# Zabbix企业级实验环境安装部署

## 准备工作

### 1.1 硬件和软件需求

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 应用名称 | 操作系统 | CPU | 内存 | 硬盘 |
| Zabbix-Server | Centos7 |  | 4G | 1T |
| Zabbix-DB | Centos7 |  | 4G | 1T |
| Zabbix-GUI | Centos7 |  | 4G | 500G |
| Zabbix-Proxy | Centos7 |  | 4G | 1T |

### 1.2 IP地址规划

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| zabbix监控环境 | **主机名hostname** | **IP地址** | | **用途** | **备注** |
| **物理IP** | **虚拟IP** |
| **ieds-1** | 192.168.0.2 | 无 | zabbix-DB（主） |  |
| **ieds-2** | 192.168.0.3 | zabbix-DB（从） |  |
| **ieds-3** | 192.168.0.4 | 192.168.0.6 | zabbix-Server（主） |  |
| **ieds-4** | 192.168.0.5 | zabbix-Server（从） |  |
| **ieds-5** | 192.168.0.7 | 无 | zabbix-GUI |  |
| **ieds-6** | 192.168.0.8 | 无 | zabbix-proxy |  |
| **ieds-7** | 192.168.0.9 | 无 | zabbix-proxy |  |
| zabbix被监控环境 | **ieds-8** |  |  |  |  |
| **ieds-9** |  |  |  |  |
| **ieds-10** |  |  |  |  |
| **ieds-11** |  |  |  |  |

### 1.3 网络配置

**ieds-1**

#vi /etc/sysconfig/network-scripts/ifcfg-enp3s0

BOOTPROTO=dhcp

IPADDR=192.168.0.2

ONBOOT=yes

#systemctl restart network.service

### 1.4 主机名配置（每台）

#vi /etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

192.168.0.2 db-master

192.168.0.3 db-slave

192.168.0.4 server-master

192.168.0.5 server-slaver

192.168.0.6 server #vip 虚拟IP

192.168.0.7 zabbix-gui

192.168.0.8 zabbix-proxy1

### 1.5 关闭防火墙及安全策略

#setenforce 0

#getenforce

#vi /etc/selinux/config

selinux=disabled

#systemctl stop firewalld.service

#systemctl disabled firewalld.service

## 2.数据库percona

### 2.1 安装步骤

**下载文件**

#tar -zxvf Percona-Server-5.7.14-8-Linux.x86\_64.ssl101.tar.gz

#mv Percona-Server-5.7.14-8-Linux.x86\_64.ssl101 /usr/local/mysql

#cd /usr/local/mysql/

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

**建立用户**

#groupadd -g 27 mysql

#useradd -g 27 -s /sbin/nologin mysql

**修改权限**

#chown -R mysql.mysql /usr/local/mysql/

**配置环境变量**

#vim ~/.bash\_profile

PATH=$PATH:$HOME/bin:/usr/local/mysql/bin

**启动percona-server**

注意不能存在文件/etc/my.cnf，否则，由于my.cnf里的不正确配置而导致mysql不能正常启动，因为mysqld脚本里面默认路径会去找/etc/my.cnf这个文件。

#mysql --verbose --help|grep my.cnf

my.cnf将会存在于以下路径，依次为优先级匹配。

/etc/my.cnf

/etc/mysql/my.cnf

/usr/local/mysql/etc/my.cnf

~/.my.cnf

但在测试的时候，发现并未读取

/usr/local/mysql/etc/my.cnf

**配置my.cnf文件**

# vi /etc/my.cnf

[mysqld]

datadir=/usr/local/mysql/data

socket=/usr/local/mysql/var/run/mysql.sock

user=mysql

# Disabling symbolic-links is recommended to prevent assorted security risks

symbolic-links=0

character-set-server=utf8

innodb\_file\_per\_table=1

[mysqld\_safe]

log-error=/usr/local/mysql/var/log/mysqld.log

pid-file=/usr/local/mysql/var/run/mysqld/mysqld.pid

**创建目录**

#mkdir -p /usr/local/mysql/var/run

#mkdir -p /usr/local/mysql/var/log

#mkdir -p /usr/local/mysql/etc

**复制配置文件**

#cp /etc/my.cnf /usr/local/mysql/etc/

**修改权限**

chown -R mysql.mysql /usr/local/mysql/

**初始化mysql**

#./bin/mysql\_install\_db \

--user=mysql \

--basedir=/usr/local/mysql/ \

--datadir=/usr/local/mysql/data/

#./bin/mysqld\_safe &

**改变权限**

#chown -R mysql.mysql /usr/local/mysql/

#chkconfig mysqld on

#/etc/init.d/mysqld start

### 2.2 创建数据库

#MysqlPassword=admin

#mysqladmin -u root password ${MysqlPassword}

Mysql>create database zabbix character set utf8;

Mysql>grant all privileges on zabbix.\* to zabbix@'localhost' identified by 'zabbix';

Mysql>grant all privileges on zabbix.\* to zabbix@'localhost' identified by 'zabbix';

### 2.3 安装zabbix-mysql

#rpm -ivh zabbix-mysql-3.2.0-1.el7.x86\_64

### 2.4 导入zabbix-server 的数据库

#mysql -uzabbix -pzabbix zabbix< /root/zabbix-3.2.0/database/mysql/schema.sql

#mysql -uzabbix -pzabbix zabbix< /root/zabbix-3.2.0/database/mysql/images.sql

#mysql -uzabbix -pzabbix zabbix< /root/zabbix-3.2.0/database/mysql/data.sql

### 2.5 MySQL主从的配置

**在MySQL主服务器192.168.0.2上配置**

1. 修改my.cnf文件

#vim /usr/local/Percona-Server-5.7.14-8-Linux.x86\_64.ssl101/etc/my.cnf

[mysqld]

......

server-id=1

log-bin=mysql-bin

log-bin=/usr/local/Percona-Server-5.7.14-8-Linux.x86\_64.ssl101/var/log/update

在MySQL命令行执行以下命令，显示主节点状态：

Mysql>show master status;

1. 建立复制权限的用户

#mysql -uroot -p

Mysql>grant replication client,replication slave on \*.\* to [repl@’192.168.0.3.%’](mailto:repl@’192.168.0.3.%’) identified by ‘zabbix\_repl’;

Mysql>flush privileges;

1. 备份数据库

#mysqldump -uzabbix -pzabbix >zabbix.sql

说明：备份完成后复制到从库，并停止对主库的写操作

**在MySQL从服务器192.168.0.3上配置**

1. 修改my.cnf文件

[mysqld]

......

server-id=2 #1,表示master；2，表示slave

log-bin=mysql-bin

1. 导入master中的数据库

#mysql -uzabbix -pzabbix <zabbix.sql

1. 配置复制权限

Mysql>change master to master\_host=’192.168.0.2’,MASTER\_USER=’repl’,MASTER\_PASSWORD=’zabbix\_repl’,MASTER\_PORT=3306,MASTER\_LOG\_FILE=’update.000002’,MASTER\_LOG\_POS=107,MASTER\_CONNECT\_RETRY=10;

说明：107、update.000002的数值要与master的position和file一致。

1. 启动MySQL slave

Mysql>start slave;

Mysql>show slave status\G;

## 3.zabbix-server

### 3.1 安装

**3.1.1 RPM安装**

#rpm -ivh zabbix-server-mysql-3.2.0-1.el7.x86\_64.rpm

**3.1.2源码安装**

#tar -zxvf zabbix-3.2.0.tar.gz

创建组和用户

#groupadd zabbix

#useradd -g zabbix zabbix

安装

#cd zabbix-3.2.0

#./configure --prefix=/usr/local/zabbix --enable-server --enable-agent --with-mysql --enable-ipv6 --with-net-snmp --with-libcurl --with-libxml2

#make && make install

### 3.2 导入数据库

#cd /usr/share/doc/zabbix-mysql-3.2.0/

#chkconfig mysql-server on

#service mysql-server start

#mysqladmin -uroot password ‘admin’;

#myql -uroot -padmin

Mysql> create database zabbix character set utf8;

Mysql>grant all privileges on zabbix.\* to zabbix@localhost identified by 'zabbix';

Mysql>flush privileges;

#mysql -uzabbix -pzabbix zabbix

#mysql -uzabbix -pzabbix zabbix < /usr/share/doc/zabbix-3.2.0/create.sql

#mkdir /var/log/zabbix

#chown zabbix.zabbix /var/log/zabbix

### 3.3 复制service启动脚本

#cp misc/init.d/fedora/core/zabbix\_\* /etc/init.d/

#chmod 755 /etc/init.d/zabbix\_\*

#sed -i “s#BASEDIR=/usr/local#BASEDIR=/usr/#g” /etc/init.d/zabbix\_server

#sed -i “s#BASEDIR=/usr/local#BASEDIR=/usr/#g” /etc/init.d/zabbix\_agentd

### 3.4 修改配置文件

#vi /usr/local/zabbix/etc/zabbix\_server.conf

DBHost=192.168.0.2 #主数据库IP地址

DBName=zabbix

DBUser=zabbix

DBPassword=zabbix

DBSocket=/usr/local/mysql/var/run/mysql.sock #与主数据库的mysql.sock路径一致

DBPort=3306 #默认端口

### 3.5 启动zabbix-server服务

#systemctl start zabbix-server.service

#systemctl enable zabbix-server.service

### 3.6 keepalived 安装

**3.6.1 安装**

#tar –zxvf keepalived-1.2.24.tar.gz

#cd keepalived-1.2.24

#./configure

#make && make install

**3.6.2 修改配置文件**

#vi /etc/keepalived/keepalived.conf

! Configuration File for keepalived

global\_defs {

notification\_email {

dba.gao@gmail.com

ixdba@163.com

}

notification\_email\_from Keepalived@localhost

smtp\_server 192.168.200.1

smtp\_connect\_timeout 30

router\_id server-master #hostname

}

vrrp\_script chk\_zabbix\_server{

script “/etc/keepalived/chk\_zabbix\_server.sh”

interval 30

weight 2

}

vrrp\_instance VI\_1 {

state MASTER #从主机配置为BACKUP

interface enp3s0

virtual\_router\_id 51

priority 111 #从主机的权重小于111

advert\_int 1

mcast\_src\_ip <IPADDR>

garp\_master\_delay 10

authentication {

auth\_type PASS

auth\_pass 1111

}

virtual\_ipaddress {

#<IPADDR>/<MASK> brd <IPADDR> dev <STRING> scope <SCOPT> label <LABEL>

192.168.0.6

}

mcast\_src\_ip 192.168.0.4

track\_script {

chk\_zabbix\_server

}

}

**3.6.3 keepalived 检测脚本**

[root@server-master]#vi /etc/keepalived/chk\_zabbix\_server.sh

#! /bin/bash

#

#

status1 = $(ps aux|grep “/usr/sbin/zabbix\_server” | grep -v grep | grep -v bash |wc -l)

if [ “${status1}” = “0” ]; then

/etc/init.d/zabbix-server start

sleep 3

status2 = $(ps aux|grep zabbix\_server |grep -v grep |grep -v bash |wc -l)

if [ “$status2” = “0” ];then

/etc/init.d/keepalived stop

fi

fi

**3.6.4 启动keepalived服务**

#systecmtl start keepalived.service

#systemctl enable keepalived.service

## zabbix-GUI

### 4.1 NGINX

安装nginx 源码安装，解压，进入解压包

#./configure

#make&make install

验证nginx 进入/usr/local/nginx/sbin 运行

#./nginx -t

安装php-fpm包

安装zabbix-web需要解决php-bcmath,php-ldap,php-mbstring

rpm -ivh php-bcmath-5.4.16-36.el7\_1.x86\_64.rpm

rpm -ivh php-bcmath-5.4.16-36.el7\_1.x86\_64.rpm

rpm -ivh php-mbstring-5.4.16-36.el7\_1.x86\_64.rpm

安装zabbix-web-mysql需要安装php-mysql依赖包

rpm -ivh php-mysql-5.4.16-36.el7\_1.x86\_64.rpm

安装zabbix-web与zabbix-web-mysql一同安装

以上安装方式为rpm安装，若需要源码安装可按如下操作

下载zabbix源码安装包 zabbix-3.2.0.tar.gz

解压tar -zxvf zabbix-3.2.0.tar.gz

进入目录 cd zabbix-3.2.0/fronteds/

该目录下包含所有前端的界面，代码为php缩写

根据web服务选择不同再分别相应配置（以apache为例）

安装apache

复制网页文件到apache目录

cp -r /zabbix-3.2.0/fronteds/php/ /var/www/html/zabbix

chown -R apache.apache /var/www/html/zabbix

开启zabbix服务

开启apache服务

配置php以及php-fpm,编辑修改 (可按照相关参数说明进行配置)

如下为php相关参数修改配置：/etc/php.ini

date.timezone=Asia/Shanghai

max\_execution\_time=300

post\_max\_size=16M

max\_input\_time=300

memory\_limit=128M

mbstring.func\_overload=2(web界面检测会报错，推荐使用默认值)

如下为php-fpm相关的参数配置： /etc/php-fpm.d/www.conf

listen=/var/run/php-fpm/php-fpm.sock

listen.allowed\_clients = 127.0.0.1

listen.owner=root

listen.group=root

**重启php-fpm服务**

**安装nginx**

下载nginx源码包，本实验中下载nginx-1.10.1.tar.gz 稳定版本

# tar -zxvf nginx-1.10.1.tar.gz

# cd nginx-1.10.1/

#./configure --prefix=/usr/local/nginx (若提示有依赖则需要先解决依赖)

#make && make install

安装完毕后会有如下信息内容，记录并保存，提供后续服务启动项设置配置信息

+ using system OpenSSL library

+ md5: using OpenSSL library

+ sha1: using OpenSSL library

+ using system zlib library

nginx path prefix: "/usr/local/nginx"

nginx binary file: "/usr/local/nginx/sbin/nginx"

nginx modules path: "/usr/local/nginx/modules"

nginx configuration prefix: "/usr/local/nginx/conf"

nginx configuration file: "/usr/local/nginx/conf/nginx.conf"

nginx pid file: "/usr/local/nginx/logs/nginx.pid"

nginx error log file: "/usr/local/nginx/logs/error.log"

nginx http access log file: "/usr/local/nginx/logs/access.log"

nginx http client request body temporary files: "client\_body\_temp"

nginx http proxy temporary files: "proxy\_temp"

nginx http fastcgi temporary files: "fastcgi\_temp"

nginx http uwsgi temporary files: "uwsgi\_temp"

nginx http scgi temporary files: "scgi\_temp"

设置nginx配置文件 /usr/local/nginx/conf/nginx.conf

worker\_processes 1;

events {

worker\_connections 1024;

}

http {

include mime.types;

default\_type application/octet-stream;

sendfile on;

keepalive\_timeout 65;

gzip on;

server {

listen 80;

server\_name localhost;

charset utf8;

access\_log /var/log/nginx/zabbix.access.log;

error\_log /var/log/nginx/zabbix.error.log;

root /usr/share/zabbix;

location / {

proxy\_buffer\_size 64k;

proxy\_buffers 32 32k;

proxy\_busy\_buffers\_size 128k;

index index.html index.htm index.php;

}

location ~ .\*\.(php|php5)?$ {

proxy\_buffer\_size 64k;

proxy\_buffers 8 64k;

fastcgi\_buffer\_size 128k;

fastcgi\_buffers 4 128k;

fastcgi\_pass unix:/var/run/php-fpm/php-fpm.sock;

fastcgi\_index index.php;

fastcgi\_param

SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;

include fastcgi\_params;

}

}

}

配置nginx开机自启动

**在/etc/init.d/目录下创建脚本**

#vi /etc/init.d/nginxd

**更改脚本权限**

#chmod 775 /etc/init.d/nginxd

**编写脚本内容**

#!/bin/sh

#Startup script for Nginx - this script starts and stops the nginx daemon

#

# chkconfig: - 85 15

# description: Nginx is an HTTP(S) server, HTTP(S) reverse proxy and IMAP/POP3 proxy server

# processname: nginx

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

# pidfile: /usr/local/nginx/logs/nginx.pid

# Source function library.

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

# Source networking configuration.

. /etc/sysconfig/network

# Check that networking is up.

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

nginx="/usr/local/nginx/sbin/nginx"

prog=$(basename $nginx)

NGINX\_CONF\_FILE="/usr/local/nginx/conf/nginx.conf"

[ -f /etc/sysconfig/nginx ] && . /etc/sysconfig/nginx

lockfile=/var/lock/subsys/nginx

start() {

[ -x $nginx ] || exit 5

[ -f $NGINX\_CONF\_FILE ] || exit 6

echo -n $"Starting $prog: "

daemon $nginx -c $NGINX\_CONF\_FILE

retval=$?

echo

[ $retval -eq 0 ] && touch $lockfile

return $retval

}

stop() {

echo -n $"Stopping $prog: "

killproc $prog -QUIT

retval=$?

echo

[ $retval -eq 0 ] && rm -f $lockfile

return $retval

}

restart() {

configtest || return $?

stop

sleep 1

start

}

reload() {

configtest || return $?

echo -n $"Reloading $prog: "

killproc $nginx -HUP

RETVAL=$?

echo

}

force\_reload() {

restart

}

configtest() {

$nginx -t -c $NGINX\_CONF\_FILE

}

rh\_status() {

status $prog

}

rh\_status\_q() {

rh\_status >/dev/null 2>&1

}

case "$1" in

start)

rh\_status\_q && exit 0

$1

;;

stop)

rh\_status\_q || exit 0

$1

;;

restart|configtest)

$1

;;

reload)

rh\_status\_q || exit 7

$1

;;

force-reload)

force\_reload

;;

status)

rh\_status

;;

condrestart|try-restart)

rh\_status\_q || exit 0

;;

\*)

echo $"Usage: $0 {start|stop|status|restart|condrestart|try-restart|reload|force-reload|configtest}"

exit 2

esac

**设置开机启动**

chkconfig --add nginxd

chkconfig nginxd on

service nginxd start

service nginxd restart

service nginxd status

### 4.2访问页面导向

浏览器中输入http://<server\_ip\_or\_name>/index.php

1.页面输出检测php环境是否正确

2.配置数据库相关信息（IP（数据库），端口，数据库名，用户，密码）

3.配置zabbix-server信息（IP（服务端）,端口，服务名标识）

设置成功后的zabbix-gui相关配置参数会保存在

/etc/zabbix/web/zabbix.conf.php 内容如下

<?php

// Zabbix GUI configuration file.

global $DB;

$DB['TYPE'] = 'MYSQL';

$DB['SERVER'] = '192.168.0.2';

$DB['PORT'] = '3306';

$DB['DATABASE'] = 'zabbix';

$DB['USER'] = 'zabbix';

$DB['PASSWORD'] = 'zabbix';

// Schema name. Used for IBM DB2 and PostgreSQL.

$DB['SCHEMA'] = '';

$ZBX\_SERVER = '192.168.0.6';

$ZBX\_SERVER\_PORT = '10051';

$ZBX\_SERVER\_NAME = 'server';

$IMAGE\_FORMAT\_DEFAULT = IMAGE\_FORMAT\_PNG;

## 5.zabbix-Agent

### 5.1 centos7版

**添加用户群组**

#groupadd zabbix

#useradd -g zabbix -m zabbix

**解压安装包，编译安装**

#tar -zxvf zabbix-3.2.0.tar.gz

#cd zabbix-3.2.0

#./configure --prefix=/usr/local/zabbix --enable-agent

# make && make install

**修改agent配置文件**

#vi /usr/local/zabbix/etc/zabbix\_agentd.conf

Server= ip.ip.ip.ip #服务端ip地址

ServerActive= ip.ip.ip.ip #服务端ip地址

Hostname=client1 #必须与创建主机时的hostname一致

**修改启动配置，添加软链接**

#cp misc/init.d/tru64/zabbix\_agentd /etc/init.d/

# chmod +x /etc/init.d/zabbix\_agentd

# ln -s /usr/local/zabbix/sbin/\* /usr/local/sbin/

# ln -s /usr/local/zabbix/bin/\* /usr/local/bin/

**设置自启动**

# vi /etc/rc.d/init.d/zabbix\_agentd

#在第二行添加如下内容

#chkconfig: 2345 95 95

#description: zabbix agent 保存后退出文件

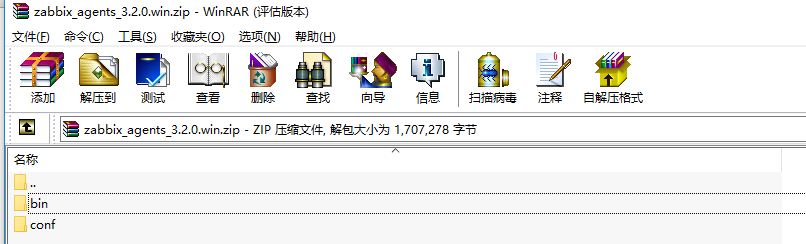
#chkconfig --add zabbix\_agentd

#chkconfig zabbix\_agentd on

#systemctl restart zabbix\_agentd

### 5.2 win7版

**下载Windows的zabbix\_agentd** http://www.zabbix.com/download.php 选择windows版本的agent下载  
  
  
  
从官方下载Zabbix Agent后，压缩包里面有2个目录，bin和conf，conf里面的配置文件是通用的。

  
bin文件夹里包含两个文件夹，一个为win32，另一个是win64，每个目录下应该有3个文件，分别 为：

  
   
根据操作系统不同，选择不同文件放在c:\zabbix目录下  
   
**添加zabbix\_agentd.win.conf文件**   
在c:\zabbix文件夹中新建zabbix\_agentd.win.conf文件，修改文件名为zabbix\_agentd.conf

配置如下内容：

Server=zabbix server的ip地址

ServerActive=zabbix server的ip地址

Hostname=本机ip或本机名

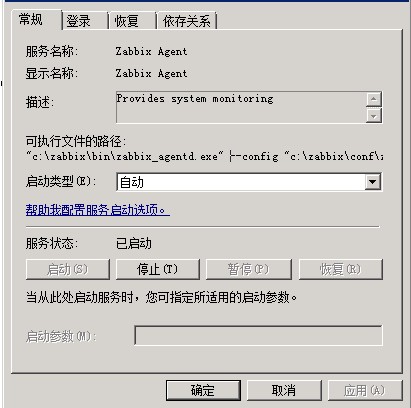
LogFile=日志存放路径

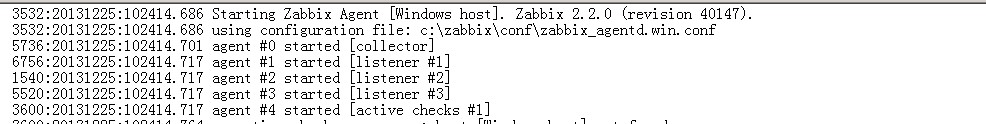
修改完成之后运行cmd命令  
  
c:\zabbix\zabbix\_agentd.exe -c c:\zabbix\zabbix\_agentd.conf -i

执行如下命令，启动客户端：  
   
c:\zabbix>zabbix\_agentd.exe -c c:\zabbix\zabbix\_agentd.conf -s

    
说明：zabbix\_agent命令参数含义：  
-c 制定配置文件所在位置；-i 是安装客户端；-s 启动客户端；-x 停止客户端；-d 卸载客户端

**查看服务并启动**  
  
到"服务"里可以看到zabbix\_agentd.exe已经添加成系统服务，启动类型为自动。



ok，agents正常运行之后，在安装文件内会看到日志文件  
  


## MYSQL监控

### 6.1 配置mysql脚本文件

给mysql赋权限：

> grant usage on \*.\* to zabbix@'localhost' identified by 'zabbix';

> flush privileges

进入zabbix解压文件的conf/zabbix\_agentd目录下

#cp userparameter\_mysql.conf /usr/local/zabbix/etc/zabbix\_agentd.conf.d/

#vi /usr/local/zabbix/etc/zabbix\_agentd.conf.d/userparameter\_mysql.conf

UserParameter=mysql.status[\*],echo "show global status where Variable\_name='$1';" | HOME=/usr/local/zabbix/etc /usr/local/Percona-Server-5.7.14-8-Linux.x86\_64.ssl101/bin/mysql -N | awk '{print $$2}'

UserParameter=mysql.size[\*],bash -c 'echo "select sum($(case "$3" in both|"") echo "data\_length+index\_length";; data|index) echo "$3\_length";; free) echo "data\_free";; esac)) from information\_schema.tables$([[ "$1" = "all" || ! "$1" ]] || echo " where table\_schema=\"$1\"")$([[ "$2" = "all" || ! "$2" ]] || echo "and table\_name=\"$2\"");" | HOME=/usr/local/zabbix/etc /usr/local/Percona-Server-5.7.14-8-Linux.x86\_64.ssl101/bin/mysql -N'

UserParameter=mysql.ping,HOME=/usr/local/zabbix/etc /usr/local/Percona-Server-5.7.14-8-Linux.x86\_64.ssl101/bin/mysqladmin ping | grep -c alive

UserParameter=mysql.version,mysql -V

说明：需要修改文件中的HOME路径为存放.my.cnf文件的目录，yum安装为/etc/zabbix并mysql命令路径

### 6.2 添加.my.cnf文件

#cd /usr/local/zabbix/etc

# vi .my.cnf

[mysql]

host=localhost

user=zabbix

password=zabbix

socket=/var/lib/mysql/mysql.sock

[mysqladmin]

host=localhost

user=zabbix

password=zabbix

socket=/var/lib/mysql/mysql.sock

### 6.3 修改zabbix\_agentd.conf

#vi zabbix\_agentd.conf

UnsafeUserParameters=1

Include=/usr/local/zabbix/etc/zabbix\_agentd.conf.d/\*.conf

### 6.4 重启agent服务

#systemctl restart zabbix\_agentd

# systemctl status zabbix\_agentd -l

## ORACLE监控

### 7.1 准备工作

**zabbix-server 安装JDK1.8**

#java -version 检查当前系统JDK版本

#rpm -qa |grep java 查找java相关程序

#yum -y remove java-1.7.0-openjdk-headless-1.7.0.91-2.6.2.3.el7.x86\_64

卸载 直到查看版本，显示如图



# mkdir -p /usr/local/java

# tar -zxvf jdk-8u101-linux-x64.tar.gz -C /usr/local/java

# vi /etc/profile

export JAVA\_HOME=/usr/local/java/jdk1.8.0\_101

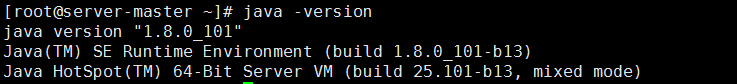
export JRE\_HOME=${JAVA\_HOME}/jre

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

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

# source /etc/profile

# java -version



安装orabbix

# mkdir -p /opt/orabbix 解压orabbix到该目录

# unzip orabbix-1.2.3.zip

# chmod -R a+x /opt/orabbix/ 赋权限

#cp /opt/orabbix/conf/config.props.sample /opt/orabbix/conf/config.props

#vim conf/config.props 编辑配置文件 参照默认配置修改

配置文件内容：

# cat /opt/orabbix/conf/config.props |grep -v "#"

ZabbixServerList=server-master #zabbix名称

server-master.Address=192.168.0.4 #zabbix服务端的IP地址

server-master.Port=10051 #zabbix服务端的监控端口

OrabbixDaemon.PidFile=./logs/orabbix.pid

OrabbixDaemon.Sleep=300

OrabbixDaemon.MaxThreadNumber=100

DatabaseList= ieds-9 #这个名称可以随便起，但是必须跟web监控、agent的主机名保持一致。

DatabaseList.MaxActive=10

DatabaseList.MaxWait=100

DatabaseList.MaxIdle=1

ieds-9.Url=jdbc:oracle:thin:@192.168.0.11:1521:orcl

#这里主要是通过JDBC来连接客户端的。orcl是数据库的实例名称。1521是监听端口

ieds-9.User=zabbix #数据库用户zabbix

ieds-9.Password=zabbix #数据库用户密码

ieds-9.MaxActive=10

ieds-9.MaxWait=100

ieds-9.MaxIdle=1

ieds-9.QueryListFile=./conf/query.props

#cp /opt/orabbix/init.d/orabbix /etc/init.d/orabbix #配置服务

# chkconfig --add orabbix

# chkconfig orabbix on

# service orabbix start 启动服务



### 7.2 设置oracle监控权限

Oracle服务器上设置监控用户及权限

#sqlplus / as sysdba

Sql>create user zabbix identified by zabbix default tablespace system temporary tablespace temp profile default account unlock;

GRANT ALTER SESSION TO zabbix;

GRANT CREATE SESSION TO zabbix;

GRANT CONNECT TO zabbix;

ALTER USER zabbix DEFAULT ROLE ALL;

GRANT SELECT ON V\_$INSTANCE TO zabbix;

GRANT SELECT ON DBA\_USERS TO zabbix;

GRANT SELECT ON V\_$LOG\_HISTORY TO zabbix;

GRANT SELECT ON V\_$LOG TO zabbix;

GRANT SELECT ON V\_$PARAMETER TO zabbix;

GRANT SELECT ON SYS.DBA\_AUDIT\_SESSION TO zabbix;

GRANT SELECT ON V\_$LOCK TO zabbix;

GRANT SELECT ON DBA\_REGISTRY TO zabbix;

GRANT SELECT ON V\_$LIBRARY CACHE TO zabbix;

GRANT SELECT ON V\_$SYSSTAT TO zabbix;

GRANT SELECT ON V\_$PARAMETER TO zabbix;

GRANT SELECT ON V\_$LATCH TO zabbix;

GRANT SELECT ON V\_$PGASTAT TO zabbix;

GRANT SELECT ON V\_$SGASTAT TO zabbix;

GRANT SELECT ON V\_$LIBRARY CACHE TO zabbix;

GRANT SELECT ON V\_$PROCESS TO zabbix;

GRANT SELECT ON DBA\_DATA\_FILES TO zabbix;

GRANT SELECT ON DBA\_TEMP\_FILES TO zabbix;

GRANT SELECT ON DBA\_FREE\_SPACE TO zabbix;

GRANT SELECT ON V\_$SYSTEM\_EVENT TO zabbix;

GRANT SELECT ON V\_$locked\_object TO zabbix;

GRANT SELECT ON dba\_objects TO zabbix;

GRANT SELECT ON dba\_tablespaces TO zabbix;

GRANT SELECT ON v\_$session TO zabbix;

说明：如果数据库是oracle11g，则需执行如下操作：

Sql>exec dbms\_network\_acl\_admin.create\_acl(acl => 'resolve.xml',description=> 'resolve acl', principal =>'ZABBIX', is\_grant => true, privilege => 'resolve');

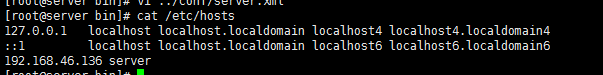
Sql>exec dbms\_network\_acl\_admin.assign\_acl(acl => 'resolve.xml', host =>'\*');

Sql>commit;

## tomcat监控

### 8.1前提条件

需要配置tomcat所在的主机中/etc/hosts文件配置



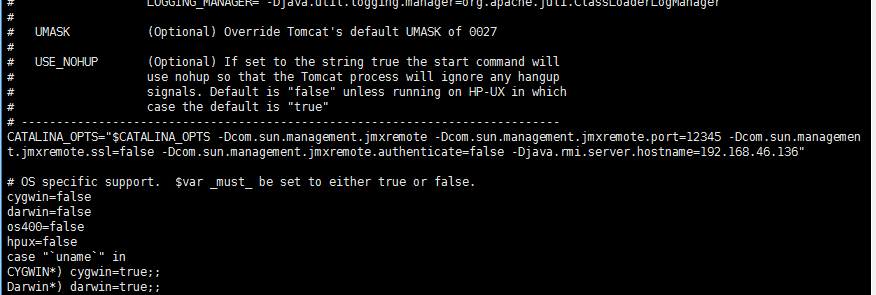
用到的文件：catalina-jmx-remote.jar cmdline-jmxclient-0.10.3.jar

### 8.2 Tomcat-Server被控端配置

   # cd /usr/java/apache-tomcat-8.0.30/bin

   # vim catalina.sh

添加如下内容：

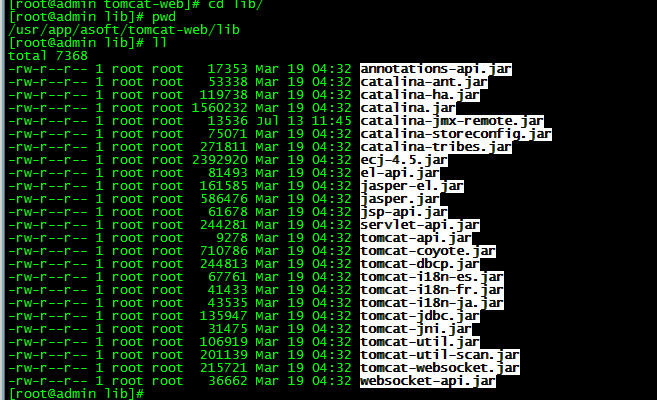


CATALINA\_OPTS="-Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.management.jmxremote.port=**12345** -Dcom.sun.management.jmxremote.ssl=false -Djava.rmi.server.hostname=**Tomcat-Server IP**"

注：1.**Tomcat-Server IP**是[客户端](http://www.07net01.com/tags-客户端-0.html" \t "_blank)(被控端) ip地址，必须写可被外部访问的IP不能是[localhost](http://www.07net01.com/tags-localhost-0.html" \t "_blank)和127.0.0.1。

2.把配置信息放到文件开头处，如图所示位置，不能放到最后，会出现打开不了端口。

3.12345是默认端口，无特殊要求尽量不要改，如修改[服务端](http://www.07net01.com/tags-服务端-0.html" \t "_blank)也同步修改。

 将这个文件catalina-jmx-remote.jar放至 

 # cd /usr/java/apache-tomcat-8.0.30/conf

   # vim server.xml

添加如下内容

<Listener className="org.apache.catalina.mbeans.JmxRemoteLifecycleListener"  rmiRegistryPortPlatform="12345" rmiServerPortPlatform="12346"/>

重启tomcat

# sh /usr/java/apache-tomcat-8.0.30/bin/shutdown.sh

# sh /usr/java/apache-tomcat-8.0.30/bin/startup.sh

# netstat -an | grep 12345

显示如下信息即为成功。

tcp      0     0    0.0.0.0:12345     0.0.0.0:\*        LISTEN

验证是否成功，可以在被控端测试。命令如下：

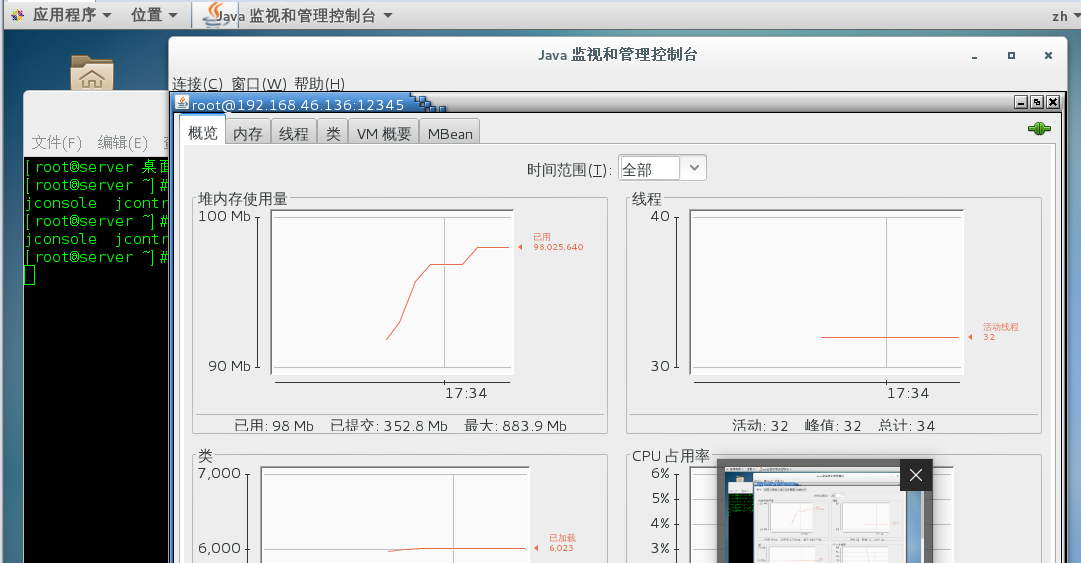
java -jar cmdline-jmxclient-0.10.3.jar - 192.168.46.136:12345 java.lang:type=ClassLoading TotalLoadedClassCount

如图显示则成功了。



通过jconsole验证



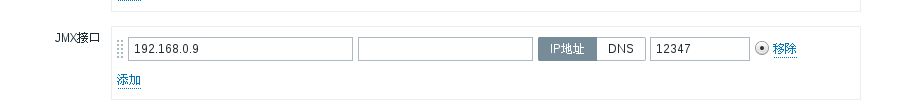


### 8.3 zabbix-gui配置

开始配置监控的主机信息未添加任何模板



配置jmx接口信息

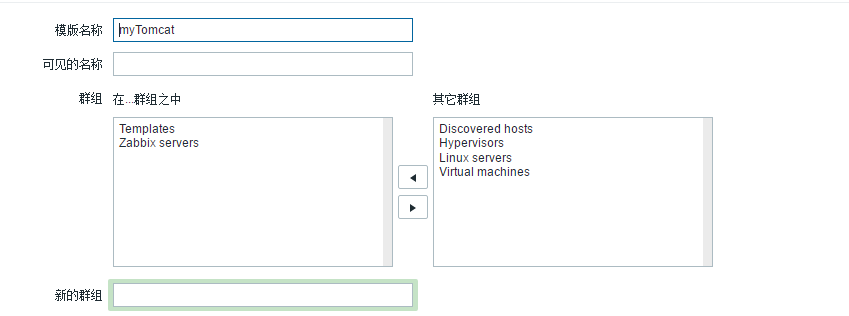


1. 模板配置

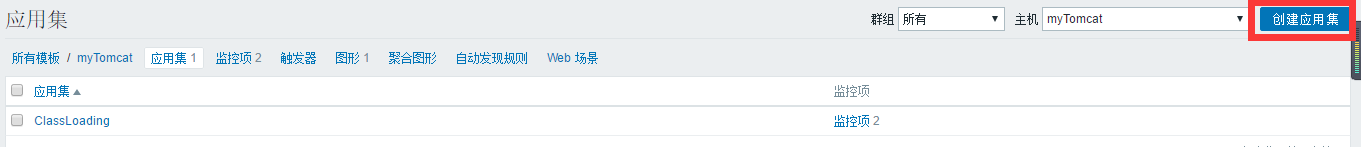
模板可以选择自带模板，或可以自定义模板

自定义模板





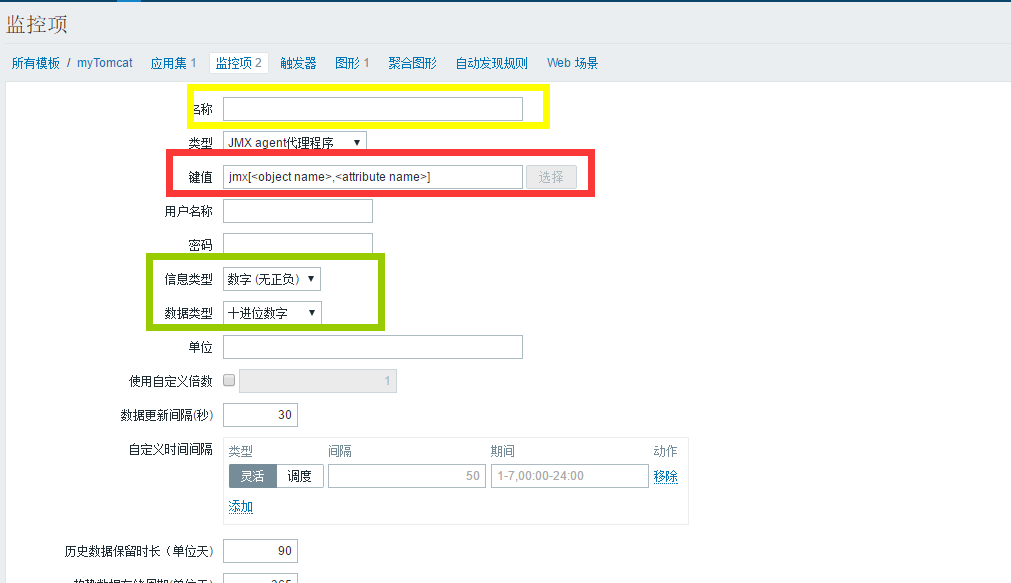
定义应用集





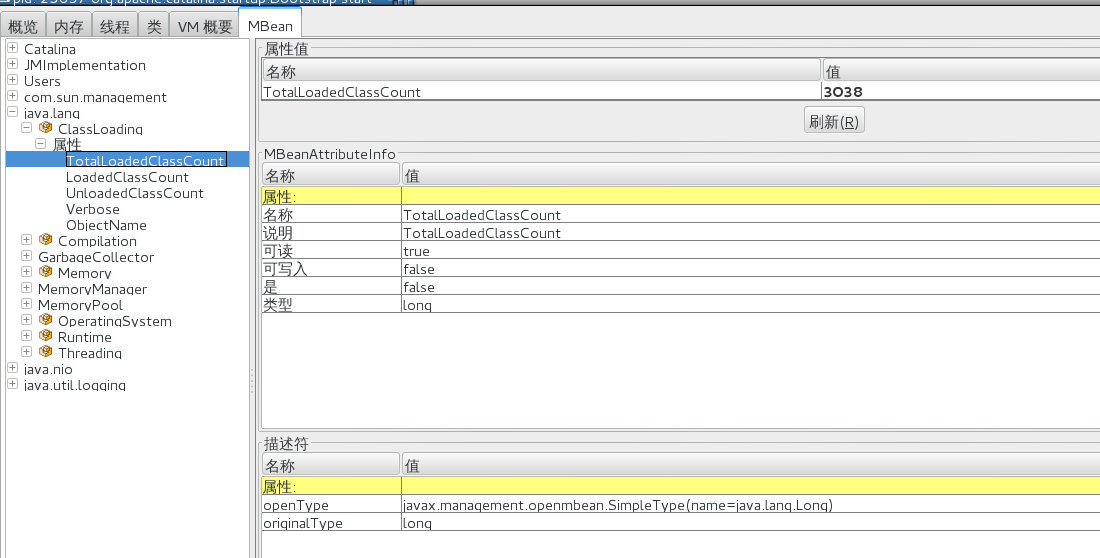
定义监控项





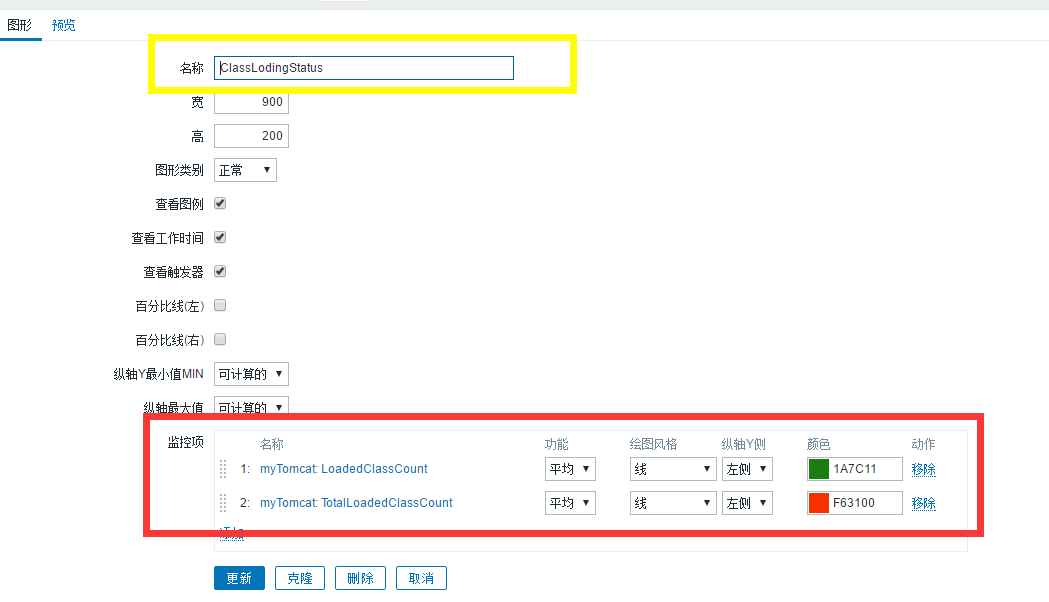
上图中的信息内容根据jconsole中获取到的信息内容填写，例如：





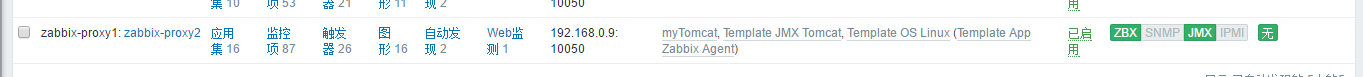
定义图形

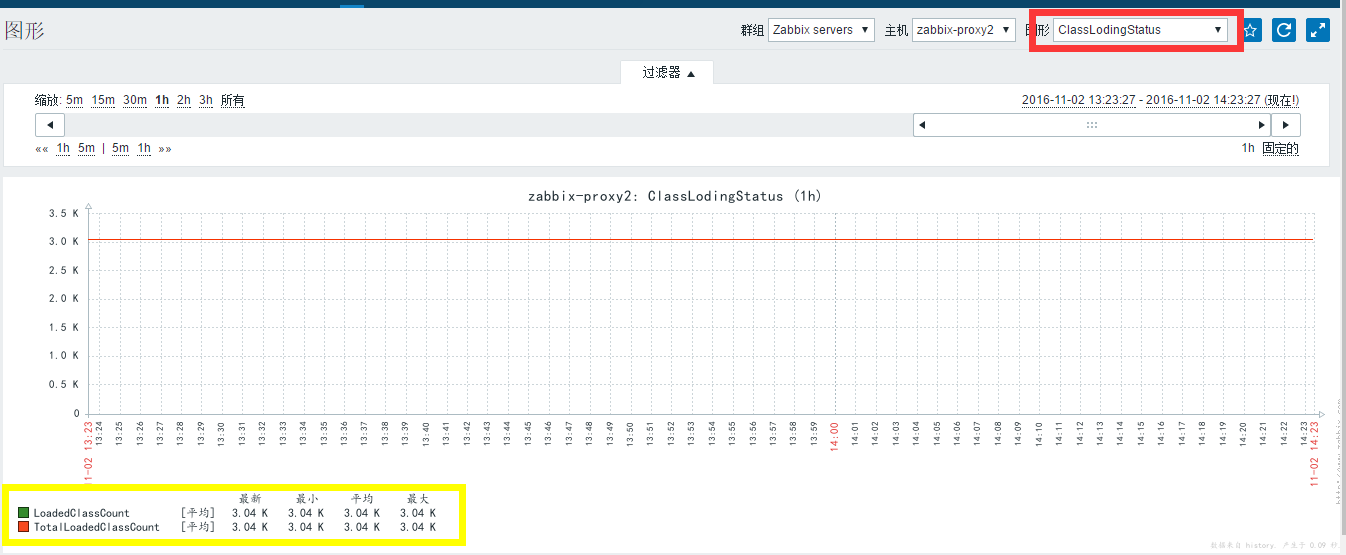




将定义模板应用监控主机上







关于模板中item的配置说明可参考官方文档中的

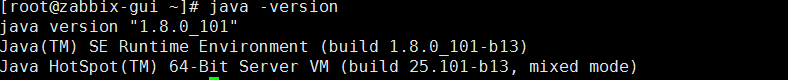
<https://www.zabbix.com/documentation/3.2/manual/config/items/itemtypes/jmx_monitoring>

|  |
| --- |
| 注意：  版本问题  Tomcat6，7，与tomcat8中的监控项配置的名称可能不一致导致无法获取数据信息，可已通过根据jconsole获取的信息进行对应监控项的信息修改。 |

## WebLogic监控

### 9.1 安装weblogic 12c

按照7.2安装JDK，这里不再赘述。



**创建组和账号**

#groupadd web

#useradd -g weblogic web

**创建指定安装目录**

#mkdir -p /usr/local/weblogic

**指定用户目录**

#chown -R weblogic:web /usr/local/weblogic

**安装**

**#**cp fmw\_12.2.1.2.0\_wls\_Disk1\_1of1.zip /usr/local/weblogic/ 注：蓝色部分是weblogic安装包路径

#su - weblogic

#cd /usr/local/weblogic

#java -jar /usr/local/weblogic/fmw\_12.2.1.2.0\_wls.jar

进入图形化安装界面。注意安装路径，点击下一步。

|  |
| --- |
| 参考：  http://wenku.baidu.com/link?url=QTto1P2XKuubBpTFCIdrRufSelYIiWm2mwMTXNyOR-lCZ0SCUa\_JiL0KsCx4EH3pFTftlDAqdqRkLtfvpTDSiiK6ZxydzyBnaGY-TVe9rEe |

### 9.2 weblogic监控

修改配置文件（以下操作都在这个目录下）

#cd /usr/local/weblogic/Oracle/Middleware/Oracle\_Home/user\_projects/domains/wl\_server/bin

添加如下选项

# vi setDomainEnv.sh

JAVA\_OPTIONS="${JAVA\_OPTIONS} -Dcom.sun.management.jmxremote=true -Dcom.sun.management.jmxremote.port=12348 -Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.management.jmxremote.ssl=false"

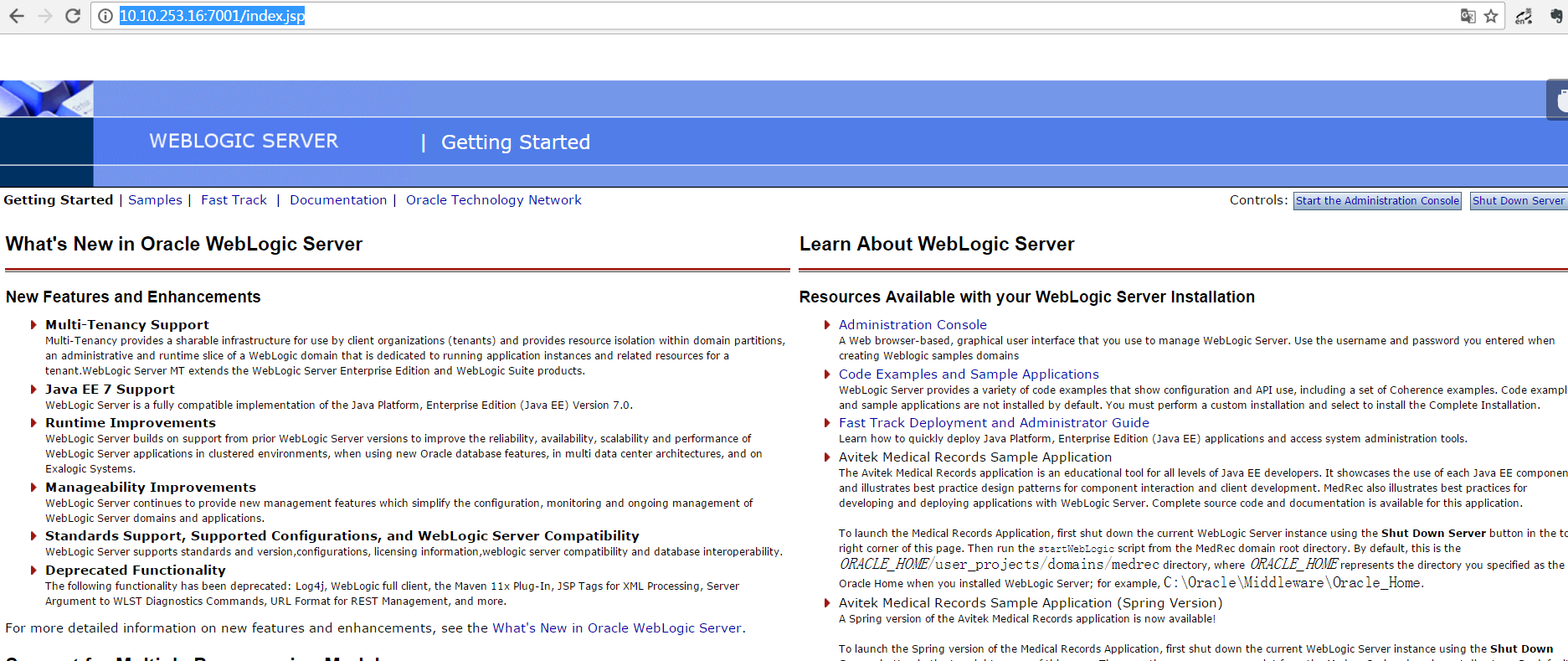
export JAVA\_OPTIONS

|  |
| --- |
| 注意：监控端口可以改变 |

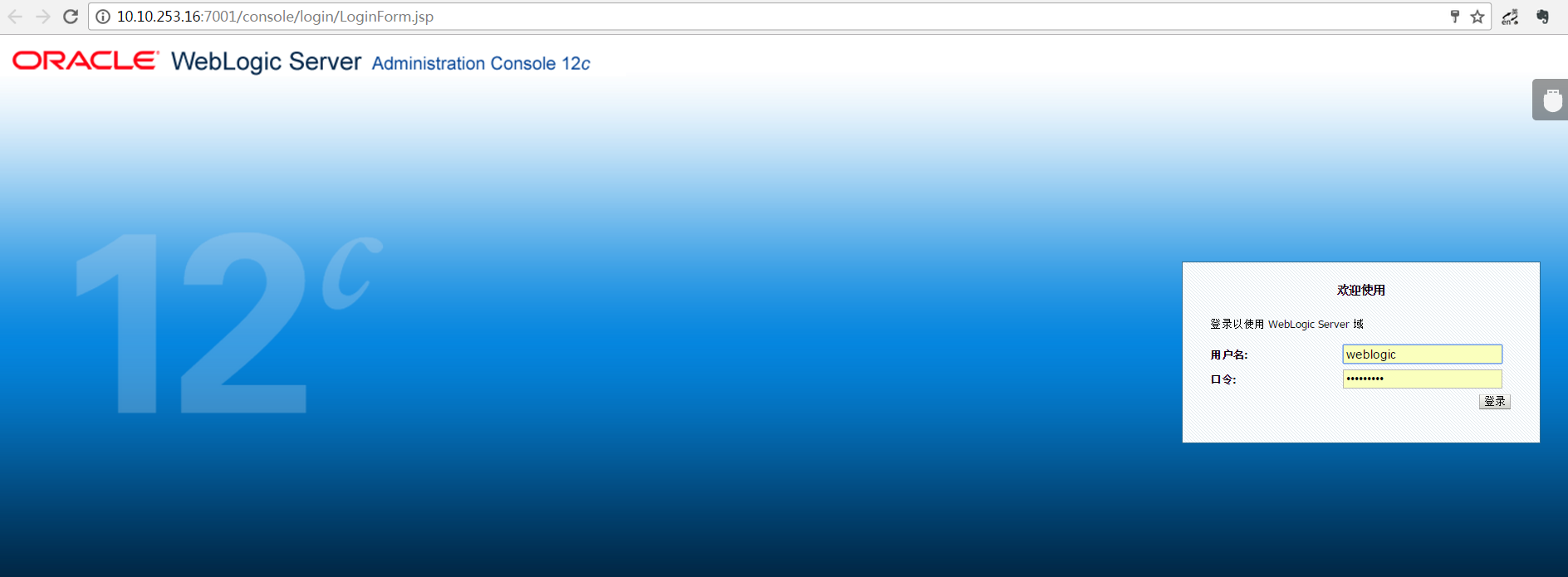
重启weblogic

# ./startWebLogic.sh

访问http://10.10.253.16:7001/index.jsp



点击“start the Administration Console”



|  |
| --- |
| 用户名：weblogic  密码：weblogic1 |

Zabbix-gui配置与tomcat监控一致

## Nginx监控

|  |
| --- |
| 前提：kernel 版本在2.6以上  本次安装openssl openssl\_devel采用rpm安装方式，也可采用yum方式安装  首先需要确认是否安装  openssl-devel包 openssl openssl-devel ssl协议上传输，MD5,SHA1等散列函数  zlib包 zlib zlib-devel http包的gzip压缩  pcre包 pcre pcre-devel 支持正则表达式 |

### 10.1 安装

进入解压目录下

#./configure --prefix=/usr/local/nginx --with-http\_stub\_status\_module --with-http\_ssl\_module --with-pcre

#make && make install

|  |
| --- |
| 参数说明：  --prefix=PATH ： 指定 nginx 的安装目录。默认 /usr/local/nginx，我的是 /usr/local/webserver/nginx  --conf-path=PATH ： 设置nginx.conf配置文件的路径。nginx允许使用不同的配置文件启动，通过命令行中的-c选项。默认为conf/nginx.conf  --user=name ： 设置nginx工作进程的用户。安装完成后，可以随时在nginx.conf配置文件更改user指令。默认的用户名是nobody。--group=name类似  --with-pcre ： 设置PCRE库的源码路径，如果已通过yum方式安装，使用--with-pcre自动找到库文件。使用--with-pcre=PATH时，需要从PCRE网站下载pcre库的源码（8.39）并解压，指定 pcre 的源码路径 ，比如：--with-pcre=/root/pcre-8.39/。perl正则表达式使用在location指令和 ngx\_http\_rewrite\_module模块中。  --with-zlib=PATH ： 指定 zlib（版本1.1.3 - 1.2.5）的源码解压目录。在默认就启用的网络传输压缩模块ngx\_http\_gzip\_module时需要使用zlib 。  --with-http\_ssl\_module ： 使用https协议模块。默认情况下，该模块没有被构建。前提是openssl与openssl-devel已安装  --with-http\_stub\_status\_module ： 用来监控 Nginx 的当前状态  --with-http\_realip\_module ： 通过这个模块允许我们改变客户端请求头中客户端IP地址值(例如X-Real-IP 或 X-Forwarded-For)，意义在于能够使得后台服务器记录原始客户端的IP地址  --add-module=PATH ： 添加第三方外部模块，如nginx-sticky-module-ng或缓存模块。每次添加新的模块都要重新编译（Tengine可以在新加入module时无需重新编译） |

配置nginx开机自启动

在/etc/init.d/目录下创建脚本

#vi /etc/init.d/nginxd

更改脚本权限

#chmod 775 /etc/init.d/nginxd

编写脚本内容

#!/bin/sh

#Startup script for Nginx - this script starts and stops the nginx daemon

#

# chkconfig: - 85 15

# description: Nginx is an HTTP(S) server, HTTP(S) reverse proxy and IMAP/POP3 proxy server

# processname: nginx

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

# pidfile: /usr/local/nginx/logs/nginx.pid

# Source function library.

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

# Source networking configuration.

. /etc/sysconfig/network

# Check that networking is up.

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

nginx="/usr/local/nginx/sbin/nginx"

prog=$(basename $nginx)

NGINX\_CONF\_FILE="/usr/local/nginx/conf/nginx.conf"

[ -f /etc/sysconfig/nginx ] && . /etc/sysconfig/nginx

lockfile=/var/lock/subsys/nginx

start() {

[ -x $nginx ] || exit 5

[ -f $NGINX\_CONF\_FILE ] || exit 6

echo -n $"Starting $prog: "

daemon $nginx -c $NGINX\_CONF\_FILE

retval=$?

echo

[ $retval -eq 0 ] && touch $lockfile

return $retval

}

stop() {

echo -n $"Stopping $prog: "

killproc $prog -QUIT

retval=$?

echo

[ $retval -eq 0 ] && rm -f $lockfile

return $retval

}

restart() {

configtest || return $?

stop

sleep 1

start

}

reload() {

configtest || return $?

echo -n $"Reloading $prog: "

killproc $nginx -HUP

RETVAL=$?

echo

}

force\_reload() {

restart

}

configtest() {

$nginx -t -c $NGINX\_CONF\_FILE

}

rh\_status() {

status $prog

}

rh\_status\_q() {

rh\_status >/dev/null 2>&1

}

case "$1" in

start)

rh\_status\_q && exit 0

$1

;;

stop)

rh\_status\_q || exit 0

$1

;;

restart|configtest)

$1

;;

reload)

rh\_status\_q || exit 7

$1

;;

force-reload)

force\_reload

;;

status)

rh\_status

;;

condrestart|try-restart)

rh\_status\_q || exit 0

;;

\*)

echo $"Usage: $0 {start|stop|status|restart|condrestart|try-restart|reload|force-reload|configtest}"

exit 2

esac

设置开机启动

#chkconfig --add nginxd

#chkconfig nginxd on

#service nginxd start

#service nginxd restart

#service nginxd status

### 10.2 Nginx监控配置

|  |
| --- |
| 首先在安装时必须添加 --with-http\_stub\_status\_module安装配置选项 |

**修改配置文件**

在nginx配置文件（/usr/local/nginx/conf/nginx.conf）中Server项中配置如下内容

#vi /usr/local/nginx/conf/nginx.conf

location /nginx\_status {

stub\_status on;

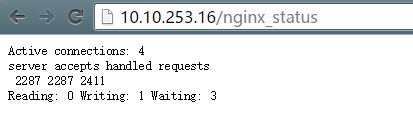
access\_log off;

}

**重启nginx**

#service nginxd restart

浏览器访问http://localhost/nginx\_status



**配置监控脚本**

Zabbix中添加监控nginx脚本，可以存在zabbix脚本路径中，本实验放置在

/usr/local/zabbix/scripts下，文件名为nginx\_status.sh

#!/bin/bash

# Script to fetch nginx statuses for tribily monitoring systems

# Author: krish@toonheart.com

# License: GPLv2

# Set Variables

BKUP\_DATE=`/bin/date +%Y%m%d`

LOG="/data/log/zabbix/webstatus.log" #nginx日志路径

HOST=127.0.0.1

PORT="80"

# Functions to return nginx stats

function active {

/usr/bin/curl "http://$HOST:$PORT/nginx-status" 2>/dev/null| grep 'Active' | awk '{print $NF}'

}

function reading {

/usr/bin/curl "http://$HOST:$PORT/nginx-status" 2>/dev/null| grep 'Reading' | awk '{print $2}'

}

function writing {

/usr/bin/curl "http://$HOST:$PORT/nginx-status" 2>/dev/null| grep 'Writing' | awk '{print $4}'

}

function waiting {

/usr/bin/curl "http://$HOST:$PORT/nginx-status" 2>/dev/null| grep 'Waiting' | awk '{print $6}'

}

function accepts {

/usr/bin/curl "http://$HOST:$PORT/nginx-status" 2>/dev/null| awk NR==3 | awk '{print $1}'

}

function handled {

/usr/bin/curl "http://$HOST:$PORT/nginx-status" 2>/dev/null| awk NR==3 | awk '{print $2}'

}

function requests {

/usr/bin/curl "http://$HOST:$PORT/nginx-status" 2>/dev/null| awk NR==3 | awk '{print $3}'

}

# Run the requested function

$1

**修改zabbix\_agent配置文件**

UnsafeUserParameters=1

并在UserParameter配置项下添加如下内容

UserParameter=nginx.accepts,/usr/local/zabbix/scripts/nginx\_status.sh accepts

UserParameter=nginx.handled,/usr/local/zabbix/scripts/nginx\_status.sh handled

UserParameter=nginx.requests,/usr/local/zabbix/scripts/nginx\_status.sh requests

UserParameter=nginx.connections.active,/usr/local/zabbix/scripts/nginx\_status.sh active

UserParameter=nginx.connections.reading,/usr/local/zabbix/scripts/nginx\_status.sh reading

UserParameter=nginx.connections.writing,/usr/local/zabbix/scripts/nginx\_status.sh writing

UserParameter=nginx.connections.waiting,/usr/local/zabbix/scripts/nginx\_status.sh waiting

**重新启动zabbix\_agent**

#service zabbix\_agentd restart

**测试验证**

|  |
| --- |
| 保证nginx服务开启， |

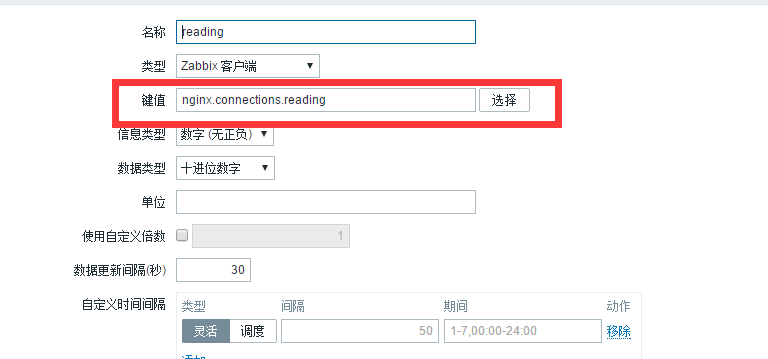
使用zabbix\_get命令来测试

#./zabbix\_get -s 127.0.0.1 -k “nginx.accepts”



**zabbix界面配置**

在zabbix-web界面中可以配置添加模板，应用集可以自定义，监控项配置如下



间隔一定时间可以通过界面最新数据去查询添加的监控项是否获取到数据