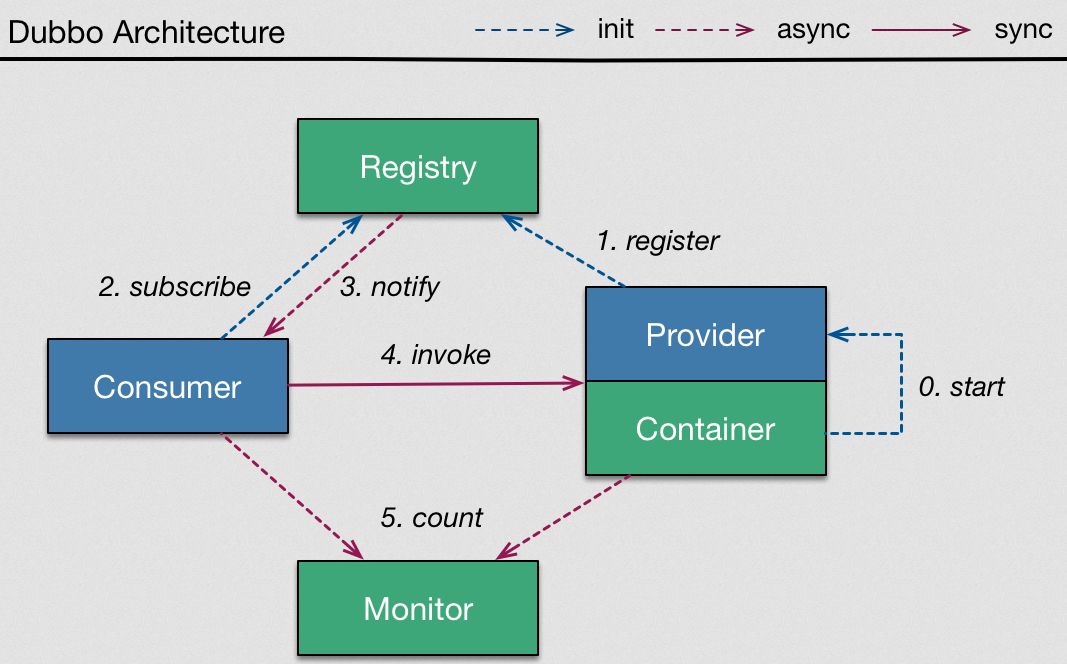
# dubbo+zookeeper整合

# dubbo+zookeeper相关概念

## dubbo

Dubbo是阿里巴巴公司开源的一个高性能优秀的服务框架，使得应用可通过高性能的 RPC 实现服务的输出和输入功能，可以和Spring框架无缝集成。

**Dubbo架构图:**



### （1）节点说明：

**Provider** 暴露服务方称之为“服务提供者”。

**Consumer**  调用[远程服务](https://baike.baidu.com/item/%E8%BF%9C%E7%A8%8B%E6%9C%8D%E5%8A%A1" \t "https://baike.baidu.com/item/Dubbo/_blank)方称之为“服务消费者”。

**Registry**  服务注册与发现的中心目录服务称之为“服务注册中心”。

**Monitor**  统计服务的调用次数和调用时间的日志服务称之为“服务监控中心”。

### 调用关系说明：

0. 服务容器负责启动，加载，运行服务提供者。

1. 服务提供者在启动时，向注册中心注册自己提供的服务。

2. 服务消费者在启动时，向注册中心订阅自己所需的服务。

3. 注册中心返回服务提供者地址列表给消费者，如果有变更，注册中心将基于长连接推送变更数据给消费者。

4. 服务消费者，从提供者地址列表中，基于软负载均衡算法，选一台提供者进行调用，如果调用失败，再选另一台调用。

5. 服务消费者和提供者，在内存中累计调用次数和调用时间，定时每分钟发送一次统计数据到监控中心。

服务注册中心: 保存信息的地方,Dubbo默认使用zookeeper作为注册中心.

## zookeeper

ZooKeeper是一个[分布式](https://baike.baidu.com/item/%E5%88%86%E5%B8%83%E5%BC%8F" \t "https://baike.baidu.com/item/zookeeper/_blank)的，开放源码的[分布式应用程序](https://baike.baidu.com/item/%E5%88%86%E5%B8%83%E5%BC%8F%E5%BA%94%E7%94%A8%E7%A8%8B%E5%BA%8F)协调服务，是[Google](https://baike.baidu.com/item/Google)的Chubby一个[开源](https://baike.baidu.com/item/%E5%BC%80%E6%BA%90" \t "https://baike.baidu.com/item/zookeeper/_blank)的实现，是Hadoop和Hbase的重要组件。它是一个为分布式应用提供一致性服务的软件，提供的功能包括：配置维护、域名服务、分布式同步、组服务等。

# dubbo安装使用

Dubbo采用全Spring配置方式，透明化接入应用，对应用没有任何API侵入，只需用Spring加载Dubbo的配置即可。

在提供方增加暴露服务配置<dubbo:service>

在消费方增加引用服务配置<dubbo:reference>

## 1、发布服务（提供方）：

<!-- 和本地服务一样实现远程服务 -->

<bean id="xxxService" class="com.xxx.XxxServiceImpl" />

<!-- 增加暴露远程服务配置 -->

<dubbo:service interface="com.xxx.XxxService" ref="xxxService" />

|  |
| --- |
| <!-- 自动扫描 -->  <context:component-scan base-package=*"com.yxw.manager.service"*></context:component-scan>  <!-- 使用dubbo发布服务 -->  <!-- 提供方应用信息，用于计算依赖关系 -->  <dubbo:application name=*"yxw-manager-service"* />  <!-- <dubbo:registry protocol="zookeeper" address="127.0.0.1:2181" /> -->  <dubbo:registry protocol=*"zookeeper"* address=*"192.168.136.133:2181"* />  <!-- 用dubbo协议在20880端口暴露服务-->  <dubbo:protocol name=*"dubbo"* port=*"20880"* />  <!-- 声明需要暴露的服务接口 -->  <dubbo:service interface=*"com.yxw.manager.service.CategoryService"* ref=*"categoryServiceImp"* timeout=*"1000000"* />  <dubbo:service interface=*"com.yxw.manager.service.ProductService"* ref=*"productServiceImp"* timeout=*"1000000"* />  <dubbo:service interface=*"com.yxw.manager.service.utils.RedisUtils"* ref=*"redisUtilsImp"* timeout=*"1000000"* /> |

## 2、调用服务（消费方）：

<!-- 增加引用远程服务配置 -->

<dubbo:reference id="xxxService" interface="com.xxx.XxxService" />

<!-- 和本地服务一样使用远程服务 -->

<bean id="xxxAction" class="com.xxx.XxxAction">

<property name="xxxService" ref="xxxService" />

</bean>

|  |
| --- |
| <!-- 消费方应用名，用于计算依赖关系，不是匹配条件，不要与提供方一样 -->  <dubbo:application name=*"yxw-manager-web"* />  <!-- 使用zookeeper注册中心暴露服务地址 -->  <!-- <dubbo:registry protocol="zookeeper" address="127.0.0.1:2181" /> -->  <dubbo:registry protocol=*"zookeeper"* address=*"192.168.136.133:2181"* />  <!-- 用dubbo协议在20880端口暴露服务-->  <dubbo:protocol name=*"dubbo"* port=*"20880"* />  <!-- 生成远程服务代理，可以像使用本地bean一样使用Service -->  <dubbo:reference id=*"testService"* interface=*"com.yxw.manager.service.TestService"* />  <dubbo:reference id=*"categoryService"* interface=*"com.yxw.manager.service.CategoryService"* />  <dubbo:reference id=*"productService"* interface=*"com.yxw.manager.service.ProductService"* />  <dubbo:reference id=*"redisUtils"* interface=*"com.yxw.manager.service.utils.RedisUtils"* /> |

# zookeeper安装使用

## 安装zookeeper(Linux)

前提：必须配置好jdk环境,因为zookeeper是使用java编写的.

# mkdir /usr/local/src/zookeeper

# tar -zxvf zookeeper-3.4.6.tar.gz -C /usr/local/src/zookeeper/

# cd /usr/local/src/zookeeper/zookeeper-3.4.6

# mkdir data

# cd /usr/local/src/zookeeper/zookeeper-3.4.6/conf

# mv zoo\_sample.cfg zoo.cfg

# vim zoo.cfg

将dataDir修改为/usr/local/src/zookeeper/zookeeper-3.4.6/conf/data/

## 启动zookeeper(Linux)

# cd /usr/local/src/zookeeper/zookeeper-3.4.6/bin

# ./zkServer.sh start

{start|start-foreground|stop|restart|status|upgrade|print-cmd}

## 开放端口号为2181的端口（Linux）

停止防火墙:

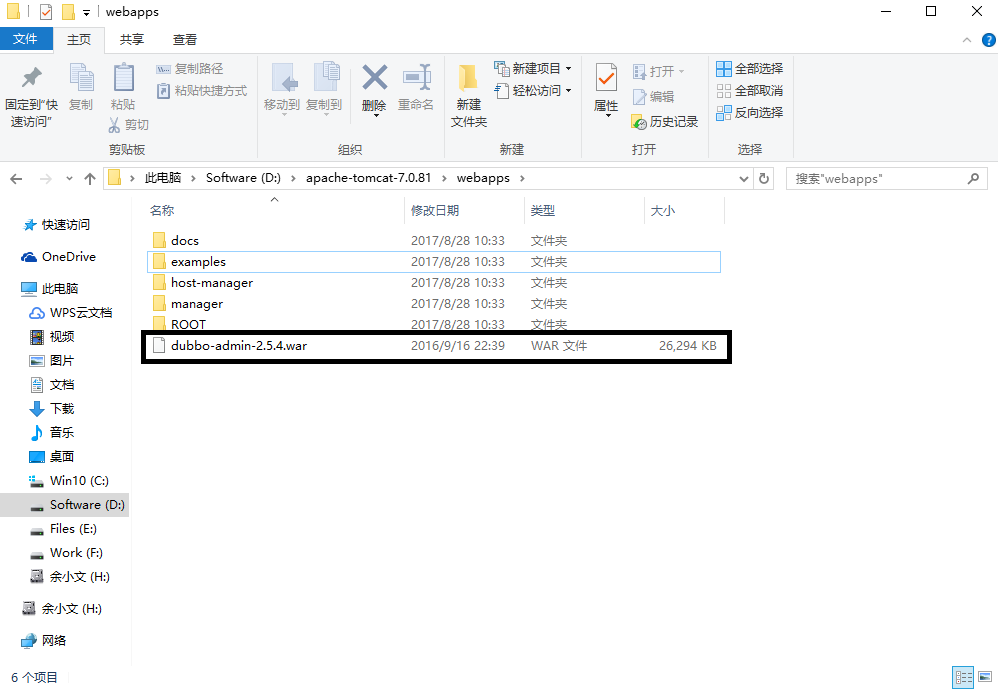
#service iptables stop

把防火墙设置为开启不启动:

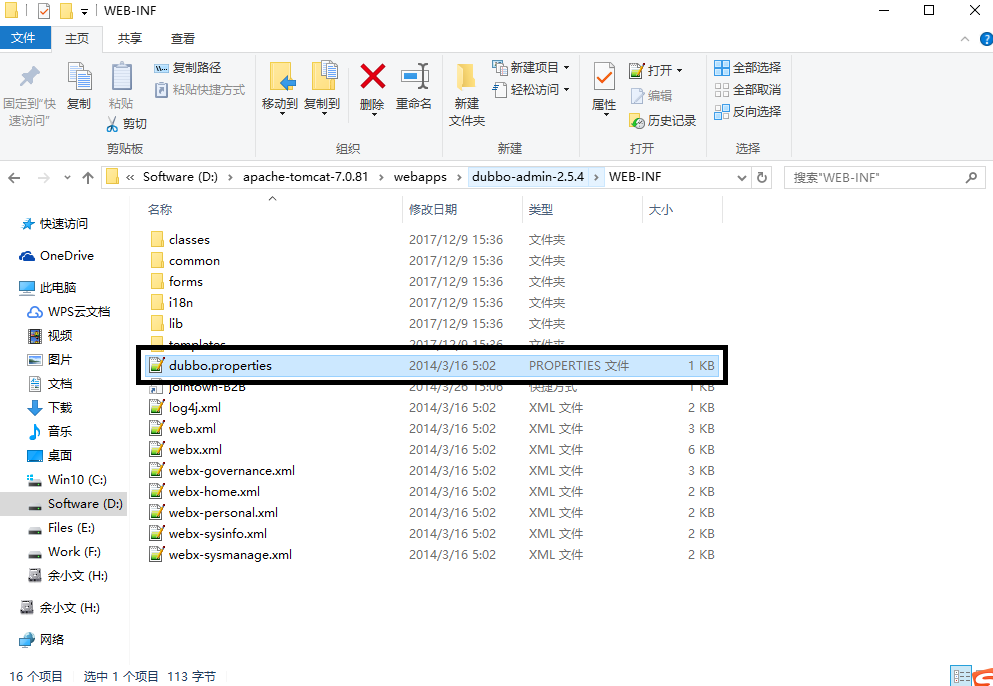
# chkconfig iptables off

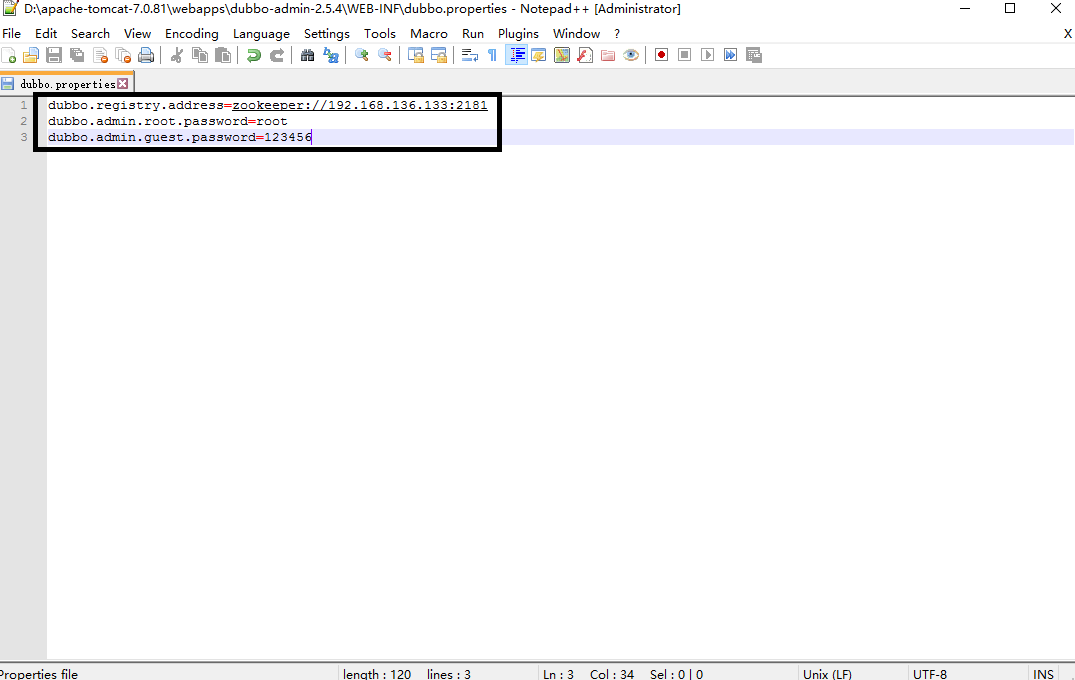
# dubbo管理后台

将dubbo-admin-2.5.4.war复制到Tomcat的webapps目录下，下次启动Tomact时会自动部署



部署成功后，修改网站配置





配置成功后重启Tomcat,访问localhost:8080/dubbo-admin-2.5.4/

账号:root

密码:root

