**构建lnmp平台**

[1. Nginx源码安装 3](#_Toc107833895)

[1.1. 解压nginx源码包 3](#_Toc913574172)

[1.2. 安装nginx的依赖包 3](#_Toc2036792500)

[1.3. 添加ssl与status模块. 3](#_Toc429055132)

[1.4. 生成快链接 3](#_Toc1754073197)

[1.5. 使用openssl工具生成ssl的key文件与公钥文件 3](#_Toc1246977006)

[1.6. 修改nginx配置文件，使其支持ssl与status模块 4](#_Toc534825487)

[1.7. 启动服务，并客户机web界面测试 4](#_Toc1531890026)

[2. 安装PHP源码包 5](#_Toc341767788)

[2.1. 安装php模块的依赖包 5](#_Toc1890612104)

[2.2. 解压PHP源码包 5](#_Toc1397941213)

[2.3. 添加模块. 5](#_Toc1225019264)

[2.4. 生成快链接 6](#_Toc906163025)

[2.5. 修改配置文件 6](#_Toc669412446)

[2.6. 起服务 6](#_Toc1140944656)

[2.7. 修改nginx配置文件，使nginx支持php． 6](#_Toc802525610)

[2.8. 编写php测试文件,并通过web测试． 7](#_Toc1027015094)

[2.9. 手动添加新模块. 8](#_Toc1426245884)

[2.9.1. 查看php模块命令: 8](#_Toc834877283)

[2.9.2. 查看php加载模块的路径 8](#_Toc714172106)

[2.10. 在php中增加ldap模块 8](#_Toc530456060)

[2.10.1. 在php.ini中添加ldap模块的路径 8](#_Toc1626669521)

[3. 安装mysql源码包 9](#_Toc534794928)

[3.1. 创建mysql用户及mysql的文件路径 9](#_Toc22801147)

[3.2. 安装依赖包 9](#_Toc2084079109)

[3.2.1. 安装cmake源码编译包 9](#_Toc1348471573)

[3.2.2. mysql源码包通过cmake编译 9](#_Toc12645400)

[3.3. 设置目录权限 10](#_Toc1118771594)

[3.4. 修改mysql配置文件my.cnf 10](#_Toc595545880)

[3.5. 初始化mysql 10](#_Toc140971497)

[3.6. 设置环境变量 11](#_Toc1885709247)

[3.7. 启动/关闭服务 11](#_Toc703379776)

[3.8. 登录mysql数据库测试，默认无密码． 11](#_Toc1054545669)

[3.9. mysq启动时utf8报错，解决方案 11](#_Toc1775018099)

[4. 通过zabbix源码,提供zabbix服务. 11](#_Toc1132434908)

[4.1. 安装依赖包 11](#_Toc661135219)

[4.2. 源码编译 12](#_Toc874511458)

[4.3. 验证zabbix服务是否安装成功. 12](#_Toc1667260395)

[4.4. 初始化Zabbix(创建数据库与数据库账户） 12](#_Toc45541597)

[4.4.1. 创建数据库，支持中文字符集 12](#_Toc1216279246)

[4.4.2. 创建可以访问数据库的账户与密码 12](#_Toc1410388852)

[4.4.3. 刚刚创建是空数据库，zabbix源码包目录下，有模板． 12](#_Toc1443482810)

[4.4.4. 上线Zabbix的Web页面 13](#_Toc293814862)

[4.4.5. 修改Zabbix\_server配置文件，设置数据库相关参数，启动Zabbix\_server服务 13](#_Toc169068229)

[4.4.6. 修改Zabbix\_agent配置文件，启动Zabbix\_agent服务 14](#_Toc2112895256)

[4.5. 初始化Zabbix 14](#_Toc1434759519)

[4.5.1. 初始化准备安装图形化软件（在安装php时已安装好了）,并修改php.ini的相关参数．并重启php-fpm服务. 14](#_Toc971593840)

[4.5.2. Web登录zabbix测试,发现没有关联到mysql数据库． 14](#_Toc992426703)

[4.6. 通过snmp监控网络设备 15](#_Toc713521755)

[4.6.1. 网络设备开启snmp读服务，并保证是互通． 15](#_Toc1806471123)

[4.6.2. 监控服务上安装yum -y install net-snmp-utils包 15](#_Toc1706598809)

[4.6.3. zabbix,web管理界面添加已开启snmp服务的网络设备 16](#_Toc1243977815)

准备工作centos7.5系统,并配置好ip.yum．准备好nginx.php.mysql.zabbix的源码包.分别为：mysql-5.5.49.tar.gz　　php-5.5.35.tar.gz　cmake-3.15.0-rc1.tar.gz

nginx-1.12.2.tar.gz　zabbix-3.4.4.tar.gz

# Nginx源码安装

## 解压nginx源码包

tar -zxf nginx-1.12.2.tar.gz

## 安装nginx的依赖包

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

## 添加ssl与status模块.

进入nginx的安装目录下，并通过configure来编译添加ssl与status模块.

cd nginx-1.12.2/

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

make && make install

## 生成快链接

ln -s /usr/local/nginx/sbin/nginx /usr/sbin/

## 使用openssl工具生成ssl的key文件与公钥文件

进入nginx的程序目录的配置目录下，/usr/local/nginx/conf下，并通过openssl工具生成ssl的key文件与 cert.pem公钥文件

cd /usr/local/nginx/conf/

openssl genrsa > cert.key

openssl req -new -x509 -key cert.key > cert.pem

## 修改nginx配置文件，使其支持ssl与status模块

server {

listen 443 ssl;

server\_name localhost;

ssl\_certificate cert.pem;

ssl\_certificate\_key cert.key;

ssl\_session\_cache shared:SSL:1m;

ssl\_session\_timeout 5m;

ssl\_ciphers HIGH:!aNULL:!MD5;

ssl\_prefer\_server\_ciphers on;

location / {

root html;

index index.html index.htm;

}

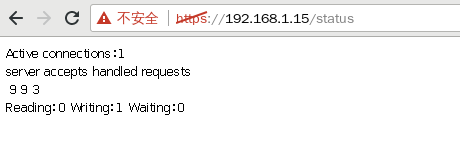
location /status {

stub\_status on;

}

## 启动服务，并客户机web界面测试

Nginx



# 安装PHP源码包

## 安装php模块的依赖包

yum -y install libxml2-devel bzip2-devel libcurl-devel libjpeg-turbo libjpeg-turbo-devel libjpeg.x86\_64 freetype-devel gd-devle　libpng-devel autoconf

下面写的是源码的对应包：

PHP和各个组件包

libxml2-2.6.30.tar.gz             xml c语言版的解析器

libmcrypt-2.5.8.tar.gz           加密算法扩展库

zlib-1.2.3.tar.gz                    压缩图像组件

gd-2.0.35.tar.gz                    处理图片的API

autoconf-2.61.tar.gz             自动配置源代码的shell脚本

freetype-2.3.5.tar.gz             操作字体的函数库

libpng-1.2.26.tar.gz               解析PNG图象格式的库

jpegsrc.v6b.tar.gz

ZendOptimizer-3.3.0a-linux-glibc21-i386.tar.gz           zend加速器

## 解压PHP源码包

tar -zxf php-5.5.35.tar.gz

cd php-5.5.35/

## 添加模块.

./configure --prefix=/usr/local/product/php-5.5.35 --with-config-file-path=/usr/local/product/php-5.5.35/etc --with-bz2 --with-curl --enable-ftp --enable-sockets --disable-ipv6 --with-gd --with-jpeg-dir=/usr/local --with-png-dir=/usr/local --with-freetype-dir=/usr/local --enable-gd-native-ttf --with-iconv-dir=/usr/local --enable-mbstring --enable-calendar --with-gettext --with-libxml-dir=/usr/local --with-zlib --with-pdo-mysql=mysqlnd --with-mysqli=mysqlnd --with-mysql=mysqlnd --enable-dom --enable-xml --enable-fpm --with-libdir=lib64 --enable-bcmath

make -j 4 && make install

## 生成快链接

ln -s /usr/local/product/php-5.5.35/ /usr/local/php

## 修改配置文件

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

cd /usr/local/php/etc/

cp php-fpm.conf.default php-fpm.conf

vim php.ini

877 date.timezone = Asia/Shanghai //设置时区

383 max\_execution\_time = 300 //最大执行时间，秒

671 post\_max\_size = 32M //POST数据最大容量

393 max\_input\_time = 300 //服务器接收数据的时间限制

404 memory\_limit = 128M //内存容量限制

## 起服务

/usr/local/php/sbin/php-fpm

ss -nltup | grep 9000

## 修改nginx配置文件，使nginx支持php．

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

...

location ~ \.php$ {

root html;

fastcgi\_pass 127.0.0.1:9000;

fastcgi\_index index.php;

include fastcgi.conf;

}

...

[root@lnmp01 etc]# nginx -s reload

## 编写php测试文件,并通过web测试．

[root@lnmp01 html]# cat index.php

<?php

echo 'hello world';

?>



## 手动添加新模块.

### 查看php模块命令:

/usr/local/php/bin/php -m

### 查看php加载模块的路径

usr/local/php/bin/php -i | grep -i extension\_dir

## 在php中增加ldap模块

php源码中自带了模块路径在php-\*\*/ext/中

cd php-5.5.35/ext/ldap/

使用源码中的phpize命令在ldap的文件中生成configure文件(前提条件：安装了autoconf的软件包)

/usr/local/php/bin/phpize

yum -y install openldap openldap-devel(安装ldap的依赖关系)

cp -frp /usr/lib64/libldap\* /usr/lib/　（为了解决./configure时产生的报错）

./configure --with-php-config=/usr/local/php/bin/php-config

make

make test

make install

### 在php.ini中添加ldap模块的路径

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

...

extension=/usr/local/php/lib/php/extensions/no-debug-non-zts-20121212/ldap.so

..（＊注意：路径需要我们自己去检查有两个．）

重启php-fpm服务,查看已加载的模块

/usr/local/php/bin/php -m | grep ldap

# 安装mysql源码包

## 创建mysql用户及mysql的文件路径

useradd -s /sbin/nologin mysql

mkdir -pv /usr/local/mysql #安装路径

mkdir -pv /usr/local/mysql/data  #数据文件存放位置

[root@lnmp01 mysql]# grep mysql /etc/shadow

mysql:!!:18061:0:99999:7:::

[root@lnmp01 mysql]# grep mysql /etc/group

mysql:x:1000:

## 安装依赖包

[root@lnmp01 mysql]# yum -y install gcc\* ncurses-devel

### 安装cmake源码编译包

tar -zxf cmake-3.15.0-rc1.tar.gz

cd cmake-3.15.0-rc1/

./configure

make && make install

### mysql源码包通过cmake编译

cd mysql-5.5.49/

/usr/local/bin/cmake -DCMAKE\_INSTALL\_PREFIX=/usr/local/mysql -DDEFAULT\_CHARSET=utf8

-DENABLED\_LOCAL\_INFILE=1 -DMYSQL\_DATADIR=/usr/local/mysql/data -DWITH\_EXTRA\_CHARSETS=all -DWITH\_READLINE=1 -DWITH\_INNOBASE\_STORAGE\_ENGINE=1 -DMYSQL\_TCP\_PORT=3306 -DDEFAULT\_COLLATION=utf8\_general\_ci -DMYSQL\_UNIX\_ADDR=/tmp/mysqld.sock -DSYSCONFDIR=/etc

注意事项：(cmake替换了./configure)  
重新编译时，需要清除旧的对象文件和缓存信息。  
# make clean  
# rm -f CMakeCache.txt

make

make && install

-DCMAKE\_INSTALL\_PREFIX=/usr/local/mysql          \    #安装路径

-DMYSQL\_DATADIR=/usr/local/mysql/data            \    #数据文件存放位置

-DSYSCONFDIR=/etc                                \    #my.cnf路径

-DWITH\_MYISAM\_STORAGE\_ENGINE=1                   \    #支持MyIASM引擎

-DWITH\_INNOBASE\_STORAGE\_ENGINE=1                 \    #支持InnoDB引擎

-DWITH\_MEMORY\_STORAGE\_ENGINE=1                   \    #支持Memory引擎

-DWITH\_READLINE=1                                \    #快捷键功能

-DMYSQL\_UNIX\_ADDR=/tmp/mysqld.sock        \    #连接数据库socket路径

-DMYSQL\_TCP\_PORT=3306                            \    #端口

-DENABLED\_LOCAL\_INFILE=1                         \    #允许从本地导入数据

-DWITH\_PARTITION\_STORAGE\_ENGINE=1                \    #安装支持数据库分区

-DEXTRA\_CHARSETS=all                             \    #安装所有的字符集

-DDEFAULT\_CHARSET=utf8                           \    #默认字符

-DDEFAULT\_COLLATION=utf8\_general\_ci

## 设置目录权限

cd /usr/local/mysql

chown -R root:mysql .　//把当前目录中所有文件的所有者所有者设为root，所属组为mysql

chown -R mysql:mysql data

## 修改mysql配置文件my.cnf

cp support-files/my-medium.cnf /etc/my.cnf　//将源码中的模板复制到mysql的配置文件中.并修改参数.

vim /etc/my.cnf

basedir = /usr/local/mysql

datadir = /usr/local/mysql/data

user = mysql

tmpdir = /tmp

## 初始化mysql

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

## 设置环境变量

将源码中的mysql管理命令,设置快捷键．

vim /root/.bash\_profile

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

source /root/.bash\_profile

## 启动/关闭服务

cd /usr/local/mysql

./bin/mysqld\_safe --user=mysql & 启动mysql服务

mysqladmin -u root -p shutdown 　关闭mysql服务

## 登录mysql数据库测试，默认无密码．

mysql -uroot

## mysq启动时utf8报错，解决方案

　　报[ERROR] COLLATION 'latin1\_swedish\_ci' is not valid for CHARACTER SET 'utf8'错误，原因可能是源码编译的是有使用了-DDefault-charset=utf8使得默认的字符集和校验规则不匹配。  
在初始化数据库时指定--character-set-server=utf8  
并修改my.cnf  
[client]  
default-character-set=utf8  
[mysqld]  
character-set-server=utf8

# 通过zabbix源码,提供zabbix服务.

## 安装依赖包

yum -y install libcurl-devel net-snmp-devel libevent-devel

## 源码编译

cd Zabbix/

tar -zxf zabbix-3.4.4.tar.gz

cd zabbix-3.4.4/

./configure --enable-server --enable-proxy --enable-agent --with-mysql=/usr/local/mysql/bin/mysql\_config --with-net-snmp --with-libcurl

（＊注意：mysql的mysql\_config的路径,源码安装时需要自己注意）

make && make install

// --enable-server安装部署zabbix服务器端软件

// --enable-agent安装部署zabbix被监控端软件

// --enable-proxy安装部署zabbix代理相关软件

// --with-mysql配置mysql\_config路径

// --with-net-snmp允许zabbix通过snmp协议监控其他设备

// --with-libcurl安装相关curl库文件，这样zabbix就可以通过curl连接http等服务，测试被监控主机服务的状态

## 验证zabbix服务是否安装成功.

ls /usr/local/etc/ zabbix配置文件

　　　ls /usr/local/bin/　zabbix管理程序

ls /usr/local/sbin/ zabbix启动程序

## 初始化Zabbix(创建数据库与数据库账户）

### **创建数据库，支持中文字符集**

　　　create database zabbix character set utf8;

### **创建可以访问数据库的账户与密码**

　　　　grant all on zabbix.\* to zabbix@"localhost" identified by "zabbix";

（注意：mysql密码的复杂度有要求）

### **刚刚创建是空数据库，zabbix源码包目录下，有模板．**

路径在/root/Zabbix/zabbix-3.4.4/database/mysql中

　　　　//使用mysql导入这些数据即可（注意导入顺序）

　　　　mysql -uzabbix -pzabbix zabbix < schema.sql

　　　　mysql -uzabbix -pzabbix zabbix < images.sql

　　　　mysql -uzabbix -pzabbix zabbix < data.sql

### **上线Zabbix的Web页面**

cd Zabbix/zabbix-3.4.4/frontends/php/

cp -a \* /usr/local/nginx/html/

chmod -R 777 /usr/local/nginx/html/\*

### **修改Zabbix\_server配置文件，设置数据库相关参数，启动Zabbix\_server服务**

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

12 ListenPort=10051

38 LogFile=/tmp/zabbix\_server.log //设置日志，仅查看以下即可

85 DBHost=localhost //数据库主机，默认该行被注释

95 DBName=zabbix //设置数据库名称

111 DBUser=zabbix //设置数据库账户

119 DBPassword=zabbix //设置数据库密码，默认该行被注释

useradd -s /sbin/nologin zabbix //不创建用户无法启动服务

zabbix\_server //启动服务

　　　　 ps aux | grep zabbix\_server //查看服务是否有启动进程

### 修改Zabbix\_agent配置文件，启动Zabbix\_agent服务

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

93 Server=127.0.0.1,192.168.2.5 //允许哪些主机监控本机

134 ServerActive=127.0.0.1,192.168.2.5 //允许哪些主机通过主动模式监控本机

145 Hostname=Zabbix server //设置本机主机名

30 LogFile=/tmp/zabbix\_agentd.log //设置日志文件

280 UnsafeUserParameters=1 //是否允许自定义key

zabbix\_agentd //启动监控agent

## 初始化Zabbix

### 初始化准备安装图形化软件（在安装php时已安装好了）,并修改php.ini的相关参数．并重启php-fpm服务.

vim /etc/php.ini

877 date.timezone = Asia/Shanghai //设置时区

383 max\_execution\_time = 300 //最大执行时间，秒

671 post\_max\_size = 32M //POST数据最大容量

393 max\_input\_time = 300 //服务器接收数据的时间限制

404 memory\_limit = 128M //内存容量限制

### Web登录zabbix测试,发现没有关联到mysql数据库．

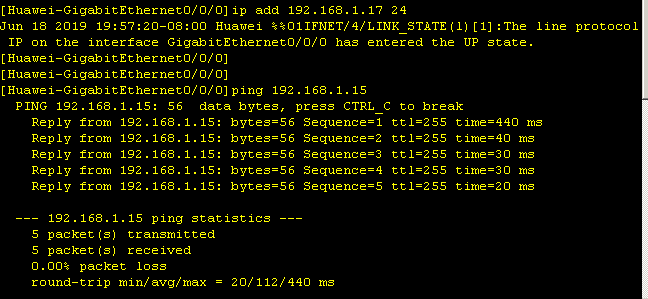
　　解决方案.查看/tmp/zabbix\_server.log中，发现zabbix开发者的关联的mysql.sock文件路径与我们自定义不一样．通过生成软连接来实现．

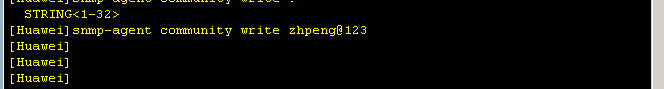
mkdir /var/lib/mysql

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

## 通过snmp监控网络设备

### 网络设备开启snmp读服务，并保证是互通．





### 监控服务上安装yum -y install net-snmp-utils包

　　使用snmpwalk来检查网络是否开启了snmp服务.

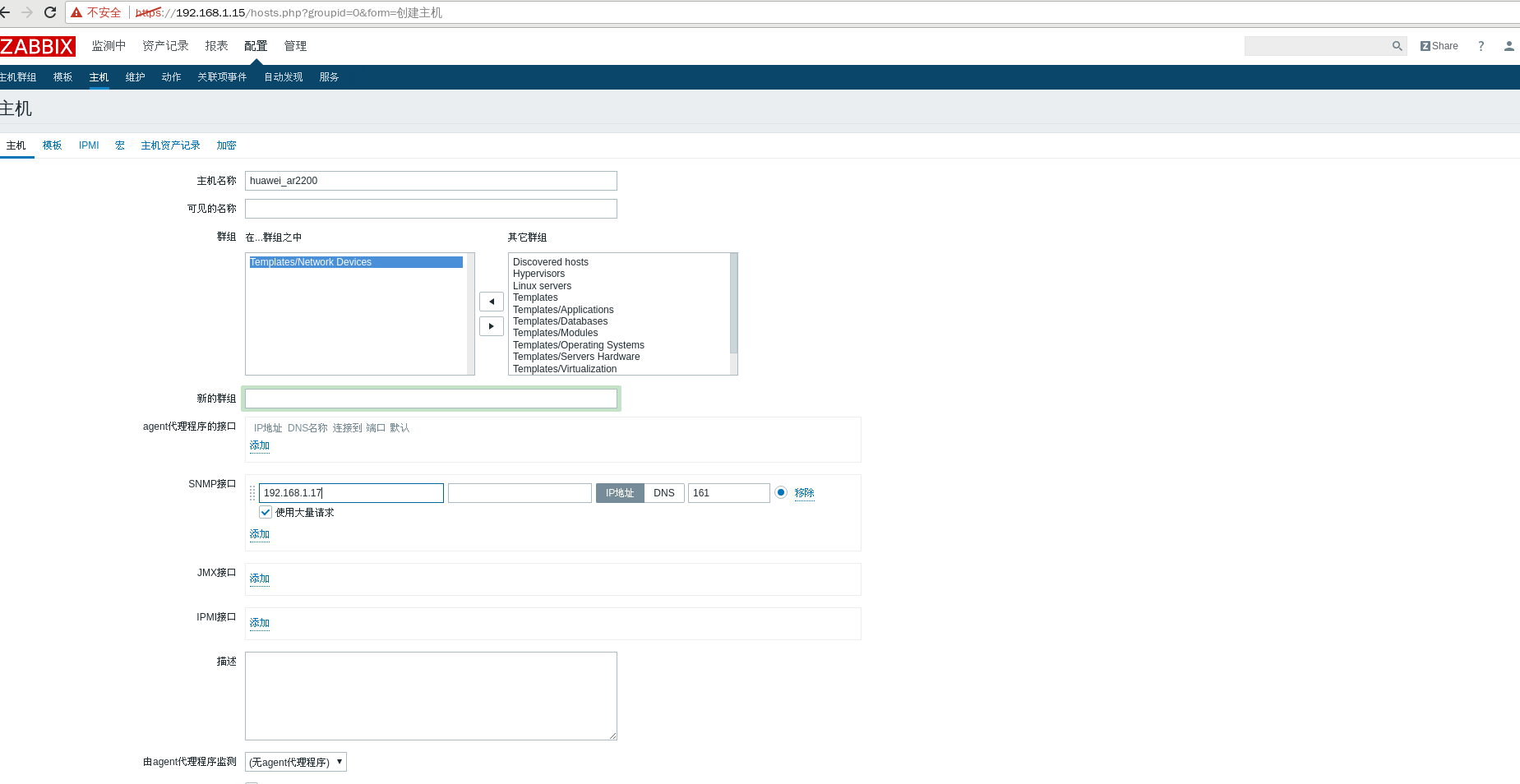
snmpwalk -v 2c -c zhpeng@123 192.168.1.17 模板

### zabbix,web管理界面添加已开启snmp服务的网络设备

在配置>主机中：1.主机名称－自定义

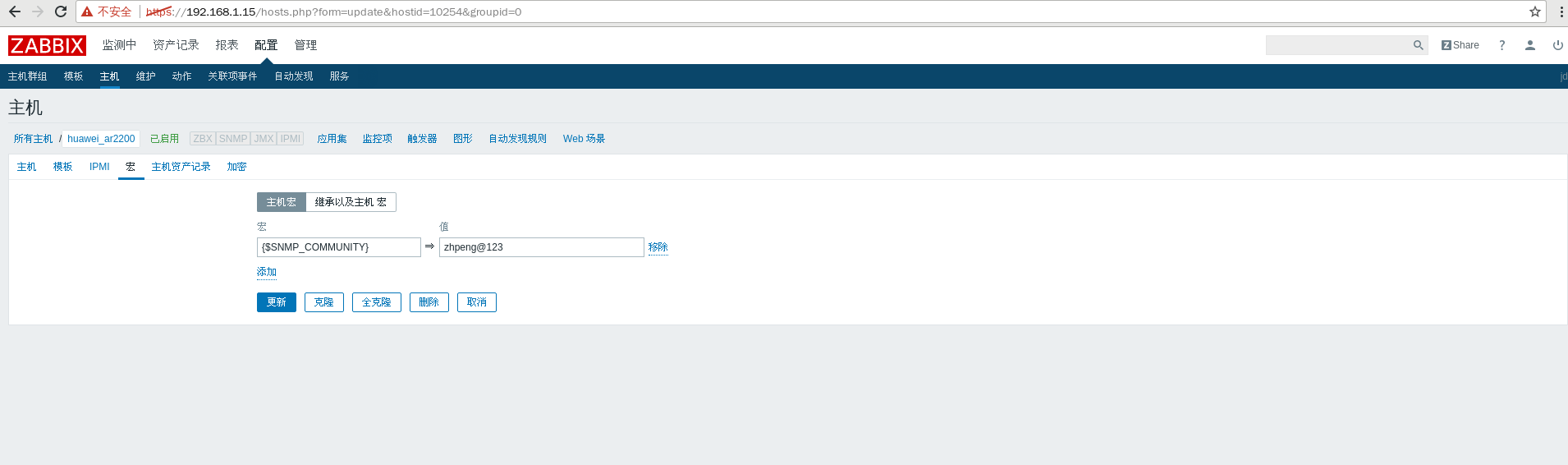
2.群组选择－Templates/Network Devices

3.SNMP接口－网络设置管理IP地址



>使用宏来定义snmp的团体名

1.主机宏{$SNMP\_COMMUNITY} 值-SNMP团体名



　　　　>选择正确的模板来监控，我监控的是华为设置,华为使用的是vrp．

