

## Лабораторная работа №7

### Мониторинг сети с помощью программ Zabbix и Wireshark

#### Теоретическая часть

Система мониторинга сервера — специальное программное обеспечение, предназначенное для непрерывного контроля и анализа работы сервера. Она позволяет отслеживать основные параметры сервера: загрузка процессора, использование памяти, объем дискового пространства, сетевая активность и другие ключевые показатели.

В простейшем случае, при работе на одной системе, например на смартфоне, возможно использовать одну из множества утилит, которая снимет значения нагрузки процессора, занимаемого объема ОЗУ или же свободного места на носителе (CPU-Z и другие). Если мы работаем на компьютере с ОС Linux, то возможно использовать консольные утилиты `htop`, `netstat` и др. Каждая из утилит применяется для мониторинга отдельных параметров - такой подход не применим для централизованного сбора информации. Поэтому необходимо использовать специализированные системы мониторинга для анализа системы.

С необходимостью систем мониторинга люди столкнулись при анализе производительности LAN сетей в рамках одного офиса. В них использовался специальный протокол SNMP (Simple Network Management Protocol), который позволял настраивать двусторонний или односторонний доступ к различным сетевым устройствам: от роутеров и коммутаторов до принтеров. На его основе настраивались такие инструменты как `Big Brother` и `nmon`, которые предоставляли информацию о сетевых нагрузках и событиях в сети, используя для этого сетевое соединение.

С появлением веб-сайтов и интернет-сервисов стало необходимо модифицировать системы. В результате были разработаны веб-ориентированные легко масштабируемые инструменты, поддерживающие интернет-протоколы. Среди популярных open-source инструментов начала 2000-х — `Zabbix`, `Nagios` и `Cacti`. В дальнейшем весь рост систем мониторинга происходил в сторону увеличения количества метрик или же разнообразия объектов для снятия этих метрик.

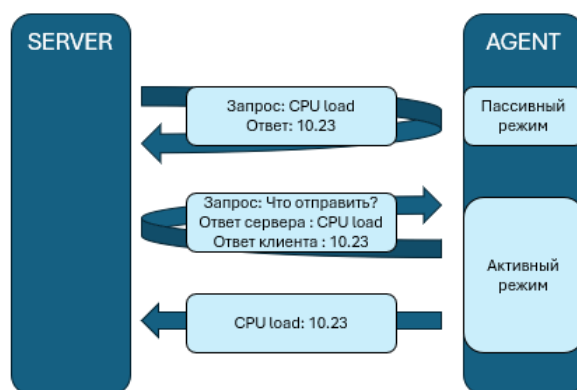
На сегодня основными системами мониторинга являются:

- **PingInfoView, Pingdom** – простые системы, использующие `ping` для проверки доступности узлов в сети. Позволяют с интервалом проверять состояние узлов и отображать доступность в режиме реального времени.
- **Zabbix** – мощная система с поддержкой агентов и безагентного сбора данных (SNMP, IPMI, HTTP и т.д.), средствами анализа данных, уведомлениями и веб-интерфейсом.

- **PRTG Network Monitor** – система с широким набором сенсоров для сбора данных (SNMP, WMI, ICMP и пр.), анализом данных, уведомлениями и удобным интерфейсом. В отличие от многих других конкурентов имеет ограничение в 100 метрик (сенсоров).
- **Graphite** — это бесплатный инструмент с открытым исходным кодом (FOSS), который строит графики числовых временных рядов, таких как производительность компьютерных систем. Graphite собирает, хранит и отображает данные временных рядов в реальном времени.
- **Prometheus** – система, в основе которой лежит база данных временных рядов (Time series database, TSDB). Поддерживает экспортеры и PushGateway (возможность отсылать метрики из кастомных скриптов и программ), уведомления и интеграцию с Grafana для визуализации.

Существуют две основные модели мониторинга:

- Push-модель – сервер мониторинга ожидает подключений от агентов для получения метрик. Модель обычно используется для мониторинга больших систем, где количество устройств может быть слишком большим для ручного сбора данных.
- Pull-модель – сервер мониторинга сам подключается к агентам мониторинга и забирает данные.



Современные системы, такие как Zabbix и Prometheus работают как Push и Pull модель. Обе системы являются очень популярными на рынке.

## Wireshark

Wireshark — программа-анализатор трафика для компьютерных сетей Ethernet и некоторых других. Имеет графический пользовательский интерфейс.

Для установки пакета необходимо воспользоваться программой:

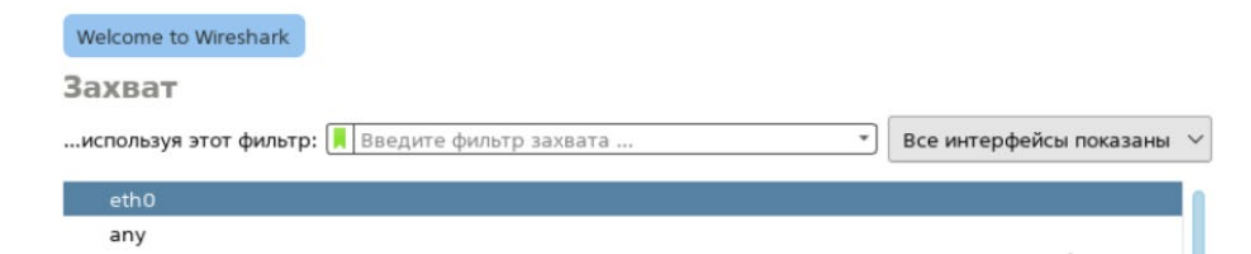
```
sudo apt install wireshark
```

При установке возможно указать, чтобы пользователи не обладающие правами суперпользователя могли производить захват трафика.

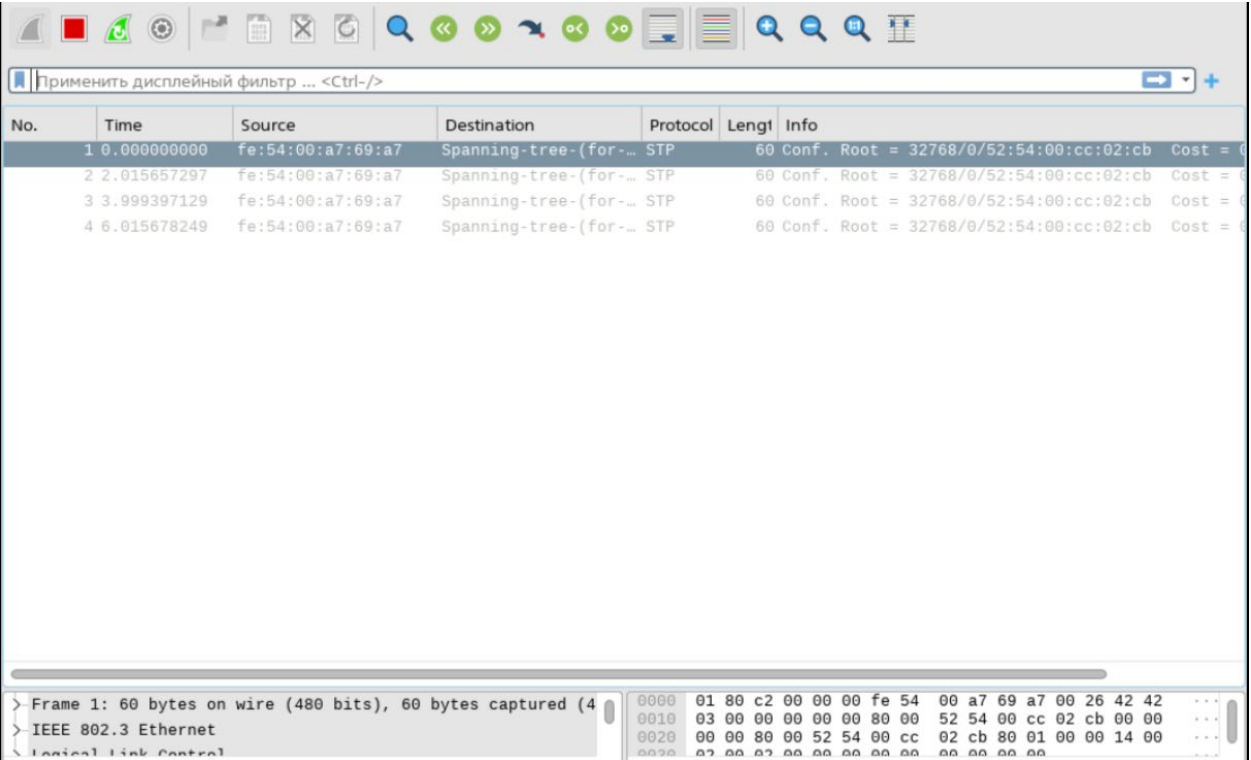
Для запуска программы воспользуйтесь командой:

```
sudo wireshark
```

После запуска программы выберите интерфейс, который вы будете прослушивать:



Далее открывается рабочее окно, в котором возможно наблюдать весь сетевой трафик, проходящий через указанный интерфейс.

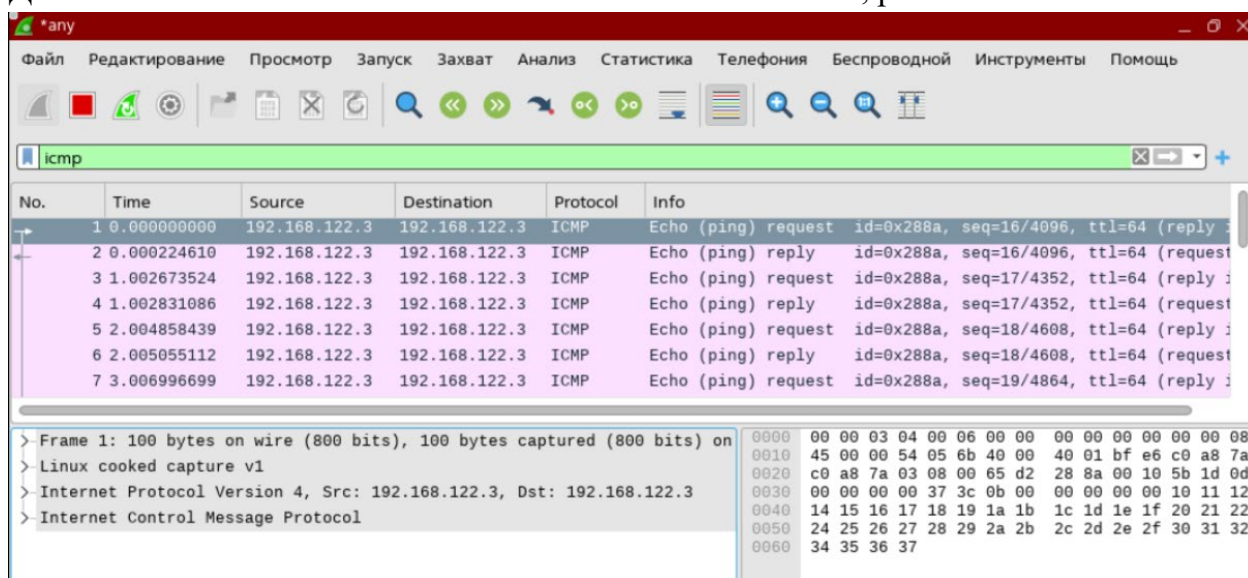


Для отбора необходимых строк возможно использовать фильтры.

Название фильтра	Комментарий	Пример
<Название протокола>	Фильтрация пакетов с определенных,	icmp
ip.src==<ip адрес>	Фильтрация по адресу источника сообщений	ip.src==192.168.122.2
ip.dst==<ip адрес>	Фильтрация по адресу назначения сообщений	ip.dst==192.168.122.3
<Название протокола>.port == <номер порта>	Фильтрация по порту	tcp.port == 80

С помощью символов || фильтры возможно объединять  
Например: icmp || tcp || udp

Для анализа пакета возможно воспользоваться окошками, расположенными ниже:

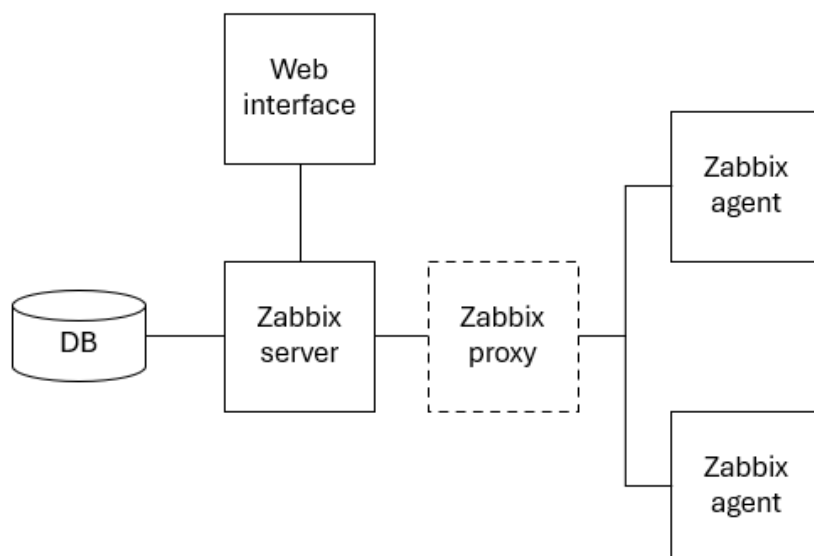


В правом окошке представлен пакет в «сыром» виде, как последовательность байт. В левом окне выводится его расшифровка.

Более подробно по программе wireshark возможно прочитать в источнике 1.

## Zabbix

Zabbix — свободная система мониторинга статусов разнообразных сервисов компьютерной сети, серверов и сетевого оборудования, написанная Алексеем Владышевым. Для хранения данных используется MySQL, PostgreSQL, SQLite или Oracle Database, веб-интерфейс написан на PHP.



**Zabbix-сервер** — ядро системы, которое дистанционно контролирует сетевые сервисы и которое является хранилищем, содержащим все конфигурационные, статистические и оперативные данные. Он является тем

субъектом в программном обеспечении Zabbix, который оповещает администраторов о проблемах с контролируемым оборудованием.

**Zabbix-прокси** собирает данные о производительности и доступности от имени Zabbix-сервера. Все собранные данные заносятся в буфер на локальном уровне и передаются Zabbix-серверу, к которому принадлежит прокси-сервер. Zabbix-прокси является идеальным решением для дистанционного контроля филиалов и других точек, в том числе сетей, не имеющих местных администраторов. Он может быть также использован для распределения нагрузки одного Zabbix-сервера. В этом случае прокси только собирает данные, благодаря чему на сервер ложатся меньшие нагрузки на ЦПУ и устройства ввода/вывода.

**Zabbix-агент** — программа контроля локальных ресурсов и приложений (таких как накопители, оперативная память, статистика процессора и т. д.) на сетевых системах, эти системы должны работать с запущенным Zabbix-агентом. Zabbix-агенты являются чрезвычайно эффективными из-за использования специфических системных вызовов для сбора информации и подготовки статистики.

**Веб-интерфейс** — часть Zabbix-сервера, и, как правило (но не обязательно), запускается на том же физическом узле, что и Zabbix-сервер. Работает на PHP, требует веб-сервер (например nginx, Apache httpd).

## Установка и настройка Zabbix

Для корректной работы сервера необходимо настроить веб-интерфейс (apache2 + php), БД и доступ к ней, Locale для поддержки английского языка.

Первым шагом требуется установить необходимые пакеты:

```
sudo apt install zabbix-server-pgsql zabbix-frontend-php php-pgsql
```

Перед началом работы необходимо назначить метки безопасности служебным пользователям postgres и zabbix:

```
sudo pdpl-user -i 63 postgres  
sudo pdpl-user -l 0:0 zabbix  
sudo usermod -a -G shadow postgres
```

Предоставить служебному пользователю postgres право чтения базы данных меток безопасности локальных пользователей:

```
sudo setfacl -d -m u:postgres:r /etc/passwd/{macdb,capdb}  
sudo setfacl -R -m u:postgres:r /etc/passwd/{macdb,capdb}  
sudo setfacl -m u:postgres:rx /etc/passwd/{macdb,capdb}
```

## Настройка apache2

В файле /etc/php/\*/apache2/php.ini раскомментировать и дописать

```
[Date]  
date.timezone = Europe/Moscow
```

В файле /etc/apache2/apache2.conf установить “AstraMode off”

Этим действием мы установили часовой пояс на веб сервере и отключили astra security mode

Необходимо проверить, что DNS работает корректно и производится преобразование доменных имен в IP адреса.

### Настройка СУБД PostgreSQL

Отредактировать конфигурационный файл /etc/postgresql/\*/main/pg\_hba.conf:

# TYPE	DATABASE	USER	ADDRESS	METHOD
local	zabbix	zabbix		trust
# IPv4 local connections:				
host	zabbix	zabbix	127.0.0.1/32	trust

Перезапустить кластер:

```
sudo systemctl restart postgresql
```

Создать пользователя и базу zabbix:

```
sudo -u postgres psql
```

```
CREATE DATABASE ZABBIX;  
CREATE USER zabbix WITH ENCRYPTED PASSWORD '12345678';  
GRANT ALL ON DATABASE zabbix to zabbix;  
ALTER DATABASE zabbix OWNER TO zabbix;
```

Далее копируется шаблон базы данных

```
sudo zcat /usr/share/zabbix-server-pgsql/{schema,images,data}.sql.gz | psql -h  
localhost zabbix zabbix  
sudo a2enconf zabbix-frontend-php  
sudo systemctl reload apache2
```

### Настройка веб интерфейса

Скопировать в файл /etc/zabbix/zabbix.conf.php

```
sudo cp /usr/share/zabbix/conf/zabbix.conf.php.example /etc/zabbix/zabbix.conf.php
```

Установить права доступа к созданному файлу:

```
sudo chown www-data:www-data /etc/zabbix/zabbix.conf.php
```

В файле /etc/zabbix/zabbix.conf.php задать значения переменных TYPE (тип используемой СУБД) и PASSWORD (пароль пользователя zabbix СУБД):

```
$DB['TYPE'] = 'POSTGRESQL';  
$DB['PASSWORD'] = '<12345678>';
```

Перезапустить службу apache2:

```
sudo systemctl reload apache2
```

В конфигурационном файле /etc/zabbix/zabbix\_server.conf раскомментировать строку, задающую пароль доступа к БД zabbix, и указать там пароль:

```
DBPassword=12345678
```

Разрешить автоматический запуск службы zabbix при перезагрузке ОС и запустить службу zabbix:

```
sudo systemctl enable zabbix-server  
sudo systemctl start zabbix-server
```

### Настройка locale

В файле /etc/locale.gen раскомментировать строку en\_US.UTF-8. Выполнить locale-gen. Открыть WEB-страницу zabbix в WEB-браузере. Имя для входа: Admin (с заглавной буквы), пароль для входа: zabbix

```
firefox localhost/zabbix
```

### Настройка Zabbix-agent

Устанавливаем на клиенте службу

```
sudo apt install zabbix-agent
```

В конфигурационном файле /etc/zabbix/zabbix\_agentd.conf меняем ip сервера на актуальный(который меняли в /etc/hosts), имя хоста устанавливаем как имя клиента.

```
Server=192.168.122.2
```

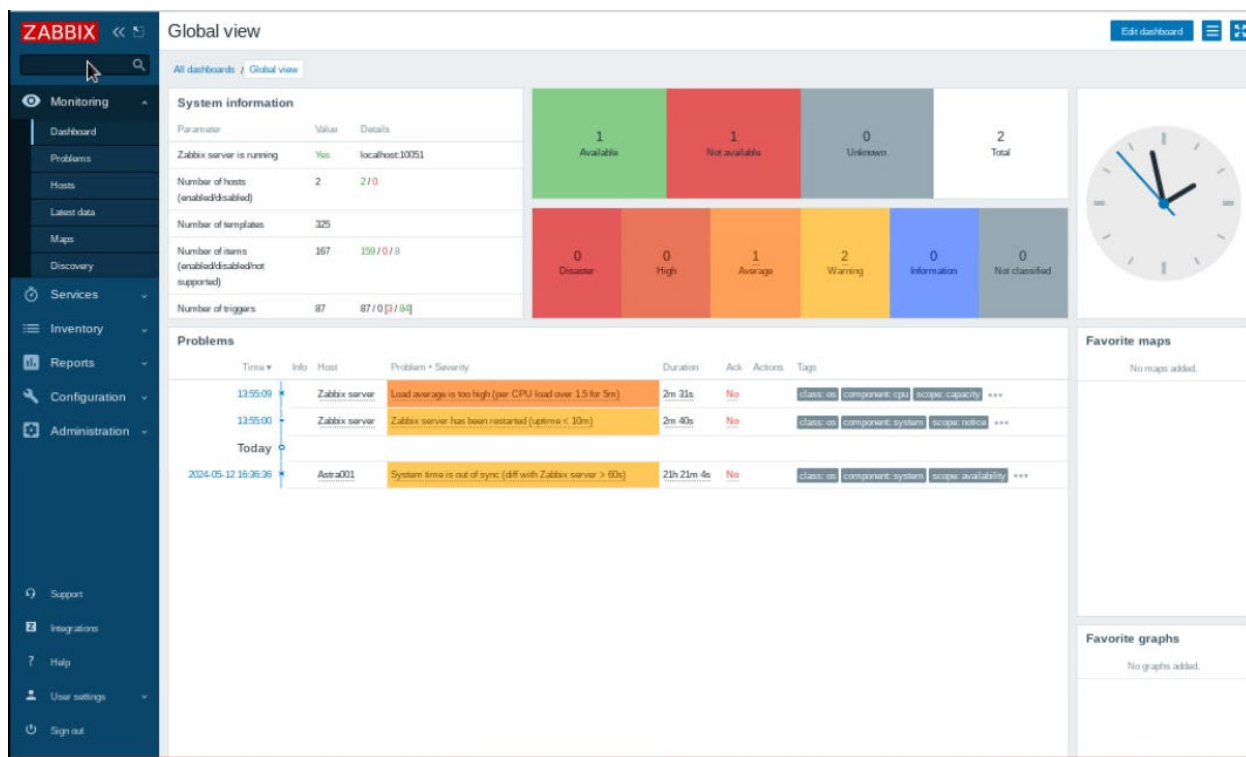
Перезапустим агент

```
sudo systemctl restart zabbix-agent
```

### Настройка

Находясь в WEB-странице zabbix



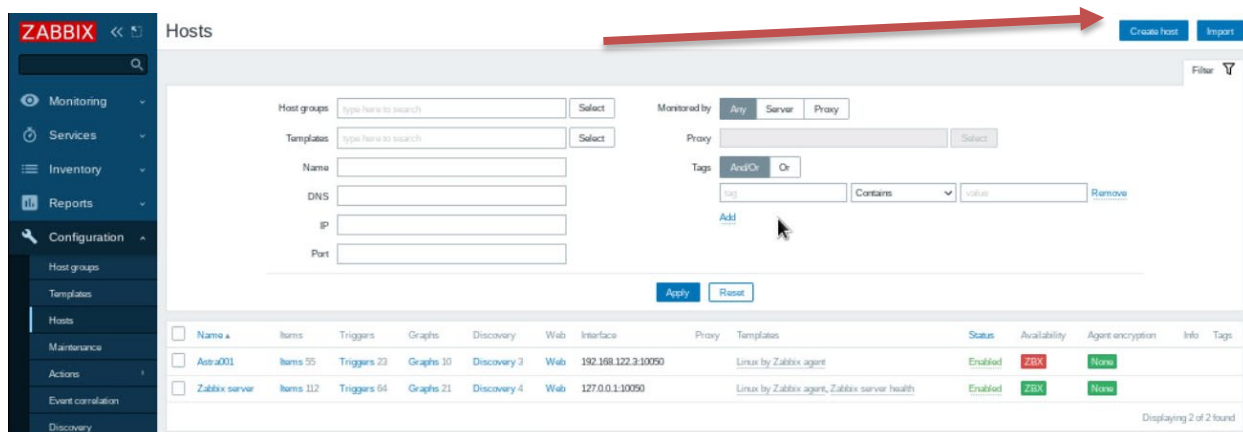


- Monitoring вкладки содержат подробную информацию о системе, хостах. Здесь можно найти ошибки и заглянуть в логи. В вкладке hosts – все подключенные серверы, с подробными данными и графиками о состоянии их метрик
- Services – возможность добавления услуг бизнес уровня для мониторинга
- Inventory – Меню Инвентаризация включает разделы обеспечивающие обзор инвентарных данных узлов сети по выбранному параметру, а также, возможность просмотра деталей инвентаризации узлов сети.
- Reports – Меню Отчеты включает в себя несколько разделов, которые содержат различные предустановленные и пользовательские отчеты, направленные на обзор таких параметров как информации о системе, триггеров и собранных данных.
- Configuration меню настроек, здесь можно выбрать шаблоны, добавить hosts, настроить события и тд.
- Administration Меню Администрирование используется в Zabbix для административных функций. Это меню доступно только пользователям с типом Super Administrators.

### Добавление хоста и метрик

Для добавления хоста перейдем во вкладку Configuration>host. В правом верхнем углу нажмем create host





Host

Host IPMI Tags Macros Inventory Encryption Value mapping

\* Host name:

Visible name:

Templates: Name:  Action: [Unlink](#) [Unlink and clear](#)

\* Groups:

Interfaces: Type:  IP address:  DNS name:  Connect to:  Port:  Default: ☒ Remove

Add

Description:

Monitored by proxy:

Enabled: ☒

[Update](#) [Clone](#) [Full clone](#) [Delete](#) [Cancel](#)

Для того, чтобы хост был доступен необходимо добавить на клиент шаблон (Linux by Zabbix agent)

Имя хоста должно совпадать с тем, которое задали на клиенте(агенте), и добавляем ip адрес агента. Если все выполнено правильно zbx значок в столбце Availability станет зеленым через какое-то время

↓

<input type="checkbox"/>	Name	Items	Triggers	Graphs	Discovery	Web	Interface	Proxy	Templates	Status	Availability	Agent encryption	Info	Tags
<input type="checkbox"/>	Astra001	<span style="border: 1px solid red;">Items 55</span>	<span style="border: 1px solid red;">Triggers 23</span>	Graphs 10	Discovery 3	Web	192.168.122.3:10050		Linux by Zabbix agent	Enabled	<span style="border: 1px solid red;">ZBX</span>	None		
<input type="checkbox"/>	Zabbix server	Items 112	Triggers 64	Graphs 21	Discovery 4	Web	127.0.0.1:10050		Linux by Zabbix agent, Zabbix server health	Enabled	ZBX	None		

Displaying 2 of 2 found

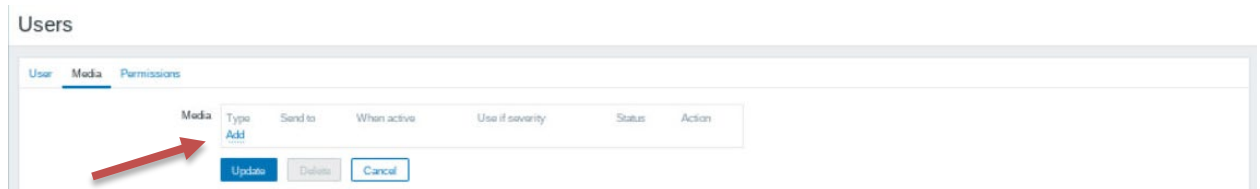
Для редактирования метрик или триггеров необходимо кликнуть на items или triggers

В случае ошибки нужно проверить конфигурацию хостов и перезапускать apache2 на сервере.

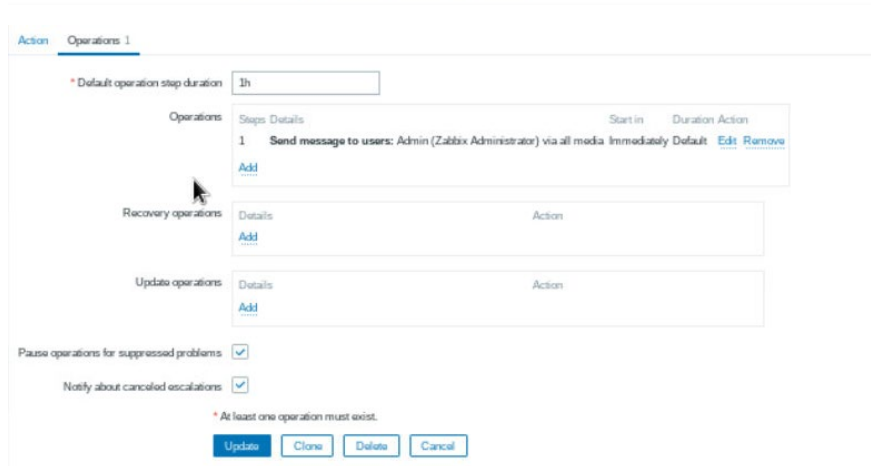
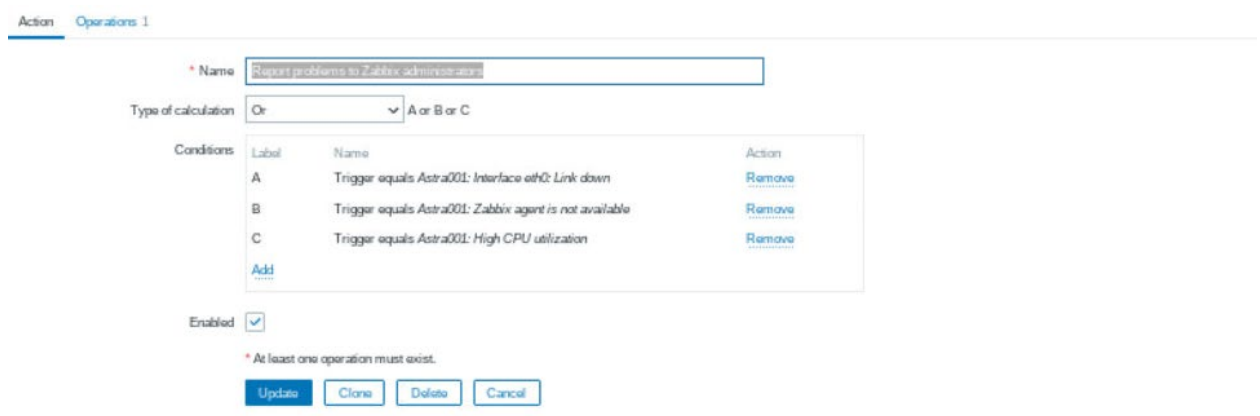
## Настройка сообщений о ключевых событиях

Необходимо зарегистрировать почту с которой будет писать заббикс сервер и добавить ее в Administrations>Media types (на примере Mail.ru, также необходимо получить пароль от почты для внешних приложений)

Далее необходимо добавить почту пользователю admin(Administrations>Users)



Сами сообщения настраиваются во вкладке Configuration>Actions>Trigger actions



После настройки, при возникновении указанных событий(перегрузка процессора, обрыв соединения) сообщения об этих событиях будут приходить на указанную почту.

## Практическая работа

### 1. Wireshark

- a. Запустите программу wireshark на сервере
- b. Установите фильтр, выбрав ip-адреса клиента, сервера и сетевого моста.
- c. Выполните следующие команды и проанализируйте результат их работы в wireshark:
  - i. ping с сервера на клиент
  - ii. ping на адрес, расположенный вне сети
  - iii. ping с клиента на сервер по доменному имени сервера
  - iv. ping с клиента на сервер по доменному имени сервера, не включенного в список DNS
  - v. открытие веб-страницы из предыдущей лабораторной работы на клиенте
  - vi. обновление адреса DHCP на клиенте с помощью команд `dhclient -r`; `dhclient`
  - vii. обмен TCP сообщениями между клиентом и сервером (запуск файлов `tcpserver.py`, `tcpclient.py`, исходные коды в приложении)
  - viii. обмен UDP сообщениями между клиентом и сервером (запуск файлов `udpserver.py`, `udpclient.py`, исходные коды в приложении)

### 2. Zabbix

- a. Установите и настройте систему мониторинга Zabbix
- b. Добавьте мониторинг следующих параметров: Нагрузка CPU, объем свободной/занятой оперативной памяти, объем свободной/занятой памяти жесткого диска
- c. Отобразите выбранные параметры в виде графиков
- d. Добавьте триггер, срабатывающий в случае перезагрузки клиентской машины, отключения сетевого интерфейса, перегрузки процессора
- e. Настройте всплывающие уведомления
- f. Настройте отправку уведомлений на вашу электронную почту
- g. Опционально: настройте отправку уведомлений в телеграмм/discord

## Список литературы:

[1] <https://wireshark.wiki>

## Приложение:

### *tcpclient.py*

```
import socket

HOST = "127.0.0.1"
PORT = 65432
bufferSize = 1024

TCPClientSocket=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
TCPClientSocket.connect((HOST, PORT))
TCPClientSocket.sendall(b"Hello, world")
data = TCPClientSocket.recv(bufferSize)
print(f"Received {data!r}")
```

### *tcpserver.py*

```
import socket

HOST = "127.0.0.1"
PORT = 65432
bufferSize = 1024

TCPServerSocket=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
TCPServerSocket.bind((HOST, PORT))
TCPServerSocket.listen()

TCPAnswer=TCPServerSocket.accept()
connection = TCPAnswer[0]
address = TCPAnswer[1]
print(f"Connected by {address}")
data = connection.recv(bufferSize)
connection.sendall(data)
```

### *udpclient.py*

```
import socket

msgFromServer = "Hello world!"
bytesToSend = str.encode(msgFromServer)

serverAddressPort = ("127.0.0.1", 65432)
bufferSize = 1024

UDPClientSocket=socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
UDPClientSocket.sendto(bytesToSend, serverAddressPort)
data = UDPClientSocket.recv(bufferSize)
print(f"Received {data!r}")
```

### *udpserver.py*

```
import socket
```

```
HOST = "127.0.0.1"
PORT = 65432
bufferSize = 1024

UDPServerSocket=socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
UDPServerSocket.bind((HOST, PORT))

UDPAnswer=UDPServerSocket.recvfrom(bufferSize)
connection = UDPAnswer[0]
address = UDPAnswer[1]
print(f"Connected by {address}")
UDPServerSocket.sendto(connection,address)
```