UNIVERCITY WORK

Farmakis Athanasios

Dai17102

DEVELOPMENT AND MANAGEMENT OF COMPUTER NETWORKS

IMUNES INSTALLATION GUIDE (WITH BUILT-IN SNMP) KAI ZABBIX (NMS) IN UBUNTU SOFTWARE

I CREATED A VIRTUAL MACHINE THROUGH VIRTUALBOX WITH SOFTWARE:

Distributor ID: Ubuntu

Description: Ubuntu 20.04 LTS

Release: 20.04

Codename: focal

ENGINE SETTINGS:

System: Basic memory: 5094 MB

Screen: Graphics memory: 128 MB

Save:15 GB

THEN I INSTALLED IMUNES AS FOLLOWS:

Via terminal

SYSTEM REQUIREMENTS

```
# apt install openvswitch-switch docker.io xterm wireshark \
make imagemagick tk tcllib util-linux
```

!!! It must be executed as root otherwise it will take off

E: Could not open lock file / var / lib / dpkg / lock-frontend - open (13: Permission denied)

E: Unable to acquire the dpkg frontend lock (/ var / lib / dpkg / lock-frontend), are you root?

Otherwise use

```
# sudo apt install openvswitch-switch docker.io xterm wireshark \
make imagemagick tk tcllib util-linux
```

IMUNES INSTALLATION

Check out the latest IMUNES source via the public github repository:

```
# git clone https://github.com/imunes/imunes.git
```

Now we need to install IMUNES and complete the virtual file system with predefined and required data. To install imunes on the system run (as root !!!):

```
# cd immune
# make install
```

File system for virtual nodes

A template file system must be created for topologies to work. This is done by issuing the following command (as root !!!):

```
# imunes –p
```

Now the IMUNES GUI can be run simply by typing the imunes command in the terminal:

```
#imunes
```

To run experiments, run it as root.

INSTALLATION OF NMS ZABBIX

ZABBIX VERSION: 5.0 LTS

OS DISTRIBUTION: Ubuntu

OS VERSION: 20.04 (Focal)

DATABASE: MySQL

WEB SERVER: Apache

INSTALL THE ZABBIX REPOSITORY

```
#wget
https://repo.zabbix.com/zabbix/5.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_5.0-1+focal_all.deb

# dpkg -i zabbix-release_5.0-1 + focal_all.deb

# apt update
```

INSTALL ZABBIX SERVER, FRONTEND, AGENT

```
# apt install zabbix-server-mysql zabbix-frontend-php zabbix-apache-conf zabbix-agent
```

CREATING AN INITIAL DATABASE

Do the following on your database server.

```
# mysql -uroot -p

password (Enter your own password!)

mysql> create database zabbix character set utf8 collate utf8_bin;
```

```
mysql> create user zabbix @ localhost identified by 'password';(Enter your own password!)

mysql> grant all privileges on zabbix. * to zabbix @ localhost ;

mysql> quit;
```

On the Zabbix server, the host enters initial formats and data. You will be prompted to enter your new password.

```
# zcat /usr/share/doc/zabbix-server-mysql*/create.sql.gz | mysql -uzabbix -p zabbix
```

CONFIGURE THE DATABASE FOR THE ZABBIX SERVER

Edit the /etc/zabbix/zabbix_server.conf file

```
DBPassword =password (Your own password!)
```

PHP CONFIGURATION FOR ZABBIX FRONTEND

Edit the /etc/zabbix/apache.conf file, remove from the comments, and set the correct time zone for you.(In all paragraphs!)

```
# php_value date.timezone Europe / Athens
```

START ZABBIX SERVER AND AGENT PROCESSES

Start the Zabbix server and agent processes and start the system.

```
# systemctl restart zabbix-server zabbix-agent apache2

# systemctl enable zabbix-server zabbix-agent apache2
```

ZABBIX INTERFACE CONFIGURATION

Log in to your newly installed Zabbix desktop: http: // server_ip_or_name / zabbix

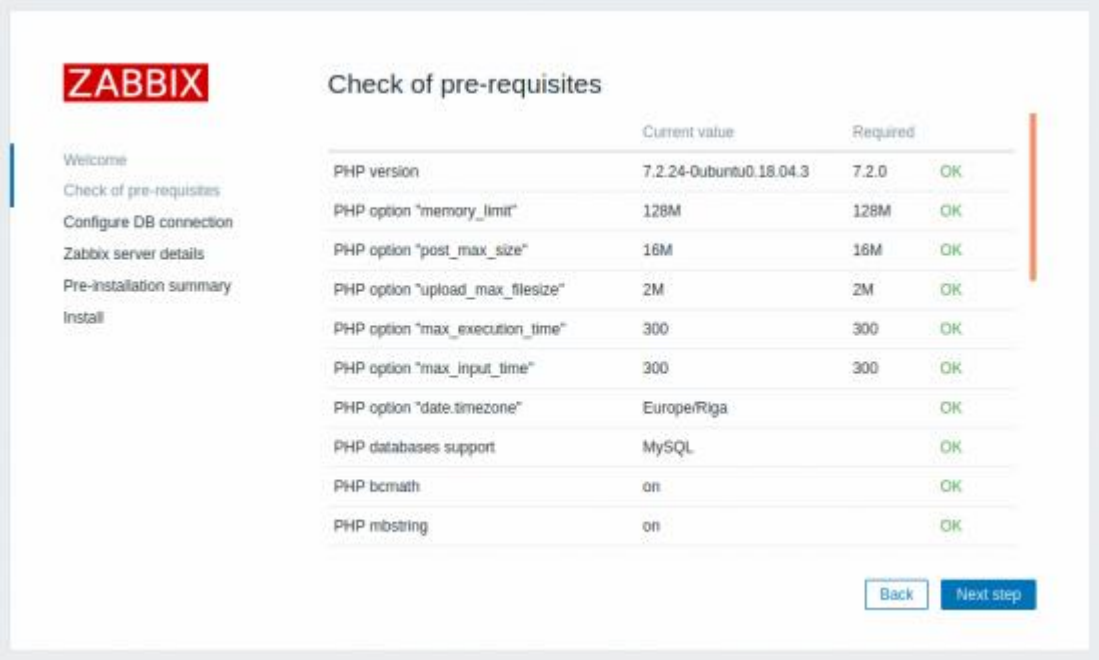
Step 1

In your browser, open the Zabbix URL: http: // <server_ip_or_name> / zabbix

You should see the first screen of the frontend installation wizard.

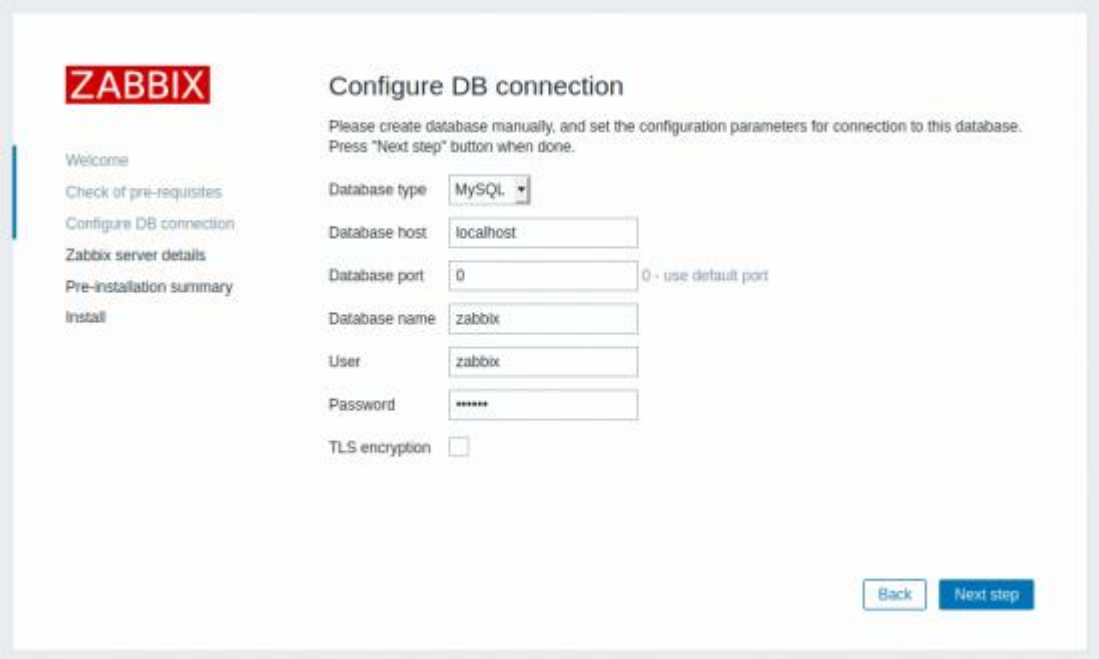
Step 2

Make sure all software requirements are met.



Step 3

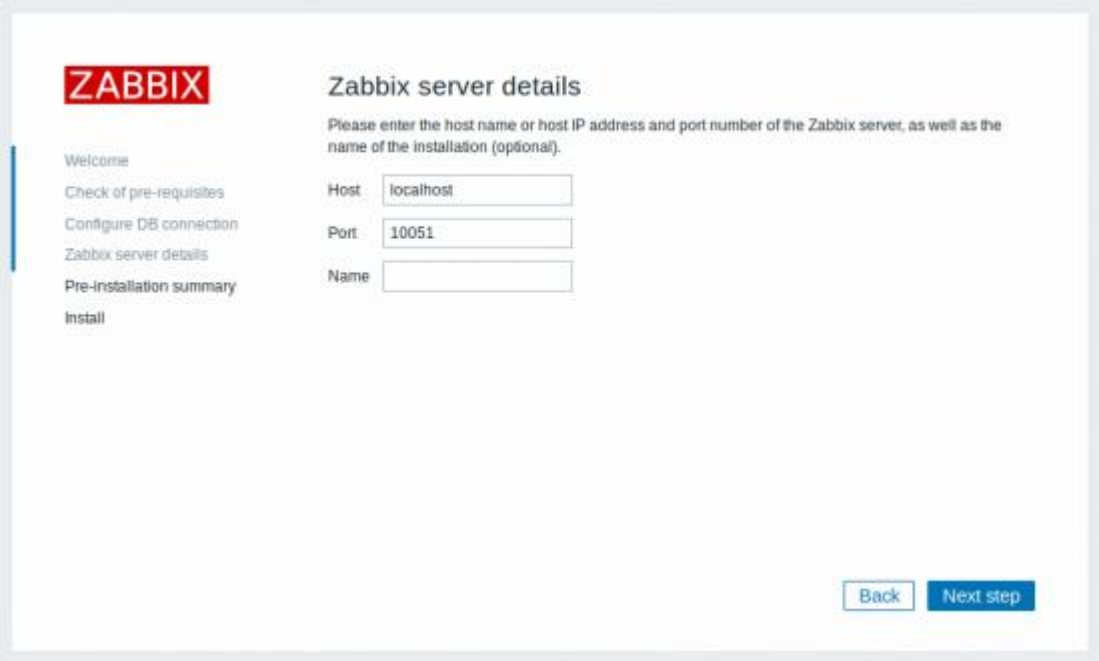
Enter login details for the database. The Zabbix database must already be created.



If the TLS encryption option is selected, the form has five additional fields for configuring the TLS database connection (MySQL or PostgreSQL only).

Step 4

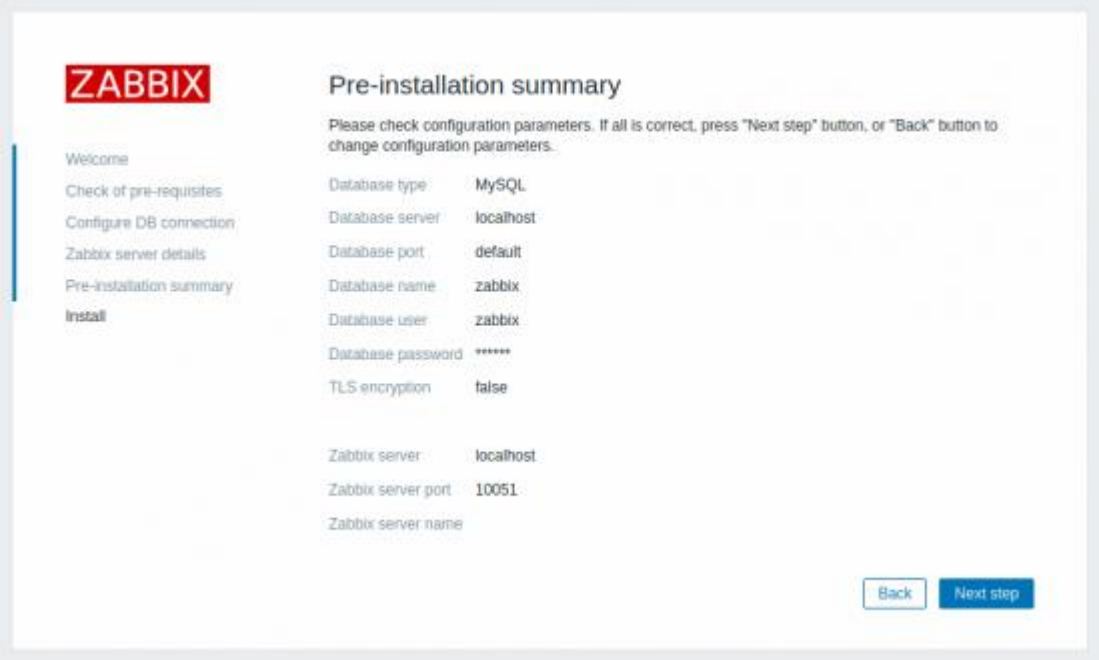
Enter Zabbix server details.



Entering a name for the Zabbix server is optional, however, if submitted, it will appear in the menu bar and page titles.

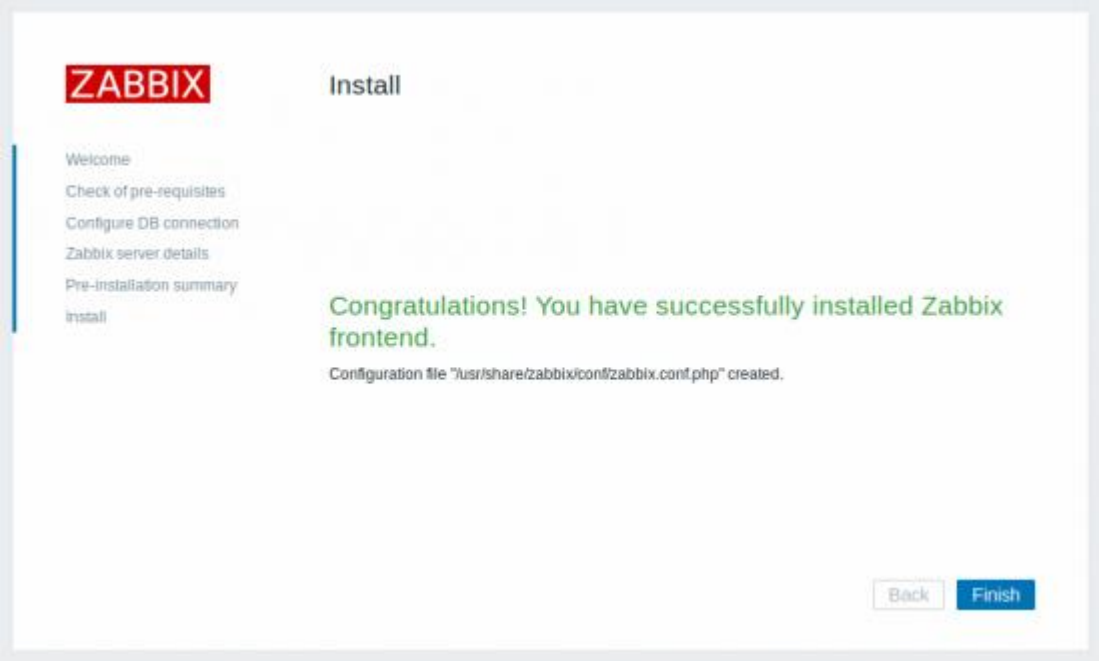
Step 5

Check a summary of the settings.



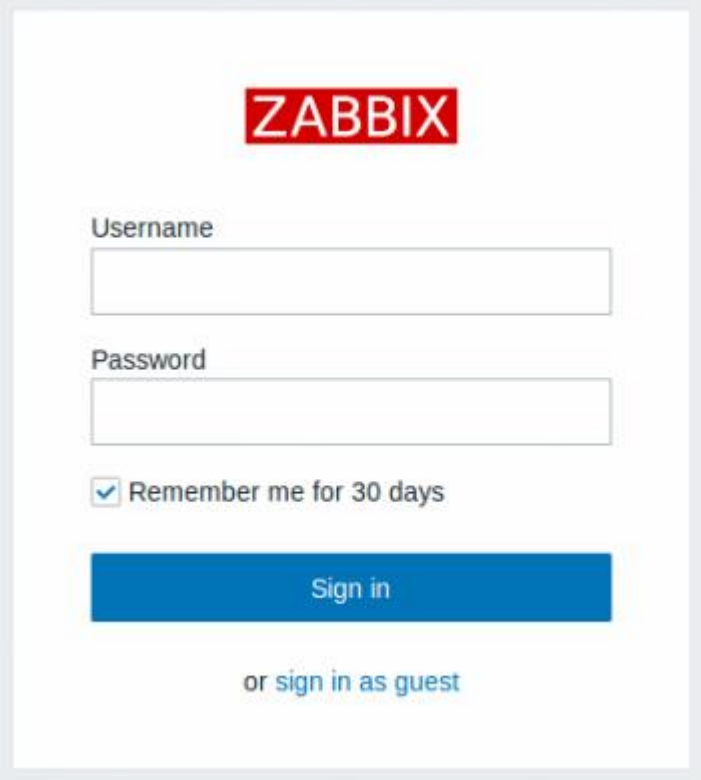
Step 6

Complete the installation.



Step 8

The Zabbix frontend is ready! The default username is Admin, zabbix password.



1 INSTALLATION OF ZABBIX DEMONS

1 DOWNLOAD THE SOURCE FILE

Go to the Zabbix download pagehttps://www.zabbix.com/download_sourcesand download the source file. After downloading, extract the sources by performing:

```
$ tar -zxvf zabbix-5.0.0.tar.gz
```

2 CREATE A USER ACCOUNT

All Zabbix daemon processes require a non-privileged user. If a Zabbix demon starts from an unauthorized account, it will run as this user.

However, if a demon starts from a "root" account, it will change to a "zabbix" user account, which must exist.

We have already created!

3 CREATE A ZABBIX DATABASE

For Zabbix servers and proxies, as well as for the Zabbix interface, a database is required. Zabbix agent execution is not required.

SQL scripts are provided to create a database schema and import the data set. The Zabbix Proxy Database only needs the schema, while the Zabbix Server Database also requires the data set at the top of the schema.

Once you have created a Zabbix database, proceed to the following steps to compile Zabbix. (We have created!)

4 CONFIGURE THE SOURCES

When configuring resources for a Zabbix server or proxy server, you must specify the type of database to be used. Only one type of database can be compiled with one server or proxy process at a time.

To see all supported configuration options, run the exported Zabbix source directory:

```
# ./configure –help
```

To configure resources for a Zabbix server and agent, you can do something like:

```
# ./configure --enable-server --enable-agent --with-mysql --
enable-ipv6 --with-net-snmp --with-libcurl --with-libxml2
```

To configure sources for a Zabbix server (with PostgreSQL, etc.), you can run:

```
# ./configure --enable-server --with-postgresql --with-net-
snmp
```

To configure the sources for a Zabbix proxy server (with SQLite, etc.), you can run:

```
# ./configure --prefix = / usr --enable-proxy --with-net-snmp
--with-sqlite3 --with-ssh2
```

To configure resources for a Zabbix agent, you can run:

```
# ./configure --enable-agent
```

or, Zabbix Agent 2:

```
# ./configure --enable-agent2
```

Notes on collection options:

The zabbix_get and zabbix_sender command line utilities are written if the --able-agent option is used.

We install the libcurl and libxml2 library:

```
# sudo apt-get update –y

#  sudo apt-get install -y libcurl-dev

#  sudo apt-get update –y

#  sudo apt-get install -y libxml2-dev
```

--with-libcurl and --with-libxml2 configuration options are required for virtual machine monitoring. - with-libcurl is also required for SMTP and web.page authentication. * Zabbix agent data. Note that cURL 7.20.0 or later is required with the --with-libcurl configuration option.

Zabbix is always compiled with the PCRE library (since version 3.4.0):

```
#sudo apt-get update # sudo apt-get install libpcre3
libpcre3-dev
```

Installation is mandatory. --with-libpcre = [DIR] only allows pointing to a specific database installation directory, instead of searching for a number of common locations for libpcre files.

You can use the --enable-static flag to statically link libraries. If you plan to distribute compiled binaries to different servers, you must use this flag to make these binaries

work without the need for libraries. Note that -enable-static does not work on Solaris.

It is not recommended to use - enable - static option when creating a server. To create the server statically you must have a static version of each external library required. There is no strict control for this in the configuration scenario.

Add an optional path to the MySQL configuration file --with-mysql = / <path_to_the_file> / mysql_config to select the desired MySQL client library when you need to use the one that is not in the default location. Useful when there are several versions of MySQL installed or MariaDB installed in parallel with MySQL on the same system.

We recommend that you use the MariaDB Connector / C library to create a proxy / proxy regardless of whether you are using a MySQL or MariaDB database server.

Use the --with-oracle flag to determine the location of the OCI API.

So we continue with ./configure

```
#. / configure
```

If ./configure fails due to missing libraries or some other circumstance, see the config.log file for more details about the error. For example, if libssl is missing, the instant error message may be misleading:

checking for main in -lmysqlclient ... no

configure: error: Not found mysqlclient library

While config.log has a more detailed description:

/ usr / bin / ld: cannot find -lssl

/ usr / bin / ld: cannot find –lcrypto

There must also be postgresql installed!

Unable to locate package postgresql-server-dev-9.1 on Ubuntu 20.04 x64

```
Solutionsudo apt install postgresql-server-dev-10
```

The same goes for the other libraries mentioned above, in addition if needed.

sudo apt-get update

sudo apt-get install libxslt-dev libxml2-dev libpam-dev libedit-dev

sudo apt-get install -t squeeze-backports postgresql-server-dev-9.1

Install the Go tools

If you are upgrading from an older version of Go, you must first remove the existing version.

```
#tar -C / usr / local -xzf go $ VERSION. $ OS- $ ARCH.tar.gz

#export PATH = $ PATH: / usr / local / go / bin
```

Note: Changes made to a profile file may not take effect until the next time you log on to your computer. To apply the changes immediately, simply run the shell commands directly or run them from the profile using a command such as the source $ HOME / .profile.

5 CREATE AND INSTALL EVERYTHING

```
$ sudo make dbschema

$ sudo make install
```

This step must be performed as a user with sufficient privileges (usually "root" or using sudo).

Running make install will install by default the binary daemons (zabbix_server, zabbix_agentd, zabbix_proxy) in / usr / local / sbin and the binary client (zabbix_get, zabbix_sender) in / usr / local / bin.

To specify a different location from / usr / local, use a --prefix key in the previous source configuration step, for example --prefix = / home / zabbix. In this case, the demon binaries will be installed in <prefix> / sbin, while the utilities in <prefix> / bin. Human pages will be installed in <prefix> / share.

6 CHECK AND EDIT CONFIGURATION FILES

edit the Zabbix factor configuration file /usr/local/etc/zabbix_agentd.conf

You must configure this file for each host with zabbix_agentd installed.

You must specify the IP address of the Zabbix server in the file. Connections from other servers will be rejected.

edit the Zabbix server configuration file /usr/local/etc/zabbix_server.conf

You must specify the database name, user, and password (if used).

The other parameters will suit your defaults if you have a small installation (up to ten monitored servers). However, you will need to change the default settings if you want to maximize the performance of your Zabbix server (or proxy server). See Performance Tuning for more details.

if you have a Zabbix proxy server installed, edit the proxy /usr/local/etc/zabbix_proxy.conf proxy configuration file

You must specify the server IP address and proxy server host name (must be known on the server), as well as the database name, user, and password (if you use any).

The full path to the database file must be specified with SQLite. No user and DB password required.

7 START THE DEMONS

Run zabbix_server on the server side.

```
shell> zabbix_server
```

Make sure your system allows 36 MB (or a little more) of shared memory, otherwise the server may not start and you will see "Cannot allocate <cache type> memory." in the server log.

Run zabbix_agentd on all monitored machines.

```
shell> zabbix_agentd
```

Make sure your system allows 2MB of shared memory, otherwise the agent may not start and you will see "Cannot allocate collector memory". in the agent log.

If you have the Zabbix proxy server installed, run zabbix_proxy.

```
shell> zabbix_proxy
```

2 INSTALL THE ZABBIX WEB INTERFACE

COPY PHP FILES

Zabbix frontend is written in PHP, so running a PHP-supported web server is required to run it. The installation is done simply by copying the PHP files from the ui directory to the HTML document directory of the web server.

Common HTML document directory locations for Apache web servers include:

- / usr / local / apache2 / htdocs (default directory when installing Apache from source)
- / srv / www / htdocs (OpenSUSE, SLES)
- / var / www / html (Debian, Ubuntu, Fedora, RHEL, CentOS)

It is recommended to use a subdirectory instead of the HTML root. To create a subdirectory and copy frontend Zabbix files to it, run the following commands, replacing the actual directory:

```
mkdir <htdocs> / zabbix

cd ui

cp -a. <htdocs> / zabbix
```

If you are installing from the Zabbix Git repository and intend to use any language other than English, you must create translation files. To do this, run:

```
locale / make_mo.sh
```

Msgfmt utility required from the gettext package!

3 JAVA GATEWAY INSTALLATION

Java gateway only needs to be installed if you want to monitor JMX applications. The Java gateway is lightweight and does not require a database.

To install from sources, first download and extract the source file.

To compile the Java gateway, run the ./configure script with the option --enable-java. It is recommended that you specify the --prefix option to request an installation path other than the default / usr / local, because installing the Java gateway will create an entire directory tree, not just one executable.

```
$ ./configure --enable-java --prefix = $ PREFIX
```

To compile and package the Java gateway into a JAR file, run make. Note that for this step you will need executable javac and jar on your route.

```
$ make
```

You now have a zabbix-java-gateway- $ VERSION.jar file in src / zabbix_java / bin. If you are comfortable running the Java portal from src / zabbix_java in the distribution directory, then you can proceed with instructions for configuring and running the Java portal. Otherwise, make sure you have enough privileges and run the installation.

```
$ make install
```

INSTALL SNMP IN IMUNES

STEP: VERIFY AND INSTALL NET-SNMP WITH RPM PACKAGES

```
#sudo apt update

#sudo apt install snapd

#sudo snap install net-snmp
```

QUAGGA INSTALLATION AT HOME

Install the Quagga Routing Demon:

```
#sudo apt-get install quagga && sudo mkdir -p / var / log / quagga && sudo chown quagga: quagga / var / log / quagga
```

Enable IPv4 and IPv6 Unicast Promotion:

```
#echo "net.ipv4.conf.all.forwarding = 1" | sudo tee -a /etc/sysctl.conf
```

```
#echo "net.ipv4.conf.default.forwarding = 1" | sudo tee -a /etc/sysctl.conf

sed 's / # net.ipv6.conf.all.forwarding = 1 / net.ipv6.conf.all.forwarding = 1 / g' /etc/sysctl.conf | sudo tee /etc/sysctl.conf

#echo "net.ipv6.conf.default.forwarding = 1" | sudo tee -a /etc/sysctl.conf

#sudo sysctl -p
```

Enable IPv4 multi-distribution forwarding:

```
#echo "net.ipv4.conf.all.mc_forwarding = 1" | sudo tee -a /etc/sysctl.conf

#echo "net.ipv4.conf.default.mc_forwarding = 1" | sudo tee -a /etc/sysctl.conf

#sudo sysctl -p
```

Create configuration files:

```
#sudo nano /etc/quagga/bgpd.conf

#sudo nano /etc/quagga/isisd.conf

#sudo nano /etc/quagga/ospf6d.conf

#sudo nano /etc/quagga/ospfd.conf

#sudo nano /etc/quagga/pimd.conf

#sudo nano /etc/quagga/ripd.conf

#sudo nano /etc/quagga/ripngd.conf

#sudo nano /etc/quagga/vtysh.conf

#sudo nano /etc/quagga/zebra.conf
```

STEP 2 AND 3

https://download.savannah.gnu.org/releases/quagga/

Download the latest version of the QUAGGA package in IMunes, and then run:

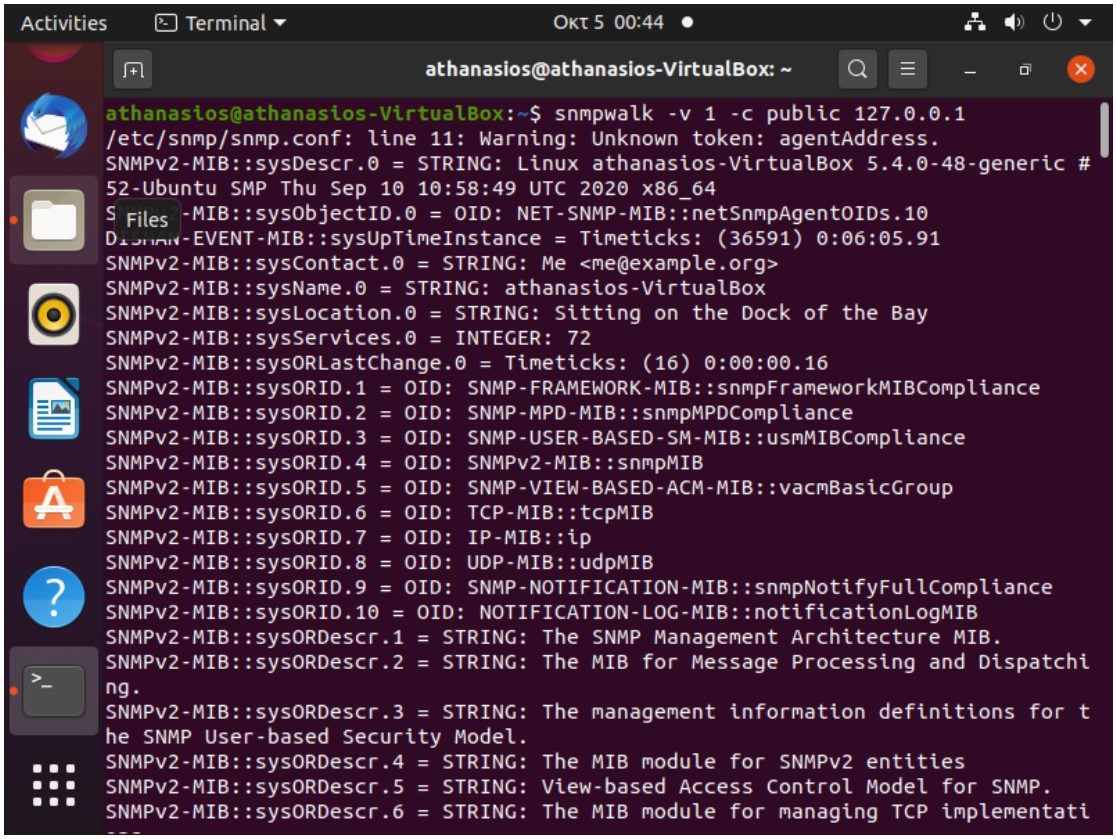
```
$ tar xvzf quagga-0.99.19.tar.gz
```

Enable snmp in quagga to receive all snmp signals from Imunes:

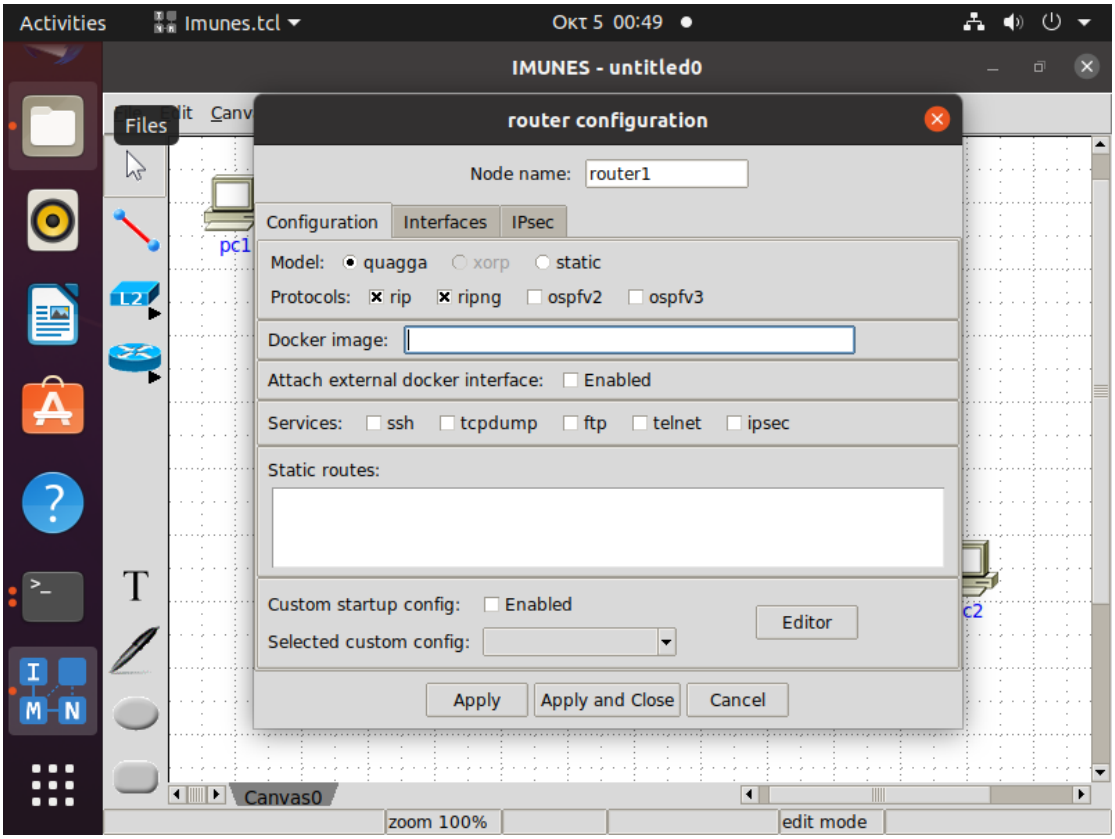
```
#cd quagga-0.99.19

# ./configure --enable-snmp

#sudo make

#sudo make install
```

We test the function of snmp:

```
#snmpwalk -v 1 -c public 127.0.0.1
```



Then in the networks that we will create in imunes we have to add the quagga option to the routers!



PROBLEMS IN ZABVICH GUI

Προβλήματα κατά την χρήση του Zabbix

Κατά την χρήση του Zabbix και του Local server που συνδεόμαστε για το index.php και το frond end του zabbix μου βγάζει error ο apache2 server και δεν μπορώ να τον ξεκινήσω ώστε να μπορέσω να μπω στο http://127.0.0.1/zabbix .

```
Job for apache2.service failed. See "systemctl status apache2.service" and
"journalctl -xe" for details.
```

Βρήκα ότι το πρόβλημα προκείπτει από λάθος του αρχείου `/etc/apache2/apache2.conf`

Όπου πρέπει να προσθέσω το

Serverhost localhost το οποίο δεν δούλεψε.

Επίσης έκανα κατάργηση εγκατάστασης και επεγκατάσταση το apache2 server αλλά πάλι μου βγάζει FAILURE .

Την προηγούμενη φορά είχε γίνει αυτό το error[Z3001] connection to database 'zabbix_db' failed: [1045] Access denied for user 'zabbix'@'localhost' (using password: YES)

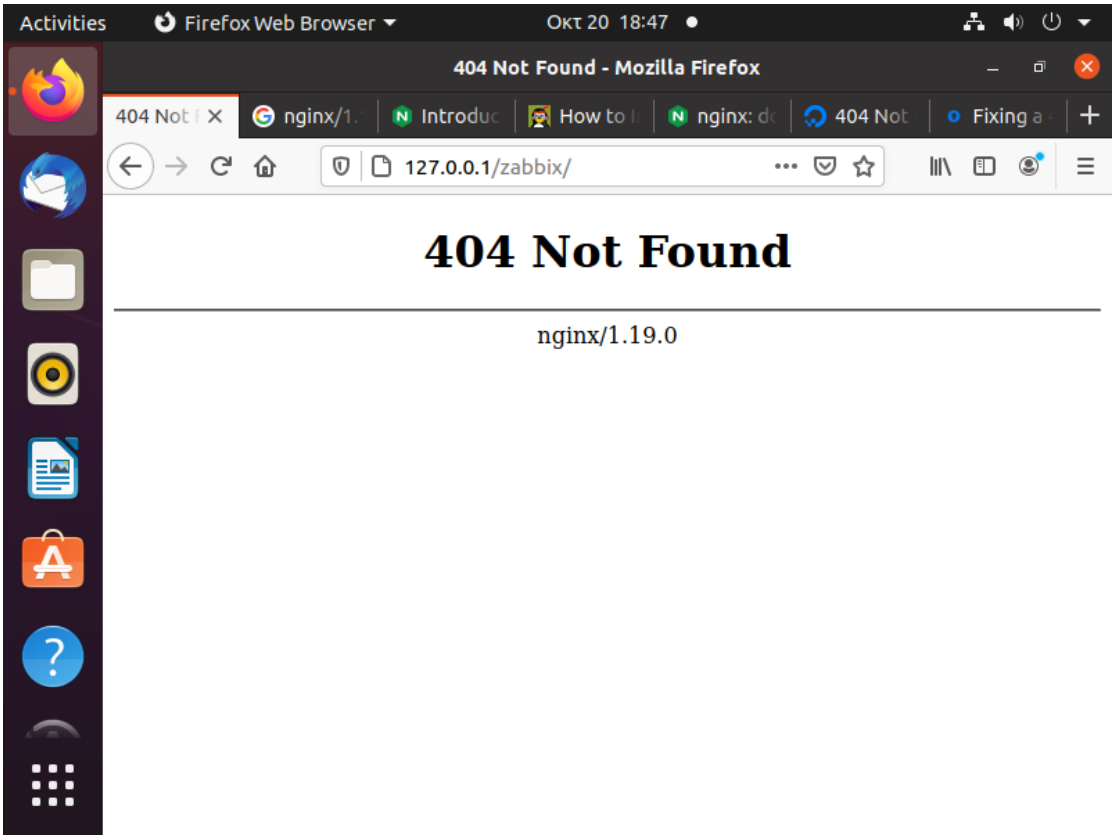
και κατάφερα να το διορθώσω πειράζοντας τα αρχεία

/etc/zabbix/zannix_server.conf και

zabbix.conf.php ώστε να έχουν ίδιες πληροφορίες , συγκεκριμμένα ίδιο host.

Το τελικό πρόβλημα είναι ότι πρέπει διαρκώς να κάνω απεγκατάσταση και επεγκατάσταση το zabbix ολόκληρο ώστε να καταφέρω να μπω στην τοπική ιστοσελίδα και να κάνω tracking.

Having this problem, after fixing the apache2.conf and zabbix_server.conf files so that they do not have differences for the correct operation, I got the following error.



Which is solved with the following steps.

Nginx.org maintains a repository for Ubuntu. We can use this repository to install the latest version of Nginx. First, create a repository source file for Nginx with the following command. Nano is a command line word processor.

```
#sudo nano /etc/apt/sources.list.d/nginx.list
```

Add the following two lines to the file.

```
#deb [arch = amd64] http://nginx.org/packages/mainline/ubuntu/ focal nginx

# deb-src http://nginx.org/packages/mainline/ubuntu/ focal nginx
```

To save the file to the Nano word processor, press CTRL + O, and then press Enter to confirm. Press CTRL + X to exit. To

verify the integrity of the packets received from this repository, we need to enter the public key Nginx using the following commands.

```
#wget http://nginx.org/keys/nginx_signing.key
```

```
#sudo apt-key add nginx_signing.key
```

Once the repository has been added to the Ubuntu 18.04 system, run the following command to update the repository information.

```
#sudo apt update
```

If you have Nginx installed from your default Ubuntu software repository, you must remove it.

```
#sudo apt remove nginx nginx-common nginx-full nginx-core
```

You may also want to back up the main Nginx /etc/nginx/nginx.conf configuration file because it will be replaced with a new nginx.conf file when we later install the latest version of Nginx.

```
#sudo cp /etc/nginx/nginx.conf /etc/nginx/nginx.conf.bak
```

Existing server block files (also known as virtual host files) will be intact. Now run the following command to install Nginx from the nginx.org repository.

```
#sudo apt install nginx
```

After installing Nginx, try the Nginx configuration.

```
#sudo nginx –t
```

If the test is successful, start Nginx.

```
#sudo systemctl start nginx
```

Enable auto-start at startup.

```
#sudo systemctl enable nginx
```

APPLICATION OF EXAMPLE IN MICROCIRCUIT AND SNMP CONTROL

Go to the Zabbix Server Control Panel and add the Network Switch as the host.

Open your browser and enter the IP address of your web server plus / zabbix.

In our example, the following URL was entered in the browser:

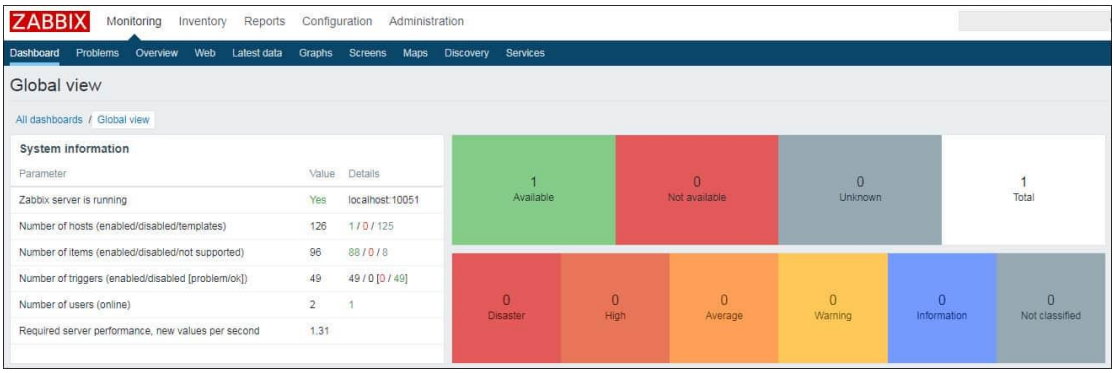
• http://127.0.0.1/zabbix

On the login screen, use the default username and password.

Default username: user

Default password: zabbix

Upon successful connection, it will be sent to the Zabbix control panel.



On the host configuration screen, you must enter the following information:

• Host name - Enter a host name to identify the switch

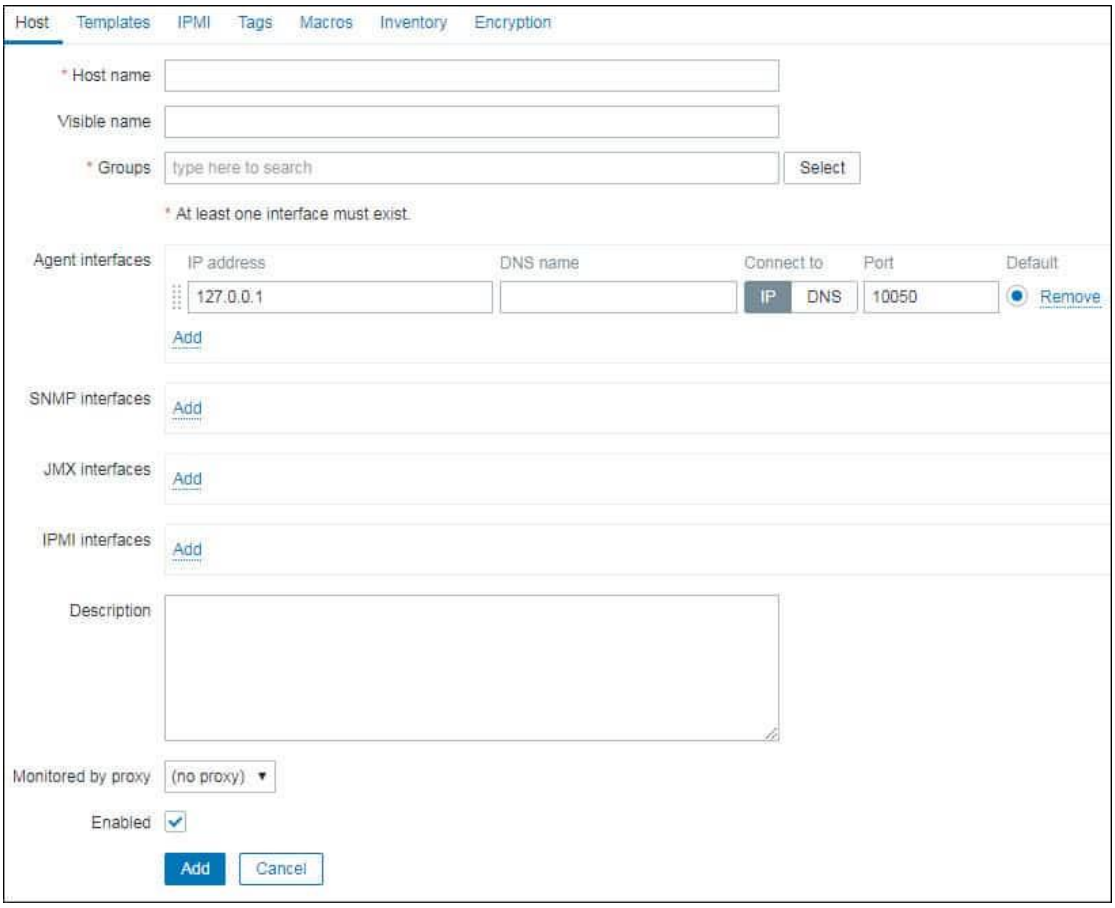
Visible host name - Repeat the host name.

• New group - Enter a name to identify a group of similar devices.

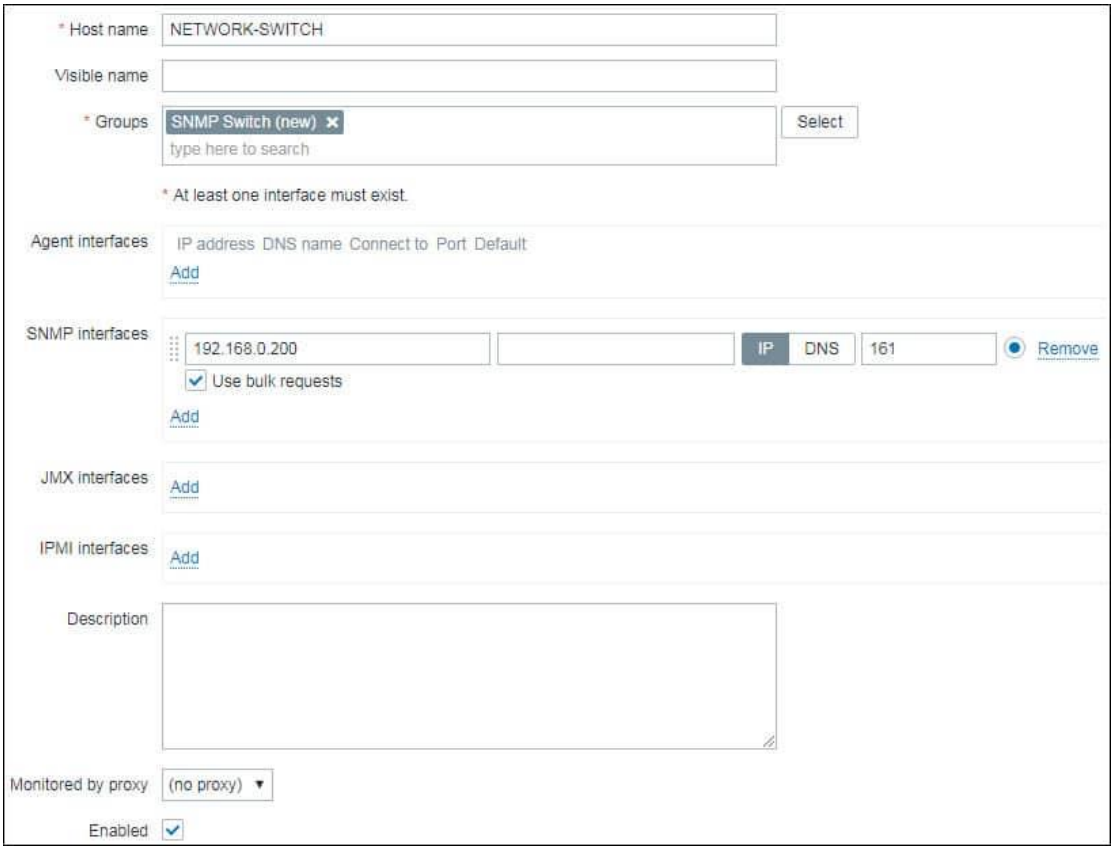
• Interface Agent - Click the Remove button.

• SNMP Interface - Click the Add button and enter the IP Switch IP address.

Here is the original image, before we configure it.



Here is the new image with our configuration.



Next, we need to configure the SNMP community that Zabbix will use to connect to the Network Switch.

Go to the Macros tab at the top of the screen.

Create a macro named: {$ SNMP_COMMUNITY}

The τιμή $ SNMP_COMMUNITY μα macro value should be the SNMP Switch Network community.

In our example, the value {$ SNMP_COMMUNITY} is GokuBlack

Next, we need to associate the host with a specific network monitoring standard.

By default, Zabbix has a wide variety of tracking templates.

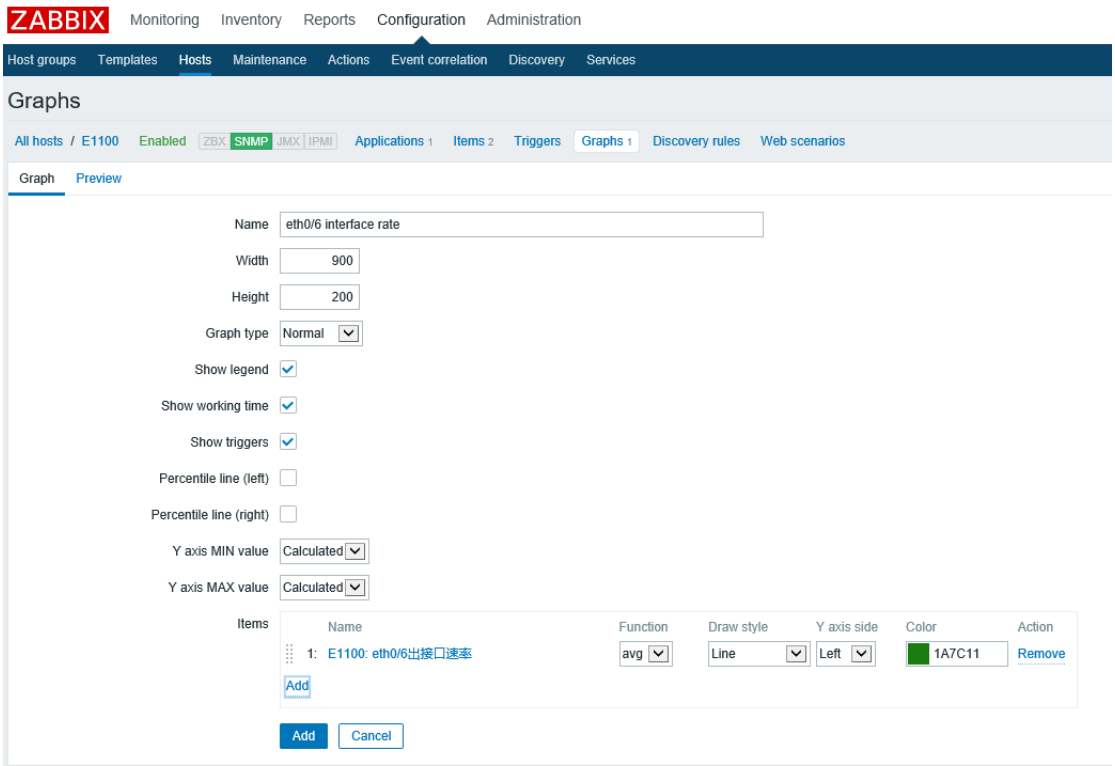
Go to the Templates tab at the top of the screen.

Locate and select the template named: Template Net Network Generic Device SNMPv2

Then we create the graph.

Configuration> Hosts

Create graph.



RESULTS


ZABBIX
Monitoring
Inventory
Reports
Configuration
Administration
Share
Dashboard
Problems
Overview
Web
Latest data
Triggers
Graphs
Screens
Maps
Discovery
Services
E1100: eth0/6 Egress interface rate
Graph
Filter
Zoom: 5m 15m 30m 1h 2h 3h All
2019-04-25 13:38:59 - 2019-04-25 14:38:59 (now!)
«« 1h 5m | 5m 1h »»
1h fixed
E1100: eth0/6出接口速率 (1h)
350 K
300 K
250 K
200 K
150 K
100 K
50 K
04/25 01:23 PM
02:00 PM
04/25 02:23 PM
last
min
avg
max
eth0/6 Egress interface rate
[all]
135.65 K
62.33 K
111.44 K
307.46 K