Диагностика системы и решение проблем

Мониторинг: выявление проблем на серверах. Обзор систем мониторинга. Мониторинг с использованием Zabbix.

[Введение](#_c713xnuw3xem)

[Установка Zabbix](#_kf5ltzkr5m7m)

[Установка zabbix-server](#_kxeccrt9r0pn)

[Настраиваем БД](#_snnby1ntbjxq)

[Установка frontend](#_j6zmzbtoibwx)

[Настраиваем zabbix-server](#_b1c6hfrml8ip)

[Настраиваем frontend](#_f0sybdkxzhhf)

[Работаем с Zabbix](#_ezmijmn5h13n)

[Добавляем агента на другую машину](#_63yoxhupmwbe)

[Добавляем и настраиваем хост на zabbix-сервере](#_ez70058qdypt)

[Более подробно: hosts, items, triggers](#_wj5w6xbs581q)

[Домашнее задание](#_3tfrjxxltv85)

[Дополнительные материалы](#_y937sk8fclye)

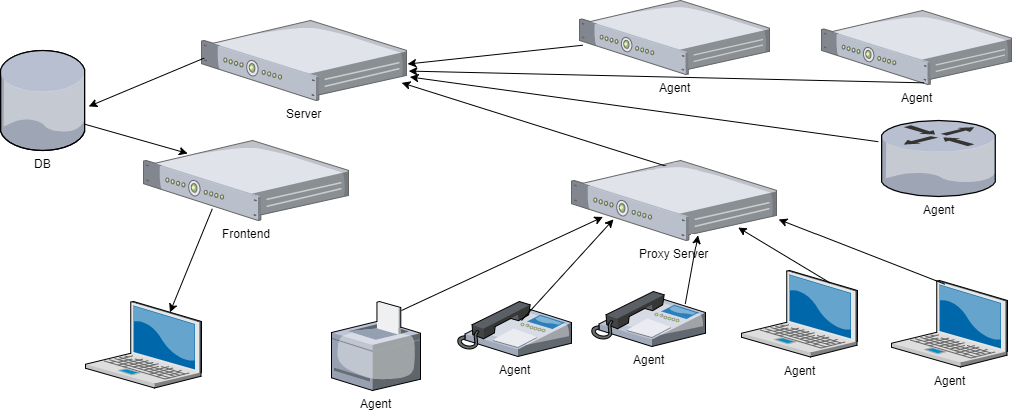
[Используемая литература](#_uvp6qax5r1ok)

# 

# Введение

Проблему лучше предупредить, чем бороться с ней, когда она уже появилась. На отдельной машине источником данных о проблемах (как уже возникших, так и потенциальных) является анализ логов. С системным логом и логами отдельных программ мы уже работали. Когда возникает проблема, связанная с работой нескольких машин через сеть, помогают сетевые утилиты (мы их рассматривали на занятии «Инструменты администрирования»). Если инфраструктура сети состоит из десятков и сотен машин, как серверов, так и иного оборудования, следующей важной задачей становится сбор и анализ данных. Здесь мы подходим к идее систем мониторинга. Такие системы строятся по клиент-серверной архитектуре и позволяют собирать данные от разных узлов (на которых работают агенты) и передавать их на сервер.

Независимо от используемой системы, архитектура систем мониторинга схожая. Выглядит она так:



Данные хранятся на сервере БД, куда складываются сервером мониторинга. Сервер мониторинга собирает данные от агентов. Это могут быть либо агенты системы мониторинга, установленные на машинах, так и встроенные SNMP-агенты (они есть в большинстве сетевых устройств). Если нет ни тех ни других, мы все равно можем проводить простейшие проверки – например с использованием протокола ICMP (доступен ли хост, открыт ли TCP-порт и т. д.)

SNMP (англ. Simple Network Management Protocol – простой протокол сетевого управления) – стандартный интернет-протокол для управления устройствами в IP-сетях. Он входит в состав стека TCP/IP и поддерживается большинством устройств (коммутаторами, маршрутизаторами, принтерами, RAID-массивами и т .д.). Работает с использованием протокола UDP, порт 161 используется агентом, 162 – менеджером.

Если агенты недоступны напрямую, может использоваться промежуточный сервер – прокси, который может опрашивать агентов от лица сервера.

Агенты могут проводить проверки двух типов, пассивные – когда сервер сам запрашивает нужную проверку, а агент отвечает, и активные – когда агент сам может отправлять серверу результаты.

Для доступа к данным используется фронтенд. В Zabbix он реализован на PHP и может использоваться вместе с Apache или nginx+php-fpm.

Наиболее известные системы мониторинга:

* Zabbix;
* Nagios;
* Monit;
* Icinga2;
* Prometheus;
* и другие, как бесплатные, так и платные.

Основная идея систем мониторинга схожая, но они различаются по своим возможностям и сложности освоения. Мы разберем систему мониторинга на примере Zabbix.

Zabbix изначально был создан для мониторинга в банковской среде, но сейчас это развитая и универсальная система мониторинга. Zabbix позволяет использовать активные и пассивные проверки, протокол SNMP, простейшие проверки, если для удаленной машины нет возможности использовать ни Zabbix-агент, ни SNMP-агент. Zabbix предоставляет большое число ключей для проверок, но также можно использовать и свои результаты (вернуть в качестве проверки результат выполнения скрипта и даже писать скрипты, которые сами сообщают Zabbix’у состояние в процессе выполнения).

Zabbix устанавливается на UNIX-подобные системы. Существуют пакеты для FreeBSD, Debian, Ubuntu, Red Hat. Для Windows серверного решения Zabbix нет: разработчики настаивают, что эта система недостаточно безопасна. Тем не менее для Windows есть агенты. Также на сайте есть готовые образы виртуальных машин с уже установленной операционной системой, zabbix-сервером и фронтендом. Кроме того, Zabbix доступен в исходных кодах.

Для Zabbix существует значительное число агентов для самых разнообразных систем, в том числе и для Windows: <https://www.zabbix.com/download>.

На сайте Zabbix есть подробная документация по использованию системы (<https://www.zabbix.com/manuals>), в том числе и на русском языке.

Zabbix использует свой протокол, который работает на TCP-портах – 10050 (агент) и 10051 (сервер)

В качестве СУБД поддерживаются MySQL, PostgreSQL, SQLIte, Oracle.

# Установка zabbix

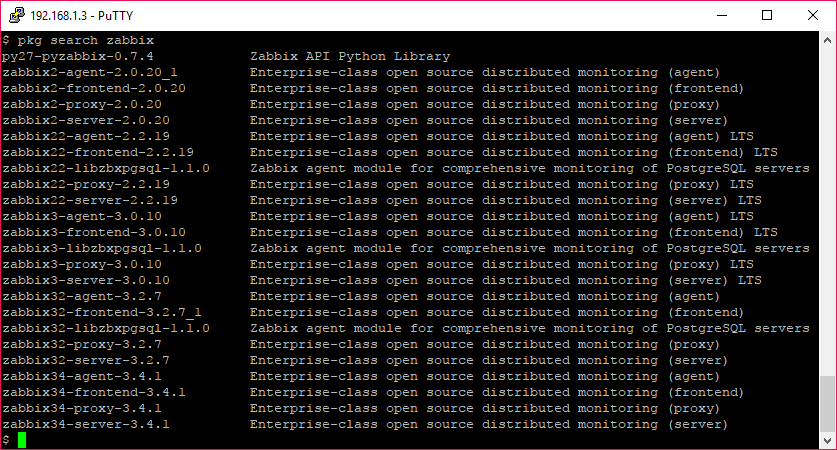
Чтобы установить из портов, найдем zabbix-server. Необходимо указать версию, может быть более новая.

|  |
| --- |
| whereis zabbix34-server |



Чтобы установить его через pkg, воспользуемся pkg search.

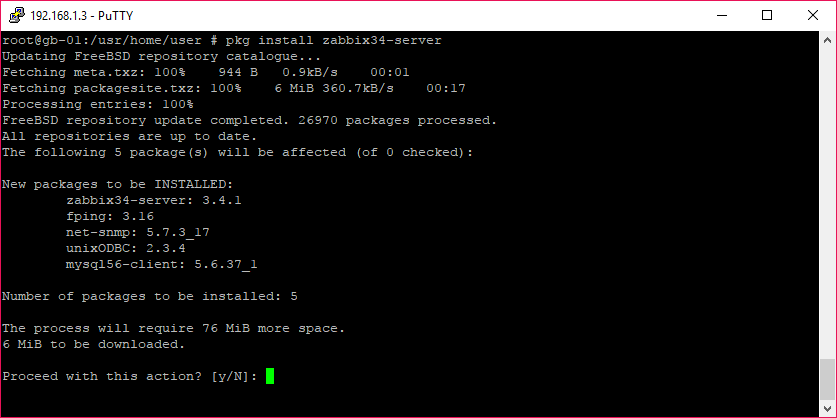
|  |
| --- |
| pkg search zabbix |



* zabbix34-server – непосредственно сервер;
* zabbix34-frontend – веб-интерфейс для сервера;
* zabbix34-agent – агент для мониторинга;
* zabbix32-proxy – прокси-сервер Zabbix, который позволяет серверу и агенту обмениваться информацией, если у сервера нет прямого доступа к агенту.

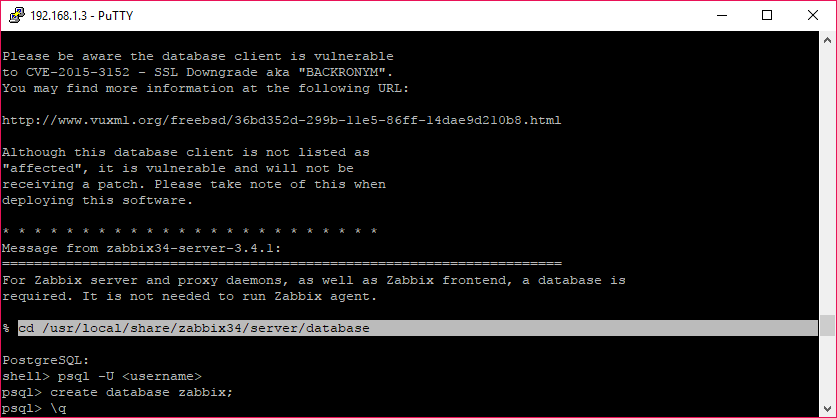
## Установка zabbix-server

Мы установим его из пакетов:

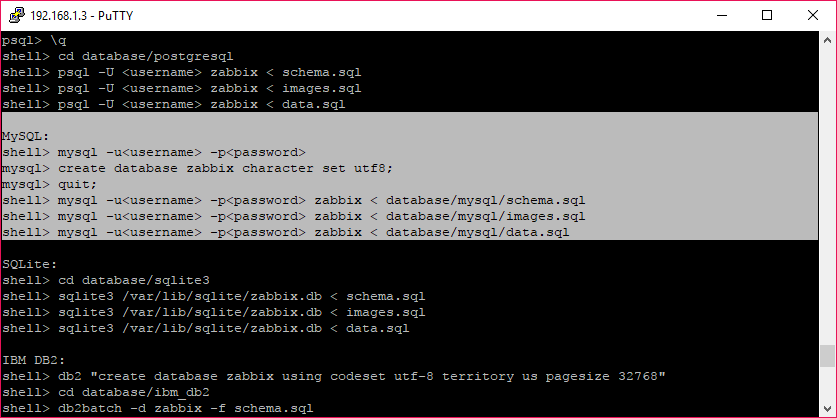


* zabbix-server – собственно сервер;
* fping – утилита для выполнения ping, которую использует Zabbix: она обладает большими возможностями (в т. ч. по проверке нескольких узлов по списку) по сравнению со стандартной утилитой;
* net-snmp – ПО для поддержки SNMP. SNMP поддерживается большинством устройств (коммутаторами, принтерами, RAID-массивами и т. д.) что позволяет контролировать даже то оборудование, на котором нет возможности поставить zabbix-agent;
* unixODBC – драйвер для работы с БД;
* по умолчанию zabbix-server собран с поддержкой mysql.

Далее нам нужно создать БД, для этого следует перейти в директорию:



и выполнить команды для используемой СУБД – их мы скопируем:



## Настраиваем БД

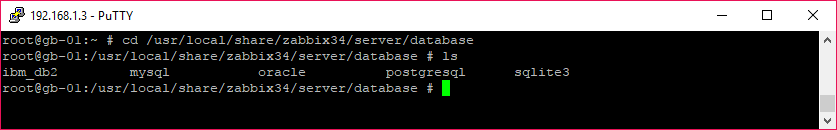
Сохраняем:

|  |
| --- |
| MySQL:  shell> mysql -u<username> -p<password>  mysql> create database zabbix character set utf8;  mysql> quit;  shell> mysql -u<username> -p<password> zabbix < database/mysql/schema.sql  shell> mysql -u<username> -p<password> zabbix < database/mysql/images.sql  shell> mysql -u<username> -p<password> zabbix < database/mysql/data.sql |

Далее переходим:

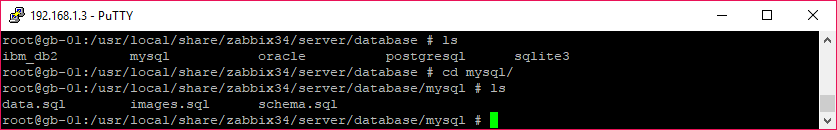
|  |
| --- |
| cd /usr/local/share/zabbix34/server/database |

Можно посмотреть содержимое директории. Здесь мы видим скрипты для разных БД:



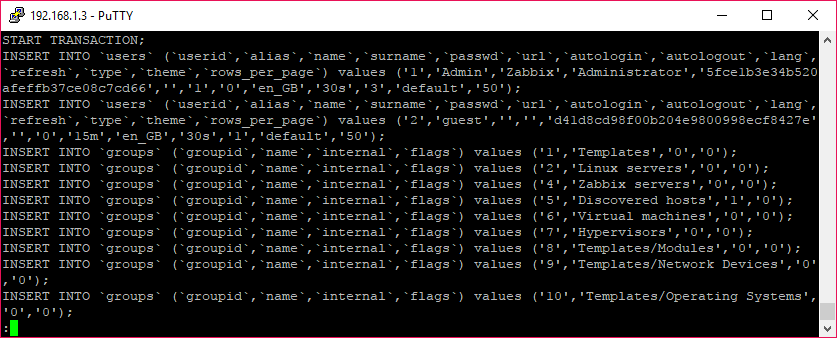
Перейдем в mysql.

|  |
| --- |
| cd mysql  ls |



Можно посмотреть скрипты:

|  |
| --- |
| less data.sql |

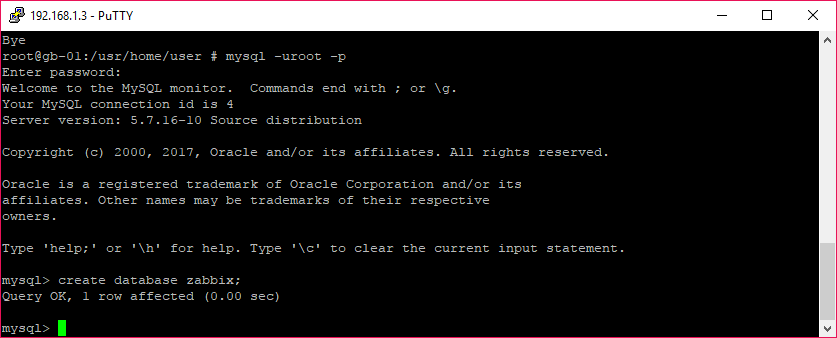


Подключаемся к mysql:

|  |
| --- |
| mysql -uroot -p |

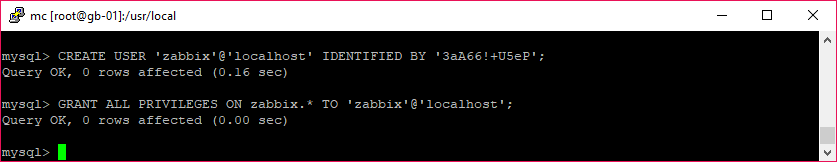
Создаем базу данных:

|  |
| --- |
| create database zabbix; |



Создадим пользователя для zabbix:

|  |
| --- |
| CREATE USER 'zabbix'@'localhost' IDENTIFIED BY '3aA66!+U5eP';  GRANT ALL PRIVILEGES ON zabbix.\* TO 'zabbix'@'localhost'; |



Выходим из клиента:

|  |
| --- |
| quit |

Убедитесь, что вы находитесь в директории с sql-скриптами (ls), иначе перейдите:

|  |
| --- |
| mysql |

Далее необходимо импортировать схему, images и данные в том порядке, в котором было указано при установке:

|  |
| --- |
| mysql -uroot -p zabbix <schema.sql  mysql -uroot -p zabbix <images.sql  mysql -uroot -pzabbix <data.sql |

либо

|  |
| --- |
| mysql -uroot -p***z1!\@Bx76+Af*** zabbix <schema.sql  mysql -uroot -p***z1!\@Bx76+Af*** zabbix <images.sql  mysql -uroot -p***z1!\@Bx76+Af*** zabbix <data.sql |

(пароль надо использовать свой).

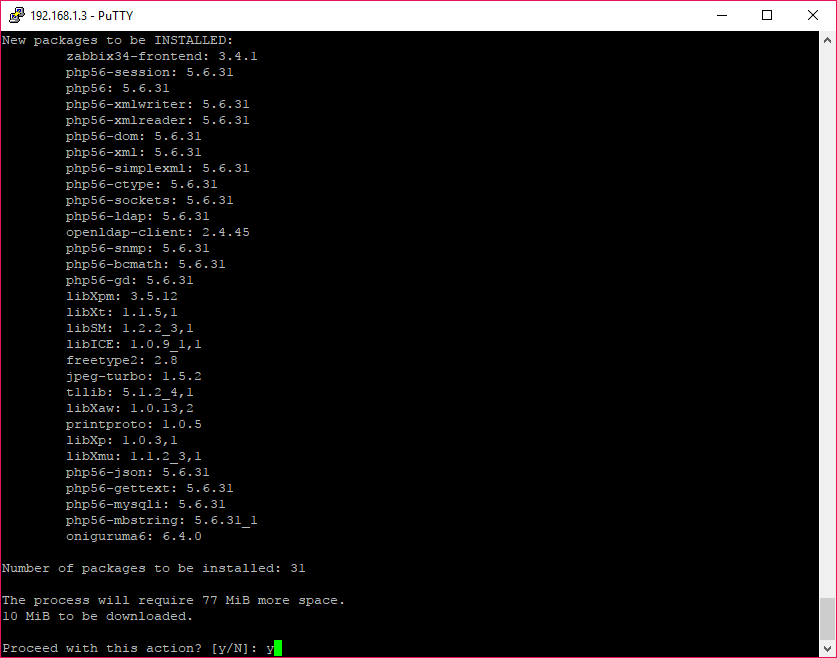
* -u – имя пользователя;
* -p – в первом варианте мы указываем, что нужно спросить пароль. Во втором мы указали пароль сразу, без пробела, например -p***m!PaS3w0RD***, если пароль – m!PaS3w0RD;
* zabbix – имя БД.

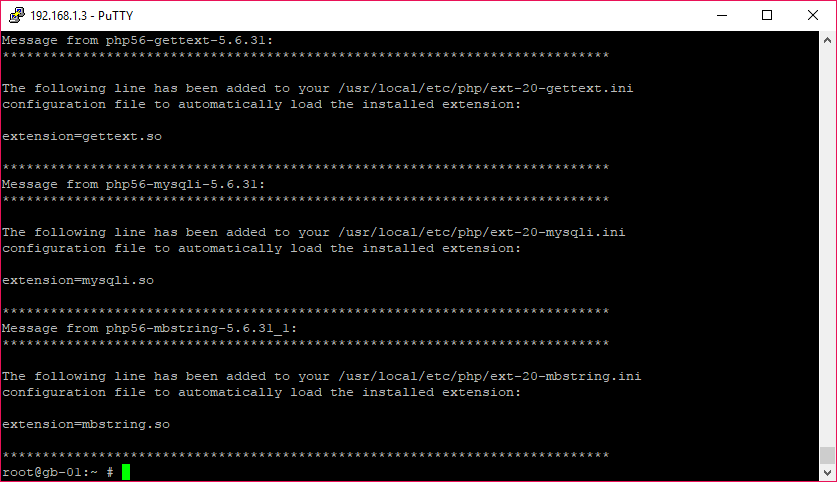
## Установка frontend

Дальше установим frontend.

Для нашей версии:

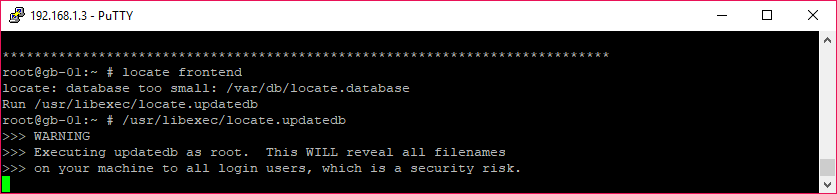
|  |
| --- |
| pkg install zabbix34-frontend |

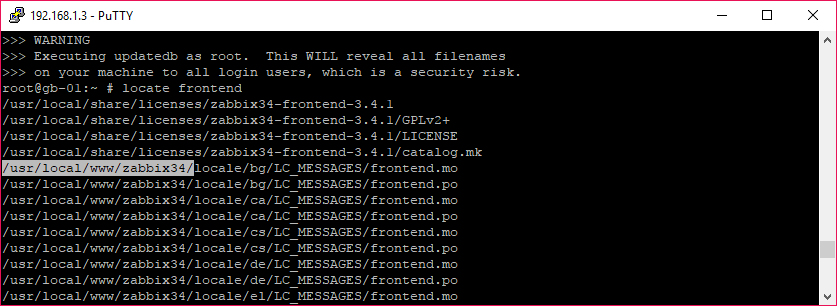




Теперь нужно настроить nginx.

Для начала найдем, где хранятся скрипты фронтенда (он написан на php).





Заменяем в /usr/local/etc/nginx/nginx.conf root:

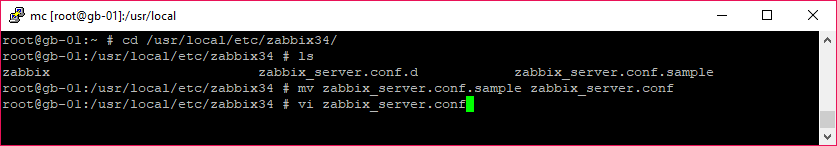
|  |
| --- |
| root /usr/local/www/zabbix34; |

Проверяем:

|  |
| --- |
| service nginx configtest |

Если все правильно, переходим к следующему этапу.

## Настраиваем zabbix-server



Исправим

|  |
| --- |
| LogFile=/tmp/zabbix\_server.log |

на

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

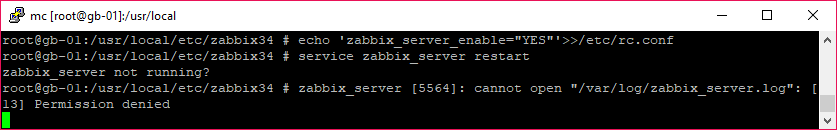
Раскомментируем и добавим пароль:

|  |
| --- |
| # DBPassword= |

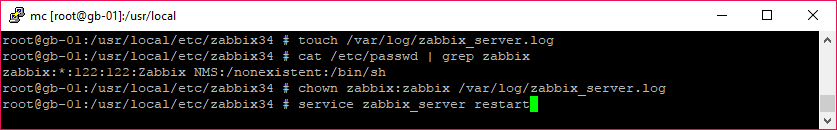
Для нашего случая:

|  |
| --- |
| DBPassword=3aA66!+U5eP |

Сохраняем и запускаем сервер

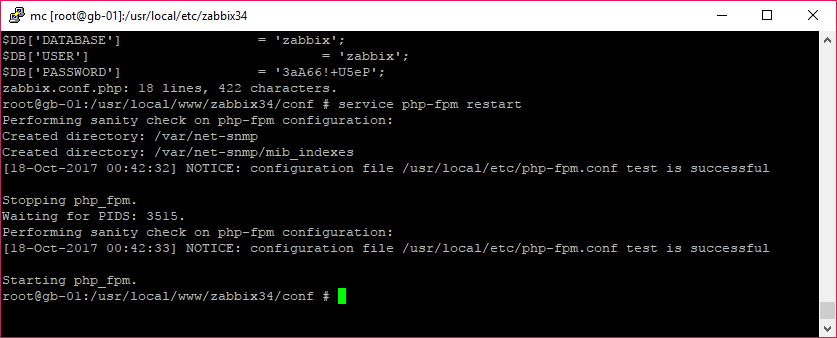


Файл необходимо создать и присвоить ему права:

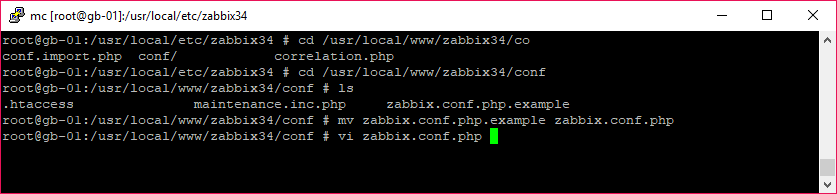


## Настраиваем frontend

Перезапустим php-fpm.



Найдем и создадим из примера zabbix.conf.php.



Аналогично надо прописать пользователя (он у нас по умолчанию zabbix) и пароль к БД.

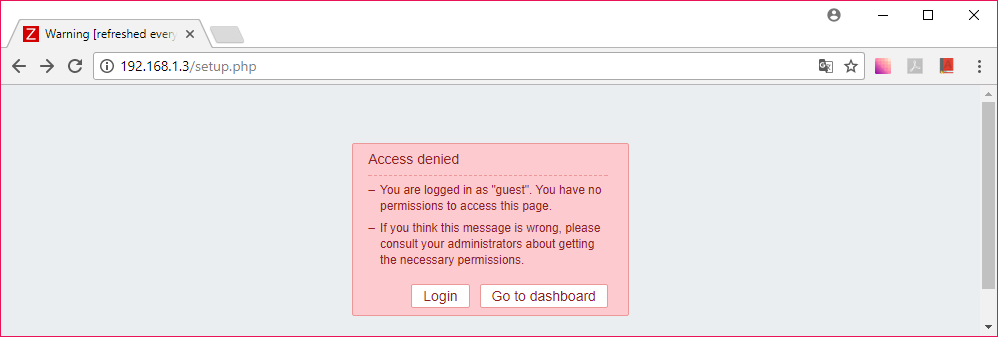
Было:

|  |
| --- |
| $DB['PASSWORD'] = ''; |

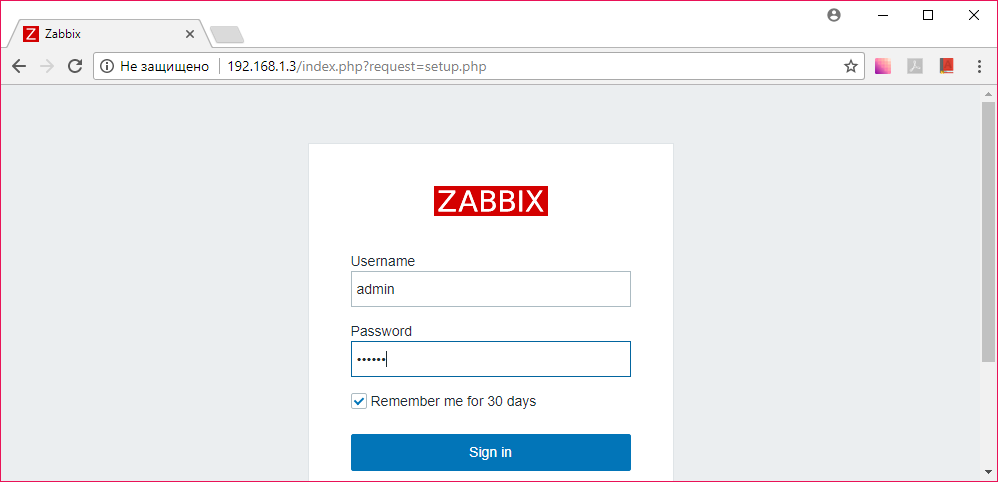
Стало:

|  |
| --- |
| $DB['PASSWORD'] = '3aA66!+U5eP'; |

Далее заходим через браузер:

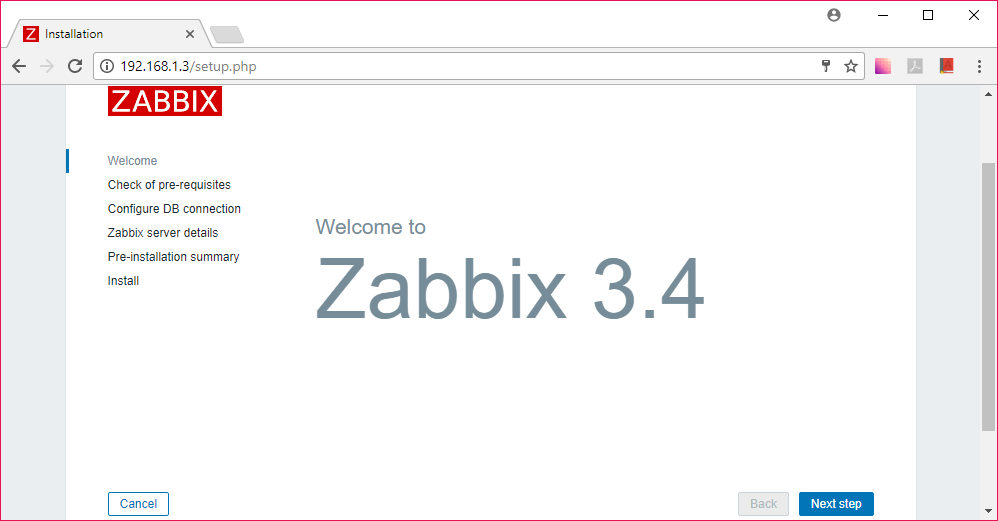


Система считает, что мы недостаточно залогинены:

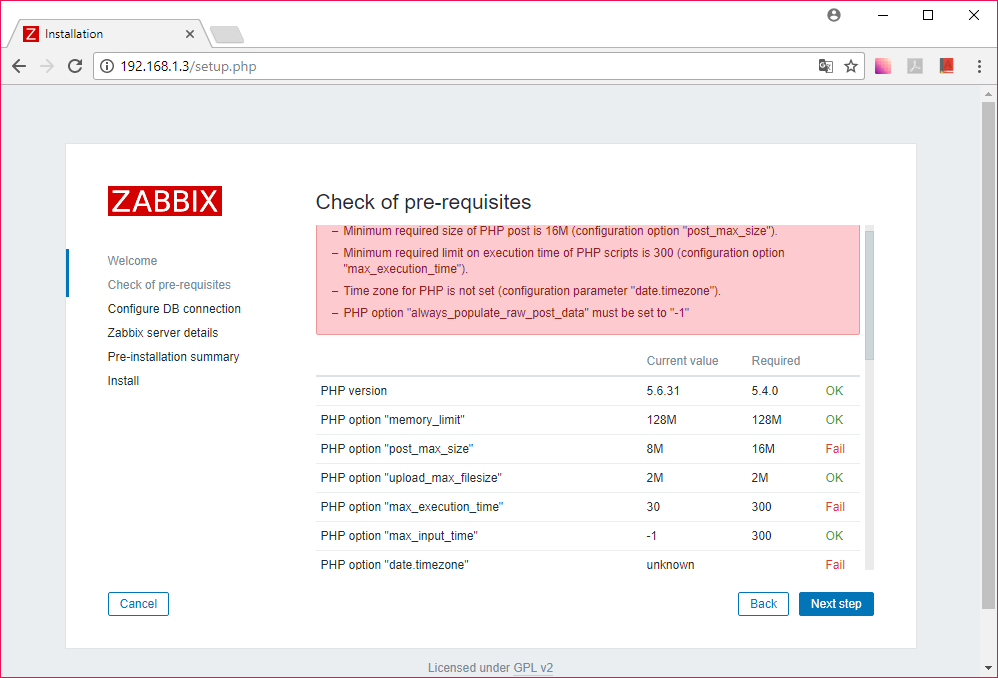


логин: admin,

пароль: zabbix.



Жмем **Next step.**



Надо исправить параметры на требуемые.

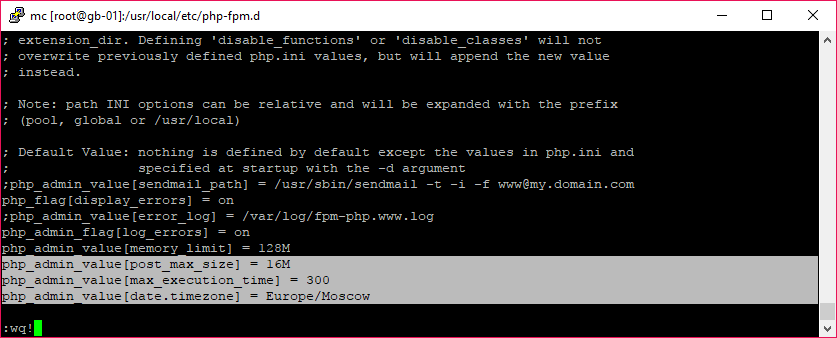
Для этого исправим php-fpm.conf:

|  |
| --- |
| vi /usr/local/etc/php-fpm.conf |

в нашем случае на:

|  |
| --- |
| vi /usr/local/etc/php-fpm.d/www.conf |

Добавляем:



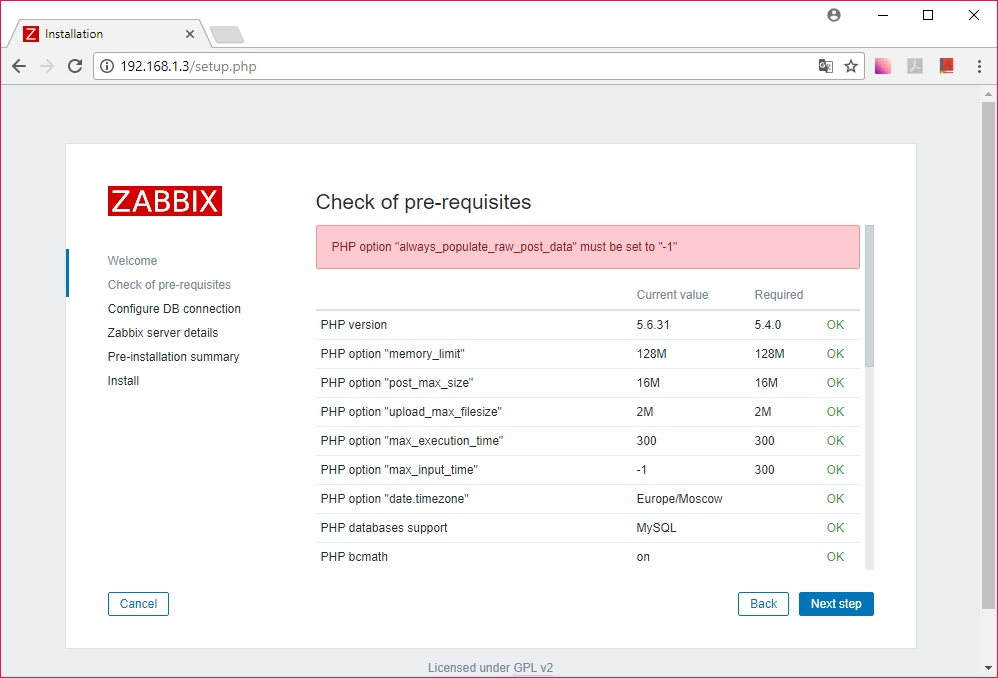
|  |
| --- |
| php\_admin\_value[post\_max\_size] = 16M  php\_admin\_value[max\_execution\_time] = 300  php\_admin\_value[date.timezone] = Europe/Moscow |

Сохраняем и перезапускаем:

|  |
| --- |
| service php-fpm restart |

Возвращаемся в браузер и жмем **Next.**

Далее:



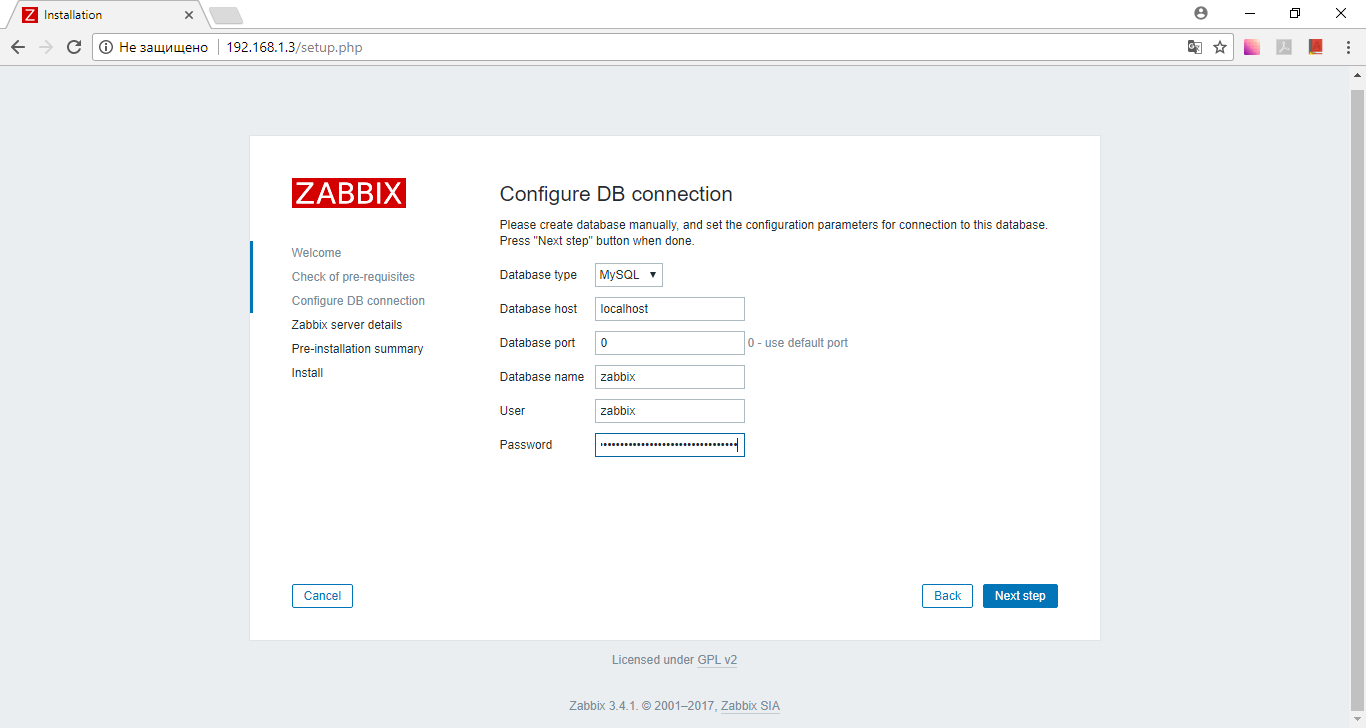
Также добавляем:

|  |
| --- |
| php\_admin\_value[always\_populate\_raw\_post\_data] = -1 |

Сохраняем и перезапускаем:

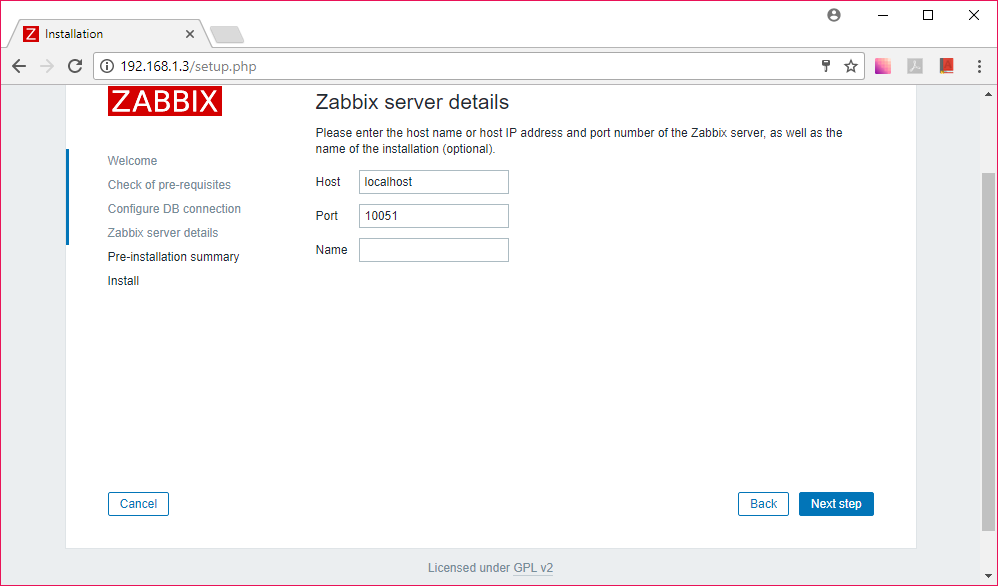
|  |
| --- |
| service php-fpm restart |

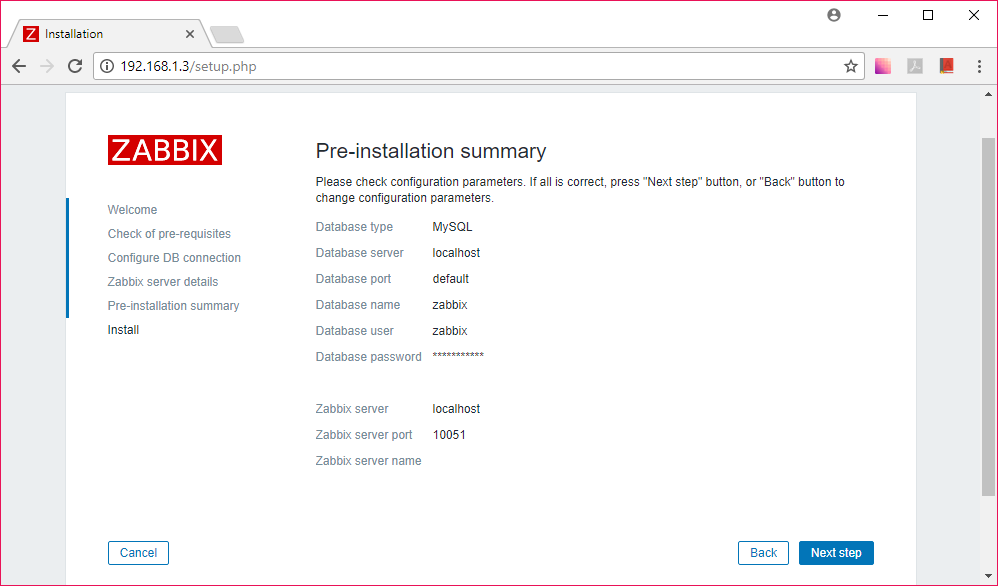
Возвращаемся в браузер и жмем **Next.**

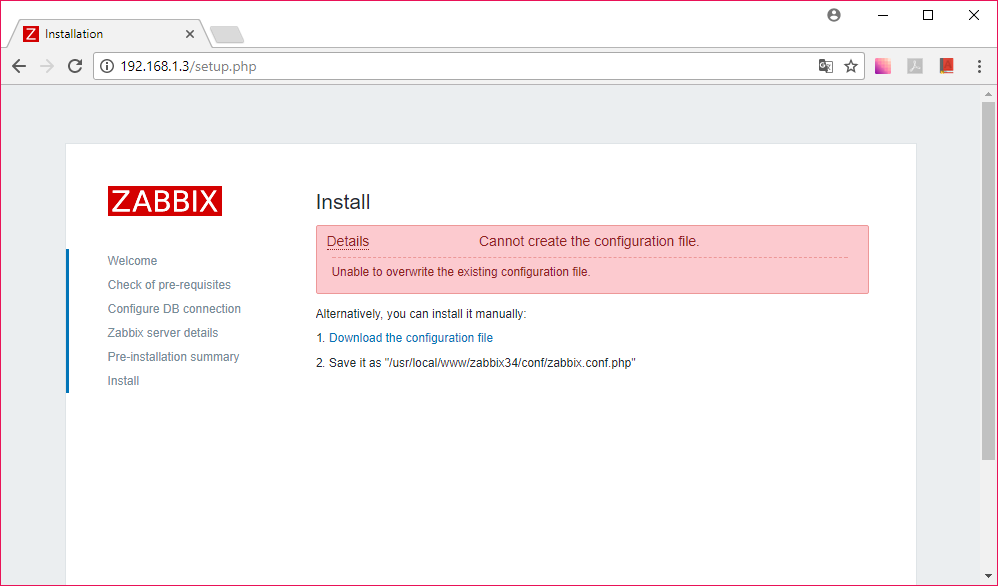


Еще раз укажем пароль для доступа к БД.

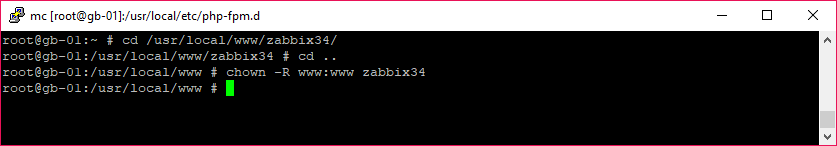
Далее:

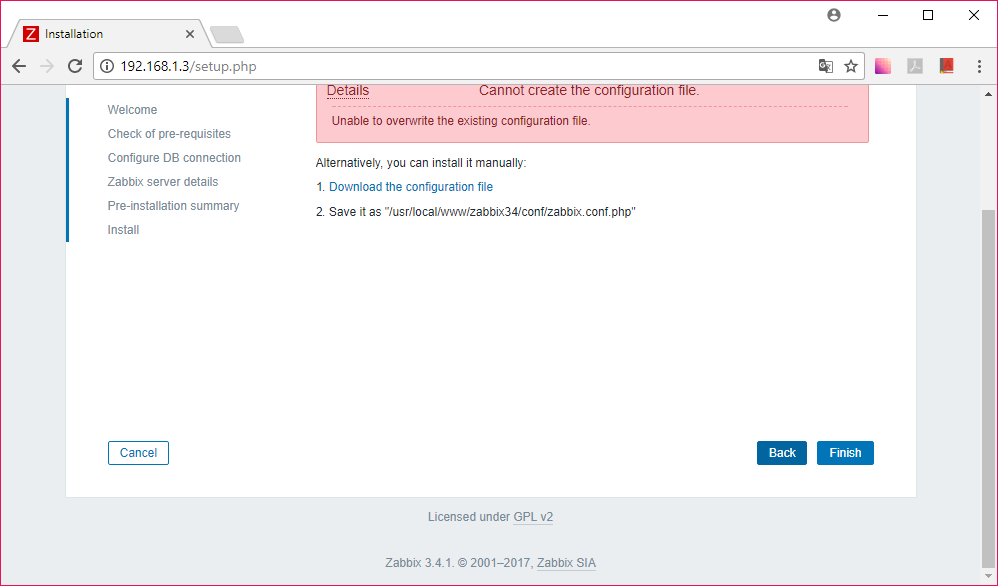




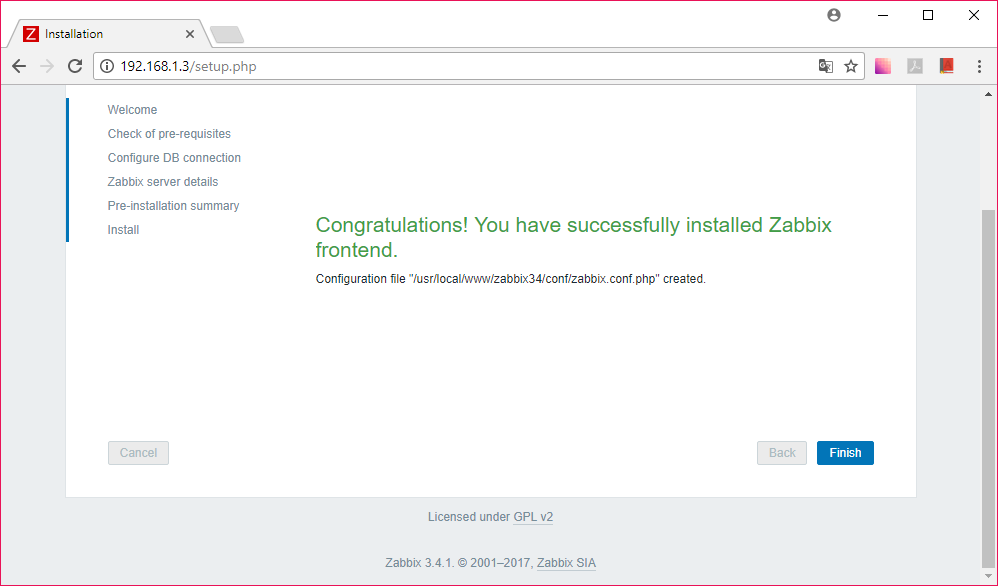


Необходимо поменять права:





Жмем **Back** и **Next Step:**



Заходим. Если система потребует логин и пароль, то логин по умолчанию Admin, пароль zabbix.

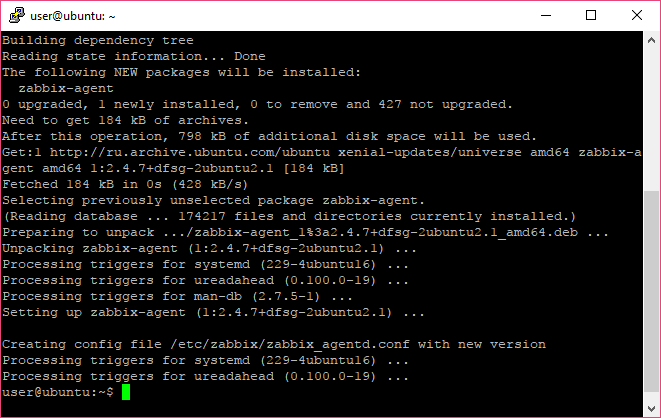
Вошли.

# Работаем с zabbix

## Добавляем агента на другую машину

Подключимся по ssh к машине с Ubuntu (в данном случае Ubuntu 16) и поставим zabbix-agent:

|  |
| --- |
| sudo apt update  sudo apt install zabbix-agent |



Отредактируем:

|  |
| --- |
| sudo vi /etc/zabbix/zabbix\_agentd.conf |

Заменим IP-адрес сервера (для пассивных проверок):

|  |
| --- |
| Server=127.0.0.1 |

на наш:

|  |
| --- |
| Server=192.168.1.3 |

Заменим IP-адрес сервера для активных проверок:

|  |
| --- |
| ServerActive=127.0.0.1 |

на:

|  |
| --- |
| ServerActive=192.168.1.3 |

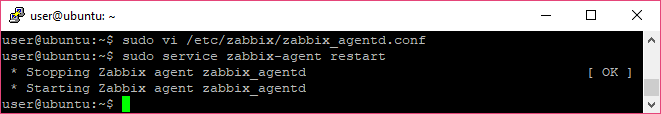
Указываем Hostname. Очень важно, что Hostname должно совпадать в конфигурационном файле и на сервере, иначе Zabbix не будет корректно работать с этим агентом.

|  |
| --- |
| Hostname=Zabbix server |

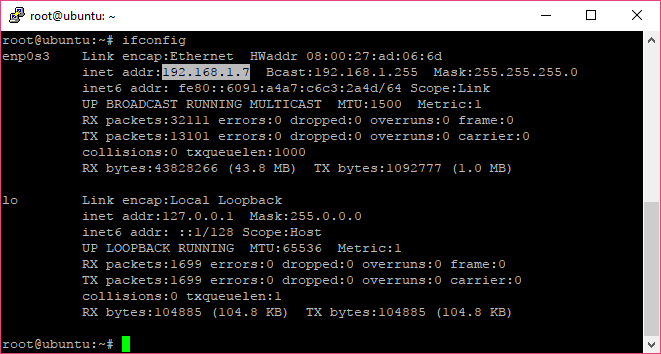
в данном случае Hostname – ubuntu, потому:

|  |
| --- |
| Hostname=ubuntu |

Сохраняем и перезапускаем:



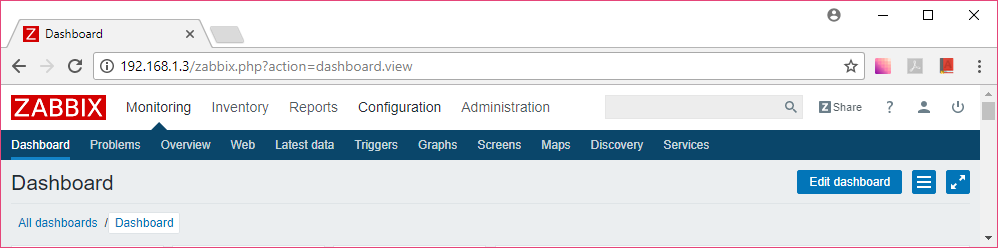
Также узнаем IP-адрес:



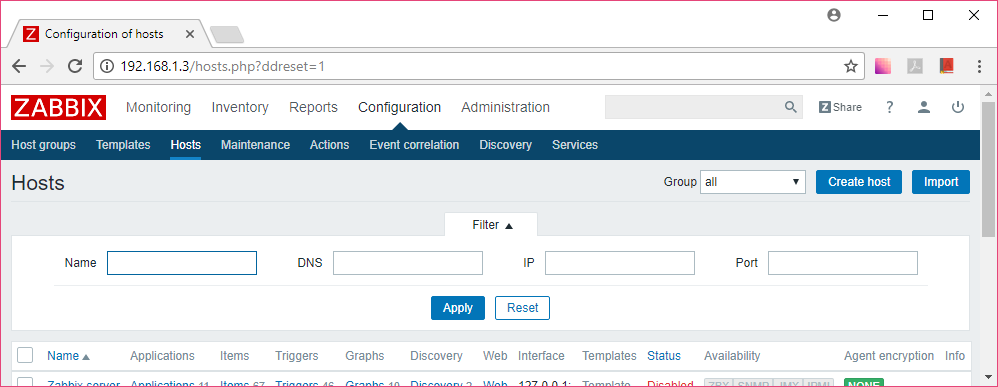
## Добавляем и настраиваем хост на zabbix-сервере

Теперь добавим хост на сервере.

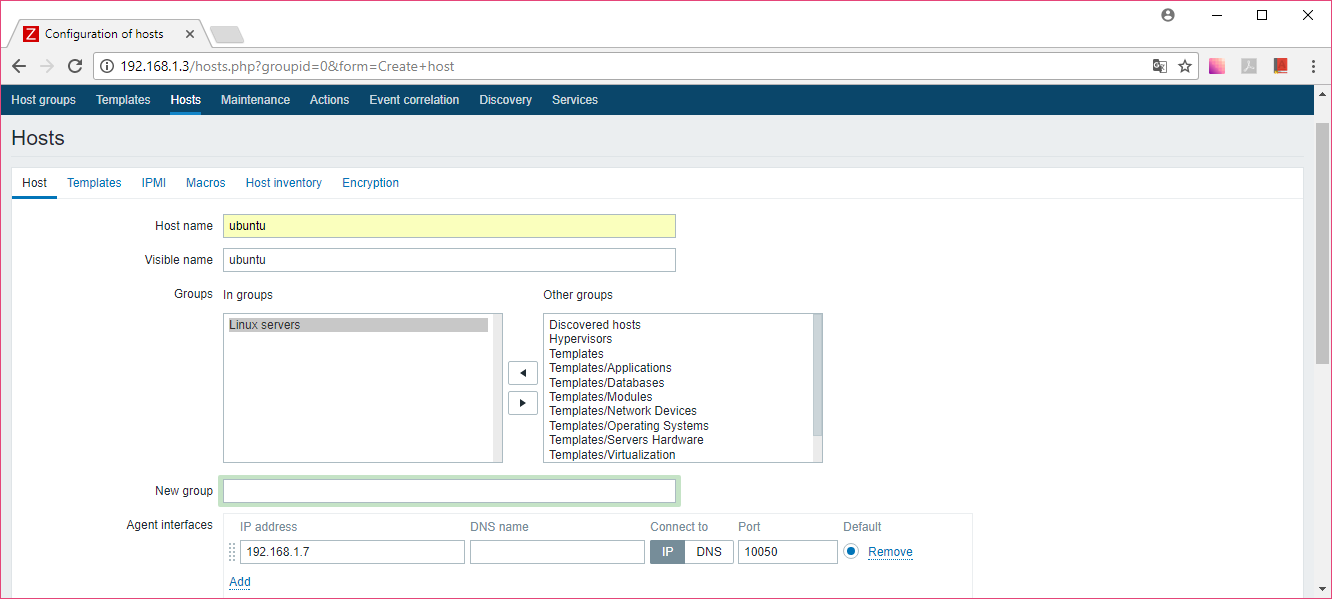
Заходим в веб-интерфейс, кликаем на **Configuration.**



Кликаем на **Hosts.**



Жмем на **Create host.**



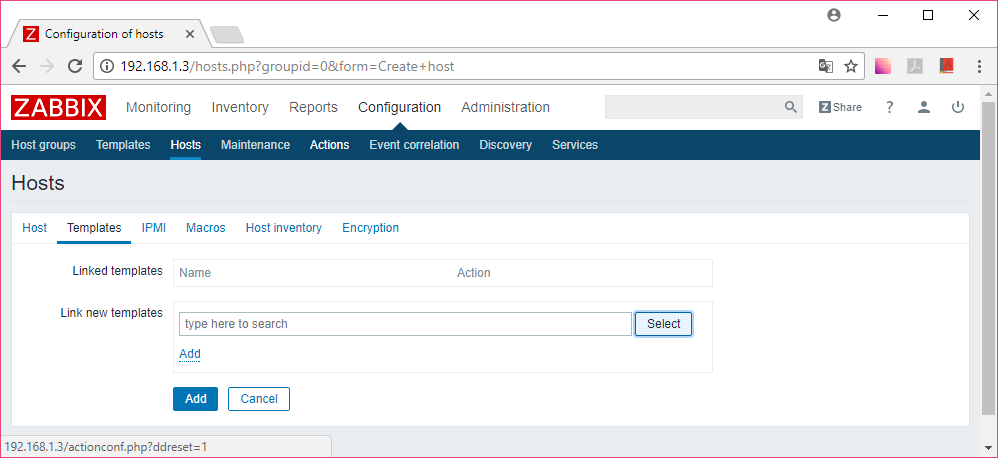
Указываем то же самое Host name, что и в конфиге агента. Также указываем имя, под которым он будет отображаться на сервере.

Добавляем группу (в данном случае Linux servers), или можно создать новую.

Также указываем IP-адрес или доменное имя, по которому мы будем работать с агентом. (Если нажать подчеркнутую ссылку **Add,** можно добавить несколько таких записей.)

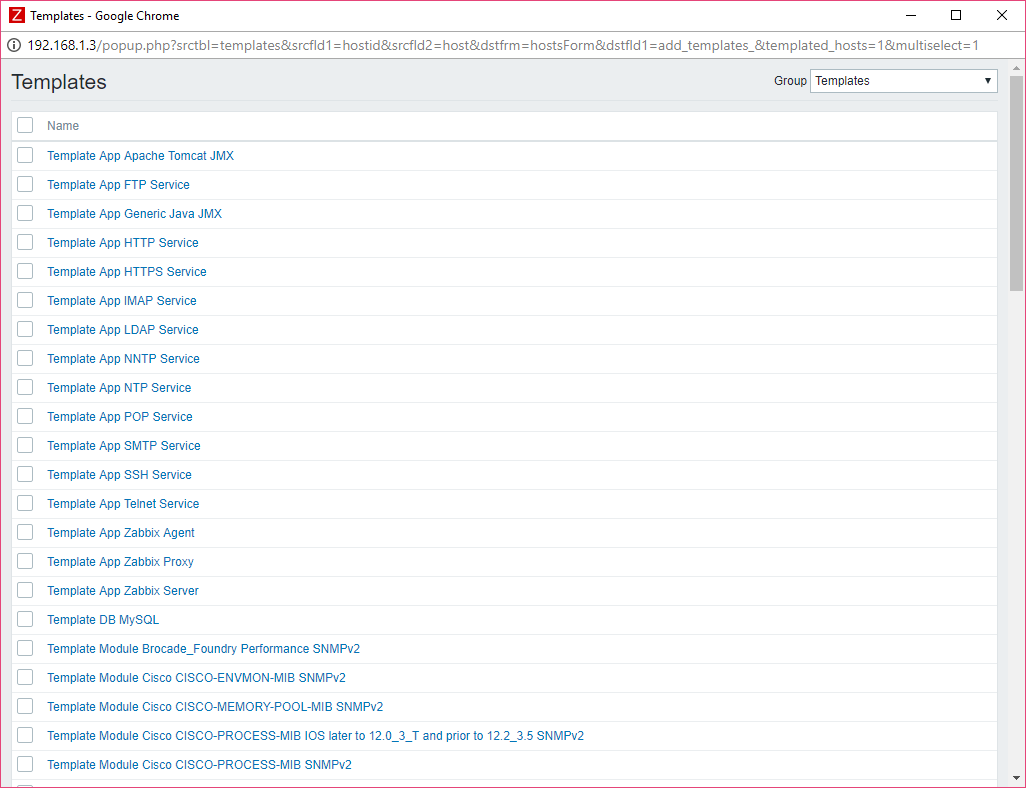
Если пролистнуть ниже, можно сохранить хост, нажав кнопку **Add,** но мы еще добавим шаблоны с проверками.

Для этого нажимаем вкладку **Templates:**

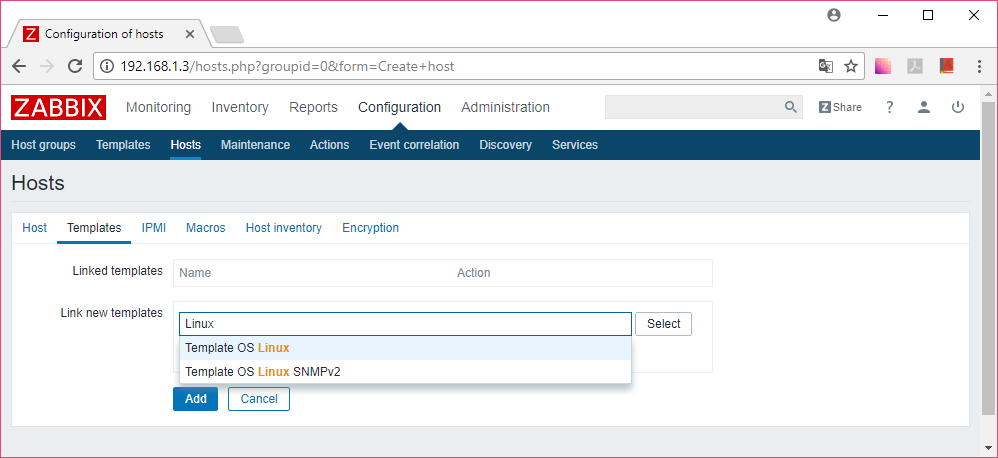


Можно нажать **Select** и выбрать из списка, а можно начать набирать, например, Linux.

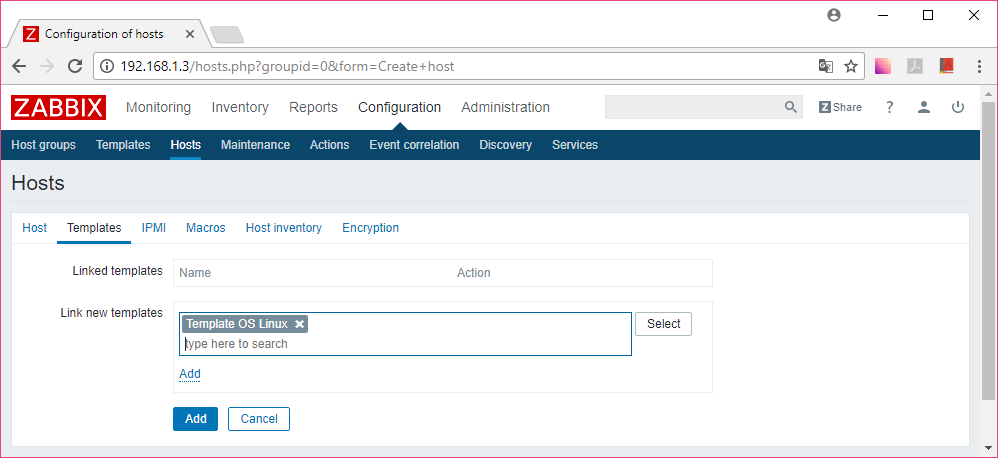
Весь список:



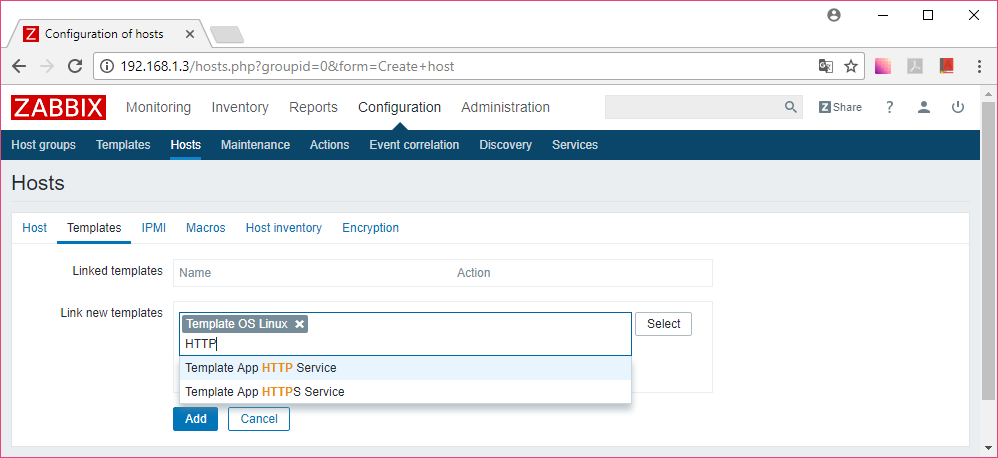
Либо начнем писать «Linux» и увидим:



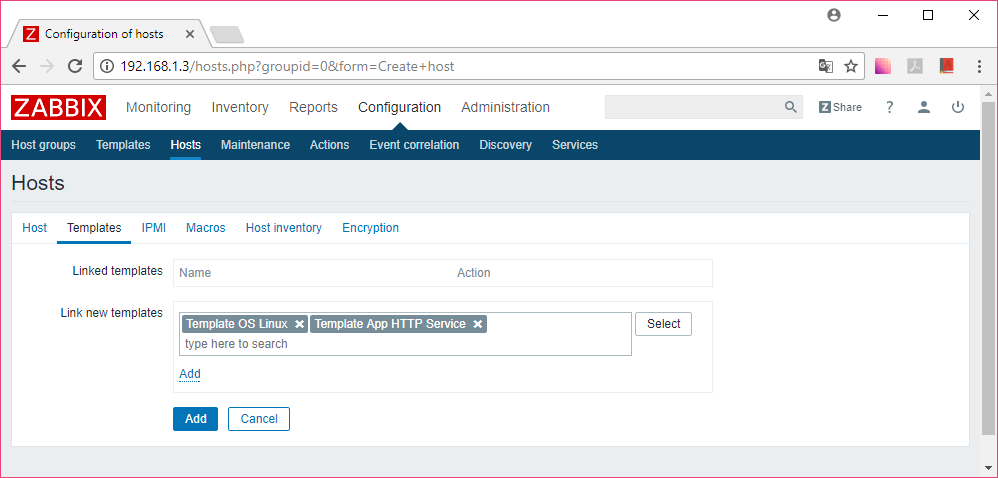
Выбираем Template OS Linux:



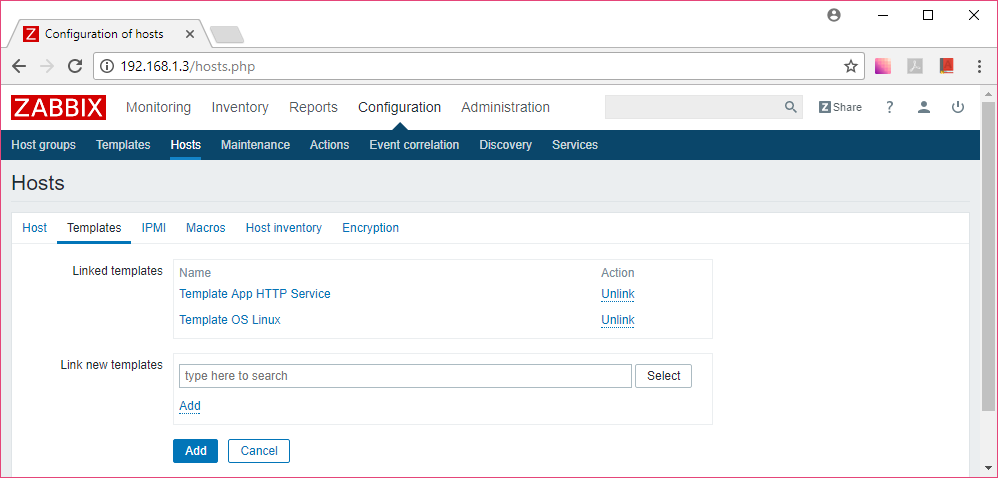
Продолжаем. Например, добавим HTTP:



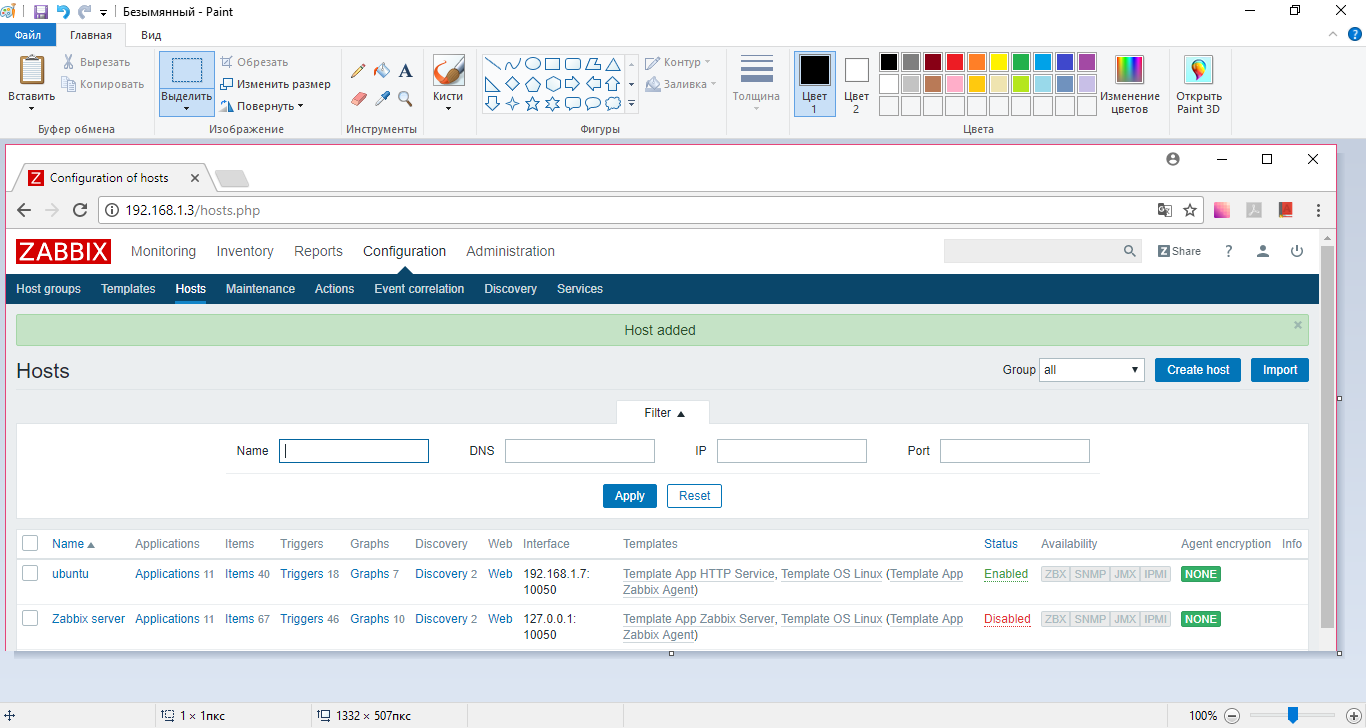
Выбираем (кликаем на нужную подсказку):



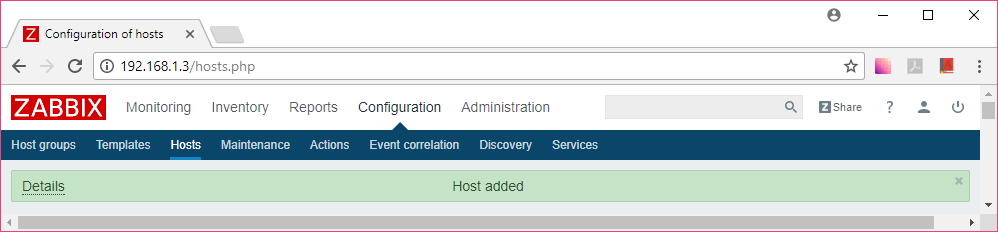
Жмем на подчеркнутую ссылку **Add:**



Жмем кнопку **Add:**

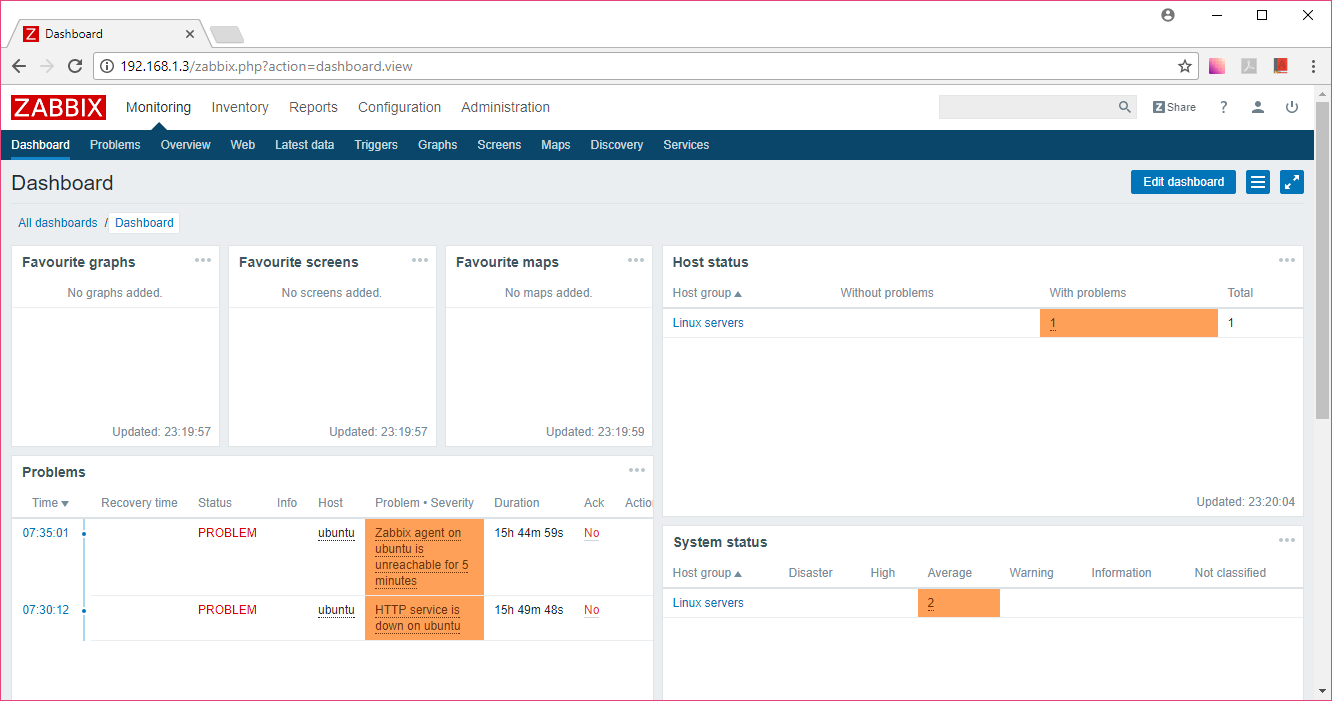


Нажимаем **Monitoring:**

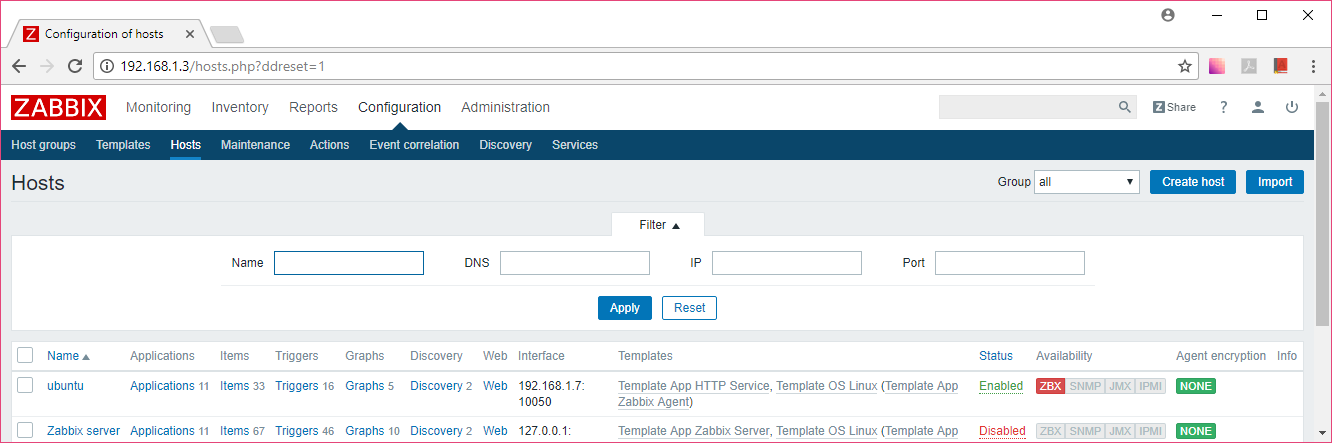
****

Жмем **Dashboard.**

Если бы мы не перезапустили zabbix-agent, то увидели бы такую картину:

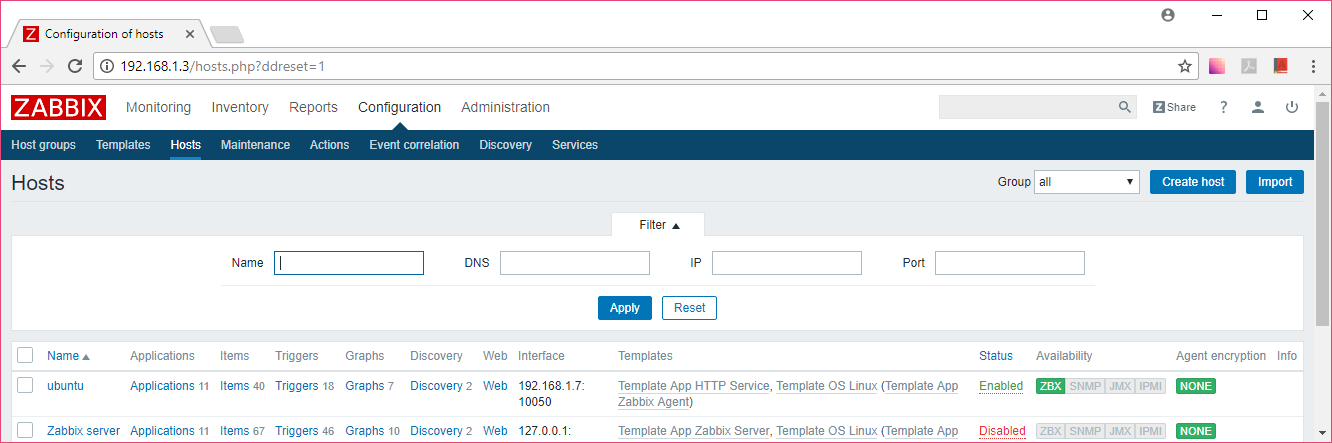


А если перейти в **Configuration,** а потом в **Hosts,** такую:



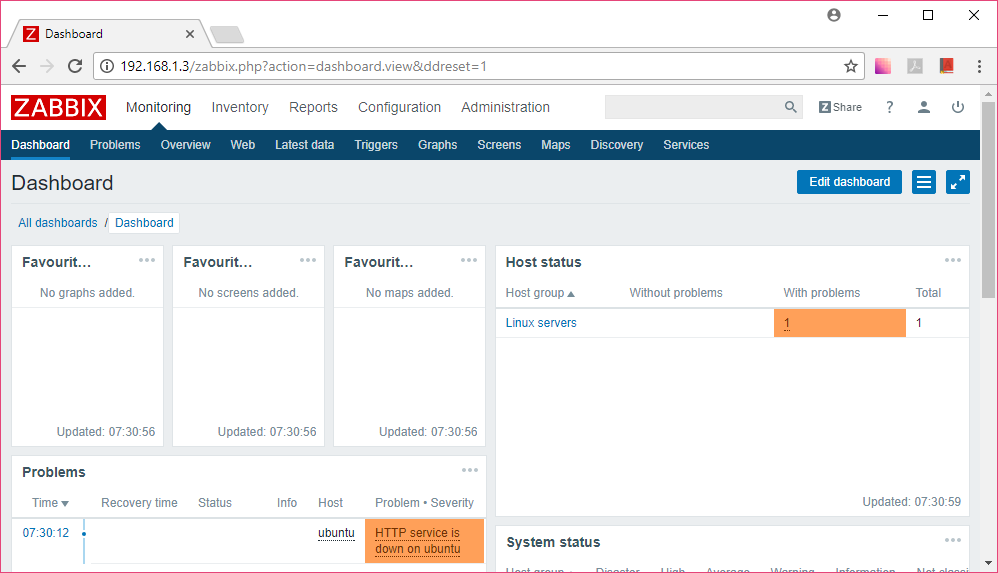
Это означает что агент не отвечает (неправильный IP-адрес, не разрешается доменное имя, если оно указано, сеть недоступна, либо неправильный IP-адрес указан в zabbix-agent, и он отвергает попытки нашего сервера подключиться к нему).

Если все верно, картинка будет такая:



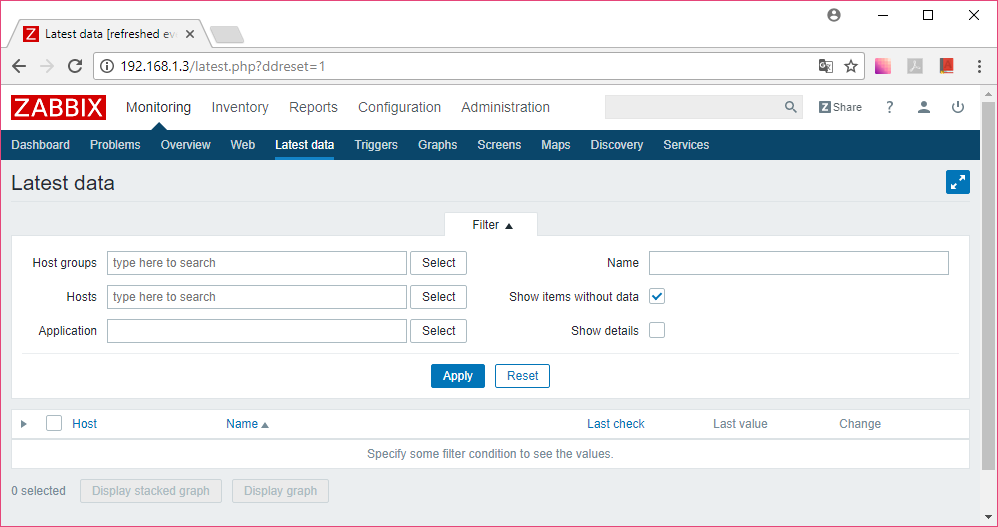
Предположим, мы все сделали правильно. Сначала настроили zabbix-agent и только потом создали host в веб-интерфейсе.

Перейдем в **Monitoring** и в **Dashboard.**

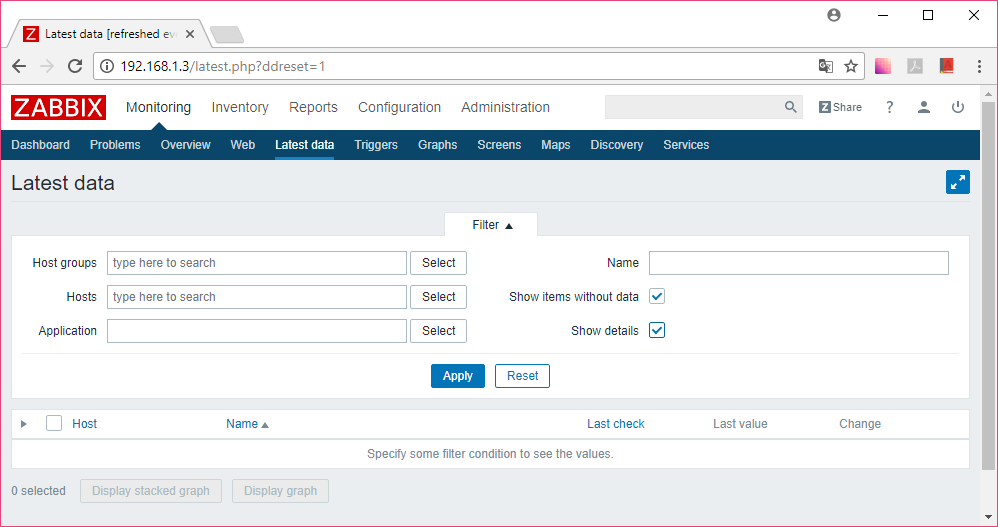


Видим, что данные уже начались собираться.

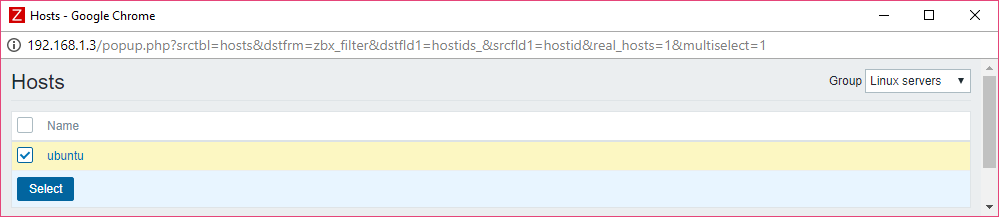
Далее нажмем **Latest Data:**

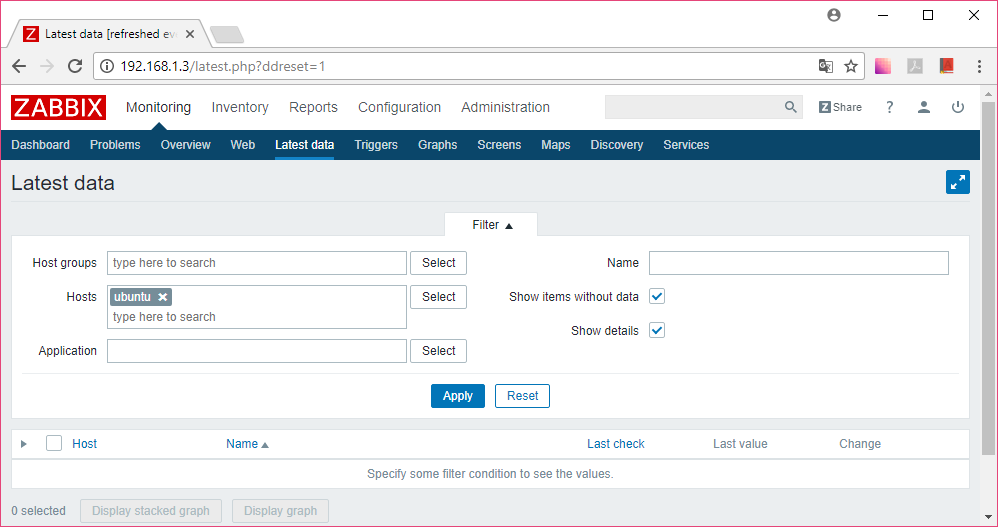


Установим **Show Details:**

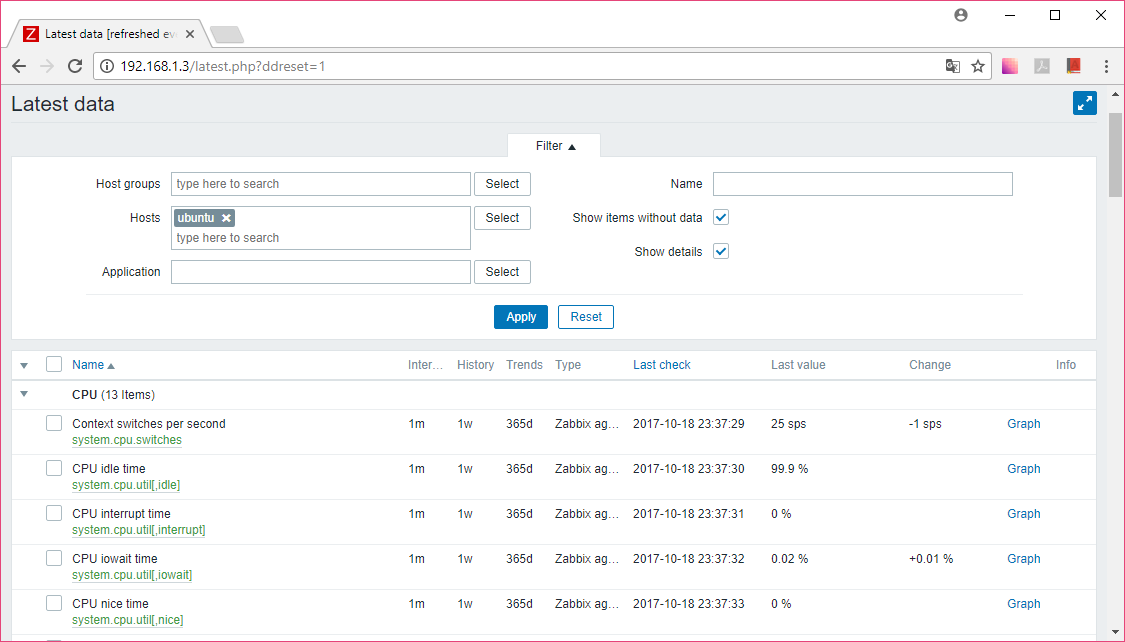


Далее выберем по хосту (**Host: Select**):

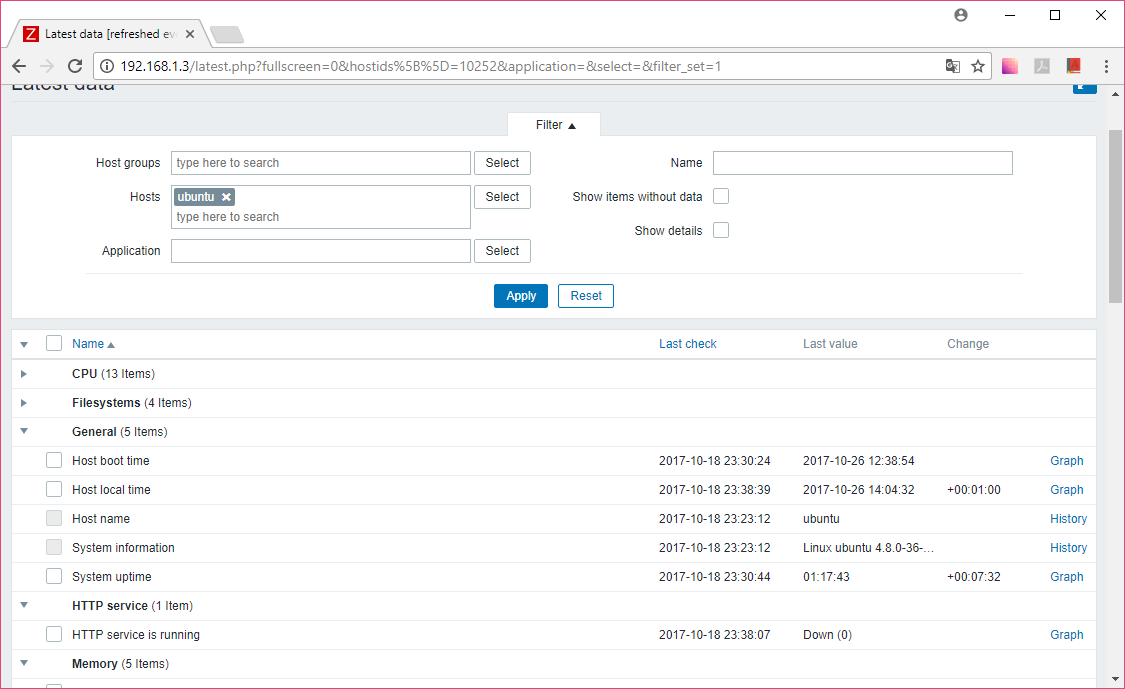




Нажмем **Apply:**



Уберем **Show Details** и **Показывать без даты,** снова **Apply:**

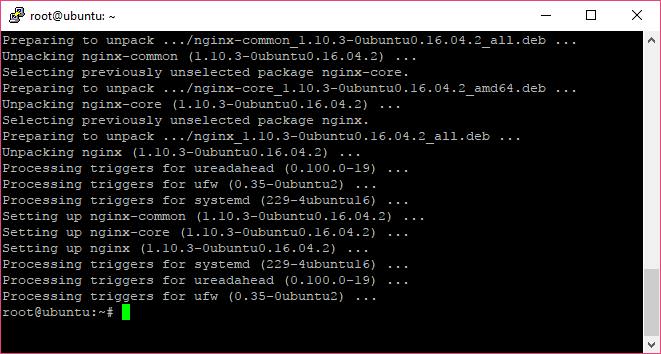


Группы items можно сворачивать и разворачивать.

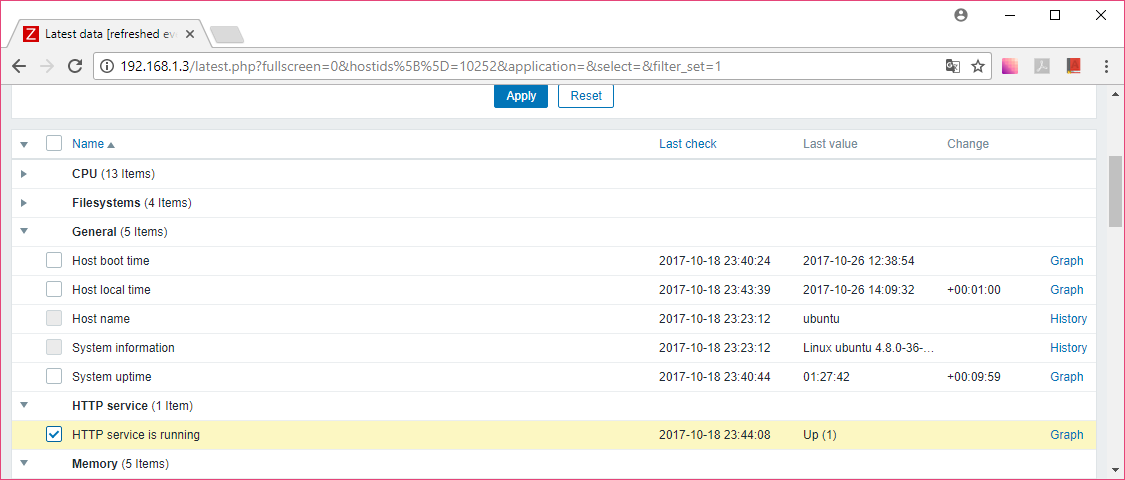
HTTP Server – down, так как он не установлен.

Установим nginx в ubuntu:

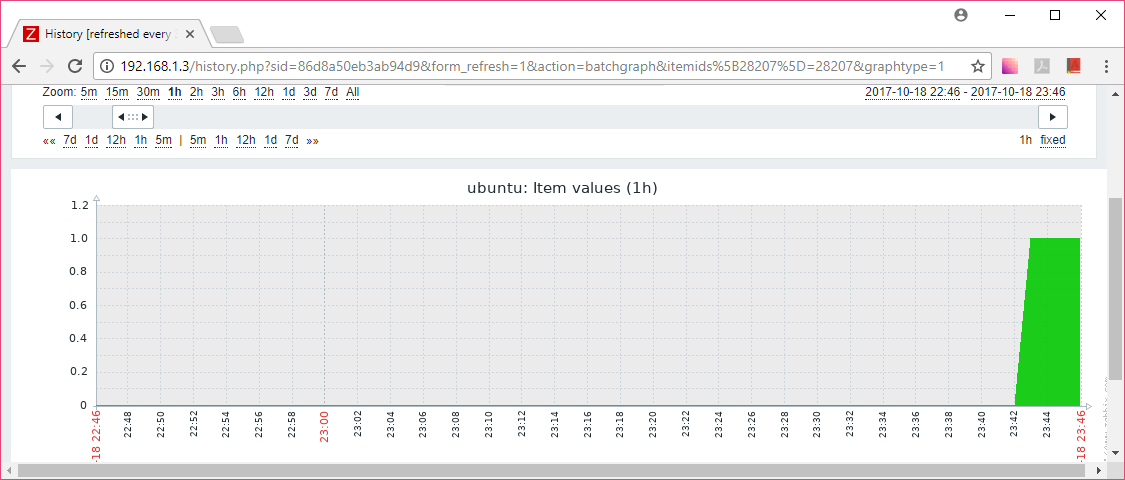
|  |
| --- |
| sudo apt install nginx |



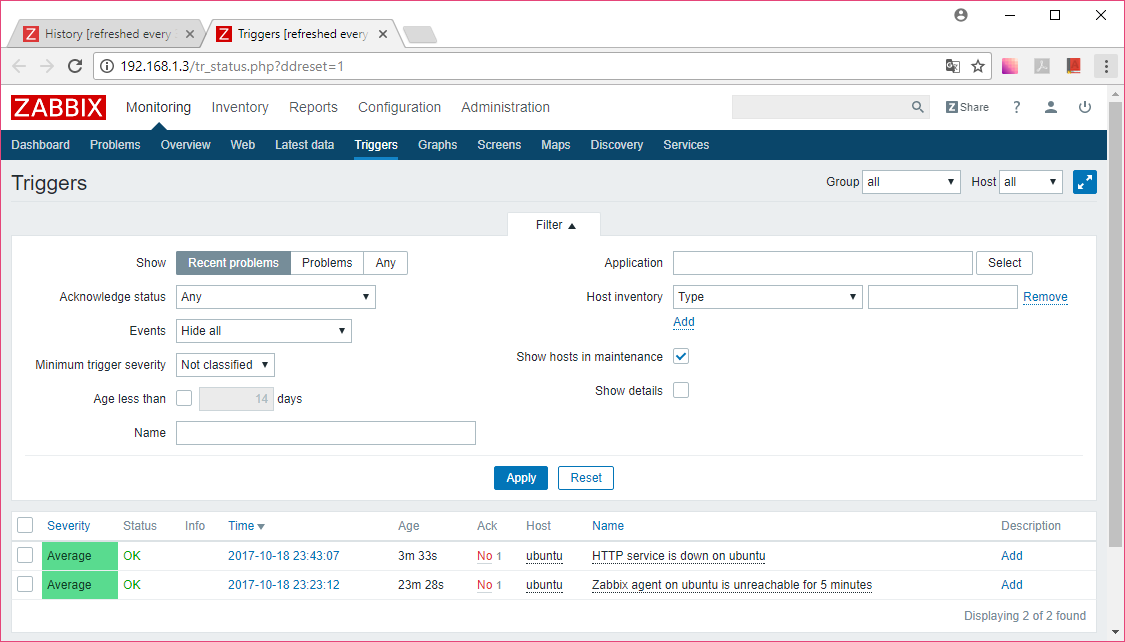
Сравним:



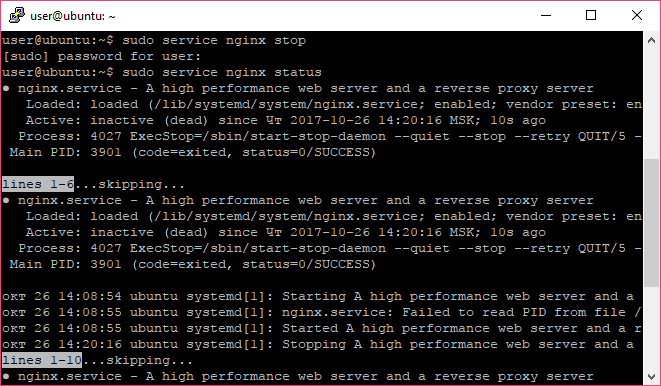
Поставим галочку и нажмем **Stacked graph:**

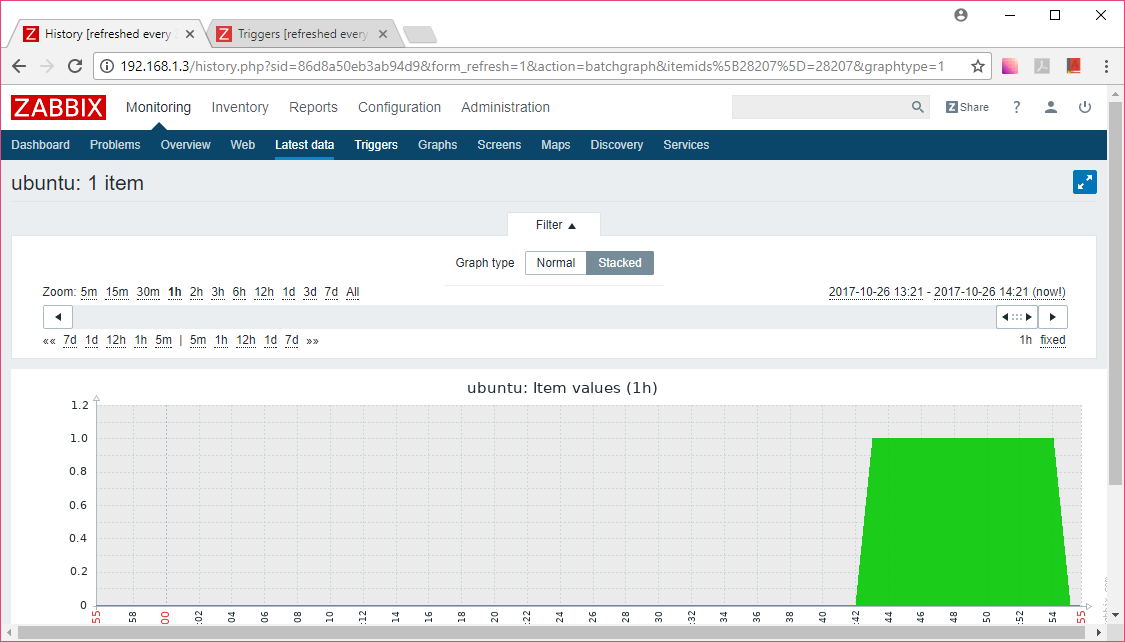


Зайдем в **Monitoring** > **Triggers.**

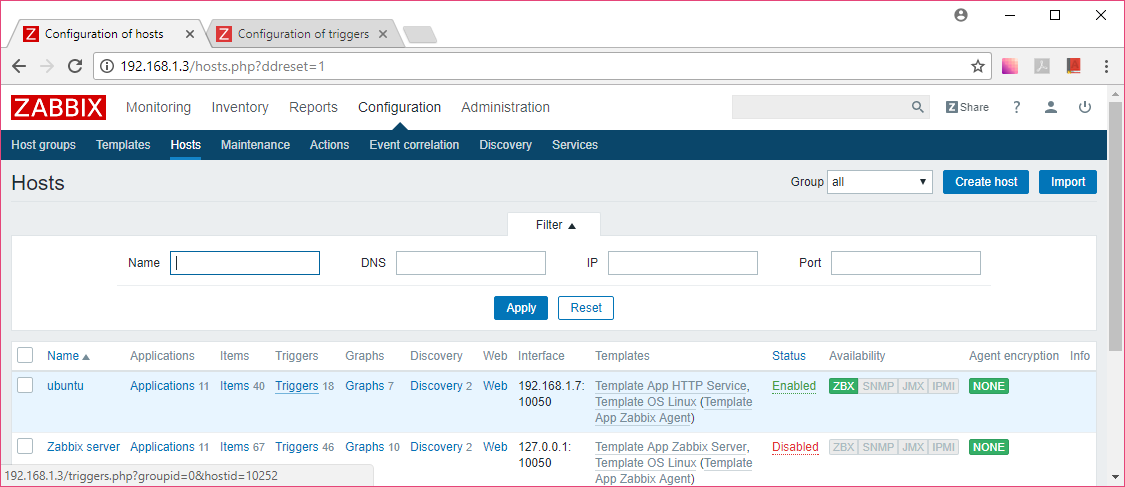


Остановим сервер:

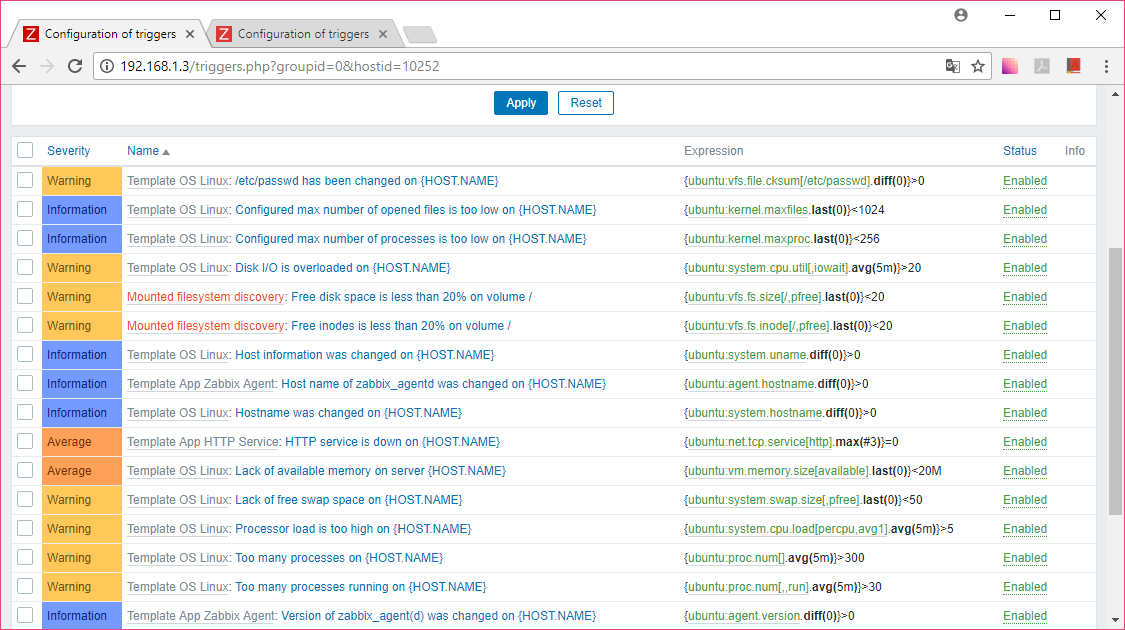




Также мы можем зайти в **Configuration** > **Hosts.**



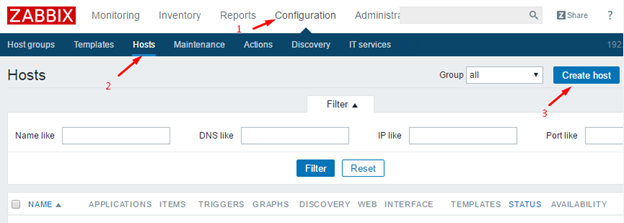
И кликнуть на **Triggers** для нашего хоста:



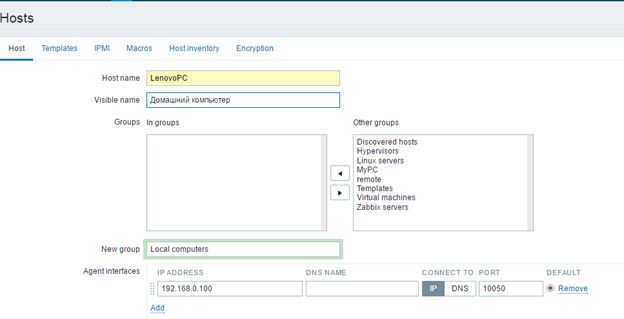
## Более подробно: hosts, items, triggers

Мы воспользовались шаблонами, которые уже включают Items и triggers, настроенные для тех или иных задач. Можно создавать свои шаблоны. Также можно создавать отдельные items и triggers, но это не всегда можно быть удобно. Разберем такой пример:

Заходим в меню **Configuration** > **Hosts.**



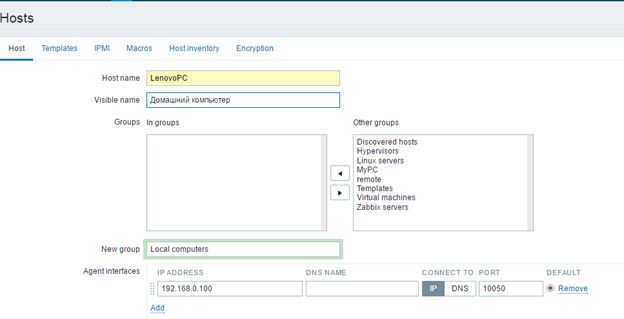
Нажимаем **Create host:**



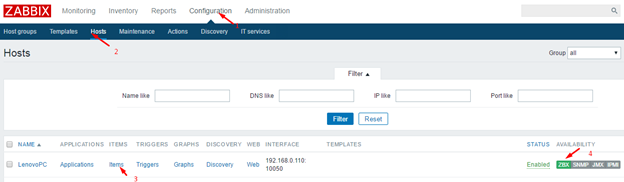
Заполняем:

* Host name – имя компьютера (hostname для Windows можно посмотреть в свойствах «Моего компьютера» или командой hostname в cmd);
* Visible Name – имя которое будет отображаться в списке;
* Groups – либо выбираем из существующих, либо можно создать сейчас (New group);
* как будем обращаться к агенту (по какому протоколу: если Agent interfaces, то собственный от Zabbix, но доступен и SNMP). Можно указать IP-адрес или доменное имя.

Сохраняем:



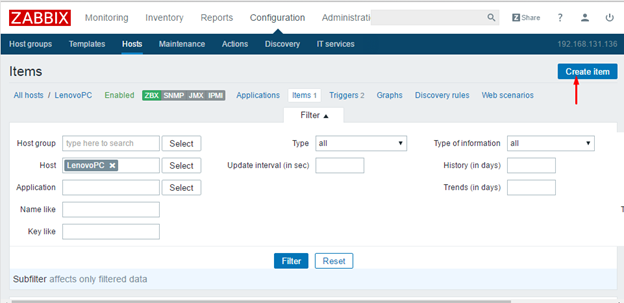
Заходим в меню в **Configuration** > **Hosts.**



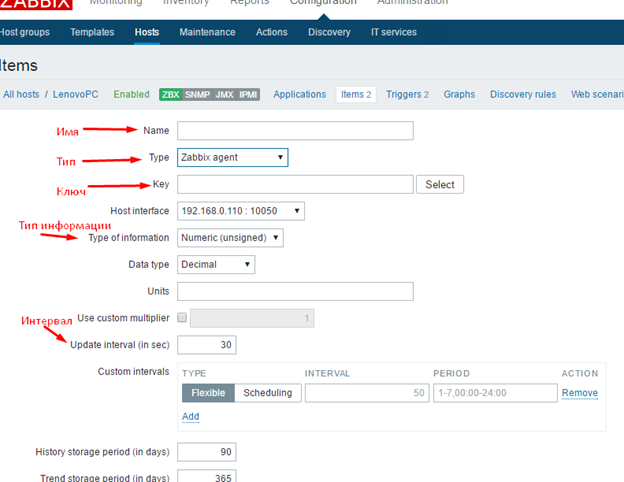
Жмем на **Items.**

Кстати, если **ZBX** отображается зеленым, то zabbix-агент-отвечает. Если нет (отображается красным), смотрите логи: возможно, что-то неверно настроено. Или вообще нет связи (No route to host).

Жмем **Create Item.**

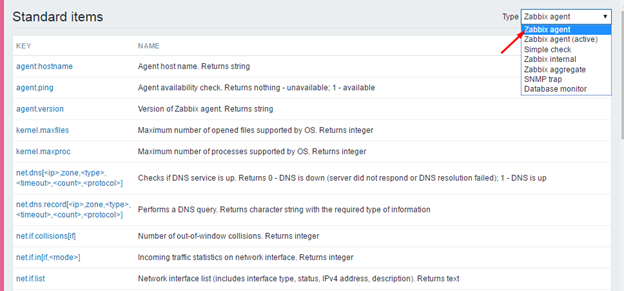


После этого появляется форма создания item:

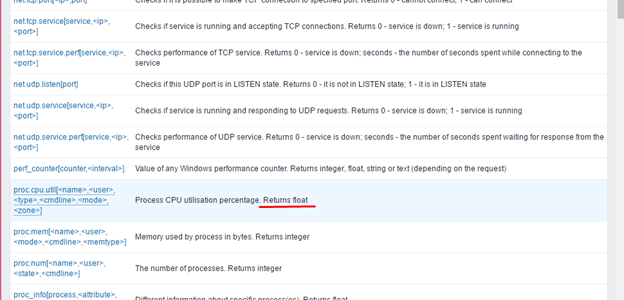


* Имя – пишем то имя, под которым он будет отображаться в списке айтемов (Items).
* Тип – выбираем, какой тип проверок использовать:
  + Zabbix agent – через установленный zabbix-агент;
  + Simple check – если zabbix-агент не установлен, то простые проверки, такие как ping и доступность сервиса по нужному нам порту (например, доступность чужого сервера, с которым есть взаимодействие через API);
  + также доступны SNMP-проверки для поддерживающих SNMP-устройств;
  + также есть активные проверки (агент инициативно присылает нам данные).
* Далее выбираем нужный нам ключ.

Нажимаем **Select,** открывается окно со списком.



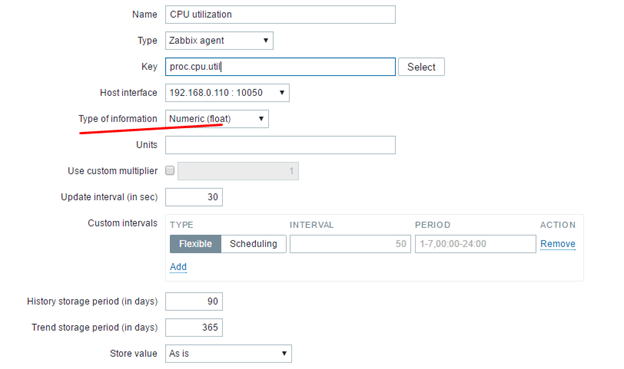
Здесь также можно выбрать тип проверки.



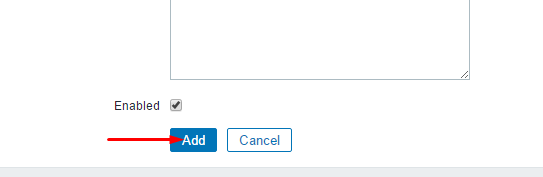
Выбираем нужный нам ключ. Обращаем внимание на тип, его же нужно указать в форме.

Не забываем выставить возвращаемый тип, в данном случае float.

Update intervals – указываем в секундах нужный нам интервал обновления.

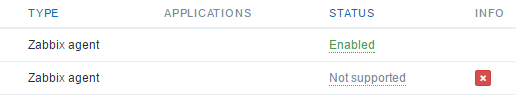


Добавляем:

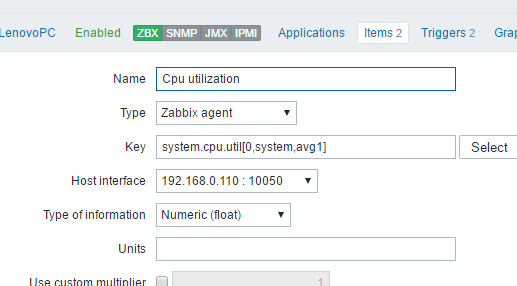


Если в списке Item отображается, как Not supported – смотрим в документации, например,

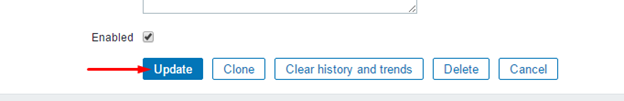
<https://www.zabbix.com/documentation/3.0/ru/manual/appendix/items/supported_by_platform>



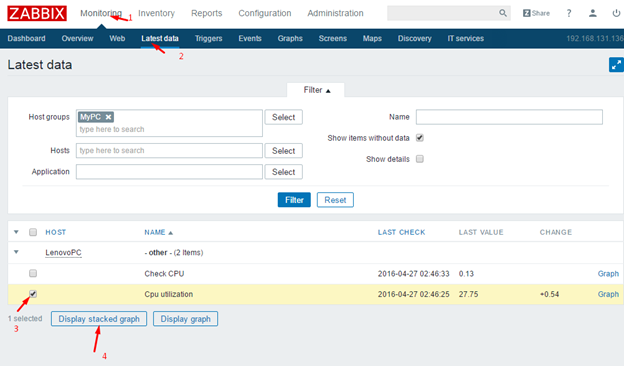
Например, при запуске агента на Windows следует вместо proc.cpu.util выбрать system.cpu.load или system.cpu.util[0,user,avg1] – разными операционными системами могут поддерживаться разные проверки.



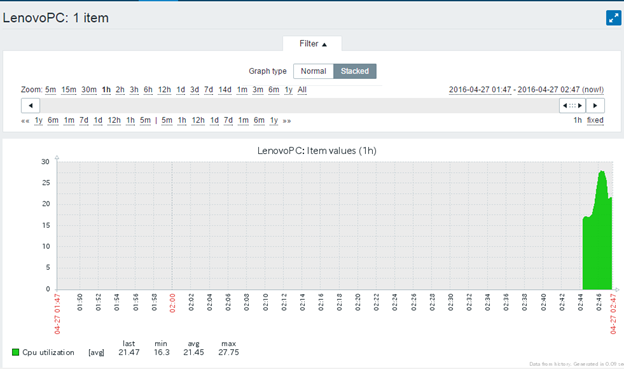
Сохраняем.



Если zabbix-агент запущен, а мы выбрали именно проверку для агента, заходим в **Monitoring** > **Latest date.**

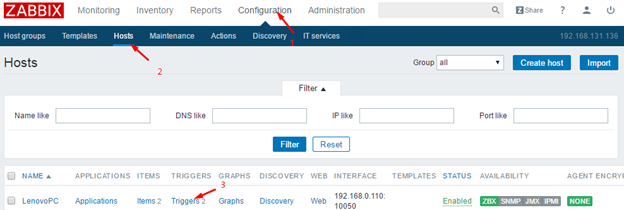


Выбираем проверку, строим график.

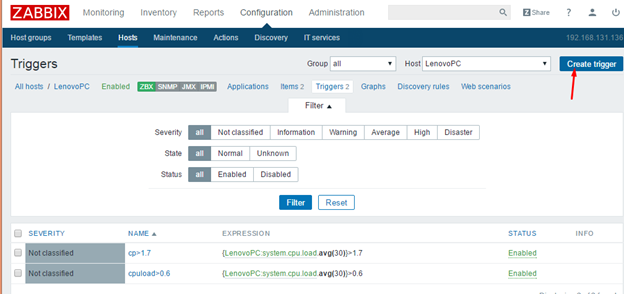


Мы можем воспользоваться существующими триггерами или создать свой.

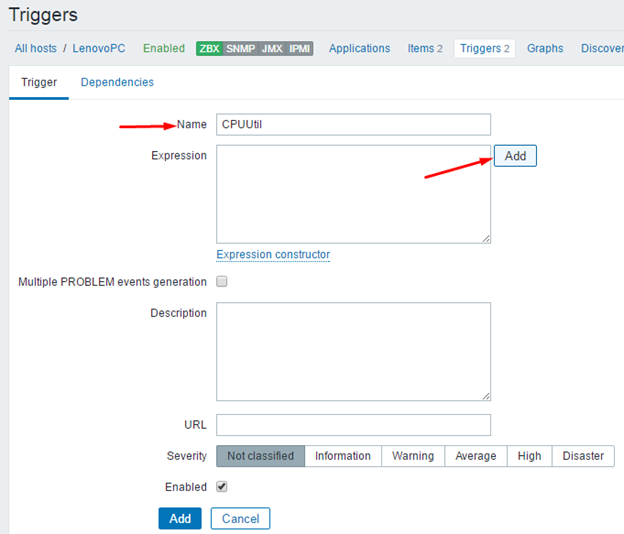
Заходим в **Configuration** > **Hosts.**



Выбираем **Trigger.**

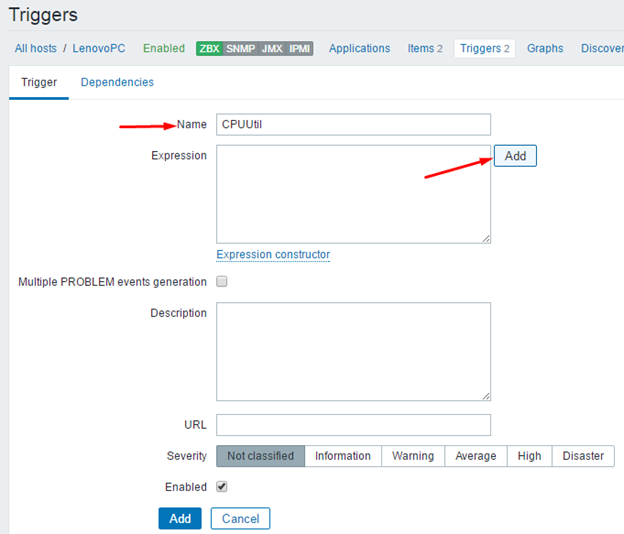


Получаем список триггеров (перечень уже созданных – например, на вкладке Triggers 2 будет уже два пункта – либо пустой).

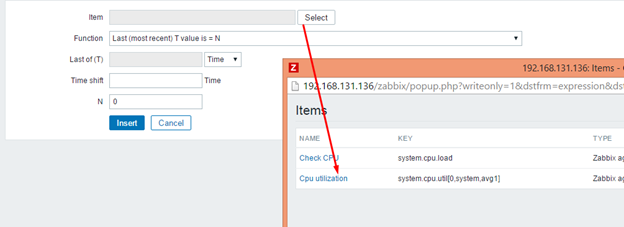


Жмем **Create trigger.**

Вводим имя триггера и жмем **Add.**

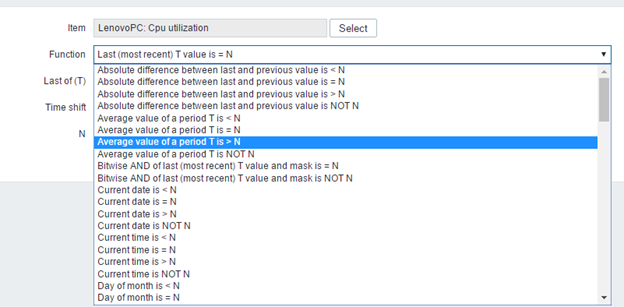


Появляется конструктор:

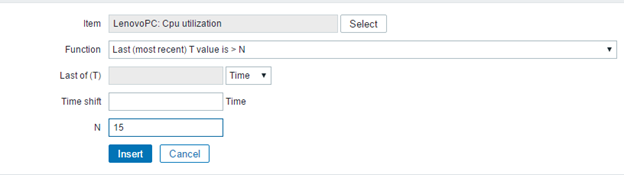


Выбираем **Item.**

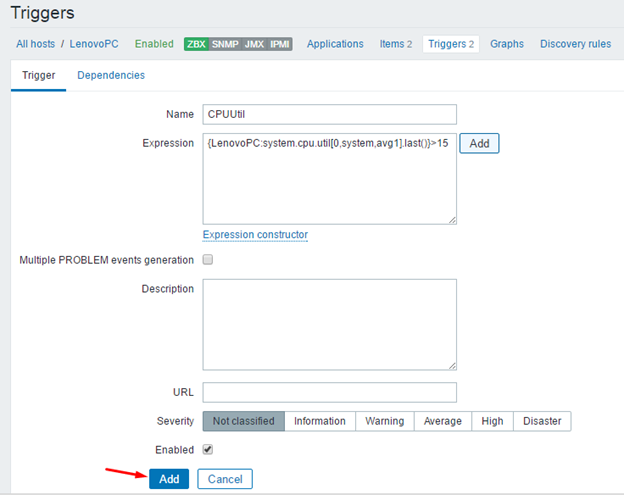
Выбираем функцию:



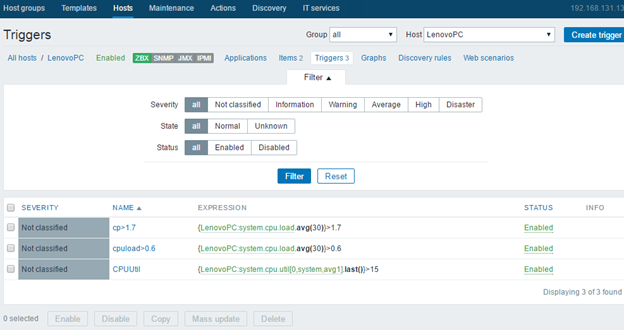
Заполняем:



Имя триггера скопируем, оно нам понадобится:

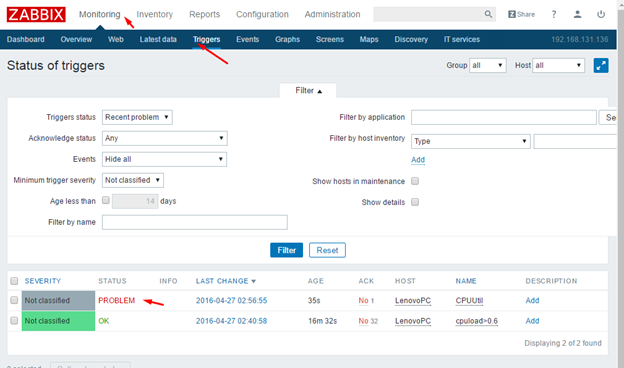


Жмем **Add.**

Добавлено.

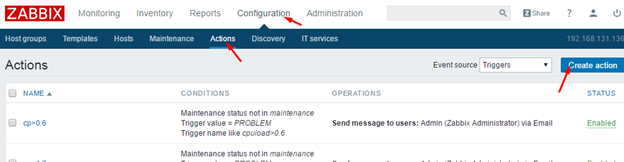
В **Monitoring** > **Triggers** можно воочию наблюдать проблему.

Статус мигает, если изменился, **Problem** отображается красным.



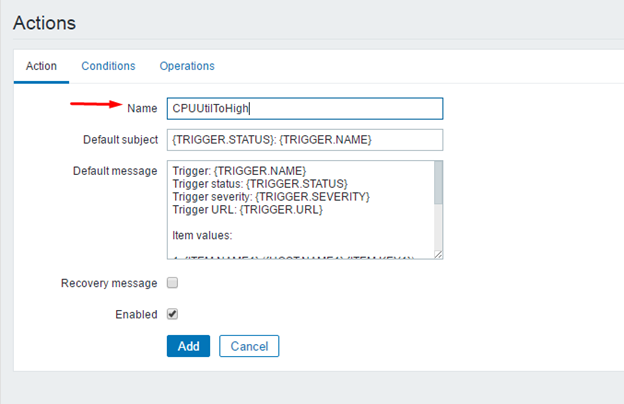
Теперь создадим действие.

**Configuration** > **Actions** > **Create action.**

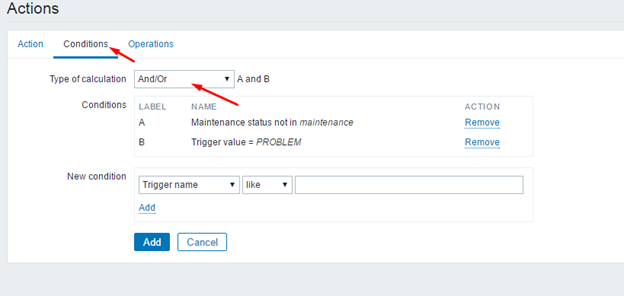


Заполняем имя действия (Name).

Можно при желании поменять шаблон письма.



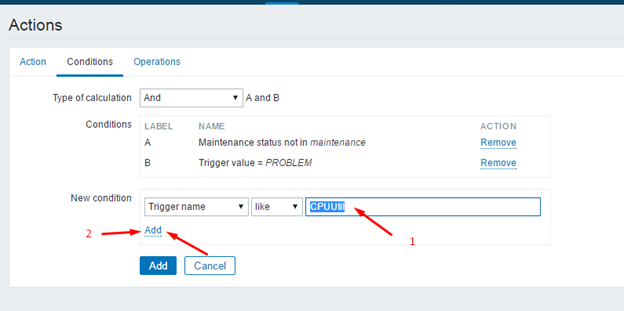
Нажимаем **Conditions.**



Выбираем **And.**

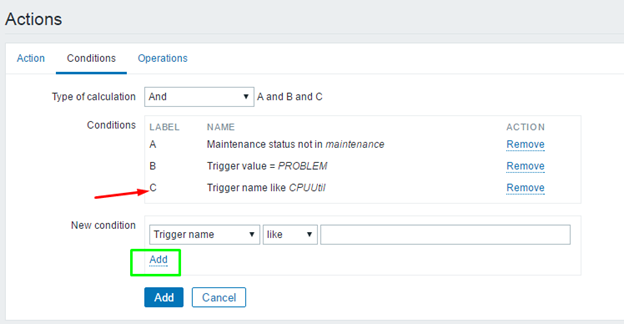
И копируем запомненное (записанное) нами имя триггера.

Trigger name like CPUUtil

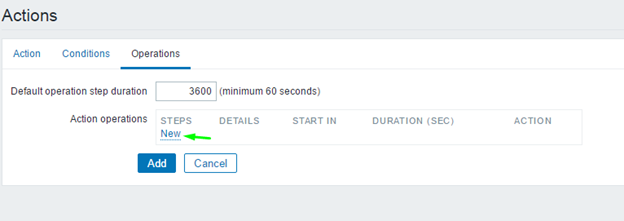


Жмем на подчеркнутую **Add** (номер 2 на рисунке, не на синюю кнопку!)

Последствия правильно нажатой **Add:**



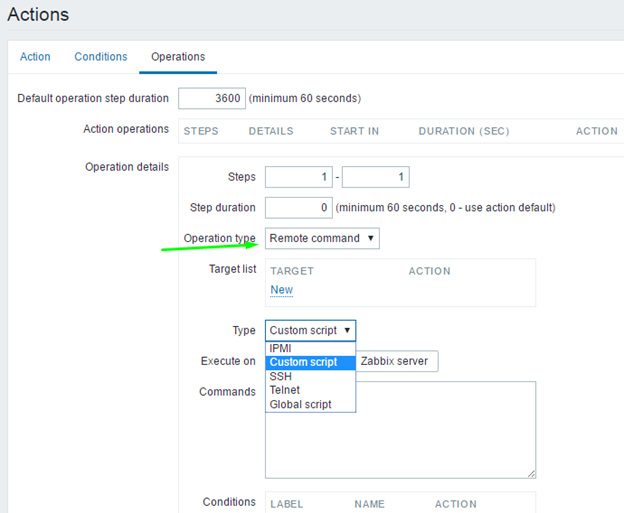
Переходим в **Operations.**



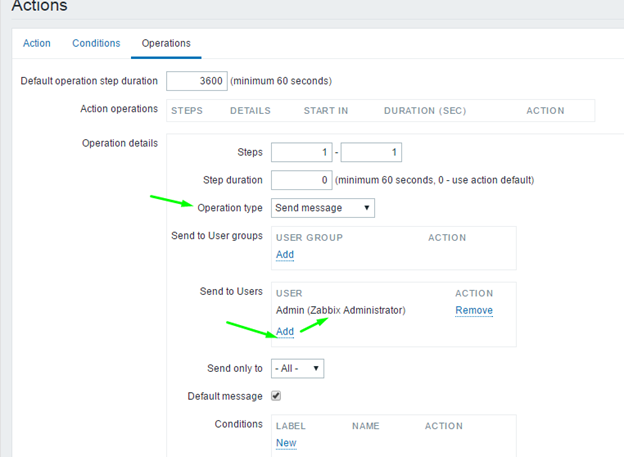
Жмем **New.**

Теперь можно выбрать, что будем делать.

Выполнить команду:

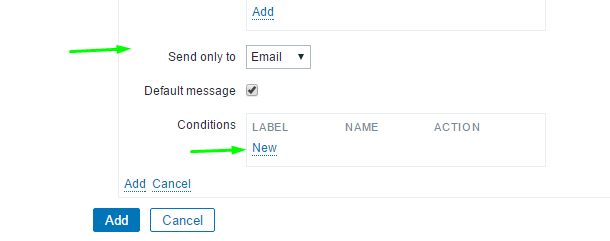


Либо отправить уведомление:

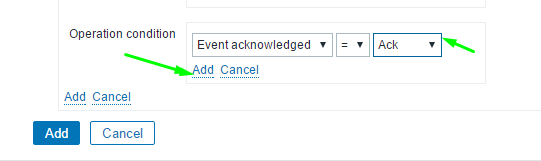


Уже выбран тип Operation type: Send message.

Send to users, жмем подчеркнутую **Add** и выбираем нашего пользователя.

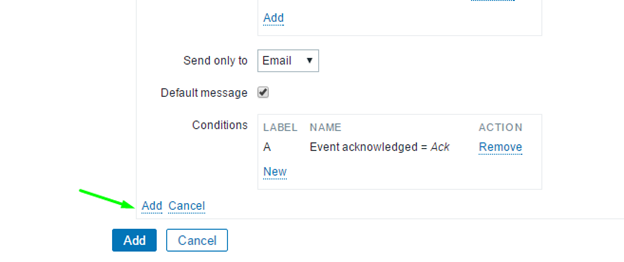


Выбираем тип оповещения и в Conditions жмем на **New**, добавляем условие:

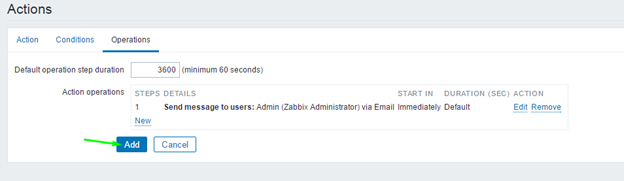


Не перепутайте кнопки **Add.** Нужная кнопка отмечена на картинке выше.

Теперь сюда:



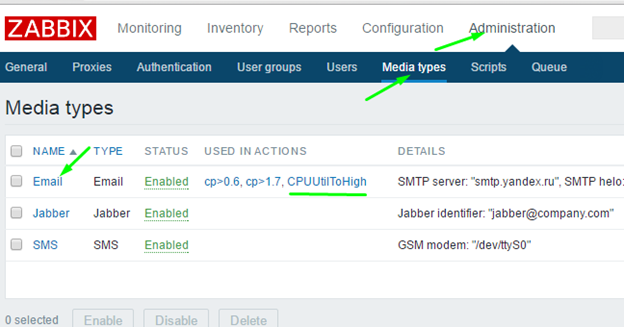
И сюда:



В списке добавился триггер:



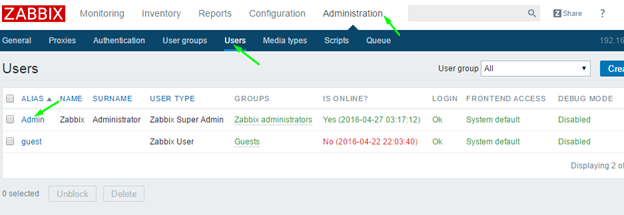
Теперь нужно настроить оповещения:



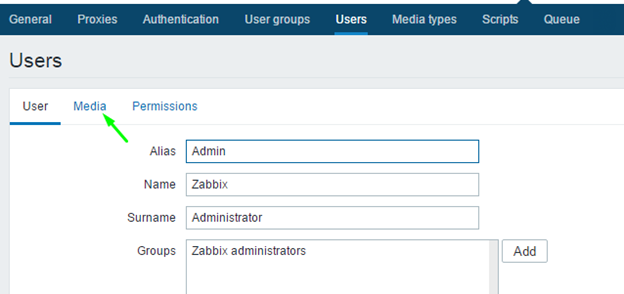
**Administration** > **Media types** > **Email**

Нажимаем на **Email** и вписываем настройки вашего SMTP-сервера (или, допустим, сервера Gmail или Yandex). Также доступны для оповещений Jabber (протокол XMPP) и SMS, если вы подключите к серверу USB-модем (главное, чтобы в серверной хорошо ловился сигнал оператора, а то упадет сеть, и связи тоже не будет).

Последний штрих:

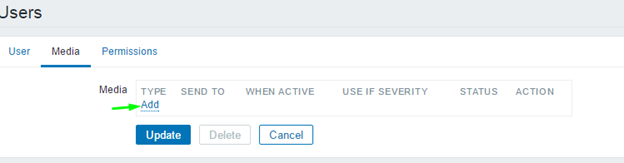


**Administration** > **Users** > **Admin.**



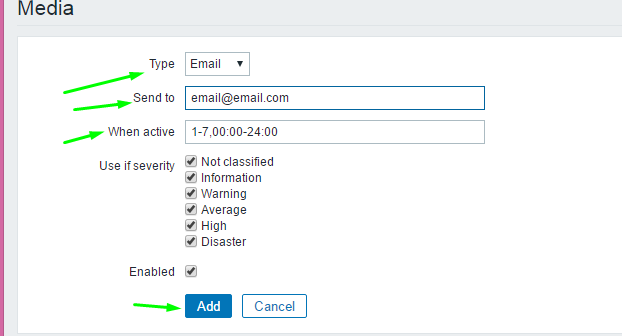
Заходим в **Media.**

Жмем **Add.**

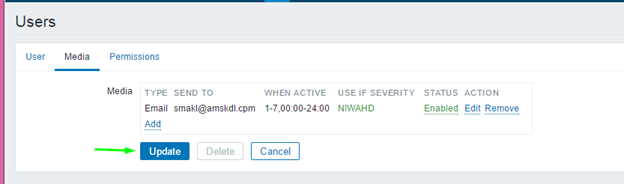


Здесь можно выбрать:

* тип оповещения;
* куда отправлять;
* в какое время (допустим, только ночью);
* тип уведомлений.



Жмем **Add.**



Теперь жмем **Update.**

# Домашнее задание

1. Установить zabbix-server, zabbix-agent, настроить базовые проверки.
2. Подключить виртуальную машину, настроить веб-проверки.

# Дополнительные материалы

1. <https://www.zabbix.com/download>
2. <https://www.zabbix.com/manuals>

# Используемая литература

Для подготовки данного методического пособия были использованы следующие ресурсы:

1. <https://ru.wikipedia.org/wiki/SNMP>
2. <https://ru.wikipedia.org/wiki/Zabbix>
3. <https://www.zabbix.com/documentation/3.0/ru/manual/appendix/items/supported_by_platform>