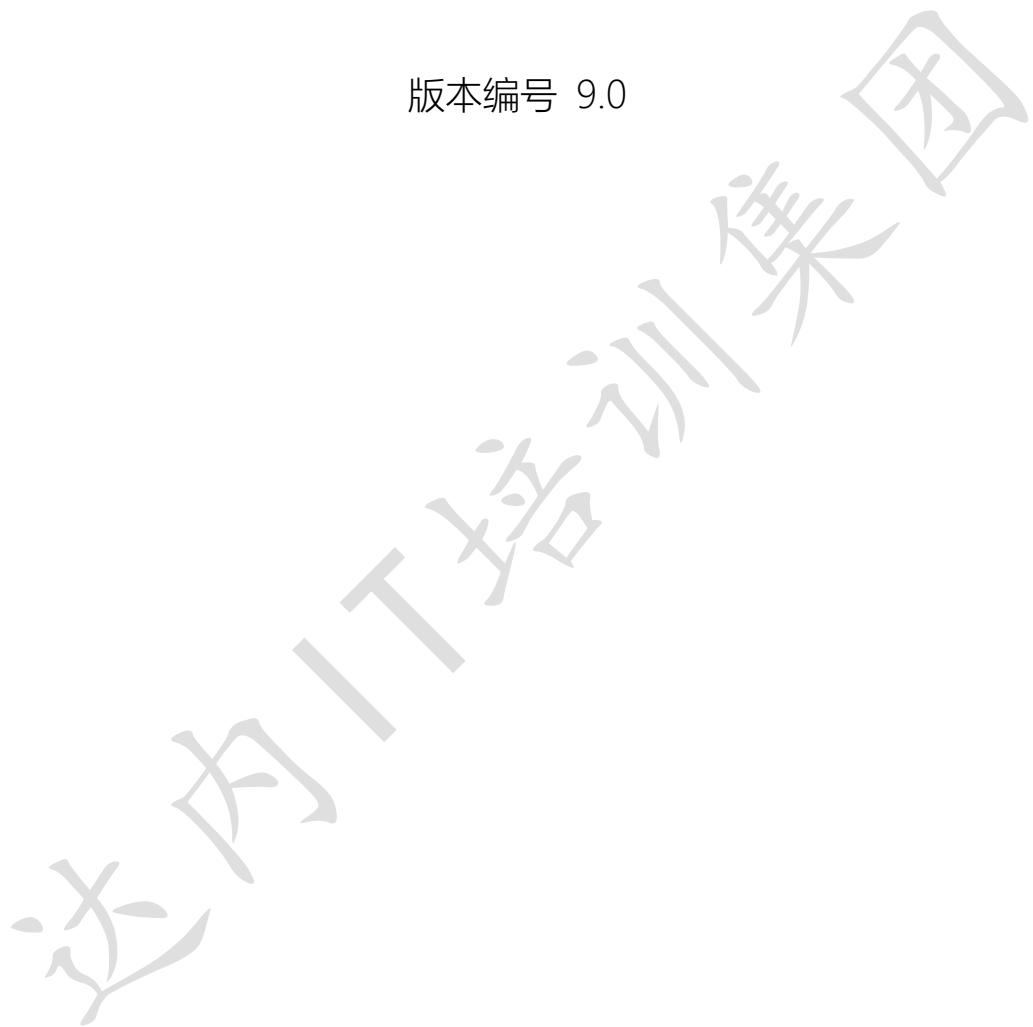


TTS 9.0 COOKBOOK

(NSD PROJECT3 DAY04)

版本编号 9.0



2020-06

达内 IT 培训集团

NSD PROJECT3 DAY04

1. 案例 1：部署 zabbix 监控服务器

- 问题

本案例要求部署在华为云上部署一台 Zabbix 监控服务器，监控其他主机。

- 安装 LNMP 环境
- 源码安装 Zabbix
- 安装监控端主机，修改基本配置
- 初始化 Zabbix 监控 Web 页面
- 修改 PHP 配置文件，满足 Zabbix 需求
- 监控 Zabbix_server 自身系统状态

- 方案

准备如表-1 所示的实验环境，配置主机名称、IP 地址。

表-1 主机列表

主机名称	IP 地址	角色
zabbix-server	192.168.1.51	Zabbix 监控服务
zabbix-proxy	192.168.1.52	Zabbix 监控代理
zabbix-agentd	192.168.1.53	Zabbix 客户端(测试用)

- 步骤

实现此案例需要按照如下步骤进行。

步骤一、部署 LNMP 环境

1)、购买华为云服务器

基础配置：无

网络配置：手动分配 IP 地址 192.168.1.51

高级配置：云服务器名称 zabbix-server

确认配置：1台

2)、更新/etc/hosts

```
[root@ecs-proxy ~]# cat >> /etc/hosts <<EOF  
192.168.1.51 zabbix-server  
EOF
```

3)、更新/root/ansible/hosts 配置文件

```
[root@ecs-proxy ~]# cat >> /root/ansible/hosts <<EOF  
[zabbix]  
192.168.1.51  
EOF
```

将最新的/etc/hosts 配置文件更新到所有的云主机上

```
[root@ecs-proxy ~]# cd /etc/ansible  
[root@ecs-proxy ansible]# ansible all -m copy -a 'src=/etc/hosts dest=/etc'
```

4)、安装 LNMP 所需软件包

```
[root@zabbix-server ~]# yum -y install gcc pcre-devel openssl-devel  
[root@zabbix-server ~]# scp root@192.168.1.252:/root/project3/DAY04/nginx-1.12.2.tar.gz /root  
[root@zabbix-server ~]# tar -xf /root/nginx-1.12.2.tar.gz  
[root@zabbix-server ~]# cd /root/nginx-1.12.2/  
[root@zabbix-server nginx-1.12.2]# ./configure --with-http_ssl_module  
[root@zabbix-server nginx-1.12.2]# make && make install  
[root@zabbix-server nginx-1.12.2]# yum -y install php php-mysql php-fpm  
[root@zabbix-server nginx-1.12.2]# yum -y install mariadb mariadb-devel mariadb-server
```

5)、修改 Nginx 配置文件

配置 Nginx 支持 PHP 动态网站,因为有大量 PHP 脚本需要执行,因此还需要开启 Nginx 的各种 fastcgi 缓存, 加速 PHP 脚本的执行速度。

```
[root@zabbix-server ~]# vim /usr/local/nginx/conf/nginx.conf  
... ...  
http{  
... ...  
fastcgi_buffers 8 16k;          #缓存php生成的页面内容, 8个16k  
fastcgi_buffer_size 32k;        #缓存php生产的头部信息  
fastcgi_connect_timeout 300;    #连接PHP的超时时间  
fastcgi_send_timeout 300;       #发送请求的超时时间  
fastcgi_read_timeout 300;       #读取请求的超时时间  
... ...  
server {
```

```
listen 8090; #将监听端口更改为 8090  
... ...  
  
location ~ \.php$ {  
root html;  
fastcgi_pass 127.0.0.1:9000;  
fastcgi_index index.php;  
include fastcgi.conf;  
}  
... ...
```

6)、启动服务

```
[root@zabbix-server nginx-1.12.2]# systemctl enable --now mariadb  
[root@zabbix-server nginx-1.12.2]# systemctl enable --now php-fpm  
[root@zabbix-server nginx-1.12.2]# /usr/local/nginx/sbin/nginx  
[root@zabbix-server nginx-1.12.2]# echo /usr/local/nginx/sbin/nginx >> /etc/rc.local  
[root@zabbix-server nginx-1.12.2]# chmod +x /etc/rc.local
```

步骤二、部署 Zabbix 服务端

1)、源码安装 Zabbix Server

多数源码包都是需要依赖包的，zabbix 也一样，源码编译前需要先安装相关依赖包。

```
[root@zabbix-server nginx-1.12.2]# yum -y install net-snmp-devel curl-devel autoconf libevent-devel  
[root@zabbix-server nginx-1.12.2]# scp root@192.168.1.252:/root/project3/DAY04/zabbix-3.4.4.tar.gz /root  
[root@zabbix-server nginx-1.12.2]# tar -xf /root/zabbix-3.4.4.tar.gz -C /root  
[root@zabbix-server nginx-1.12.2]# cd /root/zabbix-3.4.4/  
[root@zabbix-server zabbix-3.4.4]# ./configure --enable-server --enable-proxy --enable-agent --with-mysql=/usr/bin/mysql_config --with-net-snmp --with-libcurl  
[root@zabbix-server zabbix-3.4.4]# 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 指定 mysql_config 路径  
# --with-net-snmp 允许 zabbix 通过 snmp 协议监控其他设备（如交换机、路由器等）  
# --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
```

2)、创建并初始化数据库

```
[root@zabbix-server zabbix-3.4.4]# mysql
```

```
mysql> create database zabbix character set utf8;
#创建数据库，数据库名称为 zabbix，支持中文字符集
mysql> grant all on zabbix.* to zabbix@'localhost' identified by 'zabbix';
#创建可以访问数据库的账户与密码，用户名是 zabbix，密码是 zabbix
```

```
[root@zabbix-server zabbix-3.4.4]# cd /root/zabbix-3.4.4/database/mysql/
[root@zabbixserver mysql]# mysql -uzabbix -pzabbix zabbix < schema.sql
[root@zabbixserver mysql]# mysql -uzabbix -pzabbix zabbix < images.sql
[root@zabbixserver mysql]# mysql -uzabbix -pzabbix zabbix < data.sql
#刚刚创建是空数据库，zabbix 源码包目录下，有提前准备好的数据
#使用 mysql 导入这些数据即可（注意导入顺序）
#-u 指定数据库用户名，-p 指定数据库密码
```

3)、修改 zabbix_server 配置并启动监控服务

修改 zabbix_server 配置文件，设置数据库相关参数，启动 zabbix_server 服务

```
[root@zabbix-server mysql]# sed -n '38p;95p;111p;119p' /usr/local/etc/zabbix_server.conf
LogFile=/tmp/zabbix_server.log #38 行，日志的位置，排错使用，仅查看以下即可(默认已经配置 OK)
DBHost=localhost # 85 行，定义数据库服务器在哪台电脑(localhost 本机)
DBName=zabbix #95 行，设置数据库名称。(默认已经配置 OK)
DBUser=zabbix #111 行，设置数据库账户。(默认已经配置 OK)
DBPassword=zabbix #119 行，设置数据库密码
```

```
[root@zabbix-server mysql]# useradd -s /sbin/nologin zabbix
[root@zabbix-server mysql]# zabbix_server
[root@zabbix-server mysql]# echo zabbix_server >> /etc/rc.local #设置开机自启
[root@zabbix-server mysql]# chmod +x /etc/rc.local
#确认连接状态，端口 10051
[root@zabbix-server mysql]# netstat -antpu | grep zabbix_server
tcp 0 0 0.0.0.0:10051 0.0.0.0:* LISTEN 13890/zabbix_server
```

提示：如果是因为配置文件不对，导致服务无法启动时，不要重复执行 zabbix_server，一定要先使用 killall zabbix_server 关闭服务后，再重新启动一次。

4)、修改 Zabbix_agent 配置文件，启动 Zabbix_agent 服务。

自定义的 key 文件一般存储在/usr/local/etc/zabbix_agentd.conf.d/目录，这里还需要修改 zabbix_agentd.conf 文件，允许自定义监控 key，来读取该目录下的所有文件。

```
[root@zabbix-server mysql]# vim /usr/local/etc/zabbix_agentd.conf
LogFile=/tmp/zabbix_agentd.log          #30 行，定义日志文件的位置(默认已经配置 OK)
Server=127.0.0.1,192.168.1.51          #93 行，允许哪些主机监控本机
```

```
ServerActive=127.0.0.1,192.168.1.51      #134 行, 允许哪些主机通过主动模式监控本机  
Hostname=Zabbix_server #145 行, 设置本机主机名  
Include=/usr/local/etc/zabbix_agentd.conf.d/    #264 行, 加载配置文件目录。  
UnsafeUserParameters=1 #280 行, 自定义监控可以传递参数。默认为 0, 表示不允许自定义 key。
```

```
[root@zabbix-server mysql]# zabbix_agentd          #启动监控 agent  
[root@zabbix-server mysql]# echo zabbix_agentd >> /etc/rc.local  #设置开机自启  
[root@zabbix-server mysql]# netstat -antpu | grep zabbix_agentd  #查看端口信息为 10050  
  
tcp 0 0 0.0.0.0:10050 0.0.0.0:* LISTEN 14095/zabbix_agentd
```

5)、部署访问页面

```
[root@zabbix-server mysql]# cp -r /root/zabbix-3.4.4/frontends/php/* /usr/local/nginx/html/  
[root@zabbix-server mysql]# chmod -R 777 /usr/local/nginx/html/
```

6)、设置监听器,添加后端服务器。

【服务器列表】—>【弹性负载均衡 ELB】—>【(自定义 ELB 名称)】—>【监听器】—>【添加监听器】，如图-1 所示。

添加监听器

① 配置监听器 ————— ② 配置后端服务器组 ————— ③ 完成

① 配置监听器

* 名称 (输入框被红色方框标记)

* 前端协议/端口 (下拉菜单显示 TCP) (输入框被红色方框标记) 取值范围1~65535

四层监听请选择TCP、UDP；七层监听请选择HTTP、HTTPS。
选择HTTPS协议时，后端协议只能使用HTTP协议。

图-1

配置后端服务器组名称，如图-2 所示。

添加监听器

① 配置监听器 ————— ② 配置后端服务器组 ————— ③ 完成

后端服务器组

新创建 使用已有

* 名称

图-2

点击刚创建的监听器名称->后端服务器组->添加，如图-3 所示。

myelb 运行中

基本信息 监听器 后端服务器组 监控 访问日志 标签

添加监听器

listener-8090 TCP/8090		
listener-ha TCP/1080		
listener-80 TCP/80		

基本信息 后端服务器组 标签

名称: server_group-zabbix
分配策略类型: 加权轮询算法
会话保持: 未开启

添加 移除

图-3

添加后端服务器，并监听端口 8090，如图-4、图-5 所示。

添加后端服务器

后端服务器的安全组规则必须放通100.125.0.0/16网段，否则会导致健康检查异常。[了解更多](#)

您最多可以添加494个后端服务器，如需申请更多配额请点击[申请扩大配额](#)。

购买云服务器 名称

云服务器	规格	私网IP地址
<input type="checkbox"/> git	1 vCPUs 1 GB s6.small.1	192.168.1.54
<input type="checkbox"/> docker	1 vCPUs 1 GB s6.small.1	192.168.1.68
<input type="checkbox"/> harbor	1 vCPUs 1 GB s6.small.1	192.168.1.67
<input checked="" type="checkbox"/> zabbix-server	1 vCPUs 1 GB s6.small.1	192.168.1.51
<input type="checkbox"/> redis-0003	1 vCPUs 1 GB s6.small.1	192.168.1.33

图-4

添加后端服务器

• 后端服务器的安全组规则必须放通100.125.0.0/16网段，否则会导致健康检查异常。 [了解更多](#)
• 使用扩展网卡之前，请先配置路由策略。 [了解如何配置](#)

私网IP地址	云服务器	已添加端口	权重	操作
192.168.1.51	zabbix-server 1 vCPUs 1 GB s6.small.1	--	<input type="text" value="1"/>	移除

图-5

7)、访问 Zabbix Web 界面，<http://华为云公网 IP:8090/>

根据错误提示，修改 PHP 配置文件，满足 Zabbix_server 的 Web 环境要求。

#第一次访问，初始化 PHP 页面会检查计算机环境是否满足要求，如果不满足会给出修改建议

#默认会提示 PHP 的配置不满足环境要求，需要修改 PHP 配置文件

```
[root@zabbix-server mysql]# yum -y install php-gd php-xml php-ldap php-bcmath php-mbstring
[root@zabbix-server mysql]# vim /etc/php.ini
max_execution_time = 300          #384 行，最大执行时间
max_input_time = 300             #394 行，服务器接收数据的时间限制
memory_limit = 128M              #405 行，内存容量限制(默认已经配置,确认下即可)
post_max_size = 32M               #672 行，POST 数据最大容量
date.timezone = Asia/Shanghai   #878 行，设置时区
[root@zabbix-server mysql]# systemctl restart php-fpm
```

修改完 PHP 配置文件后，再次使用浏览器访问 zabbix-server 服务器，则会提示如图-6 所示信息。

ZABBIX

Check of pre-requisites

	Current value	Required
PHP version	5.4.16	5.4.0 OK
PHP option "memory_limit"	128M	128M OK
PHP option "post_max_size"	32M	16M OK
PHP option "upload_max_filesize"	2M	2M OK
PHP option "max_execution_time"	300	300 OK
PHP option "max_input_time"	300	300 OK
PHP option "date.timezone"	Asia/Shanghai	OK
PHP databases support	MySQL	OK
PHP bcmath	on	OK
PHP mbstring	on	OK
PHP option "mbstring.func_overload"	off	off OK
PHP sockets	on	OK

[Back](#) [Next step](#)

图-6

在初始化数据库页面，填写数据库相关参数，如图-7，图-8 所示。

ZABBIX

Configure DB connection

Please create database manually, and set the configuration parameters for connection to this database.
Press "Next step" button when done.

Welcome	Database type	MySQL ▾	数据库类型
Check of pre-requisites	Database host	localhost	数据库IP地址，本机为localhost
Configure DB connection	Database port	3306	0 - use default port
Zabbix server details	Database name	zabbix	数据库名称
Pre-installation summary	User	zabbix	数据库用户名
Install	Password	*****	数据库帐号密码 此为 zabbix

图-7

ZABBIX

Zabbix server details

Please enter the host name or host IP address and port number of the Zabbix server, as well as the name of the installation (optional).

Welcome	Host	192.168.1.51
Check of pre-requisites	Port	10051
Configure DB connection	Name	ZabbixServer
Zabbix server details		
Pre-installation summary		
Install		

图-8

在登陆页面，使用用户(admin)和密码(zabbix)登陆，如图-9 所示。



图-9

登陆后在 Zabbix 界面点击右上角，设置语言环境为中文 Chinese(zh_CN)，如图-10、图-11 所示。

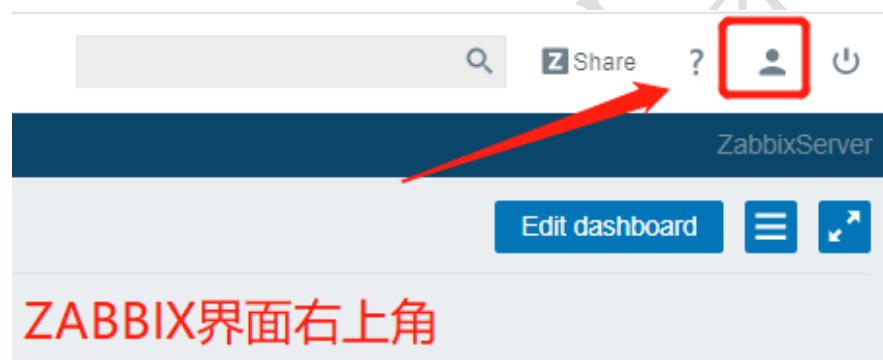


图-10

User profile: Zabbix Administrator

User Media Messaging

Language Chinese (zh_CN)

Theme System default

Auto-login

Auto-logout 15m

Refresh 30s

Rows per page 50

URL (after login)

Update Cancel

图-11

2. 案例 2：制作 Zabbix 客户端 RPM 软件包

- 问题

本案例要求使用 zabbix-3.4.4 版本的源码软件，生成对应的 RPM 软件包。

安装 rpm-build 软件包，编写 SPEC 配置文件，创建新的 RPM 软件包。

- 步骤

实现此案例需要按照如下步骤进行。

步骤一、安装 rpm-build 软件

1)、安装 rpm-build 软件包

```
[root@nginx-0001 ~]# yum -y install rpm-build
```

2)、生成 rpmbuild 目录结构。

#有报错，无需理会，可以看到 rpmbuild 目录已经创建完成

```
[root@nginx-0001 ~]# rpmbuild -ba nginx.spec  
error: failed to stat /root/nginx.spec: No such file or directory
```

```
[root@nginx-0001 ~]# ls /root/rpmbuild/  
BUILD BUILDROOT RPMS SOURCES SPECS SRPMS
```

3)、准备工作，将源码软件复制到 SOURCES 目录。

```
[root@nginx-0001 ~]# scp root@192.168.1.252:/root/project3/DAY04/zabbix-3.4.4.tar.gz  
/root/rpmbuild/SOURCES/
```

4)、创建并修改 SPEC 配置文件

```
[root@nginx-0001 ~]# cat /root/rpmbuild/SPECS/zabbix.spec  
Name:zabbix          #源码包软件名称。【不能错】  
Version:3.4.4         #源码包软件的实际版本号。【不能错】  
Release: 22           #发布序列号，标明第几次打包，后面可使用%{release}引用  
Summary: zabbix agentd #软件包的内容概要。自定义  
  
#Group:               #软件分组。(一个组中可以有多个软件包，输入组名安装软件包，可都安装。)
```

```
License: GPL      #软件授权方式，通常就是 GPL
URL: www.zabbix.com          #网址
Source0: zabbix-3.4.4.tar.gz  #源码包文件名。 【不能错】

#BuildRequires:           #源码编译的依赖。(写和写都不能解决依赖，仅起到标注作用。)
#Requires:                 #yum 安装的依赖。

%description             #定义 rpm 包的描述信息
This is zabbix rpm from dayu

%post #rpm 包安装后执行的脚本。今后有人在使用这个软件包安装时，就会触发下面的脚本内容。
useradd -s /sbin/nologin zabbix

%prep #rpm 包安装前执行的脚本
%setup -q #自动解压源码并 cd

%build                  #定义编译软件包时的操作
./configure --enable-agent    #配置源码 【需要修改】
make %{?_smp_mflags}

%install
make install DESTDIR=%{buildroot}

%files
%doc                   #对哪个目录打包用来做 rpm 包
/usr/local/sbin/zabbix_agentd
/usr/local/bin/zabbix_get
/usr/local/bin/zabbix_sender
/usr/local/etc/zabbix_agentd.conf.d
/usr/local/etc/zabbix_agentd.conf
/usr/local/share/man/man1/zabbix_get.1
/usr/local/share/man/man1/zabbix_sender.1
/usr/local/share/man/man8/zabbix_agentd.8

%changelog
```

2、使用配置文件创建 RPM 包

1)、安装依赖软件包

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

2)、rpmbuild 创建 RPM 软件包

```
[root@nginx-0001 ~]# rpmbuild -ba /root/rpmbuild/SPECS/zabbix.spec
[root@nginx-0001 ~]# ls /root/rpmbuild/RPMS/x86_64/
zabbix-3.4.4-22.x86_64.rpm zabbix-debuginfo-3.4.4-22.x86_64.rpm
```

3. 案例 3、Zabbix 自动化监控

- 问题

本案例要求配置 Zabbix 自动发现机制。

- 沿用案例二制作的 rpm 包，为三台 Nginx 部署 Zabbix 客户端。
- 创建 Web 监测
- 创建自动发现规则
- 创建自动发现后的动作，添加主机、为主机链接模板。

- 方案

当 Zabbix 需要监控的设备越来越多，手动添加监控设备越来越有挑战，此时，可以考虑使用自动发现功能，自动添加被监控主机，实现自动批量添加一组监控主机功能。

- 步骤

实现此案例需要按照如下步骤进行。

步骤一、为三台 Nginx 部署 Zabbix 客户端。

1)、将 Zabbix 客户端软件分发到三台 Nginx 服务器上。

```
[root@ecs-proxy ansible]# ansible web -m copy -a "src=/root/project3/DAY04/zabbix-3.4.4-22.x86_64.rpm dest=/root/"
```

2)、安装软件包

```
[root@ecs-proxy ansible]# ansible web -m shell -a "yum -y install /root/zabbix-3.4.4-22.x86_64.rpm"
[root@ecs-proxy ansible]# chmod +x /root/project3/DAY04/zabbix_agent.sh
[root@ecs-proxy ansible]# ansible web -m script -a "/root/project3/DAY04/zabbix_agent.sh"
```

3)、Zabbix 客户端启动是否成功

```
[root@nginx-0001 ~]# netstat -antpu |grep zabbix_agent
tcp 0 0 0.0.0.0:10050 0.0.0.0:* LISTEN 27128/zabbix_agentd
```

```
[root@nginx-0002 ~]# netstat -antpu | grep zabbix_agent
tcp 0 0 0.0.0.0:10050 0.0.0.0:* LISTEN 11430/zabbix_agentd
```

```
[root@nginx-0003 ~]# netstat -antpu |grep zabbix_agent
tcp      0      0 0.0.0.0:10050          0.0.0.0:*      LISTEN      1454/zabbix_agentd
```

步骤二、创建 Web 监测

监控网站的指标：平均下载速度、响应时间、HTTP 状态码

zabbix 提供了 web 监测功能，监控到站点的响应时间，还可以根据站点返回的状态码等。

1)、创建模板，如图-12 所示

模板名称可自定义，可选择现有群组，也可以在下方写新群组的名称，会自动创建。

图-12

创建完成后，点击群组，能够快速刚刚创建的模版，点击 Web 监测，如图-13 所示。

ZABBIX 监测中 资产记录 报表 配置 管理

主机群组 模板 主机 维护 动作 关联项事件 自动发现 服务 ZabbixServer

模板 群组 mytem 创建模板 导入

过滤器 ▲

名称 | 应用 重设

名称▲	应用集	监控项	触发器	图形	聚合图形	自动发现	Web监测	链接的模板	已链接到
Template Web Mon	应用集	监控项	触发器	图形	聚合图形	自动发现	Web监测		

显示 已自动发现的 1 中的 1

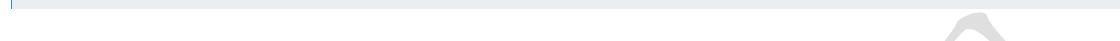


图-13

2)、在 Web 监测中创建 Web 场景，如图-14 所示

ZABBIX 监测中 资产记录 报表 配置 管理

主机群组 模板 主机 维护 动作 关联项事件 自动发现 服务 ZabbixServer

Web 监测 群组 mytem 主机 Template Web Mon 创建 Web 场景

所有模板 / Template Web Mon 应用集 监控项 触发器 图形 聚合图形 自动发现规则 Web 场景

过滤器 ▲

状态 所有 已启用 停用的 应用 重设

名称▲	步骤数量	间隔	尝试次数	认证	HTTP 代理	应用集	状态
未发现数据 显示 已自动发现的 0 中的 0							

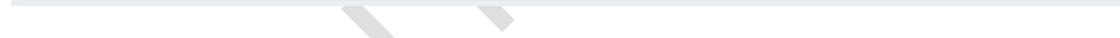


图-14

名称，自定义填写。客户端模拟用户去访问网站使用的浏览器类型。

ZABBIX 监测中 资产记录 报表 配置 管理

主机群组 模板 **主机** 维护 动作 关联项事件 自动发现 服务

Web监测

所有模板 / Template Web Mon 应用集 监控项 触发器 图形 聚合图形 自动发现规则 Web 场景

场景 步骤 认证

名称	one page						
应用集 没有发现应用集							
新的应用集							
更新间隔	1m						
尝试次数	1						
客户端	Internet Explorer 11.0						
HTTP 代理 http://[user[:password]@]proxy.example.com[:port]							
变量	<table border="1"> <tr> <td>名称</td> <td>值</td> </tr> <tr> <td>名字</td> <td>值</td> </tr> <tr> <td colspan="2">添加</td> </tr> </table>	名称	值	名字	值	添加	
名称	值						
名字	值						
添加							
头	<table border="1"> <tr> <td>名称</td> <td>值</td> </tr> <tr> <td>名字</td> <td>值</td> </tr> <tr> <td colspan="2">添加</td> </tr> </table>	名称	值	名字	值	添加	
名称	值						
名字	值						
添加							
已启用	<input checked="" type="checkbox"/>						
添加 取消							

图-15

添加步骤，监测目前的 nginx-0001、nginx-0002、nginx-0003 网站。如图-16 所示。

ZABBIX 监测中 资产记录 报表 配置 管理

主机群组 模板 **主机** 维护 动作 关联项事件 自动发现 服务

Web监测

所有模板 / Template Web Mon 应用集 监控项 触发器 图形 聚合图形 自动发现规则 Web 场景

步骤 **认证**

步骤	名称	超时	URL	要求的	状态码	动作

[添加](#) [取消](#)

名称 web1

URL http://192.168.1.11

[分析](#)

名称: web2
URL: http://192.168.1.12

分析

名称: web3
URL: http://192.168.1.13

图-16

3)、关联模版

将 Zabbix Server 设备开启，点击状态下“停用的”即可开启，如图-17 所示。

名称	应用集	监控项	触发器	图形	自动发现	Web 监测	状态	可用性	agent 加密	信息
Zabbix server	应用集 11	监控项 68	触发器 46	图形 11	自动发现 2	Web 监测 127.0.0.1:10050	停用的	ZBX SNMP JMX IPMI 无		

图-17

关联刚才新创建的模版，如图-18 所示。

所有主机 / Zabbix server 已启用 ZBX | SNMP | JMX | IPMI | 应用集 11 | 监控项 68 | 触发器 46 | 图形 11 | 自动发现规则 2 | Web 场景

主机 模板 IPMI 宏 主机资产记录 加密

链接的模板

名称	动作
Template App Zabbix Server	取消链接 取消链接并清理
Template OS Linux	取消链接 取消链接并清理

链接指示器

Template Web Mon ×

添加

更新 克隆 全克隆 删除 取消

图-18

添加完成后，点击【监测中】->【Web 监测】，查看由哪台主机进行监测。如图-19 所示。当点击名称，即可看到网页检测的数据图。

ZABBIX 监测中 资产记录 报表 配置 管理

仪表板 问题 概述 Web监测 最新数据 触发器 图形 聚合图形 拓扑图 自动发现 服务 ZabbixServer

Web监测

主机	名称▲	步骤数量	最近检查记录	状态
Zabbix server	one page	3	2020-08-30 23:31:44	正常

显示 已自动发现的 1 中的 1

图-19

因为字体的原因，打开的页面显示出很多乱码。需要向/usr/local/nginx/html/fonts 中导入字体。

```
[root@zabbix-server ~]# scp 192.168.1.252:/root/project3/DAY04/simkai.ttf /usr/local/nginx/html/fonts/
[root@zabbix-server ~]# ls /usr/local/nginx/html/fonts/
DejaVuSans.ttf simkai.ttf
[root@zabbix-server ~]# mv /usr/local/nginx/html/fonts/simkai.ttf /usr/local/nginx/html/fonts/DejaVuSans.ttf
mv: overwrite '/usr/local/nginx/html/fonts/DejaVuSans.ttf'? yes
```

导入字体刷新页面，即可看到下载速度的图表和响应时间的图表，如图-20 所示。

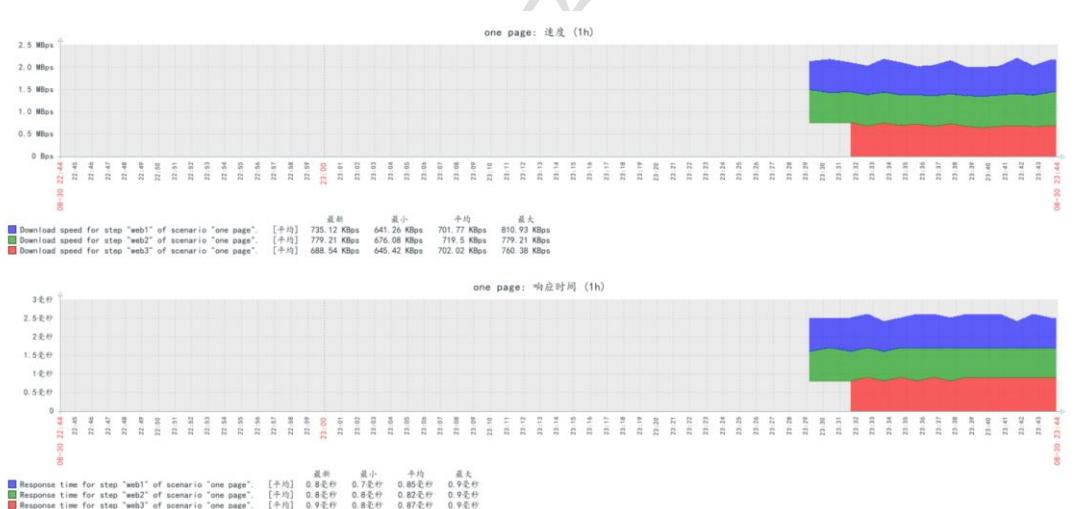


图-20

步骤二、自动发现规则

1)、创建自动发现规则

通过 Configuration (配置) --> Discovery (自动发现) --> Create discovery rule (创建发现规则)，如图-21 所示。



图-21

2) 填写规则

填写自动发现的 IP 范围（逗号隔开可以写多个），多久做一次自动发现（默认为1小时，仅实验修改为1m），如图-22 所示。

自动发现规则

名称 **名称自定义**

由agent代理程序自动发现

IP范围 **发现的范围**

更新间隔 **1分钟做一次自动发现**

检查 **新的 添加检查方式**

检查类型 端口范围

添加 取消

设备唯一性准则 IP地址

已启用

添加 取消

图-22

3、创建动作

1)、创建 Action 动作

通过 Configuration (配置) --> Actions (动作) --> Actions Event source(事件源)：自动发现(Discovery)-->Create action (创建动作)，如图-23 所示。

The screenshot shows the Zabbix web interface. At the top, there's a header with the Tedu.cn logo and navigation links like '监测中', '资产记录', '报表', '配置', '管理'. Below the header is a dark blue navigation bar with tabs: '主机群组', '模板', '主机', '维护', '动作' (which is highlighted with a red box), '关联项事件', '自动发现', and '服务'. On the right side of the header, there are icons for search, share, help, and user. The main content area has a title '动作' (Actions). At the top right of this area, there are buttons for '事件源' (Event Source), '自动发现' (Auto-discovery), and a large red-bordered '创建动作' (Create Action) button. Red arrows point from the text '如图-23所示。' to the 'Create Action' button and the '自动发现' button.

图-23

2) 配置 Action 动作具体行为

配置动作，添加动作名称，添加触发动作的条件。如图-24 所示。

This screenshot shows the 'Create Action' dialog box. At the top, it says '动作' (Actions) and '操作' (Operations). The 'Name' field contains 'add_web'. In the '新的触发条件' (New trigger condition) section, there is a dropdown menu '主机IP地址' (Host IP Address), an equals sign '=', and the value '192.168.1.10-20'. Below this is a red-bordered '添加' (Add) button. Underneath the condition, there is a checked '已启用' (Enabled) checkbox. At the bottom of the dialog are '添加' (Add) and '取消' (Cancel) buttons. Red boxes highlight the 'Name' field, the '添加' button under the condition, and the 'Enabled' checkbox.

图-24

点击操作 (触发动作后要执行的操作指令)，操作细节：与模板链接 (HTTP 模板 OS Linux 模版)。如图-25 所示。

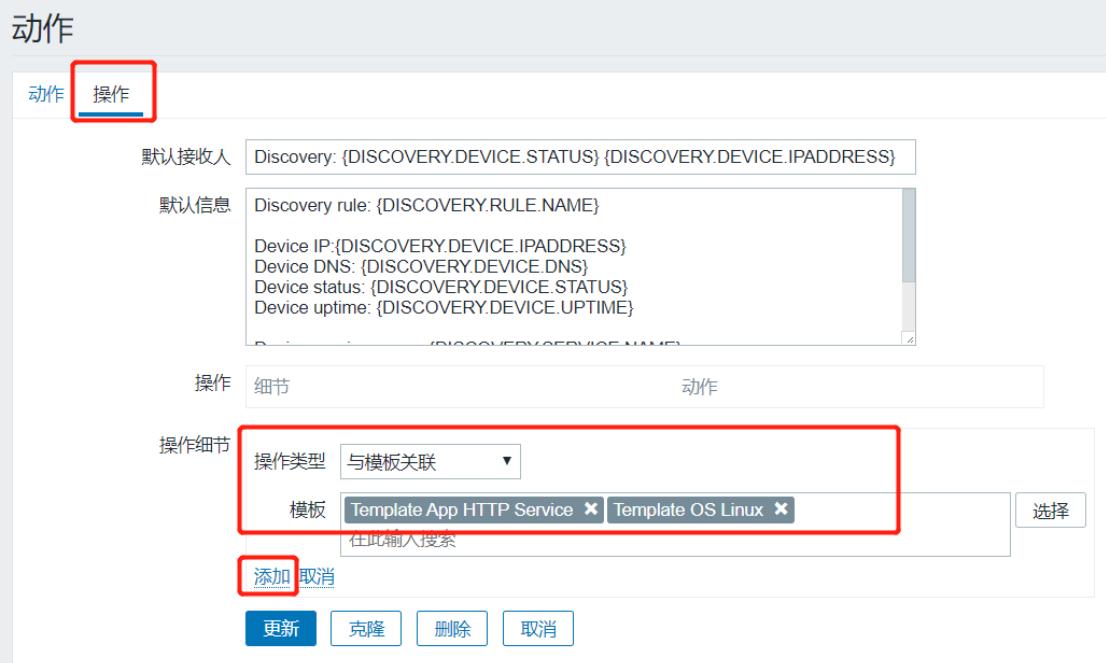


图-25

【注意：每台被监控主机必须要属于一个主机组，在这里会自动生成一个 Discovered hosts 群组，把自动发现的主机加入。】

通过 Configuration (配置) --> Hosts (主机)，如图-26 所示。

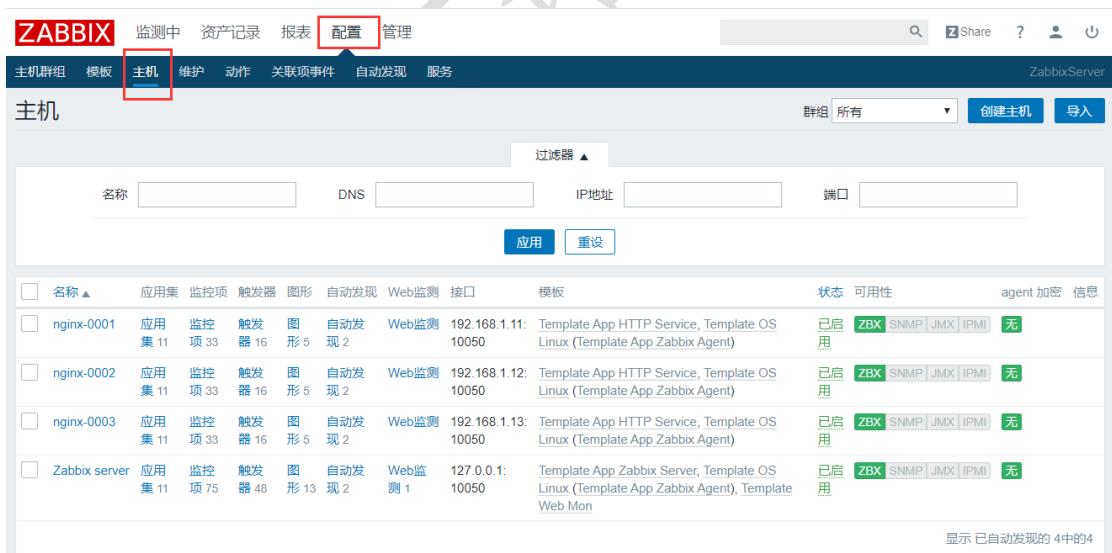


图-26

4. 案例 4、自定义 Zabbix 监控项目

- 问题

使用 Zabbix 实现自定义监控，实现以下目标：监控 nginx-0003 主机登录用户数量。需

要使用 Zabbix 自定义 key 的方式实现自定义监控，参考如下操作步骤：

- 创建自定义 key
- 创建监控项目
- 创建监控图形

- 步骤

实现此案例需要按照如下步骤进行。

步骤一、被监控主机创建自定义 key (在 192.168.1.13 上操作)

1)、创建自定义 key

Zabbix 虽然自带了许多 Key，能满足大多数的监控需求。但是真正在生产上还是有很多不足的。所以 Zabbix 还提供了一种自定义 Key 来实现这种需求。

自定义 key 语法格式为：UserParameter=自定义 key 名称,命令。

自定义的 key 文件一般存储在/usr/local/etc/zabbix_agentd.conf.d/目录，这里还需要修改 zabbix_agentd.conf 文件，允许自定义监控 key，来读取该目录下的所有文件。

【注意，之前在自动化案例中已经将该配置的注释去掉，此处只是为了确认。】

```
[root@nginx-0003 ~]# sed -n '264p' /usr/local/etc/zabbix_agentd.conf  
Include=/usr/local/etc/zabbix_agentd.conf.d/  
[root@nginx-0003 ~]# cat /usr/local/etc/zabbix_agentd.conf.d/count.login.num  
UserParameter=count.login.num,w --no-header | wc -l
```

2) 测试自定义 key 是否正常工作

```
[root@nginx-0003 ~]# killall zabbix_agentd  
[root@nginx-0003 ~]# zabbix_agentd  
[root@nginx-0003 ~]# zabbix_get -s 127.0.0.1 -k count.login.num  
2
```

步骤二、创建监控模版

1) 添加监控模板

登录 Zabbix Web 监控控制台，通过 Configuration(配置)-->Template(模板)-->Create template(创建模板)，填写模板名称，新建模板群组。如图-27，图-28 所示。



图-27



图-28

2) 创建应用集

创建完成模板后，默认模板中没有任何应用集、监控项、触发器、图形等资源。这里需要点击模板后面的 Application (应用集) 链接打开创建应用的页面。如图-29 所示。



图-29

点击 Application (应用集) 后，在该页面中点击 Create application (创建应用集) 按钮。如图-30 所示。



图-30

设置应用名称，名称可以任意，如图-31 所示。



图-31

3) 创建监控项目 item (监控项)

与创建应用一样，在模板中还需要创建监控项目。点击 items (监控项)，并在刷新出的新页面中选择 Create items (创建监控项) 创建项目。如图-32、图-33 所示。

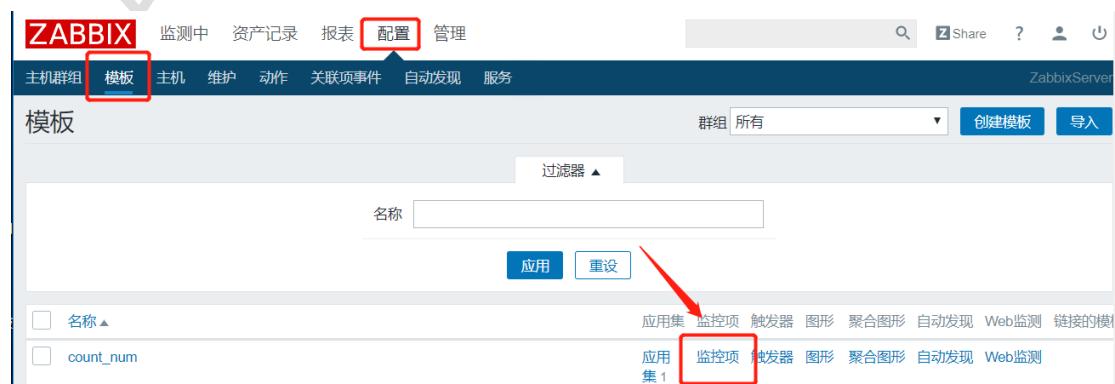


图-32



图-33

接下来，还需要给项目设置名称（名称可以任意）及对应的自定义 key（必须与前面自定义的监控 key 名称一致）。如图-34 所示。

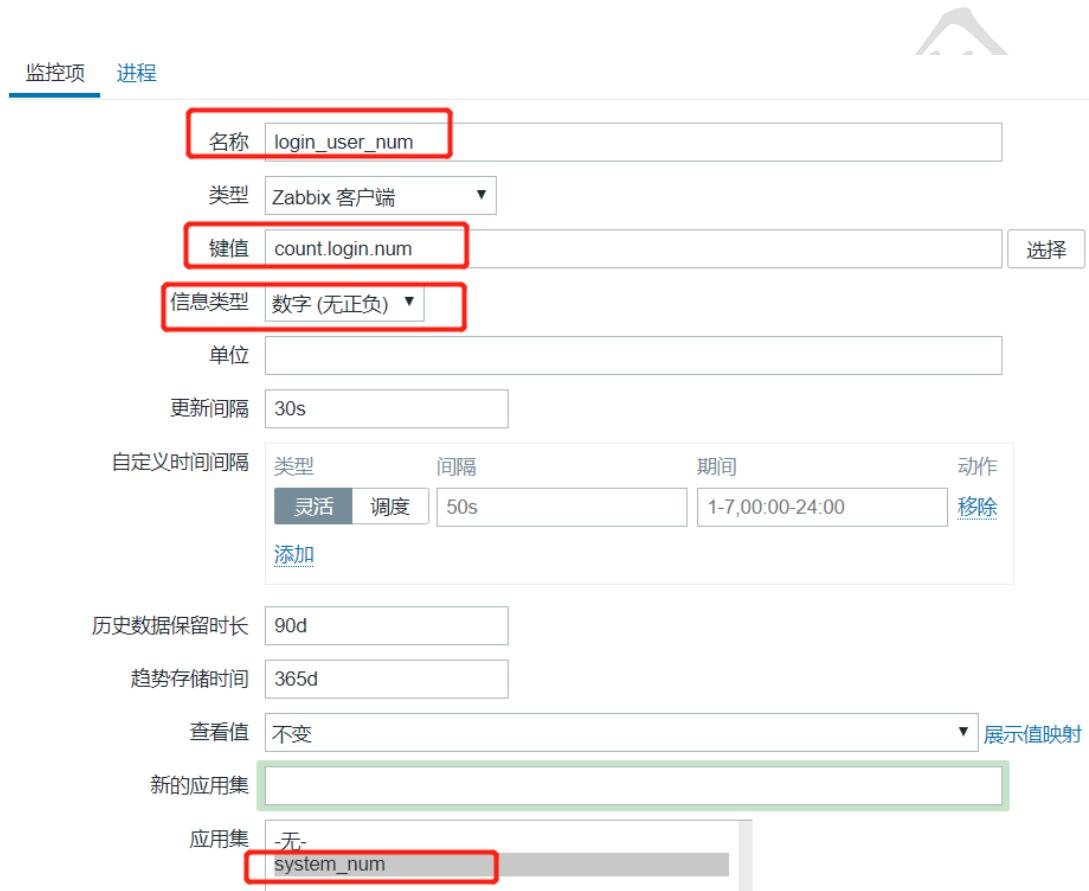


图-34

4) 创建图形

为了后期可以通过图形的方式展示监控数据，还需要在模板中创建图形，设置方法与前面的步骤一致，在监控模板后面点击 Graph (图形) 即可创建图形，设置监控图形基于什么监控数据。如图-35，图-36 所示。

图-35

图-36

5) 将模版链接到被监控主机 nginx-0003 上。

将完整的监控模板制作完成后，就可以将模板链接到主机实现监控功能了。首先找到被监控主机 Configuration (配置) --> Hosts (主机)。如图-37 所示。

ZABBIX 监测中 资产记录 报表 配置 管理

主机群组 模板 **主机** 维护 动作 关联项事件 自动发现 服务

ZabbixServer

主机 群组: 所有 创建主机 导入

过滤器 ▲

名称	DNS	IP地址	端口

应用 重设

名称	应用集	监控项	触发器	图形	自动发现	Web监测	接口	模板	状态	可用性	agent 加密	信息
192.168.1.11	应用集 11	40	触发器 18	图形 7	自动发现 2	Web监测 192.168.1.11:10050	Template App HTTP Service, Template OS Linux (Template App Zabbix Agent)	已启用 ZBX SNMP JMX IPMI 无				
nginx-0002	应用集 11	40	触发器 18	图形 7	自动发现 2	Web监测 192.168.1.12:10050	Template App HTTP Service, Template OS Linux (Template App Zabbix Agent)	已启用 ZBX SNMP JMX IPMI 无				
nginx-0003	应用集 11	40	触发器 18	图形 7	自动发现 2	Web监测 192.168.1.13:10050	Template App HTTP Service, Template OS Linux (Template App Zabbix Agent)	已启用 ZBX SNMP JMX IPMI 无				

图-37

点击需要的被监控主机链接，打开监控主机设置页面，在 Template (模板) 页面中选择需要链接到该主机的模板，在此选择刚刚创建的模板 count_num 添加即可。如图-38 所示。

主机群组 模板 **主机** 维护 动作 关联项事件 自动发现 服务

主机 所有主机 nginx-0003 已启用 ZBX SNMP JMX IPMI 应用集 11 监控项 40 触发器 18 图形 7 自动发现规则 2

模板 PMI 宏 主机资产记录 加密

链接的模板

名称	动作
Template App HTTP Service	取消链接 取消链接并清理
Template OS Linux	取消链接 取消链接并清理

链接指示器

count_num ×
在此输入搜索

添加

更新 克隆 全克隆 删除 取消

图-38

6) 查看监控数据图形

点击 Monitoring (监控中) -->Graphs (图形)，根据需要选择条件，查看监控图形。如图-39 所示。

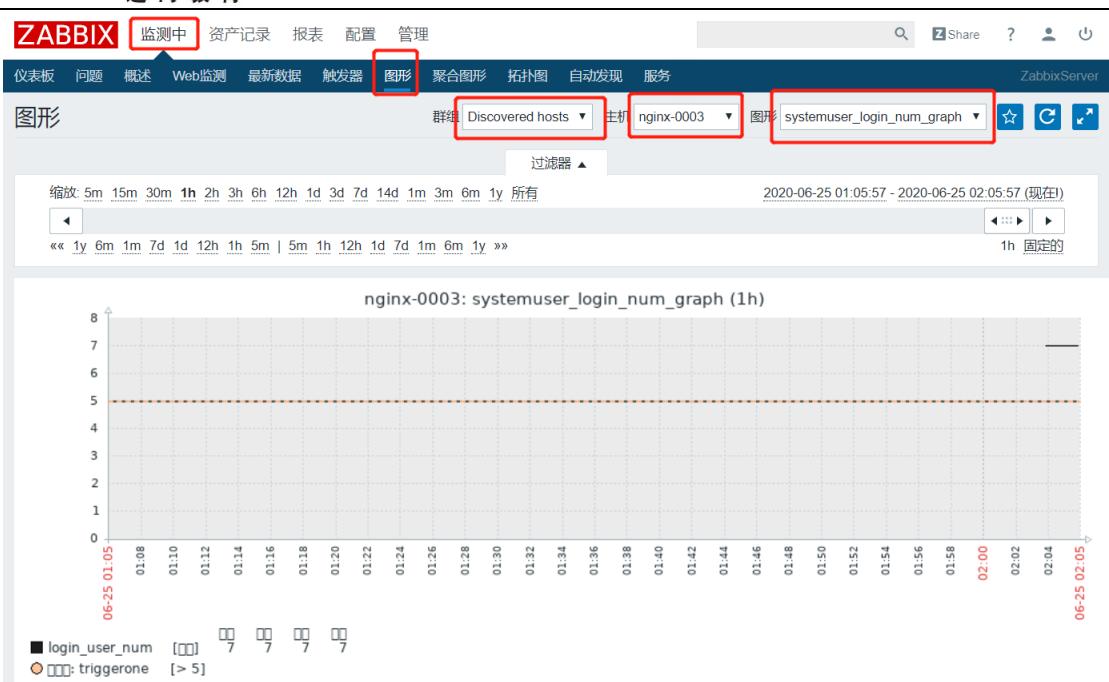


图-39

5. 案例 5、监控报警

- 问题

沿用前面的 Zabbix 练习环境，使用 Zabbix 实现报警功能。

- 注册 163 网易邮箱账号，并且申请授权码
- 设置邮件服务器及收件人邮件
- 当系统登录用户数量超过 5 人时发送报警邮件
- 步骤

实现此案例需要按照如下步骤进行。

步骤一、注册 163 网易邮箱账号，申请授权码

授权码是用于登录第三方邮件客户端的专用密码,启用授权码,避免密码泄漏造成邮箱安全隐患,使用邮件客户端更安心。

1) 登陆网易邮箱, 【设置】—>【POP3/SMTP/IMAP】, 如图-40 所示。



图-40

2) 将两个服务开启。(开启需要手机发送短信。发送完后, 点击“我已发送”), 如图-41 所示。



图-41

出现授权码, 用于第三方邮件客户端登录。如图-42 所示。



图-42

步骤二、创建触发器规则

1)、创建触发器

通过 Configuration (配置) --> Templates (模板)，找到我们在案例四中创建的模板 count_num，点击模板后面的 triggers (触发器)。如图-43 所示。

The screenshot shows the Zabbix configuration interface. The top navigation bar has tabs for '监测中' (Monitoring), '资产记录' (Assets), '报表' (Reports), '配置' (Configuration), and '管理' (Management). The '配置' tab is highlighted. Below the navigation is a sub-navigation bar with '主机群组', '模板' (selected and highlighted with a red box), '主机', '维护', '动作', '关联项事件', '自动发现', and '服务'. The main content area is titled '模板' (Templates). It shows a table with one row for 'count_num'. The table columns include '名称' (Name), '应用集' (Application Set), '监控项' (Metrics), '触发器' (Triggers), '图形' (Graphs), '聚合图形' (Aggregation Graphs), '自动发现' (Auto Discovery), 'Web 监测' (Web Monitoring), and '链接的模板' (Linked Templates). The '触发器' column for 'count_num' contains a red box around the word '触发器'. At the bottom of the table are buttons for '应用' (Apply) and '重设' (Reset).

图-43

点击创建触发器按钮，如图-44 所示。

The screenshot shows the Zabbix configuration interface. The top navigation bar has tabs for '监测中' (Monitoring), '资产记录' (Assets), '报表' (Reports), '配置' (Configuration), and '管理' (Management). The '配置' tab is highlighted. Below the navigation is a sub-navigation bar with '主机群组', '模板' (selected and highlighted with a red box), '主机', '维护', '动作', '关联项事件', '自动发现', and '服务'. The main content area is titled '触发器' (Triggers). It shows a table with one row for 'count_num'. The table columns include '所有模板' (All Templates), '群组' (Group), '主机' (Host), and a '操作' (Operations) column. The '操作' column for 'count_num' contains a red box around the '创建触发器' (Create Trigger) button. At the bottom right of the screen, there is a large red arrow pointing towards this button.

图-44

2) 配置触发器

设置触发器名称，点击 add[添加]表达式，填写表达式：监控项为登录系统的用户数量，最近账户数量大于 5。如图-45、图-46 所示。

触发器

[所有模板](#) / [count_num](#) [应用集 1](#) [监控项 1](#) [触发器](#) [图形 1](#) [聚合图形](#) [自动发现规则](#) [Web 场景](#)
[触发器](#) [依赖关系](#)

名称	triggerone
严重性	未分类 信息 警告 一般严重 严重 灾难
表达式	<input type="text"/>
添加	
表达式构造器	
事件成功迭代	表达式 恢复表达式 无
问题事件生成模式	单个 多重
事件成功关闭	所有问题 所有问题如果标签值匹配

图-45

监控项	count_num: login_user_num	选择
功能	最新的T值 > N	
最后一个(T)	<input type="text"/> 时间	时间
排班	<input type="text"/> 时间	
N	5	
插入 取消		

图-46

步骤三、设置邮件

1) 创建 Media

通过 Administration (管理) --> Media Type (报警媒体类型) --> 选择 Email (邮件)。

如图-47 所示。

The screenshot shows the Zabbix Administration interface under the 'Media' tab. The '报警媒介类型' (Alert Media Types) section is displayed. A red arrow points to the 'Email' entry in the list.

名称	类型	状态	用于动作中	细节
Email	电子邮件	已启用		SMTP服务器: "smtp.163.com", SMTP HELO: "dayuwenzi", SMTP电邮: "dayuwenzi@163.com"
Jabber	Jabber	已启用		Jabber 标识符: "jabber@company.com"

图-47

设置邮件服务器信息，设置邮件服务器及发件人邮件账户信息。如图-48 所示。

【注意：此处的密码为邮箱的授权码】

ZABBIX 监测中 资产记录 报表 配置 管理

一般 agent代理程序 认证 用户群组 用户 报警媒介类型 脚本 队列

报警媒介类型

报警媒介类型 选项

名称	Email
类型	电子邮件
SMTP服务器	smtp.163.com
SMTP服务器端口	25
SMTP HELO	dayuwenzi
SMTP 电邮	dayuwenzi@163.com
安全链接	无 STARTTLS(纯文本通信协议扩展) SSL/TLS
认证	无 Username and password
用户名	dayuwenzi@163.com
密码	*****
已启用	<input checked="" type="checkbox"/>

更新 克隆 删除 取消

图-48

2)为用户添加 Media (设置收件人信息)

在 Administration (管理) -->Users (用户) 中找到选择 Admin 账户。如图-49 所示。

ZABBIX 监测中 资产记录 报表 配置 管理

一般 agent代理程序 认证 用户群组 用户 报警媒介类型 脚本 队列

用户

别名	用户名第一部分	姓氏	用户类型	群组	是否在线?	登录	前端访问	调试模式	状态
<input type="checkbox"/> Admin	Zabbix	Administrator	超级管理员	Zabbix administrators	是 (2020-06-25 00:05:29)	正常	系统默认	停用的	已启用

图-49

点击 Admin 账户后，在弹出的界面中选择 Media (报警媒介) 菜单-->点击 Add(添加) 报警媒介。如图-50 所示。



图-50

点击 Add (添加) 后, 在 Meida Type (类型) 中填写报警类型, 收件人, 时间等信息。如图-51 所示。

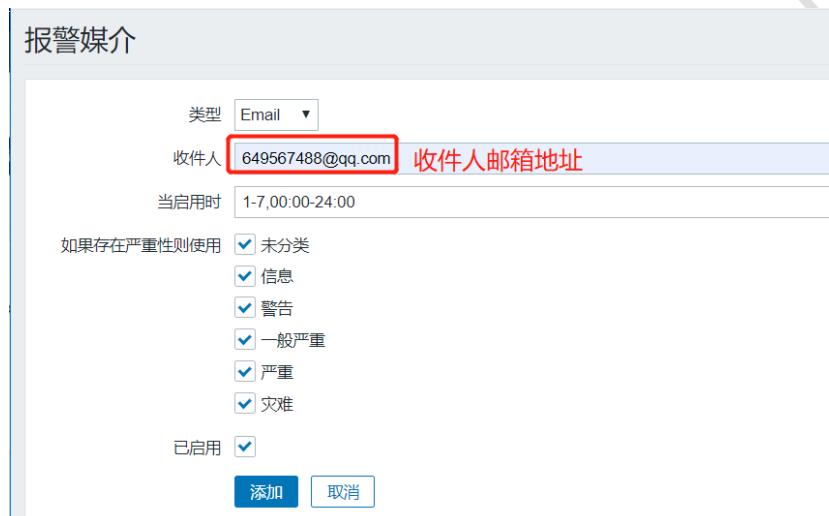


图-51

步骤四、创建 Action 动作

1) Action 动作

Action (动作) 是定义当触发器被触发时的时候, 执行什么行为。

通过 Configuration (配置) -->Actions (动作) -->Create action (创建动作), 注意事件源选择触发器。如图-52 所示。



图-52

2) 配置 Action 动作的触发条件

填写 Action 动作的名称, 配置什么触发器被触发时会执行本 Action 动作 (登录用户数

量大于 5)。如图-53 所示。

The screenshot shows the 'Actions' configuration page. At the top, there are tabs: 动作 (selected), 操作, 恢复操作, and 确认操作. Below the tabs, there is a form for creating a new action:

- 名称:** report_163mail (highlighted with a red box)
- 条件:** 标签 A, 名称 维护状态 非在 维护 (Action A, Maintenance Status not in Maintenance)
- 新的触发条件:** 触发器 = count_num: triggerone (Trigger type = count_num: triggerone). This section is highlighted with a red box.
- 已启用:**
- 操作:** 移除 (Remove) (disabled)
- 按钮:** 添加 (Add), 取消 (Cancel)

图-53

3) 配置 Action 动作的具体行为

配置动作的具体操作行为 (发送信息或执行远程命令), 无限次数发送邮件, 60 秒 1 次, 发送给 Admin 用户, 如图-54、图-55 所示。

The screenshot shows the 'Actions' configuration page with the '操作' tab selected. It includes the following fields:

- 默认操作步骤持续时间:** 1h
- 默认接收人:** Problem: {TRIGGER.NAME}
- 默认信息:** Problem started at {EVENT.TIME} on {EVENT.DATE}
Problem name: {TRIGGER.NAME}
Host: {HOST.NAME}
Severity: {TRIGGER.SEVERITY}

Original problem ID: {EVENT.ID}
{TRIGGER.URL}
- 维护期间暂停操作:**
- 操作:** 步骤 (highlighted with a red box), 新的 (highlighted with a red box)
- 按钮:** 添加 (Add), 取消 (Cancel)

图-54

操作	步骤	细节	开始于	持续时间	动作																																				
1	发送消息给用户: Admin (Zabbix Administrator) 通过 Email		立即地	默认	编辑 移除																																				
<div style="border: 1px solid #ccc; padding: 10px;"> <p>操作细节</p> <p>步骤 <input type="text" value="1"/> - <input type="text" value="1"/> (0 - 无穷大)</p> <p>步骤持续时间 <input type="text" value="0"/> (0 - 使用默认)</p> <p>操作类型 <input type="button" value="发送消息"/></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%;">发送到用户群组</td> <td><input type="text" value="用户群组 Zabbix administrators"/></td> <td style="width: 20%;">动作</td> <td>移除</td> </tr> <tr> <td colspan="2">添加</td> <td colspan="2"></td> </tr> <tr> <td>发送到用户</td> <td><input type="text" value="用户"/></td> <td>动作</td> <td>添加</td> </tr> <tr> <td colspan="2">仅送到 Email</td> <td colspan="2"></td> </tr> <tr> <td colspan="4"> <input checked="" type="checkbox"/> 默认信息 </td> </tr> <tr> <td>条件</td> <td>标签</td> <td>名称</td> <td>动作</td> </tr> <tr> <td colspan="4"> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%;">新的</td> <td style="width: 20%;"></td> <td style="width: 20%;"></td> <td style="width: 20%;"></td> </tr> </table> </td> </tr> <tr> <td colspan="4" style="text-align: center;"> 更新 取消 </td> </tr> </table> </div>						发送到用户群组	<input type="text" value="用户群组 Zabbix administrators"/>	动作	移除	添加				发送到用户	<input type="text" value="用户"/>	动作	添加	仅送到 Email				<input checked="" type="checkbox"/> 默认信息				条件	标签	名称	动作	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%;">新的</td> <td style="width: 20%;"></td> <td style="width: 20%;"></td> <td style="width: 20%;"></td> </tr> </table>				新的				更新 取消			
发送到用户群组	<input type="text" value="用户群组 Zabbix administrators"/>	动作	移除																																						
添加																																									
发送到用户	<input type="text" value="用户"/>	动作	添加																																						
仅送到 Email																																									
<input checked="" type="checkbox"/> 默认信息																																									
条件	标签	名称	动作																																						
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%;">新的</td> <td style="width: 20%;"></td> <td style="width: 20%;"></td> <td style="width: 20%;"></td> </tr> </table>				新的																																					
新的																																									
更新 取消																																									

图-55

4) 购买弹性公网 IP, 按需付费, 并绑定 zabbix-server 主机。如图-56、图-57 所示。

购买弹性公网IP ②

计费模式	<input type="radio"/> 包年/包月	<input checked="" type="radio"/> 按需付费	
区域	<input type="text" value="华北-北京四"/>		
弹性公网IP仅支持绑定在处于相同区域的云资源上。购买后不能更换区域, 请谨慎选择。			
线路	<input checked="" type="radio"/> 全动态BGP	<input type="radio"/> 静态BGP	
<input checked="" type="checkbox"/> 不低于99.95%可用性保障			
公网带宽	<input type="radio"/> 按带宽计费 流量较大或较稳定的场景	<input type="radio"/> 按流量计费 流量小或流量波动较大场景	加入共享带宽 多业务流量错峰分布场景
带宽大小	<input type="text" value="1"/> 2 5 10 100 200		<input type="radio"/> 自定义 <input type="text" value="1"/> - +
带宽名称	<input type="text" value="bandwidth-4040"/>		

图-56

IPv6公测中，诚邀您立即体验：IPv6 EIP
公测期间IPv6转换功能免费，带宽正常收费。



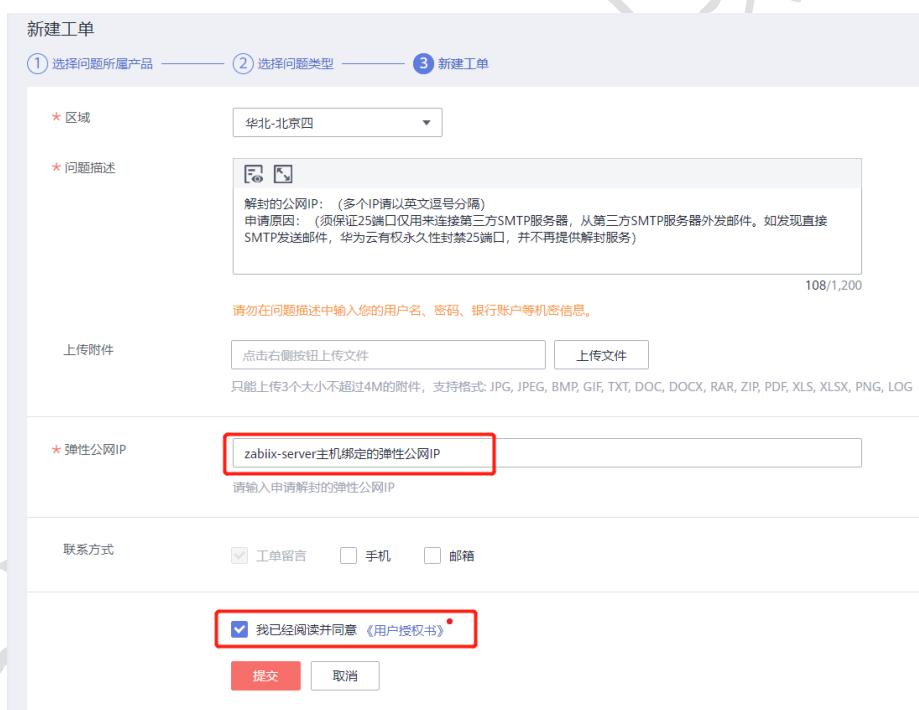
The screenshot shows the 'Elastic Public IP' management page. A table lists an IP entry: 121.36.101.254, bound to a Zabbix server named 'zabbix-server'. The 'zabbix-server' row is highlighted with a red box.

图-57

5)、开启 25 端口

为了提升华为云 IP 地址发邮件的质量,基于安全考虑,TCP 25 端口出方向默认被封禁,无法使用 TCP 25 端口连接外部地址。如果存在特殊场景,您必须在云服务器上使用 TCP 25 端口进行对外连接,请提交 25 端口解封申请。

【工单】→【新建工单】→【弹性云服务器】→【25 端口解封】→【新建工单】
填入 zabbix-server 主机的弹性公网 IP, 不需要通过手机、邮件联系, 6 遵守许可, 点击提交即可。如图-58 所示。



The screenshot shows the 'Create Work Order' page. Step 3: Create Work Order is selected. The 'Problem Description' section contains a note about unblocking port 25 for SMTP. The 'Elastic Public IP' field is filled with 'zabbix-server' and has a red box around it. The 'I have read and agree to the User License Agreement' checkbox is checked. The 'Submit' button is at the bottom.

图-58

6)、测试效果

开启终端登录 nginx-0003 (登录用户数大于 5), 然后登录监控端 Web 页面, 在仪表盘中查看问题报警 (需要等待一段时间)。并查看自己的 QQ 邮箱。如图-59 所示。



图-59

[练习]

告警主机: {HOSTNAME1}
告警时间: {EVENT.DATE} {EVENT.TIME}
告警等级: {TRIGGER.SEVERITY}
告警信息: {TRIGGER.NAME}
告警项目: {TRIGGER.KEY1}
问题详情: {ITEM.NAME} : {ITEM.VALUE}
当前状态: {TRIGGER.STATUS} : {ITEM.VALUE1}
事件 ID: {EVENT.ID}

6. 案例 6: Zabbix 分布式部署

• 问题

本案例要求部署 Zabbix 分布式

- Zabbix 分布式 Proxy 安装
- Zabbix 分布式监控 Linux

• 方案

Zabbix Proxy 可以代替 Zabbix Server 检索客户端的数据，然后把数据汇报给 Zabbix Server，并且在一定程度上分担了 Zabbix Server 的压力。Zabbix Proxy 可以非常简便的实现了集中式、分布式监控。

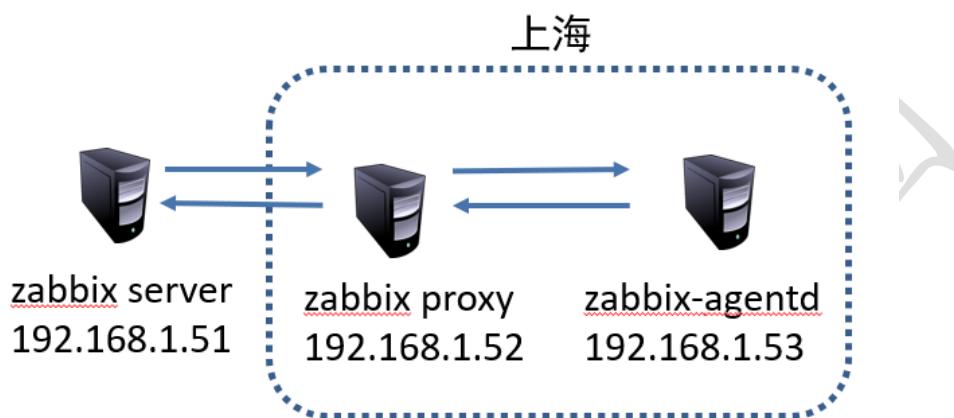
Zabbix Proxy 使用场景:

监控远程区域设备

监控本地网络不稳定区域

当 Zabbix 监控上千设备时，使用它来减轻 Server 的压力

简化分布式监控的维护



- 步骤

实现此案例需要按照如下步骤进行。

步骤一、Zabbix 分布式 Proxy 安装

1)、购买华为云服务器

基础配置: 无

网络配置: 手动分配 IP 地址 192.168.1.52

高级配置: 云服务器名称 zabbix-proxy

确认配置: 1 台

基础配置: 无

网络配置: 手动分配 IP 地址 192.168.1.53

高级配置: 云服务器名称 zabbix-agentd

确认配置: 1 台

2)、安装软件

```
[root@zabbix-proxy ~]# scp root@192.168.1.252:/root/project3/DAY04/zabbix-3.4.4.tar.gz /root
```

```
[root@zabbix-proxy ~]# yum -y install net-snmp-devel curl-devel autoconf libevent-devel
[root@zabbix-proxy ~]# yum -y install mariadb mariadb-server mariadb-devel
[root@zabbix-proxy ~]# systemctl enable --now mariadb
[root@zabbix-proxy ~]# useradd -s /sbin/nologin zabbix
[root@zabbix-proxy ~]# tar -xf zabbix-3.4.4.tar.gz
[root@zabbix-proxy ~]# cd zabbix-3.4.4/
[root@zabbix-proxy zabbix-3.4.4]# ./configure --enable-proxy --enable-agent --with-mysql --with-net-snmp --with-libcurl
[root@zabbix-proxy zabbix-3.4.4]# make && make install
```

查看 Zabbix Proxy 的版本

```
[root@zabbix-proxy zabbix-3.4.4]# /usr/local/sbin/zabbix_proxy -version
```

3)、Mysql 数据库初始化

```
[root@zabbix-proxy ~]# mysql
MariaDB [(none)]> create database zabbix character set utf8;
Query OK, 1 row affected (0.00 sec)
```

```
MariaDB [(none)]> grant all privileges on zabbix.* to zabbix@'127.0.0.1' identified by 'zabbix123';
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> use zabbix;
MariaDB [zabbix]> source /root/zabbix-3.4.4/database/mysql/schema.sql;
```

4) 、Zabbix 分布式 Proxy 配置

```
[root@zabbix-proxy ~]# sed -n "24p;43p;85p;150p;161p;176p;184p;198p" /usr/local/etc/zabbix_proxy.conf
Server=192.168.1.51      #第 24 行, zabbix server 服务器的地址或主机名
Hostname=shproxy          #第 43 行, 代理服务器名称, 需要与 zabbix server 添加代理时候 proxy name 一致。
LogFile=/tmp/zabbix_proxy.log      #第 85 行, 代理服务器的日志。
DBHost=127.0.0.1          #第 150 行, 数据库地址。
DBName=zabbix              #第 161 行, 数据库名。
DBUser=zabbix              #第 176 行, 数据库用户名。
DBPassword=zabbix123       #第 184 行, 数据库登陆的密码
DBPort=3306                #第 198 行, 数据库的端口。
[root@zabbix-proxy ~]# zabbix_proxy
[root@zabbix-proxy ~]# netstat -atnpu | grep zabbix_proxy
tcp 0 0 0.0.0.0:10051 0.0.0.* LISTEN 28582/zabbix_proxy
```

5)、创建代理

通过管理--> agent 代理程序, 点击创建代理, 如图-60、图 61 所示。



图-60

注意:agentd代理程序名称一定要与192.168.1.52(zabbix-proxy主机)配置文件中Hostname设置的名称一致。

agent代理程序名称	shproxy
系统代理程序模式	<input checked="" type="radio"/> 主动式 <input type="radio"/> 被动式
主机 agent代理程序的主机	
其它主机	Zabbix server
描述	

图-61

步骤二、Zabbix 分布式监控 Linux

1)、部署一台 Zabbix 客户端

也可以使用案例 2 制作的 RPM 包完成 Zabbix 客户端的快速部署。

```
[root@zabbix-agent ~]# yum -y install pcre-devel
[root@zabbix-agent ~]# useradd -s /sbin/nologin Zabbix
[root@zabbix-agent ~]# scp root@192.168.1.252:/root/project3/DAY04/zabbix-3.4.4.tar.gz /root
[root@zabbix-agent ~]# tar -xf /root/zabbix-3.4.4.tar.gz
[root@zabbix-agent ~]# cd /root/zabbix-3.4.4/
[root@zabbix-agent zabbix-3.4.4]# ./configure --enable-agent
```

```
[root@zabbix-agent zabbix-3.4.4]#make && make install
[root@zabbix-agent      zabbix-3.4.4]#sed      -n      '30p;93p;134p;145p;264p;280p'
/usr/local/etc/zabbix_agentd.conf
LogFile=/tmp/zabbix_agentd.log
Server=192.168.1.52          #此处指向 zabbix-proxy 主机的 IP 地址
ServerActive=192.168.1.52      #此处指向 zabbix-proxy 主机的 IP 地址
Hostname=zabbix-agent
Include=/usr/local/etc/zabbix_agentd.conf.d/
UnsafeUserParameters=1

[root@zabbix-agent zabbix-3.4.4]# zabbix_agentd
[root@zabbix-agent zabbix-3.4.4]# netstat -atnp | grep zabbix_agentd
tcp 0 0 0.0.0.0:10050 0.0.0.0:* LISTEN 11940/zabbix_agentd
```

2)、创建主机

通过 Configuration (配置) --> Hosts (主机) , 如图-62、图-63 所示。



图-62

由 agentd 代理程序监测选择创建好的 agent 代理程序(shproxy)。

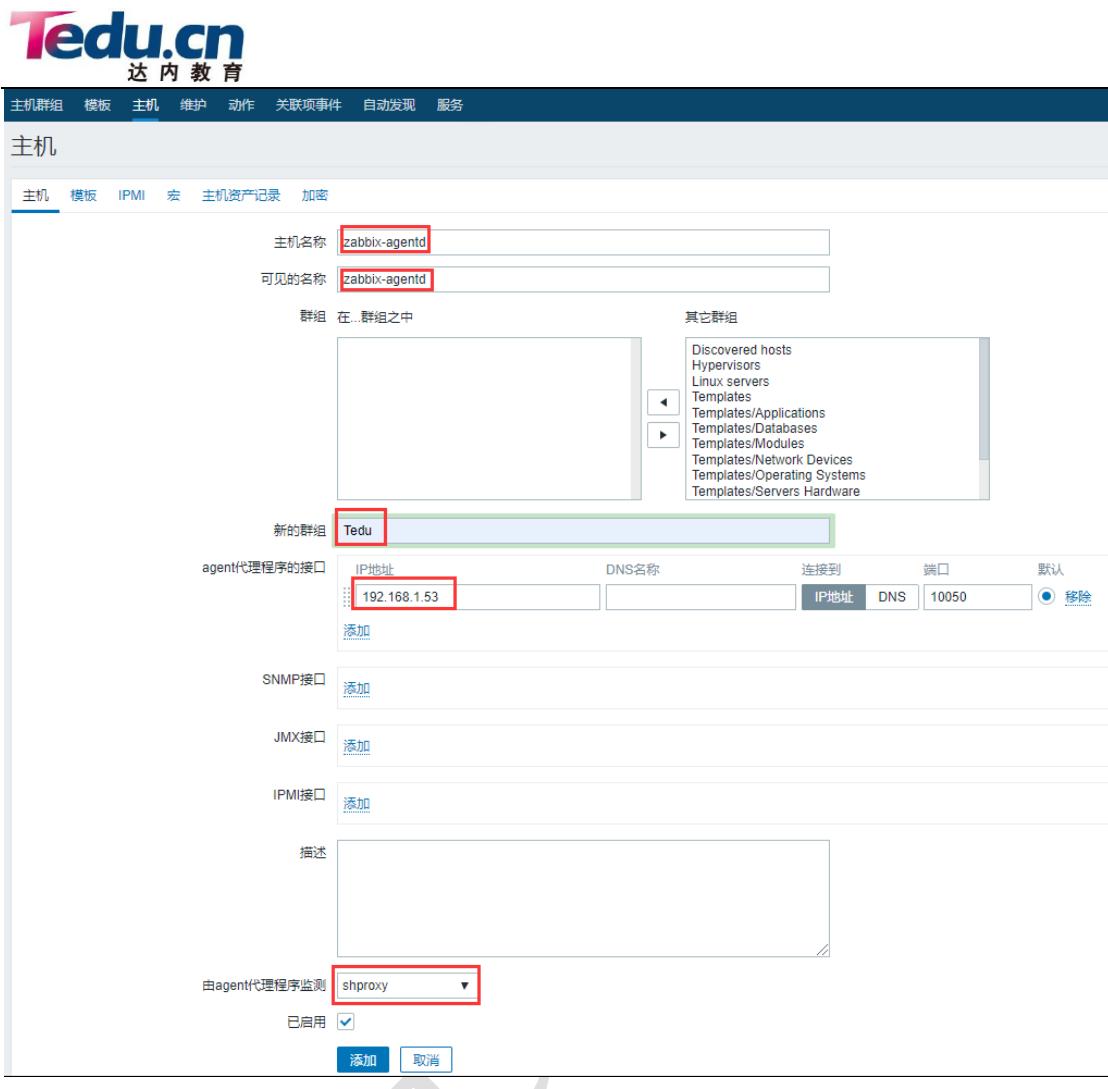


图-63

在 Template (模板) 页面中选择需要链接到该主机的模板，在此选择模板。如图 64 所示。

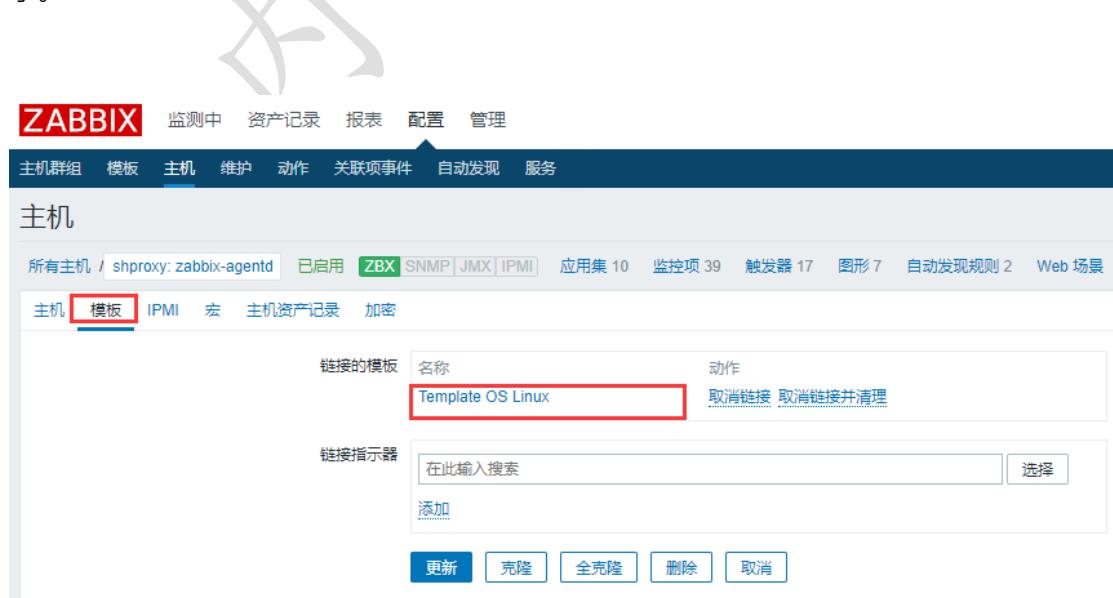


图-64

3)、查看监控数据图形

点击 Monitoring (监控中) -->Graphs (图形)，根据需要选择条件，查看监控图形。如图-65 所示。



图-65

7. 案例 7：Grafana 数据可视化

- 问题

本案例要求部署 Grafana

- 安装 Grafana 与配置
- Grafana 展示 Zabbix 数据
- 方案

Grafana 是一款采用 go 语言编写的开源应用，主要用于大规模指标数据的可视化展现。权威的资料网站是官网 (<http://docs.grafana.org/>)。

- 步骤

实现此案例需要按照如下步骤进行。

步骤一、安装 Grafana 与配置

1) 安装软件

```
[root@zabbix-server ~]# scp 192.168.1.252:/root/project3/DAY04/grafana-6.3.6-1.x86_64.rpm /root
[root@zabbix-server ~]# yum localinstall /root/grafana-6.3.6-1.x86_64.rpm
[root@zabbix-server ~]# systemctl enable grafana-server --now
[root@zabbix-server ~]# netstat -anptu | grep grafana
tcp6 0 0 :::3000 ::*:20920/grafana-serve
```

2) 创建监听器并添加后端服务器【监听端口 3000】，如图-66、图-67、图-68 所示。

The screenshot shows the 'Create Listener' configuration page. The 'Name' field is set to 'listener-3000'. The 'Frontend Protocol/Port' dropdown is set to 'TCP' with the port value '3000' highlighted with a red box. A note below states: '四层监听请选择TCP、UDP；七层监听请选择HTTP、HTTPS。选择HTTPS协议时，后端协议只能使用HTTP协议。' (For Layer 4 listening, choose TCP or UDP; for Layer 7 listening, choose HTTP or HTTPS. When choosing HTTPS protocol, the backend protocol can only be HTTP.)

图-66

(1) 配置监听器 ————— (2) 配置后端服务器组 —————

The screenshot shows the 'Create Backend Server Group' configuration page. The 'Backend Server Group' tab is selected. The 'Name' field is set to 'server_group-grafana'. The 'New Creation' button is highlighted with a blue box.

图-67

添加后端服务器

The screenshot shows the 'Add Backend Server' configuration page. A note at the top states: '后端服务器的安全组规则必须放通100.125.0.0/16网段，否则会导致健康检查异常。了解更多' (The security group rules for the backend server must allow traffic through the 100.125.0.0/16 network segment, otherwise it will cause health check anomalies. Learn more). The 'Batch Add Port' field is set to '3000' and highlighted with a red box. The table below lists the server details:

Private IP Address	Cloud Server	Added Port	Weight	Operation
192.168.1.51	zabbix-server 1 vCPUs 1 GB s6.small.1	--	1	Remove

图-68

3) 访问 <http://公网 IP:3000>，登录 grafana 管理界面。如图-69 所示。

登录用户名 admin 密码 admin

第一次登录时，需要修改密码。

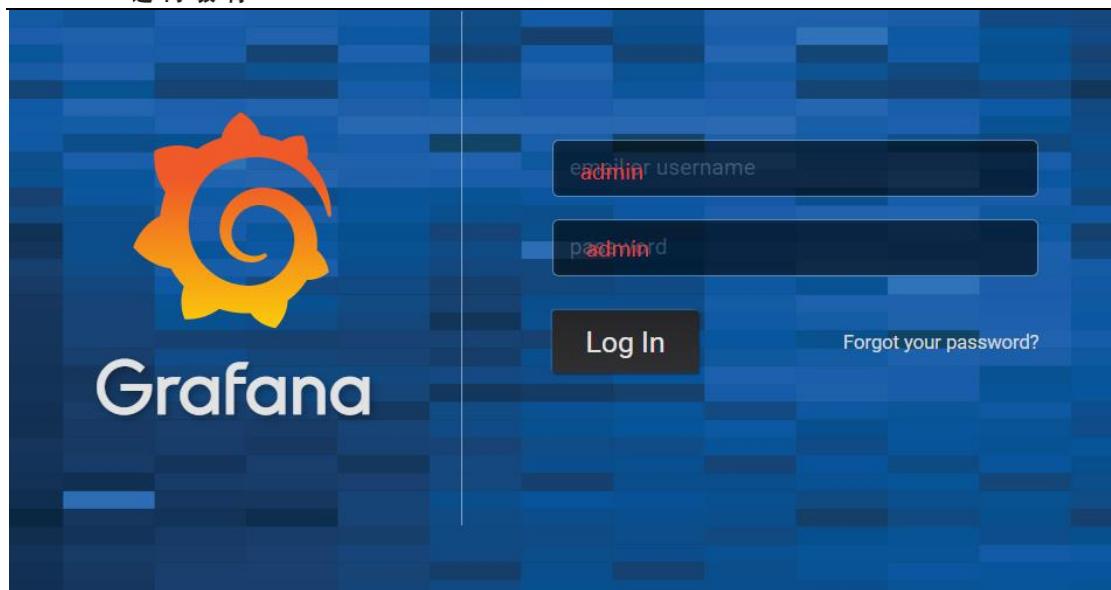


图-69

4) 安装插件，支持 Grafana 连接 zabbix-server

Grafana 是一个纯静态的仪表盘，本身并没有数据，需要配置数据源从哪里获取，Grafana 支持从 Zabbix 中获取数据。Grafana 优化了图形的展现，可以用来做监控大屏。

方式一：在线安装插件（需要连接外网，较慢。）

```
[root@zabbix-server ~]# grafana-cli -help
#列出远端可用的插件
[root@zabbix-server ~]# grafana-cli plugins list-remote
#安装插件
[root@zabbix-server ~]# grafana-cli plugins install alexanderzobnin-zabbix-app 3.12.2
installing alexanderzobnin-zabbix-app @ 3.12.2
from: https://grafana.com/api/plugins/alexanderzobnin-zabbix-app/versions/3.12.2/download
into: /var/lib/grafana/plugins

? Installed alexanderzobnin-zabbix-app successfully

Restart grafana after installing plugins . <service grafana-server restart>
[root@zabbix-server ~]# systemctl restart grafana-server #重启服务
```

方式二：手动下载离线安装插件

若在线方式安装超时，可以手动下载安装

第一步：访问官网，找到要下载的插件，并且选择版本，下载。

第二步：将插件的压缩包释放到/var/lib/grafana/plugins 默认目录下，重启服务即可。

刷新之后，在主页中会出现一个 App，点击后面的【Enable now】。如图-70 所示。

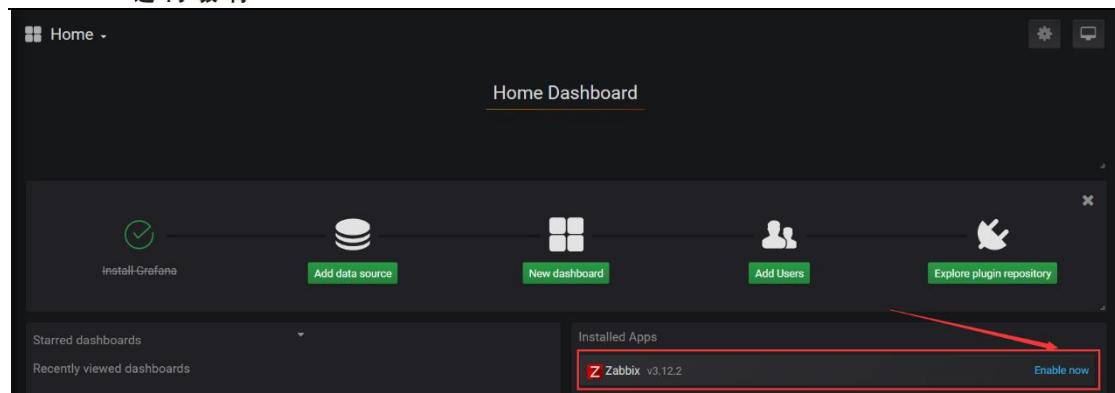


图-70

激活插件，点击【Enable】，如图-71所示。

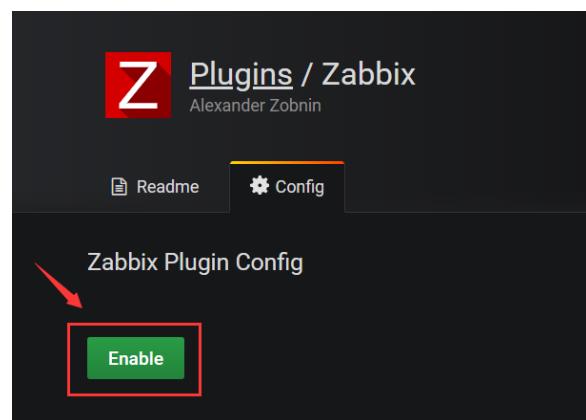


图-71

5) 安装好插件 alexanderzobnin-zabbix-app，点击【Data Source】就可以添加数据来源。如图-72、图-73 所示。

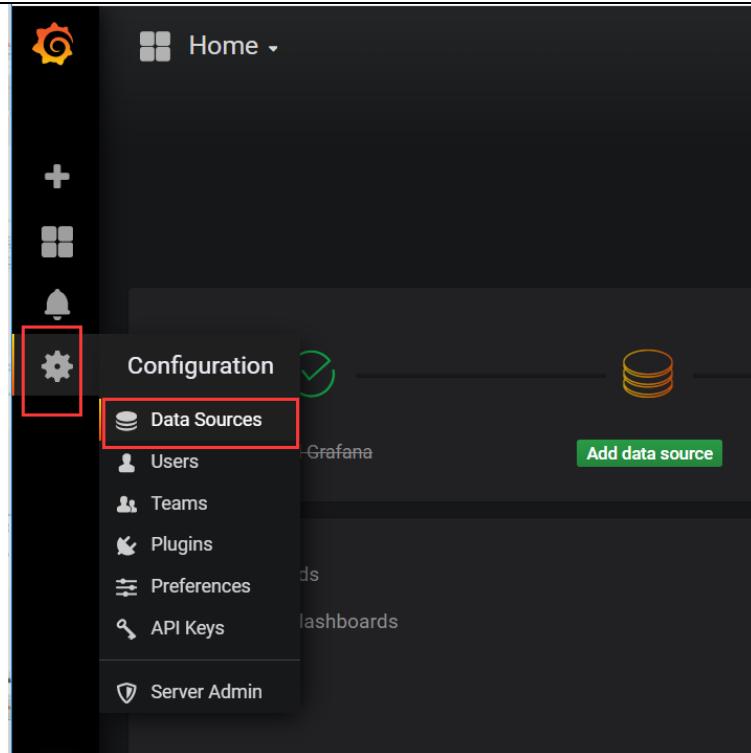


图-72

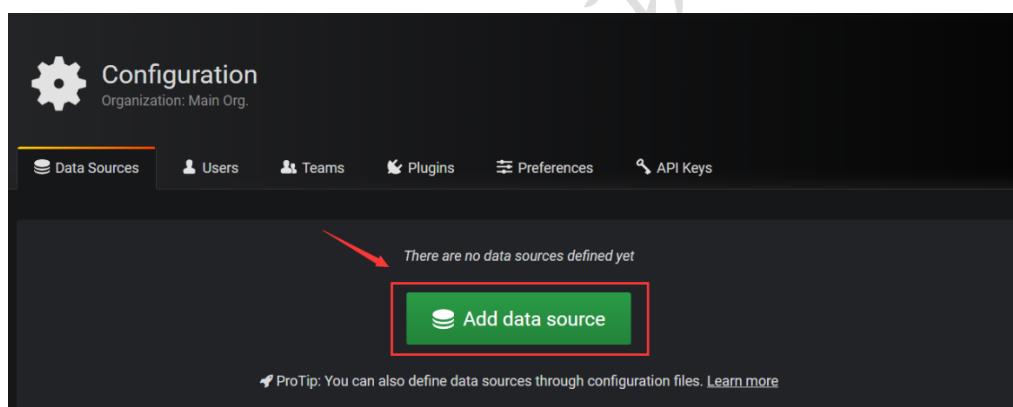


图-73

在搜索框里输入 zabbix 即可快速找到，如图-74 所示。

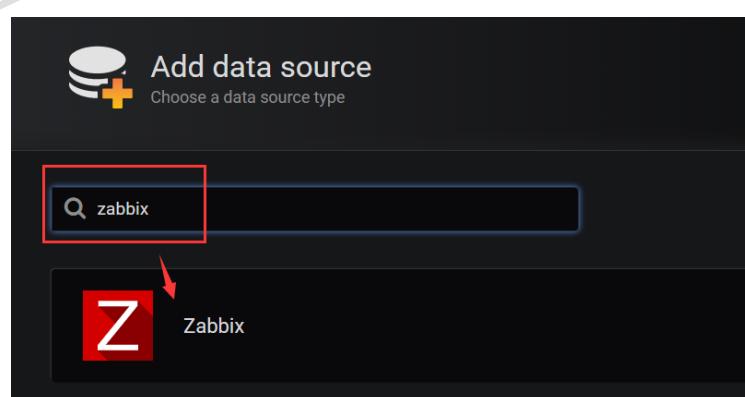


图-74

在 URL 中输入 Zabbix 的接口地址，如：http://192.168.1.51:8090/api_jsonrpc.php

在 Username 和 Password 分别输入 Zabbix 的登录用户及密码。如图-75 所示。

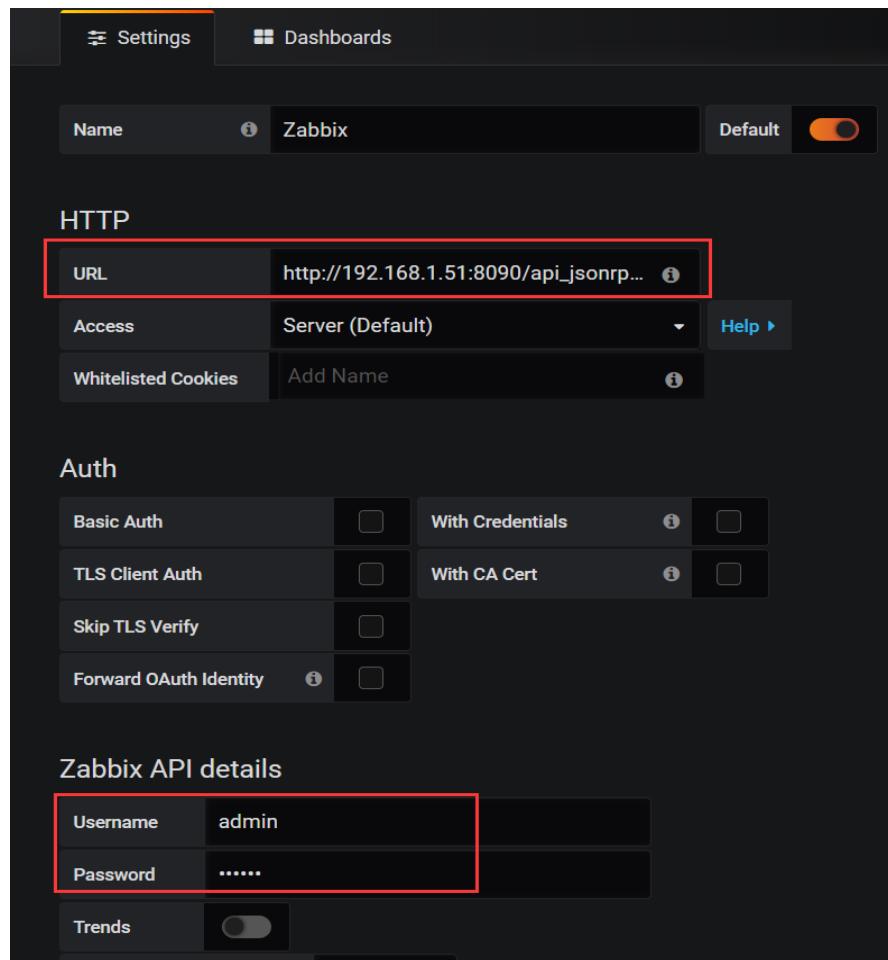


图-75

Grafana 的 logo，即当前页为 Grafana 的主页，在任何页面点击 Grafana 的 logo，都会跳转到主页。如图-76 所示。



图-76

新建按钮，用于创建 Dashboard，文件夹、以及导入外部的 Dashboard。如图-77 所示。



图-77

用于查看或管理 Dashboard。如图-78 所示。



图-78

Explore(探索)，主要用于快速编写查询语句，来查询数据中的数据。如图-79 所示。



图-79

告警设置，可以设置邮件、钉钉、短信的告警。如图-80 所示。



图-80

设置。包括这是 Data Sources (数据源)、Users (用户)、Teams (创建团队)、Plugin (插件查找)、preferences (个人性设置) API keys (API 密钥)。如图-81 所示。

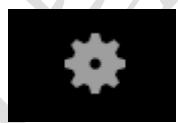


图-81

管理设置，包括 Users (用户创建)、Org (组织创建)、Settings (设置参数查看)、stats (Grafana 本身状态信息统计)、upgrade (Grafana 软件升级)。如图-82 所示。



图-82

步骤二、Grafana 展示 Zabbix 数据

1) 创建仪表盘，如图-83 所示。

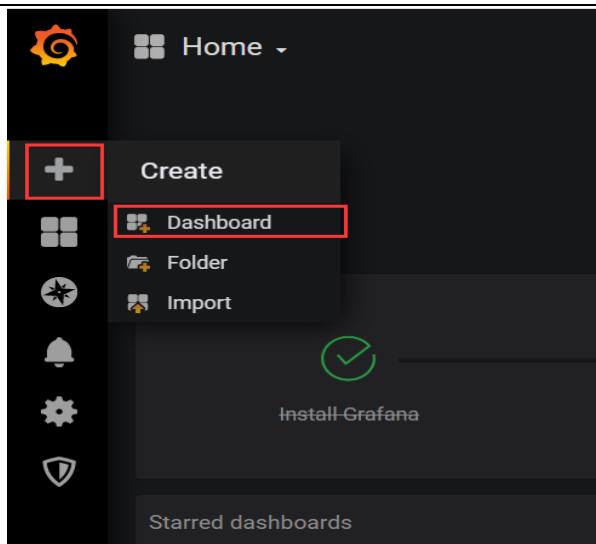


图-83

自定义仪表盘名称，保存仪表盘。如图-84、图-85 所示。

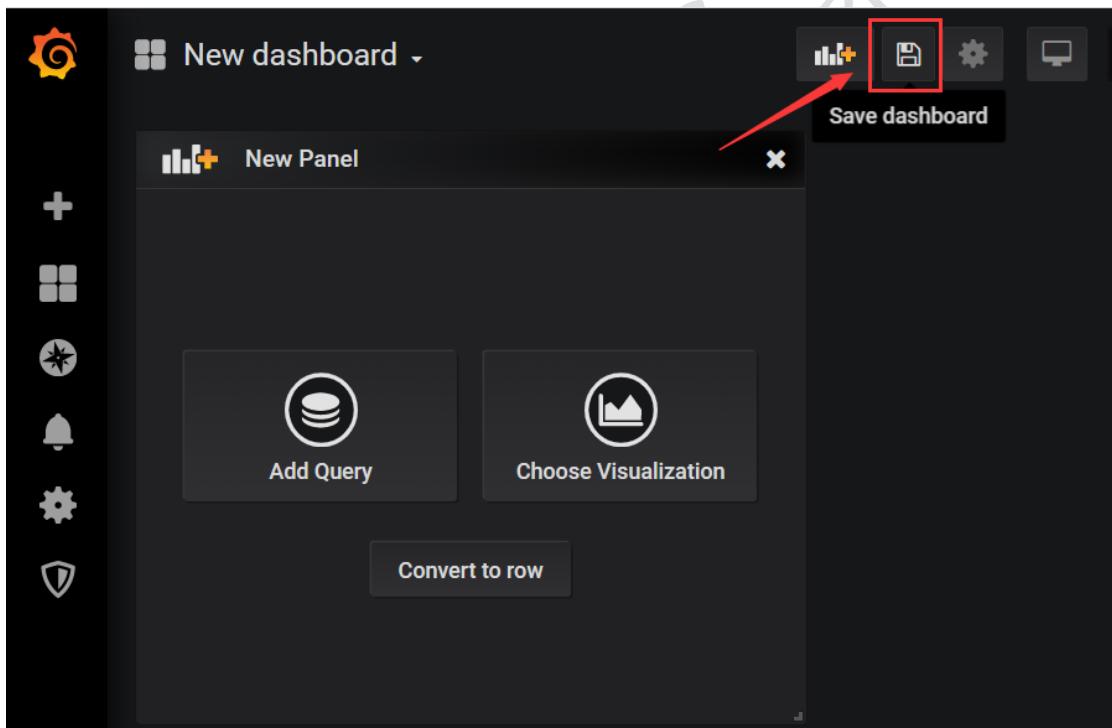


图-84

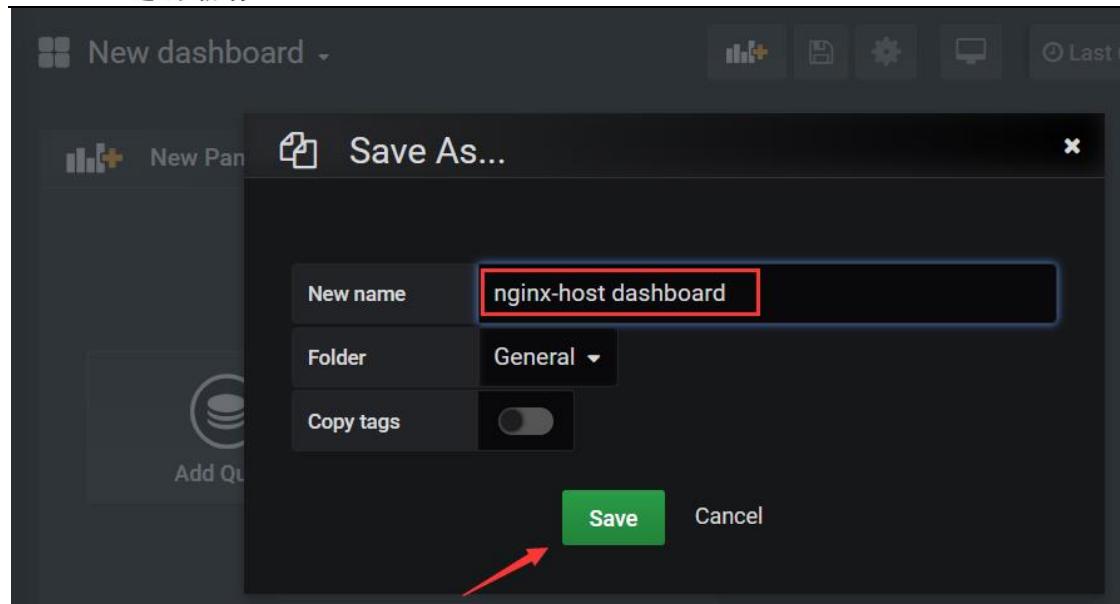


图-85

2)、添加图形 (CPU 负载), 如图-86。

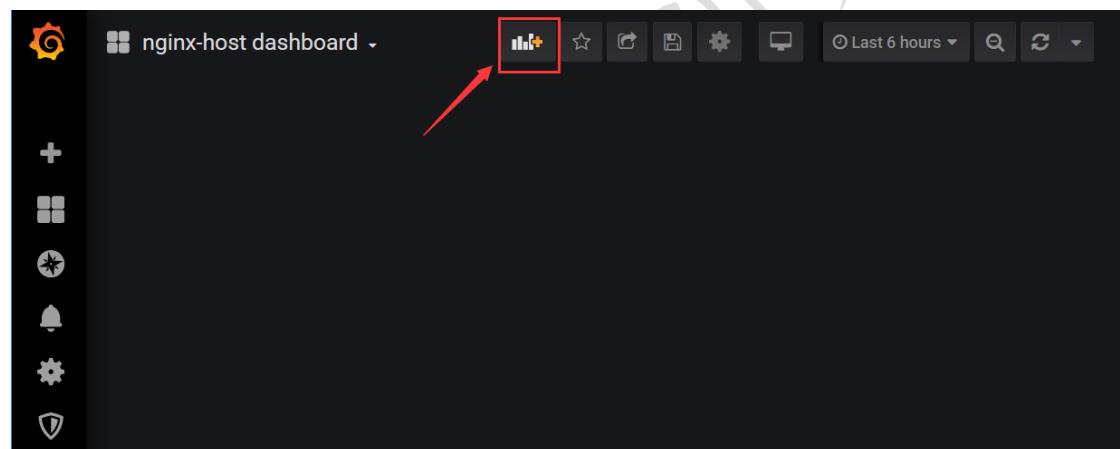


图-86

创建图表, 如图-87 所示。

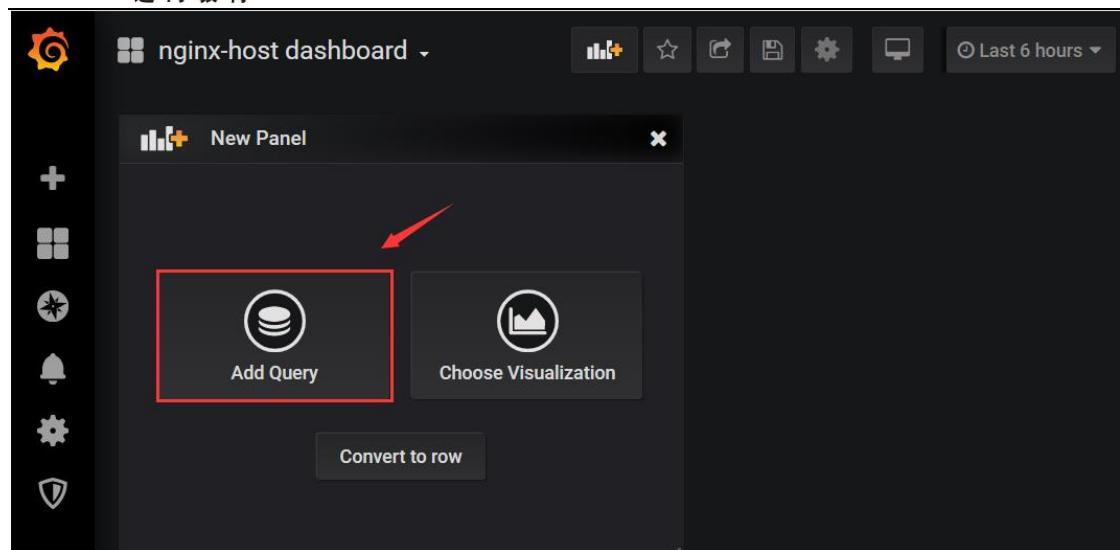


图-87

显示 192.168.1.11(nginx-0001) 主机的负载，如图-88 所示。

Item 使用正则表达式，匹配所有的负载。

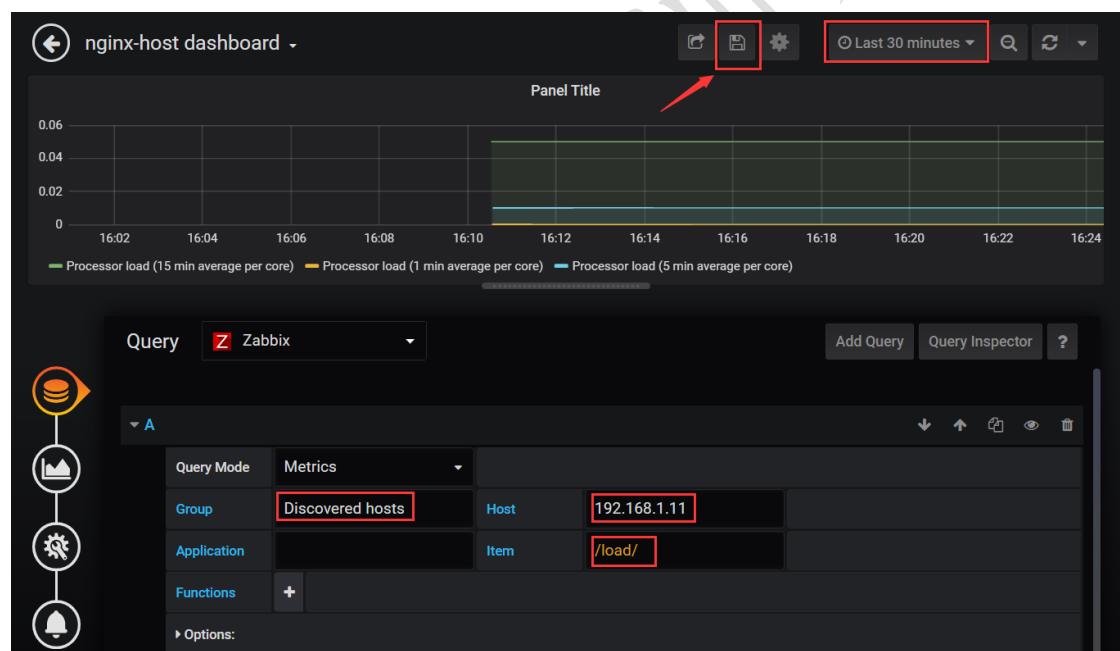


图-88

3)、更改标题，如图-89、图-90 所示。

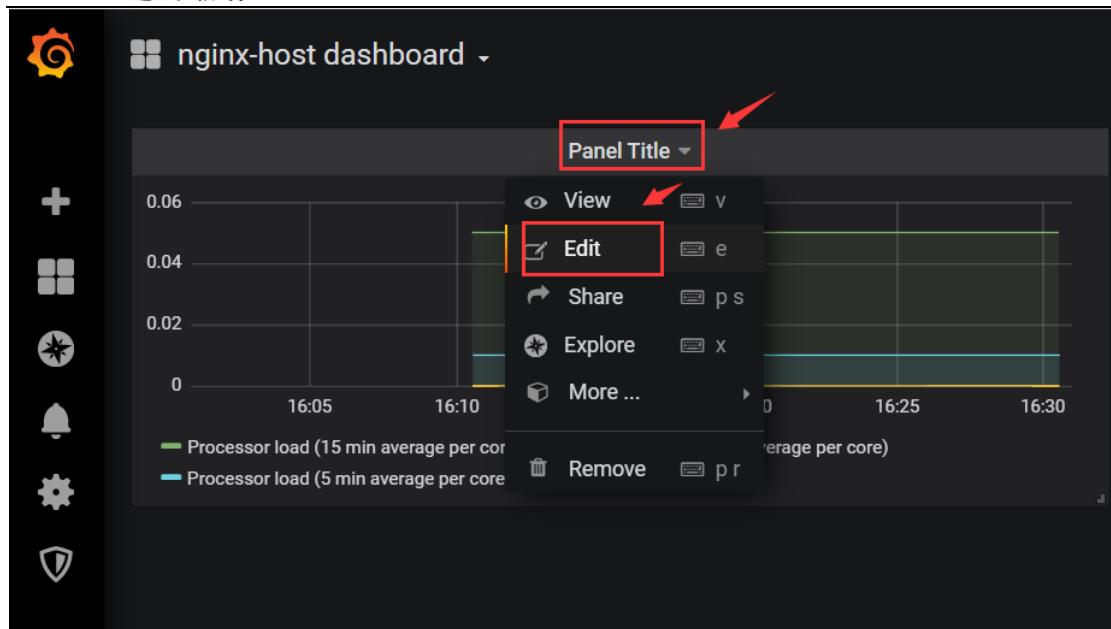


图-89

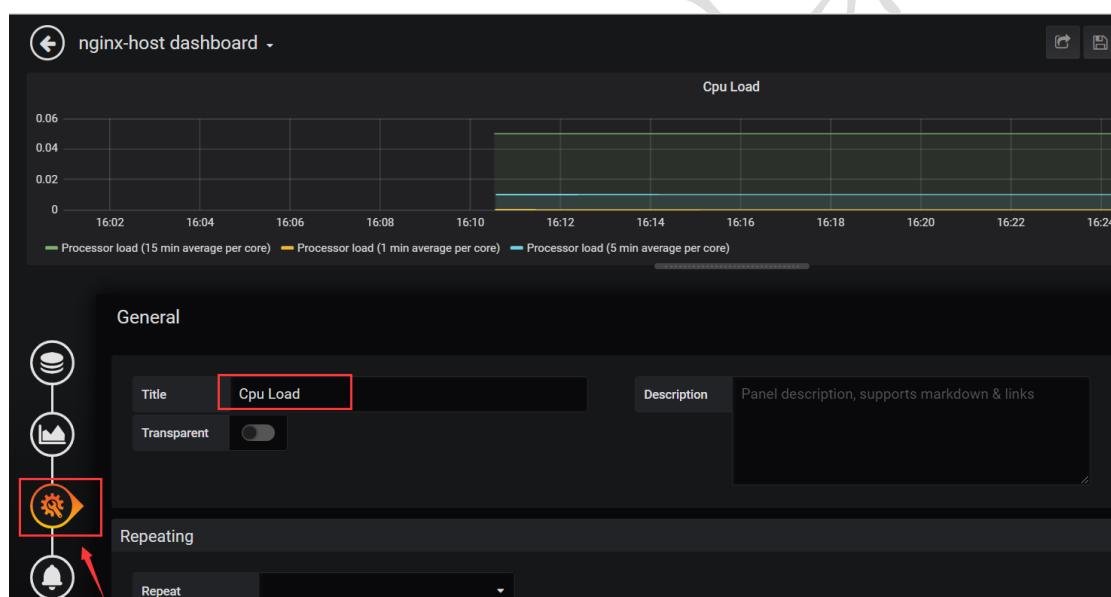


图-90

4) 、优化图表，如图-91 所示。优化完成后，进行保存。

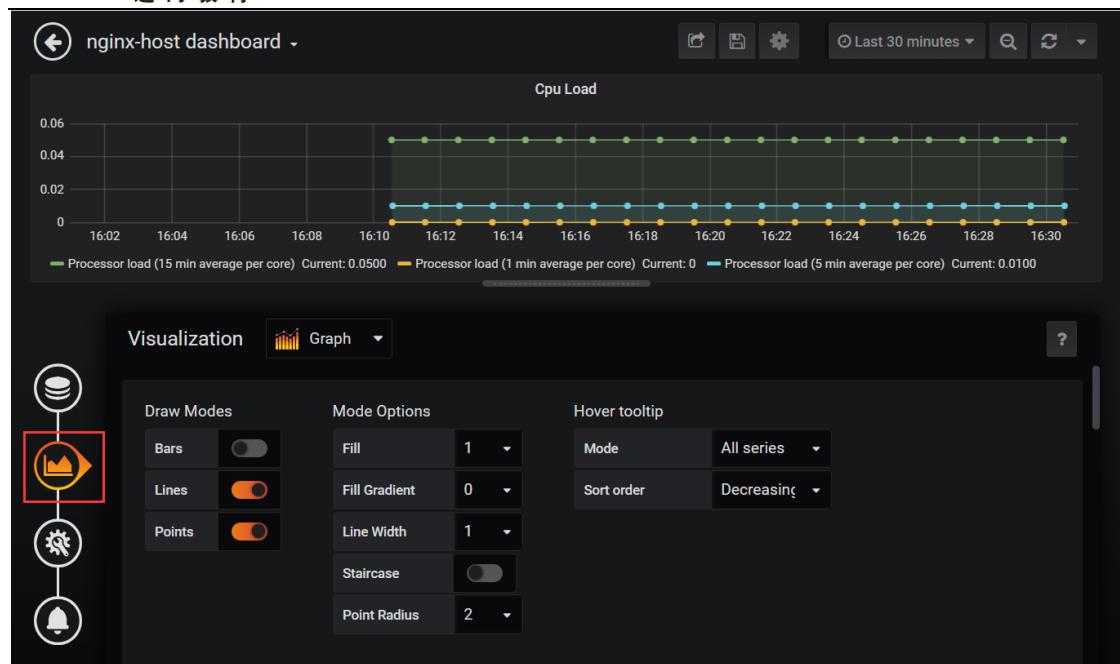


图-91

5)、添加图形（服务器网卡流量）。对现有的图表，进行复制更改，如图-92 所示。

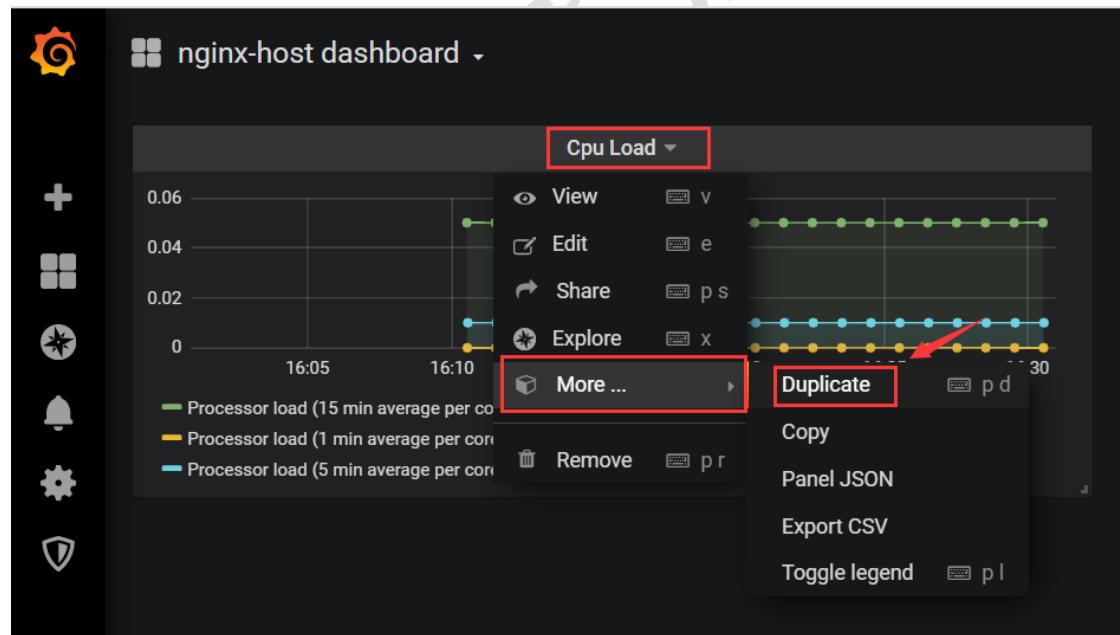


图-92

对复制出来的图标进行编辑，如图-93、图-94、图-95 所示。

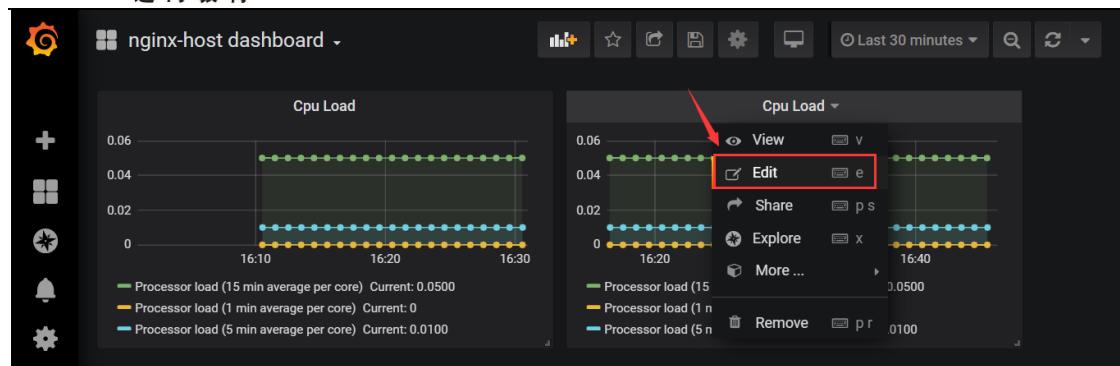


图-93

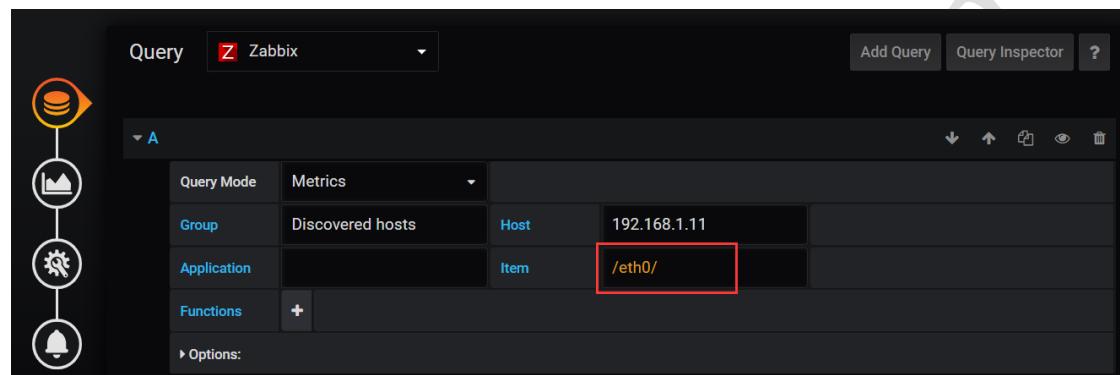


图-94

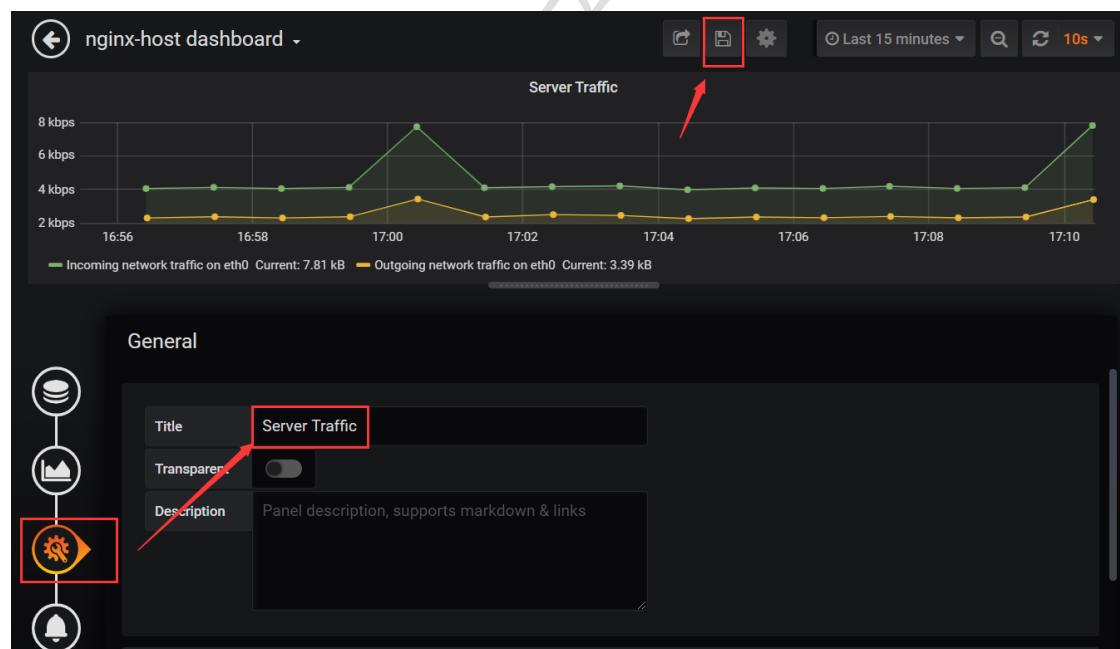


图-95

8. 案例 8：部署企业私有镜像仓库 Harbor

- 问题

本案例要求在华为云服务器上构建私有镜像仓库

- 部署私有镜像仓库 harbor
- 部署 docker 主机测试

- 方案

私有镜像仓库有许多优点

- 1) 节省网络带宽，针对于每个镜像不用每个人都去中央仓库上面去下载，只需要从私有仓库中下载即可；
- 2) 提供镜像资源利用，针对于公司内部使用的镜像，推送到本地的私有仓库中，以供公司内部相关人员使用。

VMware 公司开源了企业级 Registry 项目 Harbor，其的目标是帮助用户迅速搭建一个企业级的 Docker registry 服务。

准备如表-2 所示的实验环境，配置主机名称、IP 地址。

表-2 主机列表

主机名称	IP 地址	角色
harbor	192.168.1.67	Harbor 私有镜像仓库
docker	192.168.1.68	Docker 主机

- 步骤

实现此案例需要按照如下步骤进行。

步骤一、购买华为云服务器更新 ecs-proxy 配置文件

1)、购买华为云服务器

基础配置：无

网络配置：手动分配 IP 地址 192.168.1.67

高级配置：云服务器名称 harbor

确认配置：1台

基础配置: 无

网络配置: 手动分配 IP 地址 192.168.1.68

高级配置: 云服务器名称 docker

确认配置: 1 台

2)、更新/etc/hosts

```
[root@ecs-proxy ~]# cat >> /etc/hosts <<EOF  
192.168.1.67 harbor  
192.168.1.68 docker  
EOF
```

3)、更新/root/ansible/hosts 配置文件

```
[root@ecs-proxy ~]# cat >> /root/ansible/hosts <<EOF  
[harbor]  
192.168.1.67  
[docker]  
192.168.1.68  
EOF  
#将最新的/etc/hosts 配置文件更新到所有的云主机上  
[root@ecs-proxy ansible]# ansible all -m copy -a 'src=/etc/hosts dest=/etc'
```

4) 部署 docker 安装源

```
[root@ecs-proxy ansible]# mkdir /var/ftp/localrepo/docker  
[root@ecs-proxy ansible]# cp -a /root/project3/DAY04/docker/* /var/ftp/localrepo/docker  
[root@ecs-proxy ansible]# createrepo --update /var/ftp/localrepo
```

5) 分发实验所需软件包

Docker Compose 是用于定义和运行多容器 Docker 应用程序的工具。

Harbor 的每个组件都是以 Docker 容器的形式构建的，所以使用 Docker Compose 来对它进行部署。

```
[root@ecs-proxy ansible]# ansible harbor -m copy -a "src=/root/project3/DAY04/docker-  
compose dest=/root"  
[root@ecs-proxy ansible]# ansible harbor -m copy -a "src=/root/project3/DAY04/harbor-offline-  
installer-v1.2.0.tgz dest=/root"
```

步骤二、部署私有镜像仓库 harbor

1)、开启路由转发

```
[root@harbor ~]# echo 'net.ipv4.ip_forward = 1' >> /etc/sysctl.conf
```

```
[root@harbor ~]# sysctl -p
```

2)、安装 docker 服务，并启动

```
[root@harbor ~]# yum -y install docker-ce
[root@harbor ~]# systemctl start docker
[root@harbor ~]# systemctl enable docker
```

3)、部署 harbor

```
[root@harbor ~]# mv /root/docker-compose /usr/local/bin/
[root@harbor ~]# chmod +x /usr/local/bin/docker-compose
[root@harbor ~]# tar -xf harbor-offline-installer-v1.2.0.tgz -C /usr/local/
[root@harbor ~]# cd /usr/local/harbor
[rootharbor harbor]# ls
common docker-compose.notary.yml harbor_1_1_0_template harbor.v1.2.0.tar.gz LICENSE prepare
docker-compose.clair.yml docker-compose.yml harbor.cfg install.sh NOTICE upgrade
```

4) 修改配置文件

```
#访问 harbor 管理界面的地址，改为本机的 IP 地址
[root@harbor harbor]# sed -i '/^hostname/s/reg.mydomain.com/192.168.1.67/'
/usr/local/harbor/harbor.cfg
[root@harbor harbor]# sed -i '121 s/80:80/8099:80/' /usr/local/harbor/docker-compose.yml
#默认是访问 80 端口，但因为 80 端口与 Nginx 集群冲突，故改变访问端口为 8099
[root@harbor harbor]# sed -i '24 s/$ui_url/$ui_url:8099/'
/usr/local/harbor/common/templates/registry/config.yml
```

```
[root@harbor harbor]# ./install.sh
.....
✓ ----Harbor has been installed and started successfully.----
```

Now you should be able to visit the admin portal at <http://192.168.1.67>.
For more details, please visit <https://github.com/vmware/harbor>.

```
[root@harbor harbor]# netstat -antpu | grep 8099
tcp6 0 0 :::8099 ::*: LISTEN 3028/docker-proxy
```

5)、设置监听器,添加后端服务器。

【服务器列表】—>【弹性负载均衡 ELB】—>【(自定义 ELB 名称)】—>【监听器】—>【添加监听器】。如图-96、图-97、图-98、图-99 所示。

添加监听器

① 配置监听器 ————— ② 配置后端服务器组 —————

★ 名称	listener-8099
★ 前端协议/端口	TCP 8099 取值范围1~65535 四层监听请选择TCP、UDP；七层监听请选择HTTP、HTTPS。 选择HTTPS协议时，后端协议只能使用HTTP协议。

图-96



添加监听器

① 配置监听器 ————— ② 配置后端服务器组 ————— ③ 完成

后端服务器组	新创建 使用已有
★ 名称	server_group-harbor
★ 后端协议	TCP
★ 分配策略类型	加权轮询算法
会话保持	<input checked="" type="checkbox"/>

图-97

< | myelb | 运行中

基本信息 | **监听器** | 后端服务器组 | 监控 | 访问日志 | 标签

添加监听器	
listener-8099	编辑 删除
TCP/8099	
listener-8090	编辑 删除
TCP/8090	
listener-ha	编辑 删除
! 2 TCP/1080	

基本信息	后端服务器组	标签
名称	server_group-harbor	
分配策略类型	加权轮询算法	
会话保持	未开启	
添加	移除	

图-98

添加后端服务器



图-99

6)、访问 Harbor 界面 (<http://公网 IP 地址:8099>)。如图-100、图-101 所示。

用户名：admin

密码：Harbor12345

图-100

项目名称	访问级别	角色	镜像仓库数	创建时间
library	公开	项目管理员	0	2020/6/16 下午10:27

图-101

步骤三、部署一台 Docker 主机用于测试上传和下载。

1)、安装 docker 软件。

Docker Registry 交互默认使用的是 https, 然而此处搭建的私有仓库只提供 http 服务, 所以当与私有仓库交互时就会报上面的错误。为了解决这个问题需要在启动 docke 时增加启动参数为默认使用 http 访问。这个是在客户机的 docker 配置文件里添加的(即上传镜像到私有仓库里或从私有仓库下载镜像的客户机)。

```
[root@docker ~]# yum -y install docker-ce
[root@docker ~]# sed -i '/ExecStart/s/$/ --insecure-registry 192.168.1.67:8099/'
/usr/lib/systemd/system/docker.service
[root@docker ~]# systemctl daemon-reload && systemctl enable docker && systemctl start docker
```

2)、测试登录 Harbor 镜像仓库。

```
[root@docker ~]# docker login http://192.168.1.67:8099
Username: admin
Password: Harbor12345
Login Succeeded
```

#密码信息会存储在/root/.docker/config.json 中。

注意：

如果没有添加--insecure-registry 192.168.1.67:8099/, 在登录时会报以下错误。

Error response from daemon: Get https://192.168.1.67:8099/v2/: http: server gave HTTP response to HTTPS client

解决方法：添加参数，重启服务即可。

3) 向私有仓库中上传镜像

```
[root@docker ~]# scp root@192.168.1.252:/root/project3/DAY04/busybox.tar /root
#导入镜像
[root@docker ~]# docker load -i /root/busybox.tar
#查看系统中镜像
[root@docker ~]# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
busybox latest be5888e67be6 2 months ago 1.22MB
#更改标签
[root@docker ~]# docker tag busybox:latest 192.168.1.67:8099/library/busybox:latest
[root@docker ~]# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
192.168.1.67:8099/library/busybox latest be5888e67be6 2 months ago 1.22MB
```

```
busybox latest be5888e67be6 2 months ago 1.22MB
```

#向私有仓库推送镜像

```
[root@docker ~]# docker push 192.168.1.67:8099/library/busybox:latest
```

The push refers to repository [192.168.1.67:8099/library/busybox]

5b0d2d635df8: Pushed

latest: digest:

```
sha256:a2490cec4484ee6c1068ba3a05f89934010c85242f736280b35343483b2264b6 size: 527
```

4) 在界面中查看刚刚上传的镜像文件。

能够看到镜像仓库数为 1, 点击项目名称, 查看该项目下的镜像文件。如图-102、图-103 所示。

项目名称	访问级别	角色	镜像仓库数
library	公开	项目管理员	1

图-102

名称	标签数	下载数
library/busybox	1	0

图-103

5) 将系统中目前有的镜像文件删除, 从私有镜像仓库中下载镜像

```
[root@docker ~]# docker rmi 192.168.1.67:8099/library/busybox:latest
```

```
[root@docker ~]# docker images
```

REPOSITORY TAG IMAGE ID CREATED SIZE
busybox latest be5888e67be6 2 months ago 1.22MB

下载镜像

```
[root@docker ~]# docker pull 192.168.1.67:8099/library/busybox:latest  
[root@docker ~]# docker images  
REPOSITORY TAG IMAGE ID CREATED SIZE  
192.168.1.67:8099/library/busybox latest be5888e67be6 2 months ago 1.22MB  
busybox latest be5888e67be6 2 months ago 1.22MB
```

明确的看到该镜像的下载次数为 1。如图-104 所示。

The screenshot shows the Harbor interface. On the left, there is a navigation sidebar with '项目' (Project) selected, followed by '日志', '系统管理' (System Management), '用户管理' (User Management), '复制管理' (Copy Management), and '配置管理' (Configuration Management). The main area shows a project named 'library'. Under 'library', there is a '镜像仓库' (Image Repository) tab. A table lists an image entry: 'library/busybox' with a '标签数' (Tag Count) of 1 and a '下载数' (Download Count) of 1. The 'download count' is highlighted with a red rectangle.

图-104

点击左侧导航栏，可以看到日志信息。如图-105 所示。

The screenshot shows the Harbor interface with the '日志' (Log) option selected in the navigation sidebar. The main area displays a table titled '日志' (Logs) with two entries. The first entry is for 'admin' pulling the 'library/busybox' image with the 'latest' tag at '2020/6/17 上午12:53'. The second entry is for 'admin' pushing the same image at '2020/6/17 上午12:49'. The entire '日志' section is highlighted with a red rectangle.

图-105