МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет кібербезпеки, комп'ютерної та програмної інженерії
Кафедра комп'ютерних інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

А.С. Савченко

"___"_____2020 р.

ДИПЛОМНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ "МАГІСТРА"

ЗА СПЕЦІАЛІЗАЦІЄЮ "ІНФОРМАЦІЙНІ УПРАВЛЯЮЧІ СИСТЕМИ ТА ТЕХНОЛОГІЇ (ЗА ГАЛУЗЯМИ)"

Тема: «Саморозгортувана віртуальна машина агент-серверної телеметрії ентерпрайз класу»

Виконав: Топорок Денис Олександрович

Керівник: професор Зіатдінов Юрій Кашафович

Нормоконтролер: _____ Райчев І.Е.

Київ 2020

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра Комп'ютерних інформаційних технологій

Галузь знань, спеціальність, спеціалізація: 12 "Інформаційні технології", 122 "Комп'ютерні науки", "Інформаційні управляючі системи та технології (за галузями)"

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Савченко А.С.
" " 2020 р.

ЗАВДАННЯ

на виконання дипломної роботи студента
Топорком Денисом Олександровичем

1. Тема роботи: " Саморозгортувана віртуальна машина агент-серверної телеметрії ентерпрайз класу"

Затверджена наказом ректора №1891/ст. від 02.10.2020р..

2. Термін виконання роботи: з 05.10.2020р. до 31.12.2020р.

3. Вихідні данні до роботи: агент серверна телеметрія, створення працездатної віртуальної машини.

4. Зміст пояснювальної записки: 1)Проаналізовано потребу телеметрії. 2) Створено віртуальну машину. 3)Проаналізовано найпопулярніші системі моніторингу.

5. Перелік обов'язкового ілюстративного матеріалу: рисунки, діаграма, а також слайди презентації доповіді у PowerPoint.

6. Календарний план-графік

№ з/п	**Завдання**	***Термін виконання***	***Підпис керівника***
1.	Формування теми дипломної роботи, постановка задачі та узгодження з дипломним керівником.	05.10.20- 6.10.20	
2	Формування структури розділів дипломної роботи	06.10.20– 10.10.20	
3	Формування та оформлення першої частини дипломної роботи	11.10.20 – 15.11.20	
4	Збір науково-технічного матеріалу до другої частини дипломної роботи	16.10.20 – 20.10.20	
5	Формування та оформлення другої частини дипломної роботи	08.11.20 – 16.11.20	
6	Розробка програмної частини згідно з завдання дипломної роботи	16.11.20 – 31.11.20	
7	Формування та оформлення третьої частини дипломної роботи	19.11.20 – 31.11.20	
8	Формування звіту та графічних матеріалів	23.11.20 – 08.12.20	
9	Підписання необхідних документів	09.12.20 – 15.12.20	
10.	Підготовка до захисту дипломної роботи	09.12.20 – 22.12.20	

7. Консультація з окремого(мих) розділу(ів) роботи:

Розділ	Консультант (посада, П.І.Б.)	Дата, підпис	
		Завдання видав	Завдання прийняв

8. Дата видачі завдання 05 жовтня 2020р.

Керівник дипломної роботи . _____ Зіатдінов Ю. К.

Завдання прийняв до виконання _____ Топорок Д.О.

(підпис випускника) (ПІБ)

РЕФЕРАТ

Пояснювальна записка до дипломної роботи "Саморозгортувана віртуальна машина агент-серверної телеметрії ентерпрайз класу " містить 79 сторінок, 43 рисунка, 4 таблиці, 6 використаних джерел.

Мета дипломного проекту: застосування передових методів, для досягнення економії ресурсів IT-відділу в компаніях.

Об'єкт дослідження: середні та великі IT компанії.

Предмет: інтеграція віртуальної машини серверного програмного забезпечення.

Метод дослідження: моніторинг ресурсів при обслуговуванні серверного обладнання.

Результат проекту: диск віртуальної машини попередньо налаштованої серверної системи.

Ключові слова: МОНІТОРИНГ ТЕЛЕМЕТРІЯ, АВТОМАТИЗАЦІЯ РОБОТИ IT ВІДДІЛУ, СТВОРЕННЯ ВІРТУАЛЬНОЇ МАШИНИ, НАЛАШТУВАНЯ СИСТЕМИ.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

PCM — процес перетворення аналогового сигналу на цифровий сигнал.

Bash — вдосконалена варіація командної оболонки Bourne shell.

Ping — службова комп'ютерна програма, призначена для перевірки з'єднань.

Traceroute —програма призначена для визначення маршрутів слідування даних.

Active Directory — інтелектуальна служба каталогів корпорації Microsoft.

Troubleshooting — форма вирішення проблем, для ремонту несправних продуктів чи процесів машини або системи.

ВСТУП

Живучи в XXI століття, ми щодня, щогодини стикаємося з потоком інформації, яку отримуємо з різних джерел і наш мозок змушений її аналізувати, сортувати і реагувати.

Але людина не може це робити постійно, а зі збільшенням обсягів інформації і не в змозі. Тому створюються системи, які автоматизують збір даних, їх аналіз та управління процесами.

З розвитком прогресу, в кожній галузі створюються все більш складні гібридні системи моніторингу, контролю та управління з елементами штучного інтелекту.

Зараз дуже важко представити відсутність цих систем на створення яких виділяються величезні кошти.

При цьому уніфікувати ці системи неможливо, оскільки в кожній галузі об'єкти застосування індивідуальні, унікальні параметри налаштування при їх впровадженні.

Їх розвиток почався до нашої ери і не мав окремого усвідомленого напряму, але в наші дні це стало невід'ємною частиною розвитку.

Розділ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Телеметрія

1.1.1 Понятя телеметрії

Телеметрія (від грец. Τῆλε «далеко» + μέτρεω - «вимірюю») - отримання інформації про значення вимірюваних параметрів (напруги, струму, тиску, температури і т. п.). Контрольованих і керованих об'єктів різними методами і засобами «метрон» - «вимір».

Отримання даних здійснюється зазвичай через датчики телеметрії (в яких можливе встановлення датчику зв’язку для роботи с телеметричними системами), або зовнішні пристрої зв'язку з об'єктом, до яких підключаються звичайні датчики.

У телеметрії суть полягає в тому, що вимірювана величина, попередньо перетворена в струм або напругу, додатково перетворюється в сигнал, який потім передається по каналу зв'язку. Таким чином, передається не сама вимірювана величина, як у Джеймса Ванна з манометром, а еквівалентний їй сигнал, параметри якого вибирають так, щоб спотворення при передачі були мінімальними.

Отже, телеметрія - це збір даних вимірювань у віддалених точках від централізованого серверу та їх автоматична передача до приймального обладнання (модему) для моніторингу. Датчики, які потребують для роботи зовнішні налаштування з набором даних для розшифрування, потрібен аналог телеметрії, телекоманди .

Даний термін зазвичай відноситься до категорії бездротових систем передачі даних (зокрема радіо, інфракрасніх та ультразвукових), а також містить дані, які надіслані через телефону або комп'ютерну мережу, оптична лінія зв'язку або інші дротові комунікації, такі як RS-232, RS-485, RS-422 та носії ліній електропередачі. Більшість нинішніх телеметричних систем мають переваги низької вартості та повсюдної дії мереж GSM зв’язку, використовуючи SMS для прийому та передачі пакетних даних телеметрії.

Телеметр - це фізичний пристрій , що застосовуються в телеметрії. Він зазвичай має у своєму складі датчик , лінію передачі між датчиком та механізмом обробки та пристрій відображення, запису або управління.

Телеметрія може бути комутована, щоб дозволити передачу безлічі потоків даних у фіксованому кадрі .

1.1.2 Історія створення

Початок розвитку промислової телеметрії лежить в епоху пари, хоча в той час датчик не називався телеметром. Прикладом є доповнення Джеймса Ватта (1736-1819) до своїх парових машин для моніторингу з (близької) відстані, таких як манометр ртутного тиску та регулятор повітряної кулі.

Хоча оригінальний телеметр відносився до дальномірного приладу (далекомірний телеметр), до кінця 19 століття той самий термін широко використовувався інженерами-електриками, застосовуючи його, при відношенні до електричних приладів, що вимірюють багато інших величин, крім відстані. Загалом телеметри включали такі датчики, як термопара, термометр опору та електричний тензометр, а також, пристрій виведення , такий як Семюел Морзе з телеграфним ехолотом і реле . У 1889 р. це змусило автора в провадженні Інституту цивільних інженерів припустити, що термін далекоміра може бути замінений на тахеометр .

У 1930-х роках використання електричних телеметрів швидко зростало. Електричний тензодатчик широко використовувався в ракетних та

авіаційних дослідженнях, а радіозонд був винайдений для метеорологічних вимірювань. Поява Другої світової війни дала поштовх розвитку промисловості, і відтепер багато з цих телеметрів стали комерційно життєздатними.

Наслідуючи ракетні дослідження, радіотелеметрія використовувалась у міру того, як розпочиналося дослідження космосу. Космічні апарати знаходяться в такому місці, де фізичний зв’язок неможливий, і радіо чи інші електромагнітні хвилі (наприклад, інфрачервоні лазери) залишаються єдиним життєздатним варіантом телеметрії. Під час пілотованих космічних місій він використовується для моніторингу не тільки параметрів транспортного засобу, а й здоров'я та підтримки життя космонавтів. Під час "холодної війни" телеметрія знайшла застосування в шпигунстві. Американська розвідка виявила, що вони можуть контролювати телеметрію з радянських випробувань ракет, будуючи власний телеметр для перехоплення радіосигналів і, отже, дізнатися багато нового про радянські можливості.

1.1.3 Галузі використання

Інформація про дистанційне вимірювання по дроту виникла ще у 19 столітті. Одним з перших схем передач даних була розроблена в 1845 році між російським зимовим палацом і штабом армії. У 1874 році французькі інженери побудували на Монблані систему датчиків погоди та глибини снігу, які передавали інформацію в режимі реального часу до Парижа . У 1901 році американський винахідник С. Міхалков запатентував сельсин, схему для передачі обертання синхронізованою інформації на відстані. У 1906 році був побудований набір сейсмічних станцій з телеметрією до Пулковської обсерваторії в Росії. У 1912 році співдружність Едісона розробила систему телеметрії для контролю електричних навантажень на своїй електромережі. Панамський канал (закінчений 1913–1914) використовував розгалужені телеметричні системи для контролю шлюзів і рівня води.

Бездротова телеметрія вперше з’явилася в радіозонді , розробленому одночасно в 1930 році Робертом Бюро у Франції та Павлом Молчановим у Росії . Система Молчанова модулювала вимірювання температури та тиску шляхом перетворення їх у бездротову азбуку Морзе . Німецька ракета V-2 використовувала систему примітивних мультиплексованих радіосигналів під назвою "Мессіна", щоб повідомити про чотири параметри ракети, але вона була настільки ненадійною, що колись Вернер фон Браун заявив, що корисніше спостерігати за ракетою через бінокль.

У США та СРСР система Мессіни була швидко замінена на кращі; в обох випадках на основі імпульсно-позиційної модуляції (PPM). Ранні радянські ракетно-космічні системи телеметрії, які були розроблені наприкінці 1940-х, використовували або PPM (наприклад, систему телеметрії Tral, розроблену OKB-MEI), або модуляцію тривалості імпульсу (наприклад, систему RTS-5, розроблену NII- 885). У Сполучених Штатах на ранніх роботах застосовували подібні системи, але згодом їх замінила імпульсно-кодова модуляція (PCM) (наприклад, у зонді Марса Mariner 4). Пізніше радянські міжпланетні зонди використовували надлишкові радіосистеми, передаючи телеметрію за допомогою PCM на дециметровій смузі та PPM на сантиметровій смузі.

Галузі використання:

- **Метеорологія**

Телеметрія використовується метеорологічними аеростатами для передачі метеорологічних даних з 1920 року.

- **Нафтогазова промисловість**

Телеметрія використовується для передачі механіки буріння та оцінки пластової інформації в реальному часі, коли бурять свердловину. Ці послуги відомі як вимірювання під час буріння та зруб під час буріння . Інформація, отримана на тисячі футів під землею під час буріння, надсилається через

свердлильний отвір на поверхневі датчики та програмне забезпечення для демодуляції. Хвиля тиску перетворюється в корисну інформацію після DSP і шумових фільтрів. Ця інформація використовується для оцінки пласта , оптимізації буріння та геокерування .

- **Автогонки**

Телеметрія є ключовим фактором сучасних автогонок, що дозволяє інженерам гонок інтерпретувати дані, зібрані під час випробувань або перегонів, та використовувати їх для правильної настройки автомобіля на оптимальну продуктивність. Системи, що використовуються послідовно, такі як у Формулі 1 , стали вдосконаленими до такої міри, що можна розрахувати потенційний час кола автомобіля, і саме цього часу очікується на зустріч водія. Приклади вимірювань на гоночному автомобілі включають прискорення (сили G) за трьома осями, показники температури, швидкість колеса та зміщення підвіски. У Формулі 1 також реєструється введення водія, щоб команда могла оцінити характеристики водія, і (у випадку аварії) FIA може визначити або виключити помилку водія як можливу причину.

Пізніші розробки включають двосторонню телеметрію, яка дозволяє інженерам оновлювати калібрування автомобіля в режимі реального часу (навіть коли він вибуває на трасі). У Формулі 1 двостороння телеметрія з’явилася на початку 1990-х років і складалася з відображення повідомлень на інформаційній панелі, яку команда могла оновити. Його розвиток тривав до травня 2001 року, коли вперше його дозволили на машинах. До 2002 року команди змогли змінити картографію двигуна та деактивувати датчики двигуна з командного центру, поки машина була на трасі. У сезоні 2003 року FIA заборонила двосторонню телеметрію з Формули-1; однак, ця технологія може використовуватися в інших видах перегонів або на дорожніх автомобілях.

Одностороння система телеметрії також застосовується в R/C гоночному автомобілі для отримання інформації датчиками автомобіля, таких як: об / хв двигуна, напруга, температури, дросель.

- **Транспорт**

У транспортній галузі телеметрія надає значущу інформацію про ефективність транспортного засобу або водія, збираючи дані від датчиків у транспортному засобі. Це робиться з різних причин, починаючи від контролю за дотриманням персоналу, страхового рейтингу до прогнозного обслуговування.

Телеметрія також використовується для зв'язку пристроїв лічильника трафіку з реєстраторами даних для вимірювання потоків руху, довжини та ваги транспортних засобів.

- **Сільське господарство**

Більшість видів діяльності, пов'язаних із добрими врожаями, залежать від своєчасної доступності даних про погоду та грунт. Тому бездротові метеостанції відіграють важливу роль у профілактиці захворювань та точному зрошенні. Ці станції передають базовій станції параметри, необхідні для прийняття рішень: температура повітря та відносна вологість повітря , кількість опадів та вологість листків (для моделей прогнозування захворювань), сонячна радіація та швидкість вітру (для розрахунку випаровування), датчики листя під напругою (WDS) та вологість ґрунту (вирішальне значення для рішень про зрошення).

Оскільки місцевий мікроклімат може суттєво відрізнятися, такі дані повинні надходити з посіву. Станції моніторингу зазвичай передають дані назад наземним радіо , хоча зрідка використовуються супутникові системи. Сонячна енергія часто використовується, щоб зробити станцію незалежною від електромережі.

- **Управління водними ресурсами**

Телеметрія має важливе значення в управлінні водою , включаючи якість води та функції вимірювання потоку . Основні програми включають AMR (автоматичне зчитування лічильників), моніторинг підземних вод , виявлення витоків у розподільних трубопроводах та спостереження за

обладнанням. Наявність даних майже у реальному часі дозволяє швидко реагувати на події в полі. Контроль за допомогою телеметрії дозволяє інженерам втручатися в такі активи, як насоси, та дистанційно вмикати та вимикати насоси залежно від обставин. Вододільна телеметрія - відмінна стратегія впровадження системи управління водними ресурсами.

- **Оборона, дослідження космосу та ресурсів**

Телеметрія використовується в складних системах, таких як ракети, космічні кораблі , нафтові вишки та хімічні заводи, оскільки вона дозволяє здійснювати автоматичний моніторинг, оповіщення та ведення записів, необхідних для ефективної та безпечної роботи. Космічні агентства, такі як NASA , ISRO , Європейське космічне агентство (ESA) та інші установи використовують телеметричні та / або телекомунікаційні системи для збору даних з космічних кораблів та супутників.

Телеметрія життєво необхідна для розробки ракет, супутників і літаків, оскільки система може бути зруйнована під час або після випробування. Інженерам потрібні критичні параметри системи для аналізу (та покращення) продуктивності системи. За відсутності телеметрії ці дані часто були б недоступними.

- **Ракетобудування**

У ракетній техніці телеметричне обладнання становить невід’ємну частину ракетних засобів, що використовуються для моніторингу положення та стану ракети-носія для визначення критеріїв припинення польоту на безпеку дальності (ціль дальності призначена для громадської безпеки). Проблеми включають екстремальне середовище (температура, прискорення та вібрація), енергопостачання , вирівнювання антени та (на великих відстанях, наприклад, у польоті в космос) час подорожі сигналу .

- **Польові випробування**

Сьогодні майже кожен тип літальних апаратів , ракет або космічних кораблів має бездротову систему телеметрії під час тестування. Аеронавігаційна мобільна телеметрія використовується для безпеки

пілотів та людей на землі під час льотних випробувань. Телеметрія бортової системи приладових випробувальних приладів є основним джерелом вимірювань у реальному часі та інформації про стан, що передається під час випробувань пілотованих та безпілотних літальних апаратів.

- **Військова розвідка**

Перехоплена телеметрія була важливим джерелом розвідки для США та Великобританії, коли випробовували радянські ракети; для цього Сполучені Штати експлуатували прослуховувальний пункт в Ірані . Врешті-решт росіяни виявили мережу збору розвідданих США та зашифрували свої телеметричні сигнали для випробувань ракет. Телеметрія також була джерелом для радянських військ, які експлуатували прослуховувальні кораблі в затоці Кардіган, щоб підслуховувати британські ракетні випробування, проведені в цьому районі.

- **Моніторинг енергії**

На заводах, будинках та будинках споживання енергії такими системами, як ОВК (Опалення, вентиляція та кондиціювання) контролюється в різних місцях; відповідні параметри (наприклад, температура) надсилаються за допомогою бездротової телеметрії до центрального місця. Інформація збирається та обробляється, що дозволяє найбільш ефективно використовувати енергію. Такі системи також полегшують технічне обслуговування .

- **Розподіл ресурсів**

Багато ресурсів потрібно розподілити на широких територіях. Телеметрія корисна в цих випадках, оскільки дозволяє логістичній системі направляти ресурси туди, де вони потрібні, а також забезпечувати безпеку цих активів; Основними прикладами цього є сухі товари, рідини та насипні сипучі речовини.

складах, терміналах доставки, транспортних перевізниках та фабриках.

- **Рідини**

Рідини, що зберігаються в резервуарах, є основним об'єктом постійної комерційної телеметрії. Це, як правило, включає моніторинг резервуарних ферм

на бензинових заводах та хімічних заводах - і розподілених або віддалених резервуарів, які необхідно поповнювати, коли вони порожні (як у резервуарах для АЗС, цистернах для мазуту для опалення або цистернах для хімічних речовин на фермах), спорожняється при повному сполученні (як при видобутку з нафтових свердловин, накопичених відходів та ново вироблених рідин). Телеметрія використовується для передачі змінних вимірювань датчиків потоку та рівня бака, що виявляють рух та/або обсяги рідини за допомогою пневматичного , гідростатичного або диференціального тиску; резервуарний ультразвуковий , радіолокаційний або допплерівський ефект відлуння; або механічні або магнітні датчики.

- **Сипучі речовини**

Телеметрія сипучих твердих речовин є загальною для відстеження та звітування про об'ємний стан та стан бункерів для кормів для зерна та худоби , порошкоподібних або гранульованих продуктів харчування, порошків та гранул для виробництва, піску та гравію та інших сипучих твердих речовин. Хоча технологія, пов'язана з моніторингом ємності з рідиною, також частково застосовується до сипучих сипучих речовин, іноді вимагається звітування про загальну вагу контейнера або інші загальні характеристики та умови, внаслідок більш складних та змінних фізичних характеристик сипучих речовин.

- **Медицина / охорона здоров’я**

Телеметрія застосовується для пацієнтів (біотелеметрія), яким загрожує порушення серцевої діяльності, як правило, у відділенні коронарної допомоги . Спеціалістів телеметрії іноді використовують для спостереження за багатьма пацієнтами в лікарні. Такі пацієнти оснащені вимірювальними, реєструючими та передавальними приладами. Журнал даних може бути корисним для діагностики стану пацієнта лікарями . Функція оповіщення може попередити медсестер, якщо пацієнт страждає на гострий (або небезпечний) стан.

У медико-хірургічних сестрах доступні системи для моніторингу стану серця або для контролю реакції на антиаритмічні препарати, такі як аміодарон .

Нове застосування для телеметрії - у галузі нейрофізіології, або нейротелеметрії. Нейрофізіологія - це дослідження центральної та периферичної нервової систем шляхом реєстрації біоелектричної активності, спонтанної чи стимульованої. При нейротелеметрії (НТ) електроенцефалограма (ЕЕГ) пацієнта віддалено контролюється зареєстрованим технологом ЕЕГ за допомогою сучасного програмного забезпечення для зв'язку. Метою нейротелеметрії є визнання зниження стану пацієнта до появи фізичних ознак та симптомів.

Нейротелеметрія є синонімом безперервного моніторингу ЕЕГ у режимі реального часу і застосовується у відділі моніторингу епілепсії, невро-реанімаційному відділенні, дитячому відділенні інтенсивної терапії та відділенні інтенсивної терапії новонароджених. Через трудомісткий характер постійного моніторингу ЕЕГ НТ, як правило, проводиться у великих академічних навчальних лікарнях із використанням власних програм, до складу яких входять технології R.EEG, персонал IT-підтримки, невропатолог та нейрофізіолог та допоміжний персонал.

Сучасні мікропроцесорні швидкості, програмні алгоритми та стиснення відеоданих дозволяють лікарням централізовано реєструвати та контролювати безперервні цифрові ЕЕГ одночасно кількох важкохворих пацієнтів.

Нейротелеметрія та постійний моніторинг ЕЕГ надають динамічну інформацію про роботу мозку, що дозволяє на ранніх термінах виявити зміни в неврологічному статусі, що особливо корисно, коли клінічне обстеження обмежене.

- **Дослідження та управління рибним господарством та дикою природою**

Телеметрія використовується для вивчення дикої природи і була корисною для моніторингу видів, що перебувають під загрозою на індивідуальному рівні. Досліджувані тварини можуть бути оснащені інструментальними мітками, які включають датчики, які вимірюють температуру, глибину та тривалість занурення (для морських тварин), швидкість та місцезнаходження

(за допомогою пакетів GPS або Argos). Теги телеметрії можуть надати дослідникам інформацію про поведінку тварин, їх функції та навколишнє середовище. Потім ця інформація зберігається (з архівними мітками), або теги можуть надсилати (або передавати) свою інформацію на супутник або портативний прийомний пристрій. Захоплення та маркування диких тварин може загрожувати їм певним ризиком, тому важливо мінімізувати ці наслідки.

- **Роздрібна торгівля**

На семінарі 2005 року в Лас-Вегасі на семінарі було відзначено введення телеметричного обладнання, яке дозволило б торговим автоматам передавати дані про продажі та інвентар до маршрутного вантажного автомобіля або до штаб-квартири. Ці дані можуть бути використані для різних цілей, таких як усунення потреби водіїв здійснити першу поїздку, щоб побачити, які предмети потрібно поповнити запас перед доставкою інвентаря.

Роздрібні продавці також використовують RFID- мітки для відстеження запасів та запобігання крадіжкам крадіжок. Більшість з цих міток пасивно реагують на зчитувачі RFID (наприклад, у касі), але доступні активні теги RFID, які періодично передають інформацію про місцезнаходження базовій станції.

- **Правоохоронні органи**

Апаратура телеметрії корисна для відстеження осіб та майна у правоохоронних органах. Кісточки комір носити засуджені на випробувальному терміні може попередити владу , якщо людина порушує умови його умовно - дострокового звільнення , наприклад, шлях відступу від дозволених меж або відвідування несанкціонованого місця. Телеметрія також дозволила машини для приманки , де правоохоронні органи можуть облаштувати автомобіль камерами та обладнанням для стеження та залишити його десь там, де вони очікують, що його вкрадуть. При викраденні телеметричне обладнання повідомляє місцезнаходження транспортного засобу, що дозволяє правоохоронним органам вимкнути двигун і заблокувати двері, коли його зупиняють співробітники, що реагують.

- **Постачальники енергії**

У деяких країнах телеметрія використовується для вимірювання кількості споживаної електричної енергії. Лічильник електроенергії взаємодіє з концентратором , а останній передає інформацію через GPRS або GSM на сервер постачальника енергії. Телеметрія також використовується для дистанційного контролю підстанцій та їх обладнання. Для передачі даних іноді використовуються системи несучих фазових ліній, що працюють на частотах від 30 до 400 кГц.

- **Соколине полювання**

У соколиному скотарстві "телеметрія" означає невеликий радіопередавач, який несе хижий птах, що дозволить власнику птаха відстежувати його, коли він не входить в поле зору.

- **Тестування**

Телеметрія використовується для тестування ворожих середовищ, небезпечних для людини. Прикладами є сховища боєприпасів, радіоактивні місця, вулкани, глибоке море та космічний простір.

- **Зв'язок**

Телеметрія використовується у багатьох бездротових системах, що працюють від акумуляторів, для інформування персоналу контролю, коли рівень заряду акумулятора досягає низької точки, а кінцевий елемент потребує нових батарей.

- **Гірничодобувна промисловність**

У гірничодобувній промисловості телеметрія виконує дві основні цілі: вимірювання ключових параметрів гірничого обладнання та моніторинг практик безпеки. Інформація, яка надається шляхом збору та аналізу ключових параметрів, дозволяє виявити основні причини неефективних операцій, небезпечних практик та неправильного використання обладнання для максимізації продуктивності та безпеки. Подальше застосування технології дозволяє обмінюватися знаннями та найкращими практиками в організації.

- **Програмне забезпечення**

У програмному забезпеченні телеметрія використовується для збору даних про використання та продуктивність програм та компонентів додатків, наприклад, як часто використовуються певні функції, вимірювання часу запуску та часу обробки, апаратного забезпечення, збоїв у роботі додатків та загальної статистики використання та / або поведінка користувача. У деяких випадках повідомляються дуже деталізовані дані, такі як окремі показники вікон, кількість використаних функцій та окремі терміни функціонування.

Цей вид телеметрії може мати важливе значення для розробників програмного забезпечення для отримання даних з найрізноманітніших кінцевих точок, які неможливо перевірити все власноруч, а також отримання даних про популярність певних функцій і про те, чи слід їм надавати пріоритет чи розглядатися для вилучення. Через занепокоєння щодо конфіденційності, оскільки програмна телеметрія може бути легко використана для профілювання користувачів, телеметрія в користувацькому програмному забезпеченні часто є вибором користувача, який зазвичай представляється як функція дозволу (що вимагає явних дій користувача для її ввімкнення) або вибір користувача під час процесу встановлення програмного забезпечення .

1.2 Моніторинг системи

1.2.1 Необхідність моніторинг системи

Інформаційні системи ускладняються завдяки розвитку IT, а саме: накопичення все більшої кількості інформації та вдосконалення збору та аналізу алгоритми. Архітектура “Інтернету речей” та мікропослуг швидко зростає і виробляє багато нових джерел даних. Коли така кількість джерел включається в певне інформаційна система з'являється новенький клас завдань. Тепер потрібно системному адміністратору пильно стежити за доступністю мережевих ресурсів та балансом їх використання. Структура

споживання цифрового вмісту також суттєво змінилася. Тоді як 20 років тому більшість веб-сайтів сьогодні містили лише прості та невеликі за розміром HTML-сторінки, компанія Netflix, орендуючи фільми та серіали, генерує до третини доби трафіку в США. Обсяги переданих даних постійно зростають - наприклад, ми віддаємо перевагу перегляду відеовмісту у форматі 4K, що вимагає в 2,5 рази більшої пропускної здатності, ніжте саме відео в FullHD. Незабаром вимоги до мережі знову значно зростуть завдяки до запровадження стандарту 8K.Ось чому і інтернет-провайдери, і користувачі потребують дедалі більше обчислювальних ресурсів. Наприклад, Amazon компанії декларує дохід від AWS інфраструктури, яка забезпечує віртуальні сервери в оренду, складає майже 5 млрд. дол. США .АНБ США також купує сервери у великих обсягах .Ризики відмов комп'ютерів та мережі зростають із збільшенням складності інфраструктури. Необхідно використовувати новий підхід для розвитку цих технічних систем, що дозволять мінімізувати кількість відмов шляхом їх прогнозування і вжиття попереджувальних заходів, з одного боку, та забезпечення швидкого реагування на несправність IT-відділу, з іншого боку.

Рішенням є розгортання системи моніторингу, якою можна швидко виявити інфраструктурні збої, прогнозують поломку, виявляють неефективну та відхилену поведінку обладнання. Давайте визначимо “моніторинг” як безперервний процес спостереження за вказаним параметри певного об'єкта, реєструючи їх значення та порівнюючи із зазначеними подані критерії. Відповідно, «система моніторингу» - це спеціалізоване програмне забезпечення, яке здійснює моніторинг параметри програмних послуг та обладнання, їх відповідність зазначеним шаблонам та перевіряє наявність послуг. У разі помилки система моніторингу повідомляє системного адміністратора і, можливо, виконує деякі дії автоматично. Є можливість контролю інфраструктурні показники (доступність вузлів з мережі, завантаження процесора, використання оперативної пам'яті, вільний простір на дисках тощо), а також індикатори обслуговування (черга запису в БД, деякі

зміни біля журналів тощо). Такі системи дозволяють не тільки швидко реагувати на проблеми, але й вирішувати їх заздалегідь, виявляючи потенційні вразливі місця. Це спосіб мінімізації або повного усунення збоїв в інфраструктурі.

1.2.2 Характеристики систем моніторингу

Перші системи моніторингу з'явилися наприкінці 1990-х. На початку У 2000-х існувало три провідні системи - MS SCOM та Nagios. Усі були розроблені для внутрішніх потреб компаній-творців, а потім виросли в великі проекти для широкого кола ділових користувачів. Однак ніхто не очікував такого інтенсивний ріст інфраструктур при розробці цих систем та виснаження їх масштабованості швидко. Системні вимоги стають занадто високими, а експлуатаційні ви трати на моніторинг основної масив узли стають порівнянними з вузлами базових IT-послуг, таких як контролери доменів або дані с склади.

У цій роботі визначаємо та описуємо ключові вимоги до моніторингу нового покоління системи, які можна використовувати в бізнесі.

- **Матеріал і методи**

Щоб вказати ключові вимоги до систем моніторингу, ми повинні описати типових ділових клієнтів програмного забезпечення такого роду.

Як правило, усі компанії, які мають складну IT-інфраструктуру, є потенційними користувачами систем моніторингу. Великі підприємства, такі як банки або урядові структури та IT компанії використовують такі системи дуже часто. Було опитано декілька експертів з питань підтримки та інфраструктури, які зараз використовують моніторинг системи для визначення ключових вимог до них.

На підставі аналізованих інтерв'ю дійшов висновку, що вимоги різняться залежно від масштаб IT-інфраструктури, але не було виявлено зв'язку між типами галузей. Це означає, що групування вимог не є необхідним. Крім того,

опитав IT-спеціалістів кількох компаній, які не мають системи моніторингу, щоб зробити висновок про їхню думку щодо таких систем загалом та щодо застосовність їх у конкретних ділових ситуаціях. Є загальнодоступні дані опису випадків використання систем моніторингу, типових труднощів у їх реалізації та підтримка. Сформульовано ключові вимоги до готової до виробництва системи моніторингу на основі аналізу всіх даних.

- **Опис вимог**

Сучасна система моніторингу корпоративного класу повинна задовольняти принаймні чотири вимоги в порядку зменшення пріоритету:

1. Надійність.
2. Низьке використання ресурсів.
3. Висока зручність та подальша можливість розвитку.
4. Низька вартість володіння.

- **Надійність**

У випадку систем моніторингу надійність означає не лише безперебійну роботу але, перш за все, гарантувати відповідь на несприятливі зміни параметрів, що контролюються. Система може автоматично реагувати на події (наприклад, перезапустити службу у разі відмови) та повідомити IT-персонал про несправність. Надійне надходження повідомлень є важливим фактором для скорочення простою послуги. Але найголовніша функція - це негайно повідомляти IT-персонал, коли будь-який параметр перевищує його регулярний діапазон значень. Це дозволяє IT службі прогнозувати вузькі місця та попереджувати вирішення можливих проблем.

Вкрай важливо збалансувати кількість вхідних повідомлень. Якщо система звертається до системного адміністратора занадто часто, він почне ігнорувати його сповіщення і, як результат, може пропустити повідомлення про серйозну несправність.

Системи моніторингу корпоративного класу збирають величезну кількість даних про хост: починаючи з завантаженням центрального процесора і

закінчується кількістю процесів, що працюють в ОС. Інформування всіх змін таких параметрів є зайвим. Тим не менше, ця інформація може бути необхідний для аналізу змін навантаження та планування розвитку інфраструктури.

Я вважаю, що необхідно виділити щонайменше 3 рівні сповіщень:

• рівень інформації - відображає всі повідомлення;

• лише важливий рівень - показує лише повідомлення про ймовірні проблеми;

• лише критичний рівень - повідомляє лише про виниклі несправності (недоступність сайтів та служб) та великі відхилення критичних параметрів.

Система моніторингу повинна мати можливість групувати повідомлення різними способами та дозволяти адміністратору управління групами та налаштування попереджень. Це робить конфігурацію повідомлень системи фіксації досить гнучкою. Наприклад, критичні повідомлення можуть бути доставлені через кілька каналів зв'язку одночасно (SMS + месенджери + телефонний дзвінок + ...).

Інформаційні повідомлення не потребують індивідуальних сповіщень - їх досить переглянути в Інтернеті інтерфейс або включити до зведеного звіту.

- **Використання ресурсів**

Типовий розмір IT-інфраструктури постійно зростає, а також обсяг даних подає. Відповідно, є більше точок відмов, на які потрібно звернути увагу - отже, система моніторингу повинна контролювати все більшу кількість вузлів.

Отже, другою важливою вимогою є ефективне використання ресурсів: мережевий діапазон ширина, ресурси процесора сервера моніторингу тощо. Витрати на експлуатацію системи моніторингу не повинно бути надмірним.

Якщо системі потрібно щосекунди передавати та приймати величезну кількість даних, запишіть на диску і прочитайте, щоб повідомити про проблему, тоді така система точно буде неефективний - витрати на його функціонування перевищуватимуть витрати на виробничі послуги.

Щоб забезпечити мінімальну функціональність системи моніторингу, необхідно передавати прапор стану (1 байт) та унікальний ідентифікатор

пристрою (GUID, 16 байт). Таким чином, мінімум розмір пакета даних - 17 байт. Пакет обсягу збільшуватиметься із збільшенням кількості інформації передається.

Змоделюємо сценарій моніторингу типової середньої компанії. Припустимо що існує сто контрольованих вузлів. Давайте підрахуємо, скільки буде інформації передається на сервер системи моніторингу в різних робочих режимах.У першому випадку агенти збиратимуть і надсилатимуть лише критичну інформацію - наявність прапору і номер машини. Якщо система запитує хост один раз на 10 секунд, то для 1 година буде надіслана до розміру 17 × 6 × 60 = 6 120 байт = 6 КБ. Тому протягом 1 години система отримає близько 600 КБ даних взагалі, що не так вже й багато. У другому випадку агенти збирають і передають всі можливі параметри про хоста: кількість процесів та їх перелік, використання процесора та оперативної пам'яті, вільний простір на дисках. Досвід експлуатації існуючих систем моніторингу дозволяє оцінити обсяг такого пакету в 2 Мб. Тоді кожен хост передасть приблизно 2 × 6 × 60 = = 720 МБ даних на годину. Отже, усі дані, оброблені протягом години, будуть розміром приблизно 70 ГБ. Цей об’єм вже досить великий, і ми повинні пам’ятати, що це кількість вузлів збільшення обсягу переданих даних також збільшиться. Якщо ми збираємо не тільки інформацію про самого хоста.

- **Зручність і подальший розвиток**

Складні системи, такі як система моніторингу, вимагають серйозних витрат на впровадження та технічне обслуговування. Тому важливо зробити систему максимально зручною в процесі встановлення та під час повсякденної роботи основного її користувача - системи адміністратор. Іншою частиною функцій системи моніторингу є можливість її подальшого використання розробка та адаптація до певних завдань. Якщо організації потрібно контролювати деякі конкретні показники, наприклад кількість пакетів, що надходять від конкретного хоста над мережі або температура на одному з датчиків, підключених до віддаленого комп'ютера, хороша система моніторингу повинна

дозволяти це робити з мінімальними технічними витратами часу та зусиль персонал.

Крім того, система моніторингу повинна бути універсальною. Якщо системний адміністратор потребує одну систему для моніторингу параметрів обладнання, а іншу - для контролю стану Інтернет серверів, тоді таким комплексом систем буде досить складно адмініструвати. Сам моніторинг є реалізацією з модульною системою, яка дозволить налаштувати агент для різних завдання. Оскільки сучасні мови програмування дозволяють легко створювати міжплатформенний код, ми також проектувати архітектуру без прив'язки до певної операційної системи. Підсистема звітності генерує звіти у різних форматах і дозволяє автоматично парувати їх доставку. Ця підсистема повинна мінімізувати взаємодію з інтерфейсом системи.

- **Вартість володіння**

Ця вимога описує не тільки вартість ліцензії, але й вартість інфраструктура для системи. Тут ми розглядаємо питання окремо, оскільки це вартість ліцензії часто перевищує вартість решти необхідної інфраструктури. Більшість систем моніторингу поширюються за однією з безкоштовних ліцензій, але комерційних рішення від HP, Microsoft або IBM приносить найбільшу частину витрат у ліцензії, а не у вартості розгортання інфраструктури.

На додаток до всього вищезазначеного, комерційні ліцензії здебільшого забороняють модифікацію програмне забезпечення, за винятком того, яке вже включене до нього виробником. Наприклад, DB Модуль моніторингу не входить до складу SCOM, і його вартість коштує кілька тисяч доларів США це від зовнішніх постачальників. Вартість розробки власного модуля не набагато нижча але залежить від рівня кваліфікації команди розробників.

Важливо мати на увазі, що основна вартість систем моніторингу полягає не в вартість ліцензії, але витрати на роботу системи та підтримку; не лише під час реалізації але також під час подальшої експлуатації. У безкоштовних системах моніторингу відсутні витрати на ліцензування, але існують не менші, а часто

навіть більші витрати на кваліфікованих спеціалістів необхідних для впровадження та підтримки системи.

Розділ 2. ВИБІР ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Найпопулярніші системи моніторингу

В останні десятиліття, поряд з розвитком інформаційних систем і мереж, так само розвивалися і системи моніторингу. Чому ж я вибрав саме Zabbix? Загалом, всі системи моніторингу поділяються на кілька груп - платні і безкоштовні, масштабовані і не масштабовані, що працюють по одному-двох протоколів моніторингу або здатні забезпечувати безліч видів моніторингу. З платними і безкоштовними системами все очевидно. Платні системи часто створюються з прив'язкою до конкретного виробника серверного або мережевого устаткування, або під конкретну виробничу задачу (і це вже буде система управління і контролю виробничих процесів). Такі системи, як правило, не масштабуються (не мають можливості вирішувати завдання, не передбачені базовим функціоналом). Існують платні системи моніторингу, масштабовані під різні завдання моніторингу, але вони мають високу вартість і впроваджуються в корпоративних мережах і на виробництвах в форматі інтегрування системи під завдання замовника. З безкоштовними системами ситуація інша. Такі системи або обмежені в функціоналі і надійності, або вимагають серйозних знань від адміністраторів для забезпечення тієї самої масштабованості. Так чи інакше, безкоштовні системи не мають інсталяцій «з коробки» - систему в будь-якому випадку доведеться встановити, адаптувати і налаштувати під поставлені завдання.

У таблиці 2.1 наведу невелику порівняльну таблицю існуючих на сьогоднішній день систем моніторингу.

Табл. 2.1

Існуючі на сьогоднішній день системи моніторингу

Платні	
PRTG	Рішення для моніторингу великих корпоративних мереж. Країна виробник - Німеччина.
OSG (Orange)	Рішення для обслуговування великих корпоративних мереж. Чудова в питаннях інтеграції під технічне завдання замовника. Країна виробник -Росія.
Solar Winds	Рішення для обслуговування великих корпоративних мереж. Країна виробник - США.
SCOM	Рішення для обслуговування серверної інфраструктури Microsoft Windows Server. Країна виробник - США.
Wats UP Gold	Рішення для моніторингу невеликих мереж. Має ряд недоліків в порівнянні з іншими платними аналогами. Країна виробник - США.
Cisco Prime	ПО для обслуговування мереж, побудованих на устаткуванні Cisco. Країна виробник - США.
SNMPc	Вузькоспеціалізоване рішення для моніторингу устаткування по протоколах SNMP і ICMP. Країна виробник -Росія (Agneco SNMPc), СШФ (SNMPc Network Manager).
Безкоштовні (вільно поширювані)	
Zabbix	Універсальна, стабільна, масштабована, широкий функціонал, можливість створення своїх інструментів. Країна виробник -Латвія.
Prometheus	Універсальна, стабільна, має спеціалізовану базу даних, здатну витримати високі навантаження. Відкритий проект.
Nagios + Cacti	Своєрідна система, що запускається і підтримується в ручному режимі.
10 Strike	Обмежений функціонал, demo- версія. Розширений функціонал - у платному режимі. Підходить для невеликих мереж. Країна виробник - Росія.
Ping	Універсальний інструмент перевірки доступності мережевих вузлів, вбудований у будь-яку ОС. Підходить для вирішення украй вузького круга завдань.

2.2 Переваги та можливості Zabbix

Zabbix увібрав в себе кращі риси сучасних безкоштовних систем моніторингу. Це відкрита система, актуальні версії стабільні і надійні, система володіє стовідсотковою масштабованість (навіть якщо якийсь унікальний інструмент моніторингу відсутня - його завжди можна сконструювати самостійно). Однак, експлуатація Zabbix вимагає певних знань. Плюс освоєння системи Zabbix в тому, що отримані знання універсальні, і знайдуть застосування в багатьох суміжних областях IT-індустрії.

Також, некоректно порівнювати платні і безкоштовні системи моніторингу, оскільки у платних систем присутній функціонал не тільки моніторингу, але і управління обладнанням.

Концепція Zabbix, що відображає багатозадачність і універсальність системи, зображена на Рис. 2.1.

Рис 2.1. Концепція Zabbix як активної системи моніторингу

Система моніторингу Zabbix, за допомогою спеціальних мережевих протоколів і інструментів (ICMP, SNMP, TCP / UDP, скрипти і агенти) збирає з об'єктів моніторингу різні параметри, що характеризують їх працездатність. Зібрані дані упорядковуються в базі даних Zabbix, аналізуються і візуалізуються в зручних для сприйняття Адміністратором форматах (графіки, карти мереж).

За умови відхилення одержуваних параметрів від встановлених еталонних значень (при спрацьовуванні тригерів) - спрацьовують інструменти оповіщення адміністратора системи моніторингу або сконструйовані адміністратором скрипти. Оповіщення може бути візуальним, звуковим, у вигляді електронного листа - передбачено кілька різних видів реакції на зафіксовані події (Рис. 2.2):

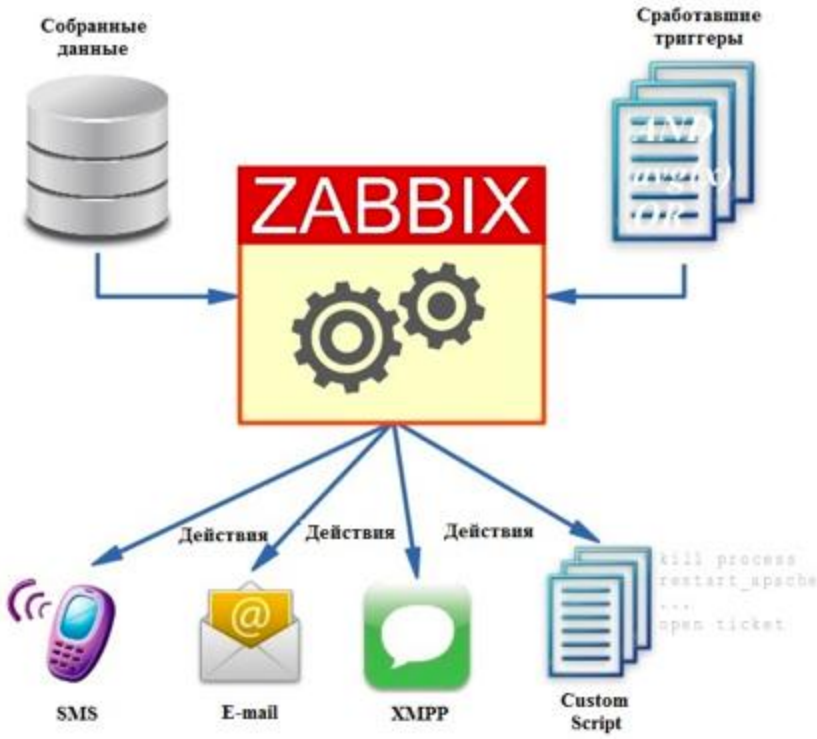

Рис. 2.2. Варіанти реакції системи на зафіксовані події

Очевидно, що система надає масу можливостей для моніторингу самих різних процесів, що відбуваються в інформаційних мережах. Розглянемо технології моніторингу, які підтримує Zabbix.

Класичний моніторинг мережевих пристроїв здійснюється за протоколами ICMP і SNMP.

Протокол ICMP служить для простої перевірки доступності мережевого вузла, з можливістю додаткового аналізу відсотка втрачених ICMP-пакетів і часу затримки. Цей протокол є базовим елементом будь-якого моніторингу, і використовується повсюдно.

Протокол SNMP - протокол, створений для моніторингу та управління мережевими пристроями. Нас буде цікавити виключно моніторинг, оскільки Zabbix не є системою управління. Всі сучасні мережеві пристрої без винятку підтримують протокол SNMP. На мережному диску існують (запрограмовані при виробництві) бази об'єктів SNMP, або так звані MIB-бази (management information base). У MIB-базах містяться як статичні об'єкти, які не змінюються в процесі роботи пристрою (інвентарні номери, моделі і назви пристроїв), так і динамічні, які з плином часу змінюють свої значення (завантаження трафіком мережевого інтерфейсу, температура і завантаження процесора, рівні помилок , і багато іншого). Виробники розміщують в мережі Інтернет документацію на MIB своїх мережевих пристроїв - по суті, це структуровані текстові файли або групи файлів, які містять інформацію про всі SNMP-параметрах пристрою, до якого вони належать, і їх можливі значення. Кожен об'єкт в MIB має свій унікальний цифровий адресу OID (Object IDentificator) і ім'я Object Name. Система моніторингу звертається до інших мережних пристроїв по протоколу SNMP і запитує цифрові значення конкретних OID. Зрозуміло, потрібно знати, який OID в MIB-базі зберігає в собі ту чи іншу значення. Для цього потрібно вміти працювати з MIB-файлами (і документацією виробника), аналізуючи які, можна отримати необхідні для вирішення поставлених завдань OID. Протокол SNMP представляє масу можливостей для отримання самої, здавалося б, несподіваною корисної інформації при опитуванні мережевих пристроїв. Наприклад, при опитуванні джерел безперебійного живлення (якщо вони мають мережеві інтерфейси) - можна отримати дані про температуру термодатчика і значення напруги в мережі електроживлення. При опитуванні

комутаторів, можна отримувати таку специфічну інформацію, як рівень помилок на мережевому інтерфейсі, а при роботі з маршрутизаторами - отримувати статистику IP SLA тестів (технологія моніторингу якості послуг зв'язку). Протокол SNMP є незамінним при моніторингу мережевого обладнання.

Доступність мережевих сервісів здійснюється за допомогою відкриття TCP-сесій на порт тестованого мережевого сервісу (мова йде про так званих безагентних перевірках, коли на стороні тестованого сервісу немає додаткового ПЗ, і аналізується лише можливість встановлення TCP-з'єднання). Для розгорнутого моніторингу серверів ОС Windows і Linux існує технологія так званих агентів, коли на об'єкт моніторингу (сервер) встановлюється додаткове ПО (Zabbix-агент), яке і відправляє серверу моніторингу Zabbix необхідні дані.

На Рис. 2.3 приведена схема навчального стенду і об'єктів, моніторинг яких буде організований і вивчений в рамках даного курсу:

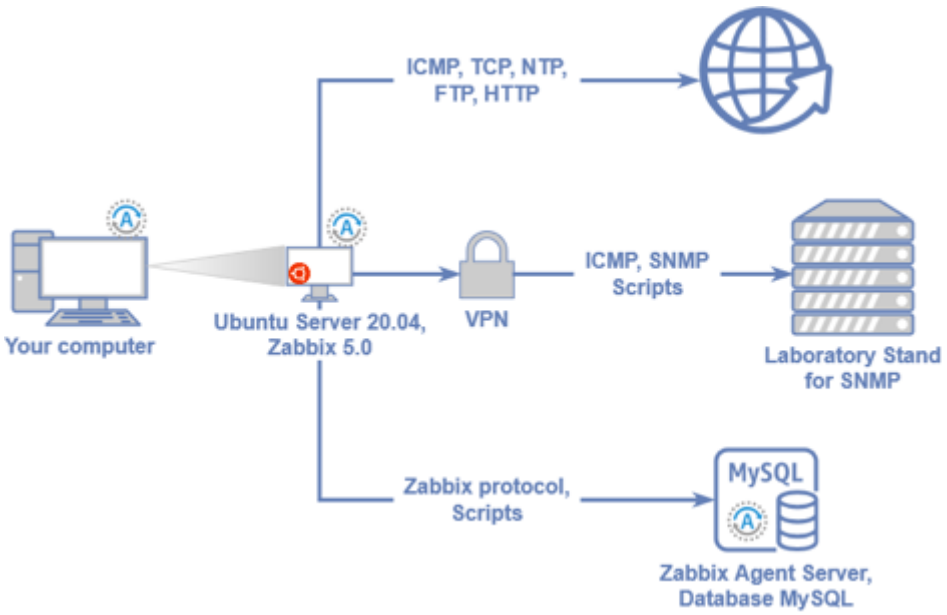

Рис. 2.3. Схема навчального стенду технології моніторингу

Необхідно пам'ятати, що використання будь-який, навіть самої просунутої системи моніторингу при обслуговуванні мережі не вирішує всіх задач, що стоять перед адміністратором.

Для отримання повноцінної картини стану мережі необхідно мати так же сервер Syslog для збору логів обладнання, сервер Netflow для аналізу трафіку, що передається по мережі (Netflow - застаріла технологія, зараз проводяться рішення для аналізу трафіку аж до 7-го рівня моделі OSI з аналізом сигнатур, тоді як Netflow охоплює тільки 3-й і 4-й рівень), софт для обліку та адміністрування конфігурацій мережевого обладнання та програмного забезпечення, ПО для обліку IP-адресного простору організації, ПО для обліку устаткування «складу» (серійні номери, моделі, дати і місця установки, інвентарні номери, дані про накладні та інше). Zabbix, як і майже всі платні рішення, не володіють таким функціоналом. Потрібно чітко усвідомлювати - до обслуговування мережевої інфраструктури потрібен комплексний підхід.

Існують так само системи оркестрації обладнання, що виробляють комплексне обслуговування і управління обладнанням. З платних рішень можна привести в приклад моновендорне ПО - Cisco Prime (обслуговування обладнання Cisco), D-View (Обслуговування комутаторів D-Link), eSight (обслуговування обладнання Huawei, так само заявлена інтеграція з обладнанням сторонніх виробників), JUNOS Space (обслуговування обладнання Juniper). З безкоштовних варто згадати Ansible - систему дистанційного конфігурування груп пристроїв по Telnet і SSH, і NOC Project - масштабну вітчизняну систему комплексного контролю і обслуговування мережевого обладнання в наймасовіших рішеннях, на великих мережах.

Далі перерахуємо інструменти, присутні в Zabbix і деяких платних системах моніторингу, але не увійшли в роботу через екзотичність, складності реалізації саме в Zabbix, або неможливості організувати лабораторні умови для їх вивчення. Сюди відносяться веб-моніторинг по протоколу HTTP (аналіз веб-сторінки на наявність певного текстового контенту), виконання віддалених команд по протоколам Telnet і SSH, зі збереженням результатів виконання цих команд в базу даних Zabbix, моніторинг серверів за технологією IPMI, Java-додатків, і платформ VmWare.

Система моніторингу Zabbix створена під ліцензією GPL (General Public License). Це означає, що його вихідний код вільно поширюється і доступний для необмеженого кола осіб. Інсталяція системи можлива на операційних системах сімейства Linux - Red Hat, Debian, Ubuntu. В даній роботі буду використовувати операційну систему Ubuntu 20.04 LTS (Focal Fossa) - даний дистрибутив стабільний, гнучкий в налаштуванні, є актуальним за версією, і репозиторій Zabbix пропонує готовий до установки пакет для цього дистрибутива. Як середовище віртуалізації я використовую Oracle VM VirtualBox (на момент написання розділу - версії 6.0), оскільки даний програмний продукт допускає безкоштовне використання в освітніх цілях.

Розділ 3. РЕАЛІЗАЦІЯ ПРОЕКТУ

3.1. Налаштування системи

3.1.1. Встановлення Ubuntu

Отже, потрібно скачати і встановити на свій ПК з офіційного сайту virtualbox.org актуальну версію середовища віртуалізації VirtualBox. Так само скачати дистрибутив Ubuntu Server 20.04 LTS 64 bit (Рис.3.1), з офіційного ресурсу проекту Ubuntu - http://ubuntu.com. При використанні інших дистрибутивів Linux працездатність не гарантовано.

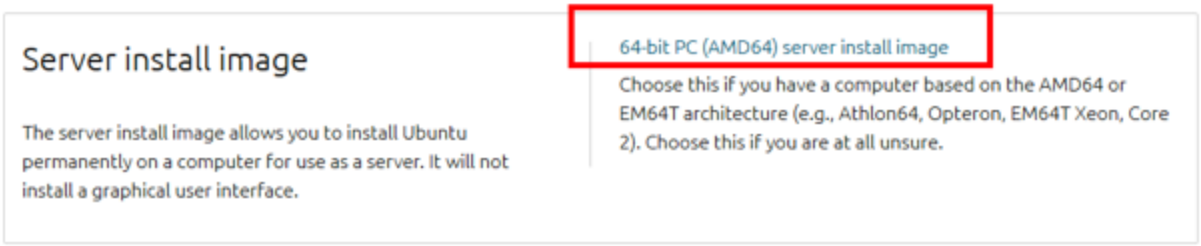

Рис. 3.1. Вибір дистрибутива ОС Linux Ubuntu

На офіційному сайті проекту, zabbix.com, в керівництві по установці системи наведені чіткі відповідності версій Zabbix дистрибутивам ОС сімейства Linux, які можна використовувати. Для створення даного проекту була обрана зв'язка ОС Ubuntu Server 20.04 + Zabbix 5.0, оскільки дистрибутив Ubuntu Server зарекомендував себе як надійна серверна платформа з широким базовим функціоналом, а Zabbix 5.0 - стабільний пакет актуальної на сьогоднішній день версії. При використанні невідповідних версій ОС і пакета - працездатність системи не гарантується, але найімовірніше, що пакет не вийде навіть встановити на невідповідний дистрибутив.

Приступимо до створення віртуальної машини. Запустив VirtualBox, і створив нову віртуальну машину (Рис. 3.2).

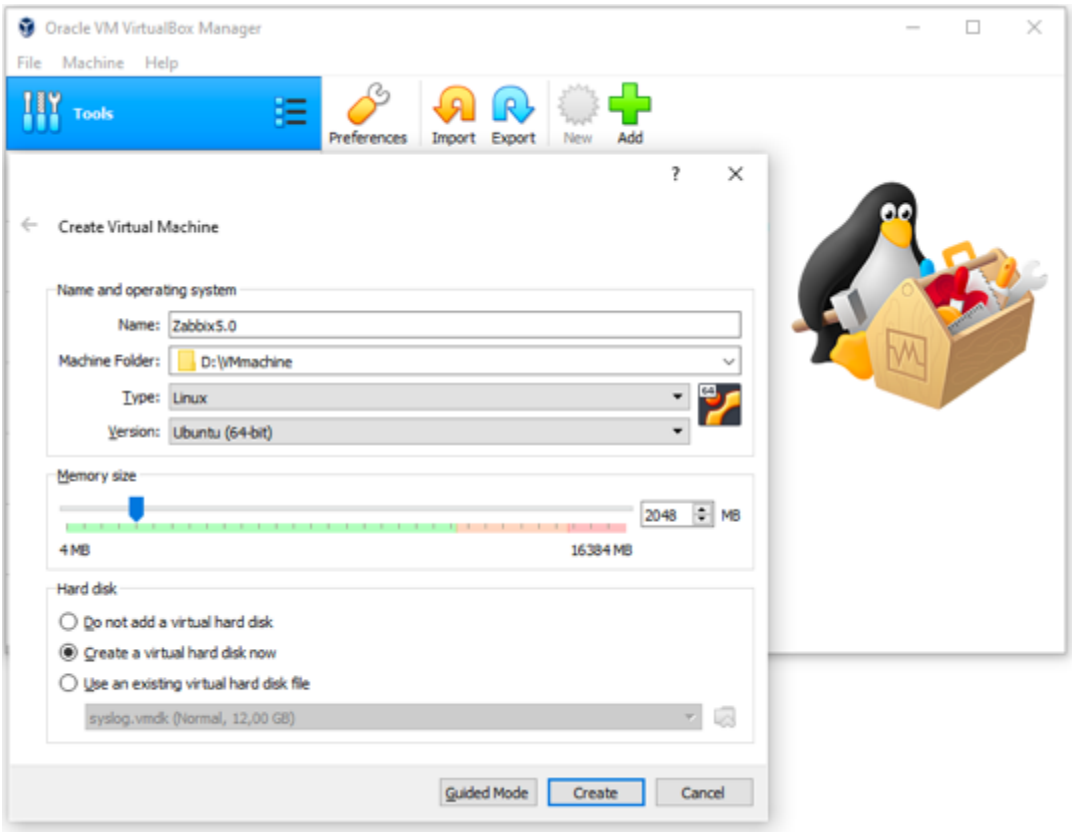

Рис.3.2. Створення віртуальної машини

Необхідно задати об'єм оперативної пам'яті, яку буде використовувати віртуальна машина. Основними споживачами системних ресурсів виступають база даних, яка обробляє всі запити і впорядковує одержувані системою моніторингу відомості, і сам Zabbix, які генерує запити до віддалених об'єктів і передає зібрані дані в базу даних.

Важливо! Необхідно надати віртуальній машині достатній обсяг оперативної пам'яті, інакше система при високих навантаженнях буде непрацездатна!

Згідно з офіційною рекомендацією Zabbix, для спостереження за мережевими вузлами в кількості 500 одиниць, досить 2 ГБ оперативної пам'яті.

Далі створив новий віртуальний жорсткий диск, виберав тип жорсткого диска VMDK, з фіксованим розміром, обсягом 10 ГБ (Рис. 3.3).

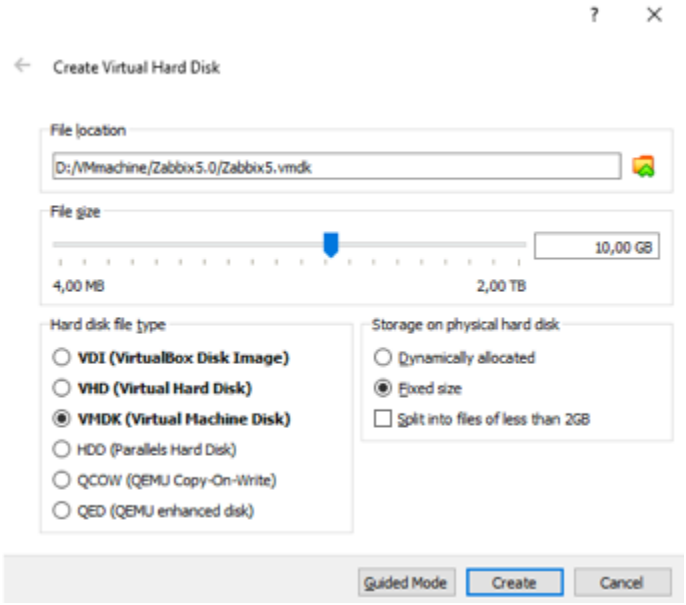

Рис.3.3. Створення жорсткого диска віртуальної машини

Такий обсяг обраний для того, щоб його вистачило для вирішення всіх завдань, які будуть розглядатися в нашому курсі. Завершіть створення віртуальної машини. У Таблиці 3.1 наведені відмінності основних типів жорстких дисків, пропонованих VirtualBox:

Табл 3.1

Відмінності основних типів жорстких дисків

Тип	Пояснение
VDI (VirtualBox Disk Image)	Собственный формат виртуальных дисков VirtualBox
VHD (Virtual Hard Disk)	Формат, применяемый в среде виртуализации Microsoft, а именно - HyperV
VMDK (Virtual Machine Disk)	Открытый формат, применяемый в среде виртуализации VmWare

У Zabbix існує методика розрахунку необхідного розміру бази даних, і її необхідно врахувати при створенні віртуальної машини: розмір бази даних

Zabbix безпосередньо залежить від кількості оброблюваних запитів в секунду і настройки очищення історії в базі даних.

Кількість оброблюваних запитів в секунду - середня кількість нових значень, які Zabbix-сервер отримує кожну секунду. Наприклад: Якщо є 1000 елементів даних з інтервалом перевірки 100 секунд, то кількість оброблюваних запитів за секунду розраховується 1000/100 = 10.

Отже, кожну секунду база даних Zabbix поповнюється десятьма новими записами.

Zabbix зберігає отримані значення певний період часу, кілька тижнів або місяців, в залежності від налаштувань. Кожне нове значення вимагає певний обсяг дискового простору для даних і індексів. Наприклад, якщо потрібно збереження 90 днів історії і кожну секунду в базу даних додається 10 нових записів, загальна кількість значень дорівнюватиме приблизно:

$$\text{Общее количество} = \left(90_{\text{дней}} \cdot 24_{\text{часа}} \cdot 3600_{\text{сек}}\right) \cdot 10 = 77760000_{\text{значений}}$$

Залежно від типу бази даних, типу отриманих значень (з плаваючою точкою, цілочисельний, рядки, файли журналів і т.д.) може знадобитися від 40 до кількох сотень байт дискового простору, для зберігання одного значення. Зазвичай одне значення займає, в середньому, 90 байт по числовим елементів даних. У нашому випадку це означає, що для зберігання всіх значень потрібно:

$$77760000_{\text{значений}} \cdot 90_{\text{байт}} = 6998400000_{\text{байт}} = \frac{6998400000_{\text{байт}}}{1024} =$$
$$= 6834375_{\text{МБ}} = \frac{6834375_{\text{МБ}}}{1024} \approx 6.7\ \text{ГБ}$$

дискового простору.

Важливо! Розмір бази даних Zabbix спочатку буде невеликим, але стане поступово збільшуватися, і зупиниться після досягнення певного моменту, залежного від налаштувань очищення бази даних (терміну зберігання значень, інструмент HouseKeeper). При впровадженні системи моніторингу Zabbix в діючій мережі - обов'язково потрібно провести розрахунок необхідної

вільного простору, виходячи із завдань моніторингу і кількості опитуваних пристроїв.

Повернемося до віртуальної машини - перед запуском залишилося вказати образ Ubuntu для завантаження, і тип мережевого підключення. Необхідно зайти в налаштування віртуальної машини, і в розділі «Носії», додати в контролер IDE, скачаний раніше ISO-образ Ubuntu Server 20.04 (Рис.3.4):

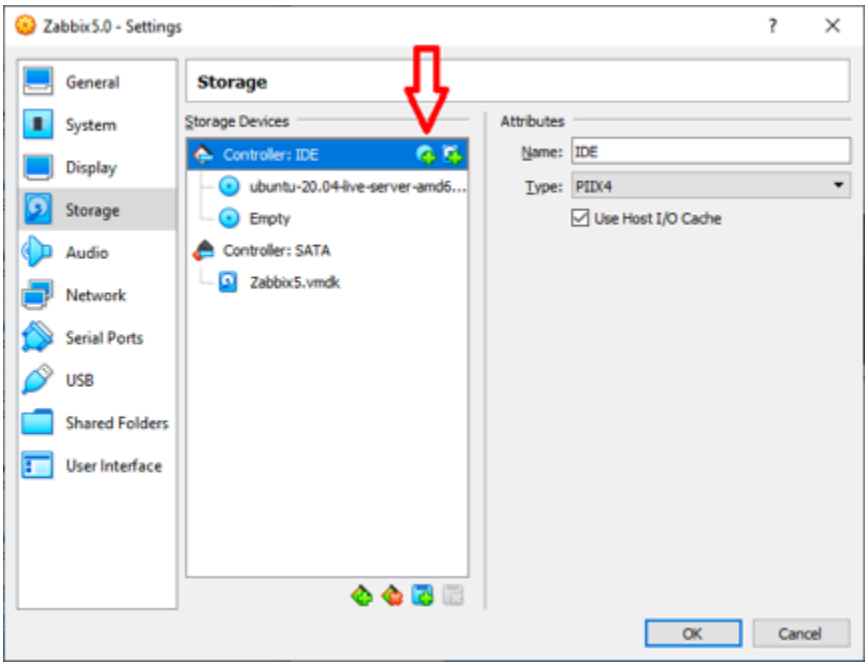

Рис.3. 4. Додавання образу Ubuntu для установки

Останній крок перед початком установки Ubuntu на віртуальну машину - організація доступу до мережі Інтернет (він обов'язково знадобиться в подальшому). Вибрав тип підключення «Мережевий міст», або Bridge, як показано на Рис. 3.5. Це означає, що віртуальний мережевий адаптер віртуальної машини буде отримувати IP-адресу з однієї підмережі з Вашим ПК, і на веб-інтерфейс Zabbix буде легко потрапити, вказуючи в браузері ПК IP-адреса віртуального хоста. Так само не буде необхідності додавати правила в віртуальний NAT віртуальної машини, для роботи з різними мережевими інструментами.

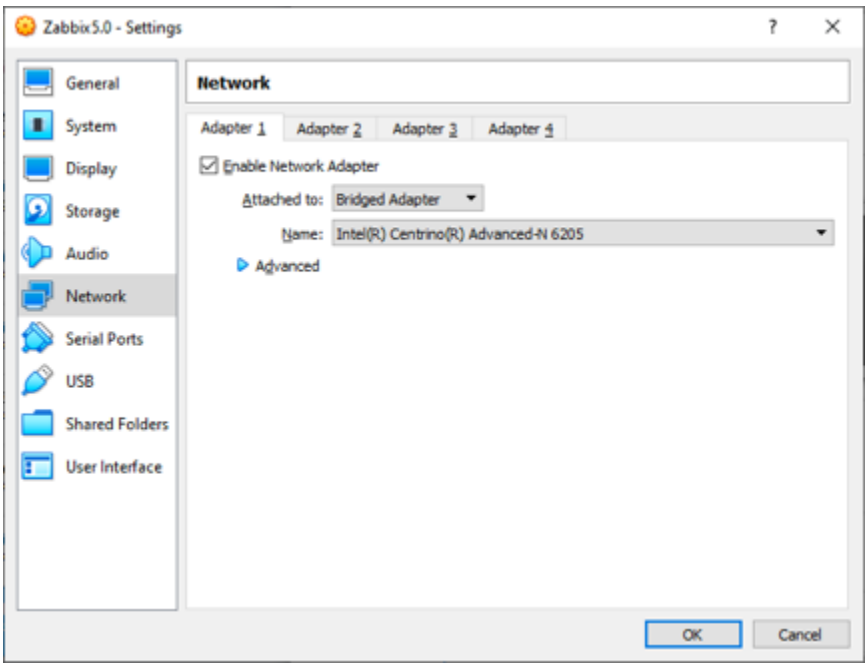

Рис. 3.5. тип мережевого підключення віртуальної машини

До речі, для спрощення подальшого виконання проекту, я використав підключення до мережі Інтернет через домашній роутер. Підключення ПК до мережі Інтернет безпосередньо не рекомендується, оскільки для віртуальної машини доведеться використовувати тип підключення NAT і надалі налаштовувати «Проброс портів».

Запустивши віртуальну машину і приступив до встановлення Ubuntu Server 20.04. Не буду детально зупинятися на процесі установки - виберіть мову установки English, розкладку клавіатури English, дія Install Ubuntu, мережевий інтерфейс, проксі (потрібно залишити поле порожнім), мережеву адресу дзеркала дистрибутивів (залиште за умовчанням), настройку використання диска Use an Entire Disk (і потім сам фізичний диск, а після – підтвердження налаштування диска). Далі програма-встановщик запропонує створити настройки профілю - ім'я машини і користувача, а так же пароль. Дані для проектного профілю наведені в Таблиці 3.2, і на Рис.3.6:

Табл. 3.2

Дані для проектного профілю

Параметр	Значение
Ваше имя	zabbix
Имя сервера	zabbix_machine
Имя пользователя	zabbix
Пароль (и подтверждение)	zabbix

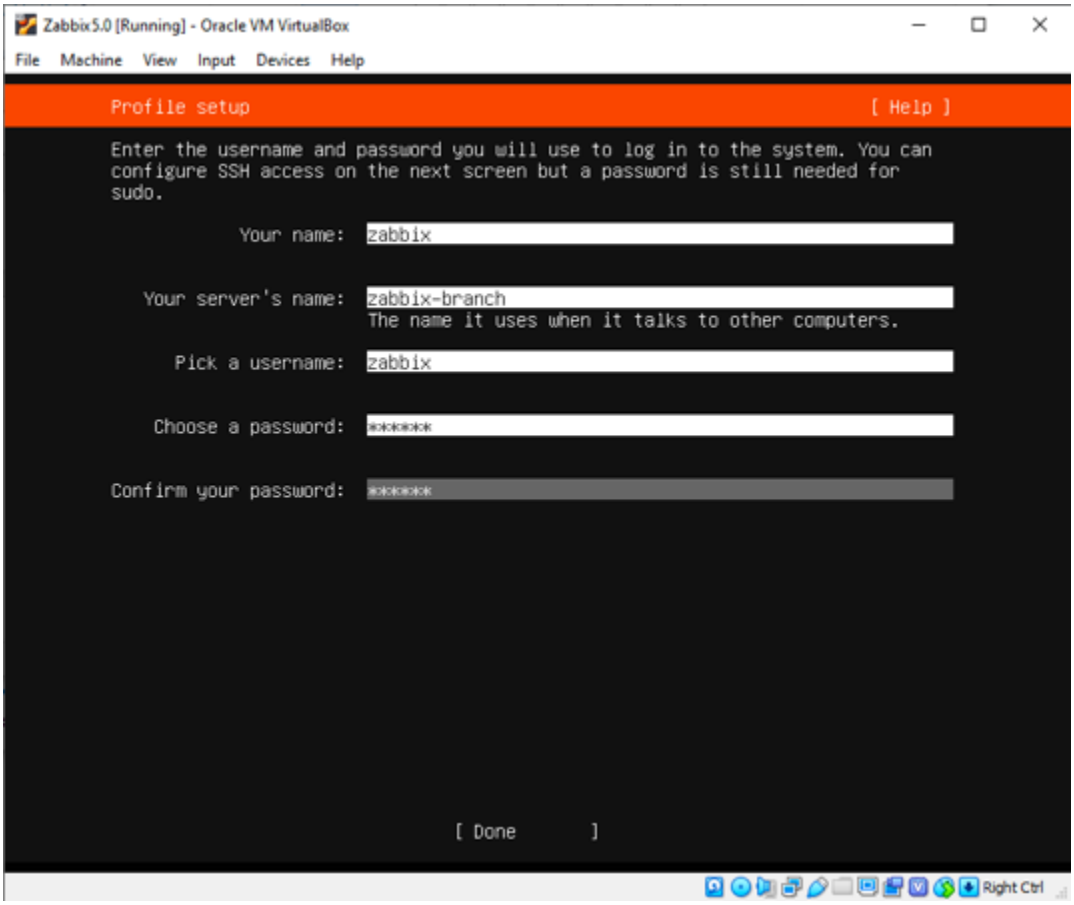

Рис. 3.6. Створення профілю

Далі установник запропонує включити в процес установки SSH-сервер - обов'язково потрібно погодитись. Далі піде вибір популярних інструментів – я пропустив цей крок. Далі піде процес установки Ubuntu Server. Після цього встановлення можно подивитися лог або перезавантажити віртуальну машину. Вибрав перезавантаження (в процесі підготовки до перезавантаження система попросить натиснути Enter).

Після перезавантаження - віртуальна машина готова до роботи. Необхідно розібратися з мережевим підключенням. В ОС Ubuntu 20.04 застосований

новий підхід до конфігурації мережевого інтерфейсу (за допомогою утиліти netplan, вже з ОС Ubuntu Server 18.04 - спостерігається відмова від утиліти net-tools). Тепер файл конфігурації мережевої карти знаходиться по шляху / etc / netplan /, і має розширення .yaml (в різних інсталяціях він буде називатися по-різному). Проаналізуйте вміст цього каталогу (Рис.. 3.7):

ls -la / etc / netplan

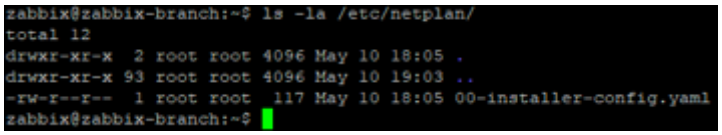

Рис.3.7. вміст каталогу /etc/netplan

Якщо у домашній / навчальної локальної мережі присутній DHCP-сервер, то віртуальна машина отримає IP-адресу автоматично. Перевірити наявність IP-адреси можна командою ifconfig, а доступ в мережу, наприклад, командою ping 8.8.8.8.

Але мені потрібна IP-адреса вручну, відредагував файл з розширенням .yaml - в моєму випадку, відповідно до Рис. 3.7, це /etc/netplan/50-cloud-init.yaml.

Важливо! Переглянути вміст каталогу можна звичної командою dir (або «dir -a», так консоль виведе всі приховані файли і папки), а можна скористатися командою «ls -la», і файли будуть виведені у вигляді списку з описом прав, розміру, і дати створення.

Для отримання прав на внесення змін з-під облікового запису, використовуйте sudo, або увійдіть в режим root, за допомогою команди sudo su:

sudo nano /etc/netplan/50-cloud-init.yaml

Налаштування DHCP виглядає так:

network:

ethernets:

enp0s3:

addresses: []

dhcp4: true

version: 2

Налаштування статичної IP-адреси виглядає інакше:

network:

ethernets:

enp0s3:

dhcp4: no

dhcp6: no

addresses: [192.168.1.1/24,]

gateway4: 192.168.1.254

nameservers:

addresses: [8.8.8.8, 8.8.4.4]

version: 2

Тепер необхідно розібратися з інтерфейсами. Перший інтерфейс, для роботи з сервером Ubuntu - інтерфейс командного рядка, або Command Line Interface (CLI). Саме в ньому буде налаштовуватися пакети, працювати зі скриптами, і взаємодіяти з сервісами системи в цілому. Другий інтерфейс, графічний інтерфейс користувача, або Graphical user interface (GUI) - інтерфейс системи моніторингу Zabbix, саме в ньому буду освоювати матеріали курсу.

Різниця між CLI і GUI, в загальному випадку, в тому, що вони призначені для вирішення різних завдань. CLI гнучкий при роботі з різними програмними засобами, майже не вимагає апаратних ресурсів, дозволяє оперувати командами майже як текстовий редактор, і добре підходить для обслуговування сервісів, настройки конфігурацій, роботи з масивами даних. GUI служить для візуалізації інтерфейсу, і дозволяє вирішувати власні завдання по роботі з різними прикладними програмами. В ОС Windows яскравий приклад відмінностей між GUI і CLI - звичний нам віконний інтерфейс і командний рядок CMD.

Для роботи з CLI можна користуватися терміналом середовища віртуалізації (в нашому випадку, це VirtualBox), але це незручно, оскільки немає можливості копіювати команди в термінал, і потрібні деякі графічні ресурси для підтримки оболонки терміналу. Крім того, якщо доведеться працювати одночасно з різними пристроями (наприклад, декількома серверами і активним мережевим обладнанням), користуватися терміналами буде просто важко. Набагато зручніше при адмініструванні серверів користуватися програмними клієнтами для віддаленого доступу.

Найпростіший, поширений і доступний клієнт віддаленого доступу - PuTTY. Його я і рекомендую використовувати. Зрозуміло, є безліч інших програмних клієнтів, більш складних, або відносяться до платних програмним продуктам. У них є ряд корисних функцій, наприклад, зберігання паролів, зручний інтерфейс для збереження сесій до серверів, багатовіконний режим роботи з декількома пристроями, додаткові надбудови для шифрованого передачі файлів, і багато іншого. Наприклад, це програмні продукти SecureCRT або Xshell.

Скачайте PuTTY і віддалено підключіться до віртуальної машини з його допомогою (Рис. 3.8):

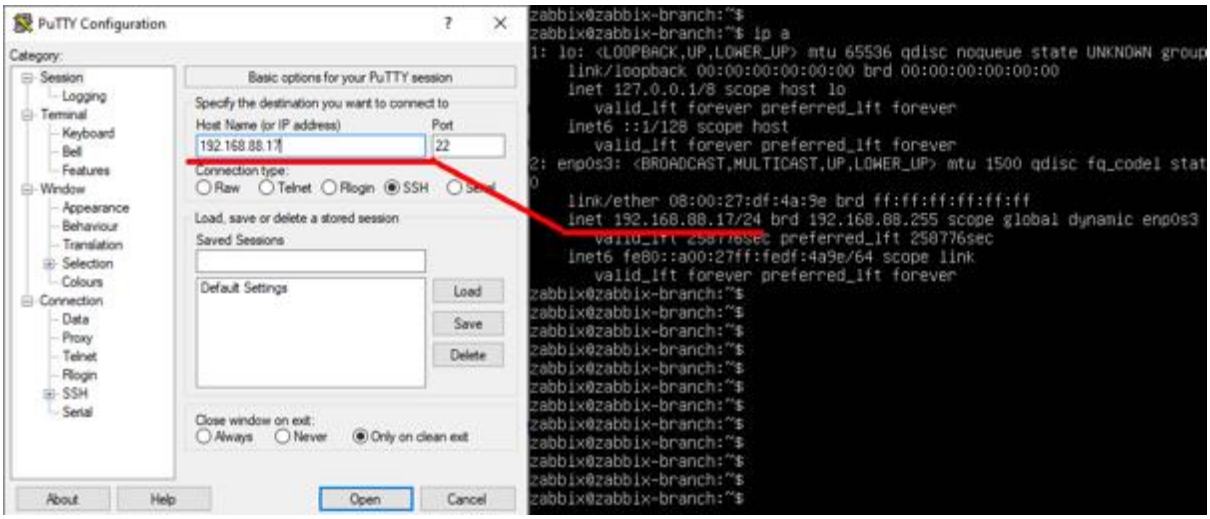

Рис. 3.8. Віддалене підключення до сервера

За допомогою команди «ip a» в терміналі сервера виводите IP-адресу мережевого інтерфейсу, і по протоколу віддаленого управління SSH, підключаєтеся до сервера клієнтом віддаленого доступу PuTTY.

В результаті, отримав запрошення від віртуального сервера, на введення логіна і пароля.

Просунуті віддалені клієнти використовуються для організації логування на стороні адміністратора (коли все, що відбувається в консолі записується в файл і зберігається на комп'ютері, з якого здійснюється віддалений доступ), і впорядкування сесій до пристроїв (в великих корпоративних мережах кількість мережевих пристроїв може досягати десятків тисяч, і кожен раз створювати сесію і налаштовувати параметри підключення незручно).

SSH є основним протоколом віддаленого управління на сьогоднішній день, в ОС Ubuntu він включений і преднастроен за замовчуванням. Його головна відмінність від попередника, першого протоколу віддаленого управління, Telnet - в шифруванні всього трафіку між клієнтом і сервером. Telnet передає всі дані - логіни і паролі, що вводяться команди і висновок на них, у відкритому вигляді (так званий plain text), і на сьогоднішній день його використання в принципі не рекомендується.

Далі необхідно оновити всі пакети (програми), встановлені на ОС. Пакети для ОС сімейства Linux зберігаються в репозиторіях (репозиторії - сховища актуальних версій пакетів для певної ОС).

Список основних репозиторіїв, з яких ОС Ubuntu викачує різні пакети, знаходиться в файлі /etc/apt/sources.list. Подивитися список можна за допомогою команди cat /etc/apt/sources.list.

Спочатку виконаємо команду, оновлюючу інформацію про пакети, що містяться в репозиторіях, а потім команду, обновляющую пакети

sudo apt-get update

sudo apt-get upgrade

Після закінчення оновлення - система готова до встановлення Zabbix.

3.1.2 Встановлення Zabbix

Приступаючи до встановлення Zabbix, потрібно пам’ятати, що для конкретних збірок Zabbix підходять строго певні ОС сімейства Linux. На офіційному сайті проекту, zabbix.com, наведені чіткі інструкції, з якого сховища встановлювати пакет для певної ОС.

Якщо використати в якості ОС дистрибутив, відмінний від використовуваного в рекомендаціях- обов'язково потрібно отримати інформацію і правильний репозиторій, і підходящі для нього версії Zabbix.

Заручившись інформацією з zabbix.com, почну установку. Спочатку зазначу репозиторій, з якого потрібно встановлювати пакет. Для проекту обрана актуальна версія Zabbix 5.0 (Рис. 3.9-3.10).

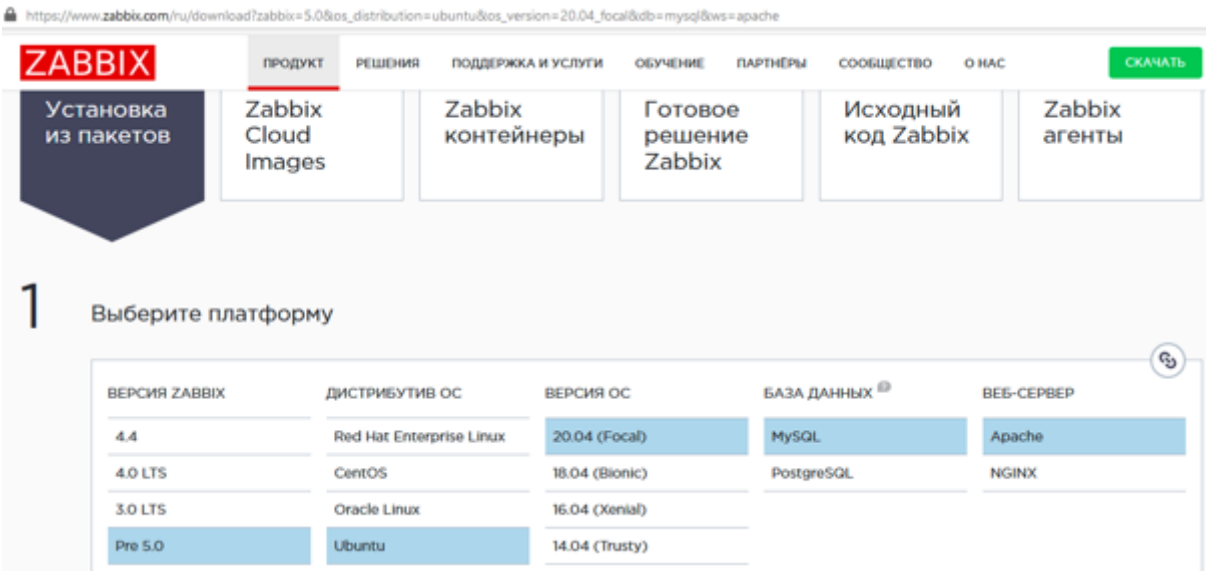

Рис.3.9. Вибір правильного дистрибутива

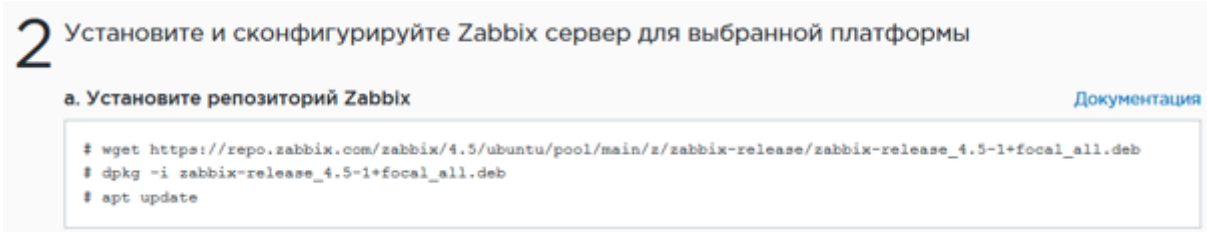

Рис.3.10. Отримання сховища

Команда wget додасть в файл /etc/apt/sources.list, що містить список репозиторіїв за замовчуванням, новий шлях, який вказує на пакет zabbix, відповідний до моєї системи ОС Ubuntu (до речі, в якості БД буде використана MySQL, через свою простоту) .

wget https://repo.zabbix.com/zabbix/4.5/ubuntu/pool/main/z/zabbix-release/zabbix-release_4.5-1+focal_all.deb

Команда dpkg -i витягує інформацію про складові релізу Zabbix версії 4.2.

dpkg -i zabbix-release_4.5-1 + focal_all.deb

Знову оновив інформації про репозиторіях.

sudo apt-get update

Список додаткових репозиторіїв, які додаються користувачем і, наприклад, є повноцінними сховищами великих продуктів (в даному випадку Zabbix), зберігається в файлі /etc/apt/sources.list.d/zabbix.list.

Команда dpkg -i актуалізувала відомості про доступні для установки модулів пакета zabbix. Після введення в консолі команди «sudo apt-get install zabbix-» та натиснувши клавішу TAB - система запропонує можливі варіанти подальшого введення. Вводячи перші літери частини команди і використовуючи натискання TAB - Ви заощадите масу часу (Рис. 3.11).

Рис.3.11. інструмент швидкого введення команд

Після всього цього поновив інформацію про репозиторії і встановив необхідні пакети для Zabbix - «back-end», веб-інтерфейс, і агент (знадобиться пізніше):

sudo apt-get install zabbix-server-mysql zabbix-frontend-php zabbix-apache-conf zabbix-agent

Погодився встановити програмне забезпечення і дочекався її закінчення. Наступний етап установки - створити базу даних для Zabbix в сервісі баз даних MySQL: Потрібно виконати скрипт (готовий набір команд) для створення бази даних Zabbix.

Для різних баз даних Zabbix пропонує готові скрипти зі створення баз даних. Zabbix підтримує такі бази даних, ка MySQL і PostgreSQL.

Для проекту досить простий БД MySQL. При впровадженні Zabbix для моніторингу, скажімо, від 500 об'єктів (наприклад, велике підприємство), буде потрібно БД PostgreSQL.

Спочатку встановимо mysql-server, потім здійснимо вхід в БД MySQL

sudo apt-get install mysql-server

sudo mysql -uroot -proot

Далі потрапляємо в командний рядок БД. Створимо базу даних zabbix_db, з кодуванням utf-8:

mysql> create database zabbix_db character set utf8 collate utf8_bin;

При правильному введенні команди система підтвердить введення повідомленням OK. У разі помилки - виведе ERROR. Далі створив користувача zabbix_us, і призначив йому права на доступ до бази (створений користувач саме для доступу до бази даних zabbix_db, це не користувач ОС).

mysql> create user zabbix_us @ localhost identified by 'zabbix_pw';

mysql> grant all privileges on zabbix_db. * to zabbix_us @ localhost;

Дана команда створює користувача і пароль для доступу до конкретної бази.

mysql> quit;

Щоб уникнути плутанини, склав таблицю 3.3 з параметрами доступу до бази даних.

Імпортуємо початкові дані. Перейдемо в каталог файлів БД zabbix і розпакуємо схему в БД zabbix_db (цей крок потрібен для створення структури таблиць в базі даних):

cd / usr / share / doc / zabbix-server-mysql

sudo zcat create.sql.gz | mysql -uzabbix_us -pzabbix_pw zabbix_db

Ця операція може зайняти кілька хвилин, дочекаймося її закінчення.

При виконанні цього пункту потрібно дотримуватися послідовності дій і орієнтуйтеся на дані Таблиці 3.1.2.1, для розуміння скоєних дій.

Наступний етап установки - зміна конфігураційного файлу zabbix_server.conf. У ньому необхідно коректно вказати назву бази даних для Zabbix, а так же логін і пароль для доступу до неї.

Відкриваю файл zabbix_server.conf:

sudo nano /etc/zabbix/zabbix_server.conf

І вношу в нього такі виправлення (таблиця 3.3):

Табл. 3.3

Параметри доступу до бази даних

Параметр базы данных	zabbix_server.conf до редактирования	zabbix_server.conf после редактирования
	#DBHost=localhost	DBHost=localhost
Название базы zabbix_db	#DBName=zabbix	DBName=zabbix_db
Пользователь zabbix_us	#DBUser=zabbix	DBUser=zabbix_us
Пароль к базе zabbix_db	#DBPassword=zabbix	DBPassword=zabbix_pw

Розпочну процес Zabbix сервера:

sudo service zabbix-server start

Важливо зробити ще один крок - додати нові процеси (Zabbix-агент і Zabbix-сервер) в автозавантаження, щоб вони запускалися автоматично при запуску системи:

sudo update-rc.d zabbix-server enable

sudo service zabbix-agent start

sudo update-rc.d zabbix-agent enable

Майже закінчив. Залишилося небагато - налаштувати конфігурацію PHP для веб-інтерфейсу Zabbix. Файл конфігурації Apache для веб-інтерфейсу Zabbix розміщується в /etc/apache2/conf.d/zabbix або /etc/apache2/conf-enabled/zabbix.conf. Майже всі параметри конфігурації PHP вже задані.

Відреагую файл zabbix.conf для веб-сервера:

sudo nano /etc/apache2/conf-enabled/zabbix.conf

В розділі:

<IfModule mod_php5.c>

Розкоментував рядок:

php_value date.timezone Europe / Moscow

В розділі:

<IfModule mod_php7.c>

Розкоментував рядок:

php_value date.timezone Europe / Moscow

Після зміни файлу конфігурації перезапустив веб-сервер apache:

sudo service apache2 restart

Отже, установка закінчена!

3.2. Перший запуск веб інтерфейсу

Здійснимо перший вхід в систему моніторингу. Нагадаю, я встанровив всю систему таким чином, що на моєму ПК (який має конкретну IP-адреса, в моєму випадку 192.168.88.17), створена віртуальна машина з віртуальним мережевим інтерфейсом, який отримав по DHCP інший IP-адресу, відмінну від IP-адреси ПК . Отримавши IP-адреса віртуальної машини за допомогою команди ifconfig. У моєму випадку IP-адреса інтерфейсу eth0 - 192.168.88.17.

Тепер скориставшись будь-яким інтернет-браузером на ПК - ввів в адресному рядку http: // IP-адрес_віртуальной_машіни / zabbix.

У моєму випадку це http://192.168.88.17/zabbix - результат на Рис. 3.12:

Рис. 3.12. перший запуск системи Zabbix

Сервіс запущено, але йому потрібно провести самотестування, перевірити працездатність всіх модулів і коректність налаштувань. Натиснувши кнопку «Next step» - Я потрапив в меню перевірки налаштувань (Рис. 3.13).

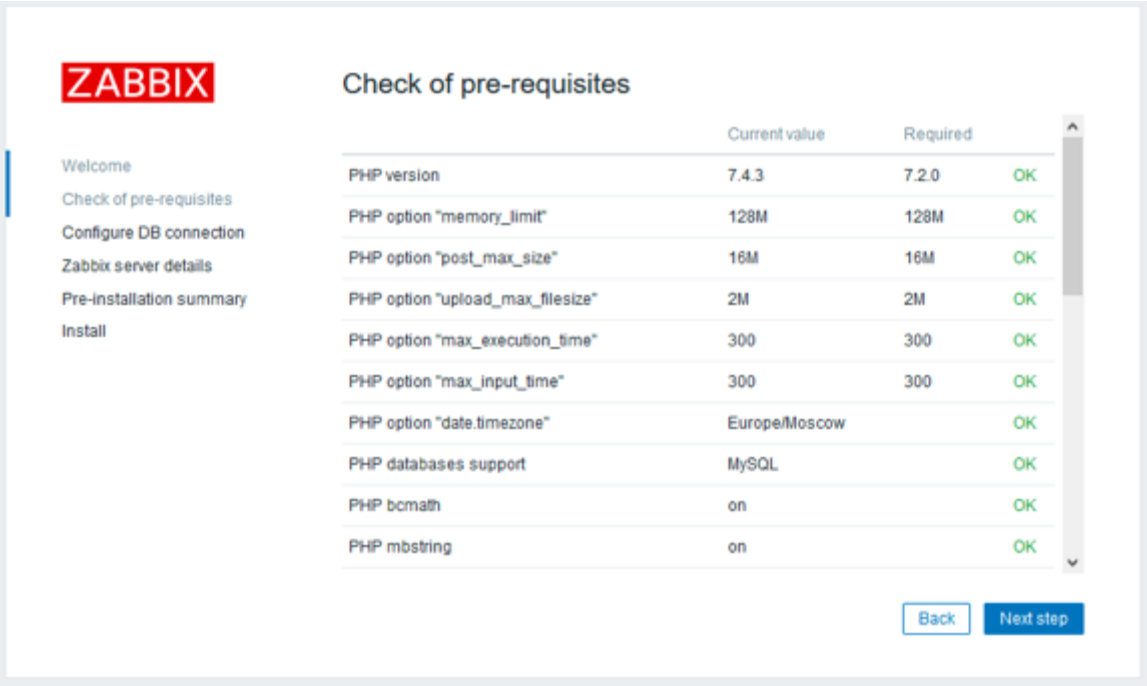

Рис.3.13. Перевірка системи

Якщо хоча б один пункт буде налаштований некоректно - Zabbix не зможе приступити до наступного кроку. Натиснувши кнопку «Next step» - Я потрапив в меню налаштувань доступу до БД (Рис..3.14), за допомогою таблиці 3.3 привів все в відповідність.

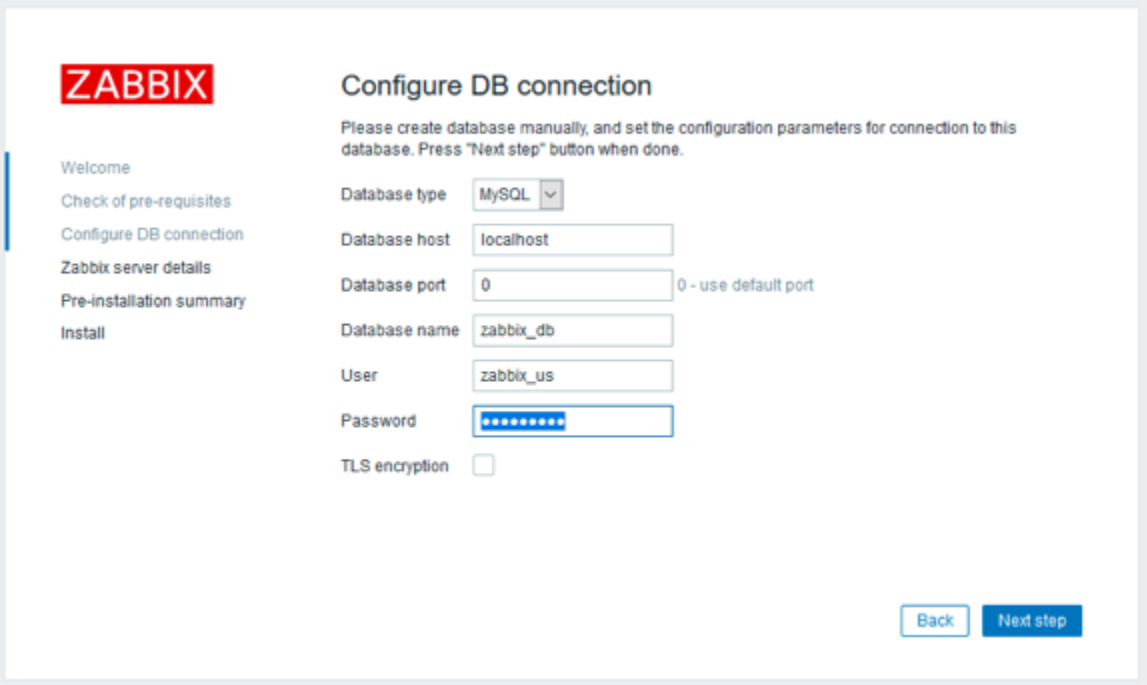

Рис.3.14. Налаштування БД

Натисніть кнопку «Next step» - Я потрапив в меню налаштувань імені (Рис. 3.15). Дайте ім'я свого облікового запису.

Рис.3.15. Налаштування імені сервера

Натиснувши кнопку «Next step» - Я потрапив в перевірку налаштувань. Ще раз натиснувши кнопку «Next step» - Я потрапив до фінального меню (Рис. 3.16).

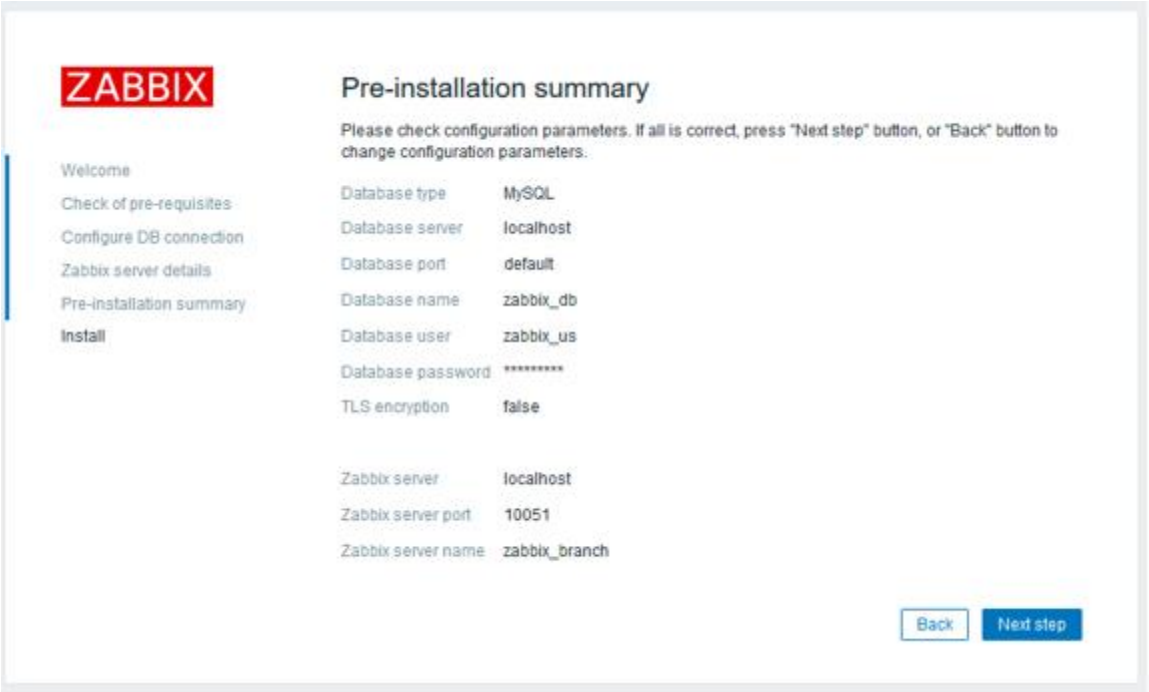

Рис.3.16. Завершення установки

Натиснувши кнопку «Next step» - я перейшов в фінальне вікно запуску системи (Рис. 3.17):

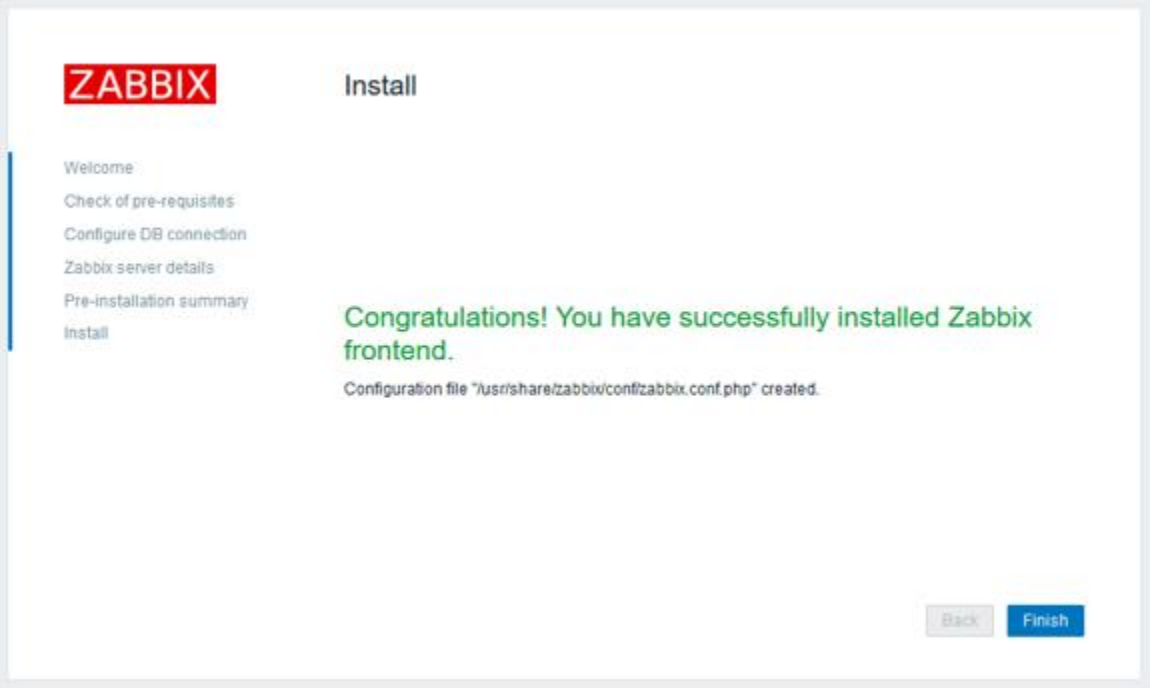

Рис.3.17. Запуск системи

І нарешті у мене в руках кращий інструмент моніторингу мережевих пристроїв серед безкоштовних вільно розповсюджуваних продуктів!

Отже, після натискання кнопки «Finish» потрапив в меню введення логіна і пароля (Рис.3.18).

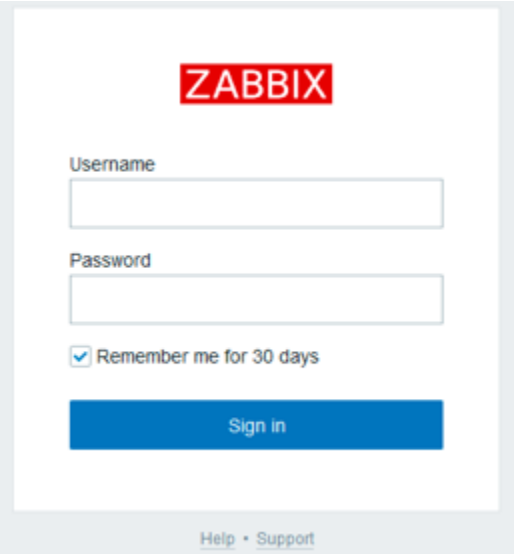

Рис.3.18. запрошення для входу в систему

Це перше меню налаштованої і готової до роботи системи моніторингу Zabbix. Логін і пароль за замовчуванням - Admin / zabbix. При першому вході Можна побачите наступне вікно - основна панель Zabbix (Рис. 3.19):

Рис.3.19. Основна панель Zabbix

Ось і успішне розгортанням системи моніторингу! Взагалі, на офіційному сайті проекту можна скачати готові інсталяції, образи, докери - але я вважаю, що необхідно усвідомлювати і розуміти етапи установки і настройки елементів системи моніторингу. А також готові інсталяції тільки на CentOs, а деякі додаткові програмні забезпечення є тільки в Debian системах, такі як libphone та esim

Коли вирішимо закінчити роботу, "виключення" ОС Ubuntu (аналог завершення роботи в Windows) здійснюється командою:

sudo shutdown -H now -P

При такому відключенні буде коректно завершуватися робота всієї віртуальної машини. Щоб подивитися можливі параметри команди shutdown, запускайте її в вигляді «shutdown --help».

Результатом виконання даного розділу повинен став скріншот веб-інтерфейсу успішно запущеної системи .моніторингу, з панеллю System Information, в якій статус запуску системи має значення «Yes» (Рис. 3.20).

Якщо буде проблема з підключенням до БД, веб-інтерфейс буде працювати, але реально система моніторингу не працюватиме, і статус запуску системи матиме значення «No».

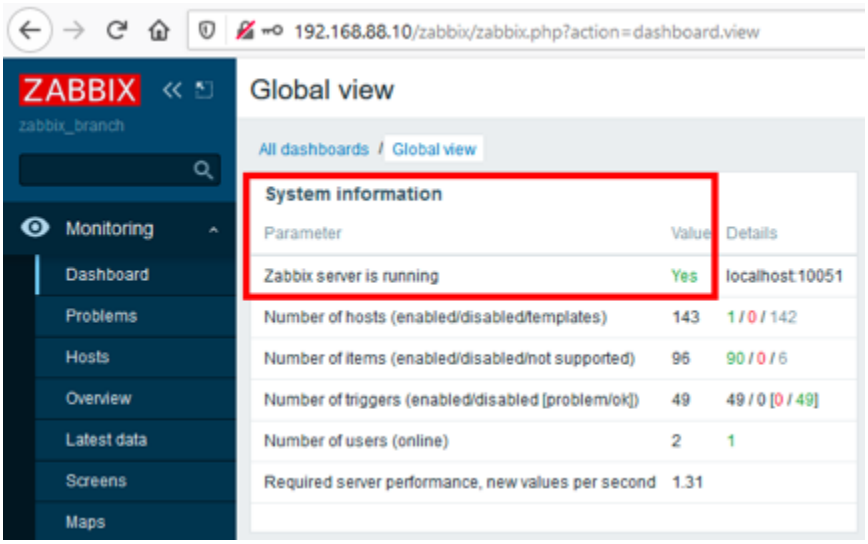

Рис.3.20. Звіт про виконану роботу

3.3 Налаштування Zabbix та знайомство с системою

Отже, в минулому підрозділі розгорнули готову для роботи систему моніторингу. Здійснимо перший вхід як користувач. Запустивши середу віртуалізації Oracle VM VirtualBox, запустив віртуальну машину і дочекався завантаження ОС. Після завантаження ОС упевніться, що з віртуальної машини доступна мережа Інтернет (при введенні команди ifconfig - інтерфейс eth0 має IP-адресу, ping 8.8.8.8 показує доступність сервера 8.8.8.8).

У браузері ПК ввів http: // IP-адрес_віртуальной_машіни / zabbix, логін і пароль за замовчуванням - Admin / zabbix. Перевірте, що веб-інтерфейс доступний (Рис. 3.21):

Рис. 3.21. Запрошення для входу в систему

Виконаю невеликий тюнинг доступу до веб-інтерфейсу. За IP-адресою віртуальної машини доступна сторінка-заглушка веб-сервера Apache2 (Рис.3.22):

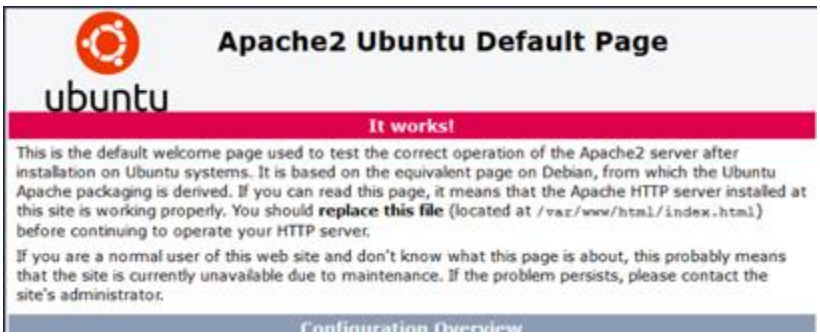

Рис. 3.22. Сторінка-заглушка веб-сервера Apache2

Змінив її, щоб при зверненні на IP-адресу віртуальної машини, потрапляв відразу в веб-інтерфейс Zabbix. Відредагував конфігураційний файл 000-default.conf, що відповідає за роботу з віртуальними хостами:

sudo nano /etc/apache2/sites-available/000-default.conf

змінив рядок

DocumentRoot / var / www / html

Привівши його до виду

DocumentRoot / usr / share / zabbix

Зберіг зміни, і перезапустив сервіс apache2

sudo service apache2 restart

Після цього веб-інтерфейс Zabbix став доступний безпосередньо по IP-адресою.

Увійшовши в систему - бачимо інтерфейс Zabbix англійською мовою. Необхідно провести русифікацію інтерфейсу (української мови немає в Zabbix), але проблема полягає в тому, що ні в збірці Zabbix 5.0, ні в ОС Ubuntu Server 20.04, російську мову не встановлено. Зробимо це.

Перейшовши в консоль віртуальної машини, і переглянувши список доступних мов командою (Рис. 3.23):

sudo locale -a

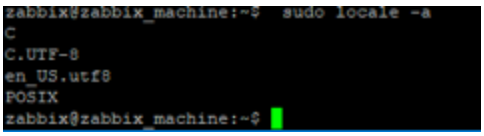

Рис. 3.23. перегляд «локалей»

Очевидно, російська мова не встановлено. Щоб подивитися, які мови (так звані «локалі» - мова і кодування. Можливо встановити на дану ОС, виконайте команду:

cat / usr / share / i18n / SUPPORTED | grep ru_

Команда cat (Рис. 3.24) служить для простого висновки вмісту текстового файлу в консоль. Приставка «| grep »вказує вивести тільки ті рядки, в яких є збіг із зазначеними після« | grep »символами (вибірка в даному випадку відбувається за випадковим збігом з« ru_ »).

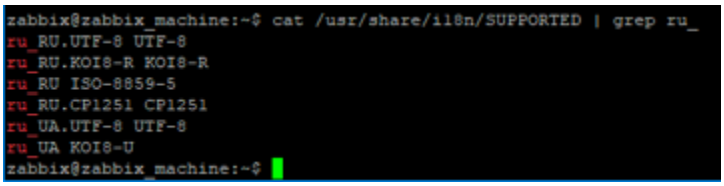

Рис. 3.24. перегляд «локалей»

Для встановлення російської мови, потрібно ввести наступні команди:

sudo locale-gen ru_RU

sudo locale-gen ru_RU.UTF-8

sudo dpkg-reconfigure locales

Після введення команди на переконфігуруванні локалей, буде запропоновано вибрати з великого списку, які локалі переконфигурировать - можна вибрати все (що займе багато часу), або тільки en і ru для UTF-8. Просто двічі виберіть Ok в з'являться меню - і будуть сконфігуровані локалі російської мови (вони вже будуть обрані за замовчуванням).

Потрібно звернути увагу, і при створенні БД для Zabbix, і при налаштуванні мови в веб-інтерфейсі, потрібно вибрати кодування utf-8 - більшість сучасних

веб-платформ за замовчуванням працюють саме на ній, в її наборі близько 100000 символів, вона використовується повсюдно. На сьогоднішній день кодування UTF-8 стала загальноприйнятою, витіснивши менш вдалі рішення.

Локаль встановлена і ініціалізовано, тепер необхідно перезавантажити веб-сервер Apache, щоб мова змінилась і став доступний в Zabbix (веб-інтерфейс Zabbix працює на базі веб-сервера Apache):

sudo service apache2 restart

Залишилося зовсім небагато - включити в настройках Zabbix російську мову (Рис. 3.25):

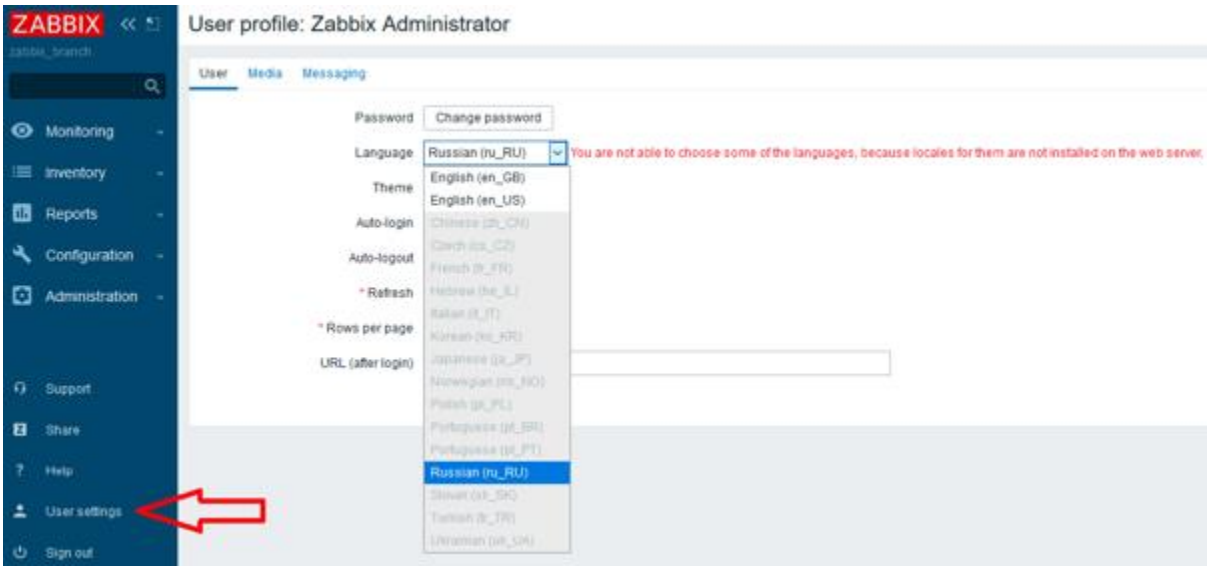

Рис.3.25. установка російської мови

Після русифікації виконаю ще кілька налаштувань системи:

- встановлю час зберігання значень в БД Zabbix (щоб уникнути її заповнення непотрібними даними і зменшення вільного місця на жорсткому диску);

- часовий пояс (щоб в системі відображався зручним для мене час);

- E-mail повідомлення про події (поки тільки підключення до поштового сервера для відправки листів).

При виборі часу зберігання зібраних даних в БД Zabbix керуюсь двома речами:

- технічними вимогами до системи моніторингу, заснованими на чітко сформульованих потребах (за який період можуть знадобитися зібрані дані при аналізі подій, що відбулися);

- здоровим глуздом (не потрібно зберігати оперативні дані за останні 3-5 років, ніхто і ніколи не буде їх аналізувати) - в залежності від поставлених завдань це може бути від 30 до 180 днів.

Щоб визначити проміжок часу зберігання зібраних даних в БД, потрібно увійти в меню Адміністрування - Загальні - Очищення історії (Рис. 3.26). Я вибрав період зберігання даних - 30 днів.

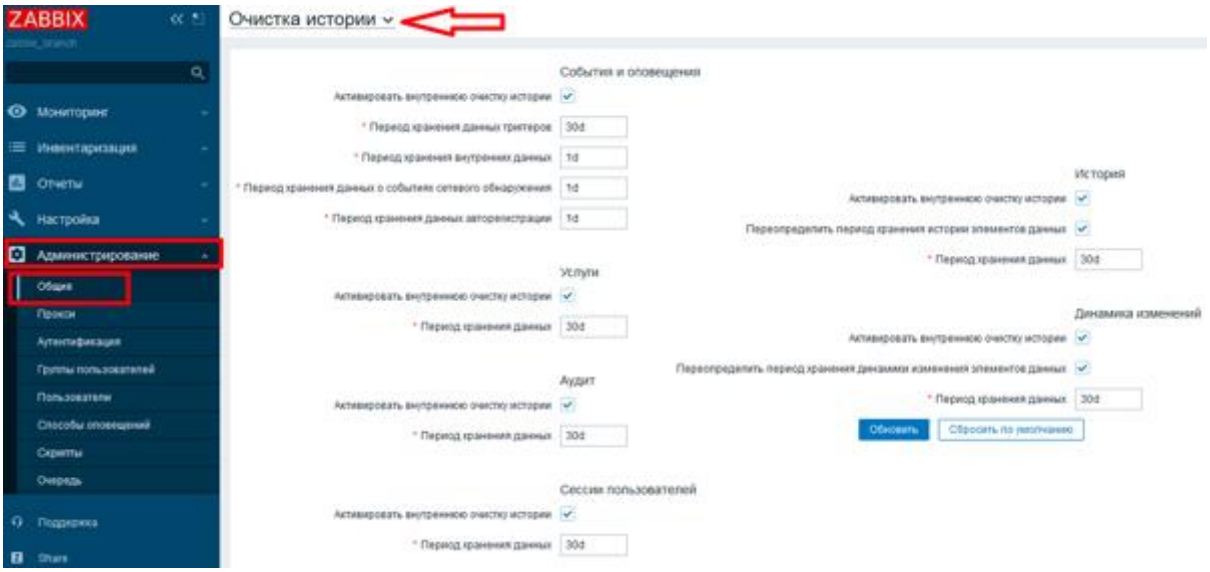

Рис. 3.26. настройка часу зберігання даних в БД

Наступний крок - встановити в Zabbix правильний час. Розберемося, звідки веб-інтерфейс Zabbix бере час. На ОС Ubuntu є системний час, в нашому випадку ОС знаходиться на віртуальній машині, по команді date показує час, встановлене в Windows (можна встановити службу ntp - «sudo apt-get install ntp», і ОС почне отримувати час зі спеціальних серверів точного часу , часовий пояс або визначається автоматично, або встановлюється вручну так само в пакеті ntp). Змінити часовий пояс (в прикладі нижче - на Київський) можна командою:

sudo timedatectl set-timezone Europe / Kyiv

Іноді виникає помилка відображення часового поясу, якщо ОС бере час з батьківської системи, на якій розгорнута віртуальна машина, хоча в CLI Ubuntu Server відображається київський час.

А ось Zabbix показує час, таке ж, як і веб-сервер Apache, який, в свою чергу, запитує у ОС Ubuntu час «за Гринвічем», і виставляє поправку на часовий пояс, зазначену в настройках файлу / etc / apache2 / conf- enabled / zabbix.conf. Щоб змінити час, що відображається в Zabbix, відредагуйте цей файл:

sudo nano /etc/apache2/conf-enabled/zabbix.conf

У ньому знайдіть параметр «php_value date.timezone», і вкажіть потрібний Вам часовий пояс, синтаксис деяких часових поясів нижче:

Europe / Kyiv
Europe / Moscow
Europe / Riga
Europe / Samara
Asia / Yekaterinburg
Asia / Novosibirsk
Asia / Krasnoyarsk
Asia / Irkutsk
Asia / Kamchatka
Asia / Magadan
Asia / Sakhalin

Я відредагував файл таким чином, щоб Zabbix відображав час р Новосибірська - «php_value date.timezone Europe / Kyiv» .Після цього потрібно перезапустити сервіс Apache, для застосування налаштувань:

sudo service apache2 restart

Фіксувати дані потрібно в тому часовому поясі, який стане точкою відліку для кореляції подій.

Якщо мова йде про моніторинг обладнання компанії, що знаходиться, наприклад, в одному місті, зручніше використовувати місцевий часовий пояс. Якщо ж об'єкти моніторингу знаходяться по всьому світу, краще використовувати 0 за грінвичем час, або час, прийнятиій в компанії. Це допоможе уникнути різночитань при аналізі подій, встановлюючи для всіх єдину точку відліку фіксації і відображення цих подій.

Єдина точка відліку часу повинна бути встановлена і в системі моніторингу, та на яку обслуговує мережевому і серверному обладнанні (необхідна кореляція зафіксованих подій в системі моніторингу та, наприклад, в логах syslog-сервера).

Але можна піти ще далі. Наприклад, необхідно вивести на екран час з кілька часових поясів. Це може бути необхідно великої диспетчерської або логістичної службі, у якій зони відповідальності розташовуються в декількох регіонах. Необхідно створити універсальний скрипт з зовнішнім аргументом, який буде трансформувати час за Гринвічем під час в заданому часовому поясі.

Меню *Моніторинг* містить всі можливі види відображення даних, що збираються. Будь-яка інформація, яку агрегує Zabbix (опитуючи різні об'єкти), відображається в загальній статистиці, а події що відбуваються поділяються за рівнями важливості, візуалізація зібраних даних передбачена у вигляді графіків, карт мереж, і всіляких звітів. Всі ці дані відображаються в різних розділах Моніторингу.

Меню *Моніторинг* - *Панель* відображає коротке зведення всієї інформації, що збирається. Тут гнучко налаштовуються панелі відображення зібраних даних, є можливість розміщувати прямі посилання на обрані графіки, комплексні екрани і карти мереж (обрані - найважливіші для адміністратора системи, по суті, швидкі посилання на важливі об'єкти).

Можна сконструювати свою першу панель моніторингу (Меню Моніторинг - Панель, кнопка «Змінити панель»). Вивчивши інструменти зміни панелі, і залишивши на ній 4 віджета (Рис. 3.27) - Інформація про систему, Актуальні

проблеми за важливістю, і дві панелі годин, для місцевого часу (локального), і за Грінвичем (на сервері).

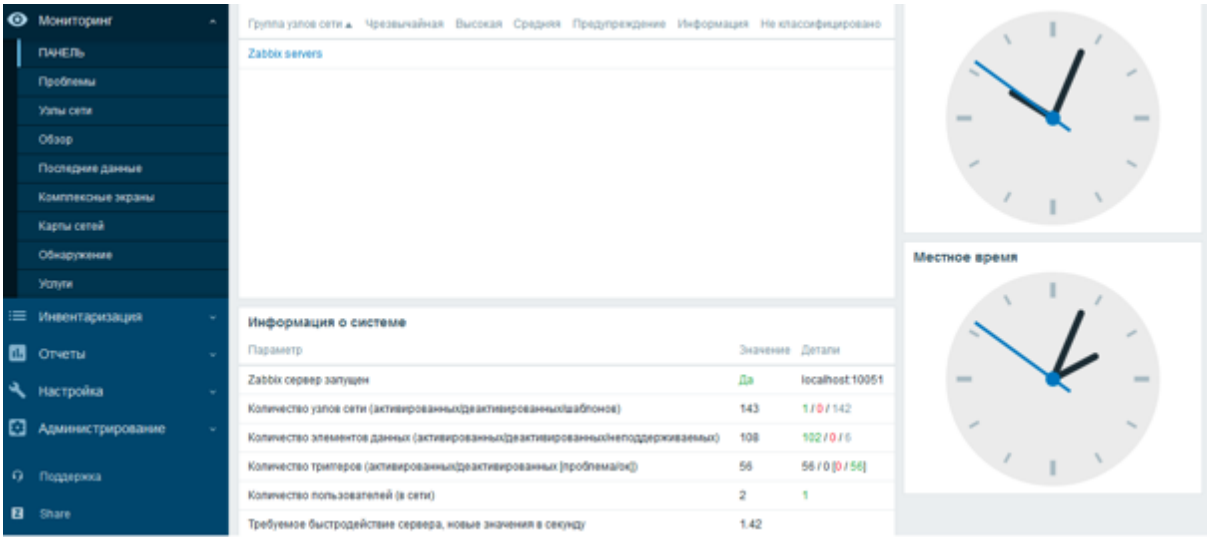

Рис. 3.27. Створення панелі моніторингу

Досвід моніторингу великої мережевої інфраструктури за допомогою Zabbix показує, що для відображення основних відомостей і проблеми в мережі досить віджета «Проблеми за важливістю», або, у всякому разі, їм необхідно навчитися користуватися. Графіки і карти мережі потрібні для аналізу і візуалізації. Але при правильній вибудуваної концепції важливість тригерів, віджет «Проблеми за важливістю» - самий інформативний інструмент в рамках Zabbix.

Тепер, для наочності освоєння системи, організую моніторинг першого об'єкта, самого Zabbix-сервера, за допомогою Zabbix-агента, який встановлю як пакет.

Відредагував конфігураційний файл агента:

sudo nano /etc/zabbix/zabbix_agentd.conf

Надав демону права root для збору даних:

AllowRoot = 1

Перезапустив Zabbix-agent:

sudo /etc/init.d/zabbix-agent restart

У Zabbix встановлено вузол мережі Zabbix server, призначений для моніторингу власних параметрів. Після того, як дав агенту права на моніторинг системи, де він встановлений, він почне збирати дані системі Zabbix і ОС Ubuntu Server.

Вибравши *Проблеми* і *Огляд* надають лістинг зареєстрованих подій (спрацювали тригерів). Вкладка *Вузли* мережі надає детальний огляд станів вузлів мережі (об'єктів моніторингу), звідси здійснюється доступ до корисних інструментів візуалізації даних. Найголовніше - звідси здійснюється доступ до графіків (Рис. 3.28):

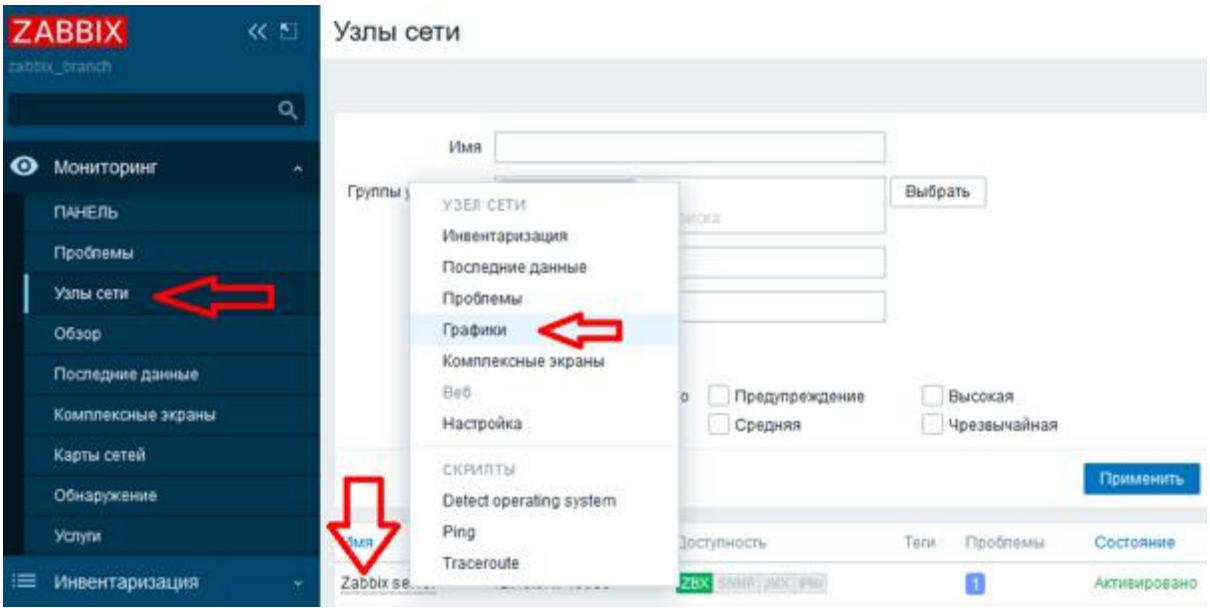

Рис. 3.28. Вкладка Моніторинг - Вузли мережі

Доступ до графіків в попередніх версіях здійснювався через окремий пункт меню Моніторингу. Зараз він організований через вкладку Вузли мережі. На мою думку, це зроблено для того, щоб підкреслити, що графіки не є єдиним інструментом візуалізації і аналізу даних. Графіки будуються на основі зібраних даних (Рис. 3.29). Кожен графік необхідно «спроектувати» на етапі створення шаблону опитування для об'єкта моніторингу. Це меню дуже популярно, оскільки саме тут можна візуалізувати події, що відбулися деякий час назад і зафіксовані системою моніторингу в числових

значеннях. Масштабованість графіків і можливість виділення певного часового відрізку дозволяють швидко отримати необхідну інформацію.

Рис. 3.29. Інструмент моніторингу Графіки

Меню *Моніторинг - Останні дані* відображає безпосередньо зібрані значення метрик, або елементів даних. Інструмент корисний для перевірки працездатності метрик як таких, і, в разі необхідності, їх налагодження, в значенні troubleshooting. Так само, тут можна за допомогою фільтрів візуалізувати отримувані значення від одного або декількох вузлів (Рис. 3.30):

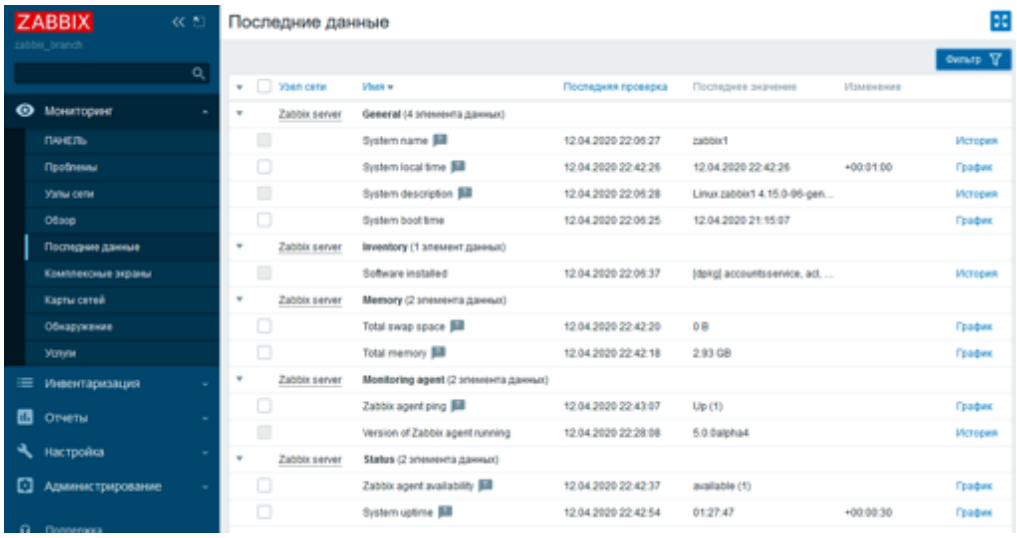

Рис. 3.30. Вкладка Моніторинг - Останні дані

Меню *Моніторинг* - *Комплексні екрани* призначене для розміщення відразу декількох об'єктів на одній веб-сторінці, це можуть бути карти, графіки, годинник (чомусь це нагадує основну панель). Комплексні екрани - інструмент візуалізації необхідних параметрів для спостережуваного процесу або об'єкта, зібраних в одному місці (наприклад, при здійсненні моніторингу сервера на відповідному комплексному екрані відображаються графіки завантаження процесора, температури, зайнятої оперативної пам'яті, і вільного місця на жорстких дисках). Однак, в Zabbix є серйозності відмінності щодо доступу та використання комплексних екранів, створюваних вручну користувачем, і автоматично при роботі шаблону опитування. Для Рис. 3.31 узятий готовий комплексний екран, встановлений в системі, і доповнений вручну.

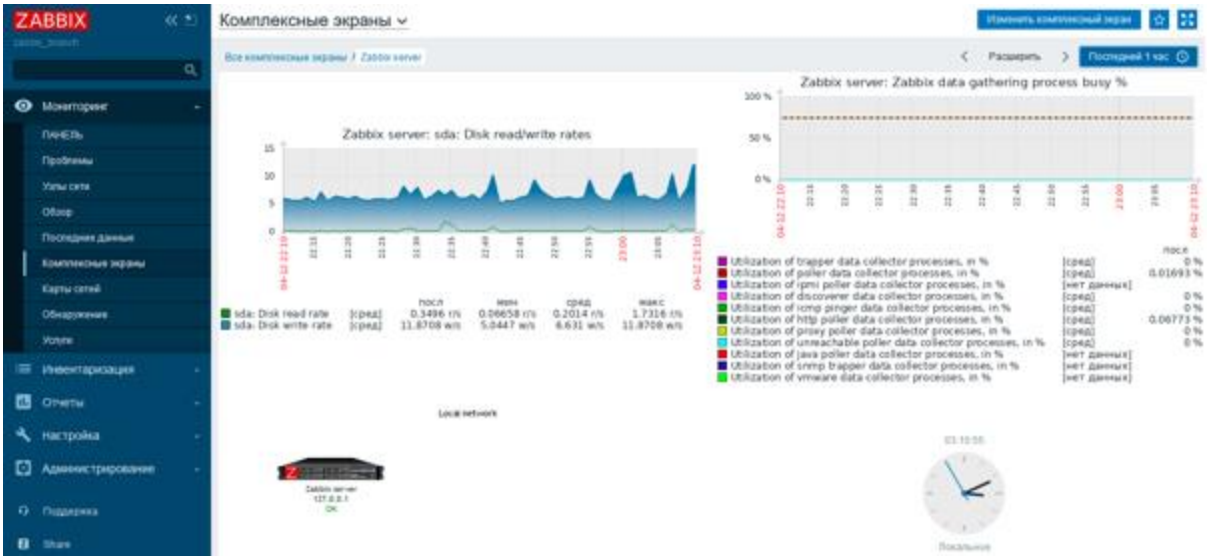

Рис. 3.31. Приклад комплексного екрану

Меню *Моніторинг* - *Карти мережі* відображає карти мережі для перегляду і редагування. Інструмент *Карти мережі* в якості основного засобу моніторингу часто використовує черговий персонал на підприємствах, де здійснюється цілодобовий моніторинг будь-яких об'єктів або процесів. Хоча, при добре продуманій системі збору різних даних з спостерігається мережі, найоб'єктивнішим інструментом моніторингу є, як не дивно, основна Панель (*Моніторинг* - *Панель*). На Картах мережі події відображаються неповноцінно,

але карти зручні для візуалізацій топологій мережевого обладнання, датчиків розумних будинків, або рознесення обладнання по місцевості, з підкладкою картинки.

Меню *Моніторинг* - *Виявлення* відображає вузли мережі, додані в Zabbix за допомогою автоматичного виявлення.

Меню *Моніторинг* - *По*слуги візуалізує переднастроєні SLA (Service Level Agreement), якщо це необхідно.

Пункт меню *Інвентаризація* містить дві вкладки - *Огляд і Вузли мережі*. Цей розділ забезпечує можливість перегляду таких деталей інвентаризації вузлів мережі, як тип обладнання, так звані PID - Product ID (моделі пристроїв), встановлене програмне забезпечення, серійні номери, час роботи. Інакше кажучи, це інструмент для візуалізації текстових даних, які можна витягти з вузлів мережі за допомогою Zabbix. У кожному розділі збору інвентарних даних буде приділятися достатньо часу, для отримання повноцінних відомостей про об'єкт моніторингу.

Меню *Звіти* є інструментом перегляду та аналізу зафіксованих системою Zabbix подій, з різними функціями вибірки, фільтрів і методів представлення даних. Крім того, в меню Звіти представлені розгорнуті звіти про дії, скоєних користувачами, і системою (наприклад, відправка повідомлень по електронній пошті).

Меню *Налаштування* - найважливіша ланка в системі Zabbix. Тут планується концепція моніторингу всього обладнання. У цьому меню зазвичай використовується тільки три основні вкладки - *Групи вузлів мережі, Шаблони, і Вузли мережі.*

Меню *Налаштування* - *Групи вузлів* мережі об'єднують в собі пристрої, в залежності від завдання, виходячи з трьох наступних принципів:

- за територіальною ознакою (всі пристрої в групі знаходяться в одній локації або офісі);

- за принципом однотипності пристроїв (серверне обладнання в одній групі, веб-сайти в інший, мережеві сервіси в третій, мережеве обладнання в четвертій, і т.п.);

- за принципом розмежування прав доступу (різним користувачам системи Zabbix призначаються різні права на перегляд або на перегляд і редагування тих чи інших груп).

При експлуатації Zabbix в великих корпоративних мережах було помічено, що пристрої краще об'єднувати в різні групи, керуючись всіма трьома принципами (за місцем розташування, за однотипності пристроїв, щодо розмежування прав доступу). Все залежить від конкретного завдання моніторингу. До речі, в головному меню *Моніторинг* - *Панель*, обладнання об'єднується в групи для перегляду, саме виходячи з приналежності об'єктів до тієї чи іншої Групі вузлів мережі.

Меню *Налаштування* - *Шаблони* містить в собі ядро системи моніторингу - шаблони опитування об'єктів. Шаблони опитування створюються під конкретні пристрої в залежності від моделі. Наприклад, якщо мова йде про мережевий пристрій передачі даних - комутатори певного виробника і серії, для нього створюється шаблон опитування по протоколу SNMP з метою фіксації навантаження трафіку на портах, завантаження процесора, і кількості помилок. Якщо ж мова йде про моніторинг веб-сайтів, створюється шаблон, який використовує встановлення tcp-сесій до веб-ресурсів, перевірки вмісту сторінок і термінів експірації доменів. Для моніторингу простоїв доступності пристроїв по протоколу ICMP - створюється відповідний шаблон, що має на увазі моніторинг будь-яких мережевих пристроїв за допомогою відправки пакетів icmp-echo.

Деякі шаблони, наприклад, для моніторингу об'єктів по ICMP - можна застосовувати до будь-яких пристроїв. Шаблони ж, використовують SNMP, створюються для кожної групи однотипних мережевих пристроїв або ж для кожної окремої моделі мережевого пристрою (якщо на підприємстві

використовується обладнання різних виробників), адже MIB-файли і OID для тих чи інших параметрів у таких пристроїв значно відрізняються.

Меню *Налаштування* - *Вузли мережі* дозволяє додавати, змінювати і видаляти об'єкти моніторингу. При створенні Вузла мережі до нього застосовується шаблон або шаблони, за якими опитуватиметься цей вузол, а так само група або групи вузлів мережі, до якої цей вузол буде належати.

Меню *Налаштування* - *Обслуговування* задає періоди обслуговування (по суті, здійснення моніторингу за розкладом) певних Вузлів мережі або Груп вузлів мережі. Даний пункт простий у використанні, але непопулярний на практиці.

Меню *Налаштування* - *Дії* дозволяє гнучко налаштовувати дії (повідомлення на електронну пошту, в месенджери, або виконання деяких скриптів). Особливість даного інструменту в тому, що дію можна створити в зв'язці з будь-якою подією, яке зафіксує Zabbix (дії можна прив'язувати до групи вузлів або одному вузлу, при фіксації одного або декількох подій - варіантів умов безліч).

Меню *Налаштування* - *Кореляція подій* дозволяє встановлювати залежності між подіями, що відбуваються. Це допомагає зменшити кількість повідомлень про події, якщо одні події залежать від інших.

Меню *Налаштування* - *Виявлення* є інструментом автоматичного додавання нових вузлів мережі.

Налаштування - *Послуги* відповідає за настройку SLA.

Меню *Адміністрування* - *Загальні* дозволяє конфігурувати параметри системи Zabbix (тему веб-інтерфейсу, додавання призначених для користувача зображень, регулярні вирази, макроси, перетворення значень та інше).

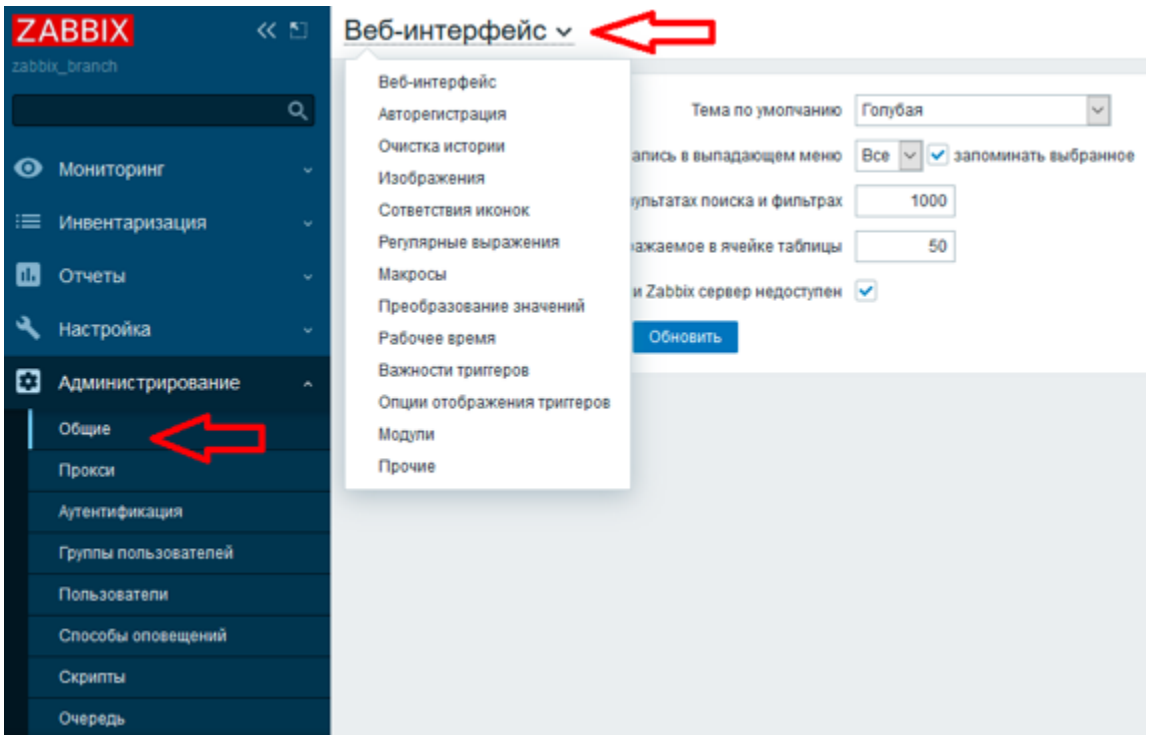

Рис. 3.32. Меню Адміністрування - Загальні

Меню *Адміністрування* - *Проксі* дозволяє організовувати розподілену інфраструктуру моніторингу Zabbix, коли кілька серверів Zabbix здійснюють моніторинг різних сегментів або процесів в мережі, а потім передають на головний сервер Zabbix. Це дозволяє знизити навантаження на основний сервер, здійснювати моніторинг «закритих» або недоступних для основного сервера сегментів мережі.

Меню *Адміністрування* - *Аутентифікація* дозволяє вибрати тип аутентифікації при вході користувача на сервер Zabbix:

- внутрішня аутентифікація має на увазі введення логіна і пароля, що зберігаються на сервері;

- LDAP-аутентифікація означає зовнішню аутентифікацію через сервіси Microsoft Active Directory або OpenLDAP;

- HTTP-аутентифікація має на увазі аутентифікацію через веб-сервер Apache (ця функція повинна бути попередньо налаштована).

Розділи меню *Адміністрування* - *Групи користувачів і учасників* дозволяють, по-перше, адмініструвати облікові записи Адміністраторів системи

моніторингу та операторів (з правами тільки на перегляд об'єктів, але не на їх зміну), а по-друге - розмежовувати їх зоною доступу до об'єктів моніторингу за групам вузлів мережі. У облікового запису Admin за замовчуванням є права на перегляд і зміна всіх груп вузлів мережі.

Меню *Адміністрування - Скрипти* оголошує команди, що виконуються на рівні ОС, наприклад, ping або traceroute.

Меню *Адміністрування - Черга* відображає кількість запитів моніторингу, з яких-небудь причин поставлених в чергу, і які очікують виконання. При коректно налагодженій системі моніторингу, надмірності ресурсів серверної платформи, відсутності незакритих сесій і дій з надмірною часом очікування - всі лічильники не повинні виходити за межі 1 хвилини.

Тепер вирішимо завдання з різними часовими поясами. Сконструюємо елементарний bash-скрипт, який буде виводити час з поправкою на часовий пояс, який буде здаватися зовнішнім аргументом. Zabbix має спеціальний каталог, в якому за замовчуванням зберігає скрипти, які використовуються для так званих зовнішніх перевірок. У файлі конфігурації Zabbix цей каталог вказано у змінній ExternalScripts:

ExternalScripts = / usr / lib / zabbix / externalscripts

При бажанні його можна змінити, я вважаю за краще працювати в ньому. Перейдемо в цей каталог і створимо наш перший скрипт (Рис.3.33):

cd / usr / lib / zabbix / externalscripts

sudo nano universal_time.sh

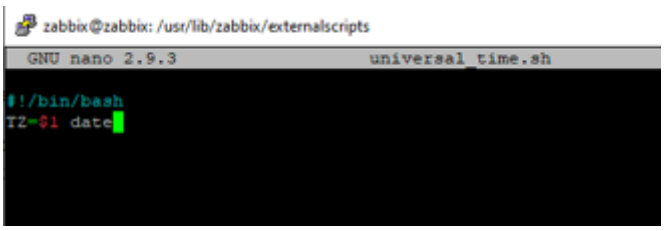

Рис. 3.33. Вміст скрипта виведення часу з поправкою на часовий пояс

Скрипт простий:

#! / Bin / bash - позначення використовуваної оболонки, bash;

TZ - Таймзона, це параметр утиліти date, що задає часовий пояс;

$ 1 - зовнішній агрумент скрипта, або змінна, яка буде змінною величиною;

date - утиліта для виведення часу в консоль;

Далі, зробимо скрипт виконуваним:

sudo chmod + x universal_time.sh

Після цього проаналізуємо його роботу (Рис. 3.34):

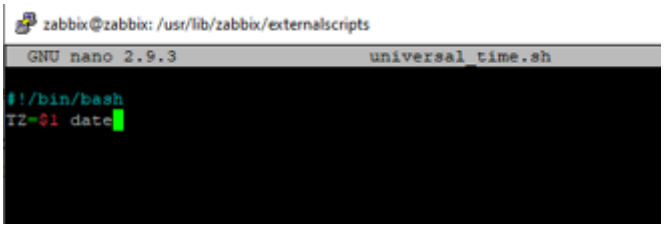

Рис. 3.34. Робота скрипта і утиліти date

Тепер створимо на існуючому вузлі мережі Zabbix_Server кілька елементів даних, для отримання значень часу в різних часових поясах. Я створюю групу елементів даних (інструмент угруповання елементів даних) з назвою «1. Time », щоб вона була вгорі після угруповання під назвою (Рис. 3.35):

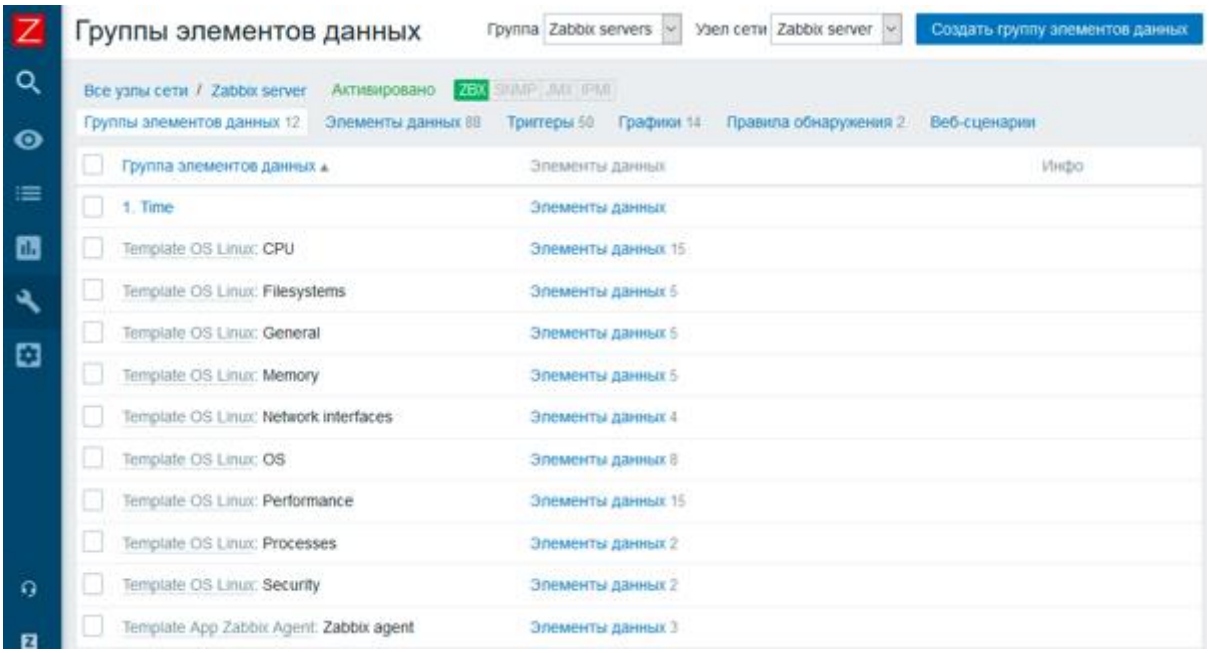

Рис. 3.35. Створення групи елементів даних

Після цього перейду до створення самих елементів даних, які і будуть запитувати у нашого сервера час з різними часовими поясами (Рис. 3.36):

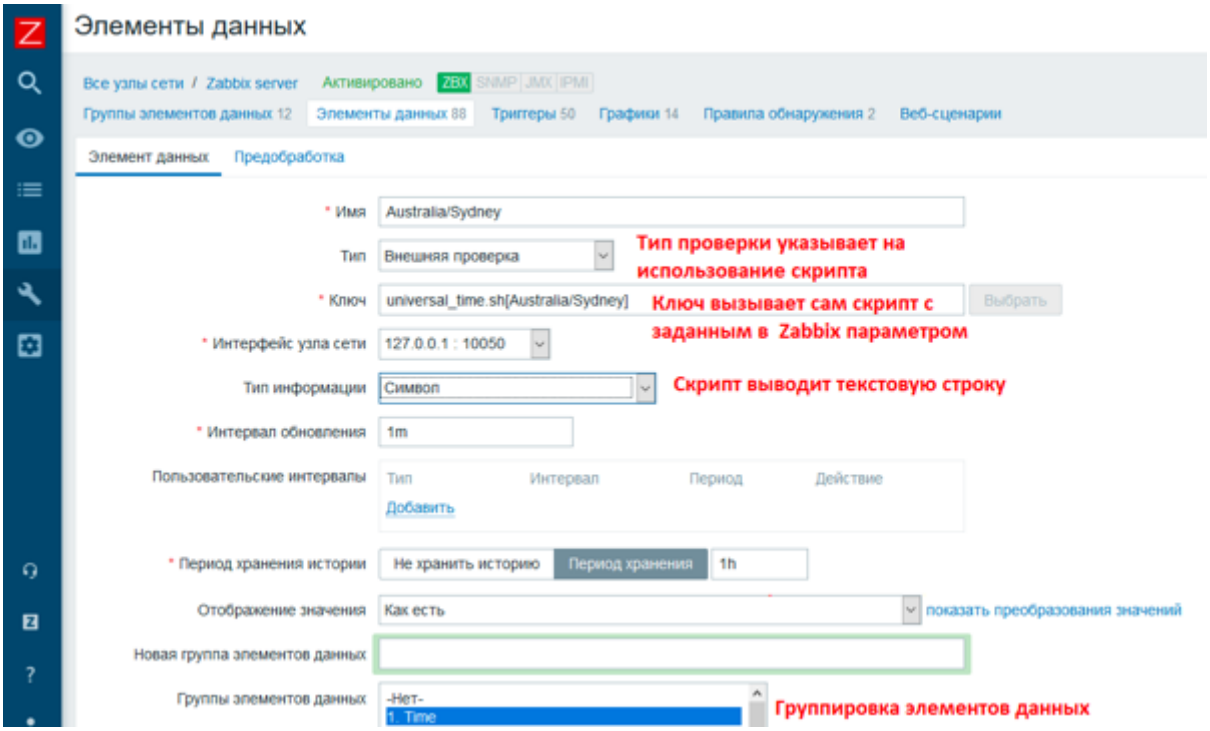

Рис. 3.36. Створення елемента даних

Для наглядності створив інші елементи даних.

Вибрав всі створені елементи даних, і вручну примусово опитав кнопкою «Виконати зараз». Після цього перевірив результат опитування в останніх даних (Рис. 3.37):

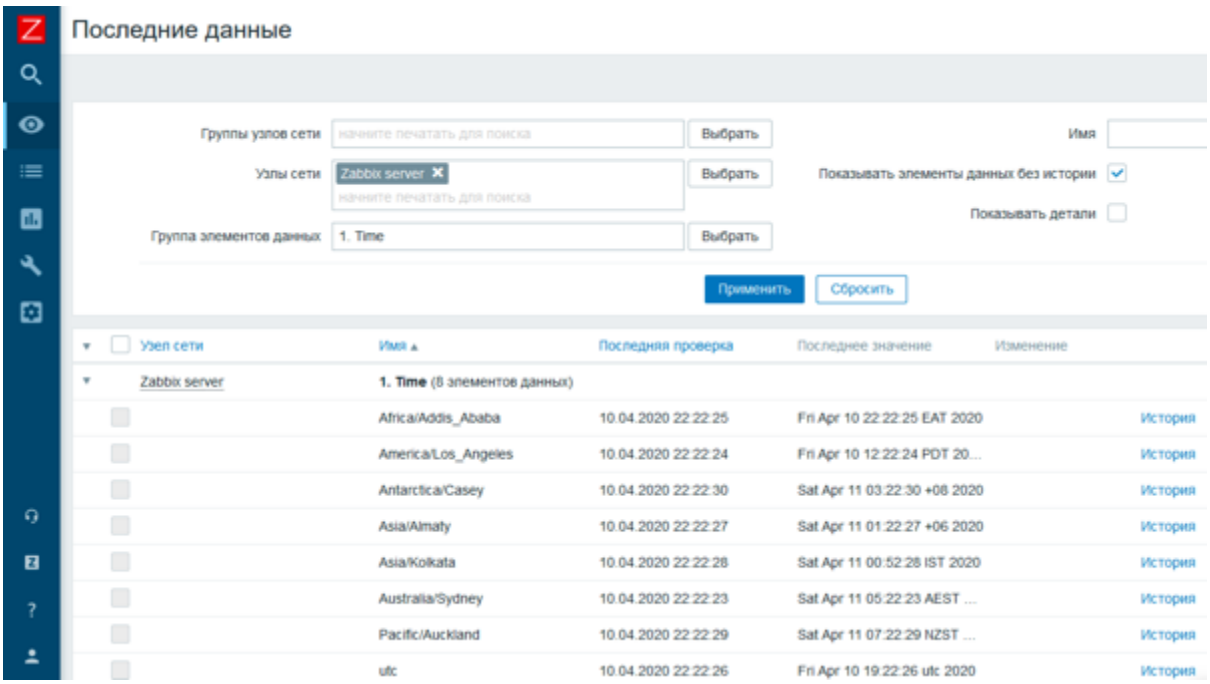

Рис. 3.37. Результат опитування в останніх даних

Тепер доповню основну панель моніторингу віджетами Простий текст і Годинник. Результат нижче (Рис. 3.38):

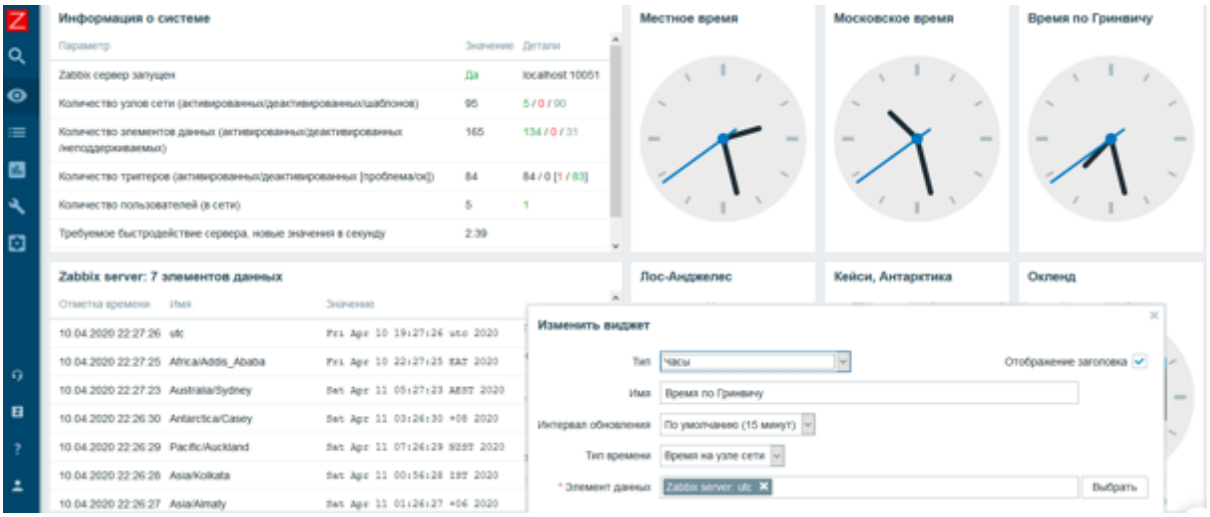

Рис.3.38. Панель моніторингу

Отже, на цьому огляд системи закінчений. У наступному розділі приступимо безпосередньо до створення віртуальної машини.

3.4 Вигрузка віртуальної машини системи

Дія потрібні для вигрузки диску віртуальної машини доступні через стандартний провідник Windows. Якщо невідомо де знаходиться папка з віртуальної машиную, це можна побачити з Virtualbox (Рис.3.39)

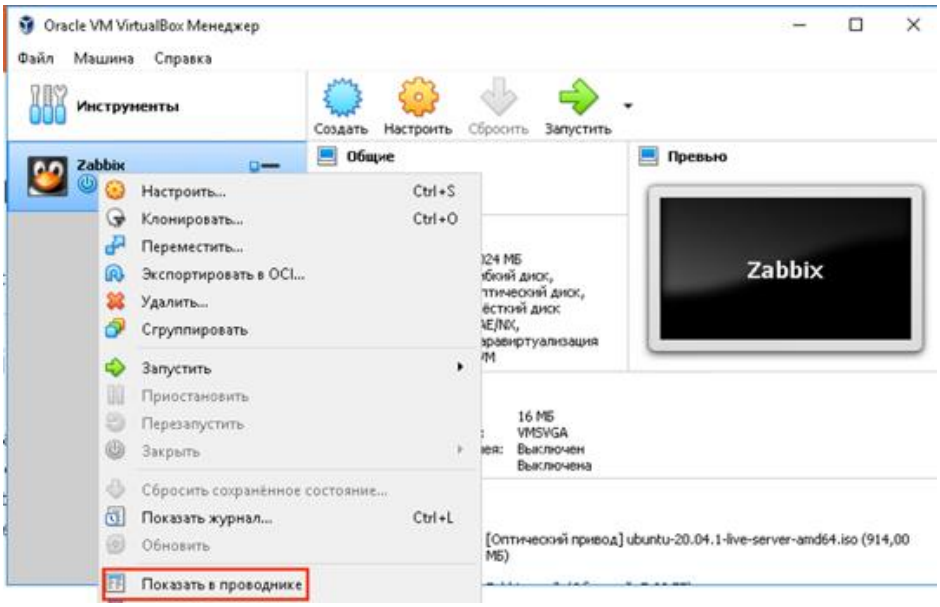

Рис.3.39. Шлях до віртуальних дисків

Перейшовши до папки з віртуальними дисками, потрібний файл для розгортування в Вашій системі має форма .vmdk. (Рис.3.40)

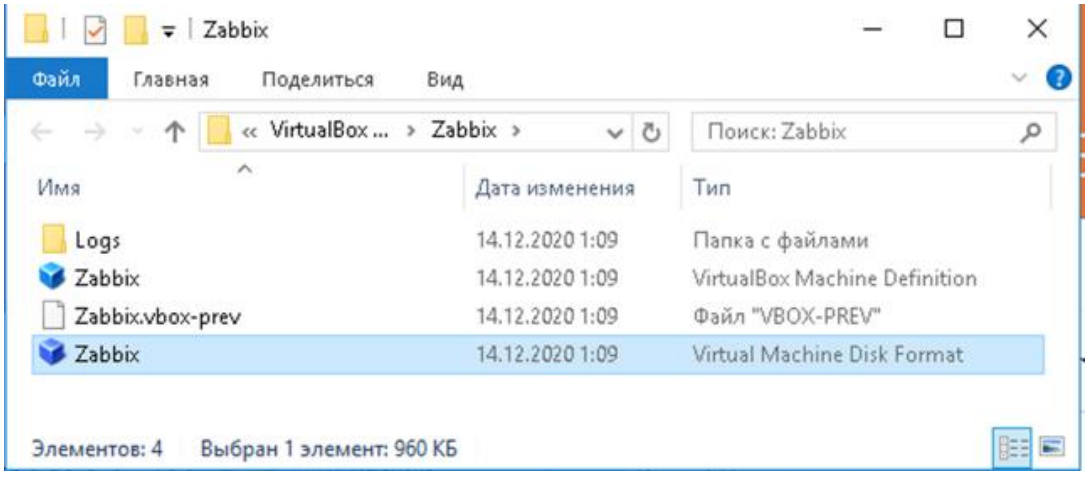

Рис. 3.40. Файл диску віртуальної машини

ВИСНОВОК

Аналіз телеметрії привів до потреби систем моніторингу в IT мережах. Також це дослідження представляє ключові вимоги систем моніторингу нового покоління. Ці системи повинні бути надійними, гнучкими та вимагати низьких експлуатаційних витрат для використання у виробництві. Крім того, я запропонував архітектуру системи, яка задовольняє ідентифікованому вимоги. Використання цієї архітектури допомагає розробити систему з реалізацією усіх вимог, включаючи швидкість та вартість володіння.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ

1. Бараш Л. Моніторинг трафіку в сетях з комутацією пакетів [Трафік

Моніторинг у мережах з комутацією пакетів]. Компьютерное обозрение , 2008, вип. 37 (654),

С. 20-25.

2. Веселуха Г. Л. Інженерний взгляд на моніторингове обладнання Центров

Інженерний погляд на моніторинг обладнання ЦОД

(Досвід впровадження)]. Електронний науковий журнал «Век якості» , 2017, вип. 2,

С. 100-111.

3. Височина О., Шматков С., Мухсін С. А. Аналіз системного моніторингу

аналіз моніторингу телекомунікаційних мереж

Системи]. Радіоелектроніка, інформатика, управління [Радіоелектроніка, інформатика,

Контроль], 2010, вип. 2, стор. 139-142.

4. Гайфулін Т., Костомаров Д. Аналіз сучасних системних моніторингів

Сучасні системи моніторингу]. Известия ТулГУ. Технические науки , 2013, вип. 9, вип. 2,

С. 51-55.

5. Телеметрія: короткий зміст концепції та обгрунтування (звіт). Біб-код : 1987STIN ... 8913455.

6. Мері Белліс, "Телеметрія"