ZABBIX

1. What is Zabbix

Zabbix was created by Alexei Vladishev, and currently is actively developed and supported by Zabbix SIA.

Zabbix is an enterprise-class open source distributed monitoring solution.

Zabbix is software that monitors numerous parameters of a network and the health and integrity of servers. Zabbix uses a flexible notification mechanism that allows users to configure e-mail based alerts for virtually any event. This allows a fast reaction to server problems. Zabbix offers excellent reporting and data visualisation features based on the stored data. This makes Zabbix ideal for capacity planning.

Zabbix supports both polling and trapping. All Zabbix reports and statistics, as well as configuration parameters, are accessed through a web-based frontend. A web-based frontend ensures that the status of your network and the health of your servers can be assessed from any location. Properly configured, Zabbix can play an important role in monitoring IT infrastructure. This is equally true for small organisations with a few servers and for large companies with a multitude of servers.

Zabbix is free of cost. Zabbix is written and distributed under the GPL General Public License version 2. It means that its source code is freely distributed and available for the general public.

2. Zabbix features

Overview

Zabbix is a highly integrated network monitoring solution, offering a multiplicity of features in a single package.

Data gathering

- availability and performance checks
- support for SNMP (both trapping and polling), IPMI, JMX, VMware monitoring
- custom checks
- gathering desired data at custom intervals
- performed by server/proxy and by agents

Flexible threshold definitions

- you can define very flexible problem thresholds, called triggers, referencing values from the backend database

Highly configurable alerting

- sending notifications can be customized for the escalation schedule, recipient, media type
- notifications can be made meaningful and helpful using macro variables
- automatic actions include remote commands

Real-time graphing

- monitored items are immediately graphed using the built-in graphing functionality

Web monitoring capabilities

- Zabbix can follow a path of simulated mouse clicks on a web site and check for functionality and response time

Extensive visualisation options

- ability to create custom graphs that can combine multiple items into a single view
- network maps
- custom screens and slide shows for a dashboard-style overview
- reports
- high-level (business) view of monitored resources

Historical data storage

- data stored in a database
- configurable history
- built-in housekeeping procedure

Easy configuration

- add monitored devices as hosts
- hosts are picked up for monitoring, once in the database
- apply templates to monitored devices

Use of templates

- grouping checks in templates
- templates can inherit other templates

Network discovery

- automatic discovery of network devices
- agent auto registration
- discovery of file systems, network interfaces and SNMP OIDs

Fast web interface

- a web-based frontend in PHP
- accessible from anywhere
- you can click your way through
- audit log

Zabbix API

- Zabbix API provides programmable interface to Zabbix for mass manipulations, 3rd party software integration and other purposes.

Permissions system

- secure user authentication
- certain users can be limited to certain views

Full featured and easily extensible agent

- deployed on monitoring targets
- can be deployed on both Linux and Windows

Binary daemons

- written in C, for performance and small memory footprint
- easily portable

Ready for complex environments

- remote monitoring made easy by using a Zabbix proxy

3. Zabbix overview

Architecture

Zabbix consists of several major software components, the responsibilities of which are outlined below.

Server

Zabbix server is the central component to which agents report availability and integrity information and statistics. The server is the central repository in which all configuration, statistical and operational data are stored.

Database storage

All configuration information as well as the data gathered by Zabbix is stored in a database.

Web interface

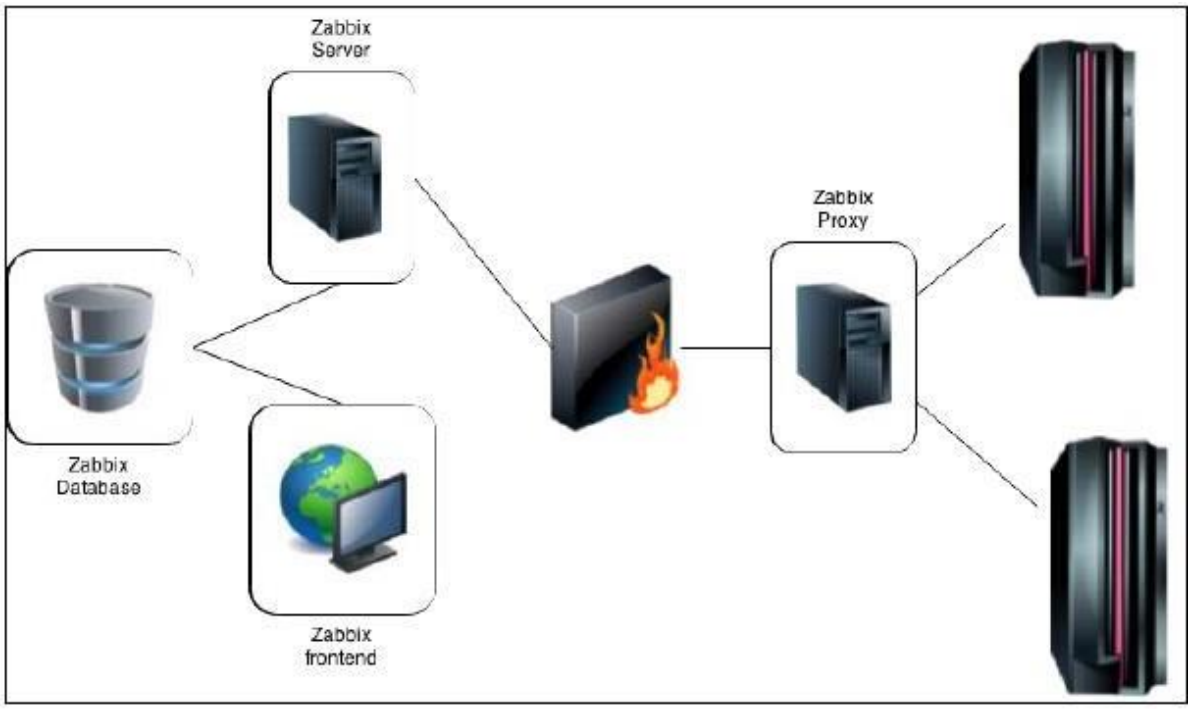
For an easy access to Zabbix from anywhere and from any platform, the web-based interface is provided. The interface is part of Zabbix server, and usually (but not necessarily) runs on the same physical machine as the one running the server.

Proxy

Zabbix proxy can collect performance and availability data on behalf of Zabbix server. A proxy is an optional part of Zabbix deployment; however, it may be very beneficial to distribute the load of a single Zabbix server.

Agent

Zabbix agents are deployed on monitoring targets to actively monitor local resources and applications and report the gathered data to Zabbix server.



Data flow

In addition it is important to take a step back and have a look at the overall data flow within Zabbix. In order to create an item that gathers data you must first create a host. Moving to the other end of the Zabbix spectrum you must first have an item to create a trigger. You must have a trigger to create an action. Thus if you want to receive an alert that your CPU load it too high on *Server X* you must first create a host entry for *Server X* followed by an item for monitoring its CPU, then a trigger which activates if the CPU is too high, followed by an action which sends you an email. While that may seem like a lot of steps, with the use of templating it really isn't. However, due to this design it is possible to create a very flexible setup.

How it works...

All our clients will be configured to communicate through the Zabbix proxy. This way we don't have to open many ports in our firewall to let all agents pass their data to the Zabbix server. The Zabbix proxy runs with its own database. This can be the same or another database as the Zabbix server. Because we keep all data in a database on our Zabbix proxy, the data stays in the proxy for some time making it also perfect for installations where we have to send data over unstable networks.

We will configure the proxy to send all data through the firewall to our Zabbix server. The Zabbix server itself can be similar to what we have seen in *Zabbix Configuration* a server with all components installed on it or it can be split up in three parts with the frontend, database, and Zabbix server installed on different hardware.

4. Install zabbix 3.4 in ubuntu16.4

Follow the below url for install zabbix.

https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-zabbix-to-securely-monitor-remote-servers-on-ubuntu-16-04

Step 1 — Installing the Zabbix Server

First, we need to install the Zabbix Server on the server where we installled MySQL, Apache, and PHP.We'll refer to this machine as the "Zabbix server" in this tutorial.

Before we install Zabbix, we need to install a few PHP modules that Zabbix needs. First, update your system's list of available packages:

```
$ sudo apt-get update
```

Then install the PHP modules Zabbix needs:

```
$ sudo apt-get install php7.0-xml php7.0-bcmath php7.0-mbstring
```

```
If we see below error
E: Unable to locate package php7.0-xml
Run below commands
-------------------------------------
sudo apt-add-repository ppa:ondrej/php
sudo apt-get update
sudo apt-get install php7.0
sudo apt-get install php7.0-mysqli
-------------------------------------
```

Now we can install Zabbix.

Zabbix is available in Ubuntu's package manager, but it's outdated, so we'll use the official Zabbix repository to install the latest stable version. Download and install the repository configuration package:

```
$ wget http://repo.zabbix.com/zabbix/3.2/ubuntu/pool/main/z/zabbix-release/zabbix-release_3.2-1+xenial_all.deb
```

```
$ sudo dpkg -i zabbix-release_3.2-1+xenial_all.deb
```

You will see the following output:

```
Output
Selecting previously unselected package zabbix-release.
(Reading database ... 55276 files and directories currently installed.)
Preparing to unpack zabbix-release_3.2-1+xenial_all.deb ...
Unpacking zabbix-release (3.2-1+xenial) ...
Setting up zabbix-release (3.2-1+xenial) ...
```

Update the package index so the new repository is included:

```
$ sudo apt-get update
```

Then install the Zabbix server and web frontend with MySQL database support:

```
$ sudo apt-get install zabbix-server-mysql zabbix-frontend-php
```

Let's also install the Zabbix agent, which will let us collect data about the Zabbix server status itself.

```
$ sudo apt-get install zabbix-agent
```

Before we can use Zabbix, we have to set up a database to hold the data that the Zabbix server will collect from its agents.

Step 2 — Configuring the MySQL Database For Zabbix

We need to create a new MySQL database and populate it with some basic information in order to make it suitable for Zabbix. We'll also create a specific user for this database so Zabbix isn't logging into MySQL with the *root* account.

Log into MySQL as the root user using the root password that you set up during the MySQL server installation:

```
If aws
$ sudo service mysql start
$ sudo mysql_secure_installation
```

ubuntu@ip-172-31-21-102:~$ **sudo mysql_secure_installation**

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

You already have a root password set, so you can safely answer 'n'.

Change the root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..

... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone to log into MariaDB without having to have a user account created for them. This is intended only for testing, and to make the installation go a bit smoother. You should remove them before moving into a production environment.

Remove anonymous users? [Y/n] n
... skipping.

Normally, root should only be allowed to connect from 'localhost'. This ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] n
... skipping.

By default, MariaDB comes with a database named 'test' that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment.

Remove test database and access to it? [Y/n] n
... skipping.

Reloading the privilege tables will ensure that all changes made so far will take effect immediately.

Reload privilege tables now? [Y/n] y
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB installation should now be secure.

Thanks for using MariaDB!
ubuntu@ip-172-31-21-102:~$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 56
Server version: 10.0.38-MariaDB-0ubuntu0.16.04.1 Ubuntu 16.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>

```
$ sudo mysql –u root –p
```

Create the Zabbix database with UTF-8 character support:

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

Then create a user that the Zabbix server will use, give it access to the new database, and set the password for the user:

```
Mysql > grant all privileges on zabbix.* to zabbix@localhost
identified by 'your_password';
```

Then apply these new permissions:

```
Mysql > flush privileges;
```

That takes care of the user and the database. Exit out of the database console.

```
Mysql > quit;
```

Next we have to import the initial schema and data. The Zabbix installation provided us with a file that sets this up for us.

Run the following command to set up the schema and import the data into the *zabbix* database. We'll use *zcat* since the data in the file is compressed.

```
$ zcat /usr/share/doc/zabbix-server-mysql/create.sql.gz | mysql –
u zabbix -p zabbix
```

Enter the password for the **zabbix** MySQL user that you configured when prompted.

This command will not output any errors if it was successful***. If you see the error ERROR 1045 (28000): Access denied for user 'zabbix'@'localhost' (using password: YES)*** then make sure you used the password for the ***zabbix*** user and not the ***root*** user.

In order for the Zabbix server to use this database, you need to set the database password in the Zabbix server configuration file. Open the configuration file in your editor:

```
$ sudo nano /etc/zabbix/zabbix_server.conf
```

Look for the following section of the file:

/etc/zabbix/zabbix_server.conf

```
### Option: DBPassword
#       Database password. Ignored for SQLite.
#       Comment this line if no password is used.
#
# Mandatory: no
# Default:
# DBPassword=
```

These comments in the file explain how to connect to the database. We need to set the `DBPassword` value in the file to the password for our database user. Add this line below those comments to configure the database:

/etc/zabbix/zabbix_server.conf

```
DBPassword=your_zabbix_mysql_password
```

That takes care of the Zabbix server configuration, but we have to make some modifications to our PHP setup in order for the Zabbix web interface to work properly.

Ref: https://tecadmin.net/install-zabbix-on-ubuntu/

Step 3 — Configuring PHP For Zabbix

The Zabbix web interface is written in PHP and requires some special PHP server settings. The Zabbix installation process created an Apache configuration file that contains these settings. It is located in the directory ***/etc/zabbix*** and is loaded automatically by Apache. We need to make a small change to this file, so open it up.

$ sudo nano /etc/zabbix/apache.conf

The file contains PHP settings that meet the necessary requirements for the Zabbix web interface. The only change you need to make is to set the appropriate timezone, which is commented out by default.

```
root@vuser-HVM-domU:~# cat /etc/zabbix/apache.conf
# Define /zabbix alias, this is the default
<IfModule mod_alias.c>
    Alias /zabbix /usr/share/zabbix
</IfModule>

<Directory "/usr/share/zabbix">
    Options FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
```

```
    <IfModule mod_php5.c>
      php_value max_execution_time 300
      php_value memory_limit 128M
      php_value post_max_size 16M
      php_value upload_max_filesize 2M
      php_value max_input_time 300
      php_value always_populate_raw_post_data -1
      php_value date.timezone Asia/Kolkata
    </IfModule>
    <IfModule mod_php7.c>
      php_value max_execution_time 300
      php_value memory_limit 128M
      php_value post_max_size 16M
      php_value upload_max_filesize 2M
      php_value max_input_time 300
      php_value always_populate_raw_post_data -1
      php_value date.timezone Asia/Kolkata
    </IfModule>
</Directory>……
```

Uncomment the time zone line, highlighted above, and change it to your time zone.

Now restart Apache to apply these new settings.

```
$ sudo systemctl restart apache2
```

You can now start the Zabbix server.

```
$ sudo systemctl start zabbix-server
```

Then check whether the Zabbix server is running properly:

```
$ sudo systemctl status zabbix-server
```

You will see the following status:

```
Output
● zabbix-server.service - Zabbix Server
   Loaded: loaded (/lib/systemd/system/zabbix-server.service; disabled;
vendor preset: enabled)
   Active: :active (running) since Thu 2017-06-08 06:40:43 UTC; 6s ago
  Process: 15201 ExecStart=/usr/sbin/zabbix_server -c $CONFFILE (code=exited,
status=0/SUCCESS)
```

. . .

Finally, enable the server to start at boot time:

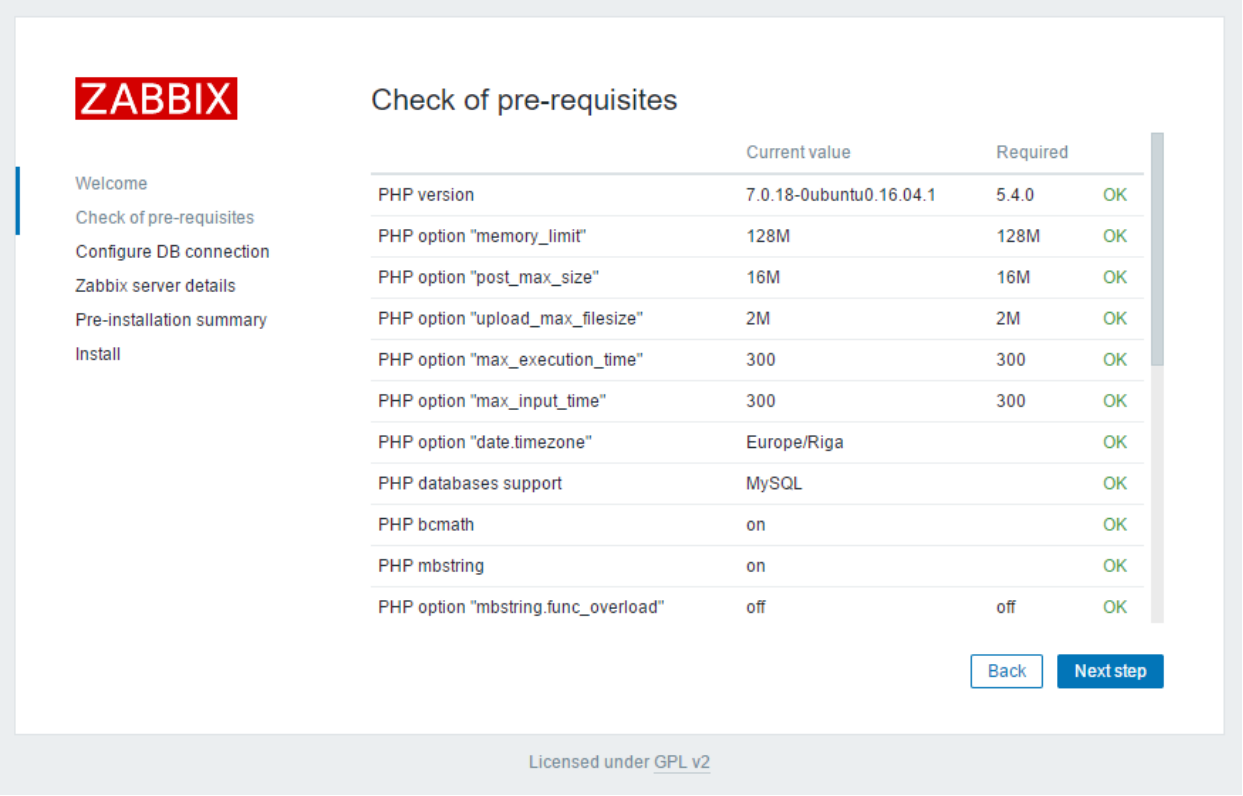
```
$ sudo systemctl enable zabbix-server
```

The server is set up and connected to the database. Now let's set up the web frontend.

Step 4 — Configuring Settings for the Zabbix Web Interface

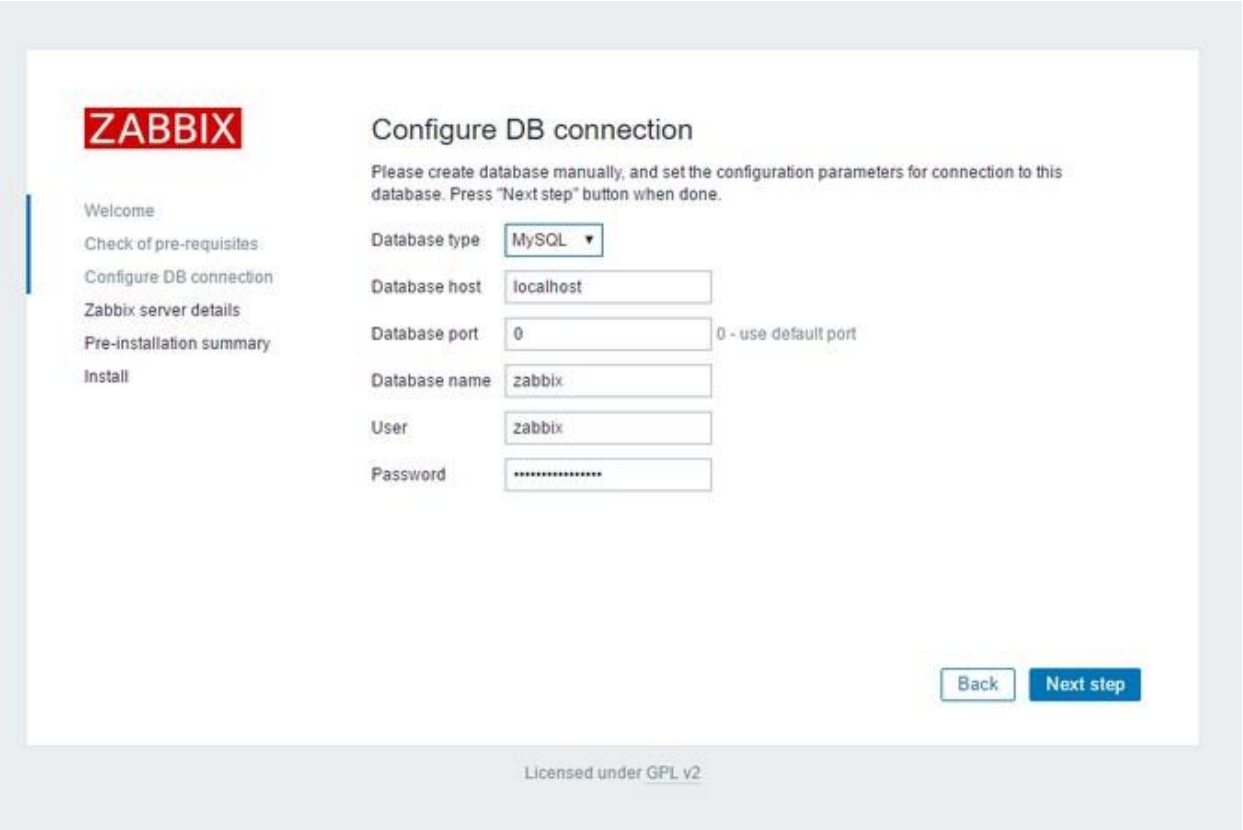
The web interface lets us see reports and add hosts that we want to monitor, but it needs some initial setup before we can use it. Launch your browser and go to the address *http://your_zabbix_server_ip_address/zabbix/.* On the first screen, you will see a welcome message. Click **Next step** to continue.

On the next screen, you will see the table that lists all of the prerequisites to run Zabbix.



All of the values in this table must be **OK**, so verify that they are. Be sure to scroll down and look at all of the prerequisites. Once you've verified that everything is ready to go, click **Next step** to proceed.

The next screen asks for database connection information.

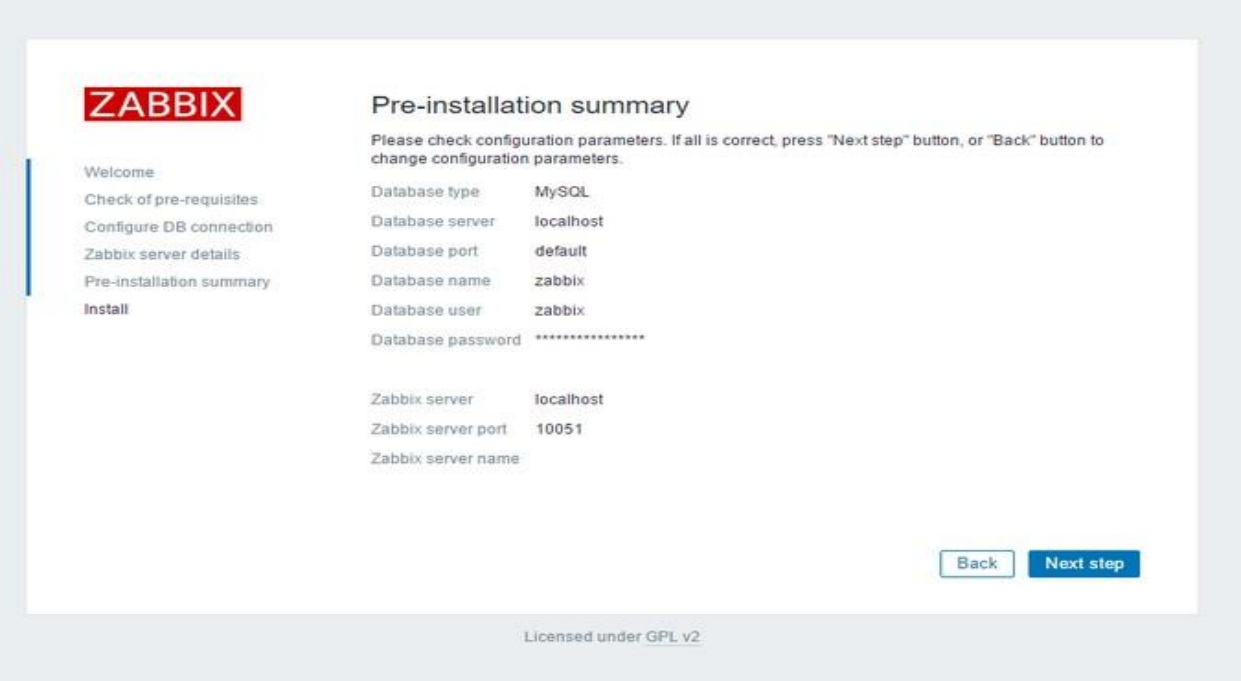


We told the Zabbix server about our database, but the Zabbix web interface also needs access to the database to manage hosts and read data. Therefore enter the MySQL credentials you configured in Step 2 and click **Next step** to proceed.

On the next screen, you can leave the options at their default values.

The **Name** is optional; it is used in the web interface to distinguish one server from another in case you have several monitoring servers. Click **Next step** to proceed.

The next screen will show the pre-installation summary so you can confirm everything is correct.



Click **Next step** to proceed to the final screen.

The web interface setup is complete! This process creates the configuration file ***/usr/share/zabbix/conf/zabbix.conf.php*** which you could back up and use in the future. Click **Finish** to proceed to the login screen. The default user is **Admin** and the password is **zabbix**.

Before we log in, let's set up the Zabbix agent on our other server.

Step 5 — Installing and Configuring the Zabbix Agent

Now we need to configure the agent software that will send monitoring data to the Zabbix server.

Log in to the second server, which we'll call the “monitored server”.

```
$ ssh sammy@your_monitored_server_ip_address
```

Then, just like on the Zabbix server, run the following commands to install the repository configuration package:

```
 $ wget
http://repo.zabbix.com/zabbix/3.2/ubuntu/pool/main/z/zabbix-release/zabbix-release_3.2-1+xenial_all.deb
```

```
$ sudo dpkg -i zabbix-release_3.2-1+xenial_all.deb
```

Next, update the package index:

```
$ sudo apt-get update
```

Then install the Zabbix agent:

```
$ sudo apt-get install zabbix-agent
```

While Zabbix supports certificate-based encryption, setting up a certificate authority is beyond the scope of this tutorial, but we can use pre-shared keys (PSK) to secure the connection between the server and agent.

So first, generate a PSK:

```
$ sudo sh -c "openssl rand -hex 32 > /etc/zabbix/zabbix_agentd.psk"
```

Show the key so you can copy it somewhere. You will need it to configure the host.

$ cat /etc/zabbix/zabbix_agentd.psk

The key will look something like this:

```
Output

cd12686e166a80aa09a227ae5f97834eaa3d5ae686d2ae39590f17ef85dd6de5
```

Now edit the Zabbix agent settings to set up its secure connection to the Zabbix server. Open the agent configuration file in your text editor:

```
$ sudo nano /etc/zabbix/zabbix_agentd.conf
```

Each setting within this file is documented via informative comments throughout the file, but you only need to edit some of them.

First you have to edit the IP address of the Zabbix server. Find the following section:

/etc/zabbix/zabbix_agentd.conf

```
### Option: Server
#       List of comma delimited IP addresses (or hostnames) of Zabbix
servers.
#       Incoming connections will be accepted only from the hosts listed
here.
#       If IPv6 support is enabled then '127.0.0.1', '::127.0.0.1',
'::ffff:127.0.0.1' are treated equally.
#
# Mandatory: no
# Default:
# Server=

Server=127.0.0.1
```

Change the default value to the IP of your Zabbix server:

/etc/zabbix/zabbix_agentd.conf

```
Server=your_zabbix_server_ip_address
```

```
Server=172.31.20.196
ServerActive=172.31.20.196:10051
```

Next, find the section that configures the secure connection to the Zabbix server and enable pre-shared key support. Find the `TSLConnect` section, which looks like this:

/etc/zabbix/zabbix_agentd.conf

```
### Option: TLSConnect
#       How the agent should connect to server or proxy. Used for active
checks.
#       Only one value can be specified:
#               unencrypted - connect without encryption
#               psk         - connect using TLS and a pre-shared key
#               cert        - connect using TLS and a certificate
#
# Mandatory: yes, if TLS certificate or PSK parameters are defined (even for
'unencrypted' connection)
# Default:
# TLSConnect=unencrypted
```

Then add this line to configure pre-shared key support:

/etc/zabbix/zabbix_agentd.conf

```
TLSConnect=psk
```

Next, locate the TLSAccept section, which looks like this:

/etc/zabbix/zabbix_agentd.conf

```
### Option: TLSAccept
#       What incoming connections to accept.
#       Multiple values can be specified, separated by comma:
#               unencrypted - accept connections without encryption
#               psk         - accept connections secured with TLS and a pre-
shared key
#               cert        - accept connections secured with TLS and a
certificate
#
# Mandatory: yes, if TLS certificate or PSK parameters are defined (even for
'unencrypted' connection)
# Default:
# TLSAccept=unencrypted
```

Configure incoming connections to support pre-shared keys by adding this line:

/etc/zabbix/zabbix_agentd.conf

```
TLSAccept=psk
```

Next, find the TLSPSKIdentity section, which looks like this:

/etc/zabbix/zabbix_agentd.conf

```
### Option: TLSPSKIdentity
#       Unique, case sensitive string used to identify the pre-shared key.
#
# Mandatory: no
# Default:
# TLSPSKIdentity=
```

Choose a unique name to identify your pre-shared key by adding this line:

/etc/zabbix/zabbix_agentd.conf

```
TLSPSKIdentity=PSK 001
```

You'll use this as the **PSK ID** when you add your host through the Zabbix web interface.

Then set the option which points to your previously created pre-shared key. Locate the TLSPSKFile option:

/etc/zabbix/zabbix_agentd.conf

```
### Option: TLSPSKFile
#       Full pathname of a file containing the pre-shared key.
#
# Mandatory: no
# Default:
# TLSPSKFile=
```

Add this line to point the Zabbix agent to your PSK file you created:

/etc/zabbix/zabbix_agentd.conf

```
TLSPSKFile=/etc/zabbix/zabbix_agentd.psk
```

Save and close the file. Now you can start the Zabbix agent and set it to start at boot time:

```
$ sudo systemctl start zabbix-agent
```

```
$ sudo systemctl enable zabbix-agent
```

For good measure, check that the Zabbix agent is running properly:

```
$ sudo systemctl status zabbix-agent
```

You will see the following status, indicating the agent is running:

```
Output

● zabbix-agent.service - Zabbix Agent
   Loaded: loaded (/lib/systemd/system/zabbix-agent.service; disabled; vendor
preset: enabled)
   Active: active (running) since Thu 2017-06-08 08:33:52 UTC; 4s ago
  Process: 18185 ExecStart=/usr/sbin/zabbix_agentd -c $CONFFILE (code=exited,
status=0/SUCCESS)
  ...
```

Our agent is now ready to send data to the Zabbix server. But in order to use it, we have to link to it from the server's web console.

Note: If you are using UFW, configure it to allow connections to port 10050

```
$ sudo ufw allow 10050/tcp
$ sudo ufw allow 10051/tcp
```

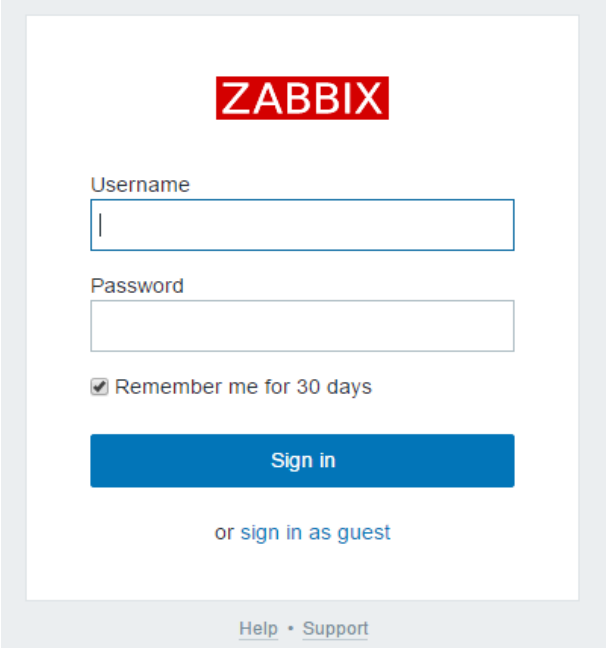
You have successfully complets to install zabbix server and agent in different server.

Now lets monitor the host just you have installed zabbix agent server…

5. Adding the New Host to Zabbix Server

A host in Zabbix is a networked entity (physical, virtual) that you wish to monitor. The definition of what can be a “host” in Zabbix is quite flexible. It can be a physical server, a network switch, a virtual machine or some application.

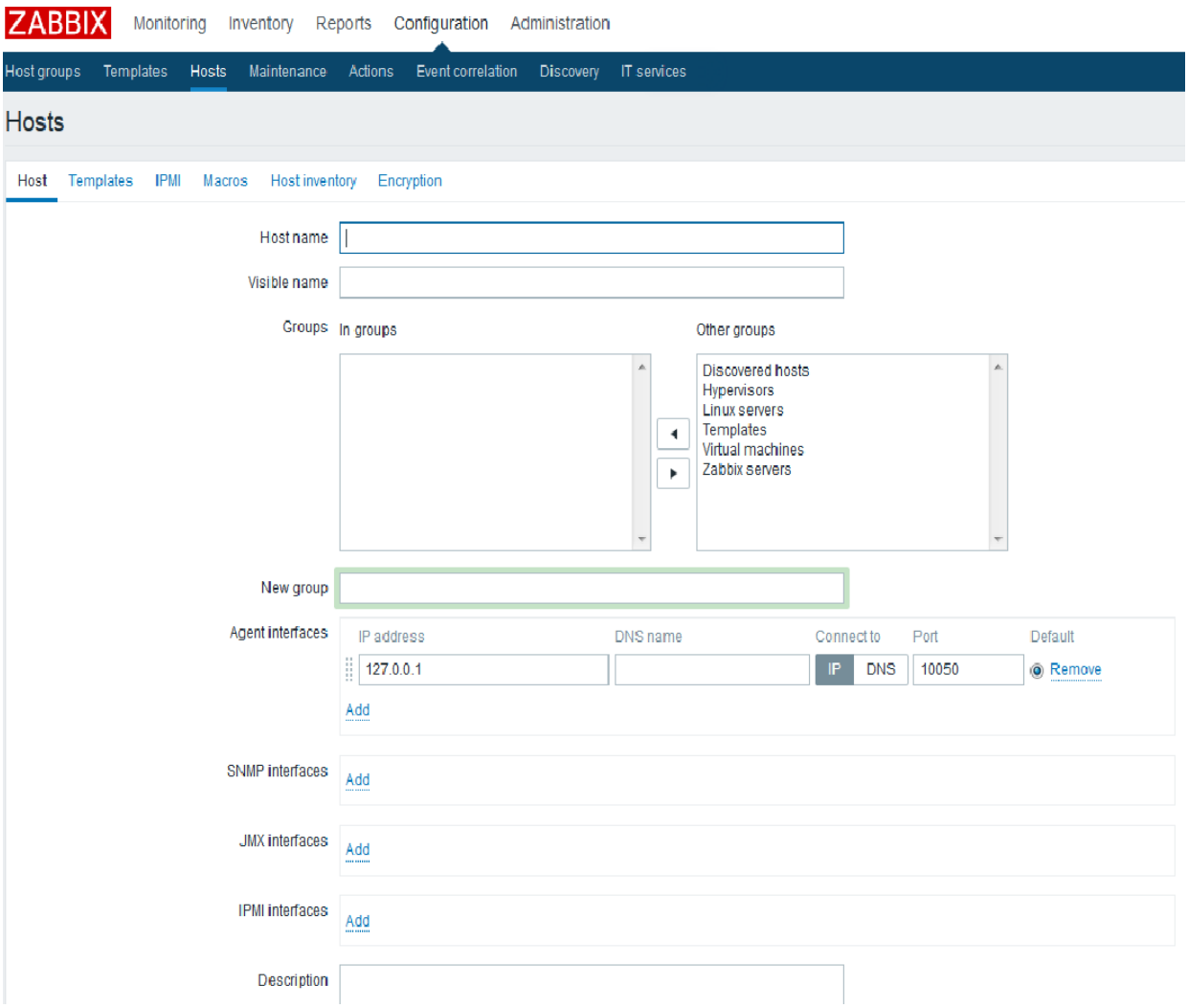
1. Login zabbix server. http://your zabbix server ip address/zabbix/



Default user name and password is Admin/zabbix

Information about configured hosts in Zabbix is available in *Configuration* → *Hosts*. There is already one pre-defined host, called 'Zabbix server', but we want to learn adding another.

To add a new host, click on *Create host*. This will present us with a host configuration form.



Host name

- Enter a host name. Alphanumerics, spaces, dots, dashes and underscores are allowed.

Groups

- Select one or several groups from the right hand side selectbox and click on **«** to move them to the 'In groups' selectbox.

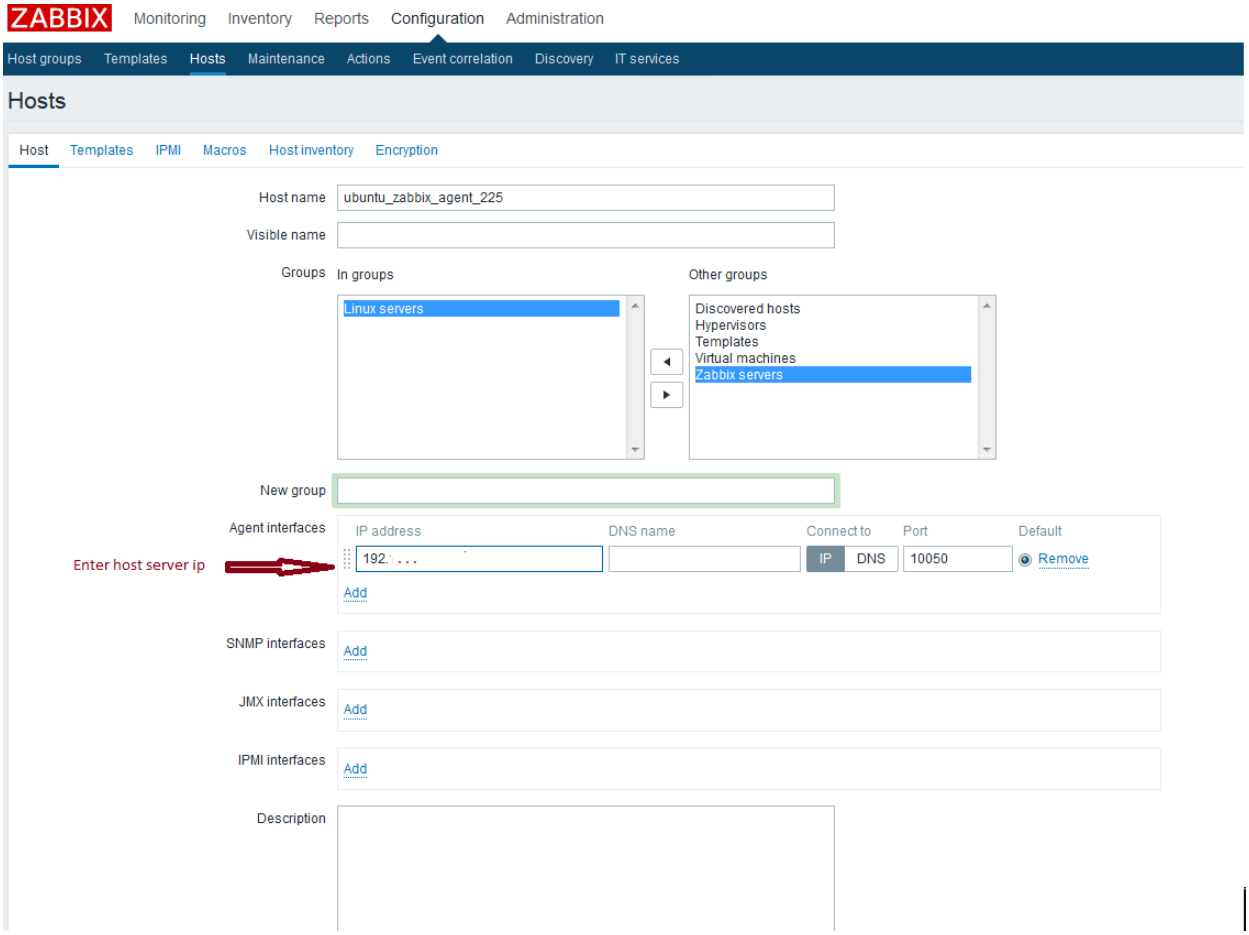
Note: All access permissions are assigned to host groups, not individual hosts. That is why a host must belong to at least one group.

IP address

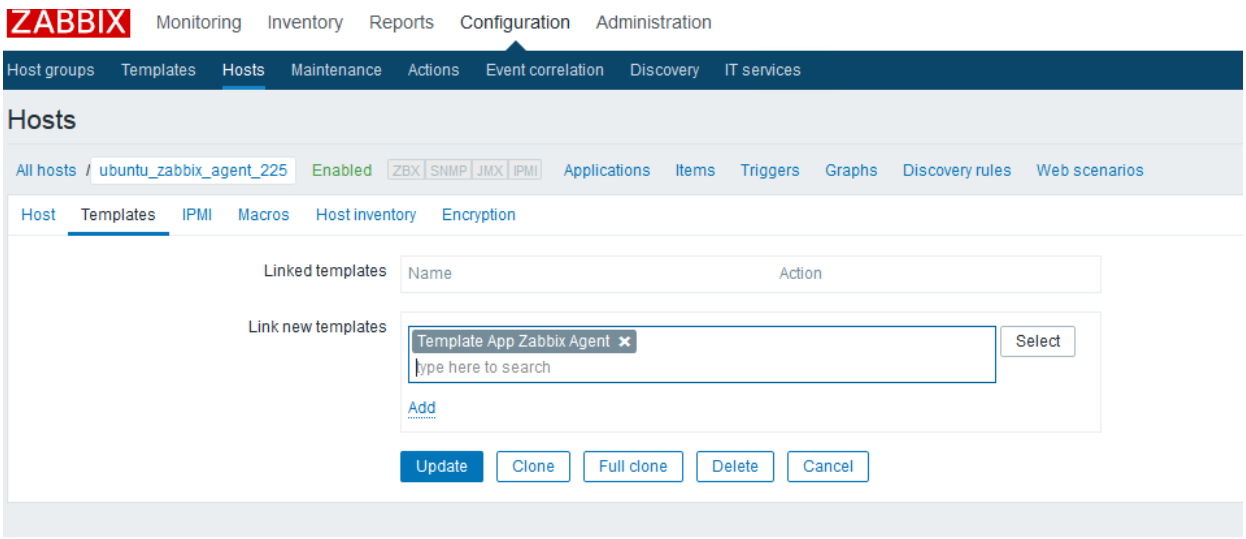
- Enter the IP address of the host. Note that if this is the Zabbix server IP address, it must be specified in the Zabbix agent configuration file 'Server' directive.

Other options will suit us with their defaults for now.

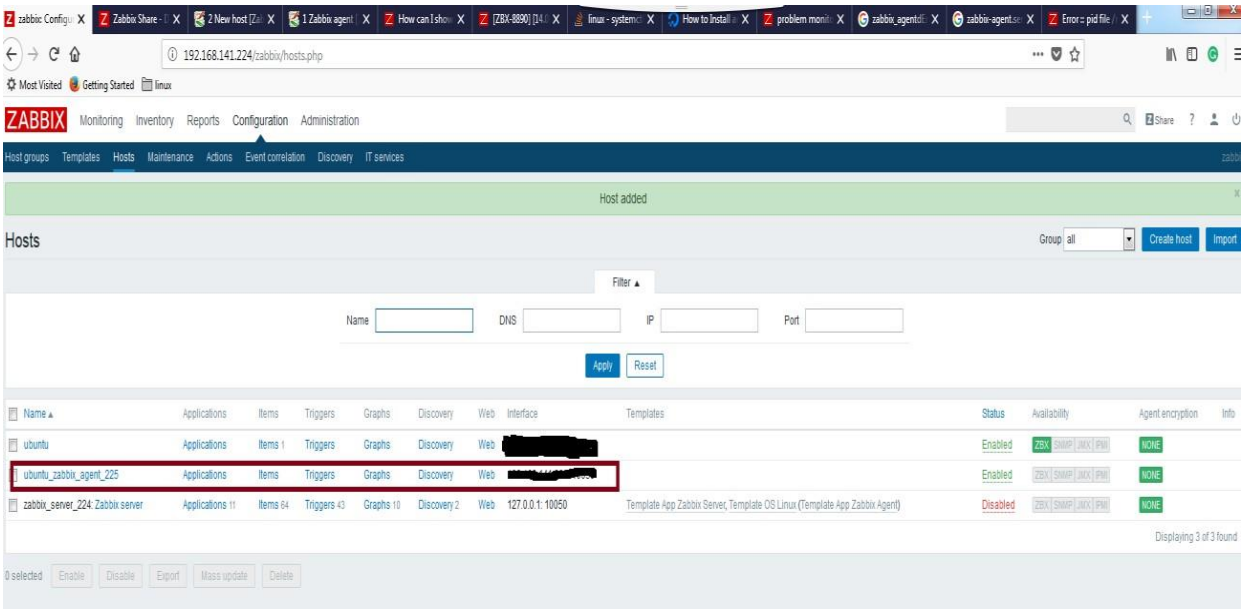
When done, click *Add*. Your new host should be visible in the hostlist.



Add template of zabbix agent and click the add button do the update .



Now we can see the newly added host in the configuration->host list.

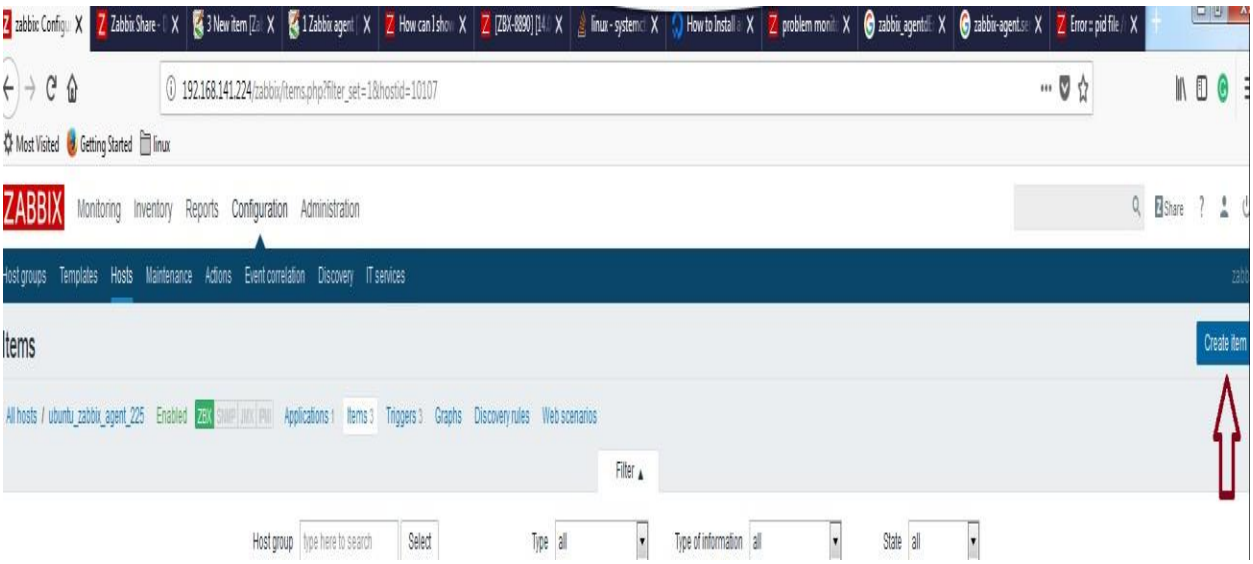


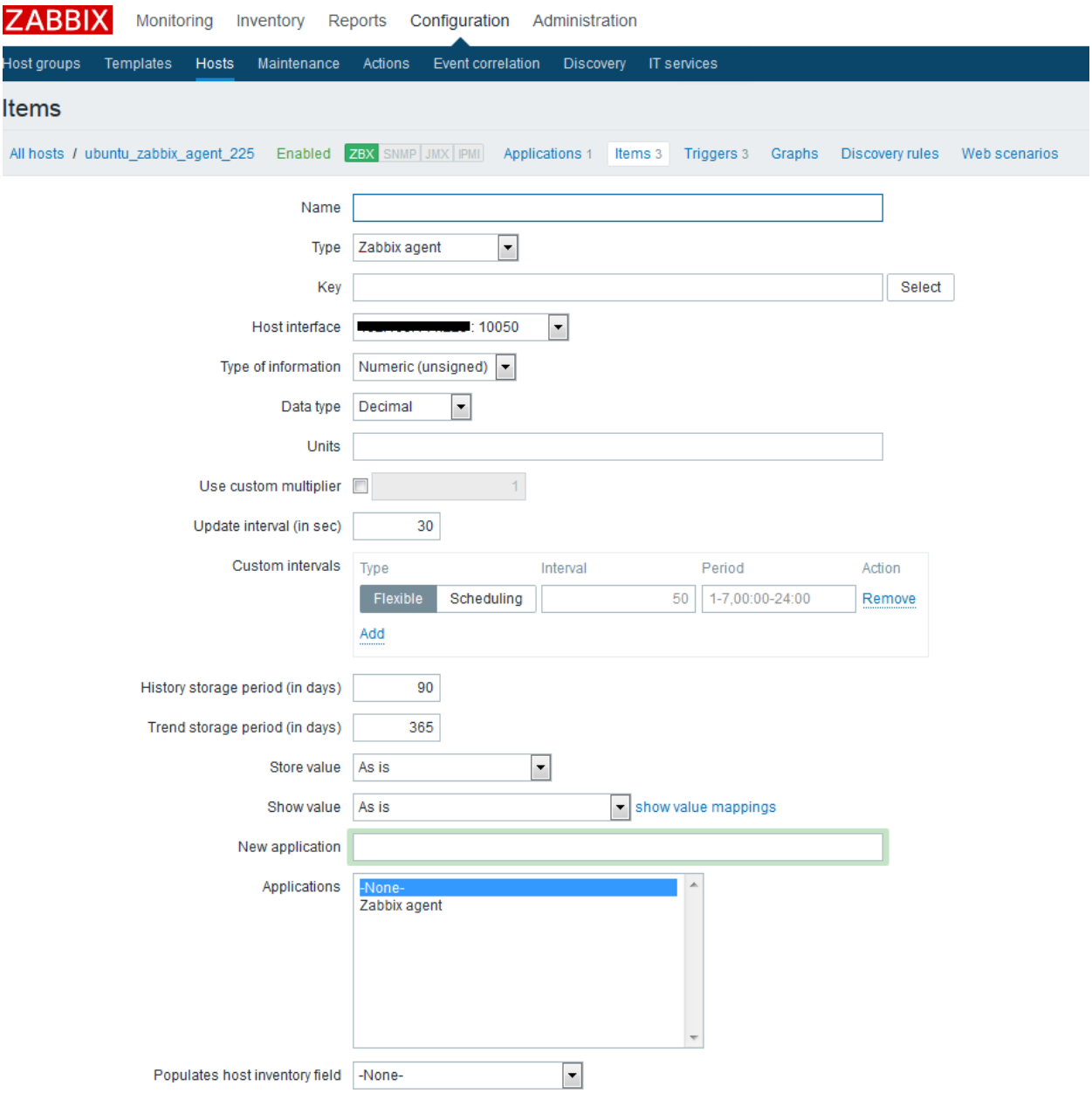
6. Create New Item.

Items are the basis of gathering data in Zabbix. Without items, there is no data - because only an item defines a single metric or what data to get off of a host.

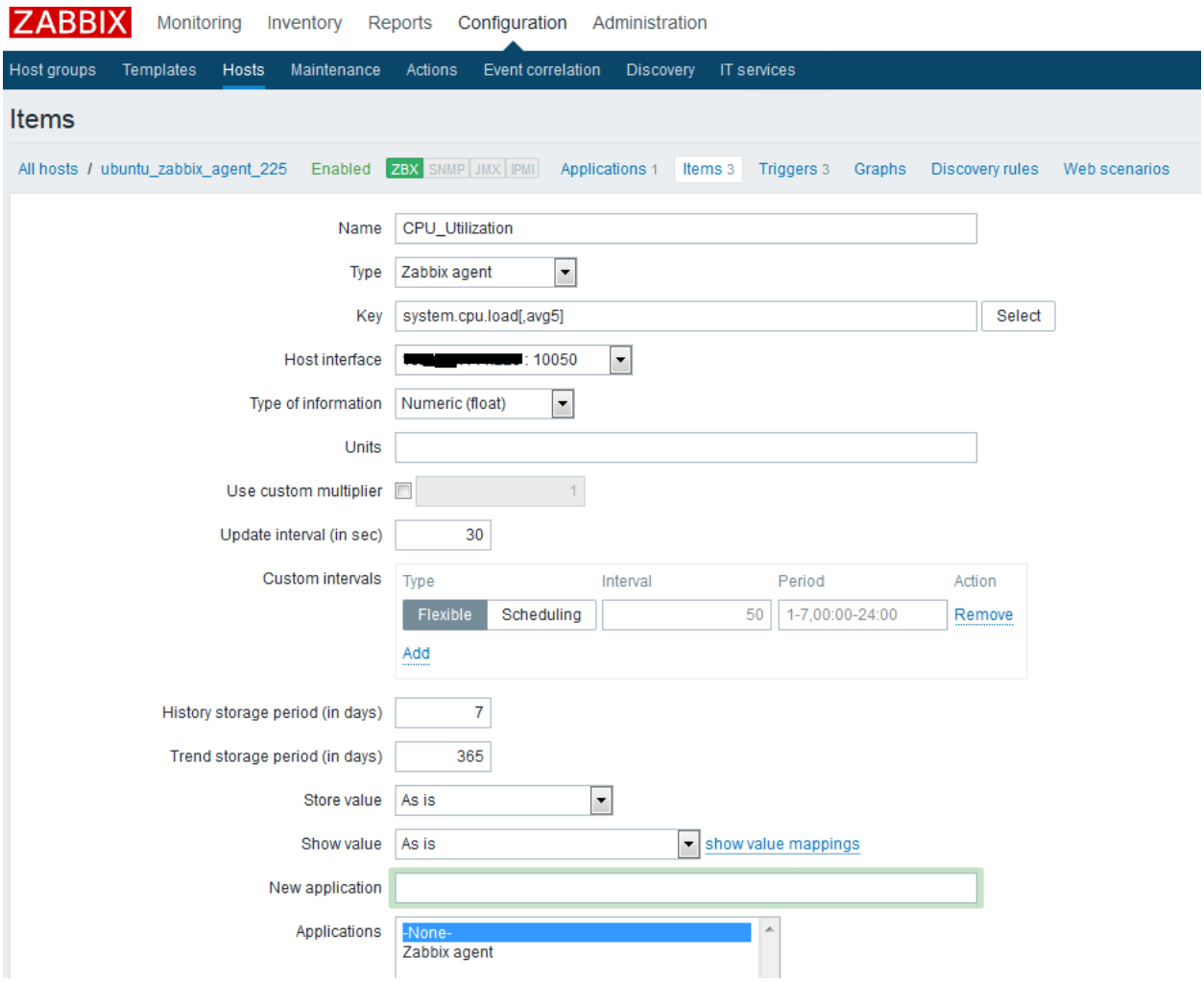
All items are grouped around hosts. That is why to configure a sample item we go to *Configuration* → *Hosts* and find the “ubuntu_zabbix_agent_225” we have created.

The *Items* link in the row of ” ubuntu_zabbix_agent_225” should display a count of '0'. Click on the link, and then click on *Create item*. This will present us with an item definition form.





Add the name of the item and select a key what exactly you want to monitor. In my case I am monitoring cpu load of my host for that I am selecting key **system.cpu.load[,avg5].**



For our sample item, the essential information to enter is:

Name

- Enter *CPU Load* as the value. This will be the item name displayed in lists and elsewhere.

Key

- Manually enter *system.cpu.load* as the value. This is a technical name of an item that identifies the type of information that will be gathered. The particular key is just one of pre-defined keys that come with Zabbix agent.

Type of information

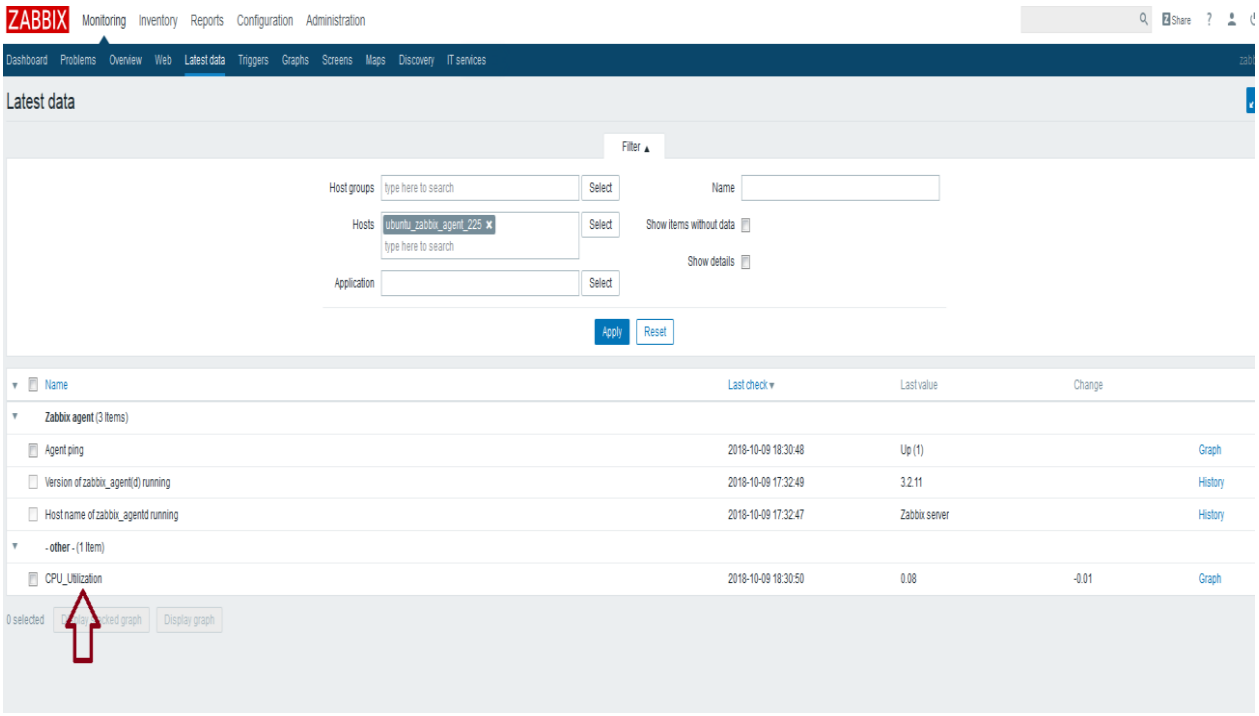
- Select *Numeric (float)* here. This attribute defines the format of expected data.

Note: You may also want to reduce the amount of days item history will be kept, to 7 or 14. This is good practice to relieve the database from keeping lots of historical values.

Seeing data

With an item defined, you might be curious if it is actually gathering data. For that, go to *Monitoring → Latest data*, select 'New host' in the filter and click on *Apply*.

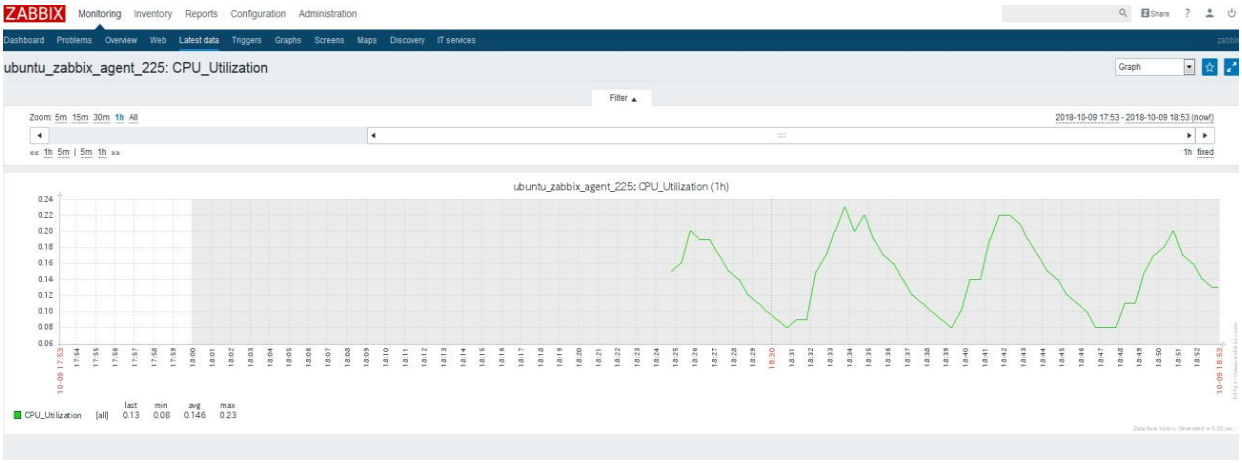
Then click on the **+** before **- other -** and expect your item to be there and displaying data.



Graphs

With the item working for a while, it might be time to see something visual. **Simple graphs** are available for any monitored numeric item without any additional configuration. These graphs are generated on runtime.

To view the graph, go to *Monitoring → Latest data* and click on the 'Graph' link next to the item.



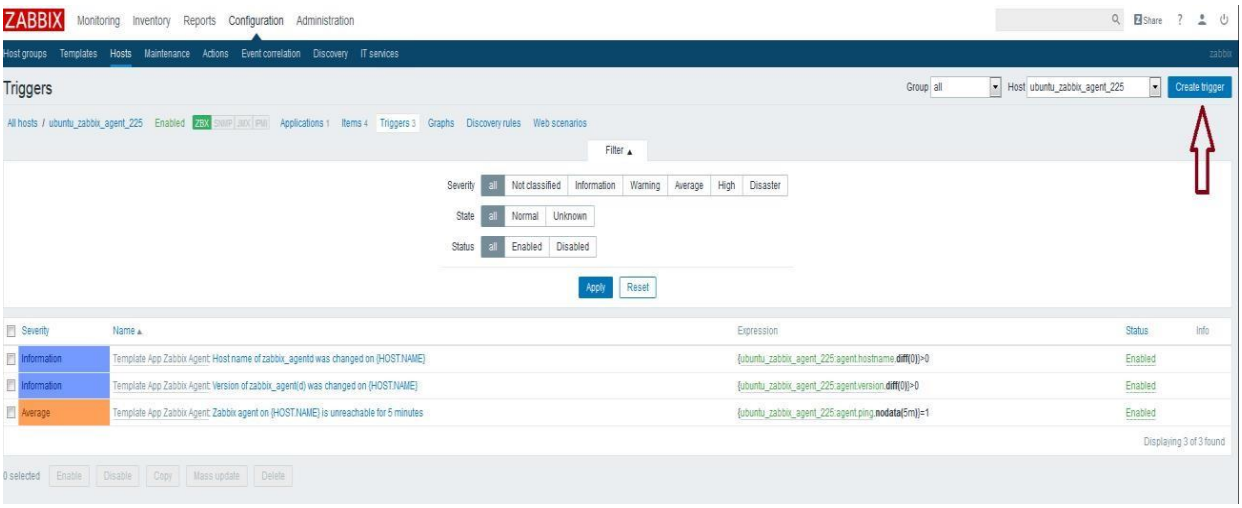
7. Create a Trigger

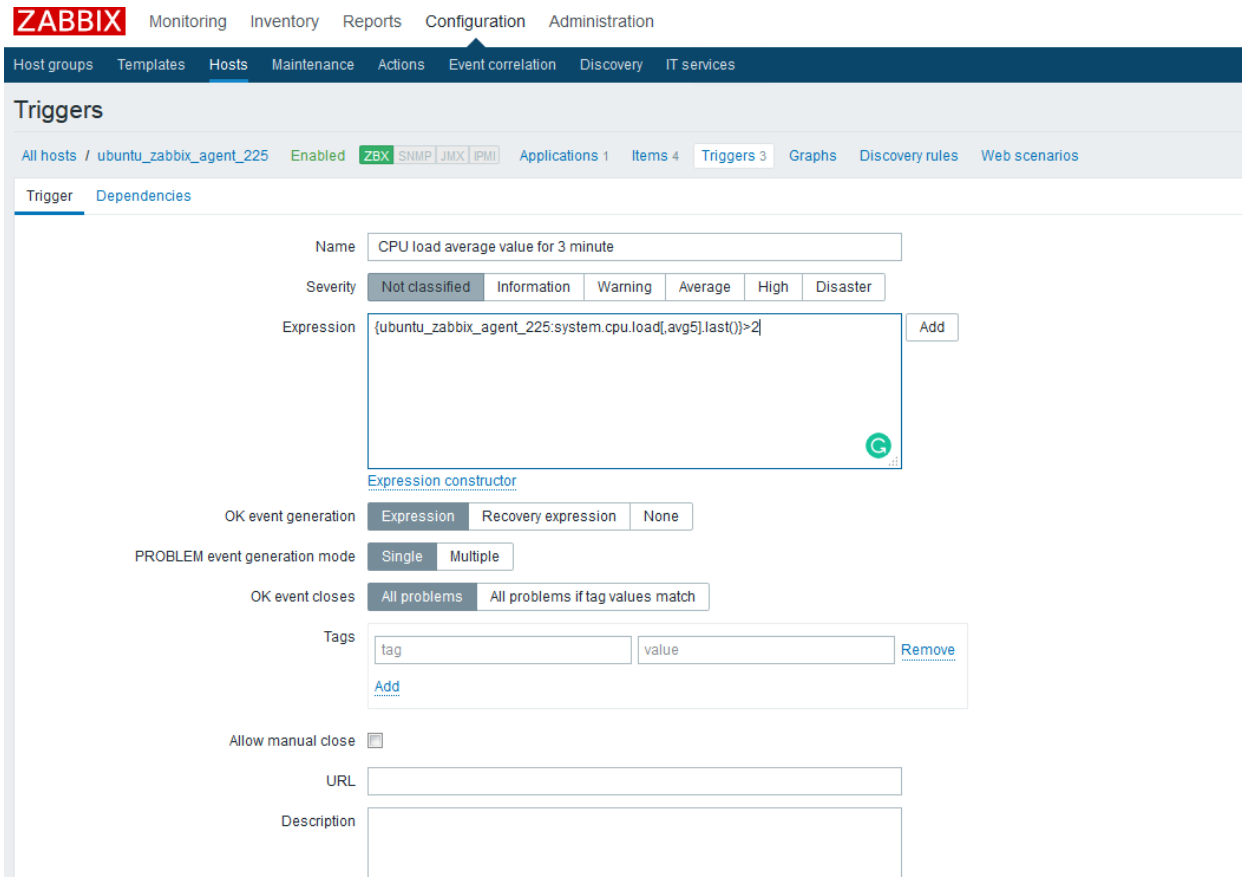
Items only collect data. To automatically evaluate incoming data we need to define triggers. A trigger contains an expression that defines a threshold of what is an acceptable level for the data.

If that level is surpassed by the incoming data, a trigger will “fire” or go into a 'Problem' state - letting us know that something has happened that may require attention. If the level is acceptable again, trigger returns to an 'Ok' state.

Adding trigger

To configure a trigger for our item, go to *Configuration* → *Hosts*, find *ubuntu_zabbix_agent_225* and click on *Triggers* next to it and then on *Create trigger*. This presents us with a trigger definition form.





For our trigger, the essential information to enter here is:

Name

- Enter *CPU load too high on 'New host' for 3 minutes* as the value. This will be the trigger name displayed in lists and elsewhere.

Expression

- Enter: {New host:system.cpu.load.avg(180)}>2

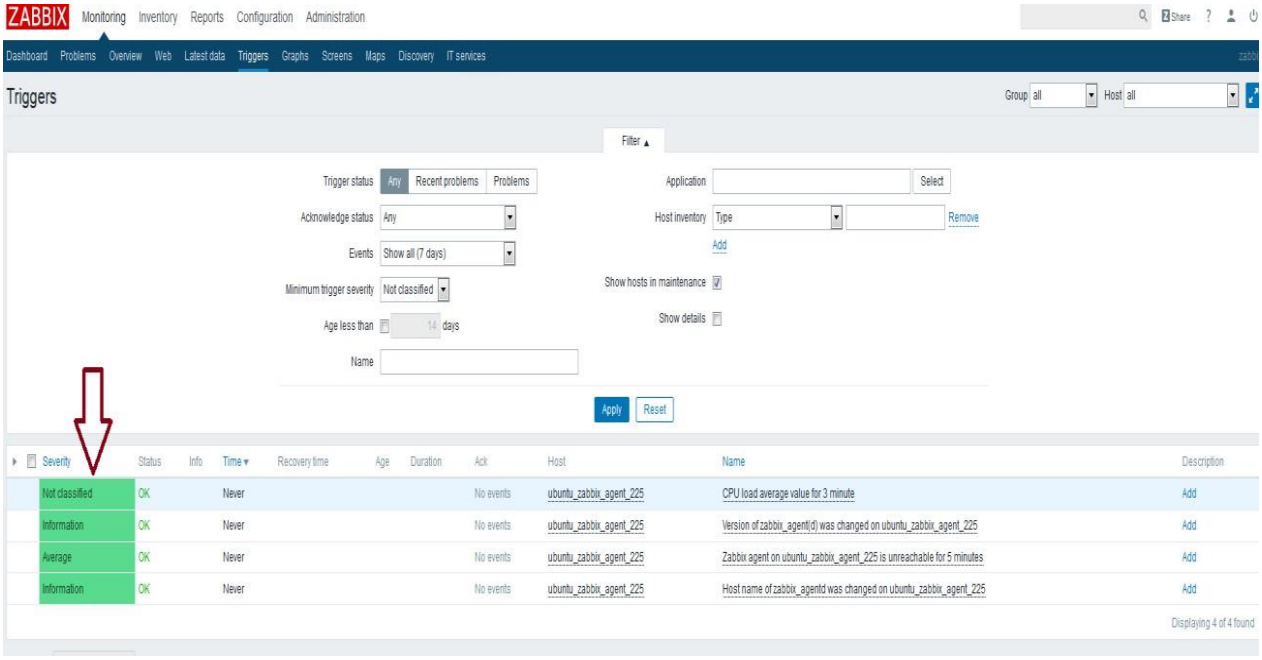
This is the trigger expression. Make sure that the expression is entered right, down to the last symbol. The item key here (system.cpu.load) is used to refer to the item. This particular expression basically says that the problem threshold is exceeded when the CPU load average value for 3 minutes is over 2. You can learn more about the syntax of trigger expressions.

When done, click *Add*. The new trigger should appear in the trigger list.

Displaying trigger status

With a trigger defined, you might be interested to see its status.

For that, go to *Monitoring* → *Triggers*. After 3 minutes or so (we asked to evaluate a 3-minute average after all) your trigger should appear there, presumably with a green 'OK' flashing in the 'Status' column.



8. Receiving problem notification

With items collecting data and triggers designed to “fire” upon problem situations, it would also be useful to have some alerting mechanism in place that would notify us about important events even when we are not directly looking at Zabbix frontend.

This is what notifications do. E-mail being the most popular delivery method for problem notifications, we will learn how to set up an e-mail notification.

E-mail settings

Initially there are several predefined notification delivery methods in Zabbix. ***E-mail*** is one of those.

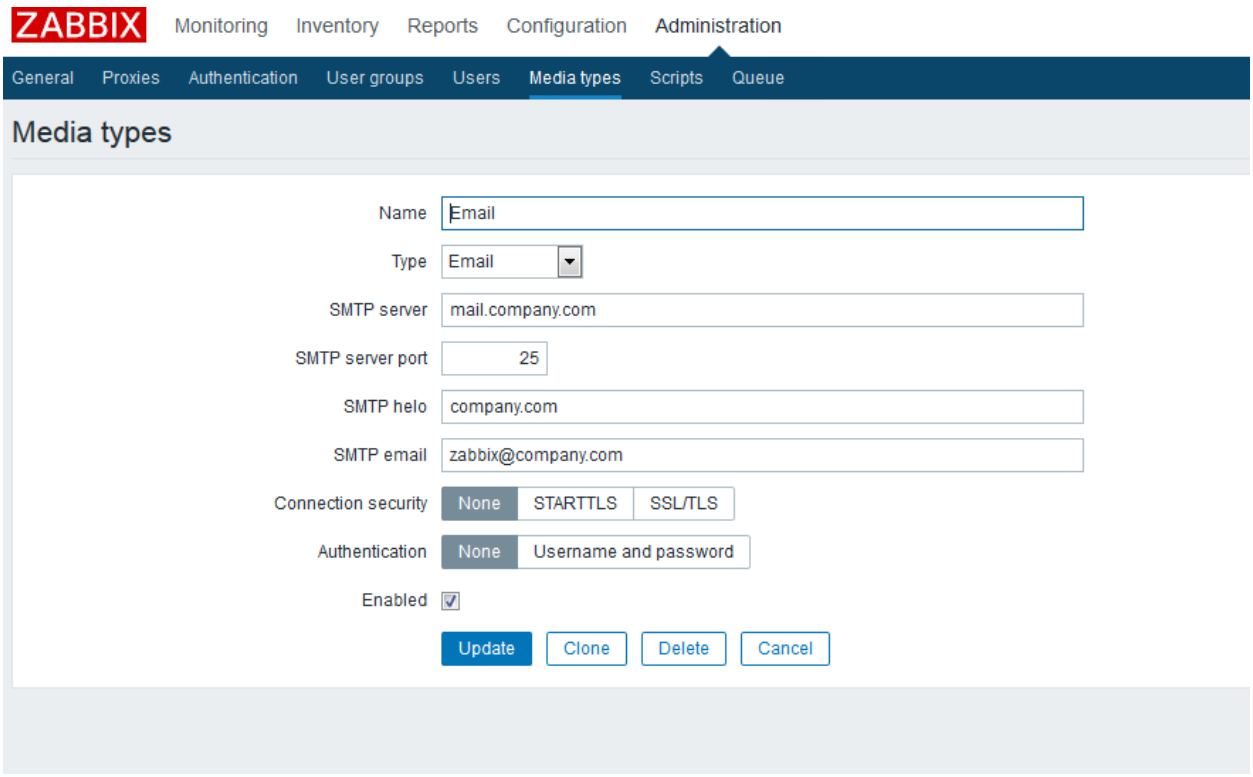
Media types.

Media are the delivery channels used for sending notifications and alerts in Zabbix.

You can configure several media types:

- E-mail
- SMS
- Jabber
- Ez Texting
- Custom alertscripts

To configure e-mail settings, go to *Administration* → *Media types* and click on *Email* in the list of pre-defined media types.



Fill the correct information to send a mail notification (we don't have a smtp server access right now).

Set the values of SMTP server, SMTP helo and SMTP e-mail to the appropriate for your environment.

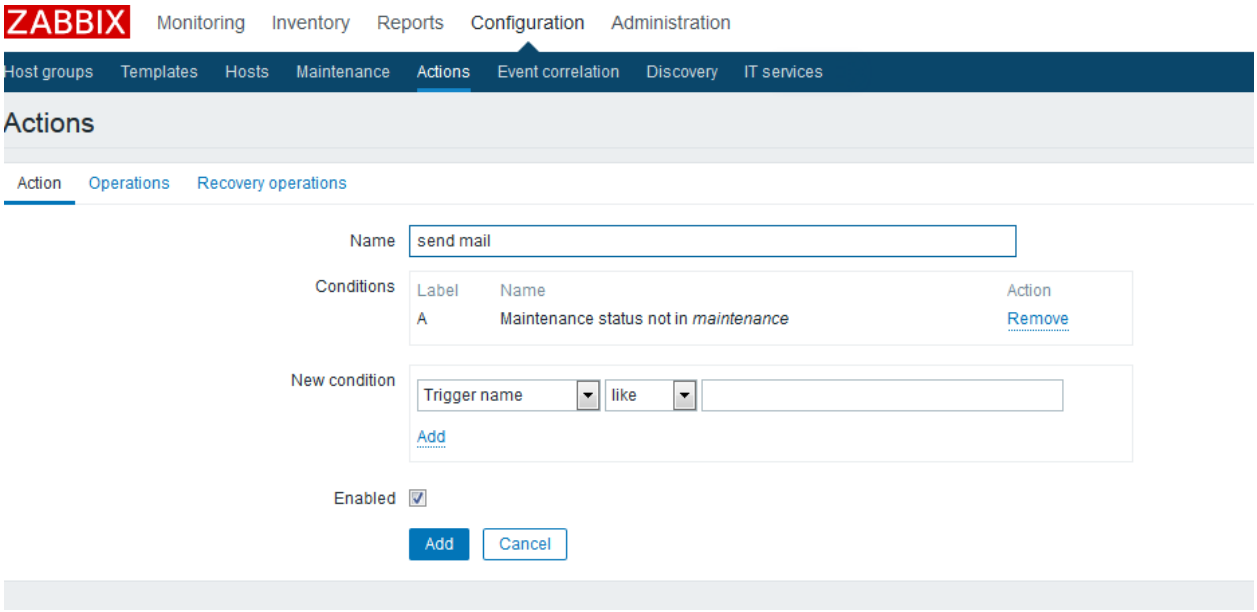
Note: SMTP email' will be used as the 'From' address for the notifications sent from Zabbix.

Press *Update* when ready.

Now you have configured 'Email' as a working media type. A media type must be linked to users by defining specific delivery addresses (like we did when configuring a new user), otherwise it will not be used.

New action

Delivering notifications is one of the things actions do in Zabbix. Therefore, to set up a notification, go to *Configuration* → *Actions* and click on *Create action*.



Fill the values in operation tab.

Please have a look below url to cinfigure email notification completely.

https://www.zabbix.com/documentation/3.4/manual/quickstart/notification

9. Create New template

Previously we learned how to set up an item, a trigger and how to get a problem notification for the host.

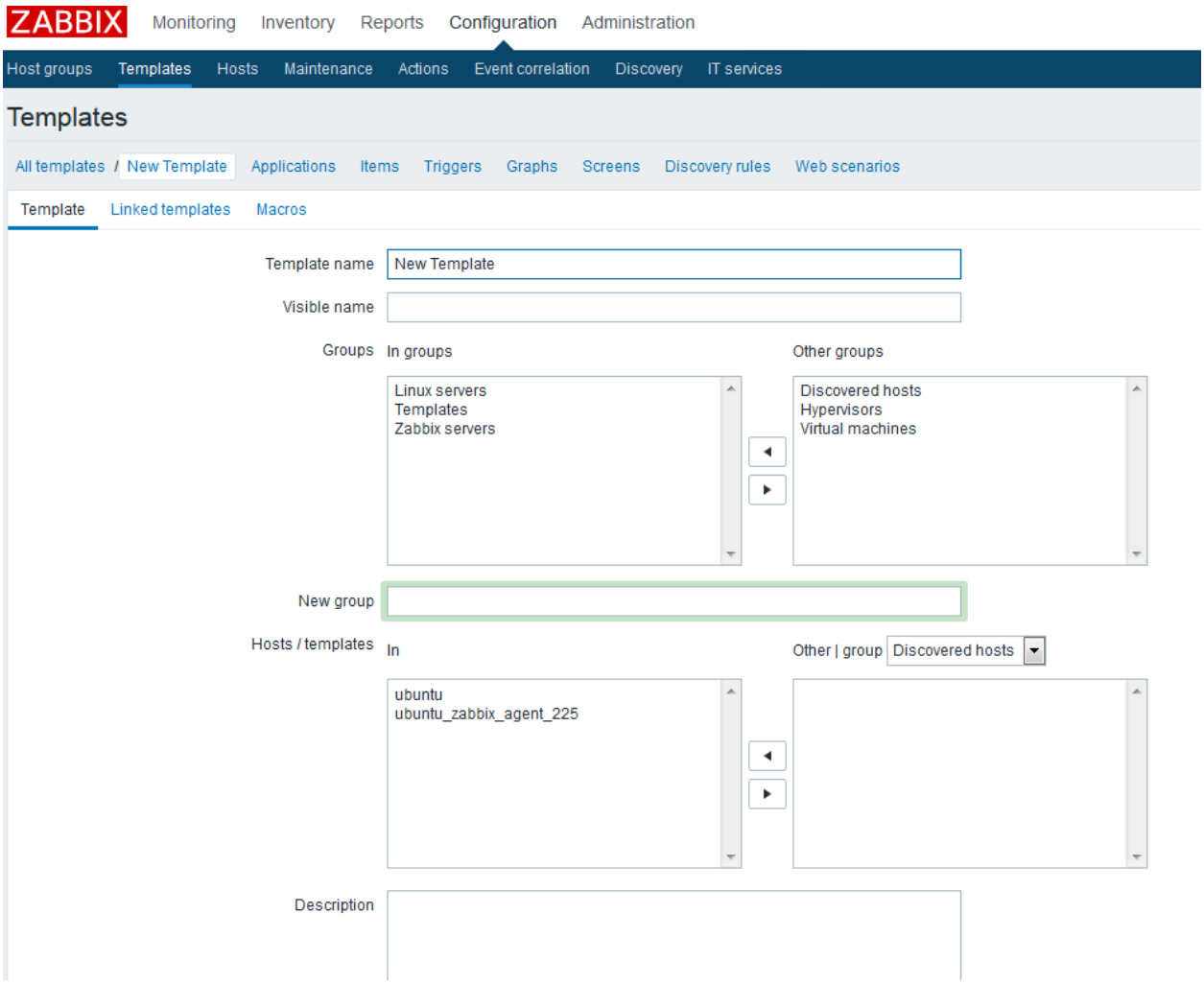
While all of these steps offer a great deal of flexibility in themselves, it may appear like a lot of steps to take if needed for, say, a thousand hosts. Some automation would be handy.

This is where templates come to help. Templates allow to group useful items, triggers and other entities so that those can be reused again and again by applying to hosts in a single step.

When a template is linked to a host, the host inherits all entities of the template. So, basically a pre-prepared bunch of checks can be applied very quickly.

Adding template

To start working with templates, we must first create one. To do that, in *Configuration* → *Templates* click on *Create template*. This will present us with a template configuration form**.**



The required parameters to enter here are:

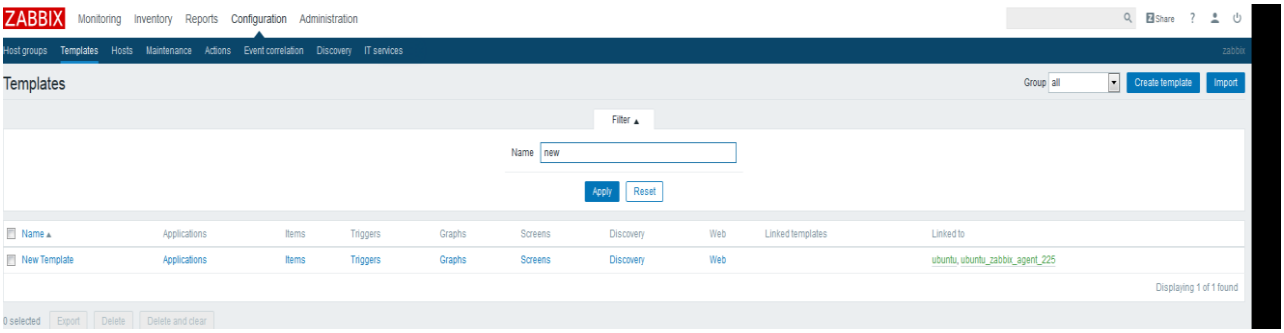
Template name

- Enter a template name. Alpha-numericals, spaces and underscores are allowed.

Groups

- Select one or several groups from the right hand side selectbox and click on **«** to move them to the 'In groups' selectbox. The template must belong to a group.

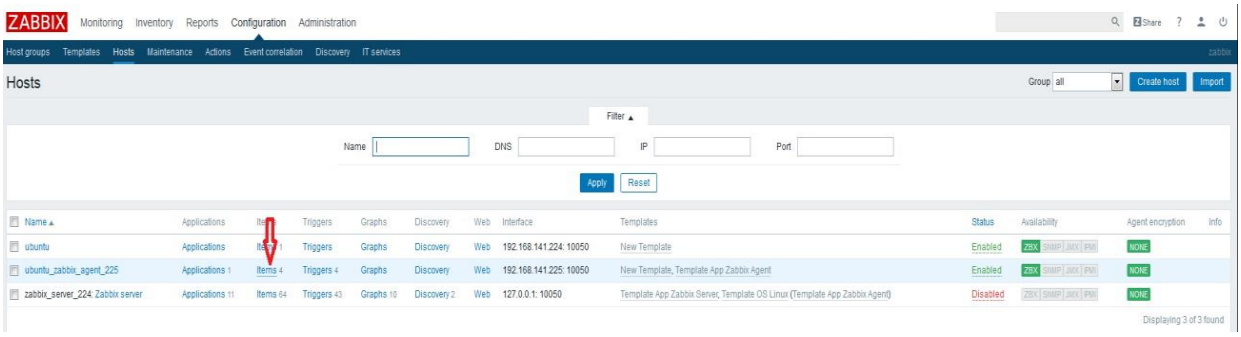
When done, click *Add*. Your new template should be visible in the list of templates.



As you may see, the template is there, but it holds nothing in it - no items, triggers or other entities.

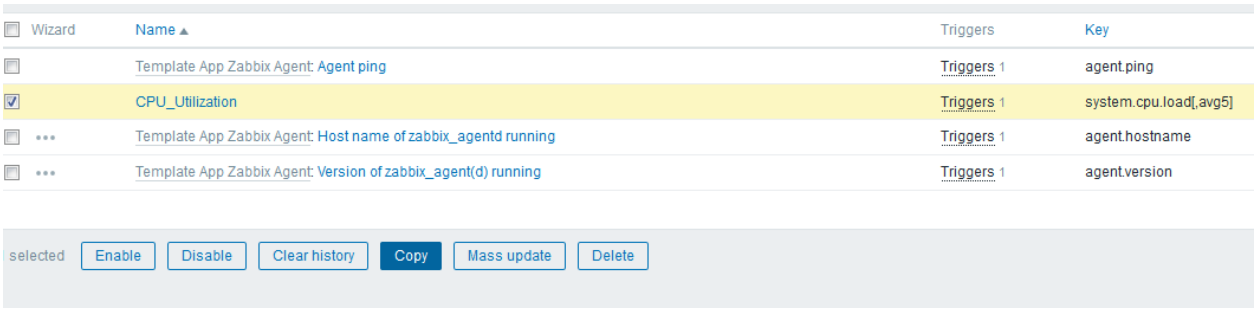
Adding item to template

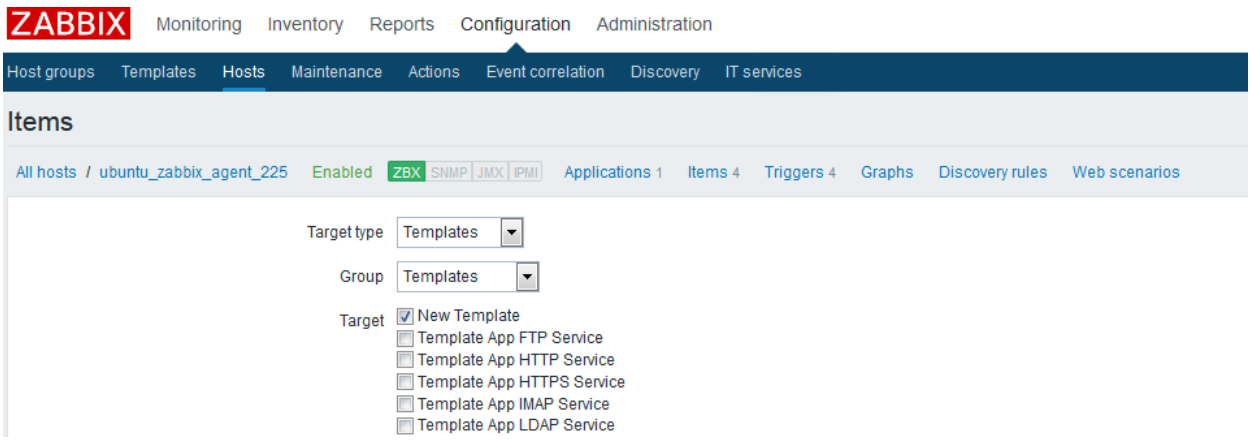
To add an item to the template, go to the item list for '`ubuntu_zabbix_agent_225`'. In *Configuration → Hosts* click on *Items* next to '`ubuntu_zabbix_agent_225`'.

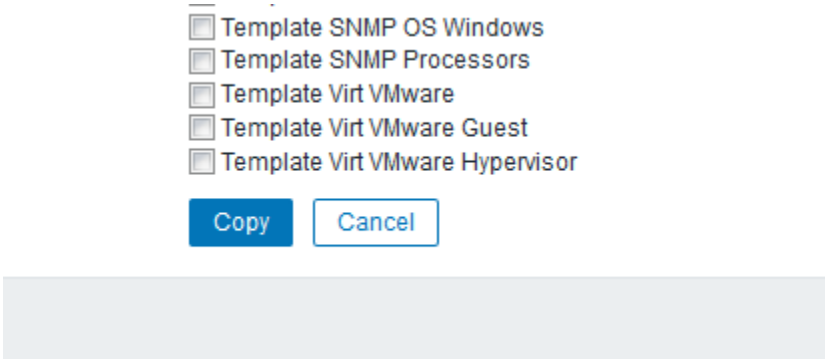


Then:

- mark the checkbox of the 'CPU Load' item in the list
- click on *Copy* below the list
- select the template to copy item to

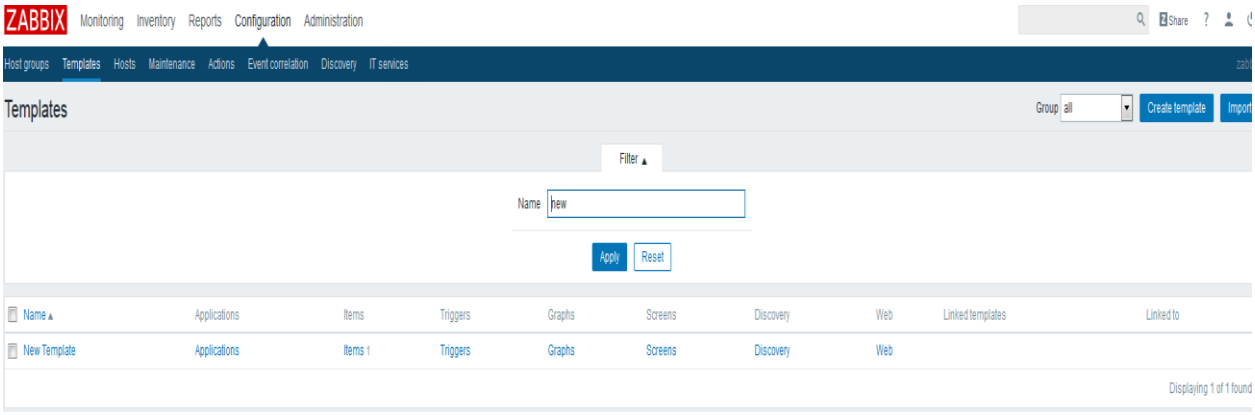






- click on *Copy*

If you now go to *Configuration* → *Templates*, 'New template' should have one new item in it.

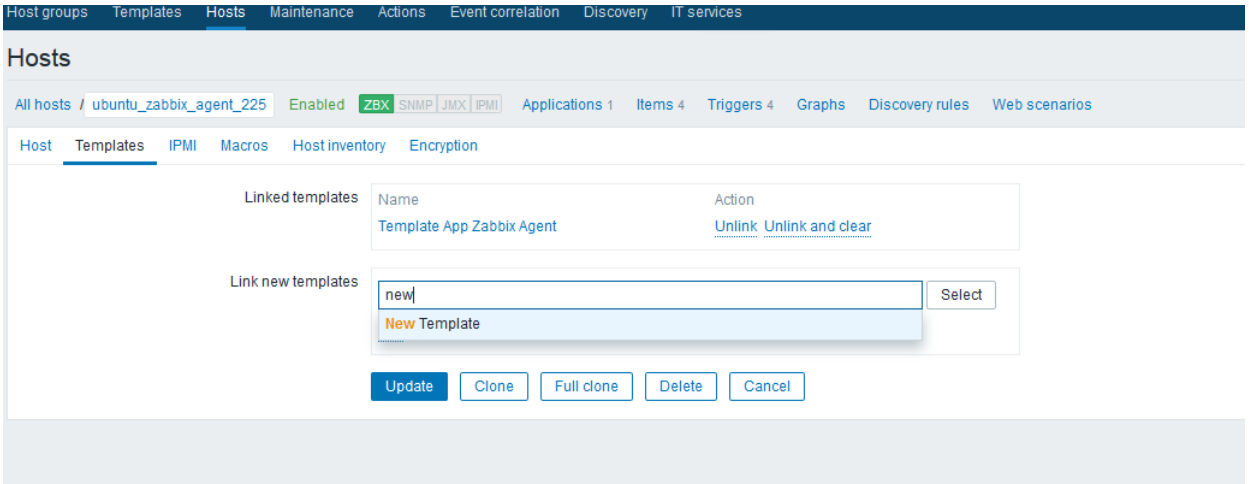


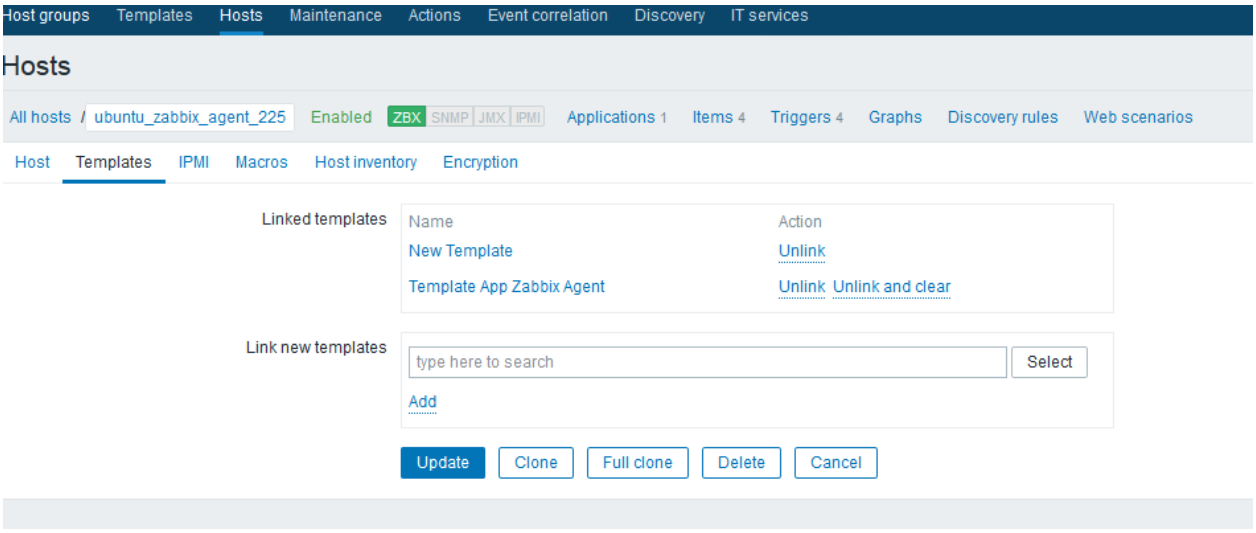
We will stop at one item only for now, but similarly you can add any other items, triggers or other entities to the template until it's a fairly complete set of entities for given purpose (monitoring OS, monitoring single application).

Linking template to host

With a template ready, it only remains to add it to a host. For that, go to *Configuration* → *Hosts*, click on 'ubuntu_zabbix_agent_225' to open its property form and go to the **Templates** tab.

There, click on *Select* next to *Link new templates*. In the pop-up window click on the name of template we have created ('New template'). As it appears in the *Link new templates* field, click on *Add*. The template should appear in the *Linked templates* list.





Click *Update* in the form to save the changes. The template is now added to the host, with all entities that it holds.

As you may have guessed, this way it can be applied to any other host as well. Any changes to the items, triggers and other entities at the template level will propagate to the hosts the template is linked to.

10. Check the zabbix server logs

1. Goto zabbbix server
2. Goto below path.

   root@vuser-HVM-domU:/var/log/zabbix# pwd

   /var/log/zabbix

   root@vuser-HVM-domU:/var/log/zabbix# ls -ltr

   total 1688

   -rw-rw-r-- 1 zabbix zabbix 39856 Sep 30 07:34 zabbix_server.log.2.gz

   -rw-rw-r-- 1 zabbix zabbix 11369 Sep 30 07:34 zabbix_agentd.log.2.gz

   -rw-r----- 1 zabbix zabbix 675255 Oct 8 07:34 zabbix_server.log.1

-rw-r----- 1 zabbix zabbix 608840 Oct 8 07:34 zabbix_agentd.log.1

-rw-r----- 1 zabbix zabbix 195453 Oct 10 15:56 zabbix_server.log

-rw-r -----1 zabbix zabbix 175864 Oct 10 15:56 zabbix_agentd.log

3. root@vuser-HVM-domU:/var/log/zabbix# cat zabbix_server.log

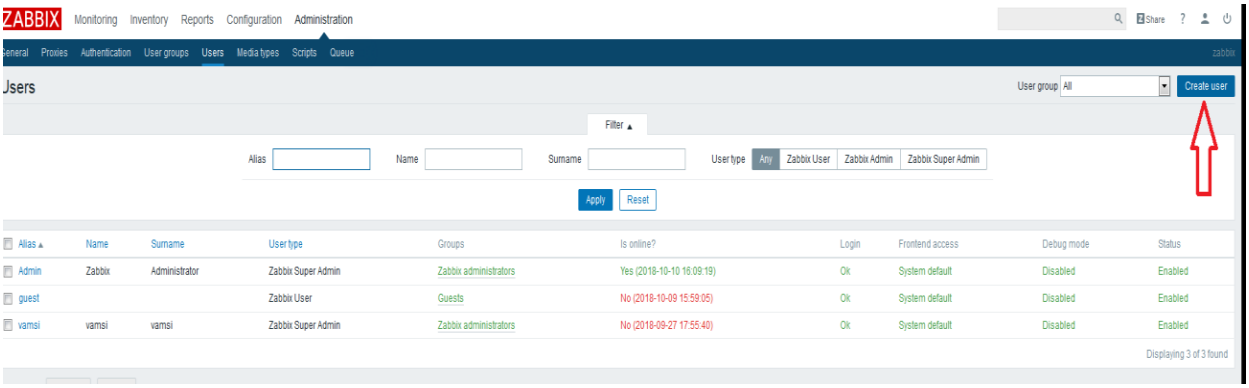
28551:20181010:152824.533 cannot send list of active checks to "127.0.0.1": host [Zabbix server] not found …….

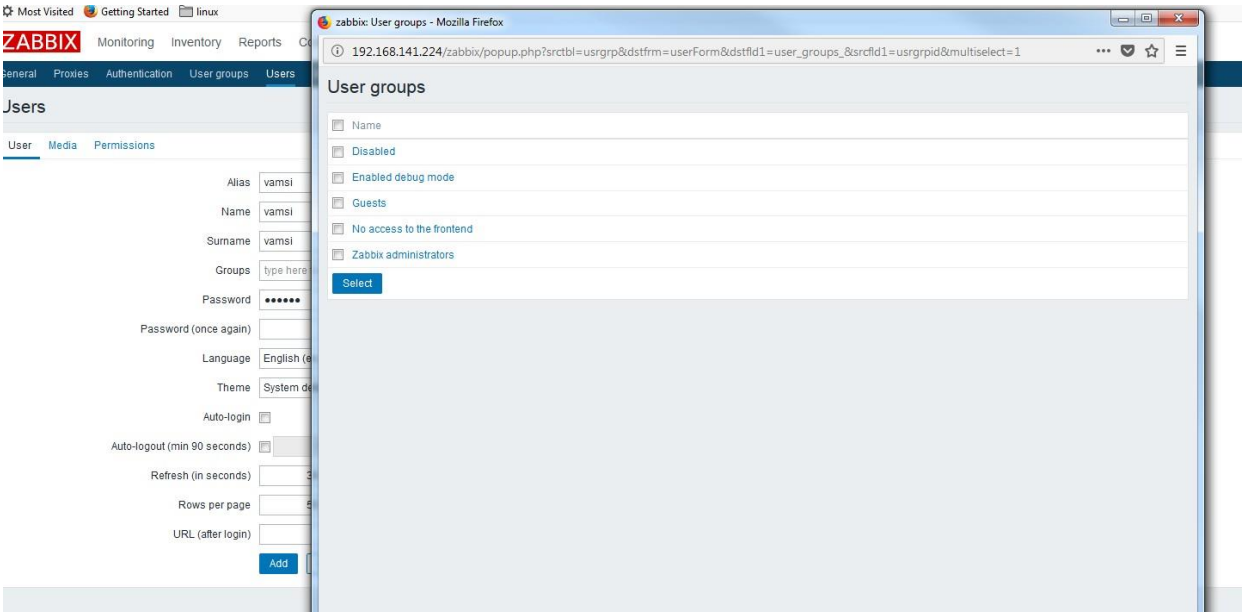
11. Check the zabbix agent logs

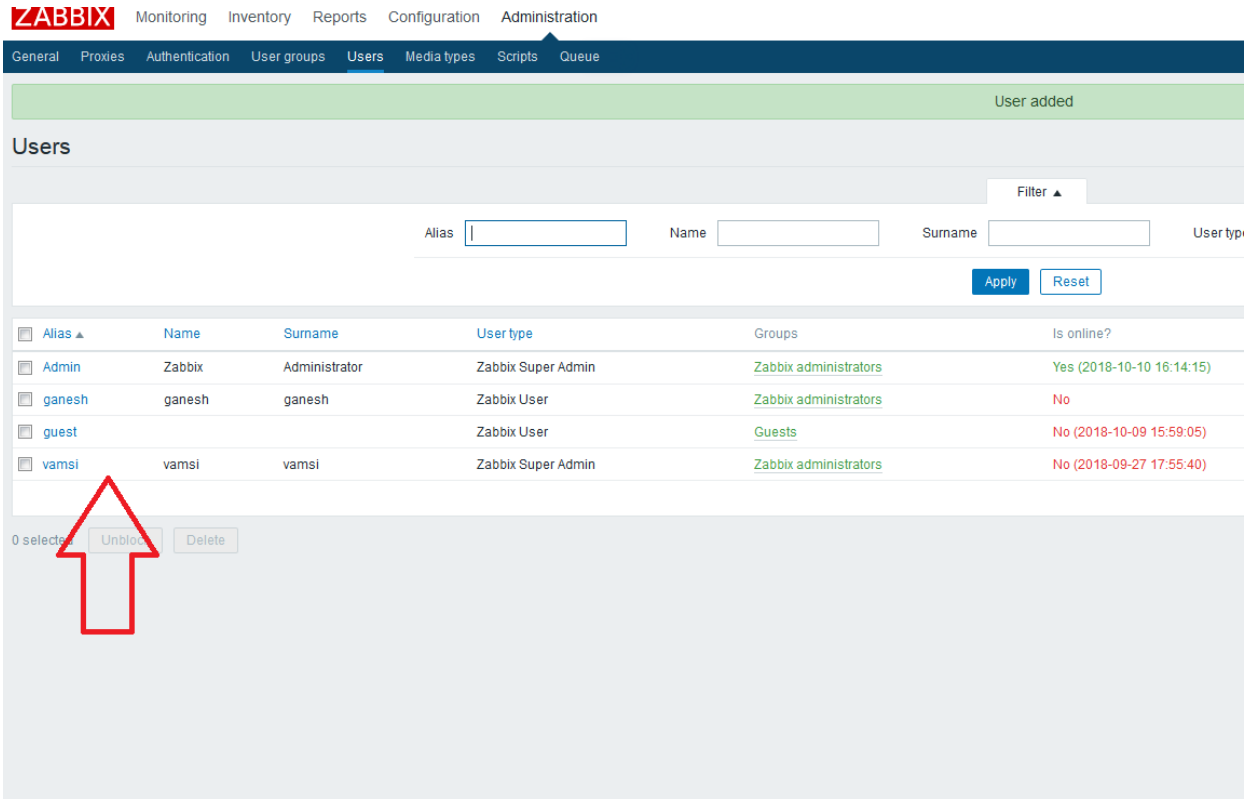
```
oot@vuser-HVM-domU:~# cd /var/log/zabbix/
oot@vuser-HVM-domU:/var/log/zabbix# pwd
var/log/zabbix
oot@vuser-HVM-domU:/var/log/zabbix# ls -ltr
otal 8
rw-rw-r-- 1 zabbix zabbix 4716 Oct  9 16:11 zabbix_agentd.log
oot@vuser-HVM-domU:/var/log/zabbix# cat zabbix_agentd.log
 23955:20181009:154925.260 Starting Zabbix Agent [Zabbix server]. Zabbix 3.2.11 (revision 76339).
 23955:20181009:154925.260 **** Enabled features ****
 23955:20181009:154925.260 IPv6 support:          YES
 23955:20181009:154925.260 TLS support:           YES
 23955:20181009:154925.260 **************************
 23955:20181009:154925.260 using configuration file: /etc/zabbix/zabbix_agentd.conf
 23955:20181009:154925.261 agent #0 started [main process]
 23956:20181009:154925.261 agent #1 started [collector]
 23957:20181009:154925.262 agent #2 started [listener #1]
 23958:20181009:154925.263 agent #3 started [listener #2]
 23959:20181009:154925.264 agent #4 started [listener #3]
 23960:20181009:154925.266 agent #5 started [active checks #1]
 23959:20181009:154925.269 invalid PSK in file "/etc/zabbix/zabbix_agentd.conf"
 23957:20181009:154925.269 invalid PSK in file "/etc/zabbix/zabbix_agentd.conf"
 23955:20181009:154925.270 One child process died (PID:23959,exitcode/signal:1). Exiting ...
 23960:20181009:154925.270 invalid PSK in file "/etc/zabbix/zabbix_agentd.conf"
 23958:20181009:154925.271 invalid PSK in file "/etc/zabbix/zabbix_agentd.conf"
abbix_agentd [23955]: Error waiting for process with PID 23959: [10] No child processes
 23955:20181009:154925.272 Zabbix Agent stopped. Zabbix 3.2.11 (revision 76339).
 24134:20181009:155303.132 Starting Zabbix Agent [Zabbix server]. Zabbix 3.2.11 (revision 76339).
 24134:20181009:155303.132 **** Enabled features ****
```

12. Create Users

1. Goto adminstration → users →create user

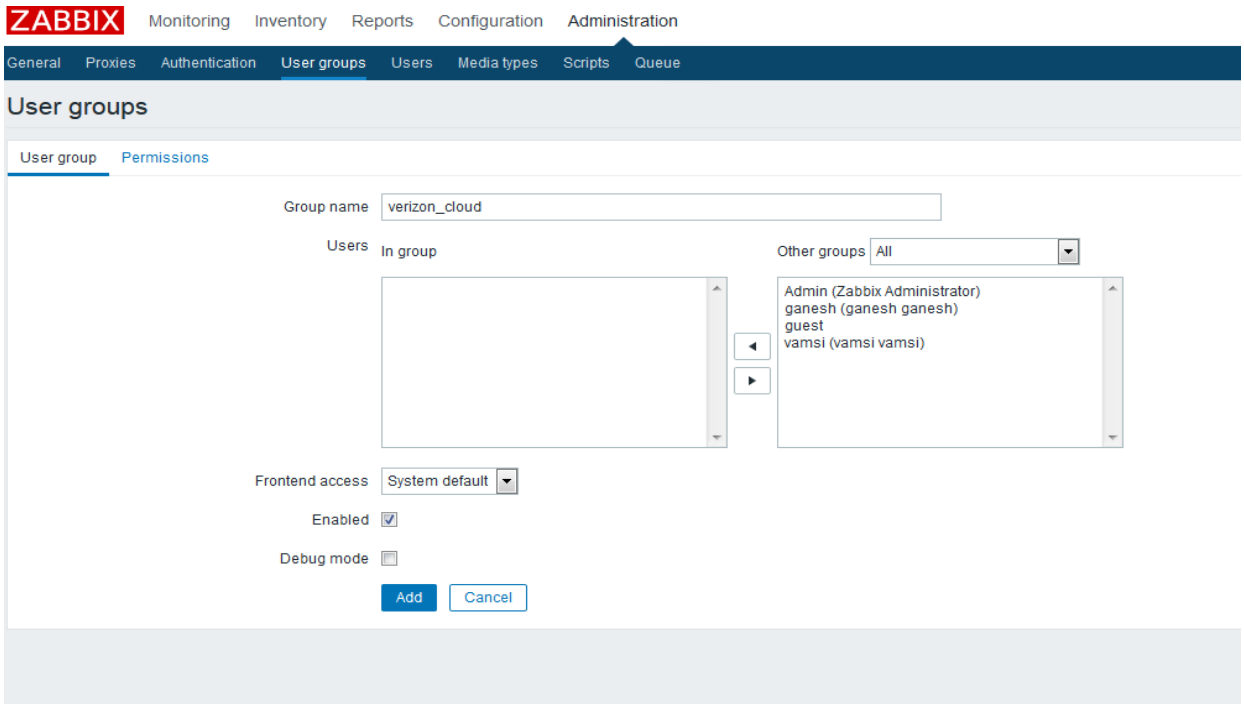


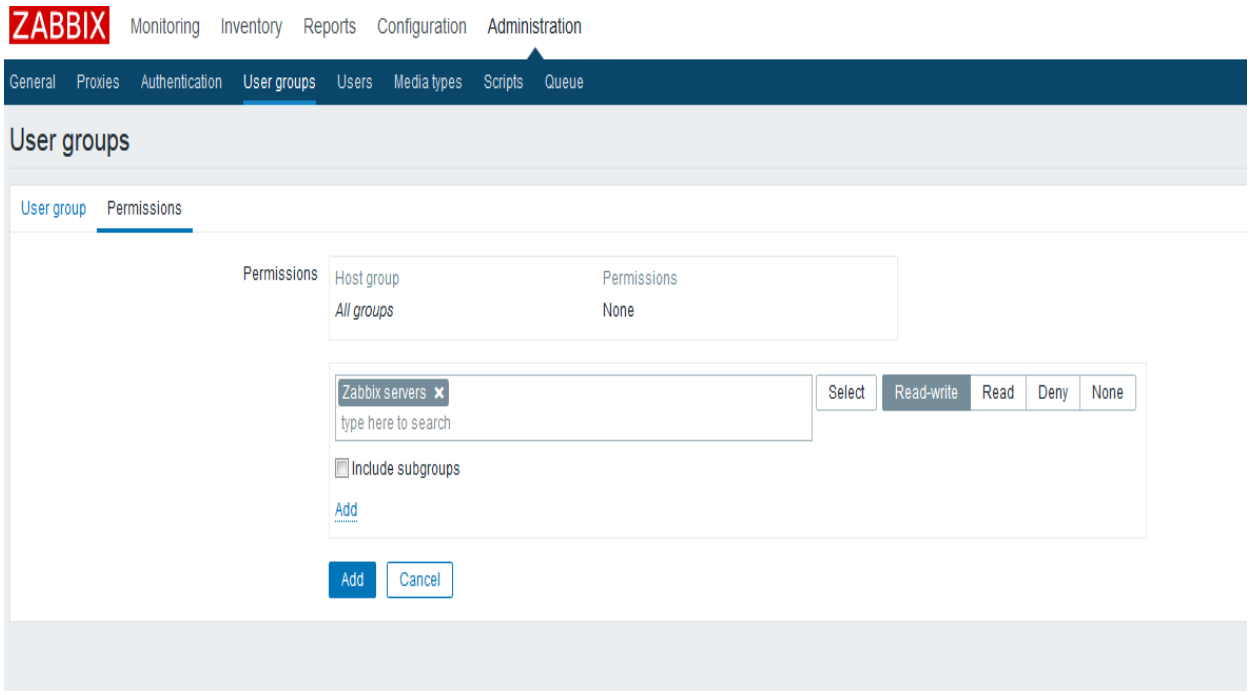




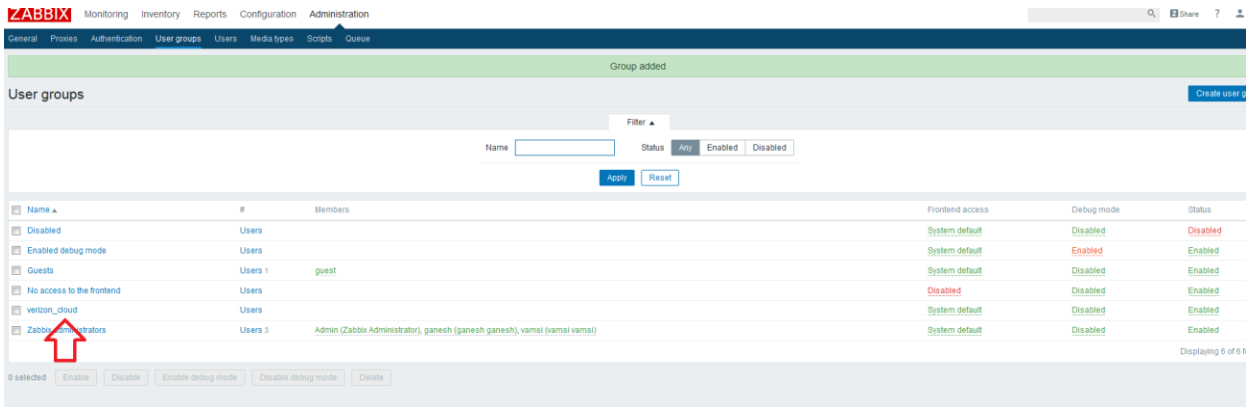
13. Create user group

Goto adminstration → User groups →create user group.





Then click add.



Topics : Screens, Slide Shows for easy monitoring.

Resources:
https://www.youtube.com/watch?v=sMOcujHk1Ig