УДК 504.064.36 DOI: 10.31673/2518-7678.2020.021220

Гринкевич Г.О., к.т.н.; **Домрачева К.О.,** к.т.н.; **Якимчук С.П.,** студент

НАЛАШТУВАННЯ СИСТЕМИ МОНІТОРИНГУ ZABBIX

Grynkevych G.O., Domracheva K.O., Yakymchuk S.P. Setting up the Zabbix monitoring system.

This article describes how to set up monitoring for different networking and network devices with different monitoring methods. Create templates to quickly configure many of the same observations.

This system has a sufficient number of monitoring methods for network devices: Zabbix agent deployed on monitored network devices for active local monitoring of resources and applications. Agent are supported on: Linux, IBM AIX, FreeBSD, NetBSD, OpenBSD, HP-UX, Mac OS X, Solaris, Windows. JMX is used to monitor applications that use JMX. The contractor that monitors JMX applications is a JAVA daemons, called the JAVA Zabbix Gateway. When Zabbix wants to know the value of the JMX counter, it simply queries the JMX gateway and the gateway does all the work for Zabbix. All requests are made using the Oracle JMX Management API. SNMP monitoring, this method is actually the standartd for many devices and program. The schematic of the underlying architecture is straightforward. It is often the only reasonable way that anyone can extract monitoring information from network switches, UPS batteries. IPMI monitoring is a hardware specification, i.e. "software independent" which means that it is not in any way connected to the BIOS. SSH monitoring, enables Zabbix to run servers without the use of an agent. This special functionality is of great value because it allows us to execute remote commands on a device that does not support Zabbix agent.

Keywords: zabbix, snmp, jmx, ipmi, templates, monitoring, agent, charts, data, items, name, status, SSH.

Гринкевич Г.О., Домрачева К.О., Якимчук С.П. Налаштування системи моніторингу Zabbix.

Налаштування системи моніторингу, підбір найоптимальніших систем для кожного мережевого пристрою. Створення шаблонів, які роблять налаштування значно простішими та використання готових шаблонів.

Огляд відомих методів моніторингу Zabbix: Дана система має достатню кількість різновидів методів моніторингу за мережевими пристроями: Zabbix агент, розгортаються на спостережуваних мережевих пристроях для активного локального моніторингу ресурсів та програм. Агенти підтримуються на: Linux, IBM AIX, FreeBSD, NetBSD, OpenBSD, HP-UX, Mac OS X, Solaris, Windows; JMX використовується для моніторингу програм які застосовують JMX. Виконавець, який виконує моніторинг програм JMX це демон Java, який називається шлюзом Java Zabbix. Коли Zabbix бажає дізнатись значення лічильника JMX, він просто запитує шлюз JMX і даний шлюз виконує всю роботу для Zabbix. Всі запити виконуються з використанням API керування JMX Oracle; SNMP моніторинг, даний метод фактично являється стандартом для багатьох приладів та програм. Схема базової архітектури є прямолінійна. Найчастіше являється єдиним розумним способом який хтось може видобувати інформацію моніторингу з мережевих комутаторів, полиць дискових пристроїв, батарей UPS; IPMI моніторинг, являється специфікацією апаратного рівня, тобто «програмно незалежний» що означає, що воно ніяким образом не ув'язується з BIOS; SSH моніторинг, надає Zabbix запускати сервера без використання агента. Ця особлива функціональність дає велику цінність, так як дозволяє нам виконати віддалені команди на пристрої який не підтримує агентів Zabbix.

Ключові слова: zabbix, snmp, jmx, ipmi, шаблони, моніторинг, агент, графіки, елементи даних, ім'я, статус, SSH.

Гринкевич А.А., Домрачева К.А., Якимчук С.П. Установка системы мониторинга Zabbix.

В этой статье описывается, как настроить мониторинг для различных сетевых и сетевых устройств с различными методами мониторинга. Создавайте шаблоны для быстрой настройки множества одинаковых наблюдений.

Эта система имеет достаточное количество методов мониторинга для сетевых устройств: агент Zabbix, развернутый на отслеживаемых сетевых устройствах для активного локального мониторинга ресурсов и приложений. Агент поддерживается: Linux, IBM AIX, FreeBSD, NetBSD, OpenBSD, HP-UX, Mac OS X, Solaris, Windows. JMX используется для мониторинга приложений, использующих JMX. Подрядчиком, который отслеживает приложения JMX, является демон JAVA, называемый шлюзом JAVA Zabbix. Когда Zabbix хочет узнать значение счетчика JMX, он просто запрашивает шлюз JMX, и шлюз выполняет всю работу для Zabbix. Все запросы выполняются с использованием Oracle JMX Management API. Мониторинг SNMP, этот метод является стандартом для многих устройств и программ. Схема базовой архитектуры проста. Зачастую это единственный разумный способ извлечения информации мониторинга из сетевых коммутаторов и батарей ИБП. Мониторинг IPMI - это аппаратная спецификация, то есть «программно-независимая», что означает, что он никак не связан с BIOS. Мониторинг SSH, позволяет Zabbix запускать серверы без использования агента. Эта специальная функциональность имеет большое значение, потому что она позволяет нам выполнять удаленные команды на устройстве, которое не поддерживает Zabbix агент.

Ключевые слова: zabbix, snmp, jmx, ipmi, шаблоны, мониторинг, агент, графики, элементы данных, имя, статус, SSH.

Вступ

Налаштування системи моніторингу, підбір найоптимальніших систем для кожного мережевого пристрою. Створення шаблонів, які роблять налаштування значно простішими та використання готових шаблонів.

Огляд відомих методів моніторингу Zabbix: Дана система має достатню кількість різновидів методів моніторингу за мережевими пристроями [1]:

- Zabbix агент, розгортаються на спостережуваних мережевих пристроях для активного локального моніторингу ресурсів та програм. Агенти підтримуються на: Linux, IBM AIX, FreeBSD, NetBSD, OpenBSD, HP-UX, Mac OS X, Solaris, Windows.

- JMX використовується для моніторингу програм які застосовують JMX. Виконавець, який виконує моніторинг програм JMX це демон Java, який називається шлюзом Java Zabbix. Коли Zabbix бажає дізнатись значення лічильника JMX, він просто запитує шлюз JMX і даний шлюз виконує всю роботу для Zabbix. Всі запити виконуються з використанням API керування JMX Oracle.

- SNMP моніторинг, даний метод фактично являється стандартом для багатьох приладів та програм. Схема базової архітектури є прямолінійна. Найчастіше являється єдиним розумним способом який хтось може видобувати інформацію моніторингу з мережевих комутаторів, полиць дискових пристроїв, батарей UPS.

- IPMI моніторинг, являється специфікацією апаратного рівня, тобто «програмно незалежний» що означає, що воно ніяким образом не ув'язується з BIOS.

- SSH моніторинг, надає Zabbix запускати сервера без використання агента. Ця особлива функціональність дає велику цінність, так як дозволяє нам виконати віддалені команди на пристрої який не підтримує агентів Zabbix.

Мета даної статті полягає в наглядному прикладі показати і пояснити налаштування Zabbix моніторингу для різних пристроїв. Також створити зручну систему з зрозумілим, логічним налаштуванням.

1. Налаштування моніторингу комп'ютерів та створення шаблону

Для початку налаштуємо моніторинг комп'ютерів.

Ми будемо проводити звичайне пінгування пристроїв в мережі. Також налаштувати моніторинг на перевірку доступності порта RDP.

Так як комп'ютерів в нас буде багато, то ми створимо шаблон. За допомогою нього ми набагато швидше будемо налаштовувати моніторинг комп'ютерів [2].

Для цього потрібно перейти в розділ «Configuration» в даному розділі необхідно вибрати «Templates» і нажати «Create template».

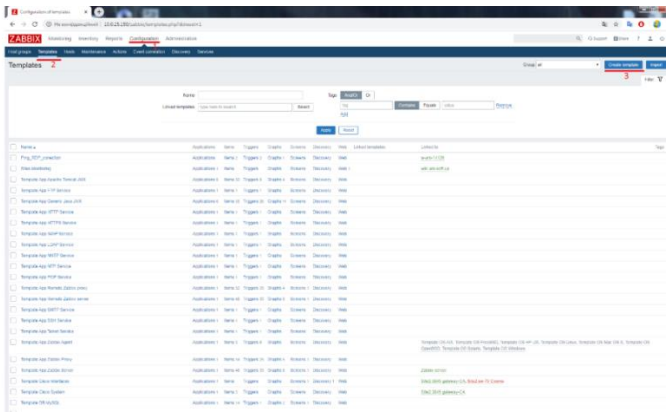

Рис. 1. Розділ «Configuration»

Перед нами появиться вікно, в якому необхідно підписати наш шаблон та добавити його в групу.

Так як це перевірка на доступність RDP та Ping, то я вирішив назвати «Ping_RDP_check» а групу я створив «Remove PC».

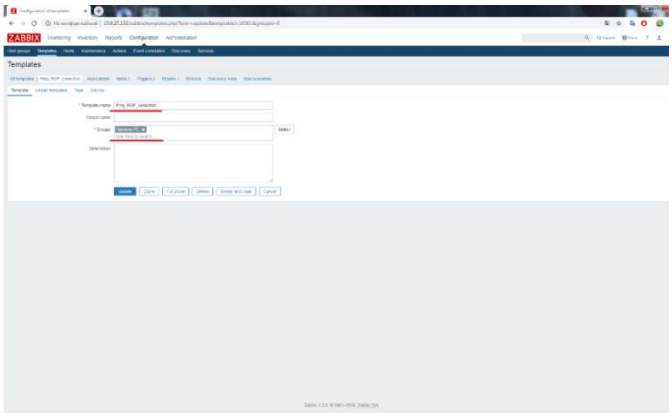

Рис. 2. Розділ «Configuration»

Наступним кроком необхідно створити предмети, які будуть перевірятись.

Для цього необхідно перейти «Items» і натиснути «Create item» [3].

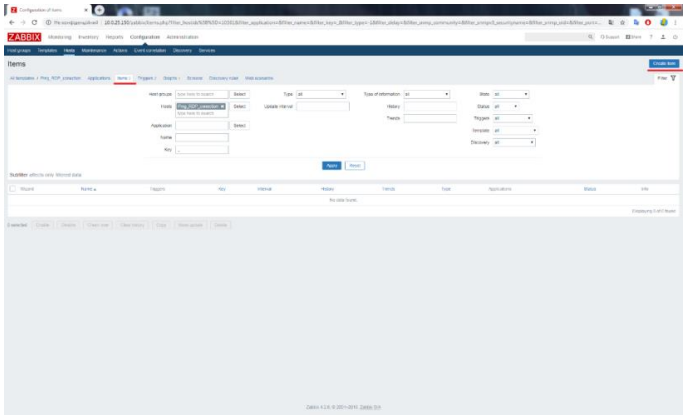

Рис. 3. Розділ «Items»

Відкриється налаштування, в яких необхідно назвати предмет, вибрати та налаштувати спосіб перевірки.

Це в нас буде звичайна перевірка тому я назвав «Ping workstation».

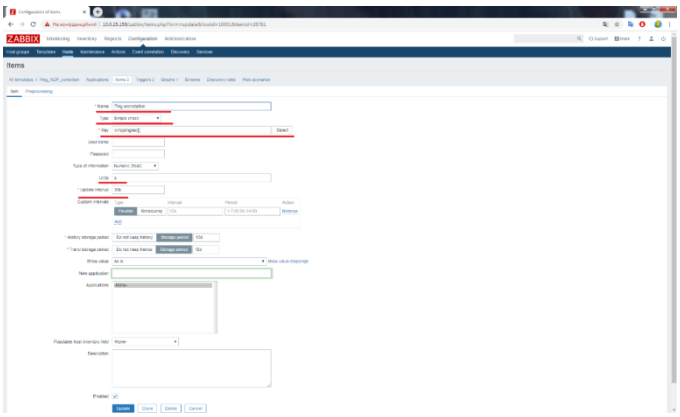

Рис. 4. Розділ «Items»

В розділі «Type» вибрав «Simple check», в «Key» натиснув «Select» та відкрилось вікно в якому можна обрати необхідний ключ та його повний опис. Нас цікавить «icppingsec» даний ключ відправляє сигнал до комп'ютера та показує його час відгуку.

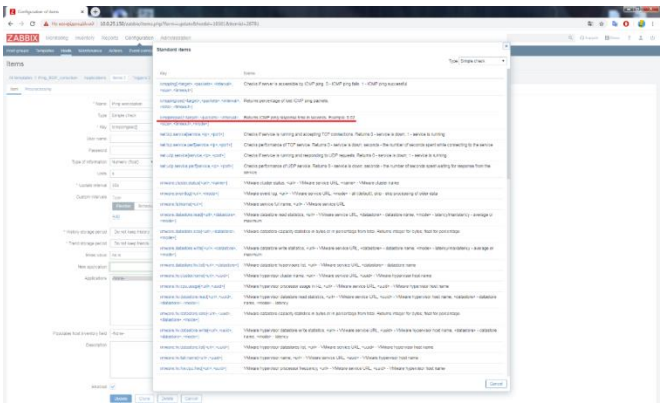

Рис. 5. Розділ «Key»

В розділі «Type of information» ми вибираємо яким типом буде появлятись інформація, нам необхідно вибрати число з комою «Numeric(Float)».

В розділі «Units» ми прописуємо «s» так як час вимірюється в секундах.

І залишається остання критичне налаштування «Update interval» ми вказуємо з якою частотою буде йти перевірка, я вибрав 30 секунд, так як на мою думку це самий оптимальний варіант.

По завершенню наживаємо «Update» в самому низу і ми створили просту перевірку.

Створюємо ще один предмет, для перевірки RDP порта на доступність.

Принцип створення такий же, тільки ключ нам необхідний «net.tcp.service[tcp, 3389]», в якому 3389-порт через який працює RDP а також необхідно вибрати тип даних «Numeric(unsigned)».

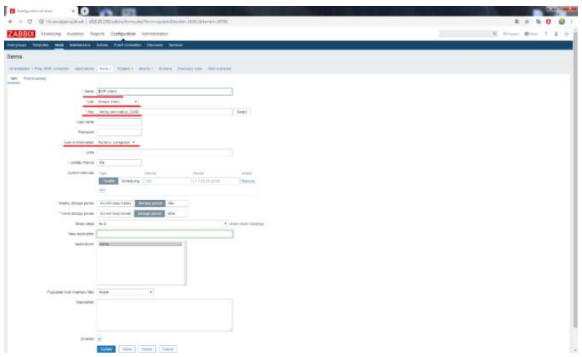

Рис. 6. Розділ «Items»

Далі ми створюємо тригери, вони створюються для того щоб система зала коли необхідно оповіщати про проблему. Для цього переходимо в «Triggers» і нажимаємо «Create trigger».

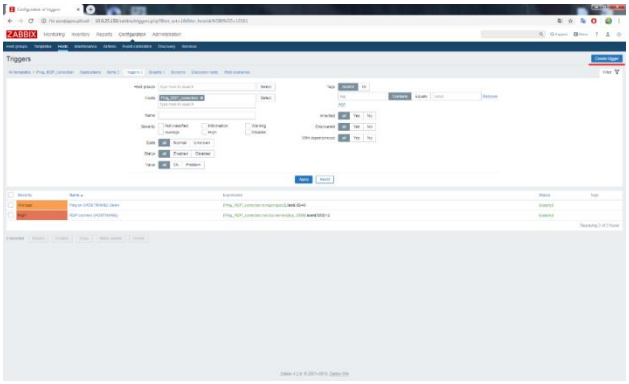

Рис. 7. Розділ «Triggers»

Відкриється вікно в якому ми називаємо тригер та підтягуємо до нього предмет і вказуємо правила.

В «Name» я прописав «Ping on {HOSTNAME} Down», де {HOSTNAME} – в момент оповіщення помилки вказувало ім'я комп'ютера з проблемою.

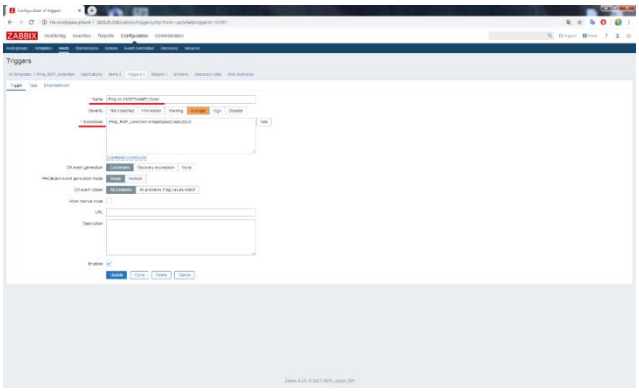

Рис. 8. Розділ «Trigers»

«Expression» ми нажимаємо «Add», після чого відкривається вікно в якому ми вибираємо наш предмет, який відповідає за просту перевірку, в «Function» ми вказуємо що дивись на останній передані дані. «Result» ми вказуємо коли спрацьовує тригер якщо відповідь від комп'ютера не приходить то буде 0. Нажимаємо «Insert» вікно закриється і зберігаємо.

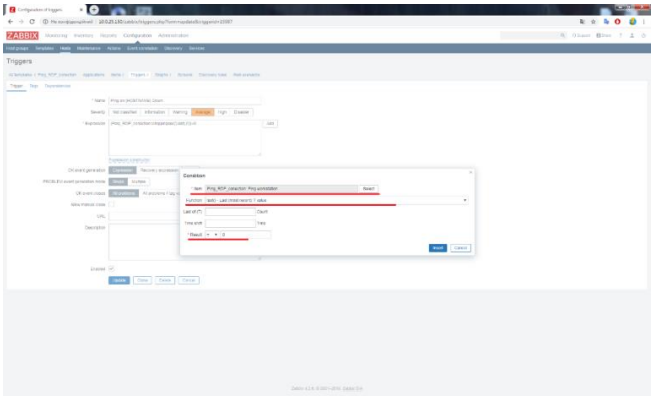

Рис. 9. Розділ «Expression»

Таку ж операцію проводимо для RDP.

Шаблон створено, тепер давайте за допомогою нього налаштуємо моніторинг комп'ютера.

Переходимо в розділ «Configuration» «Hosts» і натискаємо «Create host».

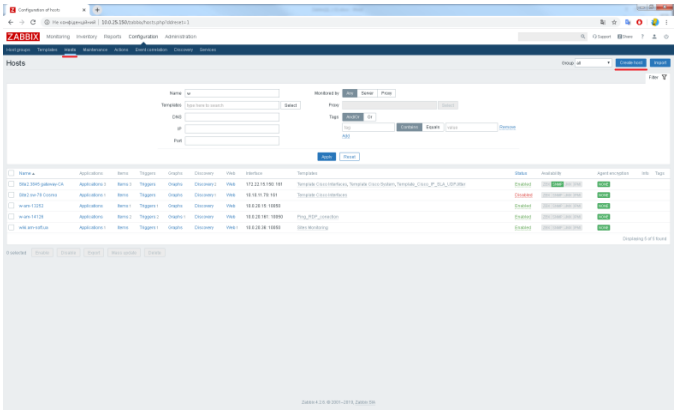

Рис. 10. Розділ «Hosts»

Після чого в нас відкриваються налаштування для нового хоста.

Host name – вказуємо назву яку буде бачити система.

Visible name – назва, яка буде відображатись для користувача

Groups – додаємо в групу «Remove PC»

Далі потрібно вибрати «Agent interfaces» і в «IP address» прописуємо адрес необхідного комп’ютера [4].

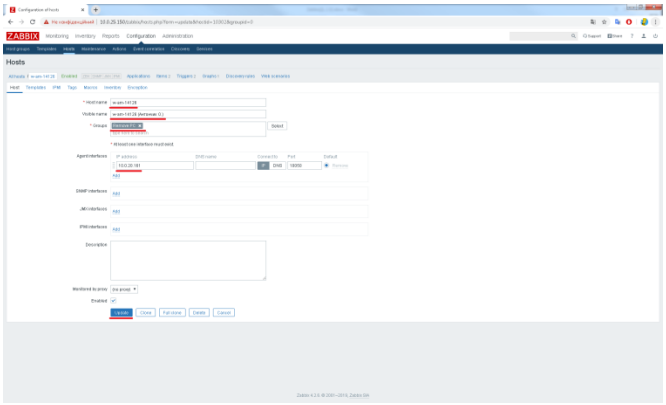

Рис. 11. Розділ «New host»

Далі переходимо в розділ «Templates» де нажимаємо «Select» і появляється вікно з усіма шаблонами, нам необхідно вибрати наш шаблон, який ми створювали. Ставимо навпроти нього галочку і нажимаємо «Select». Вікно закриється і необхідно натиснути «add» і зберегти, після чого шаблон буде додано до нашого хоста.

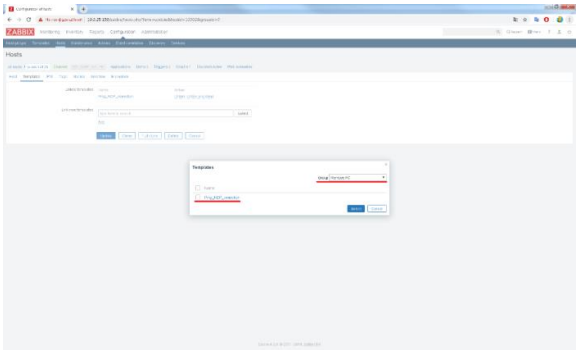

Рис. 12. Розділ «Link new templates»

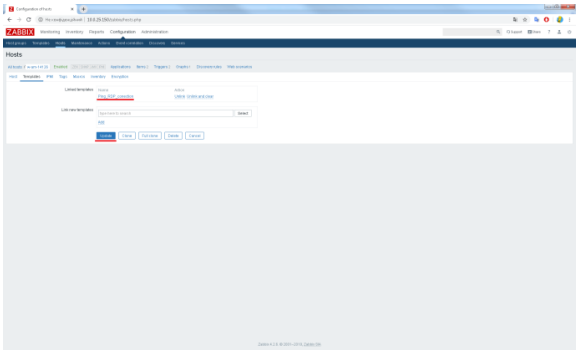

Рис. 13. Розділ «Linked templates»

Для перевірки, можна відкрити створений хост та перейти в пункт «Items» та «Triggers» і перевірити, що створились необхідні елементи.

Дане налаштування завершено, по такому шаблону в подальшому будуть підключатись всі необхідні комп'ютери. Можна було використовувати агент, але в моєму випадку він споживав би набагато більше ресурсів.

2. Налаштування моніторингу серверів та сховищ.

Дане обладнання буде перевірятись за допомогою агентів. Налаштування для різних систем налаштовується по різному. Необхідно моніторити сховище даних під керуванням системи XigmaNAS.

XigmaNAS безкоштовна операційна система для мережефого сховища. Дана система є прямим нащадком проекта Nas4Free який також є мережевим сховищем.

Для початку необхідно підготувати саме сховище для того щоб система Zabbix могла отримувати дані. Для початку, необхідно зайти на Web-інтерфейс системи XigmaNAS.

Необхідно відкрити доступ підключення через SSH. Для цього потрібно перейти в розділ «Службы» «SSH», поставити галочку навпроти в правому верхньому куті для того щоб включити протокол. Також потрібно поставити галочки навпроти: «Определяет использование проверки подлинности…»; «Указывает, позволен ли вход от имени...»

Ще необхідно створити обліковий запис та групу для агента. В розділі «Доступ» «Пользователи групы» вибираємо «Групы». Відкриється вікно, в якому необхідно вказати ім'я нашої групи та описати її призначення, зберігаємо. Повертаємось на один пункт назад і нажимаємо «Пользователи», вказуємо ім'я в розділі «Основная група» вибираєму групу, яку ми створили. Також в розділі «Домашний каталог» прописуємо /home/zabbix та зберігаємо.

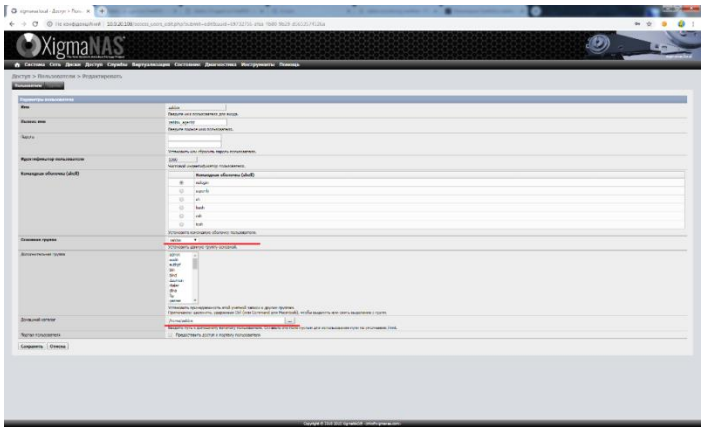

Рис. 14. Розділ «XigmaNAS»

Після того як ми підключились чере SSH нам доступний командний рядок через який ми розпочнемо встановлення агента.

Спочатку можна обновити деякі пакети системи.

 # pkg update

По завершенні оновлення розпочинаємо встановлення агента. Нам необхідно вибрати версію агента таку ж, як версія Zabbix і прописати наступні команди.

 # pkg install zabbix-agent42-agent -команда для встановлення самого агента для версії Zabbix 4.2

Після завершення встановлення агента необхідно зробити так, щоб він міг автоматично запускатись а також змінити власника сервісв, так як створений автоматично користувач, буде видалено після першого ж перезавантаження.

 # cd chown zabbix:zabbix /var/run/zabbix

 # sysrc zabbix_agent_enable=YES - для того, щоб агент автоматично запускався після перезавантаження всього сховища.

Тепер необхідно налаштувати агент. Потрібно скопіювати та перейменувати файл який відповідає за конфігурацію агента.

 # cp /usr/local/etc/zabbix42/zabbix_agent.conf.sample /usr/local/etc/zabbix42/zabbix_agent.conf

Переходимо до редагування, відкриваємо файл.

 # nano /usr/local/etc/zabbix42/zabbix_agent.conf

В даном файлі необхідно знайти рядок

«**#Server=**»

«**#ServerActive=**»

Потрібно прописати IP адрес нашого Zabbix сервера та розкоментувати, для цього потрібно видалити «#».

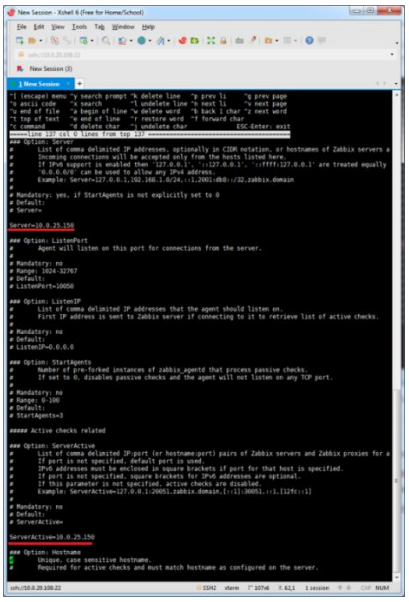

Рис. 15. Розділ «zabbix_agent.conf»

Також необхідно знайти рядок «**#Hostname**» розкоментувати його, та прописати назву та рядок «**LogFile=**» і вказуємо новий шлях «/var/log/zabbix/zabbix_agentd.log».

Зберігаємо файл, та запускаємо агента.

 # service zabbix_agent start

Перевіряємо чи сервіс запустився.

 # service zabbix_agent status

Більша частина налаштувань завершена. Тепер необхідно зайти на Zabbix сервер та створити новий хост, підписати, вибрати спосіб моніторингу «Agent interface», вказати IP адрес нашого сховища і зберегти. Всі необхідні налаштування підтянуться автматично через невеликий проміжок часу.

Висновки

Як можна вже зробити висновок, налаштування моніторингу для кожного окремого завдання є чисто індивідуальним заняттям, так як в кожного різні задачі та підхід до них. В даній статті було детально описано, як налаштувати моніторинг певних мережевих пристроїв. Так як алгоритм загальний здебільшого у всіх не відрізняється, здебільшого відрізняються тільки невеликі модифікації. Їх підганяють користувачі під себе.

Список використаної літератури

1. XigmaNAS - The original open source Network Attached Storage distribution [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: https://www.xigmanas.com/wiki/doku.php. (13.03.2020).

2. Zabbix documentation [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: https://www.zabbix.com/documentation/current/start. (13.03.2020).

3. Zabbix appliance [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: https://www.zabbix.com/documentation/current/manual/appliance. (13.03.2020).

4. Service monitoring [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: https://www.zabbix.com/documentation/current/manual/it_services. (13.03.2020).

Автори статті

Гринкевич Ганна Олександрівна – кандидат технічних наук, доцент, доцент кафедри телекомунікаційних систем та мереж, Державний університет телекомунікацій, Київ, Україна.

Домрачева Катерина Олексіївна – кандидат технічних наук, доцент кафедри телекомунікаційних систем та мереж, Державний університет телекомунікацій, Київ, Україна.

Якимчук Сергій Петрович – студент, Державний університет телекомунікацій, Київ Україна.

Authors of the article

Grynkevych Ganna Oleksandrivna - Candidate of Sciences (technical), Associate Professor, Associate Professor of the Department of Telecommunication Systems and Networks, State University of Telecommunications, Kyiv, Ukraine.

Domracheva Kateryna Oleksiivna - Candidate of Sciences (technical), Associate Professor of the Department of Telecommunication Systems and Networks, State University of Telecommunications, Kyiv, Ukraine.

Yakymchuk Serhii Petrovych - student, State University of Telecommunications, Kyiv, Ukraine.

Дата надходження в редакцію: 29.04.2020 р. Рецензент: д.т.н., доцент В.Ф. Заїка