

Лабораторная работа №7

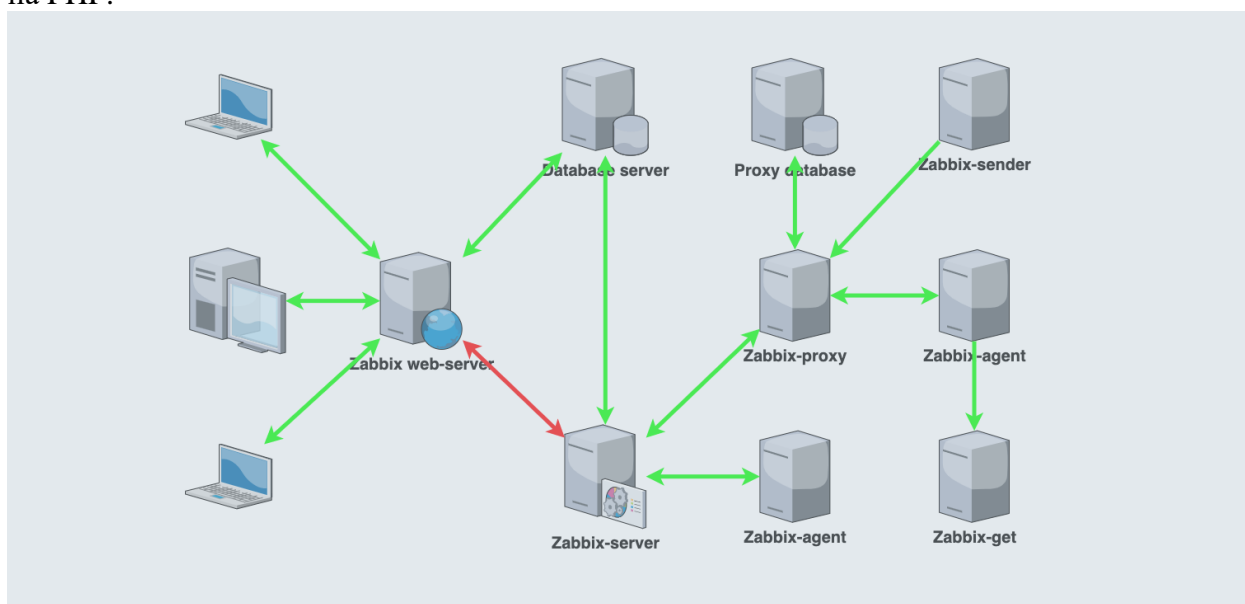
Мониторинг сети с помощью программ Zabbix и Wireshark

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.

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

- Установить необходимые пакеты: `sudo apt install zabbix-server-pgsql zabbix-frontend-php php-pgsql`
- веб интерфейс(apache2 + php)
- БД и доступ к ней
- Locale для поддержки английского

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

В файле `/etc/hosts`:

```
127.0.0.1      localhost
#127.0.1.1    astra002
192.168.122.3  astra001
192.168.122.30 asrtra002.example.com astra002
```

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;
\q
```

Выдать СУБД привелегии

```
usermod -a -G shadow postgres
setfacl -d -m u:postgres:r /etc/parsec/macdb
setfacl -R -m u:postgres:r /etc/parsec/macdb
setfacl -m u:postgres:rx /etc/parsec/macdb
setfacl -d -m u:postgres:r /etc/parsec/capdb
setfacl -R -m u:postgres:r /etc/parsec/capdb
setfacl -m u:postgres:rx /etc/parsec/capdb
pdp1-user -l 0:0 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
```

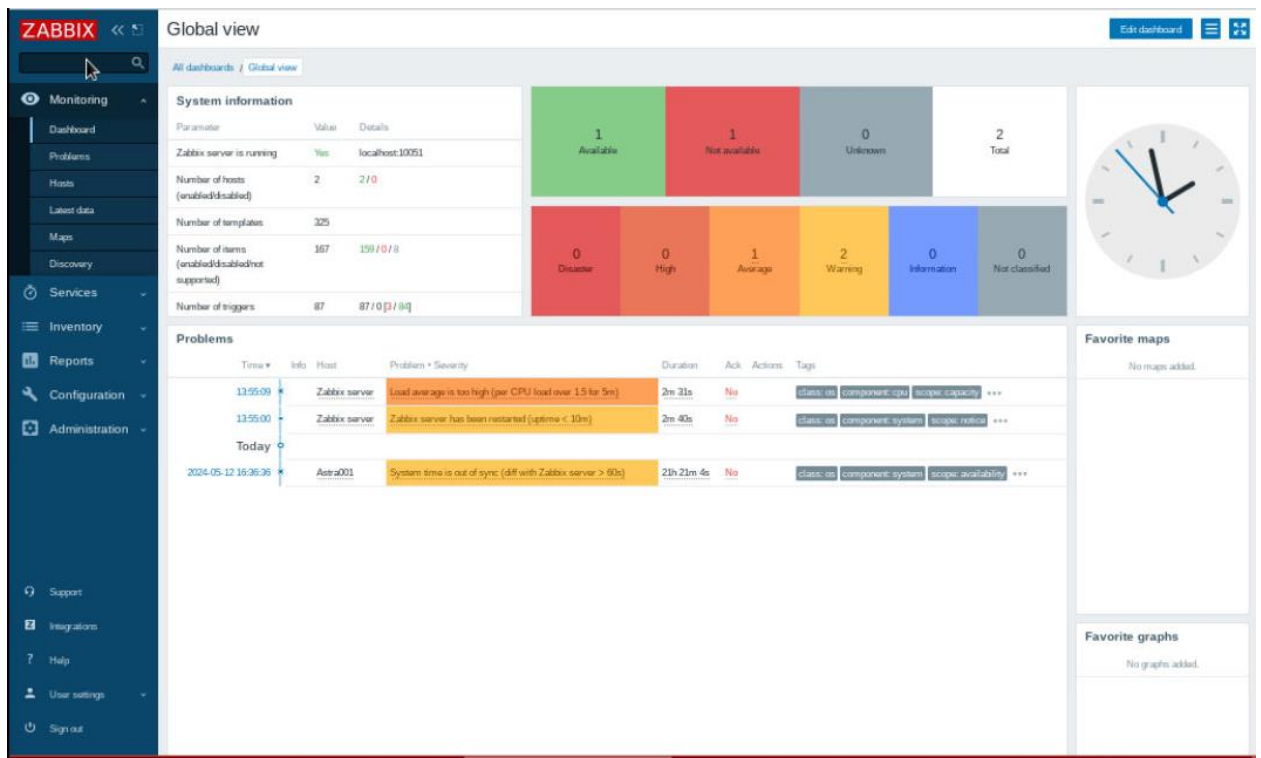
Настройка Zabbix-agent

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.

Настройка

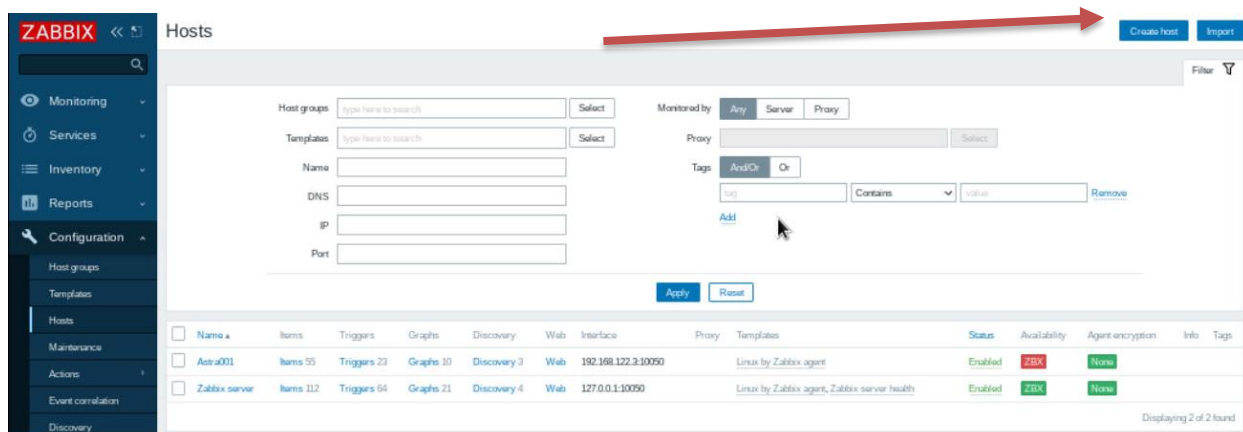
Находясь в WEB-странице zabbix



- 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 хосты, 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: Astra001

Visible name: Astra001

Templates: Name: Linux by Zabbix agent, Action: Unlink, Unlink and clear

* Groups: Templates/Operating systems

Interfaces: Type: Agent, IP address: 192.168.122.3, DNS name: , Connect to: IP, DNS, Port: 10050, Default: Remove

Add

Description:


Monitored by proxy: {no proxy}

Enabled: ☒

Buttons: 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



	Name	Items	Triggers	Graphs	Discovery	Web	Interface	Proxy	Templates	Status	Availability	Agent encryption	Info	Tags
<input type="checkbox"/>	Astra001	Items 55	Triggers 23	Graphs 10	Discovery 3	Web	192.168.122.3:10050		Linux by Zabbix agent	Enabled	ZBX	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

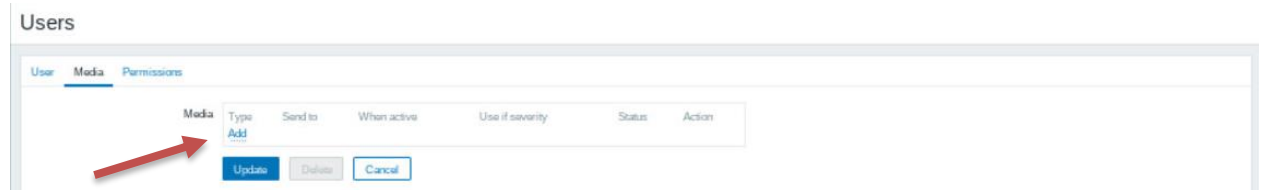
Для редактирования метрик или триггеров необходимо кликнуть на items или triggers

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

Настройка сообщений о ключевых событиях

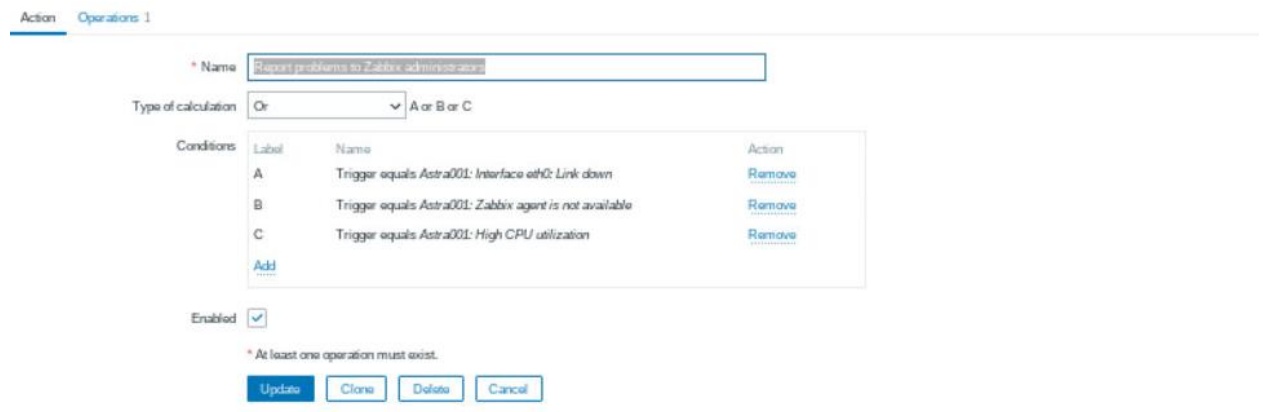
Необходимо зарегистрировать почту с которой будет писать заббикс сервер и добавить ее в Administrations>Media types (на примере Mail.ru, также необходимо получить пароль от почты для внешних приложений)

Далее необходимо добавить почту пользователю admin(Administrations>Users)

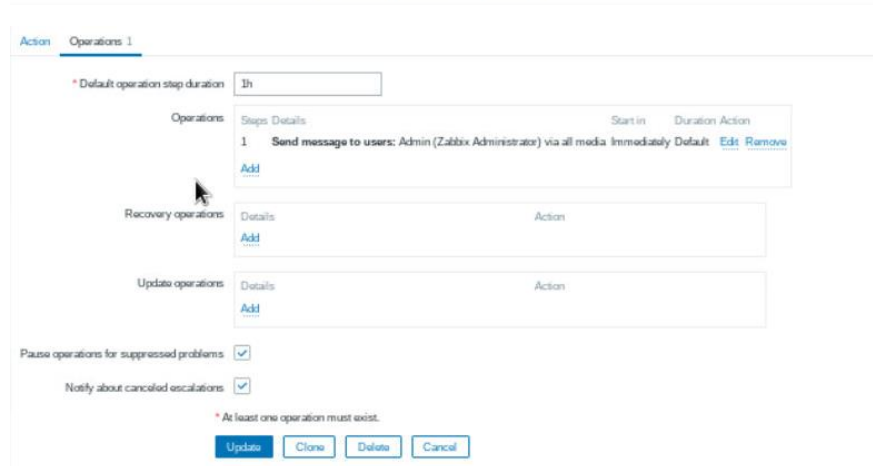


The screenshot shows the 'Users' configuration page with the 'Media' tab selected. A red arrow points to the 'Add' button in the 'Media' section. The 'Media' section includes fields for 'Type', 'Send to', 'When active', 'Use if severity', 'Status', and 'Action'. Below these fields are 'Update', 'Delete', and 'Cancel' buttons.

Сами сообщения настраиваются во вкладке Configuration>Actions>Trigger actions



The screenshot shows the 'Trigger actions' configuration page. The 'Name' field is set to 'Send problems to Zabbix administrator'. The 'Type of calculation' is set to 'Or'. The 'Conditions' section lists three conditions: 'Trigger equals Astra001: Interface eth0: Link down', 'Trigger equals Astra001: Zabbix agent is not available', and 'Trigger equals Astra001: High CPU utilization'. The 'Enabled' checkbox is checked. The 'Update' button is highlighted.



The screenshot shows the 'Operations' configuration page. The 'Default operation step duration' is set to '1h'. The 'Operations' section lists one operation: 'Send message to users: Admin (Zabbix Administrator) via all media'. The 'Recovery operations' and 'Update operations' sections are empty. The 'Pause operations for suppressed problems' and 'Notify about canceled escalations' checkboxes are checked. The 'Update' button is highlighted.

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 с помощью команды

```
sudo wireshark
```

- 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 лабораторной работы 4 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 (для этого установите почтовый клиент, например Evolution)
- g. Опционально: настройте отправку уведомлений в телеграмм/discord

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)
```