**centos7.3 测试服务器配置**

sshpass -p Szzx20170830 ssh root@192.168.1.200

用户   密码  
root   123456

useradd zhufukun  
passwd zhufukun

**1 JDK安装**

解压 jdk-8u131-linux-x64.tar.gz安装包

[root@localhost zhufukun]# mkdir /usr/java

[root@localhost zhufukun]# tar -zxvf jdk-8u131-linux-x64.tar.gz -C /usr/java

5、设置环境变量

[root@localhost zhufukun]# vim /etc/profile

在最前面添加：

export JAVA\_HOME=/usr/java/jdk1.8.0\_131

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

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

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

6、执行profile文件

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

这样可以使配置不用重启即可立即生效。

7、检查新安装的jdk

[root@localhost zhufukun]# java -version

显示：

java version "1.8.0\_131"

Java(TM) SE Runtime Environment (build 1.8.0\_131-b11)

Java HotSpot(TM) 64-Bit Server VM (build 25.131-b11, mixed mode)

**1 CentOS 7**

默认安装好之后是没有自动开启网络连接的ifconfig 安装sudo apt install net-tools

**2 gogs**

1 git 工具安装  
2 go 环境安装  
3 mysql安装  
4 添加git 用户（gitdata）  
5 下载gogs二进制包  
 位置：/home/gitdata/gogs/gogs  
     切换用户  
         su gitdata  
     运行  
         /home/gitdata/gogs/gogs web &  
     #可能存在权限问题：chown gitdata ~/gogs  
       
6 打开3000端口&&登录http://192.168.1.200:3000  
 进行配置（迁移则直接跳过该步骤）  
7 根据需要进行配置  
 位置：/home/gitdata/gogs/custom/conf/app.ini

配置(网页配置后自动出现配置)：

SSH\_ROOT\_PATH 必须填写为 /home/gitdata/.ssh 否则可能导致ssh无法登录root问题

[server]  
PROTOCOL = http  
DOMAIN = localhost  
HTTP\_ADDR = 0.0.0.0  
HTTP\_PORT = 3000  
SSH\_ROOT\_PATH = /home/gitdata/.ssh

**3 gogs 迁移：**

1 将所有代码 gogs-repositories复制到 /home/gitdata  
2 将数据库gogs 导入到目标mysql数据库  
管理员 帐号fgecctv 密码fgecctv4000

问题：代码可复制但是不能上传 `pre-receive hook declined`  
解决：删除每个仓库里的pre-receive 文件  
mv gogs-repositories/iot/cloudringad.git/hooks/pre-receive gogs-repositories/iot/cloudringad.git/hooks/pre-receive.old

**4 mongodb**

问题：2017-09-08T09:31:39.246+0800 warning: Failed to connect to 127.0.0.1:27017, reason: errno:111 Connection refused  
解决：服务没有开启，这个端口没有服务在使用。重新正确启动mongo

**5 nginx安装**

一. gcc 安装  
安装 nginx 需要先将官网下载的源码进行编译，编译依赖 gcc 环境，如果没有 gcc 环境，则需要安装：  
yum install gcc-c++  
​  
二. PCRE pcre-devel 安装  
PCRE(Perl Compatible Regular Expressions) 是一个Perl库，包括 perl 兼容的正则表达式库。nginx 的 http 模块使用 pcre 来解析正则表达式，所以需要在 linux 上安装 pcre 库，pcre-devel 是使用 pcre 开发的一个二次开发库。nginx也需要此库。命令：  
yum install -y pcre pcre-devel  
​  
三. zlib 安装  
zlib 库提供了很多种压缩和解压缩的方式， nginx 使用 zlib 对 http 包的内容进行 gzip ，所以需要在 Centos 上安装 zlib 库。  
yum install -y zlib zlib-devel  
​  
四. OpenSSL 安装  
OpenSSL 是一个强大的安全套接字层密码库，囊括主要的密码算法、常用的密钥和证书封装管理功能及 SSL 协议，并提供丰富的应用程序供测试或其它目的使用。  
nginx 不仅支持 http 协议，还支持 https（即在ssl协议上传输http），所以需要在 Centos 安装 OpenSSL 库。  
yum install -y openssl openssl-devel  
wget -c https://nginx.org/download/nginx-1.12.1.tar.gz  
1 配置  
 ./configure   
2 编译  
 make  
 make install  
3 查找  
 whereis nginx  
 # usr/local/nginx  
4 启动  
 nginx  
 nginx -s stop # kill  
 nginx -s quit # 等待nginx处理完任务后停止  
 nginx -t           # 测试conf文件  
 nginx -s reload # 可以更新配置  
5 重启：先停止再启动  
 nginx -s quit  
 nginx   
6 开机启动  
 vi /etc/rc.local 添加一行 /etc/rc.d/rc.local的链接  
 /usr/local/nginx/sbin/nginx  
 chmod 755 rc.local  
7 配置    
vim /etc/nginx/nginx.conf  
/usr/local/nginx/sbin/nginx -s quit  
/usr/local/nginx/sbin/nginx /etc/nginx/nginx.conf

**6 mount**

将整块磁盘1T mount到系统上

查看文件系统是否已挂载df -h

xiao@xc:~/模板$ sudo mount /dev/sda /mnt/disk/

mount: wrong fs type, bad option, bad superblock on /dev/sda,

  missing codepage or helper program, or other error  
   In some cases useful info is found in syslog - try  
   dmesg | tail or so.

**7 mysql 安装配置**

mysql  
mysql -h 120.76.214.138 -P 3336 -u datauser -pfgecctv714 -D cloudringproperty  
mysql -h 192.168.1.200 -P 3306 -udatauser -p123456   
mysql -h 192.168.1.200 -P 3306 -u datauser -p

创建用户：  
CREATE USER 'username'@'host' IDENTIFIED BY 'password';  
CREATE USER 'pig'@'%' IDENTIFIED BY '123456';   
授权：  
GRANT SELECT, INSERT , UPDATE, DELETE ON \*.\* TO 'pig'@'%'  
GRANT ALL ON \*.\* TO 'datauser'@'%';  
INSERT INTO user(host, user, ssl\_cipher,x509\_issuer,x509\_subject) values('localhost', 'datauser', '','','');  
​  
忽略大小写：  
vim /etc/my.cnf  
[mysqld]  
lower\_case\_table\_names=1

**8 emqtt 安装**

http://www.emqtt.io/downloads/latest/centos7-rpm  
rpm -ivh --force emqttd-centos7-v2.3-beta.3-1.el7.centos.x86\_64.rpm   
vim /etc/emqttd/emq.conf  
dashboard:http://192.168.1.200:18083  
emqttd console # 可以看到各种端口  
​  
dashboard:http listen on 0.0.0.0:18083 with 2 acceptors.  
mqtt:tcp listen on 127.0.0.1:11883 with 16 acceptors.  
mqtt:tcp listen on 0.0.0.0:1883 with 16 acceptors.  
mqtt:ws listen on 0.0.0.0:8083 with 4 acceptors.  
mqtt:ssl listen on 0.0.0.0:8883 with 16 acceptors.  
mqtt:wss listen on 0.0.0.0:8084 with 4 acceptors.  
mqtt:api listen on 127.0.0.1:8080 with 4 acceptors.  
​  
# 端口修改  
306 listener.tcp.external = 0.0.0.0:1885  
​  
# 配置：/etc/emqttd  
# 日志   
/var/log/emqttd/

**9 iptables 防火墙配置**

敲黑板： 防火墙开好了端口一定要看下服务是不是绑定了本地，比如supervisor, mongo这些  
filter链：根据数据包的分析资料”对比”预先定义的规则内容，若数据包数据与规则内容相同则进行动作，否则就继续下一条规则的比对。所以位置很重要  
[root@localhost ~]# vim /etc/sysconfig/iptables  
[root@localhost ~]# service iptables restart  
Redirecting to /bin/systemctl restart iptables.service  
修改配置，重启over

iptables-restore < /etc/iptables.rules  
service iptables restart

iptables -A INPUT -p tcp --dport 20 -j ACCEPT   
iptables -I INPUT -p tcp --dport 4000:5000 -j ACCEPT

1 链管理命令：  
 -P 设置默认策略： iptables -P INPUT DROP 把默认对则拒绝，所有远程连接都被拒绝  
   -F 清空规则链 ： iptables -t nat -F PREROUTING  
   iptables -t noat -F 清空nat表的所有链  
   -N 支持用户新建一个链：iptables -N inbound\_tcp\_web   
   -X 删除用户自定义链  
   -E 自定义链重命名  
   -Z 清空链即链中默认规则的额计算器  
      
2 规则管理：-A 追加 -I 3 插入到第三行 -R 3 取代第三行 -D 3 删除  
iptables -I 16 INPUT -p tcp --dport 9001 -j ACCEPT  
iptables -I INPUT -s 0.0.0.0/0 -p tcp --dport 9001 -j ACCEPT   
3 查看管理命令 -L  
 -n 数字显示 -v 详细显示 -vv -vvv  
 -x 在计数器上显示精确值，不做单位换算  
 --line-numbers 显示规则行号  
 -t nat:显示所有的关卡信息

1 通用匹配：源地址目标地址的匹配  
 -s :指定作为源地址匹配，这里不能指定主机名称，必须是IP | IP/MASK|0.0.0.0/0.0.0.0  
 而且地址可以去翻，加一个！ 表示除了哪个ip之外  
 -d: 目标地址  
 -p: 匹配协议 TCP/UDP/ICMP  
 -i eth0: 从这块网卡流入的数据  
 流入一般用在INPUT PREROUTING  
 -o eth0: 从这块网卡流出的数据  
 OUTPUT POSTROUTING  
2 扩展匹配 隐含扩展  
-p tcp --dport xx-xx: 指定目标端口或连续端口  
-p tcp --sport: 指定源端口  
--tcp-flags : tcp的标志位 SYN ACK FIN PSH RST URG  
--tcp-flags syn,ack,fin,rst syn = --syn  
 # 表示检查这厮个标志位，syn必须为1,其余为0,表示三次握手第一次包  
-p udp --dport/--sport  
-p icmp:   
3 显式匹配  
-m multiport --dport 21,23,80

-j ACTION  
DROP: 悄悄丢弃  
REJECT： 拒绝  
ACCEPT： 接受  
DNAT  
SNAT  
MASQUERADE： 源地址伪装  
REDIRECT 重定向：主要用于实现端口重定向  
MARK： 打防火墙标记  
RETURN 返回

**10 开机启动配置**

vim /etc/rc.d/rc.local

/usr/local/nginx/sbin/nginx &  
/usr/local/mongodb/bin/mongod --config /usr/local/mongodb/bin/mongodb.conf &  
/home/git/gogs/gogs web &

**11 python**

1 pip 安装  
 yum install python-pip  
 # 如果python-pip 不存在，使用yum -y install epel-release  
 pip install --upgrade pip  
 pip install -r requirements.py

2 MySQLdb 安装  
 # yum install python-devel mysql-devel zlib-devel openssl-devel  
 pip install mysql-python  
3 connector 安装  
 # pip install mysql-connector  
 # sudo yum install mysql-connector-python-rf  
 pip install mysql-connector==2.1.4

**12 pip**

pip freeze > requirements.txt  
pip install -r requirements.txt

**13 静态IP配置**

[root@localhost ~]# vim /etc/sysconfig/network-scripts/ifcfg-enp2s0  
TYPE=Ethernet  
BOOTPROTO=static # dhcp  
DEFROUTE=yes  
PEERDNS=yes  
PEERROUTES=yes  
IPV4\_FAILURE\_FATAL=no  
IPV6INIT=yes  
IPV6\_AUTOCONF=yes  
IPV6\_DEFROUTE=yes  
IPV6\_PEERDNS=yes  
IPV6\_PEERROUTES=yes  
IPV6\_FAILURE\_FATAL=no  
IPV6\_ADDR\_GEN\_MODE=stable-privacy  
NAME=enp2s0  
UUID=b43a3c76-cf79-4ec7-b4dc-0e424d1f0098  
DEVICE=enp2s0  
ONBOOT=yes  
​  
IPADDR0=192.168.1.200  
NETMASK=255.255.255.0  
GATEWAY=192.168.1.1

**14 superivor**

yum install supervisor  
vim /etc/supervisord.conf 去掉注释  
[inet\_http\_server]         ; inet (TCP) server disabled by default  
port=127.0.0.1:9001       ; (ip\_address:port specifier, \*:port for all iface)  
username=user             ; (default is no username (open server))  
password=123               ; (default is no password (open server))  
​  
 ; 修改127.0.0.1:9001 为 0.0.0.0：9001

vim /etc/supervisord.d/cloudring\_cn.conf # 添加应用  
[program:userserver\_cn]  
command=python /ali/cloudring\_cn/userserver/server.py  
autostart=true  
autorestart=true  
autostart=true  
startsecs=5  
startretries=10000  
redirect\_stderr=true  
stopasgroup=true  
killasgroup=true  
stdout\_logfile=/ali/supervisor/log/userserver\_cn/shutdown.log  
stdout\_logfile\_maxbytes=10MB

supervisord # 开启服务  
supervisorctl reload  
supervisorctl start all/appname  
supervisorctl stop all/appname

**15 scp**

scp -P62 -r fgeuser@120.76.214.91/ali/supervisor .

**16 SELinux**

该服务与较多服务冲突，使用繁琐，建议关闭  
 1 查看状态/usr/sbin/sestatus -v    
 2 设置为 permissive: setenforce 0  
 3 修改/etc/selinux/config 文件  
 将SELINUX=enforcing改为SELINUX=disabled  
 重启机器即可

**17 redis**

# 安装  
yum install redis-server  
​  
# 命令位置  
/usr/bin/redis-cli  
/usr/bin/redis-server  
​  
vim /etc/redis.conf  
# 不注释掉，远程访问报错：Connection closed by server.  
# bind 127.0.0.1  
port 6379  
logfile /var/log/redis/redis.log  
dir /var/lib/redis # 数据库  
requirepass 123456  
​  
# 启动方式  
redis-server /etc/redis.conf --loglevel verbose --port 6379  
redis-cli -h 127.0.0.1 -p 6379 -a 123456  
service redis restart  
​  
# python连接  
from redis import Redis  
conf = {'host': '192.168.1.200',  
 'port': 3306,  
 'db': 0,  
 'password': '123456'}  
r = Redis(\*\*conf)  
r.set('a', 100)  
print(r.get("a"))

**18 网络接口**

1 网络接口命名  
 eth0: ethernet 以太网接口  
 wifi0: 无线局域网 无线网络接口  
 ath0: Atheros 指Atheros芯片所包含的无线网络接口  
 lo: local 简写， 本地环回接口

**19 jenkins**

用户名： xiao  
密码： 1992  
vim .jenkins/secrets/initialAdminPassword  
---------------  
tomcat:启动  
./startup.sh  
./shutdown.sh  
ps -ef |grep java  
/etc/init.d/iptables status  
​  
1、下载tomcat  
2、下载jenkins：http://updates.jenkins-ci.org/download/war/  
3、unzip tomcat  
4、将jenkins.war放到tomcat\webapps下  
5、tomcat\bin\startup.sh启动tomcat：./startup.sh  
注：Linux中新添加的文件需添加执行权限，cd到tomcat\bin：chmod+x \*  
6、启动成功后登录：http://127.0.0.1:8080/jenkins  
-jenkins初始化密码:  
 位置:/home/xiao/.jenkins/secrets/initialAdminPassword  
-安装插件

**构建配置**

1 登录

帐号 admin

密码 vim .jenkins/secrets/initialAdminPassword

2 新建 -》 构建一个自由风格的软件项目

3 源码管理：

配置仓库url, 帐号密码，相应分支。

4 构建 -》 Execute shell

#!/bin/bash

echo "==================================="

echo "======python automatic script======"

echo "==================================="

cd /ali/cloudring\_cn/

git pull

if [ $? -ne 0 ]; then

#echo "cloudring\_cn command failed!!! $time" >> /home/jenkins/jenkins.log

exit 1

else

echo "========================================="

echo "git pull ===[/home/cloudring\_cn]=== ok!!!"

echo "========================================="

#echo "git pull cloudring\_cn ok!!! $time" >> /home/jenkins/jenkins.log

fi

5 一般root用户执行该指令，需要将root的公钥加入git服务器的

~/.ssh/authorized\_keys中, 否则 pull会失败

**20 git**

git remote add origin https://github.com/xiaowenlong100/dj-mysite-polls.git

git push -u origin master

# 填好公钥后仍需要密码

# 路径不全导致代码拉取失败

git clone gitdata@192.168.1.200:/home/gitdata/gogs-repositories/python/cloudring.git

gitdata@192.168.1.200:python/cloudring.git

# 本地拉取远程非master分支

git checkout -b release origin/release

**21 信息**

wenlong.xiao@czbsit.com

1992mn

**22 telnet centos7**

1 安装

yum -y install xinetd telnet telnet-server

2 允许root帐号登录

vi /etc/securetty

#添加两行

pts/0

pts/1

3 添加防火墙

vi /etc/sysconfig/iptables

-A INPUT -p tcp -m state --state NEW -m tcp --dport 23 -j ACCEPT

4 重启服务，关闭firewalld

systemctl restart iptables

systemctl disable firewalld

systemctl stop firewalld

5 注册服务

systemctl enable telnet.socket

systemctl start telnet.socket

systemctl enable xinetd

systemctl start xinetd

6 测试

telnet localhost

**192.168.1.28 ubuntu**

192.168.1.28 workway HeMei457@#

sudo mongod --config /etc/mongod.conf

mongo 192.168.1.28:27017

**nginx安装**

一. 安装gcc依赖库  
sudo apt-get install build-essential

sudo apt-get install libtool

​  
二. 安装pcre依赖库   
PCRE(Perl Compatible Regular Expressions) 是一个Perl库，包括 perl 兼容的正则表达式库。nginx 的 http 模块使用 pcre 来解析正则表达式，所以需要在 linux 上安装 pcre 库，pcre-devel 是使用 pcre 开发的一个二次开发库。nginx也需要此库。命令：  
sudo apt-get update

sudo apt-get install libpcre3 libpcre3-dev

​  
三. 安装zlib 依赖库  
zlib 库提供了很多种压缩和解压缩的方式， nginx 使用 zlib 对 http 包的内容进行 gzip ，所以需要在 Centos 上安装 zlib 库。  
sudo apt-get install zlib1g-dev  
​  
四. 安装OpenSSL依赖库  
OpenSSL 是一个强大的安全套接字层密码库，囊括主要的密码算法、常用的密钥和证书封装管理功能及 SSL 协议，并提供丰富的应用程序供测试或其它目的使用。  
nginx 不仅支持 http 协议，还支持 https（即在ssl协议上传输http），所以需要在 Centos 安装 OpenSSL 库。  
sudo apt-get install openssl

五、nginx安装

#下载最新版本：

wget http://nginx.org/download/nginx-1.11.3.tar.gz

#解压：

tar -zxvf nginx-1.11.3.tar.gz

#进入解压目录：

cd nginx-1.11.3

1 配置  
 ./configure   
2 编译  
 make  
 make install  
3 查找  
 whereis nginx  
 # usr/local/nginx  
4 启动

cd /usr/local/nginx/sbin  
 ./nginx  
 ./nginx -s stop # kill  
 ./nginx -s quit # 等待nginx处理完任务后停止  
 ./nginx -t           # 测试conf文件  
 ./nginx -s reload # 可以更新配置  
5 重启：先停止再启动  
 ./nginx -s quit  
 ./nginx   
6 开机启动  
 vi /etc/rc.local 添加一行 /etc/rc.d/rc.local的链接  
 /usr/local/nginx/sbin/nginx  
 chmod 755 rc.local  
7 配置    
vim /etc/nginx/nginx.conf

**FTP**

sudo apt install vsftpd

service vsftpd start

service vsftpd stop

**Redis**

sudo apt install redis-server

# 命令位置  
/usr/bin/redis-cli  
/usr/bin/redis-server  
​  
vim /etc/redis.conf  
# 不注释掉，远程访问报错：Connection closed by server.  
# bind 127.0.0.1  
port 6379  
logfile /var/log/redis/redis.log  
dir /var/lib/redis # 数据库  
requirepass 123456  
​

#检测后台进程是否存在

ps -ef |grep redis

#检测6379端口是否在监听

netstat -lntp | grep 6379

kill pid

# 启动方式

#加上`&`号使redis以后台程序方式运行

redis-server &

redis-server  
redis-server /etc/redis.conf --loglevel verbose --port 6379

redis-cli  
redis-cli -h 127.0.0.1 -p 6379 -a 123456

redis-cli shutdown

**mongodb**

vim /etc/mongodb/mongo.conf  
dbpath = /etc/mongodb/db  
logpath = /etc/mongodb/log/mongodb.log  
​  
mkdir /etc/mongodb/db  
mkdir /etc/mongodb/log  
启动  
mongod --config=/etc/mongodb/mongo.conf --logappend --fork  
​  
验证  
client = pymongo.MongoClient(  
   host="192.168.1.28",  
)  
db = client.get\_database("xiao")  
if not db.get\_collection('xiao'):  
   db.create\_collection("a\_collection")  
coll = db.get\_collection('xiao')  
print('result: {}'.format(coll.insert\_one({"a": "abc"})))  
​  
mongo 客户端:  
show dbs  
use xiao  
show collections  
db.xiao.find()

Tomcat启动

启动tomcat服务  
方式一：直接启动 ./startup.sh  
方式二：作为服务启动 nohup ./startup.sh &  
方式三：控制台动态输出方式启动 ./catalina.sh run 动态地显示tomcat后台的控制台输出信息,Ctrl+C后退出并关闭服务  
解释：  
通过方式一、方式三启动的tomcat有个弊端，当客户端连接断开的时候，tomcat服务也会立即停止，通过方式二可以作为linux服务一直运行  
通过方式一、方式二方式启动的tomcat，其日志会写到相应的日志文件中，而不能动态地查看tomcat控制台的输出信息与错误情况，通过方式三可以以控制台模式启动tomcat服务，  
直接看到程序运行时后台的控制台输出信息，不必每次都要很麻烦的打开catalina.out日志文件进行查看，这样便于跟踪查阅后台输出信息。tomcat控制台信息包括log4j和System.out.println()等输出的信息。

**Linux命令**

新建删除用户，修改密码

useradd zhufukun

passwd

userdel zhufukun

查询用户所属组

groups zhufukun

新建组

groupadd zfk

修改用户所属组

usermod -g zfk zhufukun

查进程：ps -ef|grep java

结束进程：kill -9 pid

查看端口

netstat -apltu|grep 8080

查看文件 ll ls

文件权限

Chmod 774 -R file

Chown zhufukun -R tomcat