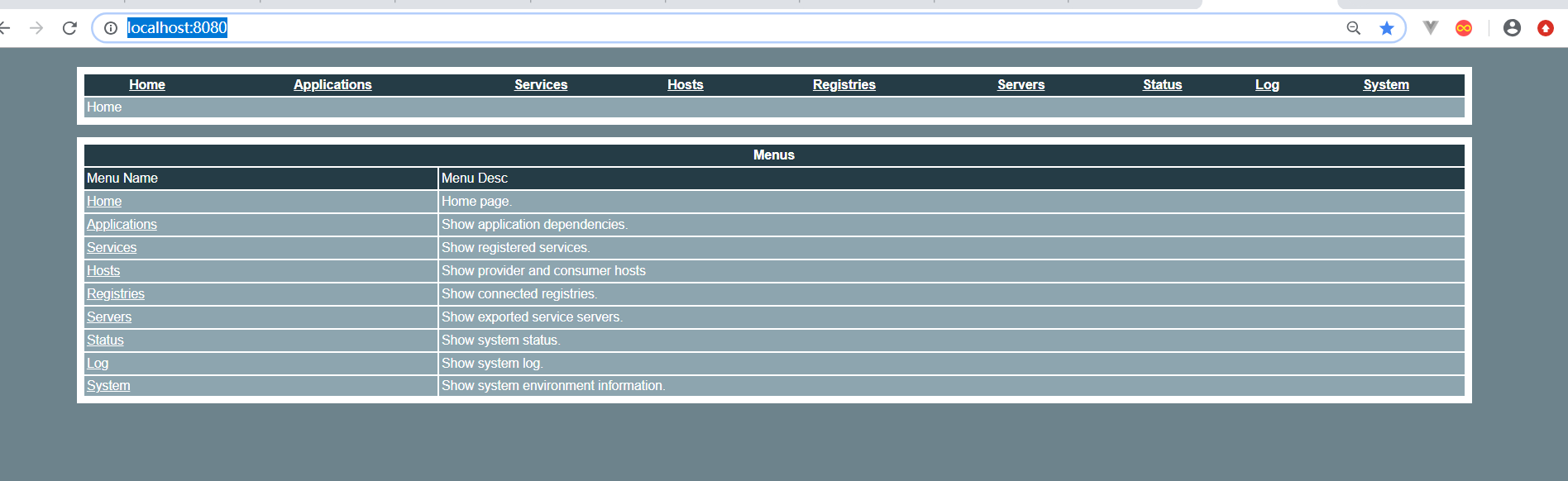
|  |
| --- |
| Docker exec -it kafka /bin/bash  cd /opt/kafka\_2.12-2.1.0/  创建topic  Bin/kafka-topics.sh --create --zookeeper zookeeper:2181 --replication-factor 1 --partitions 1 --topic test01  查看topic列表  Bin/kafka-topics.sh --list --zookeeper zookeeper:2181  运行生产这发送消息  Bin/kafka-console-producer.sh --broker-list localhost:9092 --topic test01  输入信息回车即发送  创建消费者接收消息  Bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic test01 --from-beginning |





示例页面

<http://localhost:8080/>



覆盖策略



# 5 docker安装使用redis

推荐使用镜像中心： daocloud.io/library/

拉取redis镜像

|  |
| --- |
| docker pull daocloud.io/library/redis:latest |

查看镜像

docker images

使用docker运行3个redis容器服务，分别使用6379 6380 6381三个端口，做集群使用

|  |
| --- |
| docker run –name redis-6379 –p 6379:6379 –d daocloud.io/library/redis  docker run –name redis-6380 –p 6380:6379 –d daocloud.io/library/redis  docker run –name redis-6381 –p 6381:6379 –d daocloud.io/library/redis |

查看redis内网IP

Docker inspect containerId

示例：

Redis-6379: 172.17.0.3:6379

Redis-6380: 172.17.0.4:6379

Redis-6381: 172.17.0.5:6379

|  |
| --- |
| 进入docker容器内部查看当前redis角色  Docker exec –it containerId /bin/bash  Redis-cli  Info replication    Slaveof host port //设置主从    哨兵模式详细配置待查询 |

# 6 docker安装使用mysql

|  |
| --- |
| 下载镜像  Docker pull mysql  运行启动  Docker run –p 3306:3306 –name mysql –e MYSQL\_ROOT\_PASSWORD=123456 –d mysql;  安装最新的mysql即mysql8，8之前的版本中加密规则是mysql\_native\_password，而在8之后加密规则是caching\_sha2\_password  解决办法有两种   1. 是更新客户端的驱动版本 2. 是修改mysql用户登录密码加密规则还原成mysql\_native\_password (一般采用这种)   在docker 中进入mysql  Docker exec –it mysql /bin/bash  Mysql –u root –p  连接登录后采用  ALTER USER ‘root’@’%’ IDENTIFIED WITH mysql\_native\_password BY ‘newpassword’; newpassword改为自己的密码  查看状态(可略过)  select host,user,plugin,authentication\_string from mysql.user;  刷新服务 (必须刷新服务,配置才生效)  flush privileges;  在客户端用navicate连接即可成功 |

该属性是消费者调服务者的方法，如果方法执行时间过长，超时未响应则抛出异常

<http://dubbo.apache.org/zh-cn/docs/user/configuration/xml.html>

从此有配置的优先级关系

# 7 docker安装使用Jenkins

|  |
| --- |
| docker pull jenkins/jenkins:lts;  docker images  docker inspect image\_id 查看jenkins版本  docker run -d --name jenkins -p 8081:8080 -v /home/jenkins:/home/jenkins jenkins/jenkins:lts;  查看jenkins的初始化密码  cat /var/jenkins\_home/secrets/initialAdminPassword |

调用超时或者什么情况可以设置重试次数，不包含第一次调用

在多服务的情况下，会均衡的往多个服务方重试调用接口

方法幂等 -》 设置重试次数

非幂等 -》 不可设置重试次数 新增

# 8 多版本 灰度发布 version

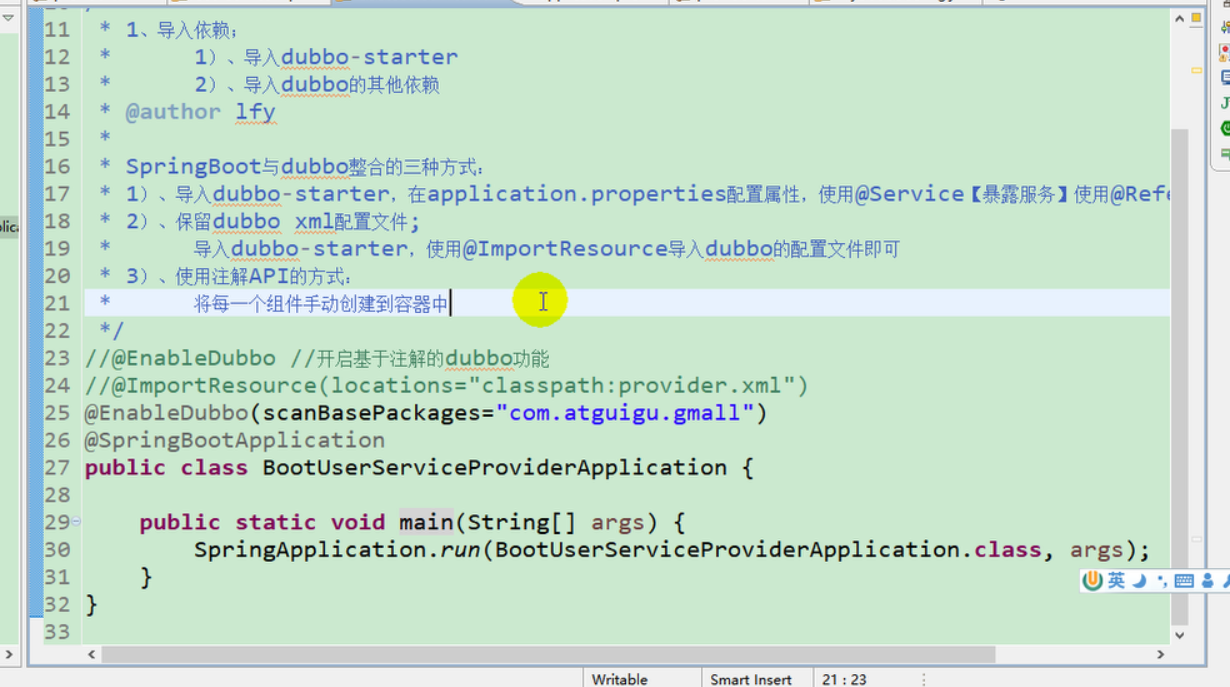
一般是在服务端上有改动产生新的接口或者方法，标记保留接口带有version

# 9 本地存根

利用本地存根来做一些调用前的判断之类

在消费者中的dubbo:reference引用中设置参数stub=”本地存根全类名”

# 10 springboot整合方式3 没办法精确到方法级别



# 11 服务降级

可在管理页面上操作服务屏蔽，暂时关闭服务，返回空

服务容错是在调用出错后返回空

## 实现动态调配服务器资源

# 12 集群容错

整合Hystrix

在服务端和消费者都引入依赖，开启容错，消费者可以设置出错后调用的方法

Spring-boot-starter-netflix-hystrix