**Zabbix安装配置手册**

目录

[1.1 简介 4](#_Toc481051847)

[1.1.1 功能介绍 4](#_Toc481051848)

[1.1.2 优缺点 5](#_Toc481051849)

[1.2 安装 5](#_Toc481051850)

[1.2.1 安装准备 6](#_Toc481051851)

[1.2.2 开始安装 30](#_Toc481051852)

[1.3 配置测试 48](#_Toc481051853)

简介

Zabbix 是一个高度集成的网络监控解决方案，可以提供企业级的开源分布式监控解决

方案，由一个国外的团队持续维护更新，软件可以自由下载使用，运作团队靠提供收费的技

术支持赢利。

官方网站：<https://www.zabbix.com/>

Zabbix 3.2 官方文档：<https://www.zabbix.com/documentation/3.2/manual>

Zabbix 通过 C/S 模式采集数据，通过 B/S 模式在 web 端展示和配置。

被监控端：主机通过安装 agent 方式采集数据，网络设备通过 SNMP 方式采集数据

Server 端：通过收集 SNMP 和 agent 发送的数据，写入数据库（MySQL，ORACLE 等） ，

再通过 php+apache 在 web 前端展示。

Zabbix 运行条件：

Server：Zabbix Server 需运行在LAMP(Linux+Apache+MySQL+PHP)环境下（或者

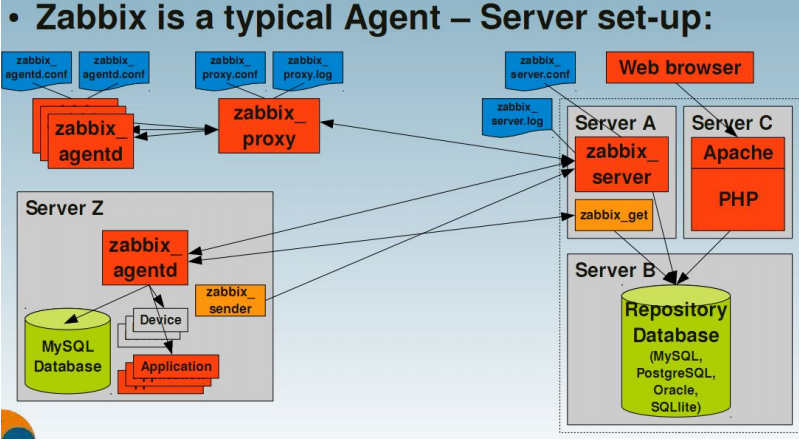
LNMP(Linux+Nginx+MySQL+PHP)） ，对硬件要求低

Agent：目前已有的 agent 基本支持市面常见的 OS，包含 Linux、HPUX、Solaris、Sun、

windows

SNMP：支持各类常见的网络设备

监控过程逻辑如图示：



功能介绍

具备常见的商业监控软件所具备的功能（主机的性能监控、网络设备性能监控、数据库

性能监控、FTP 等通用协议监控、多种告警方式、详细的报表图表绘制）：

支持自动发现网络设备和服务器（可以通过配置自动发现服务器规则来实现）

支持自动发现（low discovery）key 实现动态监控项的批量监控（需写脚本）

支持分布式，能集中展示、管理分布式的监控点

扩展性强，server 提供通用接口（api 功能） ，可以自己开发完善各类监控（根据相关接

口编写程序实现）

编写插件容易，可以自定义监控项，报警级别的设置。

数据收集

可用和性能检测

支持 snmp(包括 trapping and polling)，IPMI，JMX，SSH，TELNET

自定义的检测，自定义收集数据的频率，服务器/代理和客户端模式

灵活的触发器，您可以定义非常灵活的问题阈值，称为触发器，从后端数据库的参考值

高可定制的报警，发送通知，可定制的报警升级，收件人，媒体类型

通知可以使用宏变量有用的变量

自动操作包括远程命令

实时的绘图功能，监控项实时的将数据绘制在图形上面

WEB 监控能力，ZABBIX 可以模拟鼠标点击了一个网站，并检查返回值和响应时间

应用 api 功能，可以方便的和其他系统结合，包括手机客户端的使用。

更多功能请查看

<https://www.zabbix.com/documentation/3.2/manual/introduction/features>

### 优缺点

优点：

开源，无软件成本投入

Server 对设备性能要求低

支持设备多，自带多种监控模板

支持分布式集中管理，有自动发现功能，可以实现自动化监控

开放式接口，扩展性强，插件编写容易

当监控的 item 比较多服务器队列比较大时可以采用被动状态，被监控客户端主动从

server 端去下载需要监控的 item 然后取数据上传到 server 端。 这种方式对服务器的负载比较小。

Api 的支持，方便与其他系统结合

缺点：

需在被监控主机上安装 agent，所有数据都存在数据库里，产生的数据据很大,瓶颈主要

在数据库。

## 安装

本文档安装系统环境为CentOS7

Zabbix版本为3.2.5

服务端IP为：192.168.50.46

客户端IP为：192.168.50.35

### 安装准备

**本文档所需软件包等附件地址为：**

**链接：http://pan.baidu.com/s/1miFjDyS 密码：khwc**

**按照本文档安装配置前，请先打开上面地址下载好相关附件**

#### 系统准备

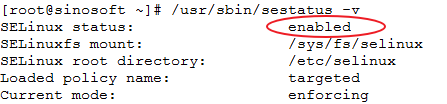
在安装Zabbix之前先对服务端和客户端的一些系统配置和运行环境进行适当的安装调整，保证Zabbix能够正常安装运行

##### 关闭Selinux

SElinux是Linux自带的一款安全加强工具，默认是开启的,在开启状态下安装LAMP/LNMP等Web环境软件会失败，所以需要关闭掉

查看Selinux状态

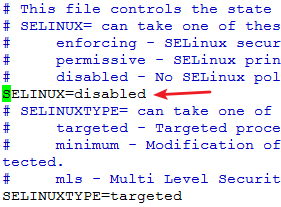
/usr/sbin/sestatus -v



临时关闭：

setenforce off

  
永久性关闭：  
vim /etc/selinux/config  
把 SELINUX=enforcing 改为 SELINUX=disabled



然后重启系统

init -- 6

重新使用Xshell连接服务器，再次查看SELinux的状态是否为关闭



尽量选择永久性关闭，避免在后面的安装使用过程中出现错误。

##### 关闭防火墙

监控机开放防火墙

firewall-cmd --zone=public --add-port=80/tcp --permanent

firewall-cmd --reload

监控机和被监控服务机上同时开放snmp的防火墙端口

firewall-cmd --zone=public --add-port=161/tcp --permanent

firewall-cmd --zone=public --add-port=161/udp --permanent

firewall-cmd --reload

监控机和被监控服务机上同时开放snmp的防火墙端口

firewall-cmd --zone=public --add-port=10050/tcp --permanent

firewall-cmd --zone=public --add-port=10050/udp --permanent

firewall-cmd --reload

也可以选择临时关闭防火墙，重启系统后不生效

systemctl stop firewalld.service

##### 设置时间同步

Zabbix对时间要求比较严格，如果服务器上时间不一致会导致数据报错，所以必须设置时间同步

安装ntpdate工具

yum -y install ntp ntpdate



设置系统时间与网络时间同步

ntpdate cn.pool.ntp.org

将系统时间写入硬件时间

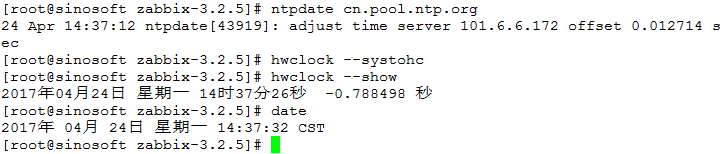
hwclock --systohc

查看硬件时间

hwclock --show

查看系统时间

date



##### 指定路径

在本文中指定软件源代码包存放位置为：/usr/local/src

源码包编译安装位置为：/usr/local/软件名字

##### 安装编译工具及库文件

编译安装软件需要先安装编译工具等系统软件包，使用如下命令安装

yum install -y apr\* autoconf automake bison bzip2 bzip2\* cloog-ppl compat\* cpp curl curl-devel fontconfig fontconfig-devel freetype freetype\* freetype-devel gcc gcc-c++ gtk+-devel gd gettext gettext-devel glibc kernel kernel-headers keyutils keyutils-libs-devel krb5-devel libcom\_err-devel libpng libpng-devel libjpeg\* libsepol-devel libselinux-devel libstdc++-devel libtool\* libgomp libxml2 libxml2-devel libXpm\* libtiff libtiff\* make mpfr ncurses\* ntp openssl openssl-devel patch pcre-devel perl php-common php-gd policycoreutils telnet t1lib t1lib\* nasm nasm\* wget zlib-devel

##### 准备安装包

服务端：

Zabbix安装包 zabbix-3.2.5.tar.gz

客户端：

Zabbix安装包 zabbix-3.2.5.tar.gz

字符集转换库 libiconv-1.15.tar.gz

上述软件包在文档附件中已经提供，可以直接使用，

将以上软件包使用XFTP工具上传到分别上传到相应服务器的/usr/local/src目录

#### Web环境准备

这里需要先在服务端上安装Web环境，便于Zabbix监控Web页面的正常访问。

常见的Web环境有LAMP(Linux+Apache+MySQL+PHP)和LNMP(Linux+Nginx+MySQL+PHP)等等，在这里推荐使用LNMP，因为Nginx相对于Apache来说支持更高的并发，而且在高并发状态下还能保持低资源低消耗高性能，适合Zabbix后面拓展监控多台机器的使用情况，故此部分为编译安装LNMP教程(CentOS7+Nginx1.6.0+MySQL5.6.19+PHP5.5.14)。

##### 准备安装包

1. Nginx安装包 nginx-1.6.0.tar.gz

2. MySQL安装包 mysql-5.6.19.tar.gz

3. PHP安装包 php-5.5.14.tar.gz

4. 支持nginx伪静态 pcre-8.35.tar.gz

5. nginx扩展 openssl-1.0.1h.tar.gz

6. nginx扩展 zlib-1.2.8.tar.gz

7. MySQL编译工具 cmake-2.8.11.2.tar.gz

8. php扩展 libmcrypt-2.5.8.tar.gz

9. php扩展 yasm-1.2.0.tar.gz

10. php扩展 t1lib-5.1.2.tar.gz

11. gd库安装包 libgd-2.1.0.tar.gz

12. gd库安装包 libvpx-v1.3.0.tar.bz2

13. gd库依赖包 tiff-4.0.3.tar.gz

14. gd库依赖包 libpng-1.6.12.tar.gz

15. gd库依赖包 freetype-2.5.3.tar.gz

16. gd库依赖包 jpegsrc.v9a.tar.gz

上述软件包在文档附件中已经提供，可以直接使用，

将以上软件包使用XFTP工具上传到/usr/local/src目录

##### 安装LNMP

###### 安装MySQL

清除痕迹

删除Centos7自带的mariadb数据库

yum -y remove mariadb\*



安装cmake

camke在这里用来作为MySQL的编译安装工具

先切换至软件包存放目录

cd /usr/local/src

解压安装包

tar -zxvf cmake-2.8.11.2.tar.gz

切换至解压出来的安装目录内

cd cmake-2.8.11.2

对即将安装的软件进行配置，检查当前的环境是否满足要安装软件的依赖关系

./configure

编译

make

安装

make install

开始安装

**创建用户目录**

添加mysql组

groupadd mysql

创建用户mysql并加入到mysql组，不允许mysql用户直接登录系统

useradd -g mysql mysql -s /bin/false

创建MySQL数据库存放目录

mkdir -p /data/mysql

设置MySQL数据库存放目录权限

chown -R mysql:mysql /data/mysql

创建MySQL安装目录

mkdir -p /usr/local/mysql

切换至软件包存放目录

cd /usr/local/src

解压安装包

tar -zxvf mysql-5.6.19.tar.gz

切换至解压出来的安装目录内

cd mysql-5.6.19

**安装配置**

配置

cmake . -DCMAKE\_INSTALL\_PREFIX=/usr/local/mysql -DMYSQL\_DATADIR=/data/mysql -DSYSCONFDIR=/etc

编译

make

安装

make install

删除系统默认的配置文件

rm -rf /etc/my.cnf

进入MySQL安装目录

cd /usr/local/mysql

生成mysql系统数据库

./scripts/mysql\_install\_db --user=mysql --basedir=/usr/local/mysql --datadir=/data/mysql

添加my.cnf配置文件到/etc目录的软连接

ln -s /usr/local/mysql/my.cnf /etc/my.cnf

把Mysql加入系统启动

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

增加执行权限

chmod 755 /etc/init.d/mysqld

加入开机启动

chkconfig mysqld on

编辑MySQL启动命令

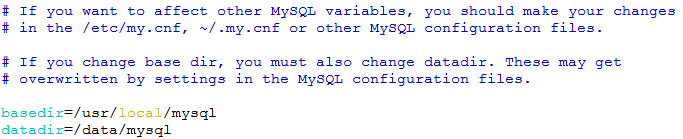
vim /etc/rc.d/init.d/mysqld

输入/basedir快速查找文件中basedir内容位置

添加MySQL程序安装路径和数据库存放目录

basedir=/usr/local/mysql

datadir=/data/mysql



保存退出

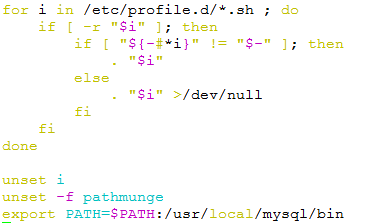
:wq

把mysql服务加入系统环境变量

vim /etc/profile

输入G快速跳转至最后一行，在最后一行下面添加下面内容

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



保存并退出

:wq

使配置立刻生效

source /etc/profile

下面这两行把myslq的库文件链接到系统默认的位置，这样在编译类似PHP等软件时可以不用指定mysql的库文件地址

ln -s /usr/local/mysql/lib/mysql /usr/lib/mysql

ln -s /usr/local/mysql/include/mysql /usr/include/mysql

创建目录

mkdir /var/lib/mysql

添加软链接

ln -s /tmp/mysql.sock /var/lib/mysql/mysql.sock

启动测试

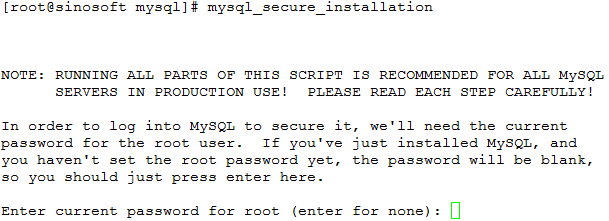
启动MySQL

service mysqld start

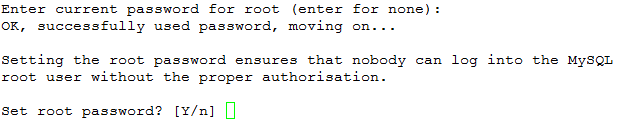


初始化安全策略

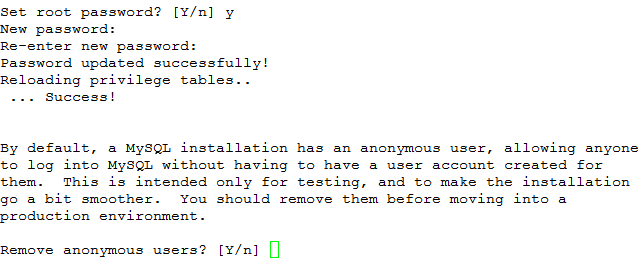
mysql\_secure\_installation



提示输入MySQL密码，这里MySQL安装完成后默认密码为空，直接回车就可以进去



进去之后提示是否设置MySQL的root用户密码，按Y 回车输入2次密码，本文这里设置MySQL密码为123456



设置完密码后接着出现下面的初始化安全配置选项，根据本文下面步骤设置

Remove anonymous users? [Y/n] #移除匿名用户,这里设置Y

Success!

Disallow root login remotely? [Y/n] #禁止root远程登录, 这里设置N,便于后面远程连接

skipping.

Remove test database and access to it? [Y/n] #删除测试数据库,这里设置Y

- Dropping test database...

Success!

- Removing privileges on test database...

Success!

Reload privilege tables now? [Y/n] # 重新加载授权表,这里设置Y

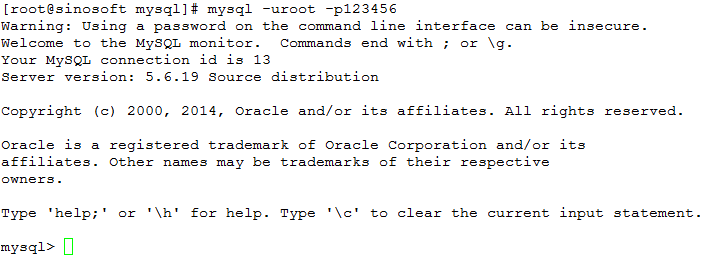
Success!

All done! If you've completed all of the above steps, your MySQL

installation should now be secure.

登陆MySQL

mysql -uroot -p123456



刷新权限

flush privileges;



重启MySQL服务

service mysqld restart



开放防火墙端口

firewall-cmd --zone=public --add-port=3306/tcp --permanent

firewall-cmd --reload

###### 安装Nginx

安装拓展

先安装以下拓展依赖包

**安装pcre**

切换至软件包存放目录

cd /usr/local/src

创建安装目录

mkdir /usr/local/pcre

解压安装包

tar -zxvf pcre-8.35.tar.gz

切换至解压出来的安装目录内

cd pcre-8.35

对即将安装的软件进行配置，检查当前的环境是否满足要安装软件的依赖关系

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

编译

make

安装

make install

**安装openssl**

切换至软件包存放目录

cd /usr/local/src

创建安装目录

mkdir /usr/local/openssl

解压安装包

tar -zxvf openssl-1.0.1h.tar.gz

切换至解压出来的安装目录内

cd openssl-1.0.1h

对即将安装的软件进行配置，检查当前的环境是否满足要安装软件的依赖关系

./config --prefix=/usr/local/openssl

编译

make

安装

make install

编辑环境变量

vim /etc/profile

在最后一行回车加入下面一行配置

export PATH=$PATH:/usr/local/openssl/bin

保存并退出

:wq

使修改后的环境变量立即生效

source /etc/profile

**安装zlib**

切换至软件包存放目录

cd /usr/local/src

创建安装目录

mkdir /usr/local/zlib

解压安装包

tar -zxvf zlib-1.2.8.tar.gz

切换至解压出来的安装目录内

cd zlib-1.2.8

对即将安装的软件进行配置，检查当前的环境是否满足要安装软件的依赖关系

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

编译

make

安装

make install

开始安装

**创建用户目录**

添加用户组

groupadd www

创建用户www并加入到www组，不允许www用户直接登录系统

useradd -g www www -s /bin/false

切换至软件包存放目录

cd /usr/local/src

解压安装包

tar -zxvf nginx-1.6.0.tar.gz

切换至解压出来的安装目录内

cd nginx-1.6.0

**安装配置**

对即将安装的软件进行配置，检查当前的环境是否满足要安装软件的依赖关系

./configure --prefix=/usr/local/nginx --without-http\_memcached\_module --user=www --group=www --with-http\_stub\_status\_module --with-http\_ssl\_module --with-http\_gzip\_static\_module --with-openssl=/usr/local/src/openssl-1.0.1h --with-zlib=/usr/local/src/zlib-1.2.8 --with-pcre=/usr/local/src/pcre-8.35

**注意：--with-openssl=/usr/local/src/openssl-1.0.1h --with-zlib=/usr/local/src/zlib-1.2.8 --with-pcre=/usr/local/src/pcre-8.35指向的是源码包解压的路径，而不是安装的路径，否则会报错**

编译

make

安装

make install

设置Nginx开机启动

新增启动文件添加下面内容

vim /etc/rc.d/init.d/nginx

#!/bin/sh

#

# 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: /etc/nginx/nginx.conf

# 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

make\_dirs() {

# make required directories

user=`$nginx -V 2>&1 | grep "configure arguments:" | sed 's/[^\*]\*--user=\([^ ]\*\).\*/\1/g' -`

if [ -z "`grep $user /etc/passwd`" ]; then

useradd -M -s /bin/nologin $user

fi

options=`$nginx -V 2>&1 | grep 'configure arguments:'`

for opt in $options; do

if [ `echo $opt | grep '.\*-temp-path'` ]; then

value=`echo $opt | cut -d "=" -f 2`

if [ ! -d "$value" ]; then

# echo "creating" $value

mkdir -p $value && chown -R $user $value

fi

fi

done

}

start() {

[ -x $nginx ] || exit 5

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

make\_dirs

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

保存并退出

:wq!

赋予文件执行权限

chmod 775 /etc/rc.d/init.d/nginx

设置开机启动

chkconfig nginx on

启动测试

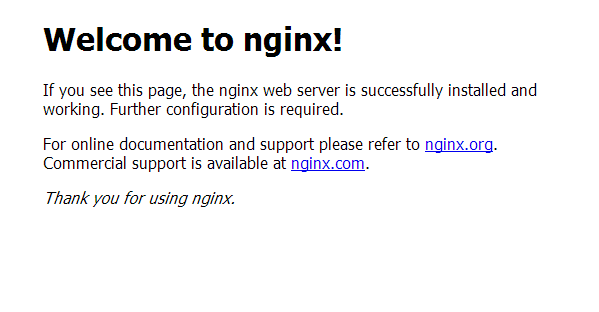
启动Nginx

/usr/local/nginx/sbin/nginx

重启Nginx

/etc/rc.d/init.d/nginx restart

在浏览器中打开服务器IP地址，会看到下面的界面，说明Nginx安装成功。



###### 安装PHP

安装拓展

先安装以下拓展依赖包

**安装yasm**

切换至软件包存放目录

cd /usr/local/src

解压安装包

tar -zxvf yasm-1.2.0.tar.gz

切换至解压出来的安装目录内

cd yasm-1.2.0

对即将安装的软件进行配置，检查当前的环境是否满足要安装软件的依赖关系

./configure

编译

make

安装

make install

**安装libmcrypt**

切换至软件包存放目录

cd /usr/local/src

解压安装包

tar -zxvf libmcrypt-2.5.8.tar.gz

切换至解压出来的安装目录内

cd libmcrypt-2.5.8

对即将安装的软件进行配置，检查当前的环境是否满足要安装软件的依赖关系

./configure

编译

make

安装

make install

**安装libvpx**

切换至软件包存放目录

cd /usr/local/src

解压安装包

tar -xvf libvpx-v1.3.0.tar.bz2

切换至解压出来的安装目录内

cd libvpx-v1.3.0

对即将安装的软件进行配置，检查当前的环境是否满足要安装软件的依赖关系

./configure

编译

make

安装

make install

**安装tiff**

切换至软件包存放目录

cd /usr/local/src

解压安装包

tar -zxvf tiff-4.0.3.tar.gz

切换至解压出来的安装目录内

cd tiff-4.0.3

对即将安装的软件进行配置，检查当前的环境是否满足要安装软件的依赖关系

./configure --prefix=/usr/local/tiff --enable-shared

编译

make

安装

make install

**安装libpng**

切换至软件包存放目录

cd /usr/local/src

解压安装包

tar -zxvf libpng-1.6.12.tar.gz

切换至解压出来的安装目录内

cd libpng-1.6.12

对即将安装的软件进行配置，检查当前的环境是否满足要安装软件的依赖关系

./configure --prefix=/usr/local/libpng --enable-shared

编译

make

安装

make install

**安装freetype**

切换至软件包存放目录

cd /usr/local/src

解压安装包

tar -zxvf freetype-2.5.3.tar.gz

切换至解压出来的安装目录内

cd freetype-2.5.3

对即将安装的软件进行配置，检查当前的环境是否满足要安装软件的依赖关系

./configure --prefix=/usr/local/freetype --enable-shared

编译

make

安装

make install

**安装jpeg**

切换至软件包存放目录

cd /usr/local/src

解压安装包

tar -zxvf jpegsrc.v9a.tar.gz

切换至解压出来的安装目录内

cd jpeg-9a

对即将安装的软件进行配置，检查当前的环境是否满足要安装软件的依赖关系

./configure --prefix=/usr/local/jpeg --enable-shared

编译

make

安装

make install

**安装libgd**

切换至软件包存放目录

cd /usr/local/src

解压安装包

tar -zxvf libgd-2.1.0.tar.gz

切换至解压出来的安装目录内

cd libgd-2.1.0

对即将安装的软件进行配置，检查当前的环境是否满足要安装软件的依赖关系

./configure --prefix=/usr/local/libgd --enable-shared --with-jpeg=/usr/local/jpeg --with-png=/usr/local/libpng --with-freetype=/usr/local/freetype --with-fontconfig=/usr/local/freetype --with-xpm=/usr/ --with-tiff=/usr/local/tiff --with-vpx=/usr/local/libvpx

编译

make

安装

make install

**安装t1lib**

切换至软件包存放目录

cd /usr/local/src

解压安装包

tar -zxvf t1lib-5.1.2.tar.gz

切换至解压出来的安装目录内

cd t1lib-5.1.2

对即将安装的软件进行配置，检查当前的环境是否满足要安装软件的依赖关系

./configure --prefix=/usr/local/t1lib --enable-shared

编译

make without\_doc

安装

make install

开始安装

注意：CentOS7系统是64位，请执行以下两条命令，否则安装php会出错（32位系统不需要执行）

\cp -frp /usr/lib64/libltdl.so\*  /usr/lib/

\cp -frp /usr/lib64/libXpm.so\* /usr/lib/

切换至软件包存放目录

cd /usr/local/src

解压安装包

tar -zvxf php-5.5.14.tar.gz

切换至解压出来的安装目录内

cd php-5.5.14

更新变量设置

export LD\_LIBRARY\_PATH=/usr/local/libgd/lib

对即将安装的软件进行配置，检查当前的环境是否满足要安装软件的依赖关系

./configure --prefix=/usr/local/php --with-config-file-path=/usr/local/php/etc --with-mysql=/usr/local/mysql --with-mysqli=/usr/local/mysql/bin/mysql\_config --with-mysql-sock=/tmp/mysql.sock --with-pdo-mysql=/usr/local/mysql --with-gd --with-png-dir=/usr/local/libpng --with-jpeg-dir=/usr/local/jpeg --with-freetype-dir=/usr/local/freetype --with-xpm-dir=/usr/ --with-vpx-dir=/usr/local/libvpx/ --with-zlib-dir=/usr/local/zlib --with-t1lib=/usr/local/t1lib --with-iconv --enable-libxml --enable-xml --enable-bcmath --enable-shmop --enable-sysvsem --enable-inline-optimization --enable-opcache --enable-mbregex --enable-fpm --enable-mbstring --enable-ftp --enable-gd-native-ttf --with-openssl --enable-pcntl --enable-sockets --with-xmlrpc --enable-zip --enable-soap --without-pear --with-gettext --enable-session --with-mcrypt --with-curl --enable-ctype

编译

make without\_doc

安装

make install

修改配置

**编辑php.ini**

复制php配置文件到安装目录

cp php.ini-production /usr/local/php/etc/php.ini

删除系统自带配置文件

rm -rf /etc/php.ini

添加软链接到 /etc目录

ln -s /usr/local/php/etc/php.ini /etc/php.ini

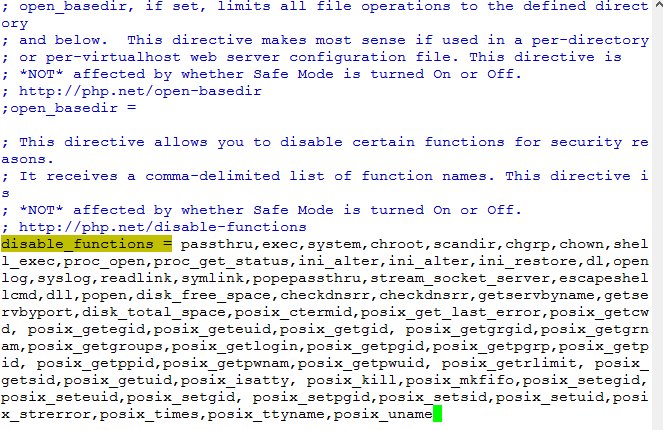
编辑配置文件

vim /usr/local/php/etc/php.ini

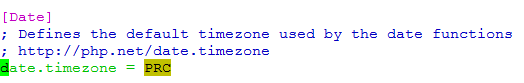
输入/disable\_functions =快速定位到文档相关内容部分，将相关部分内容修改为：

disable\_functions = passthru,exec,system,chroot,chgrp,chown,shell\_exec,proc\_open,proc\_get\_status,ini\_alter,ini\_alter,ini\_restore,dl,openlog,syslog,readlink,symlink,popepassthru,stream\_socket\_server,escapeshellcmd,dll,popen,disk\_free\_space,checkdnsrr,checkdnsrr,getservbyname,getservbyport,disk\_total\_space,posix\_ctermid,posix\_get\_last\_error,posix\_getcwd, posix\_getegid,posix\_geteuid,posix\_getgid, posix\_getgrgid,posix\_getgrnam,posix\_getgroups,posix\_getlogin,posix\_getpgid,posix\_getpgrp,posix\_getpid, posix\_getppid,posix\_getpwnam,posix\_getpwuid, posix\_getrlimit, posix\_getsid,posix\_getuid,posix\_isatty, posix\_kill,posix\_mkfifo,posix\_setegid,posix\_seteuid,posix\_setgid, posix\_setpgid,posix\_setsid,posix\_setuid,posix\_strerror,posix\_times,posix\_ttyname,posix\_uname

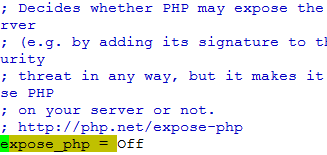
意思为列出PHP可以禁用的函数，如果某些程序需要用到这个函数，可以删除，取消禁用。



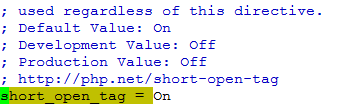
输入/ date.timezone =快速定位到文档相关内容部分，修改为date.timezone = PRC，意思为设置时区为中国



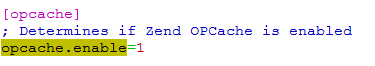
输入/ expose\_php = On快速定位到文档相关内容部分，修改为：expose\_php = Off ，意思为禁止显示php版本的信息



输入/ short\_open\_tag = Off快速定位到文档相关内容部分，修改为：short\_open\_tag = ON ，意思为支持php短标签



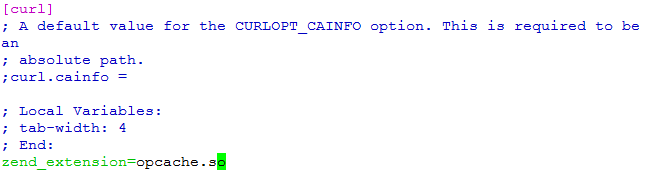
输入/ opcache.enable=0快速定位到文档相关内容部分，去掉前面分号，修改为opcache.enable=1 ，意思为php支持opcode缓存



输入/ opcache.enable\_cli=0快速定位到文档相关内容部分，去掉前面分号，修改为opcache.enable\_cli=1



输入G快速跳转至最后一行，在最后一行添加：zend\_extension=opcache.so ，意思为开启opcode缓存功能



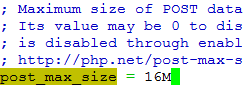
保存并退出

:wq!

编辑修改PHP配置，符合zabbix安装要求

vim /etc/php.ini

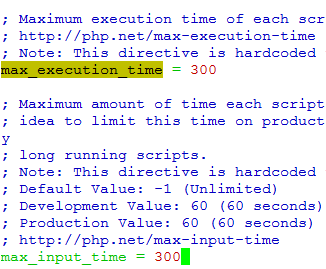
输入/ post\_max\_size快速定位到文档相关内容部分，修改为post\_max\_size =16M



输入/ max\_execution\_time快速定位到文档相关内容部分，修改为

max\_execution\_time =300

max\_input\_time =300



保存退出

:wq

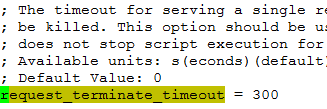
**编辑php-fpm.conf**

编辑修改

vim /usr/local/php/etc/php-fpm.conf

输入/ request\_terminate\_timeou快速定位到文档相关内容部分，修改为

request\_terminate\_timeout = 300



保存退出

:wq

重启php-fpm

service php-fpm reload



拷贝模板文件为php-fpm配置文件

cp /usr/local/php/etc/php-fpm.conf.default /usr/local/php/etc/php-fpm.conf

添加软连接到 /etc目录

ln -s /usr/local/php/etc/php-fpm.conf /etc/php-fpm.conf

编辑php-fpm配置文件

vim /usr/local/php/etc/php-fpm.conf

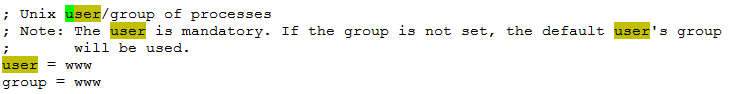
输入/user快速定位到文档user内容部分

设置php-fpm运行账号为www

user = www

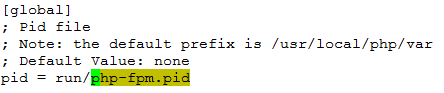
设置php-fpm运行组为www

group = www



输入/php-fpm.pid快速定位到文档php-fpm.pid内容部分

取消前面的分号

  
保存并退出

:wq

设置php-fpm开机启动

拷贝php-fpm到启动目录

cp /usr/local/src/php-5.5.14/sapi/fpm/init.d.php-fpm /etc/rc.d/init.d/php-fpm

添加执行权限

chmod +x /etc/rc.d/init.d/php-fpm

设置开机启动

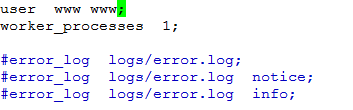
chkconfig php-fpm on

**配置Nginx支持PHP**

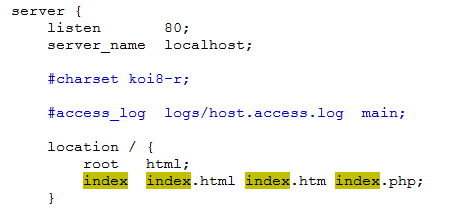
vim /usr/local/nginx/conf/nginx.conf

修改/usr/local/nginx/conf/nginx.conf 配置文件,需做如下修改

user www www; #首行user去掉注释,修改Nginx运行组为www www；必须与/usr/local/php/etc/php-fpm.conf中的user,group配置相同，否则php运行出错



index index.html index.htm index.php; #添加index.php



# pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000

#

location ~ \.php$ {

root html;

fastcgi\_pass 127.0.0.1:9000;

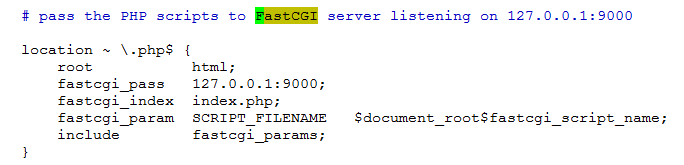
fastcgi\_index index.php;

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

include fastcgi\_params;

}

#取消FastCGI server部分location的注释,注意fastcgi\_param行的参数,改为$document\_root$fastcgi\_script\_name,或者使用绝对路径



修改完后保存并退出

:wq!

重启nginx

/etc/init.d/nginx restart

启动测试

启动php-fpm

service php-fpm start

进入nginx默认网站根目录

cd /usr/local/nginx/html/

删除默认测试页

rm -rf /usr/local/nginx/html/\*

新建index.php文件

vim index.php

添加下面内容

<?php

phpinfo();

?>

保存退出

:wq!

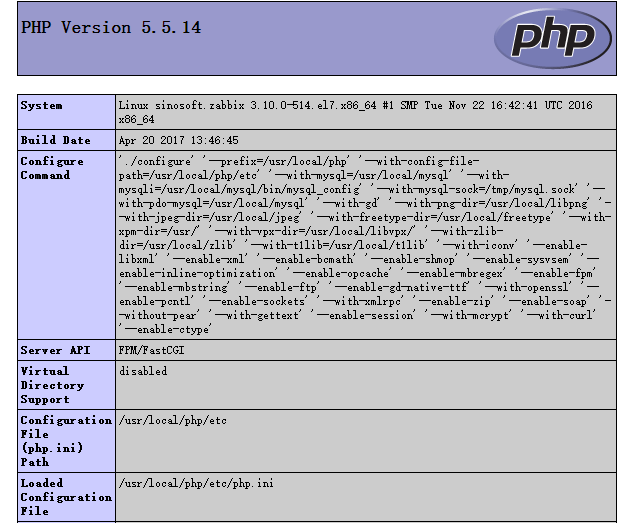
设置目录所有者

chown www.www /usr/local/nginx/html/ -R

设置目录权限

chmod 700 /usr/local/nginx/html/ -R

在浏览器中打开服务器IP地址，会看到下面的界面



至此，CentOS 7编译安装Nginx1.6.0+MySQL5.6.19+PHP5.5.14环境完成。

### 开始安装

#### 服务端

**安装前先执行前面安装准备步骤。**

* + - * 1. 创建数据库

进入软件包下载目录

cd /usr/local/src

解压安装

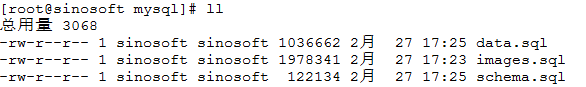
tar -zxvf zabbix-3.2.5.tar.gz

切换至mysql数据库创建脚本目录

cd /usr/local/src/zabbix-3.2.5/database/mysql

列出文件，可以看到有schema.sql、images.sql、data.sql这三个文件

ll



输入密码，进入MySQL控制台

mysql -u root -p123456

创建数据库zabbix，并且数据库编码使用utf8,采用utf8\_general\_ci的数据库校对规则

create database zabbix default character set utf8 collate utf8\_general\_ci;

进入数据库

use zabbix

新建账户zabbix，密码123456,赋予此用户在zabbix库里所有权限，且仅能在本机登陆

grant all privileges on \* to 'zabbix'@'%' identified by '123456' with grant option;

刷新系统授权表

flush privileges;

导入脚本文件到zabbix数据库

source /usr/local/src/zabbix-3.2.5/database/mysql/schema.sql

source /usr/local/src/zabbix-3.2.5/database/mysql/images.sql

source /usr/local/src/zabbix-3.2.5/database/mysql/data.sql

注意：请按照以上顺序进行导入，否则会出错。

导入完成后输入exit退出MySQL命令窗口

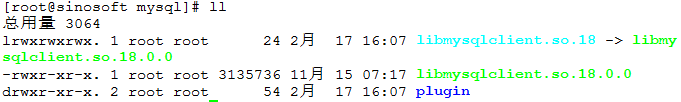
切换到MySQL的动态链接库目录

32位系统为/usr/lib/mysql，注意系统版本,文件版本可能不一样，这里是18.0.0

cd /usr/lib64/mysql

查看文件名字

ll



添加软连接

ln -s libmysqlclient.so.18.0.0 libmysqlclient.so

使配置立即生效

/sbin/ldconfig

* + - * 1. 创建用户

创建用户组zabbix

groupadd zabbix

创建用户zabbix，并且把用户zabbix加入到用户组zabbix中

useradd zabbix -g zabbix -s /bin/false

* + - * 1. 安装依赖包

安装依赖包

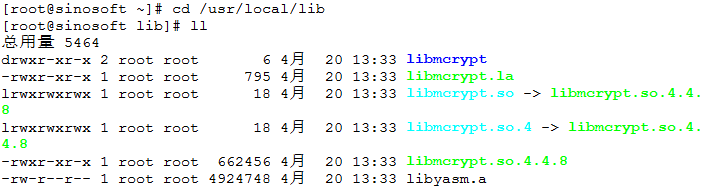
yum -y install net-snmp-devel curl curl-devel mysql-devel

切换到库文件目录

cd /usr/local/lib

查看版本名

ll



添加字符集转换库文件软连接

ln -s /usr/local/lib/libiconv.so.4 /usr/lib/libiconv.so.4

使配置立即生效

/sbin/ldconfig

* + - * 1. 执行安装

进入安装目录

cd /usr/local/src/zabbix-3.2.5

./configure --prefix=/usr/local/zabbix --enable-server --enable-agent --with-net-snmp --with-libcurl --with-libxml2 --enable-proxy --with-mysql=/usr/bin/mysql\_config #配置

make #编译

make install #安装

ln -s /usr/local/zabbix/sbin/\* /usr/local/sbin/ #添加系统软连接

ln -s /usr/local/zabbix/bin/\* /usr/local/bin/ #添加系统软连接

说明：find / -name mysql\_config 查找位置，如果没有mysql\_config，需要安装yum install mysql-devel

* + - * 1. 添加服务端口

添加zabbix服务对应的端口

vim /etc/services

编辑services文件，输入G跳转至最后一行，在最后一行下面添加以下代码

zabbix-agent 10050/tcp # Zabbix Agent

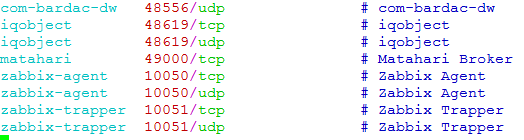
zabbix-agent 10050/udp # Zabbix Agent

zabbix-trapper 10051/tcp # Zabbix Trapper

zabbix-trapper 10051/udp # Zabbix Trapper

保存退出

:wq



* + - * 1. 修改配置文件

切换到Zabbi安装路径的etc目录

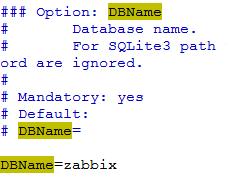
cd /usr/local/zabbix/etc

修改zabbix\_server配置文件

vim /usr/local/zabbix/etc/zabbix\_server.conf

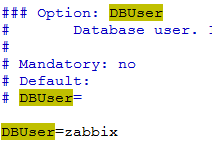
输入/DBname快速定位到文档相关内容部分，将数据库名称修改成前面[创建数据库](#_创建导入数据库)步骤时创建的数据库名字

DBName=zabbix #



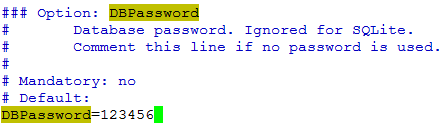
输入/DBUser快速定位到文档相关内容部分，将数据库用户名修改成前面[创建数据库](#_创建导入数据库)步骤时创建的用户名

DBUser=zabbix



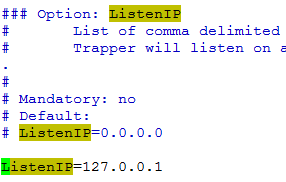
输入/DBPassword快速定位到文档相关内容部分，去掉前面注释，改为前面[创建数据库](#_创建导入数据库)步骤时创建的zabbix用户对应数据库密码

DBPassword=123456

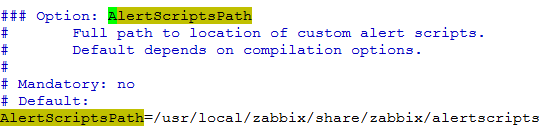


输入/ListenIP快速定位到文档相关内容部分，将第二行ListenIP=127.0.0.1的注释打开，表示数据库ip地址

ListenIP=127.0.0.1



输入/ AlertScriptsPath快速定位到文档相关内容部分，修改zabbix运行脚本存放目录

AlertScriptsPath=/usr/local/zabbix/share/zabbix/alertscripts # 

保存退出

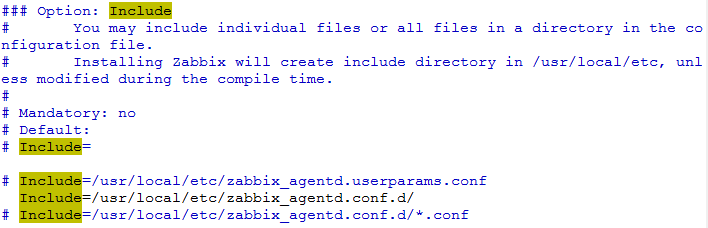
:wq

编辑zabbix\_agentd配置文件

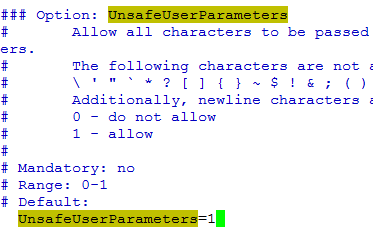
vim /usr/local/zabbix/etc/zabbix\_agentd.conf

输入/ Include快速定位到文档相关内容部分，修改zabbix运行脚本存放目录

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



输入/ UnsafeUserParameters快速定位到文档相关内容部分，设置UnsafeUserParameters=1 ，表示启用自定义key



保存退出

:wq!

根据刚才的配置创建zabbix运行脚本

mkdir -p /usr/local/etc/zabbix\_agentd.conf.d

* + - * 1. 添加开机启动

复制服务端启动脚本到系统启动目录

cp /usr/local/src/zabbix-3.2.5/misc/init.d/fedora/core/zabbix\_server /etc/rc.d/init.d/zabbix\_server

复制客户端启动脚本到系统启动目录

cp /usr/local/src/zabbix-3.2.5/misc/init.d/fedora/core/zabbix\_agentd /etc/rc.d/init.d/zabbix\_agentd

添加服务端启动脚本执行权限

chmod +x /etc/rc.d/init.d/zabbix\_server

添加客户端启动脚本执行权限

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

添加开机启动

chkconfig zabbix\_server on

chkconfig zabbix\_agentd on

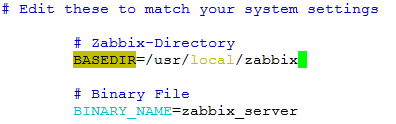
修改zabbix开机启动脚本中的zabbix安装目录

编辑服务端配置文件

vim /etc/rc.d/init.d/zabbix\_server

修改zabbix安装目录

BASEDIR=/usr/local/zabbix/



保存退出

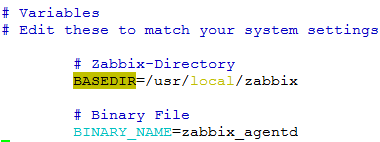
:wq

编辑客户端配置文件

vim /etc/rc.d/init.d/zabbix\_agentd

修改zabbix安装目录

BASEDIR=/usr/local/zabbix/



保存退出

:wq

* + - * 1. 配置web站点

切换到zabbix的安装包目录

cd /usr/local/src/zabbix-3.2.5

将zabbx web的页面文件拷贝至nginx的默认站点目录下

cp -r /usr/local/src/zabbix-3.2.5/frontends/php /usr/local/nginx/html/zabbix

指定zabbix web页面文件的拥有者为nginx用户组

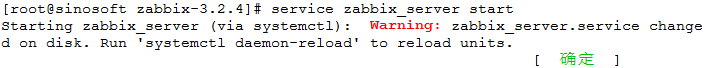
chown www.www -R /usr/local/nginx/html/zabbix

备注：/usr/local/nginx/html为Nginx默认站点目录 www为Nginx运行账户

* + - * 1. 启动服务

启动zabbix服务端

service zabbix\_server start



启动出现警告，按照提示输入下面命令后重启zabbix服务端就好

systemctl daemon-reload

service zabbix\_server restart



启动zabbix客户端

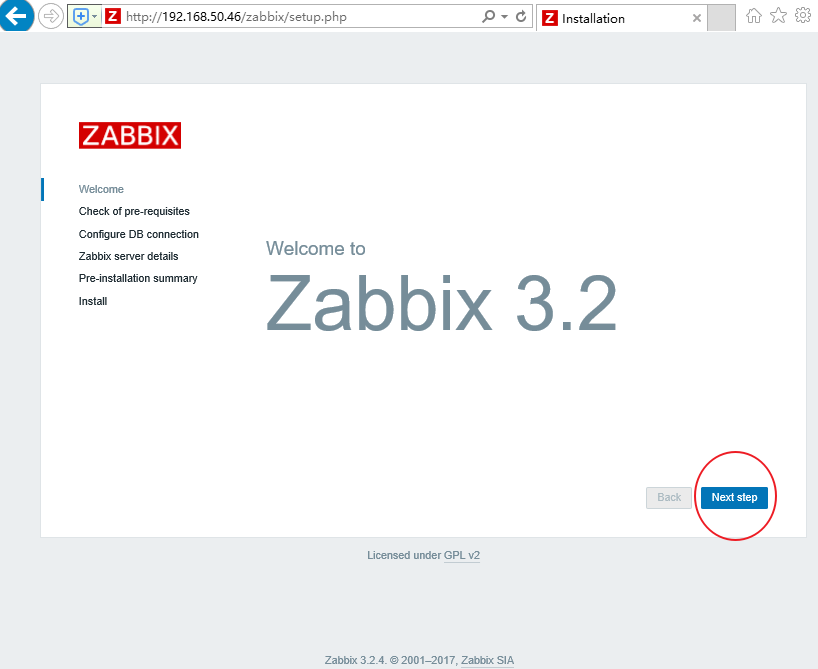
service zabbix\_agentd start



* + - * 1. 安装Web

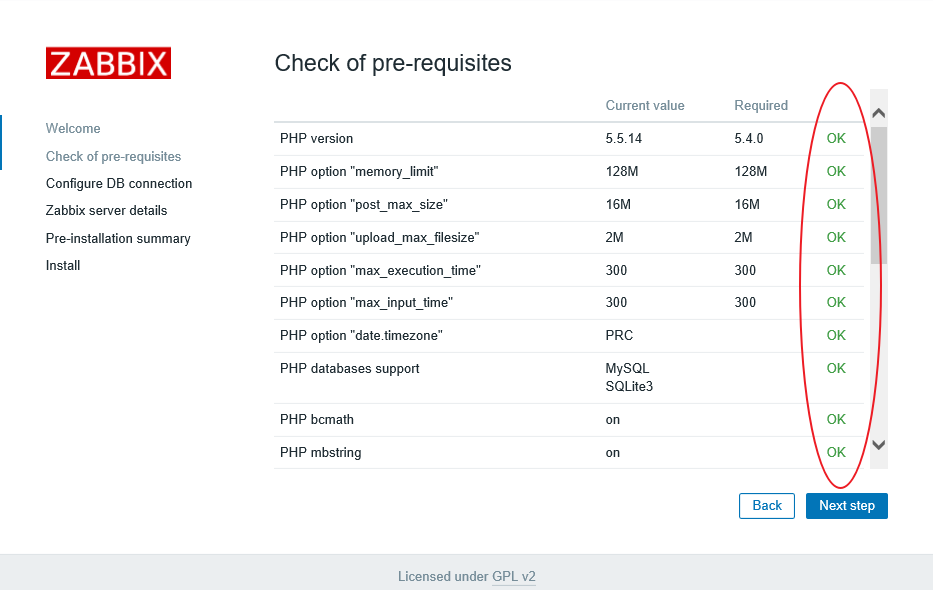
在浏览器中打开：

<http://192.168.50.46/zabbix/setup.php>



点next step下一步，然后查看最右边一列状态是否全OK，如果是红色的unknown Fail

字样则是前面配置有误，根据左边的PHP配置参数返回前面配置步骤中按要求重新配置



下一步配置MySQL数据库信息

Database：MySQL

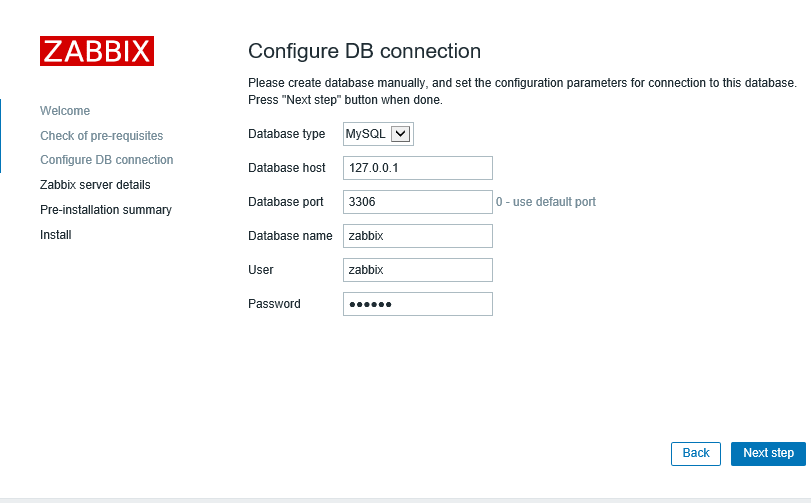
Database host：127.0.0.1

Database port：3306

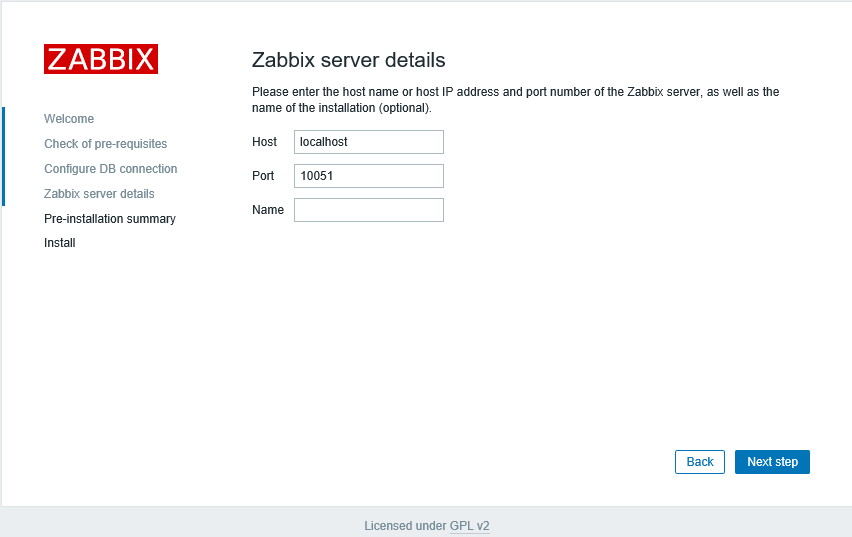
Database name：zabbix

User：zabbix

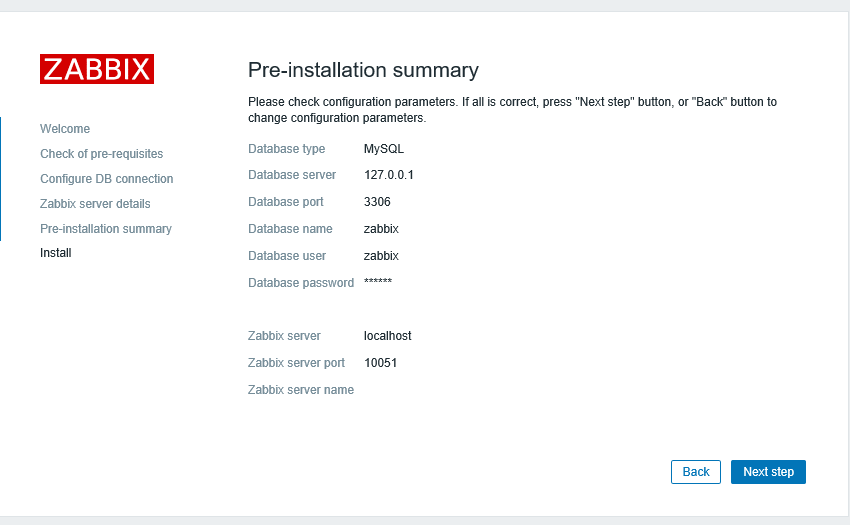
Password：123456



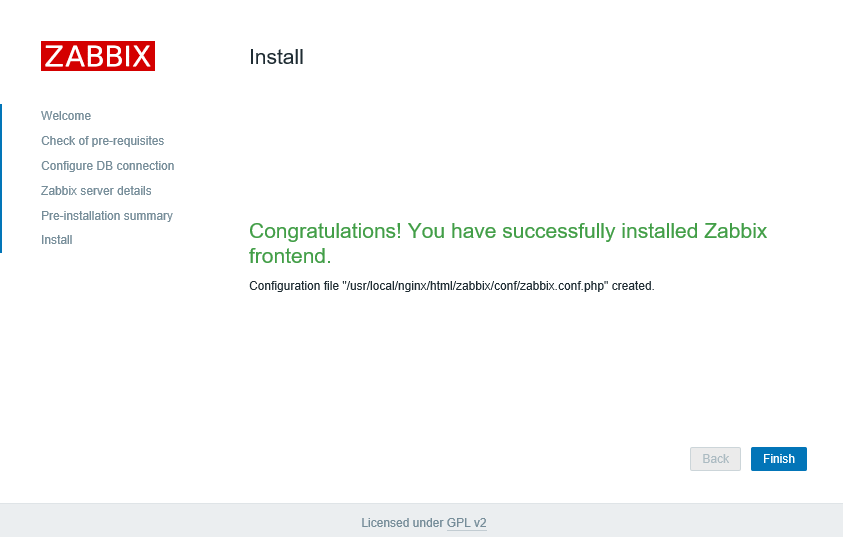
下一步输入zabbix服务器的主机名或主机IP地址和端口号，以及安装名称（可选），这里直接选默认就好



下一步确认前面输入的信息



下一步安装成功



点击Finish安装完成

* + - * 1. 汉化

更改zabbix默认语言为简体中文、替换监控图像上系统默认的中文字体

1. 修改系统配置文件，让web页面支持简体中文显示

编辑修改

vim /usr/local/nginx/html/zabbix/include/locales.inc.php

'zh\_CN' => array('name' => \_('Chinese (zh\_CN)'), 'display' => false),

修改为

'zh\_CN' => array('name' => \_('Chinese (zh\_CN)'), 'display' => true),

保存退出

:wq

1. 替换监控图像上系统默认的字体

默认字体不支持中文，如果不替换，图像上会显示乱码

在Windows系统中的C:\Windows\Fonts目录中复制出一个中文字体文件，例如msyh.ttf

把字体文件msyh.ttf上传到zabbix站点根目录下fonts文件夹中

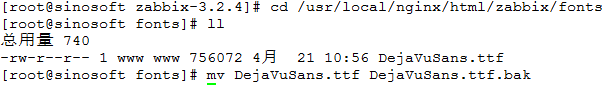


本文附件中已经提供好字体文件，直接上传至该目录

cd /usr/local/nginx/html/zabbix/fonts

备份默认的字体文件：DejaVusSans.ttf.bak

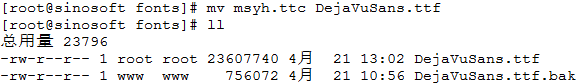
mv DejaVuSans.ttf DejaVuSans.ttf.bak



修改msyh.ttf名称为DejaVusSans.ttf

mv msyh.ttc DejaVuSans.ttf

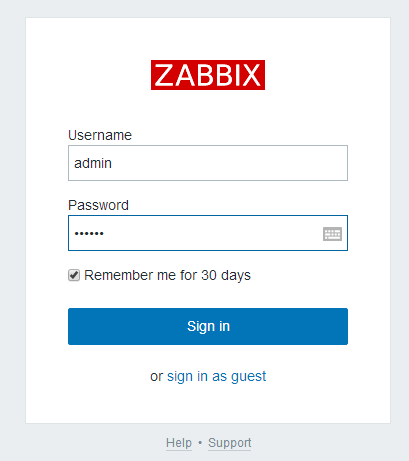
ll

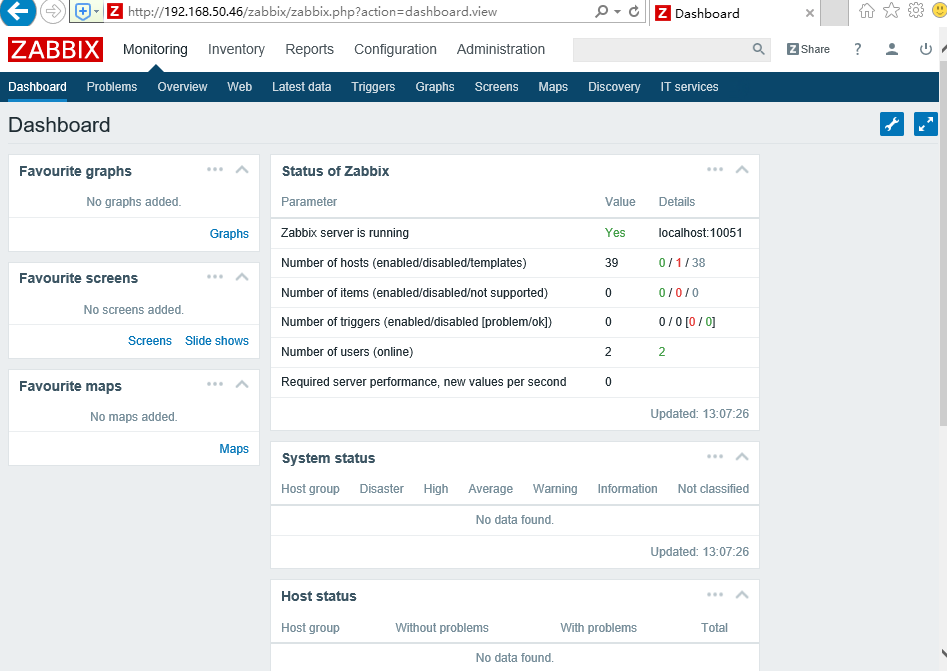


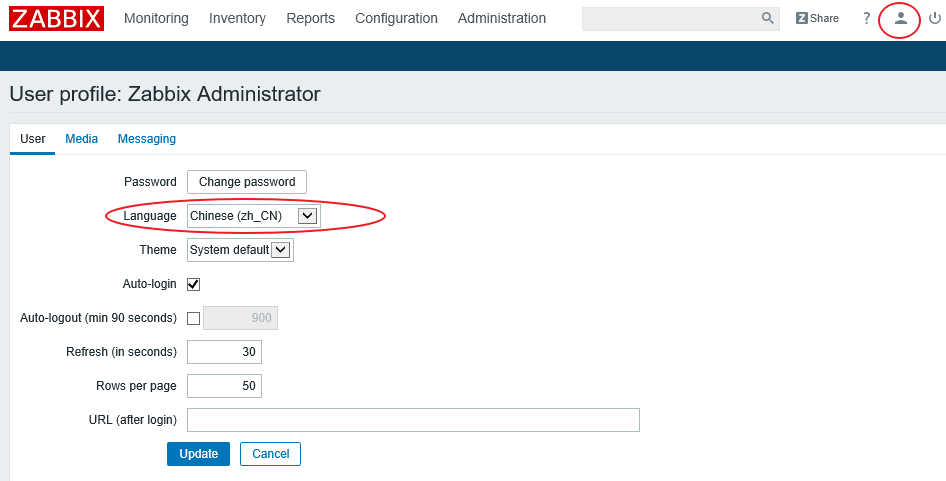
3、在浏览器中打开：

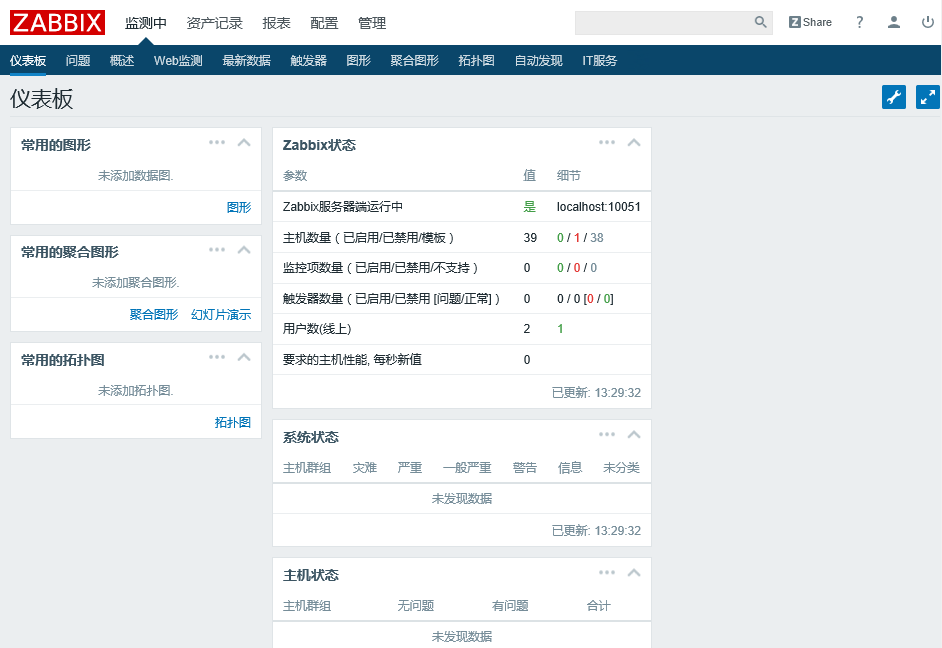
<http://192.168.50.46/zabbix>

用户名是admin 密码是zabbix









此时汉化成功

#### 客户端

##### Linux端

**安装前先执行前面安装准备步骤。**

###### 安装依赖包

切换至软件包存放目录

cd /usr/local/src

解压字符集转换库安装包

tar -zxvf libiconv-1.15.tar.gz

切换至解压出来的安装目录内

cd libiconv-1.15

对即将安装的软件进行配置，检查当前的环境是否满足要安装软件的依赖关系

./configure --prefix=/usr/local

编译

make

安装

make install

添加软连接

ln -s /usr/local/lib/libiconv.so.2 /usr/lib/libiconv.so.2

使配置立即生效

/sbin/ldconfig

###### 创建用户

创建用户组zabbix

groupadd zabbix

创建用户zabbix，并且把用户zabbix加入到用户组zabbix中

useradd zabbix -g zabbix -s /bin/false

执行安装

切换至软件包存放目录

cd /usr/local/src

解压安装包

tar zxvf zabbix-3.2.5.tar.gz

切换至解压出来的安装目录内

cd zabbix-3.2.5

对即将安装的软件进行配置，检查当前的环境是否满足要安装软件的依赖关系

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

编译

make

安装

make install

添加系统软连接

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

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

添加服务端口

添加zabbix服务对应的端口

vim /etc/services

编辑services文件，输入G跳转至最后一行，在最后一行下面添加以下代码

zabbix-agent 10050/tcp # Zabbix Agent

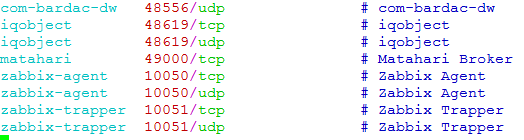
zabbix-agent 10050/udp # Zabbix Agent

zabbix-trapper 10051/tcp # Zabbix Trapper

zabbix-trapper 10051/udp # Zabbix Trapper

保存退出

:wq



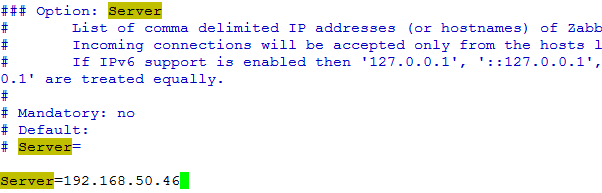
修改配置文件

修改zabbix\_agentd配置文件

vim /usr/local/zabbix/etc/zabbix\_agentd.conf

输入/ Server快速定位到文档相关内容部分，指定服务器IP

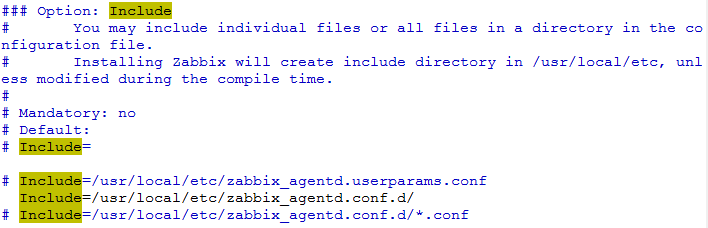
Server=192.168.50.46



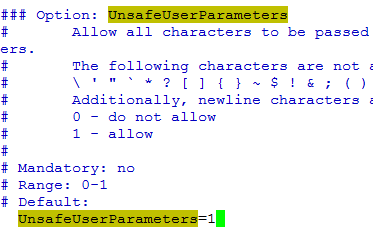
备注：192.168.50.46是Zabbix服务端IP地址

输入/ Include快速定位到文档相关内容部分，修改zabbix运行脚本存放目录

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



输入/ UnsafeUserParameters快速定位到文档相关内容部分，设置UnsafeUserParameters=1 ，表示启用自定义key



保存退出

:wq

根据刚才的配置创建zabbix运行脚本

mkdir -p /usr/local/etc/zabbix\_agentd.conf.d

添加开机启动

复制客户端启动脚本到系统启动目录

cp /usr/local/src/zabbix-3.2.5/misc/init.d/fedora/core/zabbix\_agentd /etc/rc.d/init.d/zabbix\_agentd

添加客户端启动脚本执行权限

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

添加开机启动

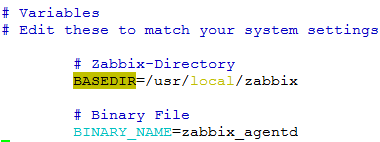
chkconfig zabbix\_agentd on

编辑客户端配置文件

vim /etc/rc.d/init.d/zabbix\_agentd

修改zabbix安装目录

BASEDIR=/usr/local/zabbix/



保存退出

:wq

启动服务

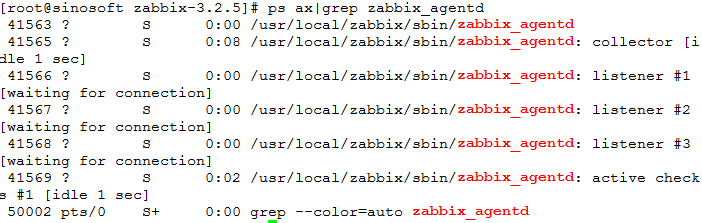
启动Zabbix客户端

service zabbix\_agentd start



检查Zabbix客户端是否正常运行

ps ax|grep zabbix\_agentd



检查Zabbix客户端是否正常运行

netstat -utlnp | grep zabbix



测试通信

测试Zabbix客户端与Zabbix服务端通信是否正常

以下代码在Zabbix服务端执行

/usr/local/zabbix/bin/zabbix\_get -s192.168.50.35 -p10050 -k"system.uptime"



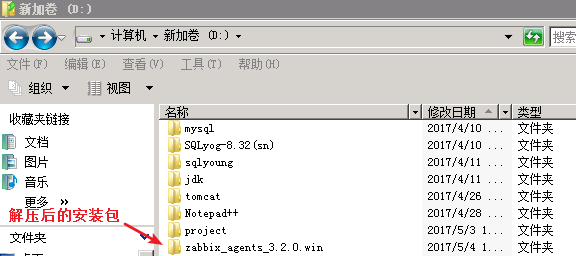
86883

有数据显示说明通信正常

备注：192.168.50.35是Zabbix客户端IP地址

##### Windows端

准备zabbix\_agentd监控客户端软件安装包(windows操作系统客户端)



解压zabbix\_agents\_3.2.0.win.zip 压缩包，里面有两个文件夹，一个是bin文件夹，另一个是conf文件夹。

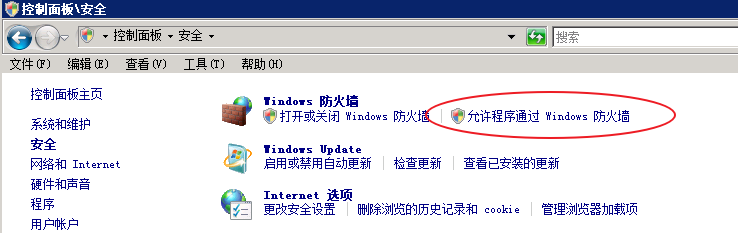
Bin文件夹里面有两个文件夹，一个是win32文件夹里存放zabbix\_agentd安装程序应用于windows 32位操作系统，一个是win64文件夹里存放zabbix\_agentd安装程序应用于windows 64位操作系统。

Conf文件夹里存放是配置文件zabbix\_agentd.win.conf

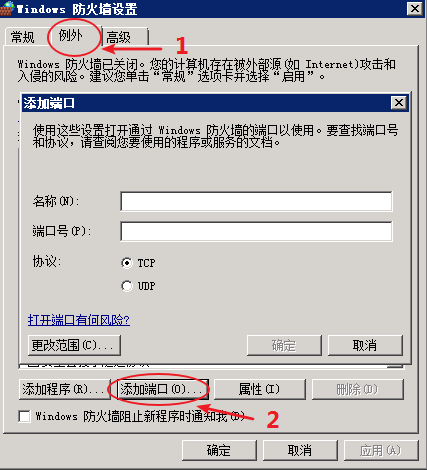
开放端口

关闭监控主机windows server 2008防火墙或防火墙入放行zabbix\_agentd客户端口号 10050 (TPC/UDP)。

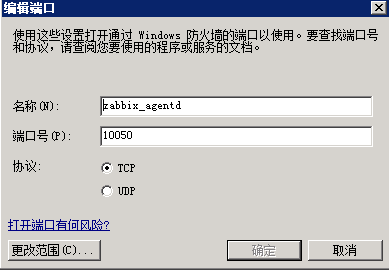
找到控制面板-----安全--允许程序通过Windows防火墙



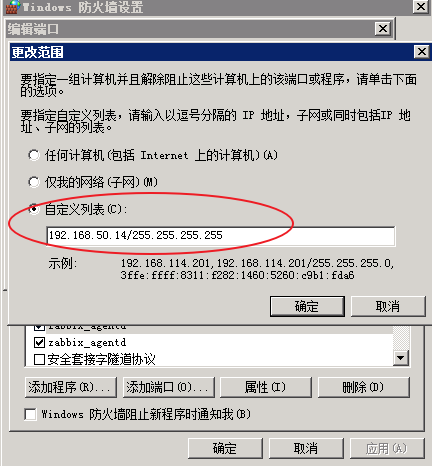
进去选择例外选项卡-----添加端口



名称输入zabbix\_agentd，端口号输入10050,协议选择TCP



选择更改范围，只允许zabbix\_server端IP 192.168.50.14通过10500端口，增加安全性



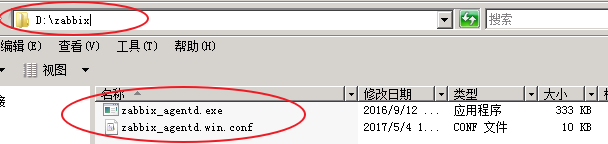
名称输入zabbix\_agentd，端口号输入10050,协议选择UTP



重复上面操作，选择更改范围，只允许zabbix\_server端IP 192.168.50.14通过10500端口，增加安全性

创建目录

在windows server 2008 操作系统 C盘目录下创建一个zabbix文件夹，把刚下载的zabbix\_agentd压缩包里的win32位文件夹的zabbix.agentd.exe安装程序和conf文件夹zabbix\_agentd.win.conf复制到windows server 2008操作系统C盘创建的zabbix文件夹里。(这里的windows server 2008 操作系统是32位的，所以拷贝的是32位安装程序，文件夹创建路径可以自定义，但后面zabbix\_agentd客户端运行安装命令需要根据创建的路径运行)



编辑配置文件

右键以文本格式编辑zabbix\_agentd.win.conf 配置文件。

修改下面几项：

EnableRemoteCommands=1 #允许在本地执行远程命令

LogRemoteCommands=1 #执行远程命令是否保存操作日志

Server = 192.168.50.14 #填写zabbix服务器IP地址

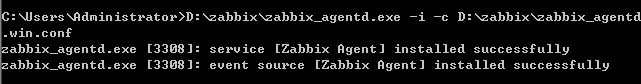
Hostname=Nginx #zabbix\_agent客户端计算机名 (被监控主机)

开始安装

桌面----开始----运行----输入cmd 打开DOS命令窗口----输入以下命令进行zabbix客户端安装。(必须要以管理员身份运行打开DOS命令窗口)

安装zabbix客户端

D:\zabbix\zabbix\_agentd.exe -i -c D:\zabbix\zabbix\_agentd.win.conf



启动zabbix服务

D:\zabbix\zabbix\_agentd.exe -s -c D:\zabbix\zabbix\_agentd.win.conf



问题解决

如果上面的都启动成功了，请忽略下面的内容

有一些小伙伴由于路径搞错，导致服务起不来，需要卸载服务再次进行安装。

首先需要在服务界面将Zabbix Agent服务给停掉

然后进行删除操作

执行

D:\zabbix\zabbix\_agentd.exe -d -c D:\zabbix\zabbix\_agentd.win.conf

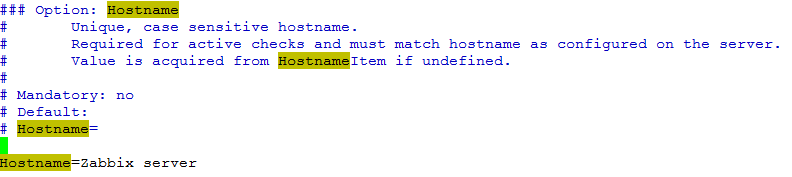
删除成功，接着再次从第一步开始。

* 1. 监控配置
     1. 监控Linux机器

这里以Zabbix服务端机器为例，因为前面安装配置过程中服务端的zabbix机器上也安装了agent客户端，可以监控服务端机器本机的性能。进入Zabbix后台管理界面按照示图先选择配置----主机----创建主机

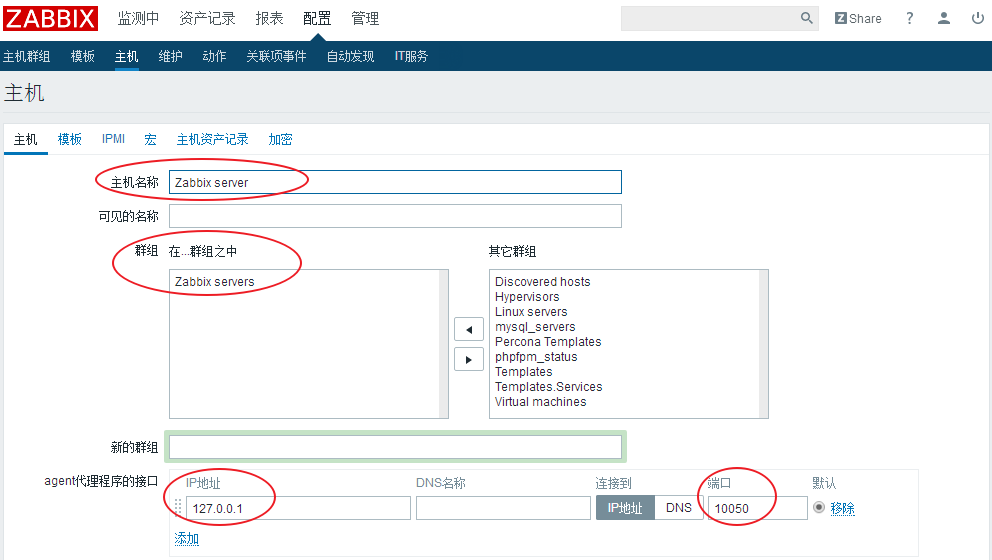


主机名称要与/usr/local/zabbix/etc/zabbix\_agentd.conf配置文件中的Hostname标签一样，这里填Zabbix server



群组选择Zabbix servers

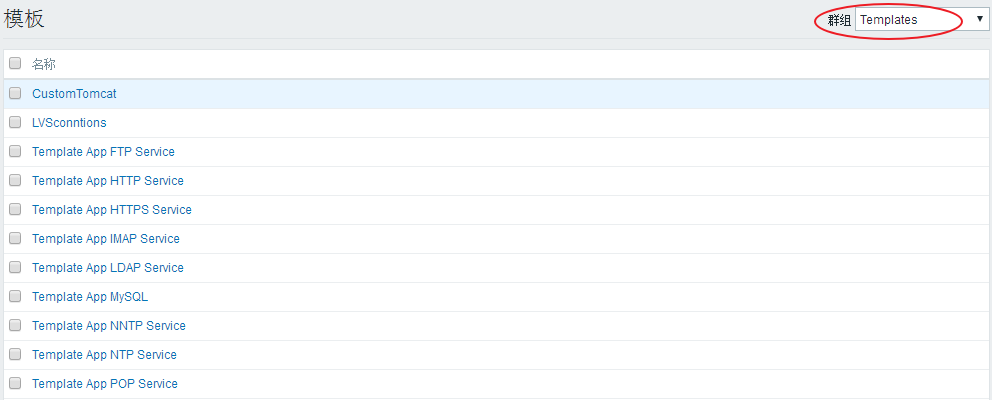
IP地址填127.0.0.1，，端口填zabbix的默认端口10050



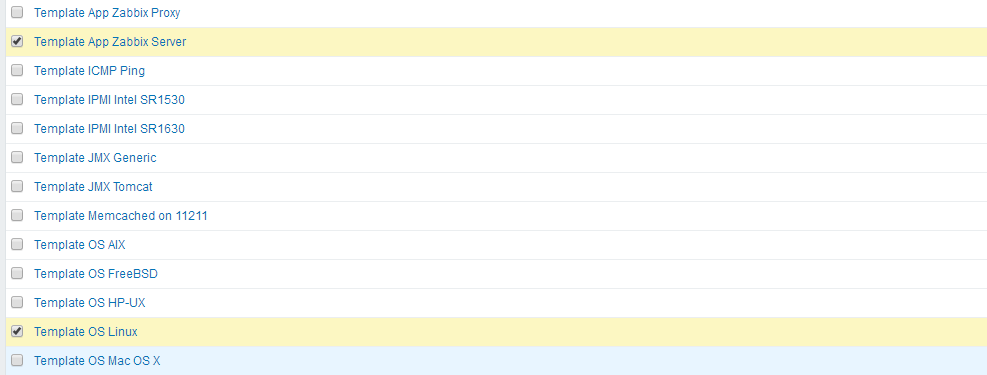
选择模版，Zabbix默认自带模版含有些常见基本监控项，可以拿来直接用，如果想监控更多细节选项则需要自己写模版。



在右上角选择Templates群组，因为自带的模版都放在这个群组里面，下面内容则是自带模版，根据名称可知道大概监控范围。



找到Template App Zabbix Server、Template OS Linux这两个模版并打上勾，然后点击最下面的选择按钮



先点击上面的添加按钮，再点击下面的添加按钮



添加之后会自动跳转到主机菜单列表，这时可以看得到应用集、监控项、触发器、图形、自动发现、Web检测等选项



配置好后点击监测中----图形----选择群组为Zabbix serves----选择主机为Zabbix server，图形选项是Zabbix服务端机器的监控项，选择任意选项就可以看到对应监控实时图形信息



* + 1. 常见监控项

常用监控项翻译

Processor load (15 min average per core) 处理器负载（每芯平均15分钟）

Processor load (5 min average per core) 处理器负载（每芯平均5分钟）

Processor load (1 min average per core) 处理器负载（每芯平均1分钟）

Interrupts per second 每秒内的设备中断数

Interrupts值包括硬Interrupts和软Interrupts，硬Interrupts会对系统性能带 来更多的不利影响。高的Interrupts值指出系统可能存在一个软件的瓶颈，可能是内核或者驱动程序。注意Interrupts值中包括CPU时钟导 致的中断（现代的xServer系统每秒1000个Interrupts值）。

CPU softirq time CPU的软中断时间

CPU system time 系统cpu时间：进程运行时，在系统区执行的时间，如（write，read等系统调用），运行的地方位于系统内存中。 系统态使用的cpu时间比

CPU user time 用户cpu时间：进程运行时，在用户区执行的时间。 用户态使用的cpu时间比

CPU steal time CPU的空闲量 虚拟机偷取时间

CPU nice time 用做nice加权的进程分配的用户态cpu时间比

CPU iowait time 详解地址： http://blog.csdn.net/lhf\_tiger/article/details/8926232

CPU interrupt time CPU的中断时间

CPU idle time 空闲的cpu时间比

Context switches per second CPU上下文切换的次数和时间（context switch） 详解地址：http://iamzhongyong.iteye.com/blog/1895728

Total swap space 交换空间总大小

Total memory 总内存

Free swap space in % swap交换空间空闲率

Free swap space 空余交换空间大小

Available memory 可用内存

System uptime 系统运行时间

Incoming network traffic on ens33 在ens33网卡中传入的网络流量

Outgoing network traffic on ens33 在ens33网卡中输出网络流量

Maximum number of processes 最大进程数

Maximum number of opened files 查看当前系统的最大文件打开数

Number of processes 进程数

Number of running processes 运行的进程数

* + 1. 监控MySQL服务器

先按照[Linux端](#_Linux端)安装步骤安装配置并启动zabbix\_agentd客户端

后续补充.....

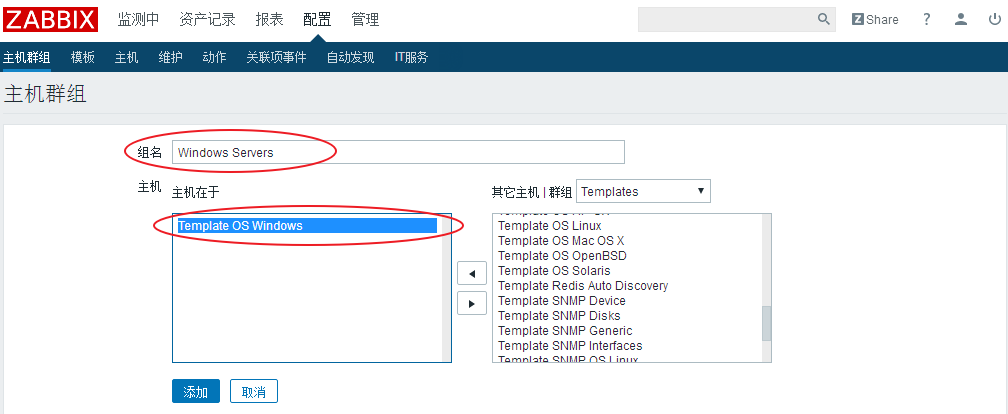
* + 1. 监控Windows机器

先按照[Windows端](#_windows端)安装步骤安装配置并启动zabbix\_agentd客户端，

然后在zabbix的后台管理界面中选择配置----主机群组----创建主机群组



组名输入Windows Servers，将下面右边的模版选项Templates----Templates OS Windows添加到左边的选项框中，点击下面添加按钮，然后页面会自动返回到主机群组列表，显示群组已添加信息

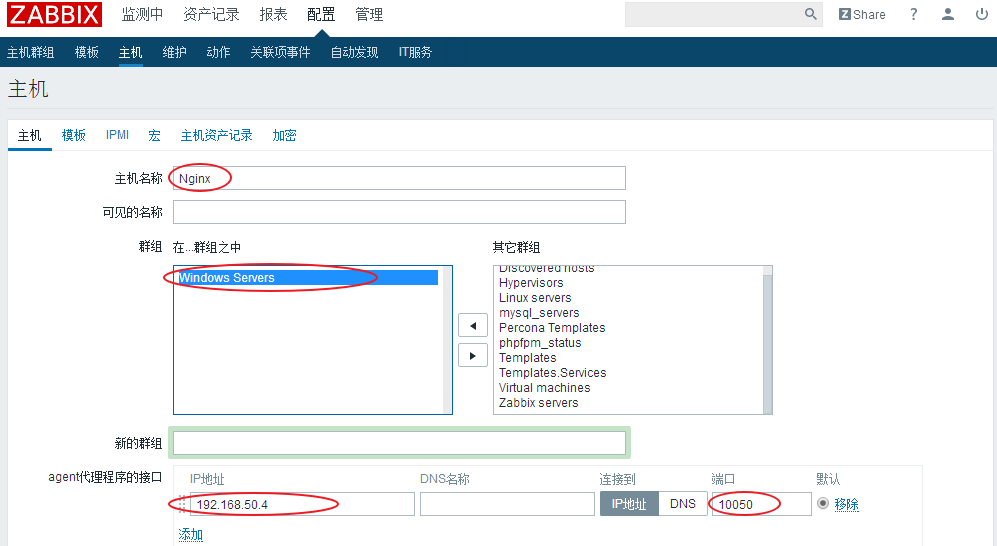


选择配置----主机----创建主机



主机名称要与zabbix\_agentd.win.conf配置文件中的Hostname标签一样，这里填Nginx，

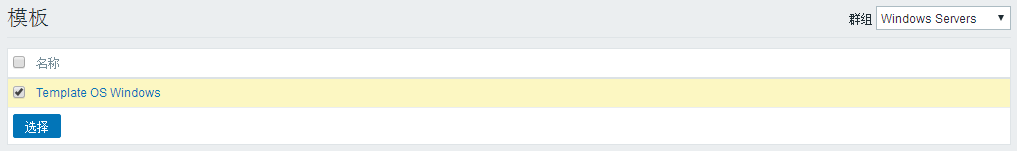
群组选择Windows Servers，IP地址填被监控机192.168.50.4，因为前面配置中服务端的zabbix机器上也安装了agent客户端，可以监控服务端机器本机的性能，端口填zabbix的默认端口10050



选择模版，Zabbix默认自带模版含有些常见基本监控项，可以拿来直接用，如果想监控更多细节选项则需要自己写模版。



在右上角选择Windows Servers群组，找到Template OS Windows这个模版并打上勾，然后点击最下面的选择按钮



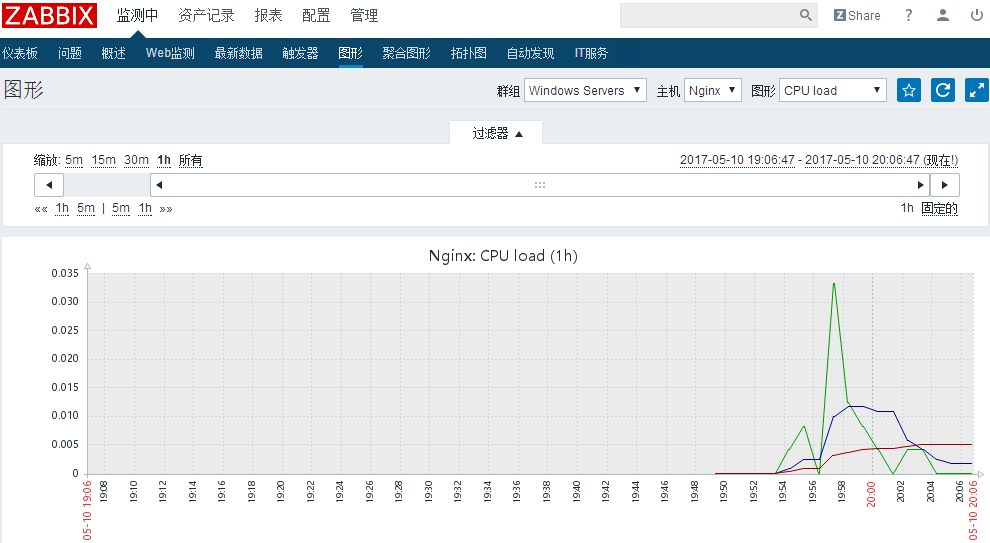
先点击上面的添加按钮，再点击下面的添加按钮



添加之后会自动跳转到主机菜单列表，选择Windows Server群组，可以看得到应用集、监控项、触发器、图形、自动发现、Web检测等选项



配置好后点击监测中----图形----选择群组为Windows Servers----选择主机为Nginx，选择任意图形选项就可以看到对应监控项实时图形信息



* + 1. 监控Web页面

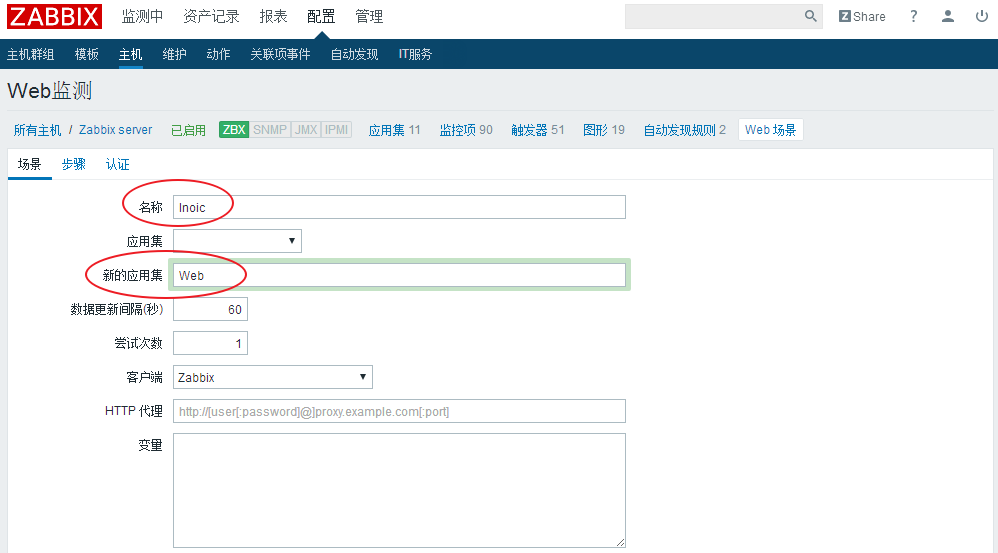
在zabbix的后台管理界面中选择配置----主机----Zabbix server主机----Web监测



选择创建Web场景



名称自己起，这里写的是要监控的项目名字Inoic,新的应用集的名字也是自己起，这里写的Web，客户端选择默认的Zabbix



选择步骤----添加



然后在弹出来的窗口中填写相关信息

名称自己起，这里填的是Inoic Web，URL填要监控的Web页面具体地址，这里填的是最近开发的一个项目地址，要求的状态码填写200，表示页面状态正常，填写完成后点更新按钮





这会添加的Web场景已经显示出来，选择监测中----Web监测----Inoic



可以看到Web界面监控详细信息了



