## 1.服务器配置

1.配置阿里云端口访问

ECS分配有公网ip外网可以访问，默认关闭只开启22(ssh远程登录)端口，登录阿里云控制台在入口开放所需端口。

控制台->云服务器ECS->更多->安全组配置->配置规则->添加安全组规则

常用端口：

8080：tomcat

3306：mysql数据库

80：http访问

22：ssh远程登录

2.安装常用编译环境

yum install -y gcc gdb strace gcc-c++ autoconf libjpeg libjpeg-devel libpng libpng-devel freetype freetype-devel libxml2 libxml2-devel zlib zlib-devel glibc glibc-devel glib2 glib2-devel bzip2 bzip2-devel ncurses ncurses-devel curl curl-devel e2fsprogs patch e2fsprogs-devel krb5-devel libidn libidn-devel openldap-devel nss\_ldap openldap-clients openldap-servers libevent-devel libevent uuid-devel uuid mysql-devel libaio unzip zip

1. 创建/usr/local/web/目录，上传安装包后解压.gz文件用tar -zxvf命令，.zip文件用unzip命令
2. 修改安装包目录名统一格式

fastdfs-5.11 fastdfs-nginx-module jdk-1.8 libfastcommon-master mycat-1.6 mysql-5.7 nginx-1.13.6 pcre-8.41 redis-4.0.9 tomcat-9.0

## 2.jdk配置

1. oracle官网下载jdk并使用filezilla上传到服务器

<http://download.oracle.com/otn-pub/java/jdk/8u161-b12/2f38c3b165be4555a1fa6e98c45e0808/jdk-8u161-linux-x64.tar.gz>

1. 配置环境变量

vim /etc/profile在文件末尾加入环境变量并加入mysql和tomcat配置并使用source /etc/profile命令使其立即生效

#jdk1.8

export JAVA\_HOME=/usr/local/web/jdk-1.8

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

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

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

#tomcat启动时间过长配置

export CATALINA\_HOME=/usr/local/web/tomcat-9.0

#mysql

export MYSQL\_HOME=/usr/local/web/mysql-5.7

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

1. 解决linux下tomcat启动慢问题，修改jdk的随机数生成配置

vim jdk-1.8/jre/lib/security/java.security

命令模式下输入/securerandom.source=file回车将搜索到的

securerandom.source=file:/dev/random

更改为

securerandom.source=file:/dev/./urandom

# 3.mysql安装

1.解压mysql-8.0压缩文件

cd /usr/local/web

tar -zxvf mysql-8.0.12-el7-x86\_64.tar.gz

2.在/data/mysql目录创建mysql数据存放目录和日志文件目录，执行脚本开始安装

mkdir -p /data/mysql/data

mkdir -p /data/mysql/log

3.vim /etc/my.cnf加入如下配置信息

[clent]

port = 3306

socket = /tmp/mysql.sock

[mysql]

#加快服务启动速度

no-auto-rehash

[mysqld]

#跳过域名解析

skip-name-resolve

#主从同步时server\_id要保证唯一

server\_id=1

port = 3306

user = root

character-set-server = utf8mb4

default\_storage\_engine = innodb

log\_timestamps = SYSTEM

socket = /tmp/mysql.sock

#安装目录

basedir = /usr/local/web/mysql-8.0

#数据目录

datadir = /data/mysql/data

#日志目录

log\_error = /data/mysql/log/mysql-error.log

slow\_query\_log\_file = /data/mysql/log/mysql-query.log

pid-file = /tmp/mysql.pid

#最大连接数，可根据SHOW GLOBAL STATUS LIKE 'max\_used\_connections'命令修改，默认128

max\_connections = 128

max\_connect\_errors = 100

#表缓存大小

table\_open\_cache = 2048

#.sql文件还原数据库时允许的最大文件大小

max\_allowed\_packet = 128M

open\_files\_limit = 65535

innodb\_buffer\_pool\_size = 1024M

innodb\_file\_per\_table = 1

innodb\_flush\_log\_at\_trx\_commit = 1

innodb\_log\_file\_size = 512M

innodb\_log\_files\_in\_group = 2

innodb\_log\_buffer\_size = 16M

innodb\_max\_dirty\_pages\_pct = 80

innodb\_lock\_wait\_timeout = 30

innodb\_data\_file\_path=ibdata1:1024M:autoextend

slow\_query\_log = 1

long\_query\_time = 1

4.修改mysql目录权限

chmod -R 755 mysql-8.0

5.执行安装命令

./bin/mysqld --initialize --explicit\_defaults\_for\_timestamp

6.vim /data/mysql/log/mysql-error.log查看最后一行末尾的初始密码（m>R1QX0sp:=z）

7.添加mysql开机自启动Esl%H)lDC2,!

cp ./support-files/mysql.server /etc/init.d/mysql

chkconfig --add mysql

chkconfig mysql on

8.启动mysql并测试登录

service mysql start

9.登录mysql控制台修改默认密码

alter user 'root'@'localhost' identified with mysql\_native\_password by '123456';

1. 开启远程登录权限

use mysql;

update user set Host=’%’ where user=root;

# 4.tomcat配置

1.需要为tomcat/bin下面的三个启动脚本赋予执行权限

chmod 755 /usr/local/web/tomcat-9.0/\*.sh

# **5.nginx配置**

**5.1安装fastdfs-nginx模块(fastdfs分布式文件系统，如果不需要可以不安装，需要安装fastdfs后才能安装这个模块)。**

1.在每一台服务器上,解压libfastcommon,进入libfastcommon-master目录执行  
./make.sh  
./make.sh install

2.安装结束后，由于FastDFS主程序设置的lib目录是/usr/local/lib，所以需要创建软链接：

ln -s /usr/lib64/libfastcommon.so /usr/local/lib/libfastcommon.so

ln -s /usr/lib64/libfastcommon.so /usr/lib/libfastcommon.so

ln -s /usr/lib64/libfdfsclient.so /usr/local/lib/libfdfsclient.so

ln -s /usr/lib64/libfdfsclient.so /usr/lib/libfdfsclient.so

3.2 解压FastDFS-nginx-module修改fastdfs-nginx-module/src的config文件，原来的内容是：

CORE\_INCS="$CORE\_INCS /usr/local/include/fastdfs /usr/local/include/fastcommon/"

修改为

CORE\_INCS="$CORE\_INCS /usr/include/fastdfs /usr/include/fastcommon"

## 5.2nginx安装

1.创建安装目录和日志目录

mkdir -p /usr/local/web/nginx-1.13/logs

2.解压nginx

cd /usr/local/web/

tar zxvf nginx-1.13.6.tar.gz

3.配置

cd /usr/local/web/nginx-1.13.6

如果安装fastdfs模块使用如下命令,必须已经安装fastdfs，否则会报错，请参考fastdfs安装配置

./configure --user=root --group=root --prefix=/usr/local/web/nginx-1.13 --with-http\_stub\_status\_module --with-http\_ssl\_module --with-http\_realip\_module --add-module=/usr/local/web/fastdfs-nginx-module/src

否则使用这个命令

./configure --user=root --group=root --prefix=/usr/local/web/nginx-1.13 --with-http\_stub\_status\_module --with-http\_ssl\_module --with-http\_realip\_module

4.编译安装

make & make install

5.如果安装有fastdfs模块，复制fastdfs-nginx-module配置文件

cp /usr/local/web/fastdfs-nginx-module/src/mod\_fastdfs.conf /etc/fdfs

修改拷贝后的配置文件主要修改如下内容

group\_name= group1#group名称

tracker\_server=192.168.1.128:22122#tracker服务器ip端口，可以写多个

tracker\_server=192.168.1.129:22122

store\_path0=/data/fastdfs/storage#storage目录

url\_have\_group\_name = true //在ＵＲＬ中包含组名

## 5.3启动

1. nginx配置文件

user root;

worker\_processes 1;

#error\_log logs/error.log;

#error\_log logs/error.log notice;

#error\_log logs/error.log info;

#pid logs/nginx.pid;

events {

worker\_connections 1024;

}

http {

include mime.types;

default\_type application/octet-stream;

#log\_format main '$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;

sendfile on;

#tcp\_nopush on;

#keepalive\_timeout 0;

keepalive\_timeout 65;

#开启文件压缩服务

gzip on;

gzip\_min\_length 1024;

gzip\_types text/plain application/javascript text/javascript;

fastcgi\_intercept\_errors on;

server {

listen 80;

server\_name wx.shifuhelp.com;

root /data/ai\_kang\_shui\_dian/;

index index.html;

client\_max\_body\_size 20m;

location =/ {

index index.html;

}

location / {

proxy\_pass http://localhost:8080/ai\_kang\_shui\_dian/;

error\_page 500 502 503 504 404 /404.html;

proxy\_set\_header x-forwarded-for $remote\_addr;

}

location ~ .\*\.(txt|html|css|js|jpg|jpeg|mp3|mp4|xls|png|gif|JPEG|PNG|JPG|bmp|BMP)$ {

root /data/ai\_kang\_shui\_dian;

expires 30d;

error\_page 500 502 503 504 404 /404.html;

}

error\_page 500 502 503 504 /404.html;

location = /404.html {

root /data/ai\_kang\_shui\_dian/;

}

}

}

2.启动

方法1

/usr/local/web/nginx/sbin/nginx -c /usr/local/nginx/conf/nginx.conf

方法2

cd /usr/local/web/nginx/sbin

./nginx

3.停止

pkill 9 nginx

**5.4添加nginx为系统服务**

1.vim /etc/init.d/nginx添加如下内容，蓝色的为安装路径，根据需要更改

#!/bin/bash

# nginx Startup script for the Nginx HTTP Server

# this script create it by caffreyxin at 2007.10.15.

# it is v.0.0.1 version.

# if you find any errors on this scripts, please contact caffreyxin.

# and send mail to xinyflove at sina dot com.

#

# chkconfig: - 85 15

# description: Nginx is a high-performance web and proxy server.

# It has a lot of features, but it's not for everyone.

# processname: nginx

# pidfile: /var/run/nginx.pid

# config: /usr/local/nginx/conf/nginx.conf

nginxd=/usr/local/web/nginx-1.13/sbin/nginx

nginx\_config=/usr/local/web/nginx-1.13/conf/nginx.conf

nginx\_pid=/var/run/nginx.pid

RETVAL=0

prog="nginx"

# Source function library.

. /etc/rc.d/init.d/functions

# Source networking configuration.

. /etc/sysconfig/network

# Check that networking is up.

[ ${NETWORKING} = "no" ] && exit 0

[ -x $nginxd ] || exit 0

# Start nginx daemons functions.

start() {

if [ -e $nginx\_pid ];then

echo "nginx already running...."

exit 1

fi

echo -n $"Starting $prog: "

daemon $nginxd -c ${nginx\_config}

RETVAL=$?

echo

[ $RETVAL = 0 ] && touch /var/lock/subsys/nginx

return $RETVAL

}

# Stop nginx daemons functions.

stop() {

echo -n $"Stopping $prog: "

killproc $nginxd

RETVAL=$?

echo

[ $RETVAL = 0 ] && rm -f /var/lock/subsys/nginx /var/run/nginx.pid

}

# reload nginx service functions.

reload() {

echo -n $"Reloading $prog: "

#kill -HUP `cat ${nginx\_pid}`

killproc $nginxd -HUP

RETVAL=$?

echo

}

# See how we were called.

case "$1" in

start)

start

;;

stop)

stop

;;

reload)

reload

;;

restart)

stop

start

;;

status)

status $prog

RETVAL=$?

;;

\*)

echo $"Usage: $prog {start|stop|restart|reload|status|help}"

exit 1

esac

exit $RETVAL

2.修改文件权限

chmod 755 /etc/init.d/nginx

3.设置开机启动

chkconfig nginx on

4.备注

启动服务：service nginx start

停止服务：service nginx stop

重启服务：service nginx reload

# 6.redis安装

1. 上传并解压缩redis安装包

cd /usr/local/web/

tar -zxvf redis-4.0.9.tar.gz

cd redis-4.0.9

1. 编译

make

1. 添加redis服务开机启动

cp utils/redis\_init\_script /etc/init.d/redis

vim /etc/init.d/redis

在第二行添加

#chkconfig: 2345 80 90

修改安装路径

EXEC=/usr/local/web/redis-4.0.9/src/redis-server

CLIEXEC=/usr/local/web/redis-4.0.9/src/redis-cli

设置开机启动

chkconfig --add redis

chkconfig on

1. 修改redis配置文件

mkdir /etc/redis

cp redis.conf /etc/redis/redis.conf

vim /etc/redis/redis.conf

daemonize no改为daemonize yes

注释bind 127.0.0.1 ::1

取消注释requirepass改为requirepass 你的密码

1. 启动redis

service redis start

1. 进入redis命令行工具

cd /usr/local/web/redis-4.0.9/src

redis-cli

# 7.mycat分库分表中间件安装

1.添加mycat环境变量

vi /etc/profile

追加 MYCAT\_HOME=/usr/local/web/mycat-1.6

source /etc/profile立即生效

2.修改默认内存大小，否则小内存机器会直接启动失败

vim /bin/startup\_nowrap.sh

将这一行改为

JAVA\_OPTS="-server -Xms512M -Xmx1024M -XX:MaxPermSize=64M -XX:+AggressiveOpts -XX:MaxDirectMemorySize=2G"

3.启动服务

bin/mycat start

4.检测启动是否成功

bin/mycat status

**8.mycat全局自增序列号配置**

1.配置server.xml

<system>

<property name="sequnceHandlerType">1</property>

</system>

2.建立自增id表

在数据库中建立一张表，存放sequence名称(name)，sequence当前值(current\_value)，步长(increment int类型每次读取多少个sequence，假设为K)等信息

原理：

Sequence获取步骤：

　　1.第一次使用该sequence时，根据传入的sequence名称，从数据库这张表中读取current\_value，和increment到MyCat中，并将数据库中的current\_value设置为原current\_value值+increment值（实现方式是基于后续的存储函数）

　　2.MyCat将读取到current\_value+increment作为本次要使用的sequence值，下次使用时，自动加1，当使用increment次后，执行步骤1)相同的操作.

MyCat负责维护这张表，用到哪些sequence，只需要在这张表中插入一条记录即可。若某次读取的sequence没有用完，系统就停掉了，则这次读取的sequence剩余值不会再使用。

-- 创建MYCAT\_SEQUENCE表

CREATE TABLE MYCAT\_SEQUENCE (

`name` VARCHAR(50) NOT NULL,

current\_value INT NOT NULL,

increment INT NOT NULL DEFAULT 1,

remark VARCHAR(100),

PRIMARY KEY(NAME)) ENGINE=INNODB;

-- – 获取当前sequence的值（返回当前值,增量）

DROP FUNCTION IF EXISTS `mycat\_seq\_currval`;

DELIMITER ;;

CREATE DEFINER=`root`@`%` FUNCTION `mycat\_seq\_currval`(seq\_name VARCHAR(50)) RETURNS VARCHAR(64) CHARSET latin1

DETERMINISTIC

BEGIN

DECLARE retval VARCHAR(64);

SET retval="-999999999,null";

SELECT CONCAT(CAST(current\_value AS CHAR),",",CAST(increment AS CHAR) ) INTO retval FROM MYCAT\_SEQUENCE WHERE NAME = seq\_name;

RETURN retval ;

END

;;

DELIMITER ;

-- 设置sequence值

DROP FUNCTION IF EXISTS `mycat\_seq\_nextval`;

DELIMITER ;;

CREATE DEFINER=`root`@`%` FUNCTION `mycat\_seq\_nextval`(seq\_name VARCHAR(50)) RETURNS VARCHAR(64) CHARSET latin1

DETERMINISTIC

BEGIN

UPDATE MYCAT\_SEQUENCE

SET current\_value = current\_value + increment WHERE NAME = seq\_name;

RETURN mycat\_seq\_currval(seq\_name);

END

;;

DELIMITER ;

-- 获取下一个sequence值

DROP FUNCTION IF EXISTS `mycat\_seq\_setval`;

DELIMITER ;;

CREATE DEFINER=`root`@`%` FUNCTION `mycat\_seq\_setval`(seq\_name VARCHAR(50), VALUE INTEGER) RETURNS VARCHAR(64) CHARSET latin1

DETERMINISTIC

BEGIN

UPDATE MYCAT\_SEQUENCE

SET current\_value = VALUE

WHERE NAME = seq\_name;

RETURN mycat\_seq\_currval(seq\_name);

END

;;

DELIMITER ;

3.插入sequence记录

INSERT INTO MYCAT\_SEQUENCE(NAME,current\_value,increment,remark) VALUES ('MYCAT', 1, 100,'测试mycat');

INSERT INTO MYCAT\_SEQUENCE(NAME,current\_value,increment,remark) VALUES ('GLOBAL', 1, 100,'全局自增id');

4.设置 sequence\_db\_conf.properties

在mycat conf目录下的sequence\_db\_conf.properties文件中添加如下内容：

MYCAT=dn3

dn3:表示我把表和函数都建在了dn3节点上

注意：DICT必须为大写

重启MyCat

# 9.Mysql主从复制

1.配置master数据库

vim /etc/my.cnf在[mysqld]下面追加如下内容

log-bin=mysql-bin

#唯一标识，一般取ip地址最后一段

server-id=1

#忽略的数据库

binlog-ignore-db=information\_schema

binlog-ignore-db=cluster

binlog-ignore-db=mysql

#需要主从复制的数据库

binlog-do-db=copy\_db

2.查看主机配置信息，需要在从机中使用

show master status;

3.配置slave数据库

vim /etc/my.cnf在[mysqld]下面追加如下内容

#和master的server-id不能一样

server-id=2

#需要主从的数据库

replicate-do-db=copy\_db

//忽略的数据库

replicate-ignore-db=mysql

重启服务配置master关联关系，进入mysql控制台执行如下命令

stop slave;

change master to master\_host='master ip',master\_port=3306,master\_user='root',master\_password='123456',master\_log\_file='mysql-bin.000004', master\_log\_pos=28125;

start slave;

master\_log\_file是在配置Master的时候的File选项， master\_log\_pos是在配置Master的Position 选项，这里要进行对应

**10.fastDfs配置**

1.安装libfatscommon,进入libfatscommon解压目录进行如下操作

make.sh

make.sh install

2.解压并进入fastdfs目录,编译安装

make.sh

make.sh install

3.配置跟踪节点

复制tracker样例配置文件，并重命名

cp /etc/fdfs/tracker.conf.sample /etc/fdfs/tracker.conf

修改tracker.conf配置文件

base\_path=/data/fastdfs/tracker

创建base\_path指定的目录

mkdir -p /data/fastdfs/tracker

启动tracker服务

/etc/init.d/fdfs\_trackerd start

设置tracker服务开机启动

chkconfig fdfs\_trakcerd on

4.配置存储节点

复制storage样例配置文件，并重命名

cp /etc/fdfs/storage.conf.sample /etc/fdfs/storage.conf

编辑配置文件，修改的内容如下

group\_name=group1 # 组名（第一组为group1，第二组为group2，依次类推...）

base\_path=/data/fastdfs/storage # 数据和日志文件存储根目录

store\_path0=/data/fastdfs/storage # 第一个存储目录，第二个存储目录起名为：store\_path1=xxx，其它存储目录名依次类推...

store\_path\_count=1 # 存储路径个数，需要和store\_path个数匹配

tracker\_server=192.168.0.200:22122 # tracker服务器IP和端口

tracker\_server=192.168.0.201:22122 # tracker服务器IP和端口

http.server\_port=88 # http访问文件的端口

创建基础数据目录

mkdir -p /data/fastdfs/storage

启动storage服务器

/etc/init.d/fdfs\_storaged start

设置storage服务开机启动

chkconfig fdfs\_storaged on

5.文件上传测试

复制并修改tracker服务器client.conf配置文件

cp /etc/fdfs/client.conf.sample /etc/fdfs/client.conf

vim /etc/fdfs/client.conf

修改内容如下

base\_path=/data/fastdfs/client

tracker\_server=120.79.185.25:22122

执行文件上传命令

/usr/bin/fdfs\_upload\_file /etc/fdfs/client.conf 文件路径

返回格式如下，则证明搭建成功

group1/M00/00/00/rBLMF1r8AO6ABN22AAAFto9GcNA36.conf

6.fastdfs-nginx-module安装

参照nginx安装配置完成nginx的安装，需要选择fastdfs-nginx-module模块

复制 fastdfs-nginx-module 源码中的配置文件到/etc/fdfs 目录，并修改

cp /usr/local/web/fastdfs-nginx-module/src/mod\_fastdfs.conf /etc/fdfs/

修改以下内容

tracker\_server=120.79.185.25:22122

group\_name=group1

url\_have\_group\_name=true

store\_path0=/data/fastdfs/storage

group\_count=1

[group1]

group\_name=group1

storage\_server\_port=23000

store\_path\_count=1

store\_path0=/data/fastdfs/storage

复制FastDFS源文件目录中HTTP相关的配置文件到/etc/fdfs目录

cd /usr/local/web/fastdfs-5.11/conf

cp http.conf mime.types /etc/fdfs/

创建数据存放目录的软链接

ln -s /data/fastdfs/storage/data/ /data/fastdfs/storage/data/M00

nginx配置文件server中修改如下内容，端口监听要和/etc/fdfs/storage.conf 中的 http.server\_port=80 相对应，可以修改

location ~/group([0-9])/M00 {

root /data/fastdfs/storage/data;

ngx\_fastdfs\_module;

}

7.负载均衡反向代理模块

参照https://blog.csdn.net/xyang81/article/details/52928230

**11.rabbitmq配置**

1.下载erlang安装包

wget -O erlang-20.3.4-1.el7.centos.x86\_64.rpm https://bintray.com/rabbitmq/rpm/download\_file?file\_path=erlang%2F20%2Fel%2F7%2Fx86\_64%2Ferlang-20.3.4-1.el7.centos.x86\_64.rpm

2.安装Erlang

yum install -y erlang-20.3.4-1.el7.centos.x86\_64.rpm

如果提示依赖冲突，卸载旧版本

rpm -e esl-erlang-20.3-1.x86\_64（冲突版本，视具体提示修改）

3.检查Erlang是否安装成功

erl -version

4.下载rabbitmq安装包

wget <https://dl.bintray.com/rabbitmq/all/rabbitmq-server/3.7.4/rabbitmq-server-3.7.4-1.el7.noarch.rpm>

5.安装rabbitmq

yum install -y rabbitmq-server-3.7.4-1.el7.noarch.rpm

6.检测rabbitmq是否安装成功

rabbitmqctl status

7.服务操作

service rabbitmq-server start #启动服务

service rabbitmq-server stop #停止服务

8.设置开机自启动

chkconfig rabbitmq-server on

9.常用命令

查看用户列表 rabbitmqctl list\_users

添加用户 rabbitmqctl add\_user 用户名 密码

删除用户 rabbitmqctl delete\_user 用户名

修改用户密码 rabbitmqctl change\_password 用户名 新密码

10.角色管理

10.1角色说明

none（普通用户）

没有控制台操作权限。

management（普通管理员）

可以查看当前用户的queues, exchanges和bindings。

可以查看和关闭当前用户的channels和connections。

可以查看当前用户的virtual hosts的统计信息。

policymaker（策略管理员）

具有management权限及查看、创建和删除当前用户的policies和parameters。

monitoring（监控管理员）

具有management权限

查看所有virtual hosts及全局的统计信息

查看所有用户的connections和channels

查看所有节点数据，如clustering和memory使用情况

administrator（超级管理员）

具有policymaker、monitoring权限

查看、创建、删除所有virtual hosts

查看、创建、删除所有users

查看、创建、删除所有permissions

可以关闭所有用户的connections

10.2查看用户角色

rabbitmqctl list\_users 用户名

10.3设置用户角色

rabbitmqctl set\_user\_tags admin 角色名称（支持同时设置多个角色）

11.权限管理

用户权限是指用户对exchange，queue的操作权限，包括配置权限，读写权限。配置权限会影响到exchange，queue的声明和删除。读写权限会影响到queue的读写消息、exchange发送消息以及queue和exchange的绑定操作。

11.1查看用户权限

rabbitmqctl list\_user\_permissions 用户名

11.2设置用户权限

rabbitmqctl set\_permissions -p 虚拟主机名称 用户名 <conf> <write> <read>

（示例如下）

rabbitmqctl set\_permissions -p / admin '.\*' '.\*' '.\*'

12.虚拟主机管理

为什么需要虚拟主机（vhost）？因为RabbitMQ只能在虚拟主机的粒度上进行权限控制。每个vhost本质上是一个mini版的RabbitMQ服务器，拥有自己的队列、交换器和绑定等。

12.1查看虚拟主机

rabbitmqctl list\_vhosts

12.2添加虚拟主机

rabbitmqctl add\_vhost 虚拟主机名称

12.3删除虚拟主机

rabbitmqctl delete\_vhost 虚拟主机名称

13.web后台管理

13.1启用后台管理插件

rabbitmq-plugins enable rabbitmq\_management

13.2登录

<http://服务器Ip:15672>

**12.jenkins配置**

1.下载jenkins.war，保存到/usr/local/jenkins目录

mkdir -R /usr/local/jenkins

wget -c -O /usr/local/jenkins/jenkins.war <http://mirrors.jenkins.io/war-stable/latest/jenkins.war>

2.启动jenkins，并后台运行(默认8080端口)

cd /usr/local/jenkins

nohup java -jar jenkins.war --httpPort=8080 >./log.log &

3.进入首页

浏览器输入服务器ip:8080端口，进入jenkins，根据提示输入页面提示的服务器指定位置的密码，验证成功后选择<安装推荐的插件>

4.插件安装成功后，创建一个管理员账户,选择保存并完成

5.确认实例配置信息，选择Save and Finish

6.服务器安装git环境

yum install curl-devel expat-devel gettext-devel openssl-devel zlib-devel gcc-c++ perl-ExtUtils-MakeMaker wget autoconf -y

cd /usr/local

wget -c -O /usr/local/git-2.9.5.tar.gz [https://mirrors.edge.kernel.org/pub/software/scm/git/git-2.9.5.tar.gz](https://mirrors.edge.kernel.org/pub/software/scm/git/git-manpages-2.9.5.tar.gz)

tar -axvf git-2.9.5.tar.gz

cd git-2.9.5

make configure

./configure --prefix=/usr/local/git

make profix=/usr/local/git

make install

echo "export PATH=\$PATH:/usr/local/git/bin" >>/etc/profile

source /etc/profile

7.服务器安装maven

cd /usr/local

wget -c <http://mirror.bit.edu.cn/apache/maven/maven-3/3.5.4/binaries/apache-maven-3.5.4-bin.tar.gz>

tar -zxvf apache-maven-3.5.4-bin.tar.gz

echo "export PATH=\$PATH:/usr/local/apache-maven-3.5.4/bin" >>/etc/profile

source /etc/profile

vim /usr/local/apache-maven-3.5.4/conf/setting.xml

找到<mirrors>标签，中间加入下列内容

<mirror>

<id>alimaven</id>

<name>aliyun maven</name>

<url>http://maven.aliyun.com/nexus/content/groups/public/</url>

<mirrorOf>central</mirrorOf>

</mirror>

8.jdk安装参考第2部分

9.jenkins配置maven、git、jdk(以下操作均在:系统管理->全局工具配置)







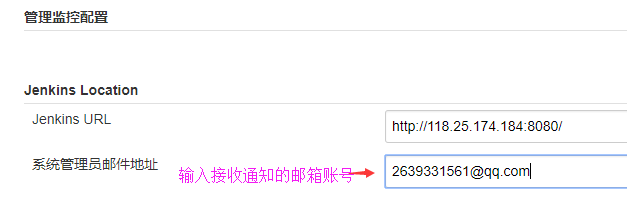
点击save，保存配置

1. 系统设置（系统管理->系统设置）

qq邮箱:设置->账户



开启IMAP/SMTP服务，安全验证成功之后会生成授权码，拷贝授权码，备用



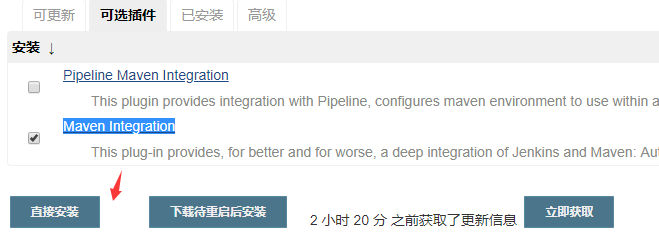


11.安装插件（系统管理->管理插件）

[Maven Integration](https://plugins.jenkins.io/maven-plugin)

SSH

可选插件选项卡中，搜索插件名称勾选后选择直接安装



勾选安装完毕后自动重启，并点击页面中显示的《打开自动刷新》

12.免密登录准备（jenkins服务器）

cd /root/.ssh/

ls -a

查看是否存在id\_rsa id\_rsa.pub文件，如果不存在则使用下面命令生成密钥

ssh-keygen //一路回车

cat id\_rsa.pub

复制公钥内容，追加到代码服务器授权文件中（代码服务器）

cd /root/.ssh

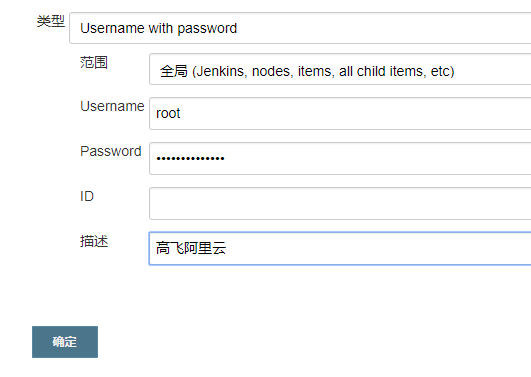
vim authorized\_keys

chmod 600 authorized\_keys

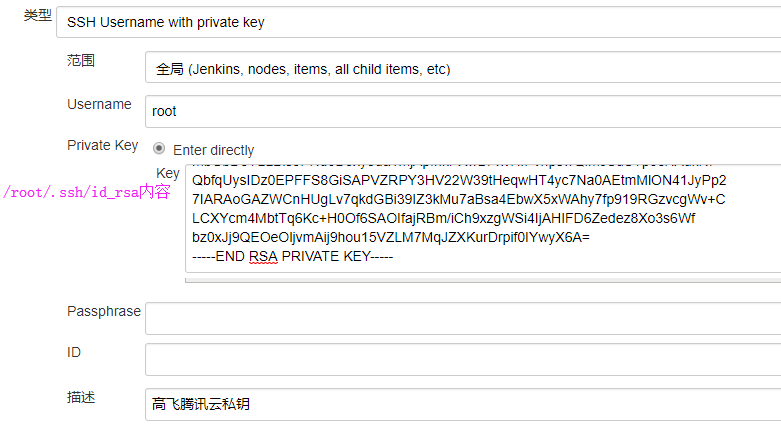
systemctl restart crond

12.添加凭据（凭据->系统->全局凭据->添加凭据）

添加一个免登录的云主机(代码服务器)



添加一个免登录的私钥凭据（jenkins服务器）



系统管理->系统设置模块设置jenkins服务器免登录SSH信息，在构建成功后，将打包后的war推送到tomcat服务器上

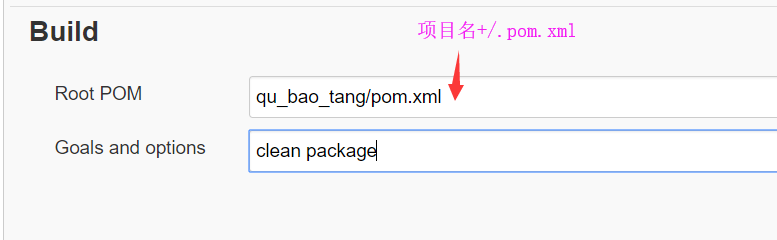


13.构建maven项目

新建一个任务，选择：构建一个maven项目









14.打包后同步到tomcat服务器（jenkins服务器和tomcat服务器都需要安装）

安装xinetd和rsync

yum install xinetd -y

yum install rsync -y

vim /etc/rsync.conf追加如下内容

read only = no

list = yes

uid = root

gid = root

#最大链接数

max connections =2

log file = /var/log/rsyncd.log

pid file = /var/run/rsyncd.pid

lock file = /var/run/rsnc.lock

启动xinetd服务

service xinetd start

使用同步命令测试文件推送到代码服务器

rsync -avz /opt/qu\_bao\_tang.war [root@120.79.95.51:/data/jenkins.war](mailto:root@120.79.95.51:/data/jenkins.war)

1. **数据库定时备份**

1.编写mysql导出数据脚本

vim /data/shell/backup\_sql.sh

追加如下内容（sql文件保存目录要提前创建）

#!/bin/sh

time=$(date "+%Y\_%m\_%d\_%H\_%M\_%S")

dir=/data/backup/ai\_kang\_shui\_dian/sql/

if [ ! -d $dir ];

then

mkdir -p $dir

fi

/usr/local/web/mysql-8.0/bin/mysqldump -uroot -p3396Zmr3Wj0VoflC -h127.0.0.1 -P40036 ai\_kang\_shui\_dian > ${dir}ai\_kang\_shui\_dian\_${time}.sql

2.添加linux定时任务

执行crontab -e，追加如下内容（每天凌晨三点启动备份脚本）

00 03 \* \* \* /data/shell/backup\_sql.sh

1. **文件定时同步**

1.主机从机分别安装rsync服务

yum install rsync

service rsyncd start

2.从机生成ssh免登录公钥

ssh-keygen -t rsa

3.将公钥发送到主机服务器

ssh-copy-id -i ~/.ssh/id\_rsa.pub root@主机ip

输入yes，之后验证主机密码

4.编写备份脚本

vim /data/shell/backup\_rsync.sh

追加如下内容

#!/bin/sh

rsync -zau --progress root@主机ip:/data/ai\_kang\_shui\_dian/ /data/ai\_kang\_shui\_dian/ -e "ssh -p 22"