## 环境搭建

防火墙的处理：<https://blog.csdn.net/qq_18300109/article/details/79172019>

### 安装JDK

* 1. yum 安装

yum install java-1.8.0-openjdk\* -y

默认安装位置：/user/lib/jvm

* 1. 配置环境变量

在/etxc/profile 文件末尾添加

export JAVA\_HOME=/usr/lib/jvm/java

export CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

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

### 2. mysql安装（安装路径：/home/work/package/mysql）

2.1 下载并安装MySQL官方的 Yum Repositor

wget -i -c <http://dev.mysql.com/get/mysql57-community-release-el7-10.noarch.rpm>

2.2 yum安装

yum -y install mysql57-community-release-el7-10.noarch.rpm

2.3 安装mysql服务器

yum -y install mysql-community-server

2.4 启动

systemctl start mysqld.service

2.5 查看mysql状态

systemctl status mysqld.service

2.6 查看初始化密码

grep "password" /var/log/mysqld.log

2.7 用初始化密码进行登录

mysql -uroot -p

输入2.6 得到的初始化密码

2.8 修改初始化密码（这里有个问题，新密码设置的时候如果设置的过于简单会报错）

select user,host from user;

ALTER USER 'root'@'localhost' IDENTIFIED BY 'new password';

ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password BY '123';

<https://www.cnblogs.com/ivictor/p/5142809.html>

ERROR **1819** (HY000): Your password does not satisfy the current policy requirements

这个其实与validate\_password\_policy的值有关。

validate\_password\_policy有以下取值：

| **Policy** | **Tests Performed** |
| --- | --- |
| 0 or LOW | Length |
| 1 or MEDIUM | Length; numeric, lowercase/uppercase, and special characters |
| 2 or STRONG | Length; numeric, lowercase/uppercase, and special characters; dictionary file |

默认是1，即MEDIUM，所以刚开始设置的密码必须符合长度，且必须含有数字，小写或大写字母，特殊字符。

有时候，只是为了自己测试，不想密码设置得那么复杂，譬如说，我只想设置root的密码为123456。

必须修改两个全局参数：

首先，修改validate\_password\_policy参数的值

set global validate\_password\_policy=**0**;

2.9 但此时还有一个问题，就是因为安装了Yum Repository，以后每次yum操作都会自动更新，需要把这个卸载掉：

yum -y remove mysql57-community-release-el7-10.noarch

2.10 设置外网访问

2.10.1 登录

mysql -uroot -pChangeme\_123

use mysql

2.10.2

查看用户

select user,host from user;

2.10.3设置host为%

update user set host='127.0.0.1' where user = 'root';

2.10.4授权用户具体权限

grant all privileges on \*.\* to 'root'@'%' identified by legaoyun@2018with grant option;

flush privileges;

create user 'zl\_'@'172.16.17.%' identified by 'legaoyun@2018'

创建只能在局域网连接Mysql的用户

create user 'zl\_db'@'172.16.17.%' identified by 'legaoyun@2018'

授予Mysql用户数据库访问权限

grant all privileges on crec\_dmp\_db.\* to 'zl\_db'@'172.16.17.%' identified by 'legaoyun@2018';

flush privileges;



### MongoDb安装

常用问题解决方案: <https://www.cnblogs.com/dunitian/p/6274437.html>

用户管理: <https://blog.csdn.net/jianleking/article/details/79715097>

3.1 下载安装包

curl -O <https://fastdl.mongodb.org/linux/mongodb-linux-x86_64-3.2.12.tgz>

3.2 解压

tar -zxvf mongodb-linux-x86\_64-3.2.12.tgz

3.3 移动到指定位置

mv mongodb-linux-x86\_64-3.2.12/ /usr/local/mongodb

3.4 创建数据和日志保存文件夹

mkdir /home/work/package/data -p

mkdir /home/work/package/logs -p

3.5 在/usr/local/mongodb/bin下新建配置

vi mongodb.conf

dbpath = /hone/work/package/mongodb/data #数据文件存放目录

logpath = /hone/work/package/mongodb/logs/mongodb.log #日志文件存放目录

port = 27017 #端口

fork = true #以守护程序的方式启用，即在后台运行

nohttpinterface = true

#auth=true

bind\_ip=0.0.0.0

3.6 配置环境变量

export MONGODB\_HOME=/usr/local/mongodb

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

3.7 重启系统配置

source /etc/profile

3.8 启动

mongod -f /usr/local/mongodb/bin/mongodb.conf

3.9 关闭

mongod -f /usr/local/mongodb/bin/mongodb.conf --shutdown

3.10开启端口

firewall-cmd --zone=public --add-port=27017/tcp –permanent

3.11 查看端口

firewall-cmd --permanent --query-port=27017/tcp

3.12 重启防火墙

firewall-cmd –reload

3.1.3 开机自启动

vim /etc/rc.local

rm -rf /home/work/package/mongodb/mongod.lock

/usr/local/mongodb/bin/mongod -f /usr/local/mongodb/bin/mongodb.conf

### 4. Redis安装

1 . 安装目录

mkdir /home/work/package/redis -p;

2. 下载服务包，并进行解压

wget http://download.redis.io/releases/redis-5.0.4.tar.gz

tar xzf redis-5.0.4.tar.gz

3. 编译

cd redis-5.0.4

make

修改配置文件

#bind 127.0.0.1

protected-mode no

make

4. 编译测试

make test

5. 进图目录utils

cd utils;

6. 执行 安装命令

sh install\_server.sh

遇到问题：

make test报错 You need tcl 8.5 or newer in order to run the Redis test

解决办法：

wget http://downloads.sourceforge.net/tcl/tcl8.6.1-src.tar.gz

sudo tar xzvf tcl8.6.1-src.tar.gz -C /usr/local/

cd /usr/local/tcl8.6.1/unix/

sudo ./configure

sudo make

sudo make install

对配置文件redis.conf进行修改

cd /utils

sh install\_server.sh

安装参数

Port : 6379

Config file : /home/work/package/redis/redis-5.0.4/conf/6379.conf

Log file : /home/work/package/redis/redis-5.0.4/log/6379.log

Data dir : /home/work/package/redis/redis-5.0.4/data/6379

Executable : /home/work/package/redis/redis-4.0.9/src/redis-server

Cli Executable : /home/work/package/redis/redis-5.0.4/src/redis-cli

Is this ok? Then press ENTER to go on or Ctrl-C to abort.

配置文件



### 5. Nginx安装

<https://blog.csdn.net/ywd1992/article/details/83095855>

安装路径：/usr/local/webserver/nginx/

* 1. 前期准备

安装编译需要的gcc和gcc-c++

yum install -y gcc gcc-c++

nginx依赖

pcre-devel、openssl-devel、zlib-devel

yum install -y pcre pcre-devel openssl openssl-devel zlib zlib-devel

下载nginx源码包并解压到当前目录

cd /home/work/package/nginx

wget http://nginx.org/download/nginx-1.12.0.tar.gz

tar -zxf nginx-1.12.0.tar.gz

* 1. nginx编译安装

生成Makefile文件

cd nginx-1.12.0

./configure --prefix=/usr/local/webserver/nginx

编译源代码并安装

make && make install

安装成功，进入/usr/local/webserver/nginx启动即可

5.3 开机启动

vi /etc/init.d/nginx

chmod 777 /etc/init.d/nginx

chkconfig --add /etc/init.d/nginx

chkconfig nginx on

### 6. Nginx配置

user root root;  
worker\_processes 4;  
  
error\_log logs/error.log;  
#error\_log logs/error.log notice;  
#error\_log logs/error.log info;  
  
pid logs/nginx.pid;  
  
  
events {  
 use epoll;  
 multi\_accept on;  
 worker\_connections 65535;  
}  
  
http {  
 include mime.types;  
 default\_type application/octet-stream;  
  
 log\_format main\_format '$remote\_addr - $remote\_user [$time\_local] "$request" '  
 '$status $body\_bytes\_sent "$http\_referer" '  
 '"$http\_user\_agent" "$http\_x\_forwarded\_for"';  
 access\_log logs/access.log main\_format;  
  
 sendfile on;  
 #tcp\_nopush on;  
 #  
 client\_body\_buffer\_size 512k;  
   
 proxy\_intercept\_errors on;  
  
 keepalive\_timeout 65;  
  
 gzip on;  
   
 client\_max\_body\_size 50M;   
   
 upstream fupin-web{  
 server 192.168.105.12:19000;  
 }  
  
 upstream fp-api{  
 server 192.168.105.12:8080;  
 }  
  
 server {  
 listen 80;  
 server\_name localhost;  
  
 #charset koi8-r;  
  
 #access\_log logs/host.access.log main;  
   
 #后台管理  
 location / {  
 #root html;  
 #index index.html index.htm;  
 proxy\_pass http://fupin-web/fupin/;  
 proxy\_redirect off;  
 proxy\_set\_header Host $host;  
 proxy\_set\_header X-Real-IP $remote\_addr;  
 proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  
 #deny  
 #allow  
 client\_max\_body\_size 10m;  
 client\_body\_buffer\_size 128k;  
  
 proxy\_connect\_timeout 90;  
 proxy\_send\_timeout 90;  
 proxy\_read\_timeout 90;  
  
 proxy\_buffer\_size 4k;  
 proxy\_buffers 4 32k;  
 proxy\_busy\_buffers\_size 64k;  
 proxy\_temp\_file\_write\_size 64k;  
 keepalive\_timeout 30;  
 }  
   
 location /wxfupin/ {  
 #root html;  
 #index index.html index.htm;  
 proxy\_pass http://fupin-web/wxfupin/;  
 proxy\_redirect off;  
 proxy\_set\_header Host $host;  
 proxy\_set\_header X-Real-IP $remote\_addr;  
 proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  
 #deny  
 #allow  
 client\_max\_body\_size 10m;  
 client\_body\_buffer\_size 128k;  
  
 proxy\_connect\_timeout 90;  
 proxy\_send\_timeout 90;  
 proxy\_read\_timeout 90;  
  
 proxy\_buffer\_size 4k;  
 proxy\_buffers 4 32k;  
 proxy\_busy\_buffers\_size 64k;  
 proxy\_temp\_file\_write\_size 64k;  
 keepalive\_timeout 30;  
 }  
   
  
 #即时租  
 location /fp-api/ {  
 #root html;  
 #index index.html index.htm;  
 proxy\_pass http://fp-api/fp-api/;  
 proxy\_redirect off;  
 proxy\_set\_header Host $host;  
 proxy\_set\_header X-Real-IP $remote\_addr;  
 proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  
 #deny  
 #allow  
 client\_max\_body\_size 10m;  
 client\_body\_buffer\_size 128k;  
  
 proxy\_connect\_timeout 90;  
 proxy\_send\_timeout 90;  
 proxy\_read\_timeout 90;  
  
 proxy\_buffer\_size 4k;  
 proxy\_buffers 4 32k;  
 proxy\_busy\_buffers\_size 64k;  
 proxy\_temp\_file\_write\_size 64k;  
 keepalive\_timeout 30;  
 }  
  
 error\_page 500 502 503 504 /50x.html;  
 location = /50x.html {  
 root html;  
 }  
 }  
}

### 7. RabbitMQ 离线安装配置

<https://www.cnblogs.com/rmxd/p/11583932.html>

### 1.下载Erlang的rpm包

　RabbitMQ是Erlang语言编写，所以Erang环境必须要有，注：Erlang环境一定要与RabbitMQ版本匹配：<https://www.rabbitmq.com/which-erlang.html>

　　Erlang下载地址：<https://www.rabbitmq.com/releases/erlang/>（根据自身需求及匹配关系，下载对应rpm包）

### 2.下载RabbitMQ的rpm包

　　RabbitMQ下载地址：<https://www.rabbitmq.com/releases/rabbitmq-server/>（根据自身需求及匹配关系，下载对应rpm包）

### 3.下载socat的rpm包

　　rabbitmq安装依赖于socat，所以需要下载socat。

　　socat下载地址：<http://repo.iotti.biz/CentOS/6/x86_64/>[socat-1.7.3.2-1.el6.lux.x86\_64.rpm](http://repo.iotti.biz/CentOS/6/x86_64/socat-1.7.3.2-1.el6.lux.x86_64.rpm)

### 4.分别安装Erlang、Socat、RabbitMQ（一定按照顺序！）

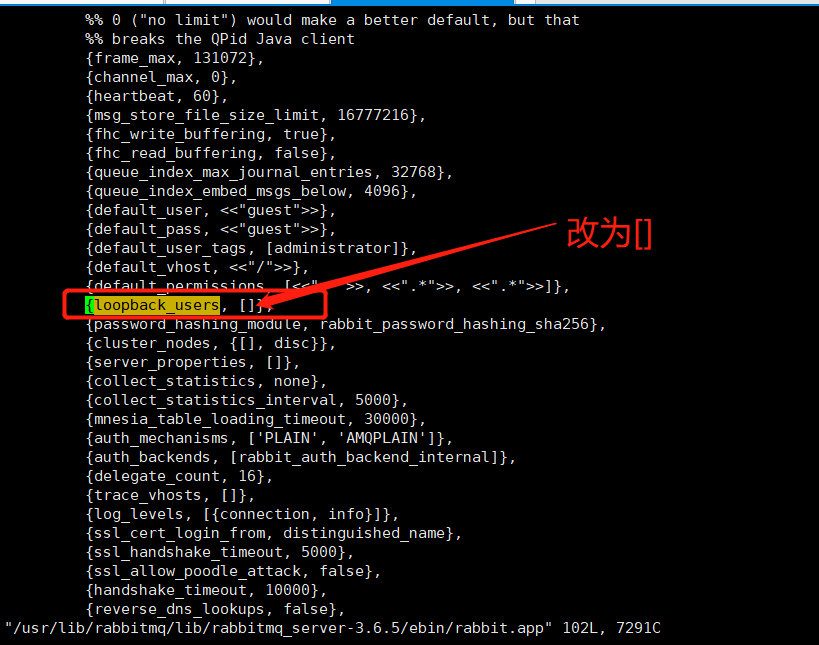
**一定按照以下顺序安装**：

　　①rpm -ivh erlang-18.3-1.el6.x86\_64.rpm

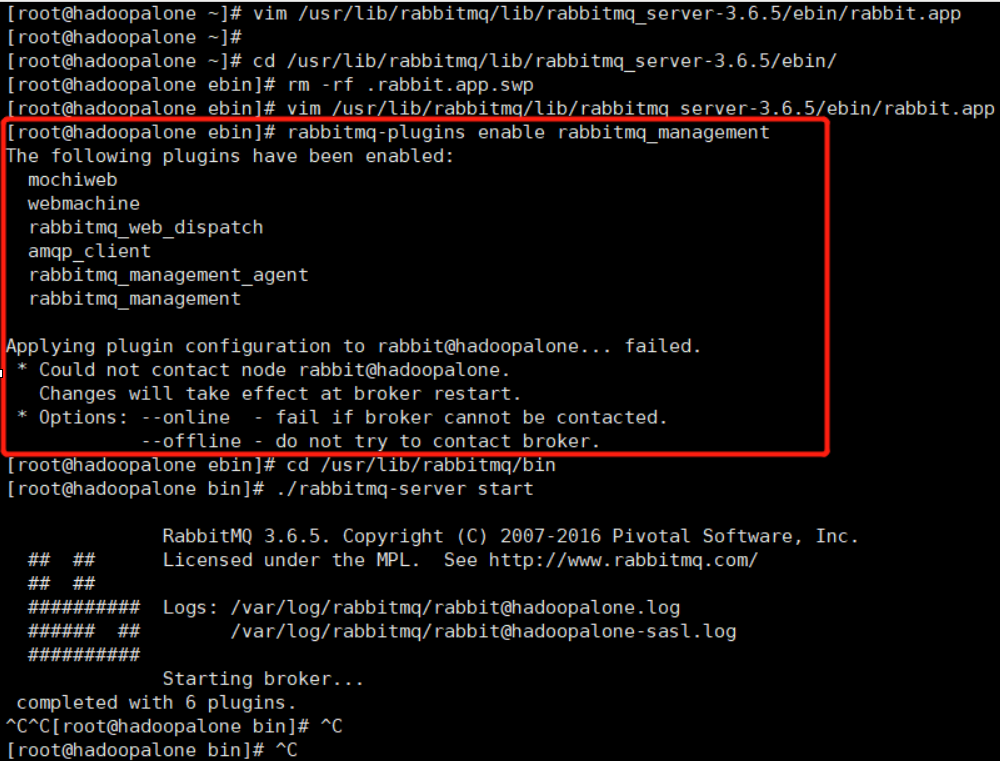
　　②rpm -ivh socat-1.7.3.2-1.el6.lux.x86\_64.rpm

　　③rpm -ivh rabbitmq-server-3.6.5-1.noarch.rpm

### 5.配置rabbitmq：vim /usr/lib/rabbitmq/lib/rabbitmq\_server-3.6.5/ebin/rabbit.app



### 6.安装管理插件：rabbitmq-plugins enable rabbitmq\_management



### 6.启动RabbitMQ，浏览器访问：[http://192.168.64.128:15672/](http://192.168.64.128:15672/#/)，出现以下界面说明安装完成！

　　①cd /usr/lib/rabbitmq/bin

　　②./rabbitmq-server start

### 8.Fastdfs安装配置

### 9. ES安装配置

<https://www.cnblogs.com/yanketao/p/10983556.html>

### 10. HDFS安装配置

## 服务部署

### 后台服务部署

#### 2.1.0部署要求

需要安装JDK,没有安装具体方法可查看1.1

#### 2.1.1 部署路径

//加版本·号

/home/work/code/package

启动脚本

#! /bin/sh

services="framework.eureka framework.auth framework.security framework.zuul framework.oss fupin.information fupin.template fupin.verification fupin.view framework-system fupin.archives"

for service in $services

do

service\_name=`ls | grep $service`

echo "`date +'%F %T'` [INFO] start $service\_name"

java -jar -Dspring.profiles.active=pro -Xms1024m -Xmx1024m $service\_name > /dev/null 2>&1 &

sleep 12s

done



停止脚本

ps -ef | grep java |grep framework | awk '{print $2;}' > pids

ps -ef | grep java |grep fupin | awk '{print $2;}' >> pids

cat pids | while read line

do

kill $line

done



查看服务状态

#### 2.1.2 每次部署代码中配置文件需要修改的位置

找到所有的生产环境配置文件application-pro.yml

##### redis配置

redis:  
 host: 119.29.199.151  
 port: 6379  
 password: legaoyun2018  
 database: 0

修改host为安装加点ip

修改密码为安装时候的密码

端口号未为安装时候端口号

##### Mongo配置

data:  
 mongodb:  
 host: 119.29.199.151  
 port: 27017  
 database: fupin

修改host为安装加点ip

database: fupin

端口号为为安装时候端口号

##### Mysql配置

datasource :  
 driver : com.mysql.jdbc.Driver  
 url: jdbc:mysql://119.29.199.151:3306/fp\_archives?useUnicode=true&characterEncoding=utf8&useSSL=false&allowMultiQueries=true  
 username : root  
 password : Changeme\_123  
 initsize : 10  
 maxActive : 20  
 minIdle : 10  
 maxWait : 120000  
 poolPreparedStatements : false  
 maxOpenPreparedStatements : -1  
 validationQuery : select 1  
 testOnborrow : true  
 testOnReturn : true  
 testWhileIdle : true  
 timeBetweenEvictionRunsMillis : 120000  
 filters : log4j,stat

修改url中的119.29.199.151:3306为安装时mysql节点的ip和端口号

修改username

修改password为设置的密码

##### Eureka:配置

eureka:  
 instance:  
 prefer-ip-address: true  
 lease-expiration-duration-in-seconds: 15  
 lease-renewal-interval-in-seconds: 5  
 client:  
 serviceUrl:  
 defaultZone: http://114.116.44.104:9871/eureka  
 registry-fetch-interval-seconds: 5

修改defaultZone为eureka 部署加点ip,和端口号（高可用用逗号分开）

##### 文件服务器配置

file:  
 service:  
 url: http://113.134.212.6:9999/

修改文件url为文件服务器的url

#### 2.1.3 数据库数据导入（存在问题，）

##### mysql数据导入

##### Mongo数据导入

### 前台服务部署

用nginx

#### 环境要求

JDK1.8 tomcat8.0

#### 部署路径

/home/work/web/tomcat(没有的话进行创建)

#### web服务部署

将web前端打的包解压到/home/work/web/tomca/webapps/fupin路径下

#### 微信端服务部署

将web前端打的包解压到/home/work/web/tomca/webapps/wxfupi路径下

#### 高可用环境部署

在nginx配置文件中的负载中添加对应的节点

### java服务和tomcat开机自启动

vi /etc/rc.local

cd /home/work/web/apache-tomcat-8.0.53/bin

./startup.sh

cd /home/work/service

./start.sh



### 数据备份

增量备份调研（nysql，mongo）

主从同步（nysql，mongo）

#### mongo数据备份

**#! /bin/sh**host=119.29.199.151  
dbname=fupin  
*echo* "*`*date +'%F %T'*`* [INFO] start backup mysql, host: $host"  
  
delete\_date=*`date -d'7 day ago' +'%Y%m%d'`*cur\_date=*`date +"%Y%m%d"`*if [ -d /home/work/backup/mongo/${delete\_date} ]  
then  
 *echo* "*`*date +'%F %T'*`* [INFO] delete /home/work/backup/mongo/${delete\_date}"  
 *rm* -rf /home/work/backup/mongo/${delete\_date}  
fi  
  
if [ -d /home/work/backup/mongo/${cur\_date} ]  
then  
 *echo* "*`*date +'%F %T'*`* [INFO] delete /home/work/backup/mongo/${cur\_date}"  
 *rm* -rf /home/work/backup/mongo/${cur\_date}  
  
 *echo* "*`*date +'%F %T'*`* [INFO] mkdir ${cur\_date}"  
 *mkdir* -p /home/work/backup/mongo/${cur\_date}  
else  
  
 *echo* "*`*date +'%F %T'*`* [INFO] mkdir ${cur\_date}"  
 *mkdir* -p /home/work/backup/mongo/${cur\_date}  
fi  
  
  
*mongodump* -h $host -d $dbname -o /home/work/backup/mongo/${cur\_date}

#### mysql数据备份

**#! /bin/sh**host=119.29.199.151  
username=root  
password=Changeme\_123  
  
*echo* "*`*date +'%F %T'*`* [INFO] start backup mysql, host: $host"  
  
delete\_date=*`date -d'7 day ago' +'%Y%m%d'`*cur\_date=*`date +"%Y%m%d"`*if [ -d /home/work/backup/mysql/${delete\_date} ]  
then  
 *echo* "*`*date +'%F %T'*`* [INFO] delete /home/work/backup/mysql/${delete\_date}"  
 *rm* -rf /home/work/backup/mysql/${delete\_date}  
fi  
  
if [ -d /home/work/backup/mysql/${cur\_date} ]  
then  
 *echo* "*`*date +'%F %T'*`* [INFO] delete /home/work/backup/mysql/${cur\_date}"  
 *rm* -rf /home/work/backup/mysql/${cur\_date}  
  
 *echo* "*`*date +'%F %T'*`* [INFO] mkdir ${cur\_date}"  
 *mkdir* -p /home/work/backup/mysql/${cur\_date}  
else  
  
 *echo* "*`*date +'%F %T'*`* [INFO] mkdir ${cur\_date}"  
 *mkdir* -p /home/work/backup/mysql/${cur\_date}  
fi  
*mysqldump* -h$host -u$username -p$password --opt fp\_archives *> /home/work/backup/mysql/${cur\_date}/fp\_archives\_backup.dump  
mysqldump* -h$host -u$username -p$password --opt fp\_base *> /home/work/backup/mysql/${cur\_date}/fp\_base\_backup.dump  
mysqldump* -h$host -u$username -p$password --opt fp\_info *> /home/work/backup/mysql/${cur\_date}/fp\_info\_backup.dump  
mysqldump* -h$host -u$username -p$password --opt fp\_template *> /home/work/backup/mysql/${cur\_date}/fp\_template\_backup.dump  
mysqldump* -h$host -u$username -p$password --opt fp\_verification *> /home/work/backup/mysql/${cur\_date}/fp\_verification\_backup.dump  
mysqldump* -h$host -u$username -p$password --opt template *> /home/work/backup/mysql/${cur\_date}/template\_backup.dump*

#### 日志备份和jar包备份

**1. 设置无需密码的ssh登陆，方法如下：**

　　假设A，B两[服务器](http://server.chinaitlab.com/)，现在需要在A机上用root登陆B机，而不需要输入密码，那我们可按照下面的步骤来做：

**1)在A机上生成钥匙对,执行以下命令：**

　　ssh-keygen -t rsa

　　Generating public/private rsa **key** pair.

　　Enter **file** in which to save the key (/root/.ssh/id\_rsa): /root/.ssh/id\_rsa

　　Enter **passphrase** (empty for no passphrase):直接回车

　　Enter same passphrase again:直接回车

　　Your identification has been saved in /root/.ssh/id\_rsa.

　　Your **public key** has been saved in /root/.ssh/id\_rsa.pub.

　　The key fingerprint is:

f6:61:a8:27:35:cf:4c:6d:13:22:70:cf:4c:c8:a0:23 root@host1

备份脚本

**#! /bin/sh**host=119.29.199.151  
username=root  
delete\_date=*`date -d'7 day ago' +'%Y%m%d'`*cur\_date=*`date +"%Y%m%d"`  
scp* -r /home/work/logs $username@$host:/home/work/backup/log/${cur\_date}  
*scp* -r /home/work/service $username@$host:/home/work/backup/service/${cur\_date}

#### 定时任务设置（**crontab**）

crontab -e

和编辑文本一样添加

#每周六三点执行

0 3 \* \* 6 /home/work/task/mysql\_backup.sh

#每周六四点执行

0 4 \* \* 6 /home/work/task/mongo\_backup.sh

#每周六四点执行

0 4 \* \* 6 /home/work/task/log\_jar\_backup.sh