高级篇

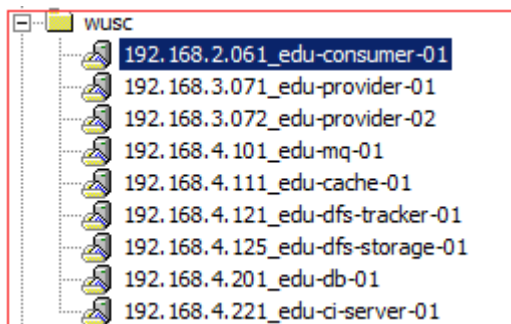**简易版支付系统部署（单节点）**

一、前期准备

1、MySQL 数据库的安装：MySQL-5.6.22，自行安装

2、Dubbo 视频教程--基础篇--第 03 节--ZooKeeper 注册中心安装

3、Dubbo 视频教程--基础篇--第 06 节--Dubbo 管理控制台的安装

4、Dubbo 视频教程--基础篇--第 10 节--Dubbo 监控中心的介绍与简易监控中心的安装

5、持续集成管理平台（SVN、Nexus、Maven、Hudson）的安装：

　　Dubbo 视频教程--基础篇--第 11 节至 18 节

6、Dubbo 视频教程--高级篇--第 21 节--ActiveMQ 的安装与使用

7、Dubbo 视频教程--高级篇--第 22 节--Redis 的安装与使用

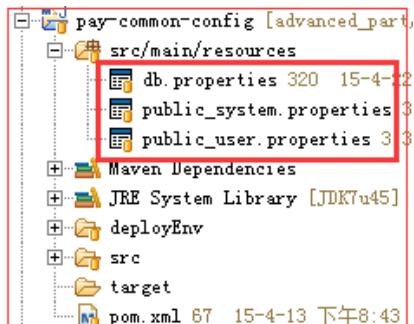8、Dubbo 视频教程--高级篇--第 23 节--FastDFS 分布式文件系统的安装与使用

二、对部署环境进行规划

创建数据库

数据库编码为：UTF-8

数据库引擎为：InnoDB

导入"基于 Dubbo 的分布式系统架构视频教程--简易版支付系统源码.rar"中的

"数据库--edu_simple_pay.rar"中的"edu_simple_pay.sql"

调整公共配置文件

应用部署前期准备

1、common 工程构建、发布到 Maven 私有库

尤其注意 pay-common-config 工程的构建（配置文件修改后需要重新构建，引用处也要重新构建）

2、facade 工程构建、发布到 Maven 私有库

部署服务

1、规划好服务部署目录，准备好服务管理脚本
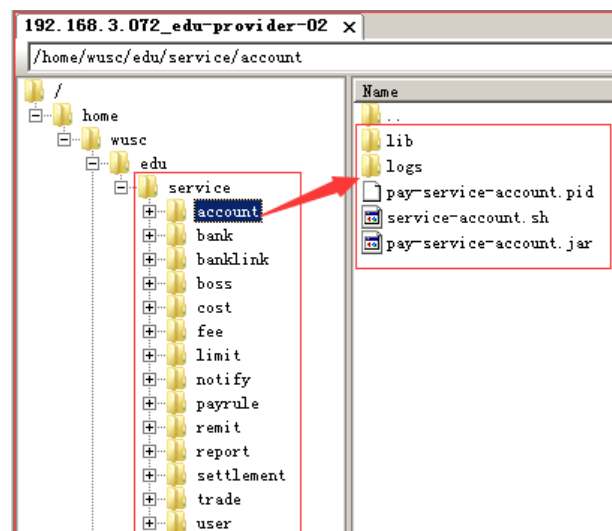
/home/wusc/edu/service/account/service-account.sh

```sh
#!/bin/sh
## java env
export JAVA_HOME=/usr/local/java/jdk1.7.0_72
export JRE_HOME=$JAVA_HOME/jre
## you just need to change this param name
APP_NAME=account
SERVICE_DIR=/home/wusc/edu/service/$APP_NAME
SERVICE_NAME=pay-service-$APP_NAME
JAR_NAME=$SERVICE_NAME\.jar
PID=$SERVICE_NAME\.pid
cd $SERVICE_DIR
case "$1" in
    start)
        nohup $JRE_HOME/bin/java -Xms128m -Xmx512m -jar $JAR_NAME >/dev/null 2>&1 &
        echo $! > $SERVICE_DIR/$PID
        echo "=== start $SERVICE_NAME"
        ;;
    stop)
        kill `cat $SERVICE_DIR/$PID`
        rm -rf $SERVICE_DIR/$PID
        echo "=== stop $SERVICE_NAME"
        sleep 5
        P_ID=`ps -ef | grep -w "$SERVICE_NAME" | grep -v "grep" | awk '{print $2}'`
        if [ "$P_ID" == "" ]; then
            echo "=== $SERVICE_NAME process not exists or stop success"
        else
            echo "=== $SERVICE_NAME process pid is:$P_ID"
            echo "=== begin kill $SERVICE_NAME process, pid is:$P_ID"
```

```
        kill -9 $P_ID
    fi
    ;;

restart)
    $0 stop
    sleep 2
    $0 start
    echo "=== restart $SERVICE_NAME"
    ;;

*)
    ## restart
    $0 stop
    sleep 2
    $0 start
    ;;

esac
exit 0
```
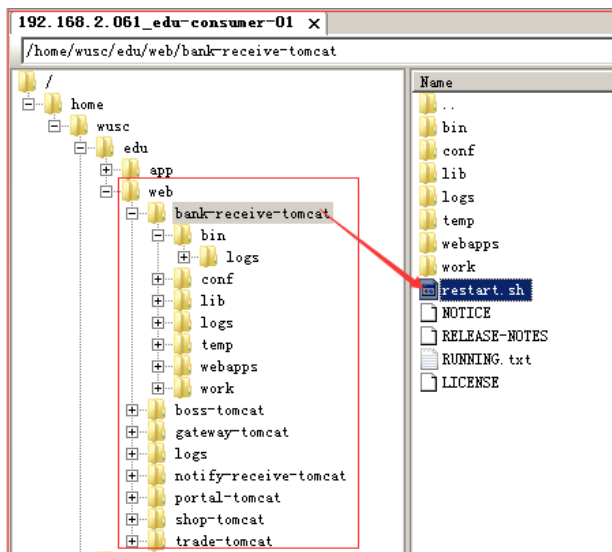
2、使用 Hudson 来自动化部署服务

3、通过 Dubbo 管控台检查各服务是否都部署成功

部署 Web 应用

1、规划好 Web 应用部署目录、端口、脚本

bank-receive-tomcat	8081
boss-tomcat	8082
gateway-tomcat	8083
notify-receive-tomcat	8084
portal-tomcat	8085
shop-tomcat	8086
trade-tomcat	8087

/home/wusc/edu/web/bank-receive-tomcat/restart.sh

```
## java env
export JAVA_HOME=/usr/local/java/jdk1.7.0_72
export JRE_HOME=$JAVA_HOME/jre
## restart tomcat
current_dir=$(cd `dirname $0`; pwd)
echo "=== current_dir is:$current_dir"
$current_dir/bin/shutdown.sh
sleep 3
rm -rf $current_dir/webapps/*/
sleep 2
$current_dir/bin/startup.sh
```

部署 APP
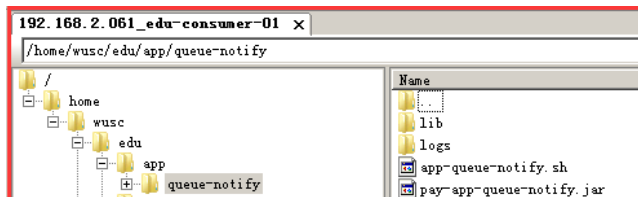
/home/wusc/edu/app/queue-notify/app-queue-notify.sh

```sh
#!/bin/sh
## java env
export JAVA_HOME=/usr/local/java/jdk1.7.0_72
export JRE_HOME=$JAVA_HOME/jre
## you only need to change next two parameters value
APP_DIR=/home/wusc/edu/app/queue-notify
APP_NAME=pay-app-queue-notify
JAR_NAME=$APP_NAME\.jar
cd $APP_DIR
## check app process weather exists
process=`ps aux | grep -w "$APP_NAME" | grep -v grep`
if [ "$process" == "" ]; then
    echo "=== $APP_NAME process not exists"
else
    echo "=== $APP_NAME process exists"
    echo "=== $APP_NAME process is : $process"
    ## get PID by process name
    P_ID=`ps -ef | grep -w "$APP_NAME" | grep -v "grep" | awk '{print $2}'`
    echo "=== $APP_NAME process PID is:$P_ID"
    echo "=== begin kill $APP_NAME process"
    kill $P_ID
    sleep 3
    P_ID=`ps -ef | grep -w "$APP_NAME" | grep -v "grep" | awk '{print $2}'`
    if [ "$P_ID" == "" ]; then
        echo "=== $APP_NAME process stop success"
    else
        echo "=== $APP_NAME process kill failed, PID is:$P_ID"
        echo "=== begin kill -9 $APP_NAME process, PID is:$P_ID"
        sleep 3
        kill -9 $P_ID
    fi
fi
sleep 2
echo "=== begin start $APP_NAME"
$JRE_HOME/bin/java -Xms128m -Xmx512m -jar $APP_DIR/$JAR_NAME >/dev/null 2>&1 &
```
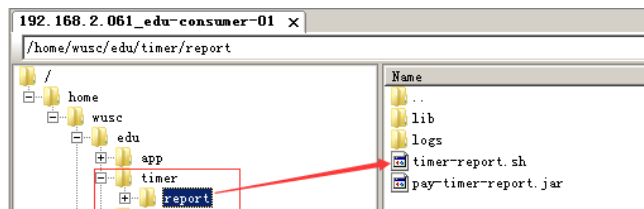
部署定时任务

定时任务的部署与调用分离

/home/wusc/edu/timer/report/timer-report.sh

```sh
#!/bin/sh
## java env
export JAVA_HOME=/usr/local/java/jdk1.7.0_72
export JRE_HOME=$JAVA_HOME/jre
## you only need to chage next two line value
SERVICE_DIR=/home/wusc/edu/timer/report
APP_NAME=pay-timer-report
JAR_NAME=$APP_NAME\.jar
sleep 1
echo "=== invoke task, please wait"
$JRE_HOME/bin/java -jar $SERVICE_DIR/$JAR_NAME >/dev/null 2>&1 &
## until process stop to print log
while true
do
    process=`ps aux | grep $APP_NAME | grep -v grep`;
    if [ "$process" == "" ]; then
        sleep 1;
        echo "=== task complete";
        sleep 3;
        break;
    else
        echo "=== process is running, please wait";
        sleep 10;
        continue;
    fi
done
```