### SSH免密码登录

服务器上的操作:

1. linux服务器上创建免密登录的用户.

useradd git

     passwd git

     至此在./home/下就会产生一个git文件夹, 即是git用户目录

2. 改变git目录为700,  .ssh文件为700,  authorized\_keys文件600. 以上文件的权限必须如此,高了低了都不行,此步骤非常重要. 如果没有.ssh和authorized\_keys文件需要自己手动创建.

本地window环境的操作，在window下打开git bash

1. 本地生成公私钥对：ssh-keygen -t rsa

在用户目录/.ssh目录下，将公钥内容添加到linux服务器 /home/git/.ssh/authorized\_keys文件中，配置完重启ssh: service sshd restart

1. ssh-copy-id -i ~/.ssh/id\_rsa.pub git@111.22.180.99

(该命令解释:  111.22.180.99 为你服务器IP地址, git为你服务器上用户名, 也就是你要为那个linux用户配置免密登录.

3 ：登录命令： ssh [git@111.22.180.99](mailto:git@111.22.180.99)

原文链接：

<https://blog.csdn.net/weixin_37281289/article/details/86658005>

### Jdk环境：

export JAVA\_HOME=/mnt/jdk/jdk1.8.0\_171

export JRE\_HOME=${JAVA\_HOME}/jre

export CLASSPATH=.:${JAVA\_HOME}/lib:${JRE\_HOME}/lib

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

source /etc/profile;

Java version;

### Tomcat解压即可

### 安装宝塔

yum install -y wget && wget -O install.sh http://download.bt.cn/install/install.sh && sh install.sh

### Mysql

# wget http://dev.mysql.com/get/mysql-community-release-el7-5.noarch.rpm

# rpm -ivh mysql-community-release-el7-5.noarch.rpm

# yum install mysql-community-server

# service mysqld restart

mysql -u root

set password for 'root'@'%'=password('zhongjihuajun@2019');

grant all privileges on \*.\* to root@'%'identified by 'juran@2019';

create user 'juranding'@'localhost' identified by 'juranding\_test';

flush privileges;

查看mysql启动时读取配置文件的默认目录

mysql --help|grep 'my.cnf'

### 5.Nginx服务器

1. yum -y install gcc zlib zlib-devel pcre-devel openssl openssl-devel

//创建一个文件夹

2.cd /usr/local

mkdir nginx

cd nginx//下载tar包

wget http://nginx.org/download/nginx-1.13.7.tar.gz  
tar -xvf nginx-1.13.7.tar.gz

1. cd /usr/local/nginx

./configure

./configure --with-http\_ssl\_module

make

make install

Cd nginx/sbin/

启动：./nginx

重启：./nginx -s reload

NGINX\_https配置教程:https://www.cnblogs.com/saneri/p/5391821.html

### 6.redis服务

1.下载

wget <http://download.redis.io/releases/redis-4.0.8.tar.gz>

tar -xzvf redis-4.0.8.tar.gz

cd redis-4.0.8

make && cd src && make install

cd ../utils

4.把redis作为一个系统的daemon进程去运行的，每次系统启动，redis进程一起启动

（1）redis utils目录下，有个redis\_init\_script脚本

cd ../utils

1. 将redis\_init\_script脚本拷贝到linux的/etc/init.d目录中，将redis\_init\_script重命名为redis\_6379，6379是我们希望这个redis实例监听的端口号

mv redis\_init\_script /etc/init.d/redis\_6379

（3）修改redis\_6379脚本的第6行的REDISPORT，设置为相同的端口号（默认就是6379）

（4）创建两个目录：/etc/redis（存放redis的配置文件），/var/redis/6379（存放redis的持久化文件）

mkdir /etc/redis

mkdir /var/redis/6379

1. 修改redis配置文件（默认在根目录下，redis.conf），拷贝到/etc/redis目录中，修改名称为6379.conf

mv redis.conf /etc/redis/6379.conf

（6）修改redis.conf中的部分配置为生产环境

daemonize yes 让redis以daemon进程运行

pidfile /var/run/redis\_6379.pid 设置redis的pid文件位置

port 6379 设置redis的监听端口号

dir /var/redis/6379 设置持久化文件的存储位置

（7）启动redis，执行cd /etc/init.d, chmod 777 redis\_6379，./redis\_6379 start

（8）确认redis进程是否启动，ps -ef | grep redis

（9）让redis跟随系统启动自动启动

在redis\_6379脚本中，最上面，加入两行注释

# chkconfig: 2345 90 10

# description: Redis is a persistent key-value database

执行命令：

chkconfig redis\_6379 on

./redis\_6379 start

### 7：NODEJS环境

1：yum install -y gcc make gcc-c++ openssl-devel wget

2：wget <https://nodejs.org/dist/v9.3.0/node-v9.3.0.tar.gz>

3：tar -xf node-v9.3.0.tar.gz

4：cd node-v9.3.0

5：./configure

6：make && make install

### 8：Go环境

1：yum update -y && yum install git wget bzip2 vim gcc-c++ ntp epel-release nodejs cmake -y

wget https://dl.google.com/go/go1.10.linux-amd64.tar.gz

tar -C /usr/local -xzf go1.10.linux-amd64.tar.gz

echo 'export GOROOT=/usr/local/go' >> /etc/profile

echo 'export PATH=$PATH:$GOROOT/bin' >> /etc/profile

echo 'export GOPATH=/root/go' >> /etc/profile

echo 'export PATH=$PATH:$GOPATH/bin' >> /etc/profile

source /etc/profile

---------------------

https://blog.csdn.net/qq\_35753140/article/details/79463735

### 9：开机自启动设置

cd /etc/rc.d/init.d

chkconfig --add appStart.sh

chkconfig appStart.sh on

chkconfig | grep appStart.sh

### 10：其他注意事项：

1：注意app和环境的启动顺序，比如说mysql->redis->springBootApp

2：写脚本的时候不要使用window编辑器

3：将脚本启动放在/etc/rc.d/init.d目录下，手动启动无问题再使用以上chkconfig 命令添加。

4：启动脚本注意文件头。

### unison安装（双端文件同步）

大致思路：双端安装unison，安装环境变量。客户端生产一个ssh公钥，server添加信任在linux下信任公钥文件位置：/home/用户/.ssh/authorized\_keys 。

* 1. windows安装包<https://www.irif.fr/~vouillon/unison/>
  2. linux下安装：yum install -y ocaml unison
  3. 查看版本：unison -version。打印版本号
  4. 配置公钥信任：

不要使用root有安全隐患。

首先创建用户: useradd unison

密码： passwd xxx

切换用户： su - unison

注意上传目录的文件权限要给到新用户。

root用户 ： chown -R unison /目录

unison用户： chmod 700 /目录

11.5将客户端公钥写入到authorized\_keys 文件：

命令：vi ~/.ssh/authorized\_keys

按 inset键粘贴内容，保存退出。

重启ssh服务： service sshd restart

11.6客户端流程：

生成ssh公钥，配置文件链接服务端，创建文件测试。

客户端配置配置文件：

#Unison preferences file

#同步目录

root = /home/unisonadm/test

#同步到server端目标目录

root = ssh://unison@ip//home/unisonadm/test/

ignore = Path runtime/\* #不同步的文件

ignore = Path application/config.php

ignore = Path application/database.php

ignore = Path LICENSE.txt

batch = true

owner = true

group = true

perms = -1

fastcheck = false

rsync = false

sshargs = -C

xferbycopying = true

log = true

logfile = /home/unisonadm/.unison/unison.log

测试:在第一个root目录下创建文件，如果服务端配置的目标地方有说明配置成功。

参考博客链接：

<https://blog.csdn.net/z13615480737/article/details/81062573>

#### 脚本appStart.sh内容：

#!/bin/bash

#chkconfig: 2345 80 90

#description:开机自动启动的脚本程序

# 开启

echo 'hello,word!'

cd /etc/rc.d/init.d

nohup /mnt/java/jre1.8.0\_161/bin/java -jar /mnt/jar/eureka-server-0.0.1-SNAPSHOT.jar > app.log &

cd /usr/local/nginx/sbin

./nginx