Zabbix

Lamp/LNMP

1. Zabbix 由以下几个组件部分构成

1. Zabbix server：负责接收agent发送的报告信息的核心组件，所有配置，统计数据及操作数据均由其组织进行。
2. Database storage: 专用于存储所有配置信息，以及由zabbix收集的数据；
3. Web interface：zabbix 的GUI 接口，通常与server 运行在同一台主机上。
4. Proxy：可选组件，常用于分布式监控环境中，代理server收集部分被监控的监控数据并统一发往server 端；
5. Agent： 部署在被监控主机上，负责接收本地数据并发往server 端或者proxy端

2. 进程

默认情况下zabbix 包含5个程序 zabbix\_agent zabbix\_get zabbix\_proxy zabbix\_server

zabbix\_sender

zabbix\_agent

客户端的守护进程，此进程收集客户端数据，例如CPU负载，内存，硬盘使用情况等

Zabbix\_get

Zabbix 工具，单独使用的命令，通常在server或者proxy 端执行获取远程客户端信息的命令，通常用于排错。

Zabbix\_sender

Zabbix 工具，用于发送数据给server或者proxy，通常用于耗时比较长的检查。很多检查非常耗时间，导致zabbix 超时，于是我们在脚本执行完毕之后，使用sender主动提交数据。

Zabbix\_server

Zabbix 服务端守护进程，zabbix\_agent zabbix\_get zabbix\_sender zabbix\_proxy 的数据最终都是提交到server

Zabbix\_proxy

Zabbix 代理守护进程，功能类似server，唯一不同的是它只是一个中转站，它需要把收集到的数据提交/被提交到server 里

1. Zabbix 监控环境中相关术语

主机(host)： 要监控的网络设备，可由IP或DNS 名称指定

主机组(host group): 主机的逻辑容器，可以包含主机和模板，但同一个组织内的主机和模板不能互相连接，主机组通常在给用户或用户组指派监控权限时使用

监控项(item): 一个特定监控指标的相关数据；这些数据来自于被监控对象；item是zabbix 进行数据收集的核心，相对某个监控对象，每个item都由”key”标识 。

触发器(trigger): 一个表达式，用于评估某监控对象的特定item内接收到的数据是否在合理范围内，也就是阀值；接收的数据量大于阀值时，触发器状态将从”ok” 转变 为”Problem”，当数据再次恢复到合理范围，又变为”OK”;

事件(event)： 触发一个值得关注的事情，比如触发器状态转变，新的agent 或重新上线的agent 的自动注册等。

动作(action)：指对于特定事件事先定义的处理方法，如发送通知，何时执行操作；

报警媒介类型(media)：发送通知的手段或者通道，如Email，jabber或者SMS等

模板(template)：用于快速定义被监控主机的预设条目集合，通常包含了item。Trigger。gragh、scrren、application 以及low-level discovery rule；模板可以直接链接至某个主机；

前端(formtemd): zabbix 的web接口

安装lnmp环境

Zabbix server 端安装

yum -y install libxml2-devel net-snmp-devel libcurl-devel mysql-devel

1. 编译安装zabbix server

wget [http://jaist.dl.sourceforge.net/project/zabbix/ZABBIX%20Latest%20Stable/3.2.3/zabbix-3.2.3.tar.gz](http://jaist.dl.sourceforge.net/project/zabbix/ZABBIX Latest Stable/3.2.3/zabbix-3.2.3.tar.gz)

tar xvf zabbix-3.2.3.tar.gz

cd zabbix-3.2.3

./configure --prefix=/application/zabbix --enable-server --enable-agent --with-mysql --enable-proxy --with-net-snmp --with-libcurl --with-libxml2

make && make install

zabbix 编译安装:

同时安装server和agent,并支持数据放入mysql数据中,可使用类似如下配置命令:

./configure --enable-server --enable-agent --with-mysql --enable-ipv6 --with-net-snmp --with-libcurl

如果仅安装server,并支持将数据放入mysql 数据中,可使用类似如下配置:

./configure --enable-server --with-mysql --with-net-snmp

如果仅安装proxy 并支持将数据放入mysql数据库中,可使用如下配置:

./configure –enable-proxy --with-net-snmp --with-mysql –with-ssh2

如果仅安装agent,可使用类似如下配置命令

./configure --enable-agent

然后进行编译

make install

添加系统软连接

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

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

2. 配置zabbix 下的mysql 环境

cd database/mysql/ 目录的sql 文件需要导入到数据库中去。

数据库配置

shell> mysql -uroot -p<password>

mysql> create database zabbix character set utf8 collate utf8\_bin;

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

mysql> quit;

shell> cd database/mysql

shell> mysql -uzabbix -p<password> zabbix < schema.sql

# stop here if you are creating database for Zabbix proxy

shell> mysql -uzabbix -p<password> zabbix < images.sql

shell> mysql -uzabbix -p<password> zabbix < data.sql

3. 配置zabbix server 端

修改zabbix配置文件,配置zabbix\_server

cd /application/zabbix/etc

vim /application/zabbix/etc/zabbix\_server.conf

LogFile=/var/log/zabbix/zabbix\_server.log

DBHost=127.0.0.1

DBName=zabbix

DBUser=zabbix

DBPassword=zabbix

Timeout=4

AlertScriptsPath=/application/zabbix/share/zabbix/alertscripts

LogSlowQueries=3000

4. 添加相关服务和端口到系统配置文件

vim /etc/services

#add zabbix port

zabbix-agent 10050/tcp #Zabbix Agent

zabbix-agent 10050/udp #Zabbix Agent

zabbix-trapper 10051/tcp #Zabbix Trapper

zabbix-trapper 10051/udp #Zabbix Trapper

5. 修改php 的配置文件php.ini 修改为下面的参数

vim /application/php/lib/php.ini

memory\_limit = 128M

post\_max\_size = 16M

upload\_max\_filesize = 2M

max\_execution\_time = 300

max\_input\_time = 300

session.auto\_start = 0

date.timezone=Asia/Shanghai

always\_populate\_raw\_post\_data = -1

6. 添加管理维护脚本

在zabbix 的源码包的解压路径下

cp misc/init.d/fedora/core/zabbix\_server /etc/init.d/

cp misc/init.d/fedora/core/zabbix\_agentd /etc/init.d/

zabbix\_server 启动

systemctl enable zabbix\_server

7. 安装nginx 配置zabbix

yum install nginx -y

mkdir /webserver/zabbix -p 创建站点目录

在zabbix 的源码路径下

cp -rp /opt/zabbix-3.2.3/frontends/php/\* /webserver/zabbix/

nginx 的配置如下

server {

listen 8089;

server\_name ljt.zabbix.com;

root /webserver/zabbix;

index index.html index.htm index.php;

location ~ \.php$ {

fastcgi\_split\_path\_info ^(.+?\.php)(/.\*)$;

if (!-f $document\_root$fastcgi\_script\_name) {

return 404;

}

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

fastcgi\_pass 127.0.0.1:9000;

fastcgi\_index index.php;

include fastcgi\_params;

}

}

nginx 的检查并且启动nginx

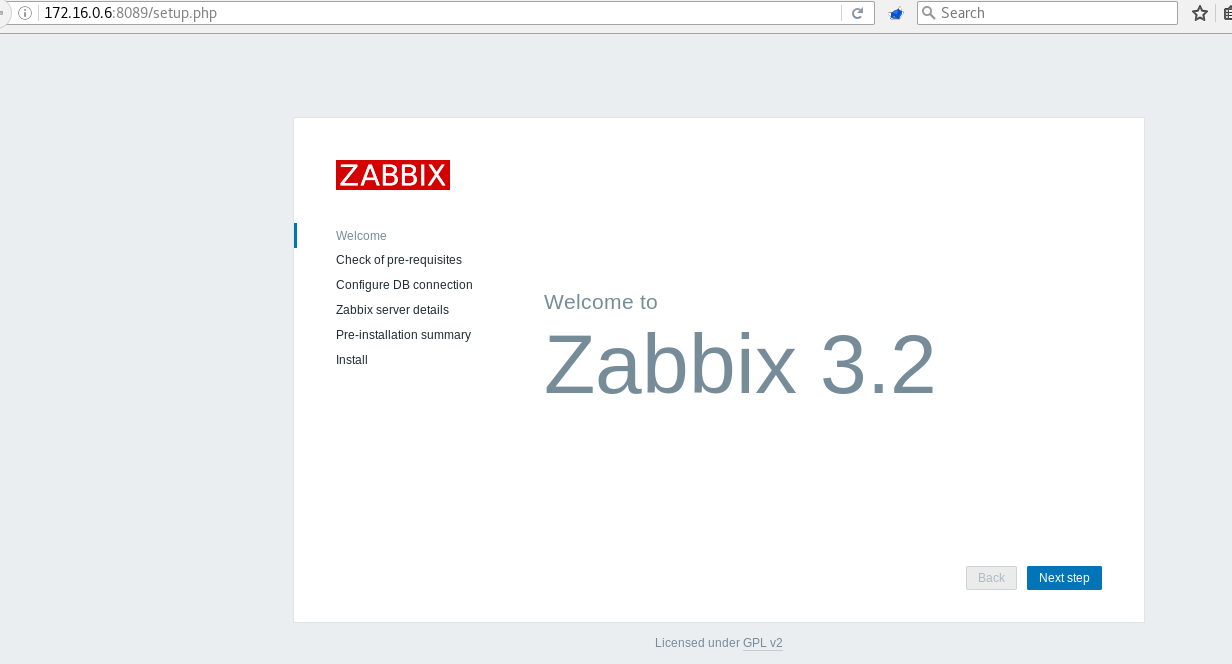
nginx -t

启动nginx

systemctl restart nginx

访问界面

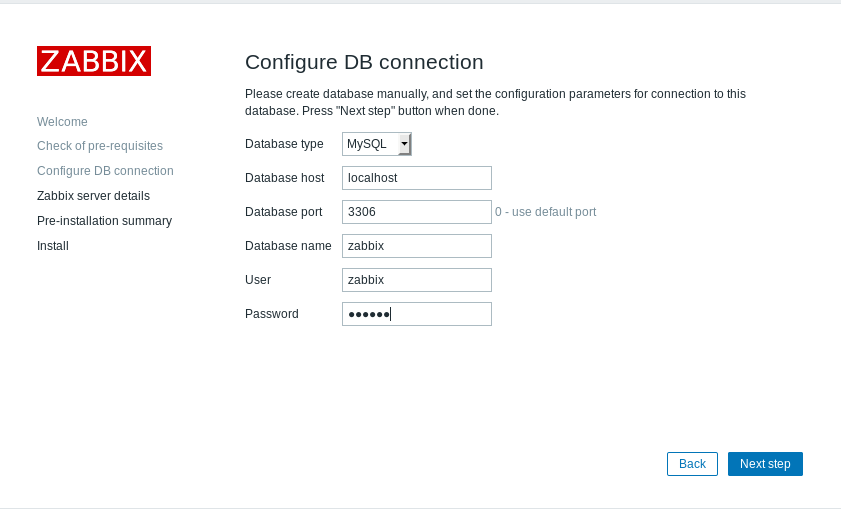
172.16.0.6:8089

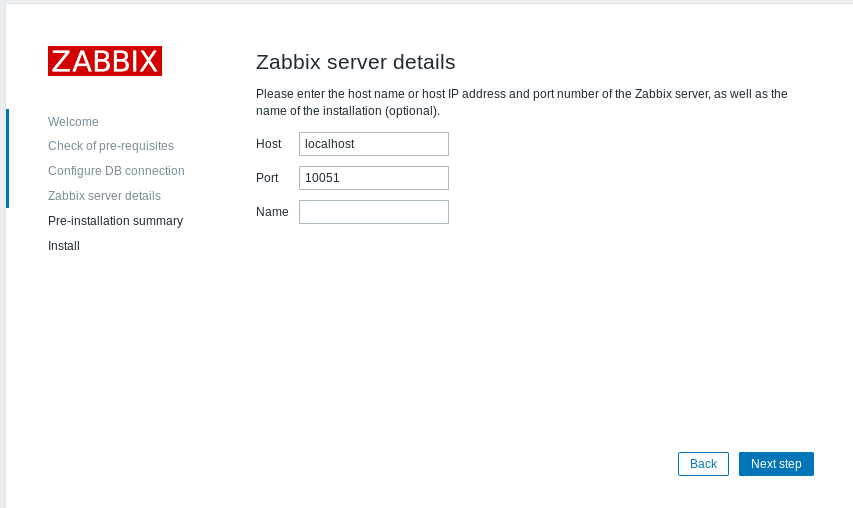


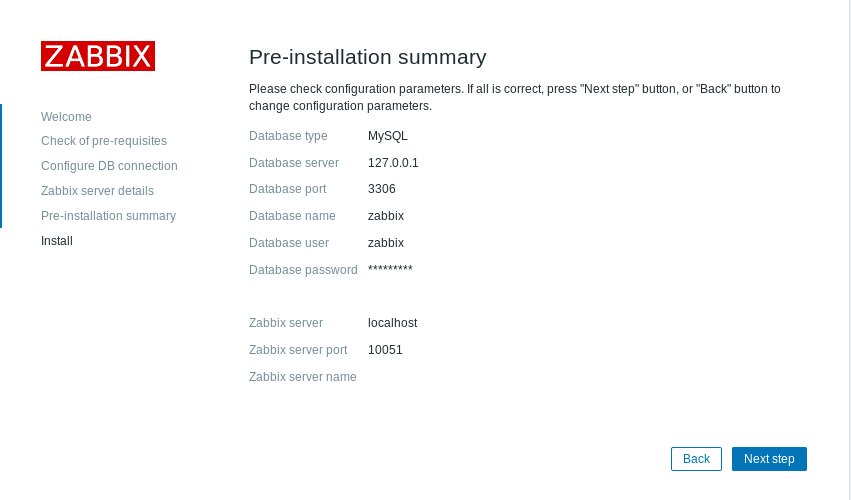
php 模块检查

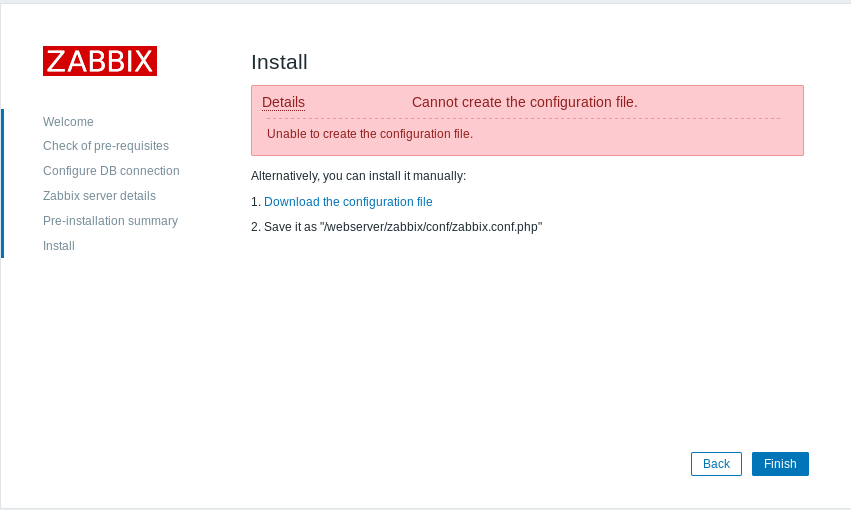


配置数据库信息

如果localhost 不通过就配置成127.0.0.1







添加配置文件

cat /webserver/zabbix/conf/zabbix.conf.php

<?php

// Zabbix GUI configuration file.

global $DB;

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

$DB['SERVER'] = '127.0.0.1';

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

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

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

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

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

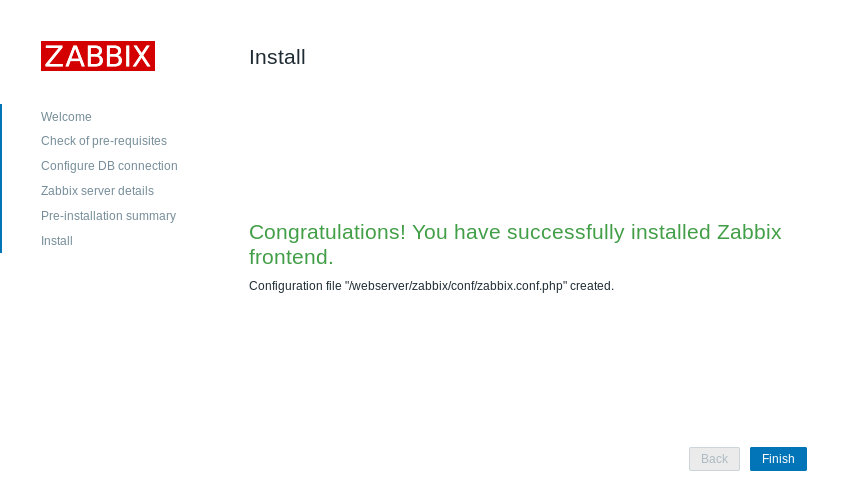
$DB['SCHEMA'] = '';

$ZBX\_SERVER = 'localhost';

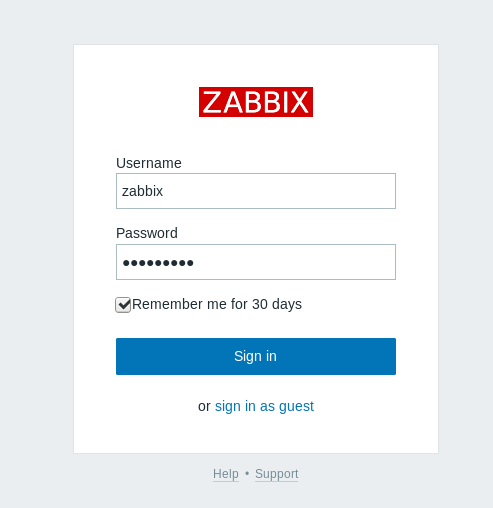
$ZBX\_SERVER\_PORT = '10051';

$ZBX\_SERVER\_NAME = '';

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

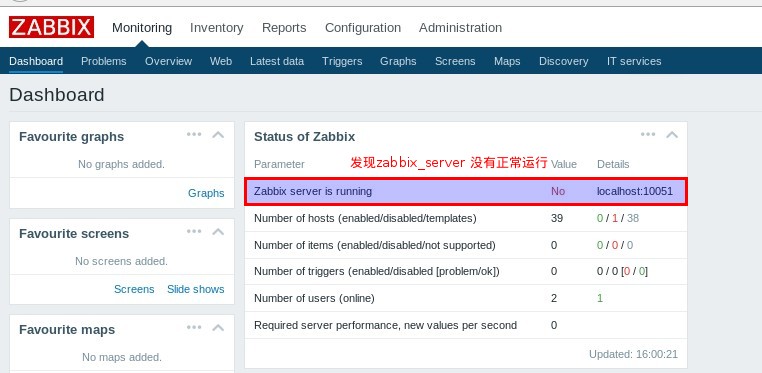


配置完毕,就可以登录了



默认用户名和密码是Admin zabbix

登录界面发现zabbix\_server 没有正常运行



systemctl status zabbix\_server 查看发现zabbix 启动有报错

Starting zabbix\_server: /usr/local/sbin/zabbix\_server: error while loading shared libraries: libmysqlclient.so.20: cannot open shared ... or direct

解决:

find / -name libmysqlclient.so.20

ln -s /application/mysql/lib/libmysqlclient.so.20 /usr/lib

ln -s /application/mysql/lib/libmysqlclient.so.20 /usr/lib64/

再次启动发现

Starting zabbix\_server: zabbix\_server [15721]: cannot open "/var/log/zabbix/zabbix\_server.log": [13] Permission denied

修改权限

chown -R zabbix:zabbix /var/log/zabbix/

systemctl restart zabbix\_server

systemctl status zabbix\_server

启动正常



agent 客户端配置

zabbix\_agent 配置和安装

zabbix\_agent 可直接通过rpm包方式进行安装,

rpm -ivh <http://repo.zabbix.com/zabbix/3.2/rhel/7/x86_64/zabbix-agent-3.2.3-1.el7.x86_64.rpm>

被动模式配置

vim /etc/zabbix/zabbix\_agentd.conf

PidFile=/var/run/zabbix/zabbix\_agentd.pid

LogFile=/var/log/zabbix/zabbix\_agentd.log

LogFileSize=0

Server=172.16.0.6

ServerActive=172.16.0.6 # server 端的ip地址

Hostname=172.16.0.1

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

UnsafeUserParameters=1 自定义key

添加相关服务和端口到系统配置文件

zabbix-agent 10050/tcp #Zabbix Agent

zabbix-agent 10050/udp #Zabbix Agent

启动

zabbix\_agentd

killall zabbix\_agentd

在zabbix\_server 测试

zabbix\_get -s 172.16.0.1 -p 10050 -k"system.uptime"

3220516

zabbix 配置文件详解

zabbix\_server 配置文件 zabbix\_server.conf

zabbix\_proxy 配置文件 zabbix\_proxy.conf

zabbix\_agent 配置文件 zabbix\_agent.conf

zabbix\_server 的配置文件

ListenPort=10051 #监听的端口

ListenIP=0.0.0.0 #监听的ip地址

zabbix\_agent 的配置文件:

StartAgents=3 # zabbix\_agent 开启的进程数

zabbix\_proxy 的配置文件:

zabbix 设置成中文之后有乱码问题解决

在win 的机器上拷贝字体

C:\[Windows](http://www.ttlsa.com/windows/)\Fonts\

拷贝字体到

sudo cp SIMSUNB.TTF /webserver/zabbix/fonts/

mv SIMSUNB.TTF SIMSUNB.ttf

修改web 前端加载的字体

vim /webserver/zabbix/include/defines.inc.php

修改前

define('ZBX\_GRAPH\_FONT\_NAME', 'DejaVuSans'); // font file name

define('ZBX\_FONT\_NAME', 'DejaVuSans');

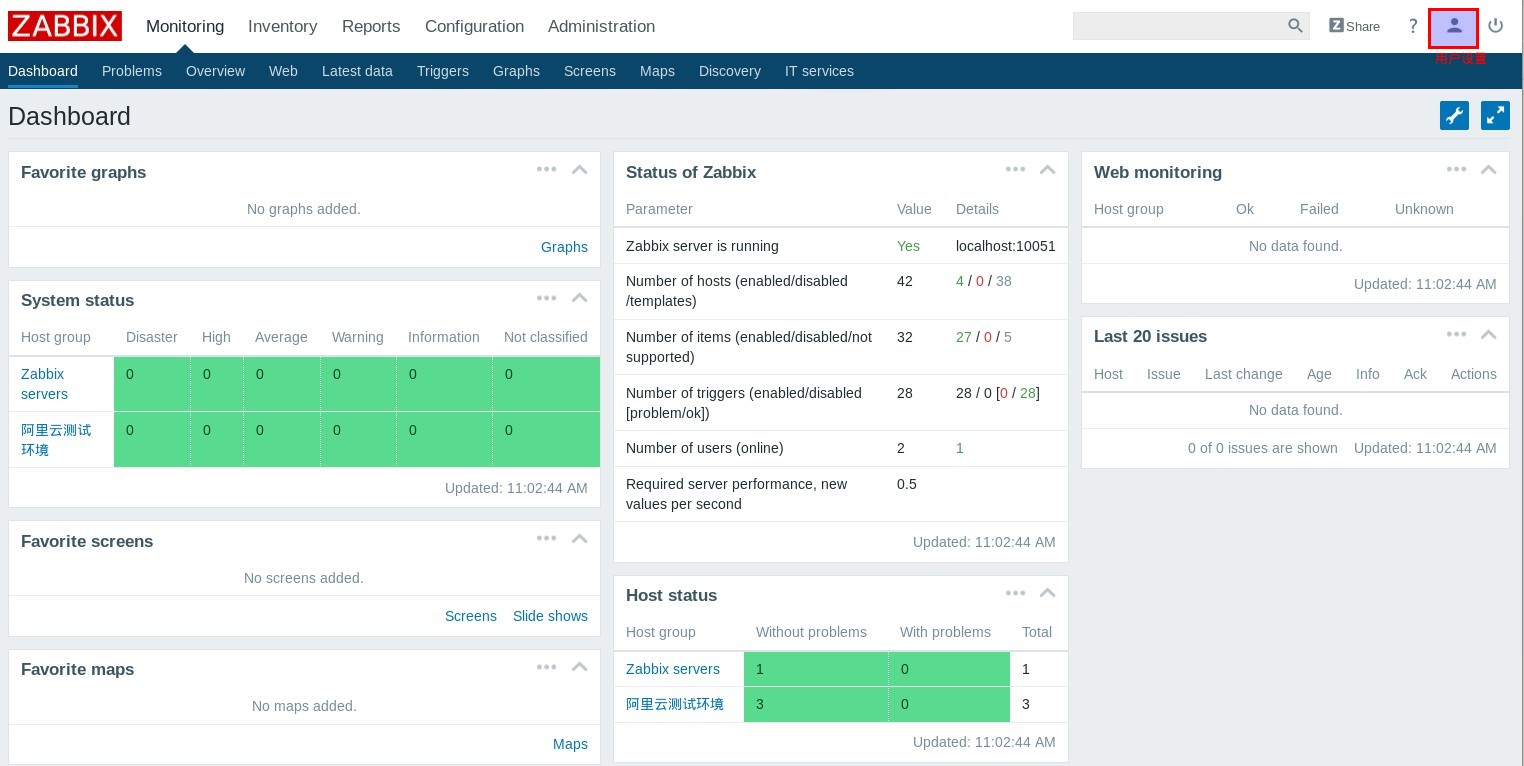
修改后

define('ZBX\_GRAPH\_FONT\_NAME', 'SIMSUNB'); // font file name

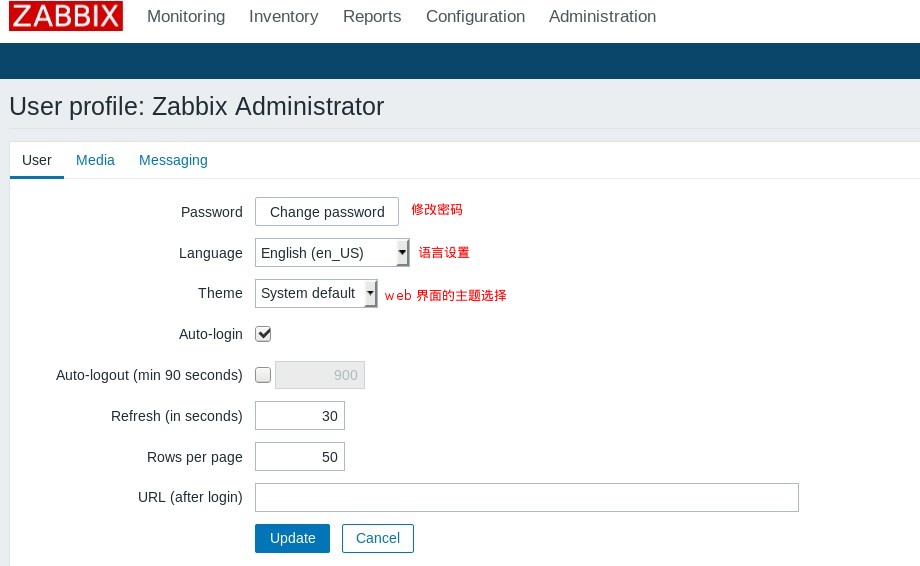
define('ZBX\_FONT\_NAME', 'SIMSUNB');

#如果是编译的php 需要去掉参数--enable-gd-jis-conv ,php编译启用enable-gd-jis-conv选项的话，那么非ASCII字符(例如汉字、拼音、希腊文和箭头)会被当成EUC-JP编码(phpinfo中美其名曰“支持JIS编码的字体”), 从而导致乱码(由于西文字体没有假名或汉字，一般表现为全部是方框)

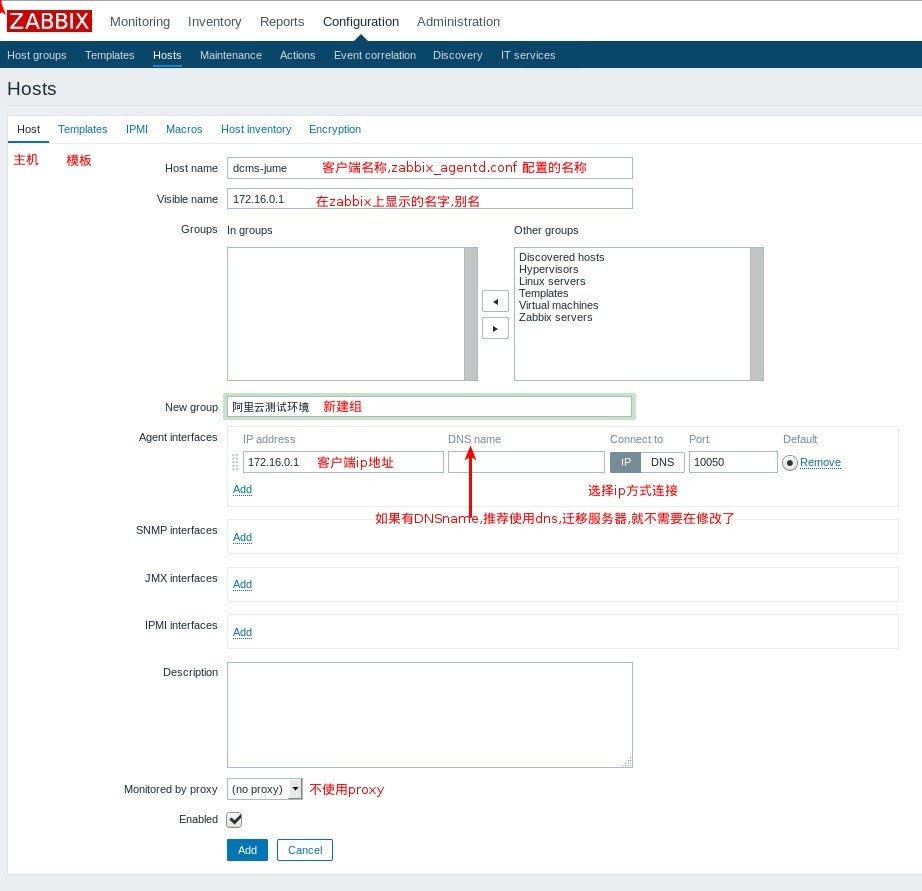
登录进行zabbix 进行设置

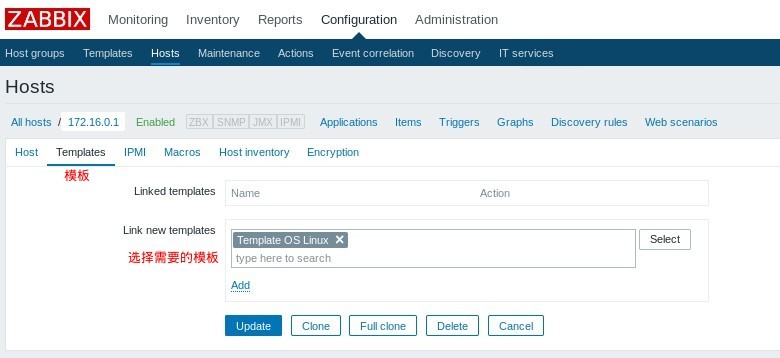


用户设置



添加主机

Configuration(配置)-----Hosts(主机)------------create hosts(创建主机)

关联监控模板

用户参数 User parameters

1. 概述

有时候当我们监控的项目在zabbix 预定义的key 中没有定义时,这时候我们可以通过编写zabbix 的用户参数的方法来监控我们要求的项目item.形象一点说zabbix 代理端配置文件中的User parameters 就相当于通过脚本获取要监控的值,然后把相关的脚本或者命令写入到配置文件中的User parameters 中,然后zabbix server 读取配置中的返回值通过处理前端的方式返回给用户

用户参数的语法:

UserParameter=<key>,<command>

例子:

UserParameter=mysql.ping,/usr/local/mysql/bin/mysqladmin ping |grep -c alive

修改web 背景为黑色



2. 让key 接受参数

通过key 也接受参数的方法使item 添加时更具备灵活性,例如系统预定义key, vm.memory.size[<mode>], 其中mode模式就是用户要接受的参数,当我们填写为free 时则返回为内存的剩余大小,如果我们填入的为userd 时这时返回的内存已经使用的大小.

相关语法:

UserParameter=key[\*],command

描述

key:key 的值在主机系统中必须是唯一的,其中\* 代表命令中接受的参数

命令:客户端系统中可执行的命令

自定义监控