## XX务部署文档

目录

[**1. 网络拓扑图 3**](#_Toc21380)

[**2. 服务组件安装 3**](#_Toc27642)

[2.1 JDK安装 3](#_Toc10449)

[2.2 Zookeeper集群安装 4](#_Toc32043)

[**3. 卡平台服务部署 6**](#_Toc25307)

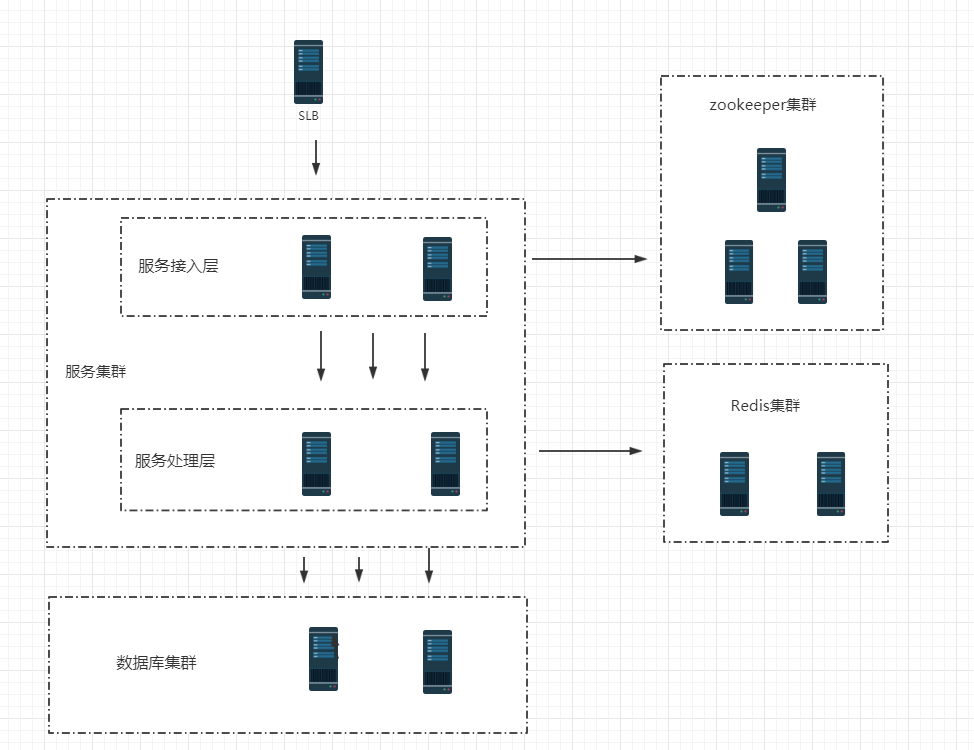
[**4.业务服务配置文件详解 6**](#_Toc29538)

[4.1 cardservice-app配置文件详解 6](#_Toc25883)

[4.2 cardservivce-api配置文件详解 8](#_Toc32757)

[4.3 posservice-api配置文件详解 9](#_Toc4485)

### 网络拓扑图



### 服务组件安装

#### JDK安装

1. 在/usr/目录下创建java目录

[root@localhost ~]# mkdir /opt/java

[root@localhost ~]# cd /opt/java

1. 下载jdk,然后解压

[root@localhost java]#

curl -O [http://download](http://download/).[Oracle](http://www.linuxidc.com/topicnews.aspx?tid=12).com/otn-pub/java/jdk/7u79-b15/jdk-7u79-linux-x64.tar.gz

[root@localhost java]# tar -zxvf jdk-7u79-linux-x64.tar.gz

1. 设置环境变量

[root@localhost java]# vi /etc/profile

在profile中添加如下内容:

#set java environment

JAVA\_HOME=/opt/java/jdk1.7.0\_79

JRE\_HOME=/opt/java/jdk1.7.0\_79/jre

CLASS\_PATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar:$JRE\_HOME/lib

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

export JAVA\_HOME JRE\_HOME CLASS\_PATH PATH

让修改生效:

[root@localhost java]# source /etc/profile

1. 验证JDK有效性

[root@localhost java]# java -version

java version "1.7.0\_79"

Java(TM) SE Runtime Environment (build 1.7.0\_79-b15)

Java HotSpot(TM) 64-Bit Server VM (build 24.79-b02, mixed mode)

代码分API和APP，API是接入，APP是业务处理；API使用的框架是DubboX，APP使用的Rpc框架是Dubbo；

#### Zookeeper集群安装

zookeeper安装在三个节点上，10.0.103.174，10.0.103.175，10.0.103.176；

1. 下载：下载zookeeper安装包： zookeeper-3.4.8.tar.gz
2. 上传：上传安装包到三个节点上；
3. 解压：在三个节点上分别执行命令：

tar -zxvf zookeeper-3.4.8.tar.gz -C /usr/local/

1. 配置环境变量： 在三个节点都上配置环境变量

vi /etc/profile

追加  
 export ZOOKEEPER\_HOME=/usr/local/zookeeper-3.4.8

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

使环境变量立即生效

source /etc/profile

1. 修改配置文件： 修改zookeeper中conf目录下的zoo\_sample.cfg为zoo.cfg

修改三个zookeeper节点中的zoo.cfg文件，修改dataDir，添加server.0、server.1、server.2

在配置文件中追加

server.1=10.0.103.174:2888:3888

server.2=10.0.103.175:2888:3888

server.3=10.0.103.176:2888:3888

zoo.cofg配置文件

# The number of milliseconds of each tick

tickTime=2000

# The number of ticks that the initial

# synchronization phase can take

initLimit=10

# The number of ticks that can pass between

# sending a request and getting an acknowledgement

syncLimit=5

# the directory where the snapshot is stored.

# do not use /tmp for storage, /tmp here is just

# example sakes.

dataDir=/usr/local/zookeeper-3.4.8/data

dataLogDir=/usr/local/zookeeper-3.4.8/logs

# the port at which the clients will connect

clientPort=2181

# the maximum number of client connections.

# increase this if you need to handle more clients

#maxClientCnxns=60

#

# Be sure to read the maintenance section of the

# administrator guide before turning on autopurge.

#

# http://zookeeper.apache.org/doc/current/zookeeperAdmin.html#sc\_maintenance

#

# The number of snapshots to retain in dataDir

#autopurge.snapRetainCount=3

# Purge task interval in hours

# Set to "0" to disable auto purge feature

#autopurge.purgeInterval=1

server.1=10.0.103.174:2888:3888

server.2=10.0.103.175:2888:3888

server.3=10.0.103.176:2888:3888

1. 在zookeeper目录下，创建data目录。在3个zookeeper节点中data目录下分别创建myid文件，并分别添加内容1、2、3；
2. 三个节点分别启动zookeeper：/usr/local/zookeeper-3.4.8/bin/zkServer.sh start

## 3. 卡平台服务部署

3.1 服务分为卡平台核心服务cardservice-app，手机端接入服务cardservice-api和刷卡设备接入服务posservice-api；

3.2 服务部署（以cardservice-app为例）

(1)打包：使用maven命令package打包，得到压缩包cardservice-app-0.0.3-SNAPSHOT-assembly.tar.gz；

(2) 上传：将压缩包cardservice-app-0.0.3-SNAPSHOT-assembly.tar.gz上传到服务器执行目录下；

(3) 解压：解压服务包，tar -xzvf cardservice-app-0.0.3-SNAPSHOT-assembly.tar.gz, 得到

服务文件夹cardservice-app-0.0.3-SNAPSHOT;

1. 修改配置文件：进入到服务文件夹cardservice-app-0.0.3-SNAPSHOT下的META-INF，

把conf-ali.properties里面的配置拷贝到conf.properties中；

1. 启动服务：进入到服务文件夹cardservice-app-0.0.3-SNAPSHOT下，执行命令bin/start.sh启动服务；

## 4.业务服务配置文件详解

#### 4.1 cardservice-app配置文件详解

#dubbo

#应用名称

dubbo.application.name=cardservice-app

#注册中心协议

dubbo.registry.protocol=zookeeper

#注册中心地址

dubbo.registry.address=127.0.0.1:2181

#是否将服务注册到注册中心

dubbo.registry=true

#使用文件缓存注册中心地址列表及服务提供者列表，应用重启时将基于此文件恢复

dubbo.registry.file=dubbo-cardservice-app.cache

#协议名称

dubbo.protocol.name=dubbo

#服务端口

dubbo.protocol.port=36601

#线程池类型，可选：fixed/cached

dubbo.protocol.threadpool=cached

#服务线程池大小(固定大小)

dubbo.protocol.threads=300

#协议的服务端和客户端实现类型，比如：dubbo协议的mina,netty等，可以分拆为server和client配置

dubbo.protocol.transporter=netty

#远程服务调用超时时间(毫秒)

dubbo.method.timeout=3000

#配置中心地址

config.zookeeper.address=127.0.0.1:2181

#数据库连接池最大活跃连接

database.maxActive=5

#数据库连接池最小空闲连接

database.minIdle=1

#数据库连接池初始化连接数量

database.initialSize=2

#redis 参数配置

redis.session.maxInactiveInterval=10080

redis.session.expirationUpdateInterval=300

redis.session.saveOnChange=false

redis.session.alwaysSaveAfterRequest=false

redis.session.initGetAllAttribute=false

redis.session.host=172.16.6.19

redis.session.port=6379

redis.session.database=1

redis.session.password=chop!@#sticks

redis.session.timeout=300

redis.session.sentinels=

redis.session.sentinelMaster=

#notify settings

#MQ topic

card.recharge.notify.topic=Vpai

#MQ消息发布者id

card.recharge.notify.producer=PID\_Vpai

#MQ消息接受者id

card.recharge.notify.consumer=CID\_Vpai

#ali MQ AccessKey

aliyun.mq.AccessKey=LTAI2Hfd232ZcbtB

#ali MQ SecretKey

aliyun.mq.SecretKey=P5jm0vRECN4XWiOl69eFNpCb0eLSNb

#ali MQ ONAddr

aliyun.mq.ONSAddr=http://onsaddr-internet.aliyun.com/rocketmq/nsaddr4client-internet

#database

#数据库驱动名称

database.driver=org.postgresql.Driver

#数据库连接地址

database.url=jdbc:postgresql://127.0.0.1:5432/bus\_card

#数据库用户名

database.username=postgres

#数据库用户名对应的密码

database.password=123456

#分配的appid

database.appid=appid\_test

#数据源配置

database.appid1=appid\_test

database.appid2=BxXyK4igVyBFHOm

database.appid3=dgxpiq6b48i71ur8nh1

#channel settings

#充值结果，刷卡消费结果通知目标的请求地址

channel.callback.url=http://112.243.253.73:8081/citizencard-web

#充值结果，刷卡消费结果通知目标的签名key

channel.callback.signkey=21232f297a57a5a743894a0e4a801fc3

#定时器的appid

timer.appId = BxXyK4igVyBFHOm

#定时器的cron表达式

timer.daily.cron=0/30 \* \* \* \* ?

#### 4.2 cardservivce-api配置文件详解

#协议名称

rest.protocol.name=rest

#rest服务端口

rest.protocol.port=6601

#服务线程池大小(固定大小)

rest.protocol.threads=500

#服务提供方最大可接受连接数

rest.protocol.accepts=500

#服务请求根路径

rest.protocol.contextpath=cardservice-api

#协议的服务器端实现类型，比如：dubbo协议的mina,netty等，http协议的jetty,servlet等

rest.protocol.server=tomcat

#协议序列化方式，当协议支持多种序列化方式时使用，比如：dubbo协议的dubbo,hessian2,java,compactedjava，以及http协议的json等

rest.protocol.serialization=fst

#dubbo

#应用名称

dubbo.application.name=cardservice-api

#dubbo服务端口

dubbo.protocol.port=36601

#注册中心协议

dubbo.registry.protocol=zookeeper

#注册中心地址

dubbo.registry.address=127.0.0.1:2181

#使用文件缓存注册中心地址列表及服务提供者列表，应用重启时将基于此文件恢复

dubbo.registry.file=dubbo-cardservice-api.cache

#远程服务调用超时时间(毫秒)

dubbo.method.timeout=3000

#配置中心地址

config.zookeeper.address=127.0.0.1:2181

#### 4.3 posservice-api配置文件详解

#协议名称

rest.protocol.name=rest

#rest服务端口

rest.protocol.port=6602

#服务线程池大小(固定大小)

rest.protocol.threads=500

#服务提供方最大可接受连接数

rest.protocol.accepts=500

#服务请求根路径

rest.protocol.contextpath=posservice-api

#协议的服务器端实现类型，比如：dubbo协议的mina,netty等，http协议的jetty,servlet等

rest.protocol.server=tomcat

#协议序列化方式，当协议支持多种序列化方式时使用，比如：dubbo协议的dubbo,hessian2,java,compactedjava，以及http协议的json等

rest.protocol.serialization=fst

#dubbo

#应用名称

dubbo.application.name=posservice-api

#dubbo服务端口

dubbo.protocol.port=36602

#注册中心协议

dubbo.registry.protocol=zookeeper

#注册中心地址

dubbo.registry.address=127.0.0.1:2181

#使用文件缓存注册中心地址列表及服务提供者列表，应用重启时将基于此文件恢复

dubbo.registry.file=dubbo-posservice-api.cache

#远程服务调用超时时间(毫秒)

dubbo.method.timeout=3000

#配置中心地址

config.zookeeper.address=127.0.0.1:2181