ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
АВТОМАТИЗОВАНИХ СИСТЕМ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Аналіз систем моніторингу для збору показників роботи сервера»

на здобуття освітнього ступеня магістра
зі спеціальності 126 Інформаційні системи та технології
(код, найменування спеціальності)
освітньо-професійної програми Інформаційні системи та технології
(назва)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Валентин ЮХИМЕНКО
(підпис) *Ім'я, ПРІЗВИЩЕ здобувача*

Виконав:
здобувач вищої освіти Валентин ЮХИМЕНКО
група ІСДМ-62

Керівник: Вадим ВЛАСЕНКО
науковий ступінь, вчене звання кандидат технічних наук

Рецензент:
науковий ступінь, вчене звання _____
Ім'я, ПРІЗВИЩЕ

Київ 2023

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення автоматизованих систем
Ступінь вищої освіти Магістр
Спеціальність 126 Інформаційні системи та технлогії
Освітньо-професійна програма 126 Інформаційні системи та технлогії

ЗАТВЕРДЖУЮ
Завідувач кафедрою ІПЗАС

_____ Каміла СТОРЧАК
«_____» _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Юхименко Валентин Миколайович
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Аналіз систем моніторингу для збору показників роботи сервера
керівник кваліфікаційної роботи Власенко В.О. кандидат технічних наук,
(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «19» 10. 2023р. No ___

2. Строк подання кваліфікаційної роботи «28» грудня 2023 р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, аналіз систем моніторингу

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
4.1.Системи моніторингу IT інфраструктури
4.2.Основні системи моніторингу серверного обладнання
4.3.Аналіз систем моніторингу для збору показників сервера

5. Перелік ілюстративного матеріалу: *презентація*
5.1.Зображення зовнішнього вигляду систем моніторингу

5.2.Зображення зовнішнього вигляду систем моніторингу для серверного обладнання
6. Дата видачі завдання «___» _____ 20__ р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Пошук літератури та джерел	19 жовтня – 1 листопада	Виконано
2	Ознайомлення з системами моніторингу та їх видами	1-5 листопада	Виконано
3	Огляд та ознайомлення з основними характеристиками систем моніторингу	5-15 листопада	Виконано
4	Ознайомлення з системами моніторингу для серверного обладнання	15-20 листопада	Виконано
5	Аналіз наявної науково-технічної літератури	20 листопада – 9 грудня	Виконано
6	Оформлення кваліфікаційної роботи	9-14 грудня	Виконано
7	Попередній захист роботи	15 грудня	Виконано
8	Подача роботи в деканат		Виконано

Здобувач(ка) вищої освіти _____ Валентин ЮХИМЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник
кваліфікаційної роботи _____ Вадим ВЛАСЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 74 стор., 27 рис., 1 табл., 17 джерел.

Мета роботи – аналіз існуючих систем моніторингу, які збирають показники серверів.

Об'єкт дослідження – системи моніторингу.

Предмет дослідження – методи, засоби моніторингу серверного обладнання.

Короткий зміст роботи: Моніторинг – це важливий процес, який допомагає організаціям, користувачам, відстежувати зміни в IT-інфраструктурі. Мета моніторингу – збір та аналіз даних про IT-сервіси і компоненти інформаційної інфраструктури та використання цих даних для контролю всіх її елементів, а також запобігання збоїв і поломок. Системи моніторингу – це програмне забезпечення, яке дозволяють відслідковувати ресурси сервера, такі як використання процесора, обсяг пам'яті, потужність сховища, продуктивність введення-виведення, час роботи мережі та інші.

КЛЮЧОВІ СЛОВА: СИСТЕМА МОНІТОРИНГУ, МОНІТОРИНГ, АГЕНТ, ІНФРАСТРУКТУРА, КЛЮЧОВІ ПОКАЗНИКИ ПРОДУКТИВНОСТІ, КРІ, ПРОАКТИВНІСТЬ.

ABSTRACT

Text part of the master's qualification work:

_74_ pages, _27_ pictures, _1_ table, _17_ sources.

The purpose of the work – analysis of existing monitoring systems that collect server indicators.

Object of research – monitoring systems.

Subject of research – methods, means of monitoring server equipment.

Summary of the work: Monitoring is an important process that helps organizations, users, track changes in the IT infrastructure. The purpose of monitoring is to collect and analyze data on IT services and components of the information infrastructure and use this data to control all its elements, as well as prevent failures and breakdowns. Monitoring systems are software that allow you to monitor server resources such as CPU usage, memory capacity, storage capacity, I/O performance, network uptime, and more.

KEYWORDS: MONITORING SYSTEM, MONITORING, AGENT, INFRASTRUCTURE, KEY PERFORMANCE INDICATORS, KPI, PROACTIVITY.

ЗМІСТ

ВСТУП

У зв’язку з постійний розвитком серверного обладнання, та ростом, розширенням його інфраструктури, необхідність якісного відстеження та моніторингу процесів та компонентів, під час його роботи є одним з ключових моментів за якими слідкують ІТ-інженери. Щоб бути в ногу з сьогоденням, компаніям необхідно відмовлятися від старих бізнес-процесів, оцифровувати їх, впроваджувати нові процеси та системи, на зразок CRM і ERP. Без особливої підготовки і захисту, всі непродумані процеси та збої несуть за собою проблеми репутаційного та фінансового плану. Якщо в компанії були лише невеликі проблеми з ІТ-інфраструктурою, компанія також може зіткнутися з нестачею прибутку, нестабільністю та низькою ефективністю. Щоб уникнути подібних результатів впроваджуються рішення з моніторингу ІТ-інфраструктури. Вони допомагають з відстеженням того, що відбувається в корпоративній мережі. В результаті значно підвищується реакція на можливі збої і атаки.

Сучасні системи моніторингу, дуже допомагають інженерам у виконанні їх роботи, та відповідно забезпеченні високого рівня продуктивності, та відмовостійкості обладнання. Оскільки якісно налаштована система моніторингу, буде охоплювати всі системи та додатки, які потребують уваги, та відповідно своєчасно сповіщатиме про аварії, або будь-які інші події.

Сьогодні підприємства вирішують використовувати велику кількість серверів як у хмарі, так і у своїх центрах обробки даних, щоб задовольнити постійно зростаючий попит. В результаті цих змін, технології моніторингу стали вирішальними. Налаштування та адміністрування кількох серверів для бізнесу та додатків стало простіше завдяки прогресу хмарних технологій.

Моніторинг серверів є важливою складовою сучасного життя, оскільки дозволяє виявляти та вирішувати проблеми з ІТ-інфраструктурою до того, як вони вплинуть на критичні бізнес-процеси. Серед основних видів систем моніторингу є комерційні та безкоштовні:

– безкоштовні системи з відкритим вихідним кодом, на кшталт Zabbix. Хоч подібні системи є умовно безкоштовними, але якість їх робіт нітрохи не гірше. Також подібна група відрізняється можливістю гнучко налаштовувати всі інструменти під корпоративні завдання. Адже за допомогою відкритого коду мережеві адміністратори можуть адаптувати рішення для стандартного моніторингу без сторонньої допомоги;

– комерційні системи з закріпленим функціоналом. Їх особливість полягає в тому, що такі системи з легкістю розгортаються та вирішують характерні завдання моніторингу. Єдиний недолік полягає в тому, що комерційні системи менш гнучкі та не так легко адаптуються під нестандартні сценарії. Комерційні системи платформного типу мають складаються з комплексу рішень, які націлені на моніторинг, управління процесів і рішення нестандартних завдань.

1 СИСТЕМИ МОНІТОРИНГУ ІТ ІНФРАСТРУКТУРИ

ІТ-моніторинг є важливою практикою, яка передбачає систематичне спостереження за інфраструктурою інформаційних технологій організації. Основною метою є забезпечення його стабільності, продуктивності та безпеки. ІТ-інструменти моніторингу мають широкий спектр опцій, починаючи від фундаментальних інструментів і закінчуючи більш просунутими рішеннями, які використовують штучний інтелект для прогнозування та запобігання збоям до їх виникнення. Завдяки цьому можна отримувати цінну інформацію про стан і поведінку своїх ІТ-активів.

Проактивний аналіз інфраструктури серверів дозволяє передбачити або виявити проблеми продуктивності, перш ніж вони стануть терміновими, і гарантує, що мережеві ресурси працюють належним чином. У 2014 році дослідження, проведене The Anthesis Group і Стенфордським університетом, виявило, що до 30 відсотків серверів у великих центрах обробки даних працювали та споживали електроенергію без генерації трафіку чи використання будь-яких циклів. Адекватний моніторинг інфраструктури міг би запобігти втраті часу, грошей і простору, пов’язаних із підтримкою непродуктивних серверів.

1.1 Визначення ІТ-моніторингу

Моніторинг ІТ-інфраструктури – це важливий процес, який допомагає організаціям, користувачам, відстежувати зміни в ІТ-інфраструктурі. Мета моніторингу – збір та аналіз даних про ІТ-сервіси і компоненти інформаційної інфраструктури та використання цих даних для контролю всіх її елементів, а також запобігання збоїв і поломок.

Однією з основних цілей ІТ-моніторингу є виявлення потенційних проблем і реагування на них у режимі реального часу. У разі виявлення незвичайних дій або аномалій, ці інструменти можуть генерувати попередження та сповіщення. Вони

дозволяють ІТ-командам вживати швидких заходів для вирішення проблем, перш ніж вони вплинуть на критичні операції. Цей проактивний підхід має вирішальне значення для мінімізації часу простою та підтримки високого рівня доступності обслуговування. ІТ-інфраструктура, яка підтримує сучасне підприємство, може мати низку складних архітектурних форм-факторів: віртуалізовані, програмно визначені, гібридні та мультихмарні, локальні та зовнішні центри обробки даних.

Для моніторингу використовують спеціалізовані системи моніторингу ІТ-інфраструктури, які збирають усі дані і об'єднують їх в єдину базу даних, де вони можуть бути структуровані і проаналізовані.

Окрім усунення несправностей і вирішення проблем, ІТ-моніторинг також відіграє ключову роль в оптимізації використання ресурсів. Ретельно відстежуючи продуктивність ІТ-компонентів, компанії можуть виявляти недостатньо використані або перевантажені ресурси та приймати обґрунтовані рішення щодо ефективного розподілу ресурсів. Це не тільки підвищує ефективність системи, але й допомагає контролювати експлуатаційні витрати. По суті, це незамінна практика, яка сприяє загальному здоров'ю, продуктивності та економічній ефективності ІТ-екосистеми організації.

Базовий моніторинг виконується за допомогою перевірок роботи пристрою, тоді як більш розширений моніторинг забезпечує детальні перегляди робочих станів, таких як середній час відповіді, кількість екземплярів програми, частота помилок і запитів, використання ЦП і доступність програмного забезпечення.

1.2 Ключові елементи моніторингу ІТ-інфраструктури

Через складність сучасної ІТ-інфраструктури рішення для моніторингу продуктивності, які допомагають керувати інфраструктурою, мають вирішальне значення для зменшення простоїв і збільшення часу відгуку.

IT-моніторинг реалізується по-різному залежно від його типу. Однак, як загальний процес, IT-моніторинг охоплює три розділи: основний (Foundation), програмне забезпечення (Software) та інтерпретацію (Interpretation).

Основний (Foundation)

Це моніторинг інфраструктури, є найнижчим рівнем стека програмного забезпечення та зосереджений на моніторингу фізичних і віртуальних пристроїв, таких як сервери або віртуальні машини (ВМ). Цей рівень, який формує основу для розширених можливостей моніторингу, передбачає моніторинг фізичних або віртуальних пристроїв, відомих як «хости». Ці хости охоплюють широкий діапазон, включаючи сервери Windows і Linux, маршрутизатори Cisco, брандмауери Nokia та віртуальні машини VMware і тому подібні. Основний рівень зосереджується на забезпеченні роботи цих хостів шляхом надсилання запитів ping. Після налаштування цей рівень надає перегляд доданих хостів, вказуючи, які з них активні чи неактивні. Ця базова інформація служить основою для розширеного моніторингу.

Програмне забезпечення (Software)

Цей розділ, який іноді називають рівнем моніторингу, аналізує дані з пристроїв у основному розділі. Зібрані тут дані включають використання процесора, навантаження, пам’ять або кількість запущених віртуальних машин. Цей рівень заглиблюється в моніторинг конкретних елементів, що працюють на цих хостах. До прикладу, на серверах Linux – простір підкачки, запущені служби, використання ЦП тощо; на серверах Windows – розмір файлу підкачки, завантаження ЦП, використання пам’яті, доступне сховище на C:/, запущені процеси тощо; для віртуалізації (наприклад, VMware) – доступність сховища даних, температура, кількість віртуальних машин, завантаження ЦП тощо.

Ці контрольовані елементи називаються «перевірками служб», і вони виконуються на хостах, указаних на базовому рівні. Процес, по суті, включає перевірку показників ефективності цих елементів. Інновації в моніторингу призвели до розробки «Автовиявлення». Ця функція дозволяє системам моніторингу сканувати та виявляти пристрої в межах попередньо визначених

підмереж або мереж. Наприклад, у випадку серверів Windows сканування підмережі дозволяє системі виявити та імпортувати всі хости в цій мережі. Система моніторингу також може визначати операційну систему цих хостів і автоматично застосовувати шаблони на основі результатів, забезпечуючи швидке отримання результату.

Інтерпретація (Interpretation)

зібрані дані та показники інтерпретуються та представлені у вигляді графіків або діаграм даних, часто на інформаційній панелі користувацького інтерфейсу (GUI). В IT, сервери та мережеві пристрої об'єднуються, щоб утворити більші об'єкти, такі як програми, веб-сайти або веб-сервіси. Основну увагу слід зосередити на моніторингу цих більших об'єктів, а не їхніх компонентів. Зрештою, головне занепокоєння – це вплив IT-проблем на бізнес та його клієнтів. Щоб вирішити цю проблему, постачальники програмного забезпечення для моніторингу запровадили «моніторинг бізнес-послуг». Моніторинг бізнес-послуг дозволяє користувачам отримати уявлення про продуктивність програм, стеків, веб-сайтів та інших складних об'єктів. Він зосереджується на їхньому здоров'ї в цілому, а не на стані їхніх компонентів. Він надає «погляд зверху вниз» на послуги, встановлюючи пріоритетність впливу на бізнес-послуги, а не аналізуючи базові компоненти. Це часто досягається за допомогою інтеграції з інструментами, які спеціально зосереджені на візуалізації даних.

1.3 Моніторинг за допомогою агента або безагентний

IT-моніторинг може працювати, за допомогою агентів або бути взагалі без агентів, відповідно через це, є безліч дискусій між адміністратора та інженерами, який моніторинг краще. Існують плюси і мінуси обох підходів, і найповніша стратегія передбачає поєднання обох.

Моніторинг за допомогою агента

Моніторинг на основі агентів зазвичай розроблений спеціально для конкретної платформи. Як наслідок, він здатний збирати та аналізувати більше даних для системи, для взаємодії з якою він був запрограмований.

Агенти – це незалежні програми, які встановлюються на контрольований пристрій для збору даних про продуктивність апаратного чи програмного забезпечення та передачі їх на сервер керування. Ці системи моніторингу часто використовуються для відстеження системних ресурсів, таких як використання ЦП і частота, або обсяг вільної оперативної пам'яті. Вони також використовуються для відображення таких елементів, як вільний простір на одному або кількох жорстких дисках, температура ЦП та інших важливих компонентів, і мережева інформація, включаючи IP-адресу системи та поточну швидкість завантаження та вивантаження. Інші можливі показники можуть включати дату й час, час безвідмовної роботи системи, ім'я комп'ютера, ім'я користувача, дані SMART жорсткого диска , швидкість вентилятора та напругу, що забезпечується джерелом живлення.

Незважаючи на те, що спеціальне програмування постачальника дозволяє отримувати детальніші дані, воно також є запатентованим, що ускладнює перехід на іншу платформу без втрати даних. Якщо компоненти інфраструктури використовують агентів, користувачам також потрібно оцінити, чи їх система моніторингу сумісна з цими системами.

Для прикладу, щоб контролювати використання сервера, IT-адміністратор встановлює агент на сервері. Сервер керування отримує ці дані від агента та відображає їх користувачеві через інтерфейс IT-системи моніторингу, часто як графік продуктивності в часі. Якщо сервер перестає працювати належним чином, інструмент сповіщає адміністратора, який може виконати певні дії, для виправлення несправності, оновити або замінити елемент, доки він не відповідатиме стандартам роботи.

Безагентний моніторинг

Безагентний моніторинг є популярним вибором, який покладається на різноманітні протоколи, такі як SNMP, WMI, SSH, NetFlow або інші, для передачі

системних даних і статистики програмному забезпеченню моніторингу. Ці вбудовані функції відстежують і керують інформацією про інфраструктуру без додаткових агентів.

Мережеві пристрої, сервери, пристрої потоку, пристрої зберігання даних і віртуальні машини, як-от VMware і Hyper-V, – це звичайні компоненти, які мають можливості моніторингу без агентів. Хороша система моніторингу інфраструктури може централізовано керувати компонентами без агентів.

Як варіант безагентного моніторингу, є апаратні системи, які відстежують подібну інформацію. Зазвичай вони займають один або кілька відсіків для дисків на передній панелі корпусу комп’ютера та або напряму взаємодіють із апаратним забезпеченням системи, або підключаються до програмної системи збору даних через USB. При будь-якому підході до збору даних система моніторингу відображає інформацію на маленькій РК-панелі або на серії невеликих аналогових чи світлодіодних цифрових дисплеїв. Деякі апаратні системні монітори також дозволяють безпосередньо контролювати швидкість вентилятора, дозволяючи користувачеві швидко налаштувати охолодження в системі. В сегменті дорогих материнських плат, є моделі, які мають встановлені з заводу виробника, певні варіанти апаратного системного монітора. Ці системи безпосередньо використовують датчики, вбудовані в систему, надаючи більш детальну та точну інформацію, ніж зазвичай надають менш дорогі системи моніторингу.

1.4 Моніторинг за допомогою SNMP

SNMP (англ. Simple Network Management Protocol – простий протокол керування мережею) – це протокол керування мережами зв'язку на основі архітектури TCP/IP. SNMP – це технологія, покликана забезпечити керування й контроль за пристроями й застосунками в мережі зв'язку шляхом обміну керівною інформацією між агентами, що розташовуються на мережних пристроях, і менеджерами, розташованими на станціях керування. SNMP визначає мережу як

сукупність мережних керівних станцій й елементів мережі (головні машини, шлюзи й маршрутизатори, термінальні сервери), які спільно забезпечують адміністративні зв'язки між мережними керівними станціями й мережними агентами. SNMP різних версій присвячений цілий ряд рекомендацій IETF (RFC). Зазвичай при використанні SNMP присутні керовані та керівні системи. До складу керованої системи входить компонент, який називається агентом, який відправляє звіти керівній системі. По суті SNMP агенти передають управлінську інформацію на керівні системи як змінні (такі як «вільна пам'ять», «ім'я системи», «кількість процесів, що працюють» тощо).

Мережа, керована SNMP, (рис. 1.1), складається з трьох ключових компонентів:

– керовані пристрої – це мережевий вузол, який реалізує інтерфейс SNMP, що забезпечує односпрямований (лише читання) або двонаправлений (читання та запис) доступ до інформації вузла. Керовані пристрої обмінюються інформацією про вузли з NMS. Керовані пристрої, які іноді називають елементами мережі, можуть бути будь-якими типами пристроїв, включаючи, але не обмежуючись, маршрутизатори, сервери доступу, комутатори, кабельні модеми, мости, концентратори, IP-телефони, IP-відеокамери, комп’ютерні хости та принтери;

– агент – програмне забезпечення, яке працює на керованих пристроях, це програмний модуль для керування мережею, який знаходиться на керованому пристрої. Агент має локальні знання інформації про керування та перекладає цю інформацію у форму, специфічну для SNMP, або з неї;

– станція керування мережею (NMS) – програмне забезпечення, яке працює на диспетчері, виконує програми, які відстежують і контролюють керовані пристрої. NMS забезпечують основну частину ресурсів обробки та пам'яті, необхідних для керування мережею. Одна чи декілька NMS можуть існувати в будь-якій керованій мережі.

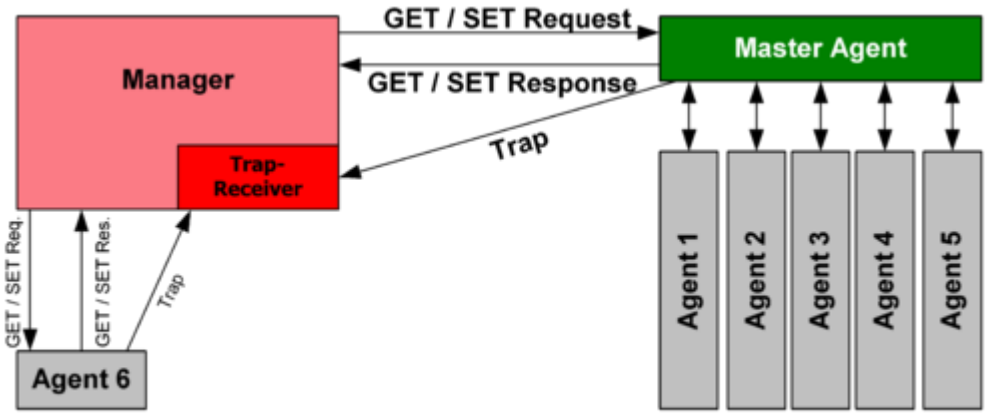


Рисунок 1.1 Принцип зв’язку SNMP

Система що керує, може отримати достовірну інформацію через операції протоколу GET, GETNEXT і GETBULK. Агент може самостійно без запиту надсилати дані, використовуючи операцію протоколу TRAP або INFORM. Керівні системи можуть також відправляти конфігураційні оновлення або контрольні запити, використовуючи операцію SET для безпосереднього управління системою. Операції конфігурування та управління використовуються тільки тоді, коли потрібні зміни у мережній інфраструктурі. Операції моніторингу зазвичай виконуються на регулярній основі.

Змінні, доступні через SNMP, організовані в ієрархії. Ці ієрархії та інші метадані (такі як тип і опис змінної) описуються Базами Керівної Інформації (англ. Management Information Bases (MIBs)).

Management Information Base (MIB)

SNMP не визначає, яку інформацію (які змінні) керована система повинна надавати. Навпаки, SNMP використовує розширювану модель, в якій доступна інформація визначається Базами Керівної Інформації (MIB – Management Information Base). Бази Керівної Інформації описують структуру керівної інформації пристроїв. Вони використовують ієрархічний адресний простір імен, що містить унікальний ідентифікатор об'єкта (англ. object identifier (OID)). Грубо кажучи, кожен унікальний ідентифікатор об'єкта визначає змінну, яка може бути прочитана чи встановлена через SNMP. MIB'и використовують нотацію, визначену в ASN.1.

Ієрархія MIB може бути зображена як дерево з безіменним коренем, рівні якого приписані різними організаціями. На найвищому рівні MIB OID'и належать різним організаціям, що займаються стандартизацією, в той час як на нижчих рівнях OID'и виділяються асоційованими організаціями. Ця модель забезпечує управління на всіх шарах мережної моделі OSI, адже MIB'и можуть бути визначені для будь-яких типів даних і операцій.

Керований об'єкт – це одна з будь-якого числа характеристик, специфічних для керованого пристрою. Керований об'єкт містить у собі один або більше екземплярів об'єкта (що ідентіфікуються за OID), які насправді є змінними.

Існує два типи керованих об'єктів:

– скалярні об'єкти визначають єдиний екземпляр об'єкта;

– табличні об'єкти визначають множинні, пов'язані екземпляри об'єктів які групуються в таблицях MIB.

Прикладом керованого об'єкта може бути atInput, який є скалярним об'єктом що містить єдиний екземпляр об'єкта, ціле число, яке показує загальну кількість вхідних пакетів AppleTalk на мережний інтерфейс маршрутизатора.

Ідентифікатор об'єкта (OID) унікально ідентифікує керований об'єкт в ієрархії MIB.

Abstract Syntax Notation One (ASN.1)

В телекомунікаціях і комп'ютерних мережах, ASN.1 є стандартною гнучкою нотацією для опису структур даних, що служать для кодування, передачі і декодування даних. ASN.1 являє собою набір правил для опису структури об'єктів, незалежних від специфічних для обладнання методик кодування, і формальну нотацію, яка дозволяє уникнути неоднозначностей.

ASN.1 це єдиний ISO та ITU-T стандарт, спочатку визначений у 1984 році як частина стандарту CCITT X.409: 1984. Пізніше, в 1988 році, завдяки його широкому застосуванню, він був виділений в окремий стандарт X.208. Значно переглянута версія 1995 року описана в X.680.

Адаптована підмножина ASN.1 Структура Керуючої Інформації (SMI) описана в протоколі SNMP для визначення наборів пов'язаних MIB об'єктів, також званих MIB модулями.

Деталізація роботи протоколу SNMP

SNMP працює на прикладному рівні набору протоколів Інтернету. Усі повідомлення SNMP транспортуються через протокол UDP (User Datagram Protocol). Агент SNMP отримує запити на UDP-порт 161. Менеджер може надсилати запити з будь-якого доступного вихідного порту на порт 161 агента. Відповідь агента надсилається назад до вихідного порту диспетчера. Менеджер отримує сповіщення (Traps та InformRequests) на порт 162. Агент може генерувати сповіщення з будь-якого доступного порту. При використанні з захистом транспортного рівня або безпекою транспортного рівня датаграм запити надходять на порт 10161, а сповіщення надсилаються на порт 10162. SNMPv1 визначає п’ять основних протокольних блоків даних (PDU – Protocol Data Unit). Два інші PDU, GetBulkRequest і InformRequest, були додані в SNMPv2, а PDU Report додано в SNMPv3. (табл. 1.1)

Таблиця 1.1 Принцип побудови PDU SNMP

IP header	UDP header	version	community	PDU-type	request-id	error-status	error-index	variable bindings

Відповідно до цього, наведено сім типів PDU SNMP, визначених полем PDU-type:

– GetRequest – запит менеджера до агента для отримання значення змінної або списку змінних. Бажані змінні вказуються у зв’язках змінних (поле значення не використовується). Отримання вказаних значень змінних виконується агентом як атомарна операція. Повертається відповідь із поточними значеннями;

– SetRequest – запит менеджера до агента на зміну значення змінної або списку змінних. Зв'язки змінних вказуються в тілі запиту. Зміни до всіх указаних змінних

повинні бути зроблені агентом як атомарна операція. Повертається відповідь із (поточними) новими значеннями для змінних;

– GetNextRequest – запит менеджера до агента для виявлення доступних змінних та їхніх значень. Повертає відповідь зі зв'язуванням змінної для лексикографічної наступної змінної в MIB. Весь MIB агента можна пройти за допомогою ітераційного застосування GetNextRequest, починаючи з OID 0. Рядки таблиці можна прочитати, вказавши OID стовпців у прив'язках змінних запиту;

– GetBulkRequest – запит менеджера до агента для кількох ітерацій GetNextRequest. Оптимізована версія GetNextRequest. Повертає відповідь із кількома прив'язками змінних, отриманими від прив'язки змінної або прив'язок у запиті. Для керування поведінкою відповіді використовуються спеціальні PDU - неповторювачі та поля максимального повторення. GetBulkRequest було представлено в SNMPv2;

– Response – повертає прив'язки змінних і підтвердження від агента до менеджера для GetRequest, SetRequest, GetNextRequest, GetBulkRequest і InformRequest. Звіти про помилки забезпечуються полями статусу помилки та індексу помилки. Хоча він використовувався як відповідь як на отримання, так і на набори, цей PDU називався GetResponse у SNMPv1;

– Trap – асинхронне повідомлення від агента до менеджера. Хоча в іншому зв'язку SNMP менеджер активно запитує інформацію від агента, це PDU, які надсилаються від агента до менеджера без явного запиту. Перехоплення SNMP дозволяють агенту повідомляти станцію керування про важливі події за допомогою небажаного повідомлення SNMP. PDU перехоплення включають поточне значення sysUpTime, OID, що визначає тип перехоплення, і додаткові прив'язки змінних. Адресація призначення для перехоплень визначається залежно від додатка способом, як правило, за допомогою змінних конфігурації перехоплень у MIB. Формат повідомлення перехоплення було змінено в SNMPv2, а PDU перейменовано на SNMPv2-Trap;

– InformRequest – підтверджене асинхронне сповіщення. Цей PDU був представлений у SNMPv2 і спочатку був визначений як зв’язок між менеджерами. Пізніші реалізації послабили оригінальне визначення, щоб дозволити зв’язок агента з менеджером. Сповіщення від менеджера до менеджера вже були можливі в SNMPv1 за допомогою перехоплення, але оскільки SNMP зазвичай працює через UDP, де доставка не гарантується, а скинуті пакети не повідомляються, доставка перехоплення не гарантується. InformRequest виправляє це, оскільки після отримання повертається підтвердження.

1.5 Використання SSH в процесі моніторингу безагентним методом

Протокол Secure Shell (SSH) – це криптографічний мережевий протокол для безпечної роботи мережевих служб у незахищеній мережі. Його найбільш відомі програми – віддалений вхід і виконання командного рядка.

Програми SSH засновані на архітектурі клієнт-сервер, з’єднуючи примірник клієнта SSH із сервером SSH. SSH працює як багаторівневий набір протоколів, що складається з трьох основних ієрархічних компонентів: транспортний рівень забезпечує автентифікацію сервера, конфіденційність і цілісність; протокол автентифікації користувача підтверджує користувача на сервері; і протокол з'єднання мультиплексує зашифрований тунель у кілька логічних каналів зв'язку.

SSH був розроблений для Unix-подібних операційних систем як заміна Telnet і незахищених віддалених протоколів оболонки Unix, таких як Berkeley Remote Shell (rsh) і пов’язаних протоколів rlogin і rexec, які використовують незахищену передачу відкритим текстом маркерів автентифікації.

SSH вперше був розроблений у 1995 році фінським комп’ютерним науковцем Тату Юлененом. Подальша розробка набору протоколів тривала кількома групами розробників, створюючи кілька варіантів реалізації. Специфікація протоколу розрізняє дві основні версії, які називаються SSH-1 і SSH-2. Найпоширенішим

стеком програмного забезпечення є OpenSSH, випущений у 1999 році як програмне забезпечення з відкритим кодом розробниками OpenBSD. Реалізації поширюються для всіх типів операційних систем, які широко використовуються, включаючи вбудовані системи.

Детальніше про SSH

Коли пара відкритих і закритих ключів створюється користувачем вручну, автентифікація по суті виконується під час створення пари ключів, а потім сеанс може бути відкритий автоматично без запиту пароля. У цьому сценарії відкритий ключ розміщується на всіх комп'ютерах, які мають надати доступ власнику відповідного закритого ключа, який власник зберігає закритим. Хоча автентифікація базується на закритому ключі, ключ ніколи не передається через мережу під час автентифікації. SSH лише перевіряє, що особа, яка пропонує відкритий ключ, також володіє відповідним закритим ключем.

У всіх версіях SSH важливо перевіряти невідомі відкриті ключі, тобто пов'язувати відкриті ключі з ідентифікаторами, перш ніж вважати їх дійсними. Прийняття відкритого ключа зловмисника без перевірки авторизує неавторизованого зловмисника як дійсного користувача.

SSH використовує криптографію з відкритим ключем для автентифікації віддаленого комп'ютера та дозволяє йому автентифікувати користувача, якщо це необхідно.

SSH можна використовувати в кількох методологіях. Найпростішим способом є те, що обидва кінці каналу зв'язку використовують автоматично створені пари відкритого та закритого ключів для шифрування мережевого з'єднання, а потім використовують пароль для автентифікації користувача.

Керування ключами OpenSSH

У Unix-подібних системах список авторизованих відкритих ключів зазвичай зберігається в домашньому каталозі користувача, якому дозволено віддалено входити в систему, у файлі ~/.ssh/authorized_keys. Цей файл поважається SSH, лише якщо він не доступний для запису нікому, крім власника та кореня. Коли відкритий ключ присутній на віддаленому кінці, а відповідний закритий ключ присутній на

локальному кінці, вводити пароль більше не потрібно. Однак для додаткової безпеки сам закритий ключ можна заблокувати за допомогою парольної фрази.

Закритий ключ також можна шукати в стандартних місцях, а його повний шлях можна вказати в параметрах командного рядка (опція – і для ssh). Утиліта ssh-keygen створює відкритий і закритий ключі, завжди в парах.

SSH також підтримує автентифікацію на основі пароля, яка шифрується автоматично згенерованими ключами. У цьому випадку зловмисник може імітувати законну сторону сервера, запитати пароль і отримати його (атака «людина посередині»). Однак це можливо лише в тому випадку, якщо обидві сторони раніше не проходили автентифікацію, оскільки SSH запам’ятовує ключ, який раніше використовувала сторона сервера. Клієнт SSH видає попередження перед тим, як прийняти ключ нового, раніше невідомого сервера. Автентифікацію пароля можна вимкнути на стороні сервера.

Використання SSH

SSH зазвичай використовується для входу на віддалену машину та виконання команд, але він також підтримує тунелювання, пересилання портів TCP і з’єднань X11; він може передавати файли за допомогою пов’язаних протоколів передачі файлів SSH (SFTP) або безпечного копіювання (SCP). SSH використовує модель клієнт-сервер.

Клієнтська програма SSH зазвичай використовується для встановлення з’єднань із демоном SSH, таким як sshd, який приймає віддалені з’єднання. Обидва зазвичай присутні в більшості сучасних операційних систем, включаючи macOS, більшість дистрибутивів Linux, OpenBSD, FreeBSD, NetBSD, Solaris і OpenVMS. Примітно, що версії Windows до Windows 10 версії 1709 не включають SSH за замовчуванням. Існують пропрієтарні, безкоштовні та з відкритим кодом (наприклад, PuTTY і версія OpenSSH, яка є частиною Cygwin) різного рівня складності та повноти. Менеджери файлів для UNIX-подібних систем (наприклад, Konqueror) можуть використовувати протокол FISH , щоб забезпечити графічний інтерфейс розділеної панелі з функцією перетягування. Програма Windows з відкритим вихідним кодом WinSCP забезпечує подібні можливості керування

файлами (синхронізація, копіювання, віддалене видалення) за допомогою PuTTY як серверної частини. І WinSCP, і PuTTY доступні в пакетах для запуску безпосередньо з USB-накопичувача, не вимагаючи встановлення на клієнтській машині. Розширення безпечної оболонки для браузера Chrome також дозволяє SSH-з'єднання без встановлення будь-якого програмного забезпечення та навіть дозволяє SSH з комп'ютера Chromebook. Налаштування SSH-сервера в Windows зазвичай включає ввімкнення функції в програмі «Налаштування». У Windows 10 версії 1709 доступний офіційний порт OpenSSH для Win32.

SSH важливий у хмарних обчисленнях для вирішення проблем з підключенням, уникаючи проблем безпеки, пов'язаних із відкриттям хмарної віртуальної машини безпосередньо в Інтернеті. Тунель SSH може забезпечити безпечний шлях через Інтернет через брандмауер до віртуальної машини.

IANA призначила для цього протоколу TCP-порт 22, UDP-порт 22 і SCTP-порт 22. IANA перерахувала стандартний TCP-порт 22 для серверів SSH як один із добре відомих портів ще в 2001 році. SSH також можна запускати за допомогою SCTP, а не TCP як протоколу транспортного рівня, орієнтованого на підключення.

Відповідно, SSH – це протокол, який можна використовувати для багатьох програм на багатьох платформах, включаючи більшість варіантів Unix (Linux, BSD, включаючи macOS від Apple і Solaris), а також Microsoft Windows. Для деяких із наведених нижче програм можуть знадобитися функції, які доступні або сумісні лише з певними клієнтами чи серверами SSH. Наприклад, використання протоколу SSH для реалізації VPN можливе, але наразі лише з реалізацією сервера та клієнта OpenSSH. Тож, якщо детальніше про використання SSH:

– для входу в оболонку на віддаленому хості (замінюючи Telnet і rlogin);

– для виконання однієї команди на віддаленому хості (заміна rsh);

– для налаштування автоматичного входу (без пароля) на віддалений сервер (наприклад, за допомогою OpenSSH);

– у поєднанні з rsync для ефективного та безпечного резервного копіювання, копіювання та віддзеркалення файлів;

– для перенаправлення порту;

– для тунелювання (не плутати з VPN, який направляє пакети між різними мережами або з’єднує два широкомовні домени в один);

– для використання в якості повноцінного зашифрованого VPN. Зауважте, що лише сервер і клієнт OpenSSH підтримують цю функцію;

– для пересилання X з віддаленого хоста (можливо через кілька проміжних хостів);

– для перегляду веб-сторінок через зашифроване проксі-з'єднання з клієнтами SSH, які підтримують протокол SOCKS;

– для безпечного монтування каталогу на віддаленому сервері як файлової системи на локальному комп’ютері за допомогою SSHFS;

– для автоматизованого віддаленого моніторингу та керування серверами за допомогою одного або кількох із описаних вище механізмів;

– для розробки на мобільних або вбудованих пристроях, які підтримують SSH;

– для захисту протоколів передачі файлів.

Використання SSH в кількох механізмах передачі файлів:

– захищена копія (SCP – source copy), яка розвинулася з протоколу RCP через SSH;

– rsync, який має бути більш ефективним, ніж SCP. Зазвичай працює через з’єднання SSH;

– протокол передачі файлів SSH (SFTP – SSH File Transport Protocol), безпечна альтернатива FTP (не плутати з FTP через SSH або FTPS);

– файли, що передаються через протокол оболонки (FISH – Files transferred over shell protocol), випущений у 1998 році, який розвинувся з команд оболонки Unix через SSH;

– швидкий і безпечний протокол (FASP – Fast and Secure Protocol), він же Aspera, використовує SSH для керування та порти UDP для передачі даних.

1.6 WMI в моніторингу безагентним методом

Інструмент керування Windows (WMI) складається з набору розширень до моделі драйвера Windows, яка забезпечує інтерфейс операційної системи, через який інструментальні компоненти надають інформацію та сповіщення. WMI – це реалізація Microsoft стандартів Web-Based Enterprise Management (WBEM) і Common Information Model (CIM) від Distributed Management Task Force (DMTF).

WMI дозволяє мовам сценаріїв (таким як VBScript або Windows PowerShell) керувати персональними комп'ютерами та серверами Microsoft Windows як локально, так і віддалено. WMI попередньо встановлено в ОС Windows 2000 – Windows 11. Він доступний для завантаження для Windows NT і Windows 95 до Windows 98.

Корпорація Майкрософт також надає інтерфейс командного рядка для WMI під назвою Windows Management Instrumentation Command-line (WMIC). Проте WMIC застарів, починаючи з Windows 10 версії 21H1, Windows 11 і Windows Server 2022.

Метою WMI є визначення власного набору незалежних від середовища специфікацій, які дозволяють спільно використовувати інформацію про керування між програмами керування. WMI визначає корпоративні стандарти керування та відповідні технології для Windows, які працюють із існуючими стандартами керування, такими як Desktop Management Interface (DMI) і SNMP. WMI доповнює ці інші стандарти, надаючи єдину модель. Ця модель представляє кероване середовище, за допомогою якого загальним способом можна отримати доступ до даних керування з будь-якого джерела.

Особливості та інструменти WMI

Для тих, хто бажає розробити один або кілька постачальників WMI, WMI пропонує багато функцій із коробки. Ось найважливіші переваги:

– інтерфейси автоматизації – оскільки WMI поставляється з набором інтерфейсів автоматизації, готових до використання, усі функції керування, які

підтримуються постачальником WMI та його набором класів, отримують безкоштовну підтримку сценаріїв із коробки. Окрім дизайну класу WMI та розробки постачальника, командам розробки та тестування Microsoft не потрібно створювати, перевіряти чи тестувати модель сценаріїв, оскільки вона вже доступна в WMI;

– інтерфейси керування.NET – оскільки простір імен System.Management покладається на існуючу систему COM/DCOM, створений постачальник WMI та його набір класів WMI автоматично стає доступним для всіх програм .NET незалежно від мови, що використовується (наприклад, C#, VB.NET). Окрім дизайну класу WMI та розробки постачальника, як для сценаріїв, командам розробників і тестувальників Microsoft не потрібно створювати, перевіряти та тестувати нові збірки для підтримки нового простору імен у .NET Framework, оскільки ця підтримка вже доступна від WMI для безкоштовно;

– інтерфейси програмування C/C++ COM/DCOM – як і більшість компонентів Windows, програмісти COM/DCOM можуть використовувати функції постачальника, які вони розробляють, на рівні інтерфейсів COM/DCOM. Як і в попередніх середовищах (сценарії та .NET Framework), споживачу COM/DCOM просто потрібно взаємодіяти зі стандартним набором COM-інтерфейсів WMI, щоб використовувати можливості постачальника WMI та його набір підтримуваних класів WMI. Щоб зробити всю інформацію про керування доступною з власних API, розробнику постачальника WMI потрібно просто взаємодіяти з набором попередньо визначених COM-інтерфейсів WMI. Це автоматично зробить інформацію про керування доступною на рівні WMI COM. Крім того, об’єктна модель COM-інтерфейсу сценаріїв дуже схожа на об’єктну модель інтерфейсу COM/DCOM, що полегшує розробникам ознайомлення зі сценаріями;

– можливості віддаленого керування через DCOM і SOAP – більше, ніж просто надання локальних можливостей COM, оскільки управління полягає в усьому віддаленому підключенні, WMI пропонує транспорт DCOM. Крім того, протокол SOAP буде доступний у Windows Server 2003 R2 через ініціативу WS-Management

під керівництвом Microsoft, Intel, Sun Microsystems і Dell. Ця ініціатива дозволяє віддалено запускати будь-які сценарії або використовувати дані WMI через певний набір інтерфейсів, що обробляють запити/відповіді SOAP. Перевага для розробника постачальника WMI полягає в тому, що коли він відкриває всі свої функції через WMI, Windows Remote Management/WS-Management, у свою чергу, також може використовувати цю інформацію (вбудовані об'єкти в екземпляри WMI не підтримуються в Windows Server 2003 R2. Це однак ціль для Vista). Рішення WMI/WS-Management забезпечує безкоштовне нанесення всіх шарів на WS-Management і відображення моделі даних CIM на SOAP. У випадку, якщо потрібно використовувати DCOM, впровадження DCOM вимагає наявності проксі DLL, розгорнутої на кожній клієнтській машині. Оскільки WMI доступний в операційній системі Windows з Windows 2000, ці проблеми усунено;

– підтримка запитів – WMI пропонує підтримку запитів WQL із коробки. Це означає, що якщо постачальник не призначений для підтримки запитів, WMI підтримує це за допомогою техніки перерахування поза постачальником;

– можливості створення подій – WMI пропонує можливість сповіщати абонента про будь-яку подію, яка його цікавить. WMI використовує мову запитів WMI (WQL) для надсилання запитів на події WQL і визначає тип подій, які потрібно повернути. Механізм створення подій із усіма пов'язаними зворотними викликами є частиною інтерфейсів WMI COM/DCOM та автоматизації. Будь-хто, хто пише провайдера WMI, може отримати переваги цієї функції безкоштовно для своїх клієнтів. Споживач вирішує, як він хоче споживати інформацію про керування, надану постачальником WMI, і відповідний набір класів WMI;

– генератор шаблонів коду – щоб прискорити процес написання постачальника WMI, включаючи всі інтерфейси COM/DCOM і відповідні визначення, команда WMI розробила майстер WMI ATL для створення шаблону коду, що реалізує постачальника. Згенерований код базується на моделі класу WMI, спочатку розробленій розробником. Розробник постачальника WMI зможе зв'язати попередньо визначені інтерфейси COM/DCOM для постачальника WMI з його

набором власних API, які отримують інформацію керування для надання. Вправа полягає в заповненні «прогалин» у коді провайдера для створення бажаної логіки інтерфейсу;

– передбачуваність – передбачуваність є важливою проблемою для IT-фахівців, оскільки вона визначає здатність людини, яка має досвід роботи з набором інтерфейсів керування компонентом Windows, застосувати ці знання відразу, інтуїтивно, до будь-якого іншого керованого компонента Windows, не вивчаючи все з самого початку. вгору. Передбачуваність для клієнта є справжньою перевагою, оскільки вона збільшує повернення інвестицій (ROI). Людина, яка стикається з такою ситуацією, просто очікує, що все буде працювати так само, виходячи зі свого попереднього досвіду. Постійне збільшення кількості COM-інтерфейсів для програмування/скриптів має величезний вплив на передбачуваність, оскільки це ускладнює автоматизацію, керування Windows і використання існуючих знань для клієнтів. WMI з CIM вирішує цю проблему, завжди показуючи ту саму об’єктну модель програмування (COM/DCOM, Automation, .NET), незалежно від керованої сутності;

– захист наявних інвестицій клієнтів – захист інвестицій клієнтів і партнерів мотивує клієнтів інвестувати в технології. Оскільки останніми роками корпорація Майкрософт багато інвестувала в написання WMI, клієнти та партнери інвестували в інструменти, які використовують можливості WMI Windows. Тому вони природно продовжують використовувати ці можливості замість того, щоб використовувати новий набір конкретних інтерфейсів для кожного керованого компонента Windows. Конкретний набір інтерфейсів означає наявність певного набору агентів або власно розробленого програмного забезпечення на основі нової моделі або набору інтерфейсів, спеціально призначених для компонента чи технології. Використовуючи можливості WMI сьогодні, клієнти та партнери можуть використовувати інвестиції, зроблені в минулому, мінімізуючи витрати на розробки, навчання та нові відкриття. Це також матиме великий вплив на

стабільність і надійність їхньої інфраструктури, оскільки вони продовжуватимуть використовувати існуючу реалізацію з удосконаленою технологією;

– забезпечте логічну та уніфіковану модель адміністрування – як коротко описано раніше у вступі, ця модель базується на галузевому стандарті під назвою CIM, визначеному DMTF (https://www.dmtf.org/). Схема на основі класів CIM визначається консорціумом конструкторів і розробників програмного забезпечення, що відповідає вимогам галузі. Це означає, що не лише Microsoft використовує можливості WMI, але й будь-які сторонні конструктори чи розробники пишуть власний код, який відповідає моделі. Наприклад, Intel робить це для деяких своїх мережевих драйверів і програмного забезпечення. HP використовує існуючі постачальники WMI та впроваджує власні постачальники WMI у програмне забезпечення HP Open View Enterprise Management. IBM використовує WMI із пакету керування Tivoli, MOM і SMS також споживають і надають інформацію WMI. Нарешті, Windows XP SP2 використовує WMI для отримання інформації про статус від антивірусного програмного забезпечення та брандмауерів.

Деякі інструменти WMI також можуть бути корисними на етапах проектування та розробки:

– компілятор MOF (MOFComp.exe): компілятор Managed Object Format (MOF) аналізує файл, що містить оператори Managed Object Format, і додає класи та екземпляри класів, визначені у файлі, до репозиторію CIM. Формат MOF – це спеціальний синтаксис для визначення представлення класу CIM у файлі ASCII (наприклад, MIB для SNMP – те, що файли MOF для CIM). MOFComp.exe входить до складу кожної інсталяції WMI. Кожне визначення, наявне в репозиторії CIM, спочатку визначається у файлі MOF. Файли MOF знаходяться в %SystemRoot%\System32\WBEM. Під час налаштування WMI вони завантажуються в репозиторій CIM.

– інструменти адміністрування WMI: Інструменти адміністрування WMI складаються з чотирьох інструментів: WMI CIM Studio, WMI Object Browser, WMI

Event Registration і WMI Event Viewer. Найважливішим інструментом для розробника постачальника WMI є WMI CIM Studio, оскільки він допомагає у початковому створенні класу WMI у сховищі CIM. Він використовує веб-інтерфейс для відображення інформації та покладається на набір компонентів ActiveX, встановлених у системі під час першого запуску. WMI CIM Studio надає можливість: підключіться до вибраної системи та перегляньте репозиторій CIM у будь-якому доступному просторі імен; шукайте класи за їх назвою, описом або назвою властивості; перегляньте властивості, методи та асоціації, пов’язані з певним класом; перегляньте екземпляри, доступні для даного класу досліджуваної системи; виконуйте запити мовою WQL; створіть файл MOF на основі вибраних класів; скомпілюйте файл MOF, щоб завантажити його в репозиторій CIM.

– WinMgmt.exe: WinMgmt.exe не є інструментом; це виконуваний файл, який реалізує службу WMI Core. У сімействі операційних систем Windows NT WMI працює як служба. На комп’ютерах під керуванням Windows 98, Windows 95 або Windows Me WMI працює як програма. У сімействі операційних систем Windows NT також можна запускати цей виконуваний файл як програму, у цьому випадку виконуваний файл запускається в контексті поточного користувача. Для цього спочатку потрібно зупинити службу WMI. Виконуваний файл підтримує деякі параметри, які можуть бути корисними під час запуску WMI як служби або програми. Розробники постачальників WMI, які можуть захотіти налагодити своїх постачальників, по суті, повинні запускати службу WMI як додаток.

– WBEMTest.exe: WBEMTest.exe – це засіб тестування WMI, який постачається разом із WMI. Цей інструмент дозволяє адміністратору або розробнику виконувати більшість завдань за допомогою графічного інтерфейсу, який WMI надає на рівні API. Хоча цей інструмент доступний у всіх операційних системах на базі Windows NT, він офіційно не підтримується Microsoft. WBEMTest надає можливість: перераховувати, відкривати, створювати та видаляти класи та екземпляри класів; обирати простір імен; виконувати запити даних і подій;

виконувати методи, пов’язані з класами або екземплярами; виконувати кожну операцію WMI асинхронно, синхронно або напівасинхронно;

– інструмент командного рядка WMI (WMIC): WMIC – це інструмент командного рядка, призначений для полегшення пошуку інформації WMI про систему за допомогою деяких простих ключових слів (псевдонімів). WMIC.exe доступний у Windows XP Professional, Windows Server 2003, Windows Vista, Windows 7, Windows Server 2008, Windows 11. Введення wmic /?в командному рядку відображає повний список параметрів і ключових слів. (У Windows 11 wmic /?відображається «WMIC застарілий», а потім текст довідки.);

– існує порт Linux для інструменту командного рядка WMI, написаного на Python, на основі Samba4 під назвою «wmi-client»;

– WBEMDump.exe: WBEMDump – це інструмент, який постачається разом із Platform SDK. Цей інструмент командного рядка постачається з власним проектом Visual C++. Інструмент може показувати класи репозиторію CIM, екземпляри або те й інше. Можна отримати ту саму інформацію, що й за допомогою WMIC. WBEMDump.exe вимагає більш конкретних знань про WMI, оскільки він не абстрагує WMI як WMIC. Однак він працює під Windows NT 4.0 і Windows 2000. Також можна виконувати методи, надані класами або екземплярами. Навіть якщо це не стандартний інструмент WMI, який постачається разом із інсталяцією системи, цей інструмент може бути дуже корисним для вивчення репозиторію CIM і функцій WMI;

– WMIDiag.vbs: Інструмент діагностики WMI – це VBScript, який можна завантажити з Microsoft тут, і це інструмент для тестування та перевірки WMI у Windows 2000 і новіших версіях. Завантаження містить досить повну документацію, а інструмент підтримує численні перемикачі. Під час запуску він створить до чотирьох текстових файлів, які: перелічують виконані кроки (файл LOG), огляд результатів (файл REPORT), файл статистики (у форматі значень, розділених комами) і, за бажанням, файл зі списком постачальники, зареєстровані на машині (PROVIDERS, також у форматі значень, розділених комами).

Згенерований файл звіту містить список виявлених проблем і потенційні способи їх вирішення;

– WMI Explorer: WMI Explorer Tool – це безкоштовно доступна програма з відкритим вихідним кодом, яку можна завантажити тут і є інструментом для перерахування та запитів до постачальників WMI у графічному інтерфейсі користувача.

1.7 Протокол ICMP в моніторингу

Протокол керуючих повідомлень Інтернету (ICMP – Internet Control Message Protocol) є допоміжним протоколом у наборі протоколів Інтернету. Він використовується мережевими пристроями, включаючи маршрутизатори, для надсилання повідомлень про помилки та операційної інформації, що вказує на успішне або невдале підключення до іншої IP-адреси, наприклад, помилка вказується, коли запитана служба недоступна або що хост або маршрутизатор не можуть бути досягнутим. ICMP – це протокол мережевого рівня, який використовується маршрутизаторами, проміжними пристроями та хостами для передачі інформації про помилки або оновлень іншим маршрутизаторам, проміжним пристроям і хостам.

ICMP відрізняється від протоколу Інтернету (IP) версії 6 або IPv6 тим, що він не пов’язаний з протоколом керування передачею (TCP) або протоколом дейтаграм користувача (UDP). Як наслідок, немає потреби, щоб пристрій з’єднувався з іншим перед надсиланням повідомлення ICMP.

Наприклад, у протоколі TCP два пристрої, які спілкуються, першими беруть участь у рукостисканні, яке займає кілька кроків. Після завершення "рукостискання" дані можуть бути передані від відправника до одержувача. Цю інформацію можна переглянути за допомогою такого інструменту, як tcpdump.

ICMP відрізняється. З'єднання не утворюється. Повідомлення просто надсилається. Крім того, на відміну від TCP і UDP, які диктують порти, на які надсилається інформація, у повідомленні ICMP немає нічого, що спрямовує її до певного порту на пристрої, який її отримає.

Повідомлення ICMP надсилаються за кількома сценаріями. Наприклад, якщо один пристрій надсилає повідомлення, яке є завеликим для обробки одержувачем, одержувач скине це повідомлення та надішле повідомлення ICMP назад до джерела. Інший приклад: мережевий шлюз знаходить коротший маршрут для переміщення повідомлення. Коли це відбувається, надсилається повідомлення ICMP, і пакет перенаправляється на коротший маршрут.

ICMP відрізняється від транспортних протоколів, таких як TCP і UDP, тим, що він зазвичай не використовується для обміну даними між системами, а також не використовується регулярно мережевими програмами кінцевих користувачів (за винятком деяких діагностичних інструментів, таких як ping і traceroute).

ICMP також використовується для діагностики мережі, зокрема для термінальних утиліт ping і traceroute:

– Traceroute. Утиліта traceroute використовується для відображення фізичного шляху маршрутизації між двома інтернет-пристроями, які спілкуються один з одним. Він планує шлях від одного маршрутизатора до іншого, який іноді називають стрибком. Використання traceroute для діагностики проблем мережі може допомогти адміністраторам знайти джерело затримки мережі;

– Ping. Утиліта ping – це простіше трасування. Він надсилає запити ping, які також називаються повідомленнями ехо-запиту, а потім вимірює час, потрібний повідомленню, щоб досягти пункту призначення та повернутися до джерела. Ці відповіді називаються ехо-відповідями. Пінги корисні для збору інформації про затримку певного пристрою. Однак, на відміну від traceroute, ping не надає графічних карт макета маршруту. Утиліта ping також часто використовується для певних атак на відмову в обслуговуванні (DoS).

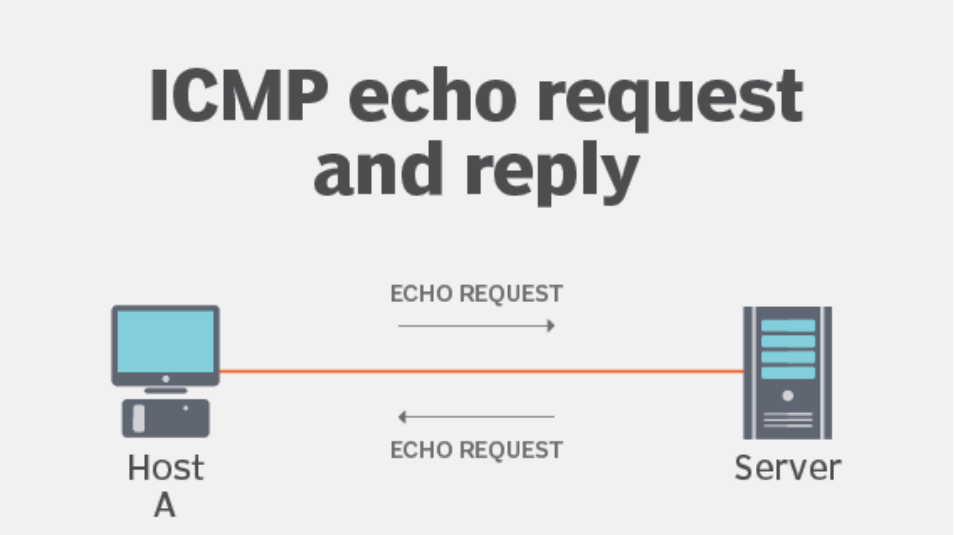


Рисунок 1.2 Запита та відповідь ICMP

Широко використовуваний Інтернет-протокол версії 4 або клас адрес IPv4 і новіший IPv6 використовують аналогічні версії протоколу ICMP – ICMPv4 і ICMPv6 відповідно.

ICMP є одним із основних протоколів пакету IP. Однак ICMP не пов'язаний з жодним протоколом транспортного рівня , таким як протокол керування передачею (TCP) або протокол дейтаграм користувача (UDP). Це протокол без з'єднання, тобто пристрою не потрібно відкривати з'єднання з цільовим пристроєм перед надсиланням повідомлення. Це відрізняється від TCP, наприклад, де з'єднання має бути встановлено перед відправкою повідомлення, встановлюючи, що обидва пристрої готові за допомогою TCP рукостискання.

Повідомлення ICMP передаються як дейтаграми та складаються з IP-заголовка, який інкапсулює дані ICMP. Дейтаграма, так само як і пакет, є автономною незалежною сутністю даних. Подумайте про це як про пакунок, який передає частину більшого повідомлення через мережу. Пакети ICMP – це IP-пакети з ICMP у частині даних IP. Повідомлення ICMP також містять повний IP-заголовок з вихідного повідомлення, тому кінцева система знає, який пакет не вдався.

Заголовок ICMP з'являється після заголовка пакета IPv4 або IPv6 і позначається як номер IP-протоколу 1. Протокол містить три параметри, які пояснюються нижче. Слідом за трьома параметрами є дані ICMP і оригінальний IP-заголовок, що визначає, який пакет не вдався.

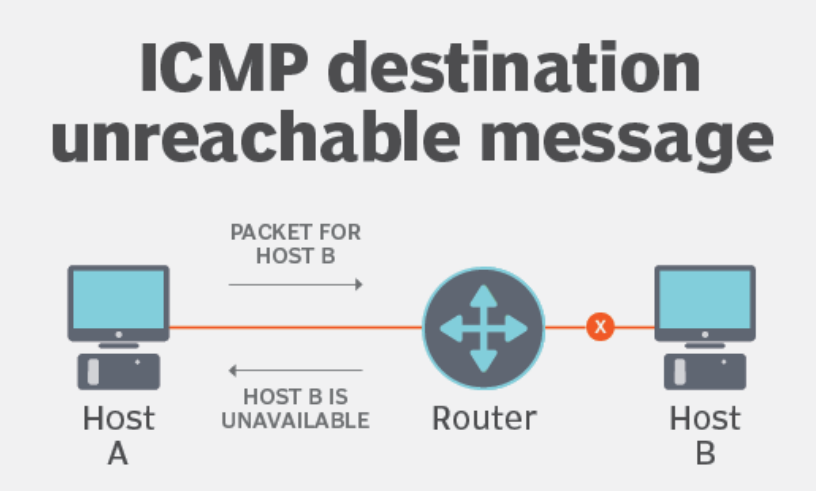


Рисунок 1.3 Принцип роботи протоколу ICMP

1.8 Види IT-моніторингу

Існує велика кількість типів моніторингу, які можна використовувати на кожному рівні IT-ландшафту (рис 1.4). Мережеве і серверне обладнання повинні постійно контролюватися на предмет їх працездатності, особливо коли поломка може призвести до незапланованих простоїв в роботі і, як наслідок, до втрати продуктивності.

Рисунок 1.4 Деякі типи IT-моніторингу

Найбільш розповсюджені види моніторингу IT-інфраструктури:

– Моніторинг інфраструктури – це процес базового рівня, який збирає та переглядає показники, що стосуються апаратного забезпечення IT-середовища та програмного забезпечення низького рівня. Інструменти моніторингу інфраструктури забезпечують еталон для ідеальної роботи фізичних систем, таким чином полегшуючи процес тонкого налаштування та скорочуючи час простою, а також дозволяючи IT-командам виявляти збої та проблеми;

– моніторинг сервера та системи – засоби моніторингу оцінюють продуктивність серверів і компонентів інфраструктури. Сервери контролюються окремо, а сукупні дані аналізуються на продуктивність мережі. Зібрані показники включають час роботи сервера та продуктивність. Апаратний монітор є звичайним компонентом сучасних серверів та персональних комп’ютерів, який може постачатися як окрема мікросхема, часто підключена через I-2-C або SMBus, або як частина рішення Super I/O, яка часто підключається через Low Pin Count (LPC). Ці пристрої дозволяють контролювати температуру в корпусі, напругу, що подається на материнську плату блоком живлення, і швидкість вентиляторів комп'ютера, які підключені безпосередньо до одного з роз'ємів вентиляторів на материнській платі. Багато з цих апаратних моніторів також мають можливості керування вентиляторами. Програмне забезпечення для моніторингу системи, таке як SpeedFan у Windows, lm_sensors у Linux, envstat у NetBSD та sysctl hw.sensors у OpenBSD та DragonFly, може взаємодіяти з цими чіпами, щоб передавати цю інформацію датчиків навколишнього середовища користувачеві;

– хмарний моніторинг – хмарні клієнти можуть переглядати певні показники, такі як використання ЦП, пам’яті ОЗП та сховища, щоб оцінити ефективність їхніх програм, але природа хмарної інфраструктури обмежує перегляд фізичних активів, на яких виконуються хмарні робочі навантаження;

моніторинг мережі – шукає проблеми, спричинені повільними або несправними мережевими компонентами або порушеннями безпеки. Показники включають час відповіді, безвідмовну роботу, помилки запитів статусу та

перевірки HTTP/HTTPS/SMTP. Додатки, що працюють через розподілену IT-інфраструктуру, можуть захоплюватися низкою мережевих вузлів, альтернативних частин додатків і служб. Моніторинг локальної мережі допомагає визначити, що внутрішня мережа організації працює належним чином, а швидкість і продуктивність знаходяться на потрібному рівні. За допомогою інструментів моніторингу IT-інфраструктури можна відстежувати швидкість передачі даних, проводити моніторинг пропускної здатності мережі, отримувати сповіщення про нові доданих мережевих пристроях і т.д. Моніторинг мережі може допомогти організації проактивно зреагувати, якщо неавторизований користувач намагається отримати доступ до мережі або спостерігається аномально висока кількість TCP-з'єднань, а також якщо виявлені проблеми з справністю мережевого обладнання;

– моніторинг безпеки – тип моніторингу зосереджений на виявленні та запобіганні вторгненням, як правило, на рівні мережі. Це включає в себе моніторинг мереж, систем і кінцевих точок на наявність вразливостей, реєстрацію доступу до мережі та визначення шаблонів трафіку для пошуку можливих порушень;

– інтеграція та моніторинг API (Application programming interface) – це набір чітко визначених методів для взаємодії різних компонентів. API надає розробнику засоби для швидкої розробки програмного забезпечення, сучасні програми та служби мають впевнену зовнішню інтеграцію для обробки, можливостей ресурсів та альтернативних цілеспрямованих процесів. Дотримання інтеграції використовується, щоб визначити надання та часові показники інтеграцій сторонніх розробників;

– моніторинг продуктивності додатків (Application performance monitoring (APM)) – APM збирає показники продуктивності програмного забезпечення на основі досвіду кінцевого користувача та споживання обчислювальних ресурсів. Приклади показників, які надає APM, включають середній час відгуку під час пікового навантаження, дані про вузькі місця продуктивності, а також час навантаження та час відгуку. Моніторинг додатків входить до сфери управління

продуктивністю додатків, концепція, яка передбачає більш широкий контроль рівня продуктивності додатків (рис 1.5);

– моніторинг веб-продуктивності – цей тип моніторингу оцінює рухомі компоненти вашої веб-служби, як-от веб-сайти, зокрема служба відповідає на запит користувача на стороні клієнта в мережі. Моніторинг включає вимірювання швидкість завантаження сторінки, помилки передачі даних, помилки завантаження тощо;

– моніторинг бізнес активності – ти моніторингу, зосереджений на вимірюванні та відстеженні бізнес-метрик. Цей тип моніторингу допомагає оцінити показники продуктивності протягом більших періодів часу. Ці інструменти відстежують такі показники, як завантаження програм, веб-продажі та інші показники, наприклад обсяг веб-трафіку.

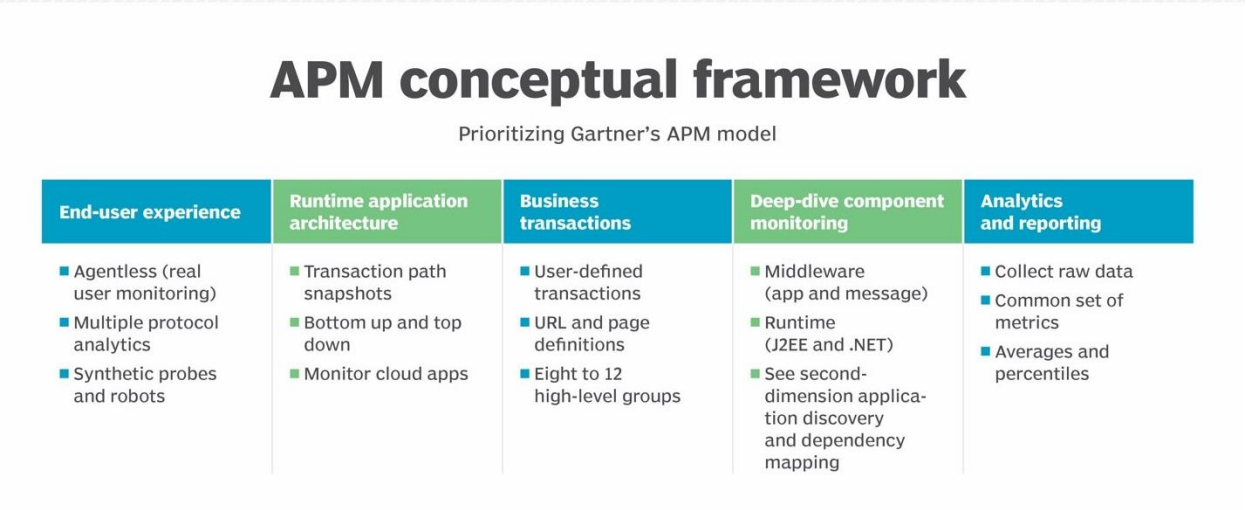


Рисунок 1.5 Концепція моніторингу продуктивності додатків

Інші інструменти, що використовуються в IT-моніторингу, можуть включати інструменти спостереження, інструменти аналізу та інструменти залучення.

– інструменти спостереження є основним типом інструментів, які відстежують ефективність роботи програмного забезпечення;

– інструменти аналізу беруть дані спостережень і додатково аналізують їх, щоб визначити, де і чому виникають проблеми з тими чи іншими компонентами;

– інструменти залучення, зосереджені на дії, на основі даних із інструментів спостереження та аналізу, щоб виконувати такі дії, як створення сповіщень або запуск іншого апаратного чи програмного забезпечення.

До прикладів сервісів, які забезпечують моніторинг різного типу, можна віднести:

– Інструменти APM: Cisco AppDynamics, BMC TrueSight, Microsoft Azure Application Insights, Datadog, Dynatrace, ManageEngine Applications Manager тощо;

– інструменти IT-інфраструктури: Microsoft System Center Operations Manager (SCOM), LogicMonitor, VMware vRealize Operations, ManageEngine OpManager тощо;

– хмарні інструменти моніторингу: Amazon CloudWatch, Microsoft Azure Monitor, Cisco CloudCenter, Oracle Application Performance Monitoring Cloud Service тощо;

– контейнери/мікросервіси/інструменти моніторингу розподілених програм: Jaeger, Kafka, Lightstep тощо;

– інструменти AIops: BigPanda, Moogsoft, New Relic, Datadog, Dynatrace тощо;

– інструменти моніторингу журналів: Fluent , Flexi stack, sumo та sumo logic тощо;

– інструменти моніторингу безпеки мережі: Cisco DNA Analytics and Assurance, LogRhythm, LiveAction LiveNX тощо.

Моніторинг інфраструктури проти управління інфраструктурою

Хоча фрази звучать однаково, існують відмінності в моніторингу інфраструктури та управлінні інфраструктурою.

– Моніторинг інфраструктури передбачає збір і перегляд даних, пов’язаних з різними компонентами обчислювальної системи підприємства. Моніторинг інфраструктури ідентифікує проблему та повідомляє про неї.

– Тоді керівництво інфраструктури отримує це повідомлення та оцінює вплив проблеми та оцінює, як пом’якшити або покращити проблему.

Ефективний моніторинг інфраструктури є ключовим компонентом оптимального управління інфраструктурою, і обидва важливі для продуктивності та прибутку функціонального бізнесу.

2 ОСНОВНІ СИСТЕМИ МОНІТОРИНГУ, ХАРАКТЕРИСТИКИ

2.1 Система моніторингу Nagios XI

Nagios XI – комплексне програмне забезпечення для моніторингу корпоративних серверів і мереж. Бізнес-версія Nagios XI, була створена на основі версії з відкритим кодом і має більше функціональних можливостей, що онзанача вимагає менше часу на адміністрування. Nagios зосереджується в основному на показниках сервера, продуктивності додатків і мережевому трафіку. Він збирає дані за допомогою агентів, розміщених як на елементах мережі, так і на компонентах, які він контролює. (рис. 2.1)

Nagios також може підключатися до мережевих комутаторів або інших компонентів, запитуючи їхній статус за допомогою простого протоколу керування мережею (SNMP). Він зв’язується з продуктами на базі Windows і збирає дані з них за допомогою протоколу Windows Management Instrumentation (WMI).

Nagios доступний як комплект для завантаження з окремими пакетами для кожного продукту, що працює на Windows або Linux. Після завантаження та інсталяції інструменту, потрібно виконати низку початкових налаштувань. Після розгортання агентів дані повинні почати надходити в Nagios і його стандартні інформаційні панелі.

Рішення легко настроюється та масштабується, що робить його ідеальним для багатьох підприємств. Однак ця висока можливість налаштування супроводжується додатковою складністю та витратами на обслуговування.

До плюсів, можна віднести:

- підтримка мережевих компонентів, таких як маршрутизатори, комутатори та інше фізичне обладнання;
- можливість налаштування; підтримує спеціальні показники;
- підтримує моніторинг серверів Windows і Linux.

До мінусів:

– обмежений набір інформаційних панелей за замовчуванням

– складний інтерфейс користувача; конфігурація не дуже зручна для користувача

– накладні витрати на технічне обслуговування та експлуатацію

Nagios XI безкоштовний для невеликих середовищ, але після семи вузлів моніторингу, потрібна річна ліцензія на підтримку та обслуговування сервера Nagios.

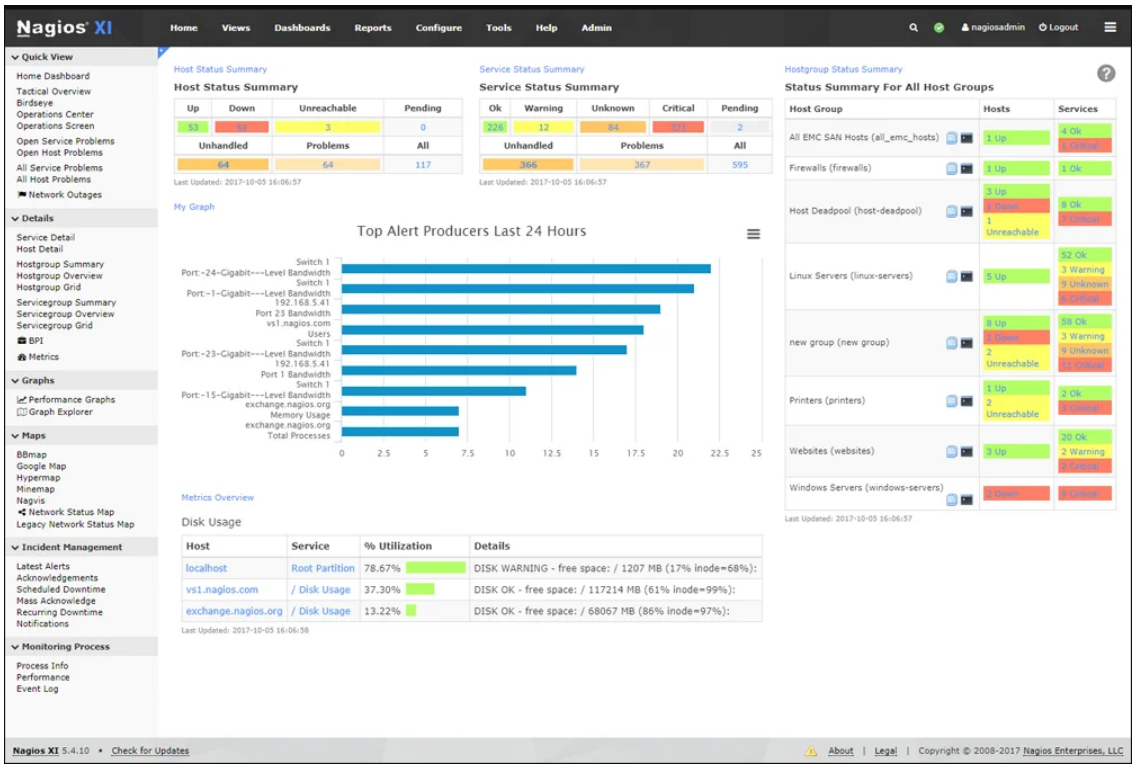


Рисунок 2.1 Зовнішній вигляд системи моніторингу Nagios

2.2 Система моніторингу Checkmk

Checkmk – побудована на базі Nagios, але сильно допрацьована. Система проста, зручна та функціональна, а поріг входу дуже низький. Розібратися зможе майже будь-хто. Моніторинг виконує на основі агентів та з snmp. Відмінно працює

автовиявлення. Достатньо під’єднати хост з агентом і Checkmk сам підбере шаблон, виявить усі відомі йому метрики та почне моніторинг. (рис. 2.2)

Має зручний та красивий веб-інтерфейс. Після додавання змін необхідно їх підтвердити, до цього вони не застосовуються.

За допомогою checkmk можна контролювати гібридну IT-інфраструктуру. Це включає сервери, мережі, програми, бази даних, хмару, контейнери, сховище та IoT. Checkmk забезпечує понад 1800 інтеграцій у бази даних, хмарні сервіси, операційні системи чи апаратне забезпечення.

Checkmk можна розгорнути за лічені хвилини з єдиного інтегрованого пакета, доступного для багатьох платформ, а також як докер-контейнер. Checkmk використовує автоматичне виявлення, щоб допомогти вибрати найбільш відповідні показники для проекту. Користувач може контролювати свою інфраструктуру за допомогою потужного моніторингу на основі агентів, а також моніторингу без агентів через HTTP/SNMP або через пряме з’єднання API.

Checkmk не пропонує стандартні моделі підписки, і дає можливість розрахувати власне рішення на їхній сторінці цін. Ціна залежатиме від вибраної версії та кількості послуг. Налаштування за умовчанням починається з 600 доларів США на місяць без урахування податків.

Основні переваги checkmk, схожі з Nagios, але трошки доповнені:

- журнал і моніторинг подій;
- моніторинг інфраструктури;
- динамічні інформаційні панелі.

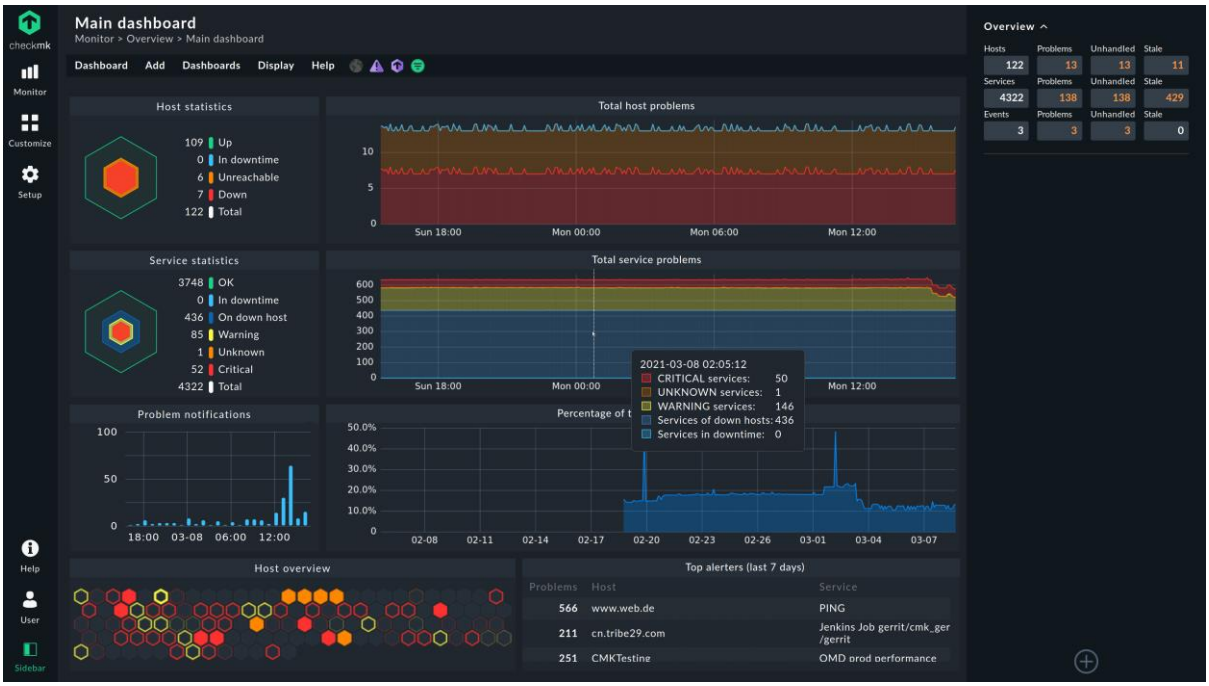


Рисунок 2.2 Зовнішній вигляд системи моніторингу Checkmk

2.3 Система моніторингу Icinga

Icinga – спочатку це був форк Nagios. Але з часом їх шляхи сильно розійшлися, тож можна вважати Icinga самостійним продуктом. Причому якісним та повністю безплатним. У системі є все, що треба для повноцінного моніторингу. Дані може збирати як за допомогою агентів, так і без них. Бекенд написано на C++, вебінтерфейс на php. Як БД підтримує MySQL, Oracle Database, PostgreSQL. Ефективний механізм моніторингу Icinga здатний контролювати всю інфраструктуру, включаючи всі центри обробки даних і хмарні хости. Після моніторингу він збирає всі результати в окремому ресурсі для подальшої оцінки. (рис. 2.3)

До собливостей можна віднести:

– веб-інтерфейс із настроюваними видами, групуванням, фільтруванням, окремим елементом, настроюваною інформаційною панеллю та інтуїтивно зрозумілим інтерфейсом;

– безпечний і захищений із SSL і обмеженнями для користувачів, сповіщеннями через сповіщення та керуванням інцидентами;

– переконлива мова конфігурації, швидка синхронізація, веб-конфігурація та автоматизація за допомогою інструментів;

– розгортання за допомогою REST API, інструментів DevOps, автоматизованих інтеграцій, розподіленого моніторингу та моніторингу на основі агентів;

– теги екземплярів, схема graphite, graphite writer, метрика, еластичний пошуковий writer та інтеграція Graylog.

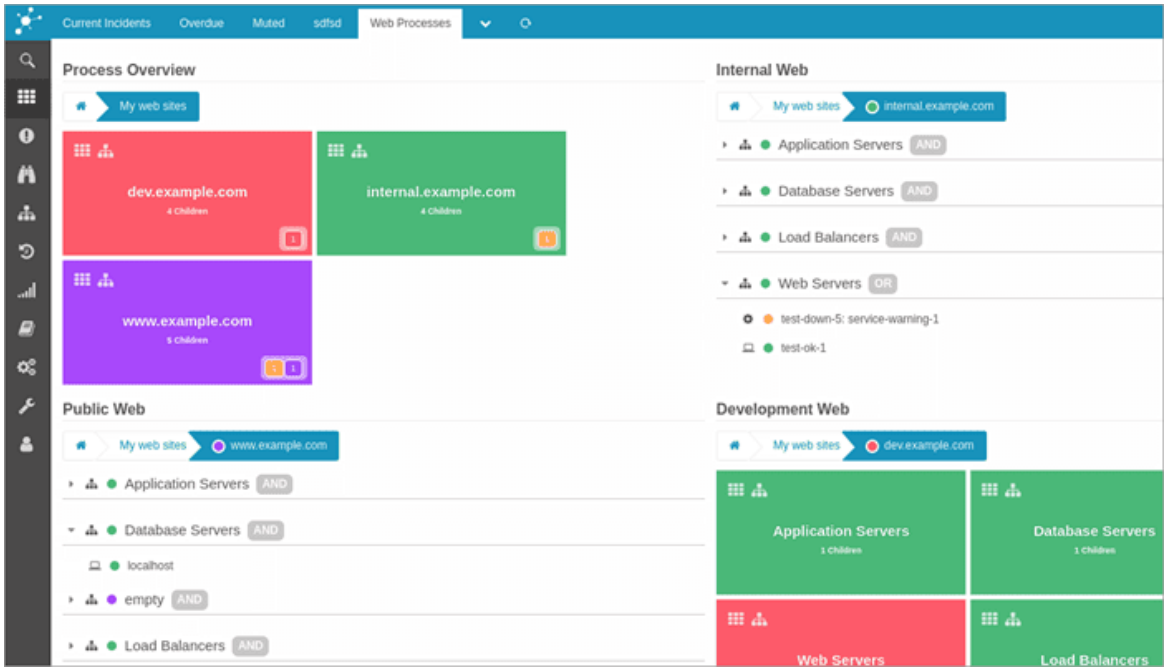


Рисунок 2.3 Зовнішній вигляд системи моніторингу Icinga

2.4 Система моніторингу Zabbix

Zabbix – це універсальний і популярний інструмент моніторингу системи з відкритим вихідним кодом, який збирає, агрегує та аналізує показники з мереж, серверів/віртуальних машин, хмар, програм, служб, баз даних тощо. Його великі можливості візуалізації дають уявлення про стан системи, а готові або налаштовані користувачем шаблони автоматизують виявлення компонентів, які потрібно контролювати. (рис. 2.4)

Zabbix – вільна система моніторингу статусів різноманітних сервісів комп'ютерної мережі, серверів та мережевого обладнання. Для зберігання даних використовують MySQL, PostgreSQL, SQLite або Oracle Database, вебінтерфейс написаний на PHP. Підтримує кілька видів моніторингу.

Simple checks – може перевіряти доступність і реакцію стандартних сервісів, таких як SMTP або HTTP, без встановлення будь-якого програмного забезпечення на хості, що спостерігається.

Zabbix agent – може бути встановлений на UNIX-подібних або Windows-хостах для отримання даних про навантаження процесора, використання мережі, дисковому просторі тощо.

External check – виконання зовнішніх програм, також підтримується моніторинг через SNMP.

До плюсів можна віднести:

– проста інтеграція на основі API з існуючими програмами;

– автоматично виявляє аномалії та прогнозує тенденції за допомогою розумних гнучких порогів;

– класифікує виявлені проблеми для ефективнішого сповіщення та прискореного аналізу першопричини;

– видимість на одній панелі з настроюваними інформаційними панелями, графіками та звітами.

До мінусів:

– комплексне розгортання та налаштування;

– немає розміщеного рішення.

Zabbix це монітонинг з відкритим кодом і є безкоштовним, але якщо є необхідність, є можливість доповнити його платними послугами, такими як технічна підтримка, консультації та підтримка оновлення/створення шаблонів.

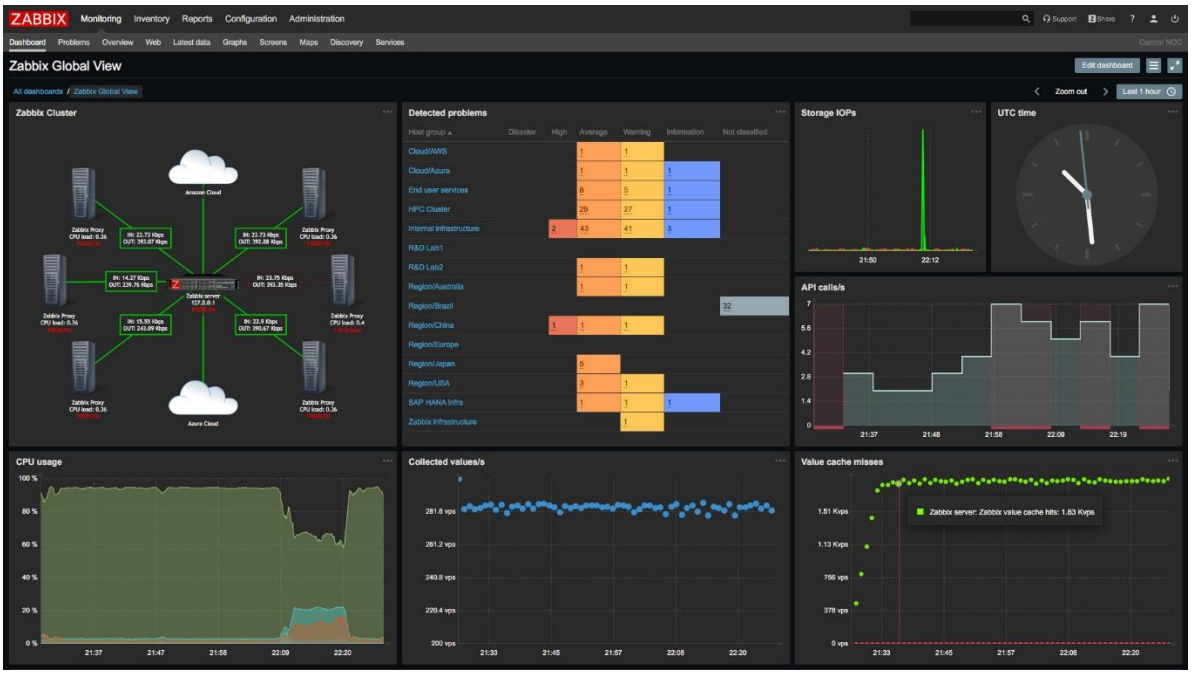


Рисунок 2.4 Зовнішній вигляд системи моніторингу Zabbix

2.5 Система моніторингу AppDynamics

AppDynamics – надає комплексне рішення для моніторингу інфраструктури, яке охоплює компоненти сервера, сховища та мережі як у хмарних, так і в гібридних середовищах. Ви можете розгорнути його локально або використовувати як службу SaaS. (рис. 2.5)

Можливості повного стека моніторингу цього інструменту допомагають співвідносити проблеми продуктивності програми з вузькими місцями інфраструктури низького рівня, прискорюючи тим самим аналіз основних причин і усунення.

Завдяки повному набору інформаційних панелей і показників AppDynamics підтримує детальні сповіщення, які можна інтегрувати зі сторонніми інструментами оповіщення та керування інцидентами, такими як ServiceNow, PagerDuty та Jira.

Якщо компанія більше орієнтована на бізнес, AppDynamics є хорошим рішенням. Більшість зацікавлених сторін завжди шукають економічну цінність у

кожній інвестиції в цифрову інфраструктуру, яку вони роблять. Менталітет AppDynamics, який орієнтований на бізнес на першому місці, дозволяє зацікавленим сторонам побачити, як їхні інвестиції в хмарну інфраструктуру впливають на ключові показники ефективності бізнесу, як-от дохід, із корельованими показниками продуктивності програм і серверів.

До плюсів можна віднести:

– кореляція показників продуктивності програми з показниками продуктивності сервера та мережі;

– виявлення аномалій і сповіщення;

– перша в бізнесі платформа спостереження з рекомендаціями щодо планування потужностей.

До мінусів:

– для розширених функцій необхідне навчання;

– неадекватні підручники та документація.

AppDynamics стягує плату за ядро ЦП. Доступна 15-денна безкоштовна пробна версія.

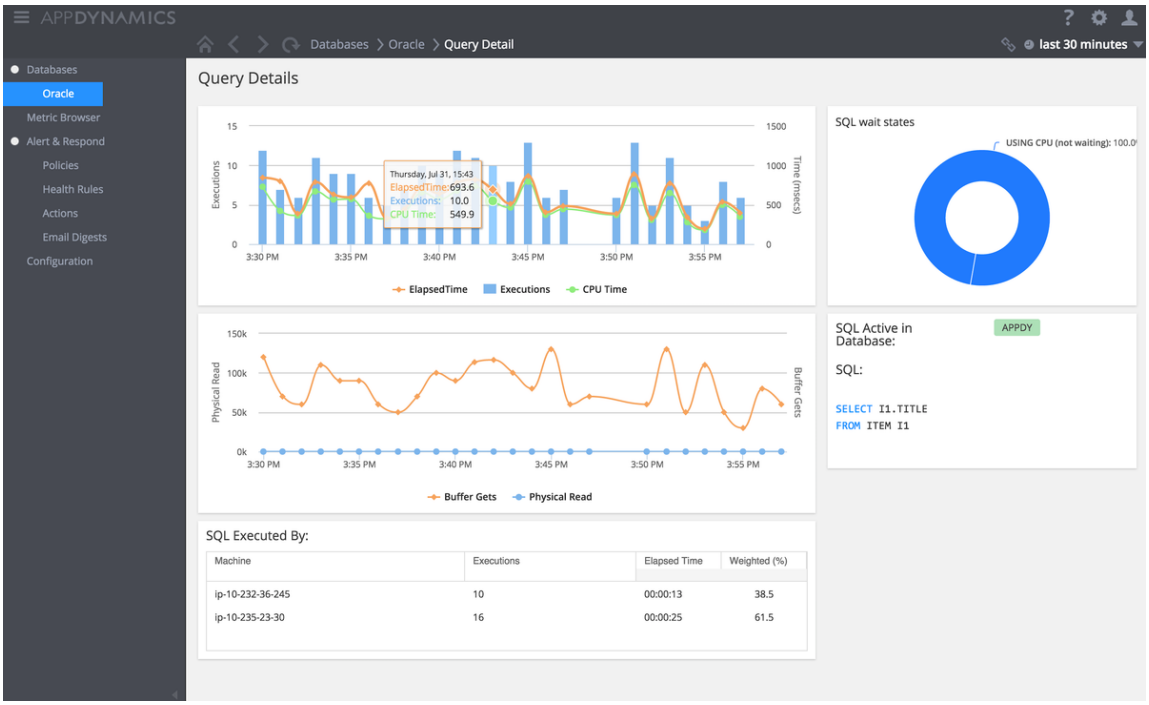


Рисунок 2.5 Зовнішній вигляд системи моніторингу AppDynamics

2.6 Система моніторингу The Elastic Stack

Рішення для моніторингу Elastic Stack (ELK Stack) поєднує в собі можливості трьох проектів з відкритим кодом: Elasticsearch, Logstash та Kibana. Elasticsearch відповідає за пошук і аналітику, тоді як Logstash допомагає вводити та перетворювати дані з різних джерел перед надсиланням їх до Elasticsearch. Kibana дозволяє візуалізувати за допомогою діаграм і графіків на основі даних, проаналізованих Elasticsearch. Ці можливості можна використовувати для показників, зібраних із багатьох джерел у вашій інфраструктурі, і для надання уявлень про стан вашого середовища. (рис. 2.6)

Інтеграція для моніторингу інфраструктури доступна через модуль Metricbeat, який співвідносить показники з різних джерел, як-от сервери, контейнери Docker, Kubernetes , та багато іншого. Модуль створює шаблони індексів у Kibana, які допомагають візуалізувати стан інфраструктури. Ви також можете налаштувати сповіщення для порогових значень на основі індексів/метрик і надсилати сповіщення електронною поштою, Microsoft Teams, Slack або іншими інтеграціями сторонніх розробників.

До плюсів можна віднести:

– можливість розміщення ELK локально або використання розміщеного рішення;

– можливість перегляду використання процесора/пам’яті та статистики на рівні процесу на інформаційній панелі Kibana;

– налаштування, аналіз і візуалізація даних у режимі реального часу для отримання глибокої інформації;

– аналізує телеметричні дані з розподілених інфраструктур у реальному часі;

– бібліотеки для кількох мов сценаріїв і програмування.

До мінусів:

– складне та багатоетапне розгортання;

– складна конфігурація інфраструктури, необхідна для забезпечення стійкості, високої доступності та зручності використання даних.

ELK має відкритий вихідний код і його можна безкоштовно завантажити та використовувати. Однак вам доведеться платити за підтримку інфраструктури (тобто обчислень), зберігання та пропускної здатності мережі, необхідних для роботи компонентів ELK, що може бути дорогим.

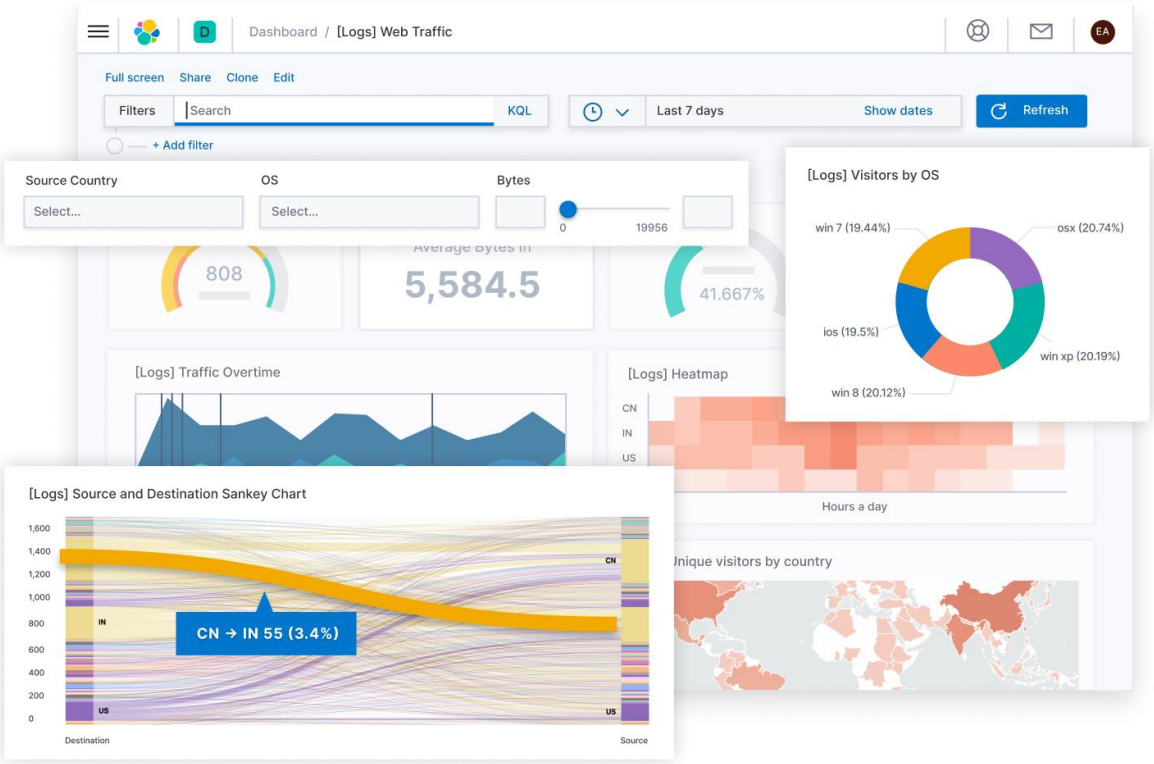


Рисунок 2.6 Зовнішній вигляд системи моніторингу The Elastic Stack

2.7 Система моніторингу LibreNMS

LibreNMS — форк Observium з ширшим функціоналом. Можна запускати в Docker, є інтеграція з Grafana, є можливість зберігати дані в InfluxDB. Система заточена на моніторинг мереж з SNMP. Має підтримку практично всіх популярних мережевих пристроїв, окрім цього, вміє моніторити Windows, Linux, FreeBSD і

тому використовує агенти. Працює на базі PHP+MySQL. Усі типові метрики підхоплює сама. (рис. 2.7)

До особливостей можна віднести:

– автоматичне виявлення - може виявити архітектуру всієї мережі за допомогою CDP, FDP, LLDP, OSPF, BGP, SNMP і ARP;

– настроювані сповіщення - надзвичайно гнучка система оповіщення, сповіщення електронною поштою, irc, slack, телеграм, MS Teams тощо;

– API доступу - повноцінний API для керування, створення графіків і отримання даних;

– білінгова система - можливість створення рахунки за пропускну здатність для портів у мережі на основі використання або передачі;

– автоматичне оновлення - функція автоматичного оновлення, завдяки виправленню помилок, новим функціям тощо, постійна актуальність системи;

– додаток для iPhone та Android - доступна рідна програма для iPhone та Android, яка забезпечує основні функції.

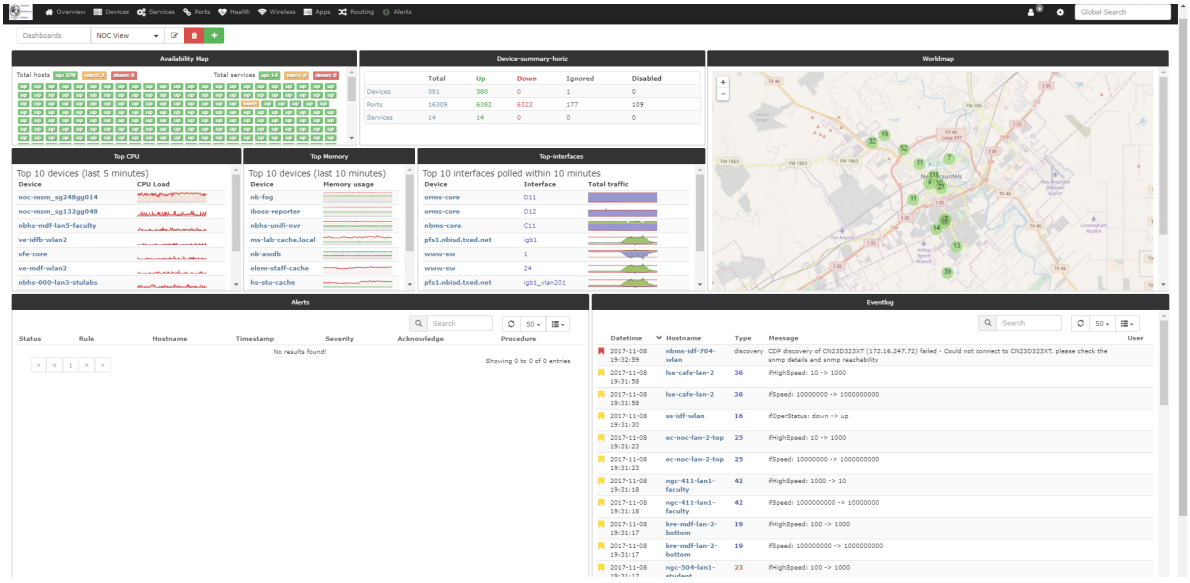


Рисунок 2.7 Зовнішній вигляд системи моніторингу LibreNMS

2.8 Система моніторингу New Relic

New Relic – моніторинг, що працює за моделлю SaaS. Вам достатньо встановити агента на сервер та вказати свій унікальний ключ. Далі агент все зробить сам, а вам залишиться тільки зайти в вебінтерфейс і дивитися метрики. Функціонал моніторингу дуже крутий, користуватися зручно. Безплатний тарифний план – 100 GB даних на місяць. (рис. 2.8)

New Relic дає вам повне уявлення про всю вашу інфраструктуру в одному місці. Один із нових інструментів моніторингу, представлений із більшою увагою до моніторингу контейнерів і серверів, New Relic об’єднує всі ваші корельовані показники в одну площину даних, що дає вам змогу виконувати рядок запиту лише кількома клацаннями миші та точно бачити, як усе пов’язано.

Він має відкриту та гнучку інтеграційну структуру, яка дозволяє інтегрувати та підтримувати різні системи, зокрема Prometheus, Kubernetes , AWS, Azure, GCP, MySQL, NGINX, Apache Kafka, Apache Cassandra та багато інших.

Налаштування програмного забезпечення моніторингу серверів New Relic займає менше п’яти хвилин, і ви можете отримати миттєве уявлення про свою систему. Налаштування є досить простим завдяки його готовій інтеграції з іншими системами та простим у використанні SDK. Якщо більшість ваших робочих навантажень пов’язані з Kubernetes і контейнерами, New Relic стане чудовим вибором для моніторингу вашої інфраструктури.

До плюсів можна віднести:

- підтримка корельованих показників;
- проактивне виявлення аномалій і сповіщення;
- доступні інтеграції для провідних хмарних провайдерів; підтримка відкритих стандартів.

До мінусів:

- немає самостійного рішення;

– крута крива навчання;

– менше контролю над керуванням сповіщеннями;

– модель ціноутворення, яка стягує плату за користувачами та за даними.

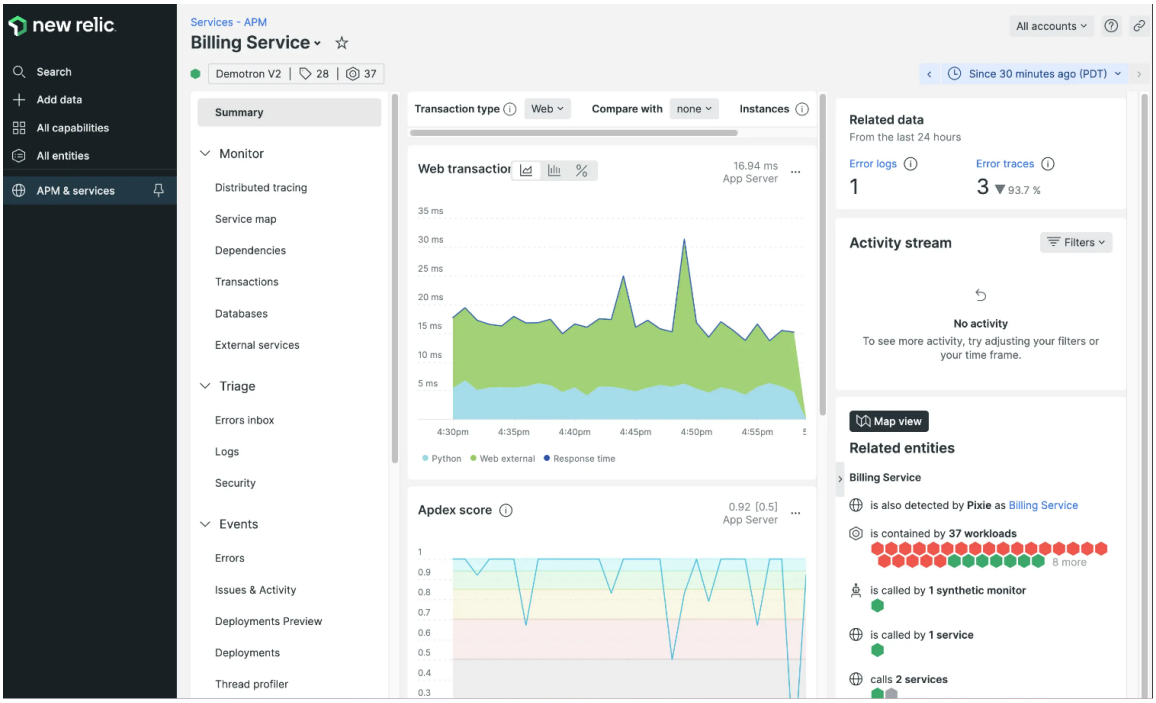


Рисунок 2.8 Зовнішній вигляд системи моніторингу New Relic

2.9 Система моніторингу Sematext Monitoring

Sematext Monitoring – повноцінне ПЗ для моніторингу IT-інфраструктури, яке забезпечує видимість локальних і хмарних розгортань в реальному часі. Він також дозволяє вам бачити стан справності вашої інфраструктури, відстежуючи програми, сервери, контейнери, процеси, інвентар, події, бази даних тощо. (рис. 2.9)

Ви можете використовувати його для моніторингу інфраструктури контейнерів, щоб отримати доступ до контейнерних додатків, що працюють у Docker або платформах оркестровки, таких як Kubernetes, Docker Swarm і Nomad. Sematext Monitoring може здійснювати автоматичне виявлення. Sematext Agent

спостерігає за вашими середовищами для сервісів, які можна підключити до самого інструменту, що полегшує процес адаптації.

Цей інструмент пропонує комплексне виявлення аномалій та інтеграцію із зовнішніми службами сповіщень для попередження інфраструктури, такими як PagerDuty, Opsgenie, Splunk On-Call (раніше VictorOps) тощо.

Sematext – ідеальне рішення для моніторингу системи. Лише за кілька хвилин налаштування, і на додаток до всього, що було зазначено вище, користувач отримує:

– автоматизоване виявлення служб і журналів; швидке підключення

– співвідносить журнали та події з консолідованою однопанельною системою, яка відображає статус і працездатність

– відстежує неправильну конфігурацію сервера та застарілі пакети

– моніторинг процесу для виявлення вузьких місць продуктивності

Sematext пропонує 14-денну безкоштовну пробну версію. Є три рівні цін: базовий (безкоштовний моніторинг інфраструктури до трьох хостів), стандартний (3,6 дол. США за хост/місяць) і професійний (5,76 дол. США за хост/місяць).

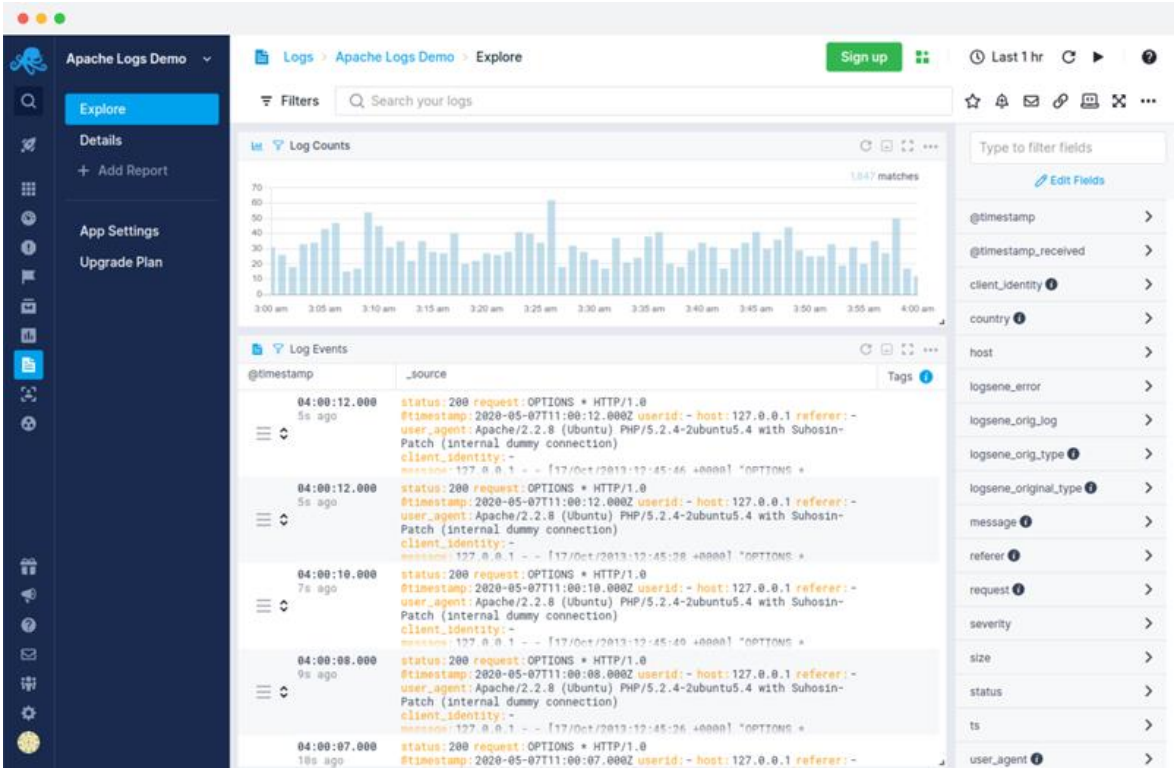


Рисунок 2.9 Зовнішній вигляд системи моніторингу Sematext Monitoring

2.10 Система моніторингу SolarWinds Server & Application Monitor (SAM)

SolarWinds Server & Application Monitor (SAM) – забезпечує поглиблений моніторинг вашої IT-інфраструктури, як локально, так і в хмарі. Він пропонує готову підтримку для понад 1 200 програм і систем, а також кілька інших шаблонів для інтеграції, створених спільнотою. (рис. 2.10)

Інструмент дозволяє відстежувати компоненти інфраструктури за допомогою WMI, SNMP, Powershell, REST API тощо. SAM має попередньо визначені конфігурації моніторингу ОС для Windows і Linux, що забезпечує швидшу адаптацію та моніторинг продуктивності.

Рішення є фантастичним кандидатом для моніторингу не лише серверів, а й усієї інфраструктури. Можливості включають автоматичний моніторинг служб сервера, моніторинг віддаленого сервера, моніторинг працездатності сервера, моніторинг серверних програм, моніторинг інвентаризації сервера та моніторинг процесів сервера. Користувач отримує покриття для контейнерів, баз даних і програм із сповіщеннями та понад 1200 інформаційних панелей із коробки.

До плюсів можна віднести:

- підтримка наскрізного моніторингу з корельованими показниками;
- автоматичне виявлення служб і відображення залежностей програм;
- підтримка та рекомендації щодо планування потужності сервера.

До мінусів:

- немає виявлення аномалій для сповіщень;
- немає підтримки федерації ідентифікації з LDAP;
- загальні та обмежені фільтри звітів на інформаційній панелі.

SolarWinds Server & Application Monitor стягує оплату за хост за місяць. Існує також 30-денна безкоштовна пробна версія.

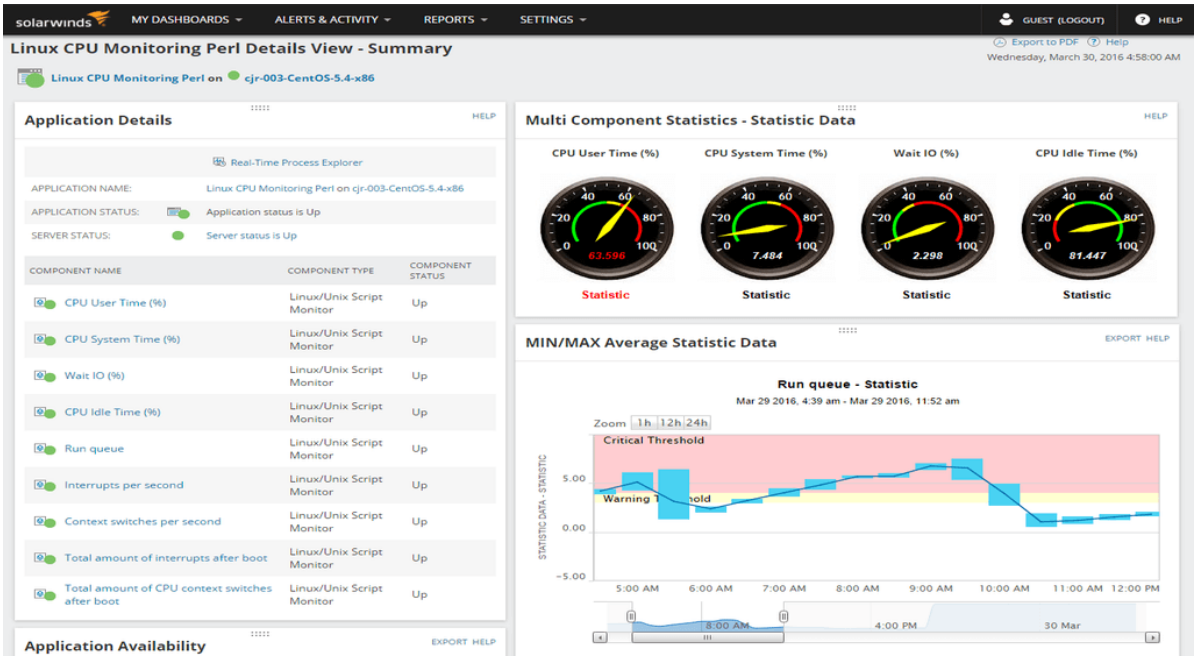


Рисунок 2.10 Зовнішній вигляд системи моніторингу SolarWinds Server & Application Monitor (SAM)

2.11 Система моніторингу Datadog

Datadog – це серверна система моніторингу інфраструктури, програм, мережі та log-журналів. Однією з видатних особливостей Datadog є те, що він забезпечує уніфіковане уявлення про моніторинг із корельованими показниками, пов’язаними з виявленням журналів серверів і трасуванням. Це зручна функція, коли ви усуваєте проблеми з продуктивністю на своїх серверах. Наприклад, відстежуючи метрики сервера разом із даними додатків, можна виявити приховані джерела затримки, такі як перевантажені хости або суперечливі бази даних.Агент із відкритим кодом Datadog підтримує понад 450 інтеграцій, зокрема Kubernetes, Docker та Apache Kafka. (рис. 2.11)

Datadog охоплює всі відповідні метрики та журнали безпеки та моніторингу системи, і ви можете використовувати Datadog API для інтеграції з рештою свого середовища. Ви також можете шукати, фільтрувати, розділяти та аналізувати

журнали та інші показники, щоб отримати реальну картину працездатності системи, а також швидко усувати виявлені проблеми.

До плюсів можна віднести:

– автоматично виявляє аномалії та генерує інтелектуальні сповіщення;

– уніфікований досвід моніторингу в хмарних і локальних середовищах;

– візуалізує топографію системи;

– консолідовані та настроювані інформаційні панелі для відображення ключових відомостей про працездатність системи, включаючи інтерфейс веб-сайту, код програми та базову інфраструктуру.

До мінусів:

– складне налаштування;

– досить складне навчання

– обмежена аналітика журналу через відсутність підтримки обробки журналу JSON.

Є можливість спробувати Datadog за допомогою безкоштовної 14-денної пробної версії. Існує три рівні цін: безкоштовний (п’ять хостів із збереженням показників протягом 1 дня), Pro (15 доларів США за хост/місяць) і Enterprise (23 доларів США за хост/місяць).

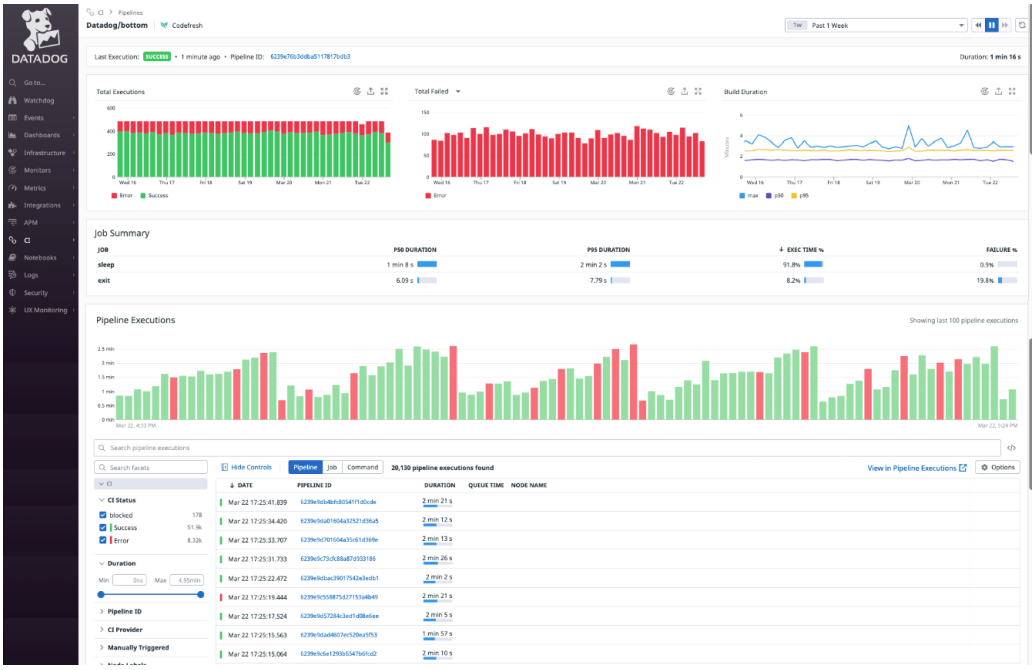


Рисунок 2.11 Зовнішній вигляд системи моніторингу Datadog

2.12 Система моніторингу Prometheus і Grafana

Prometheus і Grafana є двома найвідомішими інструментами моніторингу з відкритим вихідним кодом, доступними наразі, дуже популярний для потреб моніторингу серверів. Для реалізації, потрібно буде встановити низку агентів збору даних, відомих як експортери, щоб надсилати показники в Prometheus, а Grafana дозволяє створювати приголомшливі інформаційні панелі, використовуючи ці показники. (рис. 2.12)

Налаштування повноцінної системи моніторингу сервера за допомогою Prometheus і Grafana потребує значної конфігурації та є складною, оскільки це рішення DIY. Однією з ключових переваг є те, що Prometheus працюватиме на вашій інфраструктурі, тобто ви не будете передавати свої показники сторонньому постачальнику.

Сповіщення про моніторинг сервера підтримуються як Grafana, так і Prometheus із інтеграцією кількох каналів, включаючи Slack, PagerDuty, Microsoft Teams та деякі інші.

Prometheus і Grafana – це потужна комбінація з відкритим вихідним кодом, яка забезпечує значну гнучкість завдяки серверній частині, яка забезпечує чудовий моніторинг продуктивності сервера.

До плюсів можна віднести:

– безкоштовний і відкритий вихідний код із величезною спільнотою відкритих вихідних кодів для підтримки;

– автоматичне виявлення служб і підтримка моделей метричного сканування як push, так і pull;

– підтримка власних показників; величезна кількість експортерів, доступних для експорту метрик до Prometheus з різних джерел.

До мінусів:

– складне та трудомістке керування екземплярами Prometheus; операційні витрати, якщо ваш персонал не знайомий з інструментом;

– потрібно вручну налаштувати експортери Prometheus і керувати ними;

– потрібне ручне налаштування для графіків і сповіщень.

Prometheus і Grafana – безкоштовні інструменти моніторингу серверів із відкритим кодом.

Рисунок 2.12 Зовнішній вигляд системи моніторингу Grafana

2.13 Система моніторингу Dynatrace

Dynatrace – це повноцінний інструмент моніторингу продуктивності сервера, доступний як у моделях програмного забезпечення як послуги (SaaS), так і в локальних моделях. Завдяки можливості відстежувати показники сервера, а також журнали сервера, Dynatrace має задовольнити більшість ваших потреб у моніторингу. (рис. 2.13)

Налаштувати Dynatrace відносно легко, оскільки потрібно лише кілька хвилин, щоб передати свої показники на красиві інформаційні панелі, щоб отримати уявлення про ЦП, пам’ять і стан мережі серверів аж до рівня процесу.

Одна з чудових особливостей Dynatrace полягає в тому, що він може показувати мережеві показники для певного процесу. Він не лише відстежує сервери, але й використовує штучний інтелект для автоматичного розуміння особливостей усієї архітектури вашої програми, включаючи оцінку доступності та проблем з продуктивністю.

З мінімальними операційними витратами та складністю Dynatrace є чудовим рішенням для моніторингу інфраструктури організації, розміщеної в кількох хмарах.

До плюсів можна віднести:

– доступна локальна версія;

– універсальна платформа з підтримкою інфраструктури, продуктивності додатків, бізнес-аналітики та хмарної автоматизації;

– виявлення аномалій і сповіщення за допомогою ШІ.

До мінусів:

– деякі обмежені функції інформаційної панелі;

– комплексний у використанні, вимагає додаткового навчання;

– відставання документації для останніх випущених функцій.

Dynatrace стягує плату залежно від кількості даних, отриманих щомісяця. Доступна 15-денна безкоштовна пробна версія.

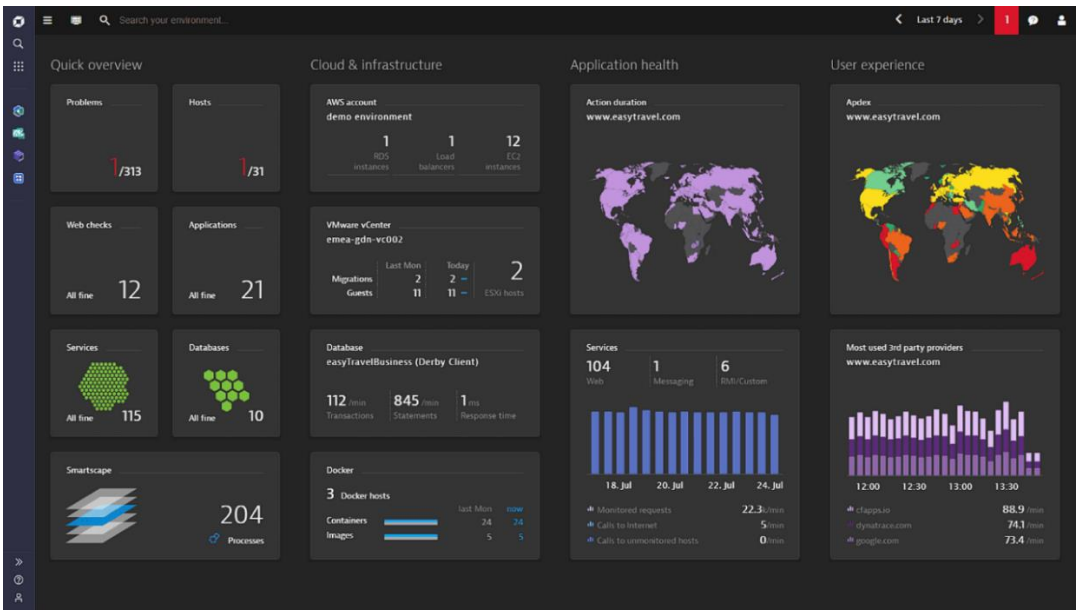


Рисунок 2.13 Зовнішній вигляд системи моніторингу Dynatrace

2.14 Система моніторингу ManageEngine OpManager

ManageEngine OpManager – це просте та економічне програмне забезпечення для моніторингу продуктивності сервера, у якому більше уваги приділяється мережі. Він включає складні можливості моніторингу сервера та мережі, такі як візуалізація потоку мережевого трафіку та наскрізний моніторинг мережі сервера. Одна цікава функція полягає в тому, що якщо ваша організація використовує VoIP, ManageEngine OpManager може просто відстежувати та звітувати про продуктивність VoIP на ваших серверах, надаючи інформацію про те, як її покращити. (рис. 2.14)

Ви також отримуєте підтримку моніторингу будь-якого фізичного пристрою з підключенням до мережі та IP-адресою, наприклад серверів, комутаторів, маршрутизаторів, балансувальників навантаження, брандмауерів, принтерів і пристроїв зберігання.

ManageEngine OpManager постійно стежить за вашою мережею та надає повну інформацію про неї та контроль над нею. Якщо ваша організація зосереджена на моніторингу мережі, особливо якщо вона базується на телекомунікаційних компаніях, ManageEngine OpManager чудово підійде.

До плюсів можна віднести:

- наскрізний моніторинг мережі;
- моніторинг фізичних пристроїв, наприклад маршрутизаторів і комутаторів;
- відстежує та усуває проблеми з продуктивністю VoIP.

До мінусів:

- немає хмарної версії SaaS;
- моніторинг продуктивності програми та мережева кореляція недоступні;
- операційні витрати через керування постійними оновленнями виправлень.

Ціна ManageEngine OpManager визначається потребами кожного клієнта. Існує безкоштовна версія, яка дозволяє контролювати до трьох пристроїв.

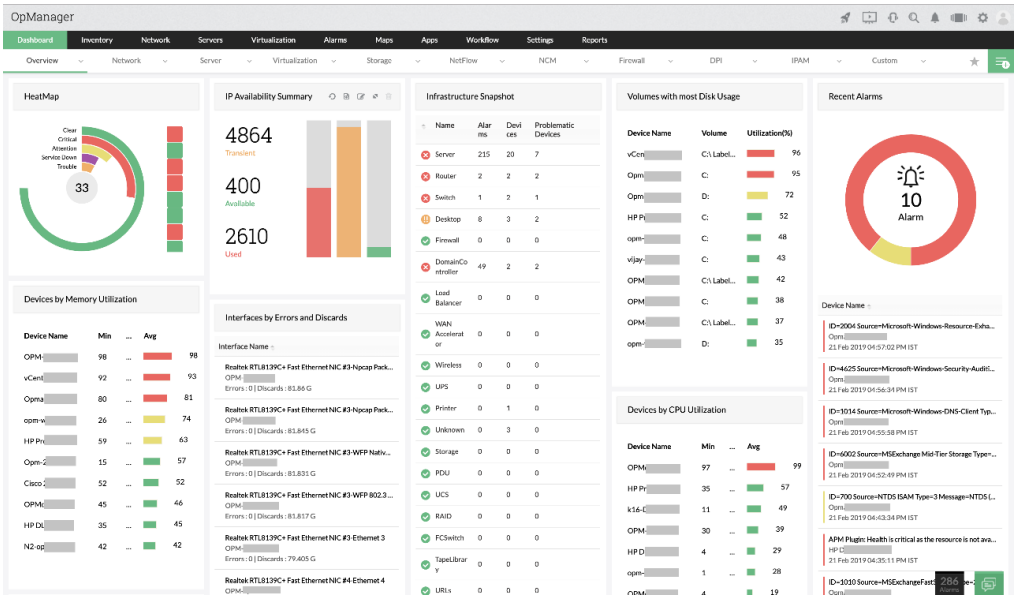


Рисунок 2.14 Зовнішній вигляд системи моніторингу ManageEngine OpManager

2.15 Система моніторингу Better Stack

Better Stack – перспективний концепт, дозволяє запитувати журнали так само, як ви надсилали б запит до бази даних за допомогою SQL-сумісного структурованого журналу управління. Пропонуючи інтеграцію в такі стеки, як Kubernetes, Heroku, Logstash, Rails, Docker або AWS тощо, для яких існує широкий набір варіантів моніторингу. Завдяки спеціально створеній технології на основі ClickHouse ви можете працювати з вашими журналами ефективніше та заощаджуйте кошти. (рис. 2.15)

Better Stack без особливих зусиль шукає в петабайтах журналів за лічені хвилини і є готовий подати сигнал тривоги, якщо буде зареєстровано будь-яку аномалію, присутність або відсутність. Завдяки множинній інтеграції з кількома інструментами DevOps Better Stack пропонує надійне рішення для моніторингу журналів.

Зібрані дані візуалізуються Grafana, що забезпечує ще більшу ефективність управління Intel. Посилена безпека є однією з головних переваг журналу моніторинг, а сам Better Stack є одним із найбезпечніших доступних інструментів.

Використання галузевих стандартів кращих практик і співпрацювати тільки з центрами обробки даних сумісний із сертифікатами DIN ISO/IEC27001, ваші дані в безпеці під час обох транзит і зберігання.

Є безкоштовна версія Better Stack, з 1 Гб на місяць обсягу даних і 72 години збереження даних. Платні рішення починаються від 24 доларів США на місяць з пакетом Freelancer, включаючи 30 ГБ щомісячного обсягу та 15-денне збереження даних. З попередньо встановленими пакетами, ви можете досягти аж до бізнес-моделі за 120 доларів США на місяць, яка пропонує 150 Обсяг даних у ГБ і можливість придбати додатковий обсяг пам’яті всього за 0,25 дол. США/ГБ. Проте, якщо ви шукаєте спеціальну цінову пропозицію, Better Stack готовий до корпоративних цілей.

Основні переваги Better Stack:

– Better Stack використовує ClickHouse , що дозволяє йому працювати з більшою ефективністю та таким чином заощадити кошти;

– добре розроблений, легкий для сприйняття ШІ в темному режимі;

– розширені функції співпраці.

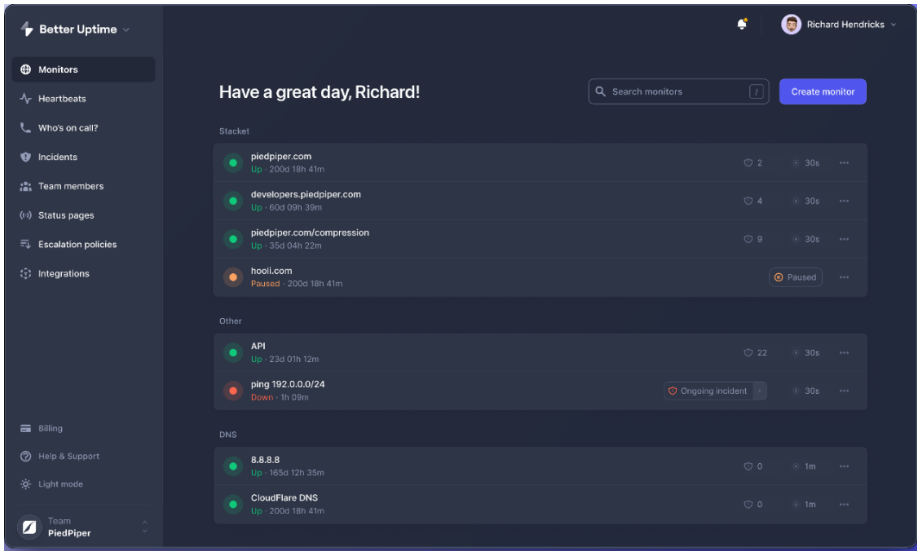


Рисунок 2.15 Зовнішній вигляд системи моніторингу Better Stack

2.16 Система моніторингу Site24x7 Infrastructure Monitoring

Site24x7 – це хмарне рішення для моніторингу, здатне контролювати такі компоненти інфраструктури, як сервери, мережі, контейнери та платформи віртуалізації. Незалежно від того, чи розміщено це локально чи в хмарі, на сервері, який відстежується, потрібно встановити агента. Site24x7 може збирати всі відповідні показники з серверів Windows і Linux і передавати інформацію в єдину консоль. Це включає в себе критичні показники продуктивності Windows, такі як використання ЦП/пам'яті/диска, служби та працездатність процесів, а також показники сервера Linux, такі як середнє завантаження та кількість потоків і обробки процесів. (рис. 2.16)

Дані, зібрані агентом, відображаються на інформаційних панелях із представленнями, що охоплюють інформацію про мережу, активність програми та показники сервера, щоб надати вам уявлення про стан інфраструктури в реальному часі. Ви можете використовувати Site24x7 для моніторингу продуктивності хостів Docker і кластерів Kubernetes. Окрім готових можливостей моніторингу інструмента, ви можете писати власні плагіни моніторингу за допомогою Shell, PowerShell, Batch, VB, Python тощо.

До плюсів можна віднести:

- можливість моніторингу понад 60 показників продуктивності серверів;
- моніторинг і аналіз служб і процесів Windows і Linux у реальному часі;
- автоматичне виявлення, відображення та моніторинг мережевих пристроїв;
- відстежує доступність і ефективність таких служб, як DNS, FTP і SMTP;
- понад 100 інтеграцій плагінів для таких програм, як MySQL і Apache.

До мінусів:

- складний у налаштуванні та конфігурації через низку доступних опцій;
- моніторинг сервера обмежений кількома технологіями.

Site24x7 пропонує безкоштовну 30-денну пробну версію. Ціни на моніторинг інфраструктури починаються від 8 доларів США на місяць до 10 серверів із можливістю придбання додаткових надбудов моніторингу.

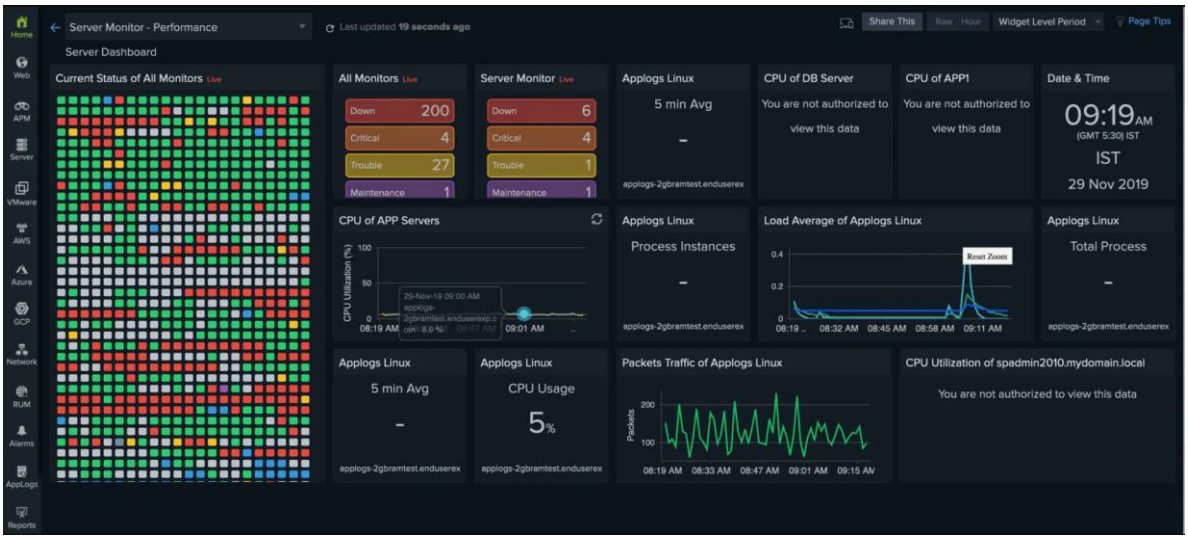


Рисунок 2.16 Зовнішній вигляд системи моніторингу Site24x7

3 АНАЛІЗ СИСТЕМ МОНІТОРИНГУ ДЛЯ ЗБОРУ ПОКАЗНИКІВ СЕРВЕРА

Моніторинг продуктивності сервера – це процес відстеження ефективності використання ресурсів вашого сервера шляхом збору та аналізу показників. Саме тут вступає в дію програмне забезпечення для моніторингу – воно гарантує безперебійну роботу програм і наявність достатнього ресурсу серверів для виконання процесів. Це означає, що він відстежує певні ключові компоненти сервера:

– ЦП: в ідеалі відсоток використання ЦП має досягати максимуму лише рідко, і піки мають бути короткими. Якщо ЦП часто перевантажується або наближається до максимального навіть у періоди непікової навантаження, це означає, що система не має достатнього запасу;

– використана пам’ять: якщо цей індикатор досягає ліміту пам’яті, слід подумати про горизонтальне масштабування серверів або додавання додаткової оперативної пам’яті;

– зберігання: знати про використання дискового сховища є вирішальним у виробничих системах – якщо на дисках закінчиться простір, призведуть до збоїв системи, та некоректної роботи сервісів;

– мережа: відстежуючи смугу пропускання та пропускну здатність мережі, ви можете бачити, як трафік доставляється на ваш сервер.

Відповідно – це детальний процес спостереження, вимірювання та аналізу ключових показників продуктивності (KPI) сервера. Відстеження часу відповіді, центральний процесор (ЦП), використання пам’яті та пропускна здатність мережі допомагають забезпечити функціональну взаємодію з користувачем.

Під час моніторингу, адміністратор:

– збирає та аналізує дані про використання ресурсів сервера;

– визначає вузькі місця;

– вирішує проблеми продуктивності.

Це дає змогу завчасно вирішувати потенційні проблеми до їх виникнення, та загострення. Таким чином веб-ресурс, сервер або програма стабільно працюють з високим показником Uptime.

Продуктивність сервера також залежить від масштабованості. Якщо виникне необхідність збільшити трафік веб-сайту, не знаючи внутрішніх показників сервера, існує ризик втратити продуктивність що приведе до негативних наслідків. Але це не єдині причини критичного моніторингу продуктивності сервера.

3.1 Переваги моніторингу продуктивності сервера

Моніторинг продуктивності сервера передбачає кілька інших ключових переваг, які є важливими для компаній, що працюють у сучасному швидкому цифровому середовищі. (рис.3.1)

Рисунок 3.1 Переваги моніторингу продуктивності сервера

– Збільшення рівня задоволення кінцевого користувача – веб-сайти або програми, що завантажуються повільно, часто призводять до розчарування кінцевих користувачів, які можуть залишити їх і ніколи не повернутися. Проактивність допомагає кінцевому користувачу, працювати швидше та плавніше;

– економія коштів та оптимізація використання ресурсів – моніторинг сервера може заощадити ваші гроші, якщо постійно відстежувати використання ресурсів і ефективність цього процесу. А покращена безпека заощадить гроші, уникнувши витоку даних та інших ризиків безпеки;

– посилена безпека – моніторинг продуктивності сервера допомагає виявити незвичайні моделі або дії, які можуть вказувати на порушення безпеки або спроби атак. Виявлення та визначення потенційних кіберзагроз допомагає захистити дані клієнтів і зберегти довіру;

– усвідомлене прийняття рішень – ключові показники ефективності сервера дають змогу отримати уявлення про внутрішні операції. Адміністратор можете визначити області, де потрібні додаткові ресурси або оптимізація. KPI внутрішніх серверів допоможуть визначити траєкторію, необхідну для досягнення цілей бізнесу.

3.2 Ключові показники моніторингу (KPI)

Моніторинг продуктивності, як правило, передбачає вимірювання показників ефективності протягом певного часу порівняно з показниками ефективності.

Ключові компоненти успішної стратегії моніторингу продуктивності сервера включають: визначення ключових показників; базові показники, пов’язані з продуктивністю сервера; звіт про додаткову цінність ключових показників. (рис. 3.2)

Рисунок 3.2 Ключові показники продуктивності

Таким чином, моніторинг продуктивності сервера здійснюється шляхом відстеження ключових показників, які забезпечують безперебійну роботу та продуктивність сервера. Деякі ефективні індикатори допомагають визначити, чи продуктивність сервера є оптимальною чи потребує покращення. Ці показники можуть включати кількість запитів за секунду, частоту помилок, час безвідмовної роботи, кількість потоків, середній час відповіді та час максимальної відповіді.

До цих показників відносяться:

– Запити на секунду (RPS – Requests Per Second) – основною функцією сервера є отримання та обробка запитів. Продуктивність сервера може погіршитися, коли кількість запитів збільшується або є нестабільною. RPS – це показник, який обчислює кількість запитів, отриманих протягом періоду моніторингу. RPS вказує на проблему продуктивності сервера, якщо виникають проблеми під час обробки запитів. Таким чином, це індикатор завантаження сервера;

– частота помилок – це небажані проблеми, які можуть погіршити продуктивність сервера. Зазвичай вони виникають, коли сервер сильно навантажений. Коефіцієнт помилок – це показник, який обчислює відсоток запитів, які не вдалися або не отримали відповідь від сервера. Це найважливіший показник, на який слід звернути увагу, коли вирішуєте проблеми з продуктивністю сервера;

– час роботи (Uptime) – найбільш критичною проблемою для будь-якої операції є доступність сервера. Час безвідмовної роботи означає, як довго сервер працював за певний період без значних збоїв. Якщо метрика безвідмовної роботи стає меншою за 99% часу використання сервера, це потребує уваги. Щодо контексту, архітектура сервера високої доступності підтримує доступність 99,999% навіть під час запланованих і незапланованих відключень, також відома як надійність Five Nines. Сервер має бути надійним для кінцевих користувачів, тому час безвідмовної роботи є хорошим показником проблем із продуктивністю;

– підрахунок потоків – параметри підрахунку потоків визначають максимальну кількість запитів, які сервер може обробляти одночасно, що може бути суттєвим показником продуктивності сервера. Коли програма генерує забагато потоків,

кількість помилок може збільшуватися. Коли кількість потоків досягає максимального порогу, запити призупиняються, доки не з’явиться вільне місце. Коли час утримання занадто довгий, користувачі стикаються з помилками тайм-ауту;

– середній час відгуку (ART – Average Response Time) і піковий час відгуку (PRT – Peak Response Time) – ART обчислює загальний час циклу запитів/відповідей, взятий для всіх запитів, поділений на кількість запитів. PRT обчислює тривалість циклів часу запиту/відповіді, щоб відстежити найдовший цикл протягом періоду моніторингу. Оцінка показників ART і PRT є найефективнішою технікою для отримання точного розуміння часу відповіді.

Налаштування критичних значень для сповіщання

Порогові сповіщення допомагають бізнесу зрозуміти обмеження можливостей сервера, завчасно запобігаючи простоям і збоям. Налаштувавши сповіщення, адміністратор уникає більших проблем у майбутньому та встановлює попередньо визначені обмеження.

Необхідно визначити допустимі межі для кожного показника на основі базових значень. Після чого, потрібно налаштувати інструмент моніторингу, щоб ініціювати сповіщання, коли певний поріг буде досягнуто, а також можна налаштувати сповіщання за допомогою будь-якого іншого каналу зв’язку, як от електронна пошта, чи доступні месенджери, що дає змогу адміністраторам вирішувати проблему з сервером, як тільки вони виникають.

Регулярна перевірка звіту продуктивності

Звіти про продуктивність пропонують цінну інформацію про тенденції, закономірності та аномалії в інфраструктурі сервера. Вони дозволяють визначити сфери, які потребують покращення, і потенційні проблеми. Необхідно запланувати регулярні перевірки ефективності. Потрібно встановити частоту на основі характеру потреб сервера, для цього, треба шукайте шаблони, області вдосконалення та значні піки в інструменті моніторингу. Залежно від цього аналізу може виникнути необхідність оптимізувати сервери додатків або виконати певне усунення несправностей.

Оптимізація налаштувань та використання ресурсів сервера

Необхідно регулярно оптимізувати ресурси сервера, щоб він працював з максимальним показником продуктивності. Спочатку необхідно оновити програмне забезпечення. Операційні системи, програми та навіть апаратне забезпечення завжди мають бути оновленими. Таким чином адміністратор запобігає вразливості безпеки та забезпечує оптимальну продуктивність. Припустимо, адміністратор помітив, що процесор перевантажується в періоди високого трафіку. Рішенням буде підвищення потужності процесора; якщо система майже досягла максимального обсягу виділеної пам'яті, варіантом виправлення, є розширення пам'яті. Основним принципом оптимізації, є пошук потенційно слабких місць та завчасне вирішення будь-яких проблем.

Регулярне технічне обслуговування сервера

Для того, щоб гарантувати, що сервер залишається у найкращій формі, необхідно виконувати регулярні завдання з технічного обслуговування.

Одним з основних кроків, необхідність робити резервні копії, щоб забезпечити цілісність даних і забезпечити швидке відновлення в разі втрати даних або збою сервера. Потрібно відстежувати та керувати використанням дискового простору, щоб запобігти проблемам, пов'язаним із вичерпанням обсягу сховища. Необхідно виконувати регулярні перевірки безпеки, щоб виявити вразливі місця та застосувати необхідні виправлення або оновлення. Також до ключових аспектів, відноситься перегляд та оптимізація журналів (log) сервера, щоб виявити тенденції, помилки або незвичну діяльність, яка може потребувати подальшого дослідження.

3.3 Ключові поради, щодо моніторингу серверного обладнання

Моніторинг продуктивності сервера дозволяє адміністраторам відстежувати поглиблену інформацію про статус і справність сервера. Для цього, під час вибору інструменту моніторингу необхідно враховувати різні фактори. (рис. 3.3).

До ключових методів моніторингу продуктивності сервера, відносяться:

– Необхідно налаштувати візуальний інтерфейс користувача – візуалізація – це графічне представлення інформації та даних за допомогою таких інструментів, як графіки, діаграми та карти. Чітке відображення всієї конструкції мережі, отримання чіткого візуального представлення ключових даних і звітування про працездатність сервера допомагають адміністраторам контролювати, розуміти та приймати рішення щодо оптимізації продуктивності сервера;

– необхідно налаштувати докладні сповіщення – сповіщення в режимі реального часу дає адміністраторам інформацію про будь-які проблеми, допомагаючи швидко їх вирішити. Детальні сповіщення, як-от автоматичні повідомлення або сповіщення від інструменту моніторингу, які пропонують рекомендовані процедури для усунення проблеми, є більш цінними, ніж прості сповіщення. Адміністратори серверів повинні спочатку перевірити серйозність проблеми та зрозуміти логічні наслідки. Якщо проблема матиме серйозний вплив на сервер, адміністратор може прийняти ефективні рішення щодо подальших кроків для її вирішення;

– необхідно регулярно проводити додатковий моніторинг стану сервера – справність сервера стосується стану основних функцій сервера. Моніторинг справності сервера відіграє важливу роль у виявленні збоїв у сервері та мережі, та може допомогти визначити налаштування роботи сервера, заміну апаратного забезпечення та оптимізацію продуктивності. Фізична перевірка може включати використання ЦП, доступність пам’яті та ємність диска. Моніторинг працездатності сервера надає дані, які можуть бути корисними для прогнозування проблем сервера, порівнюючи поточні та історичні дані.

Потрібно розуміти, чи сумісні всі компоненти, налаштувати зручний дешборд, розуміти чи потрібні інтеграція з якимись іншими інструментами, зважити ціну та якість, та забезпечити надійність і відповідність стандартам безпеки.

Рисунок 3.3 Фактори вибору системи моніторингу

Враховуючи всі ці фактори, необхідно бути впевненим що:

– інструмент, має бездоганно інтегруватися з серверною інфраструктурою. Він має підтримувати операційні системи (наприклад, Windows або Linux), платформи та технології, на які покладається бізнес. Перш ніж шукати рішення, необхідно переглянути документацію інструмента та переконатися, що він сумісний із поточним серверним середовищем. (Це стосується як хмарного хостингу, так і локальних серверів.);

– інструмент має забезпечувати настроювані дешборди, для відстеження продуктивності сервера в реальному часі. Таким чином адміністратор зможе швидко визначити проблеми та тенденції. Необхідно обирати рішення, яке дозволить створювати спеціальні інформаційні панелі відповідно до поточних потреб і вподобань. Організована, інтуїтивно зрозуміла інформаційна панель і платформа мають велике значення для моніторингу продуктивності сервера;

– крім того, необхідно обирати інструмент, який інтегрується з іншими сервісами та додатками. Наприклад, ви хочете, щоб ваше програмне забезпечення для моніторингу ефективності інтегрувалося з хмарними постачальниками та

інструментами керування інцидентами. Можливості інтеграції можуть оптимізувати ваші процеси, зменшити ручні зусилля та підвищити ефективність вашої команди;

– незважаючи на те, що вартість завжди є одним з ключових факторів, дуже важливо збалансувати між ціною та якістю. Необхідно оцінити характеристики та визначити, чи виправдана вартість відповідно до поточних вимог;

– надійна та швидка команда технічної підтримки є важливою для будь-якого інструменту моніторингу серверів. Необхідно обирати систему моніторингу, яка має швидку та ефективну службу підтримки, щоб вирішити проблеми та мінімізувати час простою. Відповідно потрібно рішення, яке пропонує комплексну підтримку через різні канали, такі як телефон, електронна пошта та чат. Якщо інструмент не пропонує підтримку, вирішення проблем може зайняти більше часу;

– хороший інструмент моніторингу має пропонувати настроювані сповіщення, щоб інформувати адміністратора про важливі події та потенційні проблеми. Необхідно обирати інструмент, який дає змогу налаштувати порогові сповіщення та налаштувати спосіб отримання сповіщень, наприклад електронною поштою, SMS або програмою. Це гарантує, що адміністратор зможе оперативно та ефективно реагувати на будь-які події в системі;

– інструмент моніторингу сервера має надавати пріоритет безпеці та відповідати галузевим стандартам і правилам. Необхідно обирати рішення, яке пропонує надійні функції безпеки, такі як шифрування, безпечне зберігання даних і контроль доступу.

3.4 Системи моніторингу продуктивності сервера (Nagios, Zabbix, AppDynamics)

Відштовхуючись від порад, щодо моніторингу серверного обладнання, можна виділити основні критерії:

– система моніторингу повинна мати можливість тонкого налаштування дешборду, детального налаштування сповіщань та метрик, та давати можливість регулярно перевіряти показники роботи сервера (програмні та апаратні);

– система моніторингу повинна бути сумісна з інфраструктурою сервера, його операційною системо, додатками та сервісами, які крутяться на сервері;

– обираючи систему моніторингу, необхідно звертати увагу не лише на її ціну, але і на якість, сервісну підтримку та безпеку з відповідністю галузевих стандартів, та відповідно забезпечення надійного зберігання даних та контроль доступу.

Серед усіх систем моніторингу, представлених на ринку, я б хотів виділити лише декілька з них, оскільки вони відповідають всім вище переліченим вимогам: Nagios XI, Zabbix та AppDynamics.

Системо моніторингу продуктивності сервера Nagios

Як і писалось раніше, Nagios XI – комплексне програмне забезпечення для моніторингу корпоративних серверів і мереж. Бізнес-версія Nagios XI, була створена на основі версії з відкритим кодом і має більше функціональних можливостей, що вимагає значно менше часу на адміністрування. Nagios зосереджується в основному на показниках сервера, продуктивності додатків і мережевому трафіку. Він збирає дані за допомогою агентів, розміщених як на елементах мережі, так і на компонентах, які він контролює, тобто за допомогою нативного протоколу. (рис. 3.4)

Рішення легко настроюється та масштабується, що робить його ідеальним для багатьох підприємств. Однак ця висока можливість налаштування супроводжується додатковою складністю та витратами на обслуговування.

Агент живе на елементі мережі, наприклад на сервері Linux. Nagios звертається до агента, щоб перевірити різні статистичні дані (наприклад, простір на диску, оперативна пам’ять, використання ЦП тощо). Агент збирає запитувану інформацію та відповідає Nagios XI. Nagios спочатку зберігає інформацію для подальших звітів, історичних діаграм і графіків, а потім іншою дією, яку він може виконати, є створення сповіщення. За допомогою сповіщення статистичні дані, які

повертає агент, можуть вказувати на занадто заповнений диск або інший стан, а згенероване сповіщення – це те, як адміністратор дізнається про потенційну проблему.

Ще один спосіб моніторингу пристроїв – це використання рідного протоколу. Існує два рідних протоколи, які використовує Nagios: один – SNMP, а інший WMI, який є специфічним для середовищ Windows. Наприклад, якщо у вас є мережевий комутатор із увімкненим і налаштованим протоколом SNMP, Nagios XI може зв’язатися, щоб дізнатися, як працює комутатор. Потім комутатор відповість своїм станом (наприклад, усе добре, порт не працює або щось інше, що може відбуватися).

Для зручного налаштування, на офіційному сайті, є повноцінна детальна інструкція.

До плюсів, можна віднести:

- підтримка мережевих компонентів, таких як маршрутизатори, комутатори та інше фізичне обладнання;
- можливість налаштування; підтримує спеціальні показники;
- підтримує моніторинг серверів Windows і Linux.

До мінусів:

- обмежений набір інформаційних панелей за замовчуванням;
- накладні витрати на технічне обслуговування та експлуатацію.

Nagios XI безкоштовний для невеликих середовищ, але після семи вузлів моніторингу, потрібна річна ліцензія на підтримку та обслуговування сервера Nagios. Базова версія коштує 1995$, в той час, як Enterprise Edition – 3495$.

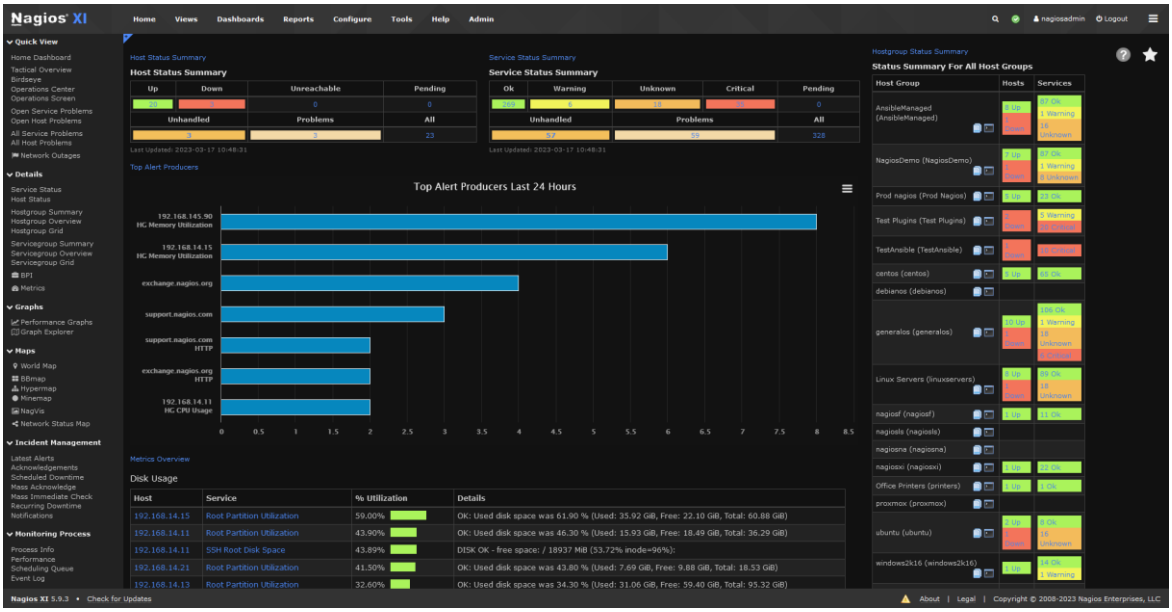


Рисунок 3.4 Дешборд Nagios XI

Система моніторингу продуктивності сервера Zabbix

Zabbix — це програмний інструмент моніторингу з відкритим вихідним кодом для різних IT-компонентів, включаючи мережі, сервери, віртуальні машини (ВМ) і хмарні служби. Zabbix надає метрики моніторингу, такі як використання мережі, завантаження процесора та використання дискового простору. (рис. 3.5)

Програмне забезпечення відстежує роботу в Linux, Hewlett Packard Unix (HP-UX), Mac OS X, Solaris та інших операційних системах (ОС); однак моніторинг Windows можливий лише через агентів. Zabbix можна розгорнути для моніторингу з агентами та без агентів. На IT-компоненти встановлюються агенти для перевірки продуктивності та збору даних. Потім агент звітує на централізований сервер керування Zabbix. Ця інформація включається у звіти або представлена візуально в графічному інтерфейсі користувача Zabbix (GUI). Якщо виникнуть проблеми щодо того, що відстежується, Zabbix надішле сповіщення або сповіщення користувачеві. Моніторинг без агентів виконує той самий тип моніторингу за допомогою наявних ресурсів у системі чи пристрої для емуляції агента.

Веб-інтерфейс Zabbix дозволяє користувачам переглядати своє IT-середовище за допомогою настроюваних інформаційних панелей на основі віджетів, графіків, мережевих карт, слайд-шоу та звітів. Наприклад, користувач може налаштувати звіт, щоб відображати показники, пов’язані як з угодами про

рівень обслуговування (SLA), так і з ключовими показниками продуктивності (KPI) щодо навантаження на ЦП.

Для зберігання даних використовують MySQL, PostgreSQL, SQLite або Oracle Database, вебінтерфейс написаний на PHP. Підтримує кілька видів моніторингу.

Simple checks – може перевіряти доступність і реакцію стандартних сервісів, таких як SMTP або HTTP, без встановлення будь-якого програмного забезпечення на хості, що спостерігається.

Zabbix agent – може бути встановлений на UNIX-подібних або Windows-хостах для отримання даних про навантаження процесора, використання мережі, дисковому просторі тощо.

External check – виконання зовнішніх програм, також підтримується моніторинг через SNMP.

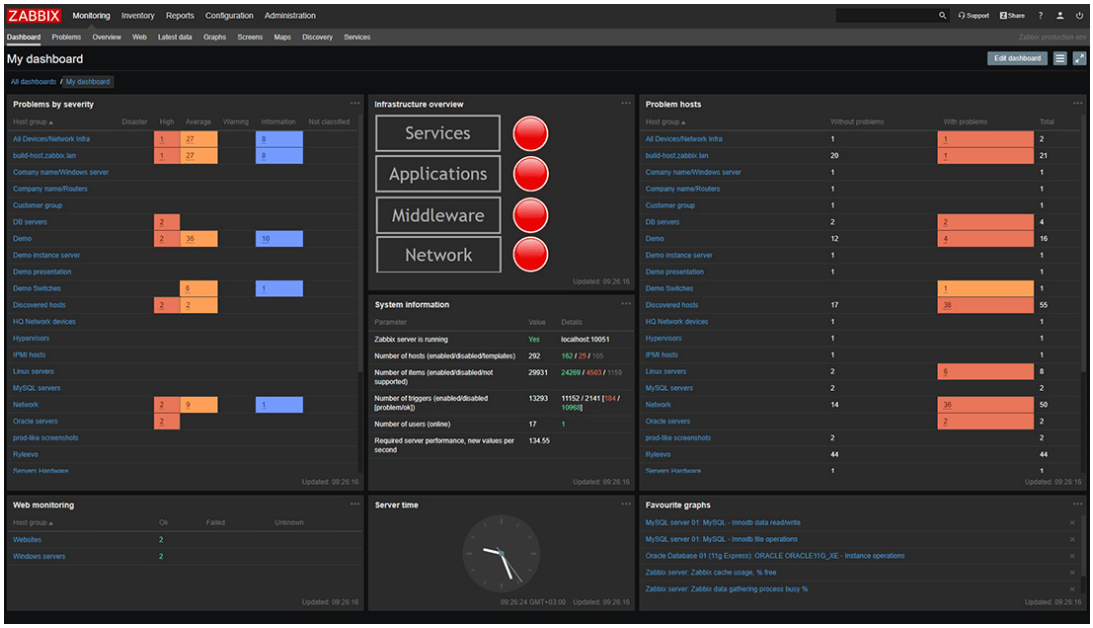


Рисунок 3.5 Дешборд Zabbix

Zabbix це монітонинг з відкритим кодом і є безкоштовним, але якщо є необхідність, є можливість доповнити його платними послугами, такими як технічна підтримка, консультації та підтримка оновлення/створення шаблонів.

Система моніторингу продуктивності сервера AppDynamics

AppDynamics – надає комплексне рішення для моніторингу інфраструктури, яке охоплює компоненти сервера, сховища та мережі як у хмарних, так і в гібридних середовищах. Це провідний продукт для керування продуктивністю

додатків (APM). Це інструмент, який відстежує інфраструктуру додатків і надає видимість на рівні коду. Він підтримується всіма основними технологіями (Java, .NET, PHP, Node.js, NOSQL тощо) і може бути встановлений як локальне рішення, так і як рішення SaaS (програмне забезпечення як послуга).

Частина програмного забезпечення під назвою «Агент» встановлюється в додатку, який потрібно контролювати. Агент збирає показники продуктивності та надсилає їх до процесу Сервера під назвою Контролер. Контролер обробляє показники та представляє їх через веб-браузер. Адміністратор моніторингу може налаштовувати оповіщення та створювати звіти за допомогою веб-інтерфейсу.

Агент постійно контролює заявку. Оскільки він використовує технологію інструментування байт-коду, Agent має підключення до кожного рядка коду. Ось як AppDynamics може забезпечити видимість рівня коду. Агенти доступні для більшості популярних технологій.

Більшість можливостей моніторингу є «нестандартними», включаючи оповіщення. Ще одна чудова функція «з коробки» – це «Відображення потоку додатків». AppDynamics виявляє різні підсистеми та серверні модулі та чудово малює їх у браузері. Дозволяє виявляти бекенди, з якими спілкується система. Крім того, AppDynamics «вивчає» поведінку додатків і автоматично встановлює базові лінії та попереджає, коли відхилення від базової лінії не є нормальним (аномалія).

Можливості повного стека моніторингу цього інструменту допомагають співвідносити проблеми продуктивності програми з вузькими місцями інфраструктури низького рівня, прискорюючи тим самим аналіз основних причин і усунення.

Завдяки повному набору інформаційних панелей і показників AppDynamics підтримує детальні сповіщення, які можна інтегрувати зі сторонніми інструментами оповіщення та керування інцидентами, такими як ServiceNow, PagerDuty та Jira.

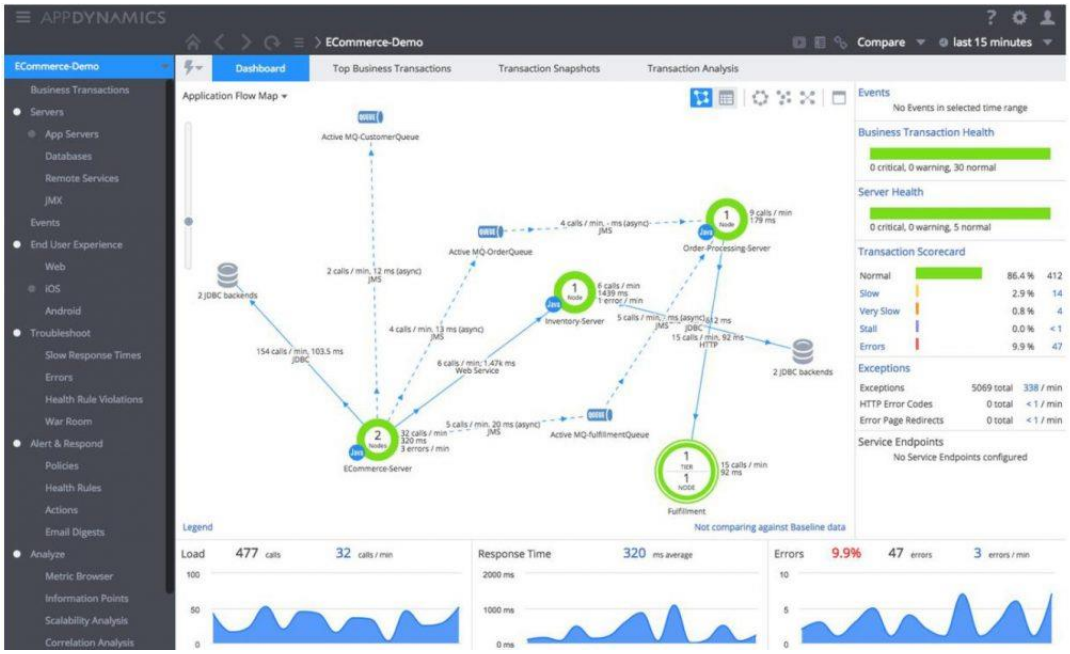


Рисунок 3.6 Дешборд AppDynamics

AppDynamics стягує плату за ядро ЦП. Доступна 15-денна безкоштовна пробна версія. Від стандартної версії за 6$ за одне ядро ЦП, на місяць, до Enterprise Edition – 90$ за одне ядро ЦП, на місяць.

3.5 Аналіз та порівняння систем моніторингу

Серед цих, вище перелічених систем моніторингу, для ефективного відстеження поведінки серверного обладнання і не тільки, кожен з інструментів, має всі можливості, щоб робити це якісно та інформативно.

Якщо порівняти Nagios, Zabbix та AppDynamics, то за даними Gartner Peer Insights, Zabbix має рейтинг 4,4 зірки з 300 відгуків, тоді як Nagios має рейтинг 4,3 зірки з 175 відгуків, а система моніторингу AppDynamics є самою популярною серед користувачів, оскільки на 900 відгуків, рейтинг становить 4,5 зірки. Всі інструменти мають свої переваги та недоліки, тому вибір залежить від потреб організації.

Nagios – це система моніторингу з відкритим вихідним кодом, яка дозволяє організаціям виявляти та вирішувати проблеми IT-інфраструктури до того, як вони

вплинуть на критичні бізнес-процеси. Це потужний і гнучкий інструмент, який можна налаштувати відповідно до конкретних потреб усієї IT-інфраструктури. Якщо порівнювати Nagios з Zabbix та AppDynamics, то Nagios незалежно від того, чи потрібно відстежувати мережеві пристрої, дані журналу чи будь-що інше, що працює від електрики – дає змогу робити це все, за допомогою потужного та масштабованого механізму моніторингу, планування потужностей, відстеження інцидентів тощо. Nagios надсилає негайні сповіщення, коли в системі спостерігається будь-яка незвичайна поведінка, щоб можна було вжити відповідних заходів щодо аварій чи алармів. Ця система моніторингу є більш дорогою, в порівнянні з Zabbix, та в умовах масштабованості, є дешевшою за AppDynamics.

Zabbix – це також система моніторингу з відкритим вихідним кодом, яке забезпечує моніторинг мережі корпоративного класу та рішення моніторингу програм. Це масштабоване, розподілене та безпечне програмне забезпечення, яке може контролювати будь-що: від серверів до програм і служб. Zabbix доступний безкоштовно та має професійні послуги та підтримку. Його високо оцінюють технічні спільноти по всьому світу. Якщо порівнювати Zabbix з Nagios та AppDynamics, то Zabbix завдяки своїй відмінній масштабованості, може підійти для використання від малих систем, до надзвичайно великих, зі значною кількістю хостів, що дає можливість контролювати тисячі показників, зібраних з різних віртуальних і фізичних машин. Маючи можливість тонко налаштувати дешборд, Zabbix є чудовим та зручним інструментом для моніторингу ключових показників, що в свою чергу допомагає системам завжди залишатися з високим показнико Uptime та KPI. Система моніторингу, є безкоштовною, але не зважаючи на це, є досить зручною навіть в такому виконанні, що дає змогу використовувати її малим бізнесам, скорочуючи витрати на моніторинг. А у випадку масштабованості – використання Zabbix дешевше ніж Nagios та AppDynamics.

AppDynamics – це інструмент моніторингу, який дозволяє відстежувати стан серверів, додатків та мережі. Якщо порівняти AppDynamics з Nagios та Zabbix, то AppDynamics має більш простий інтерфейс користувача та більш широкий

функціонал, проте дуже погано реалізована інтеграція з AWS, якщо це можна так назвати. А також, істотним мінусом, є зависока ціна, в порівняння з відносно безкоштовним Zabbix, та Nagios з його місячною передплатою.

Відповідно, підсумувавши все, якщо потрібен інструмент моніторингу, який має більшу кількість налаштувань та можливостей, то Nagios та Zabbix можуть бути кращим вибором. Інструменти Zabbix і Nagios використовуються для моніторингу системи. Для вибору інструменту моніторингу єдиним фактором, який потрібно враховувати, є вимоги до інструменту та використання, з урахуванням яких, можна обрати найкращий варіант, в певних умовах і під певні задачі. Обидва інструменти здатні контролювати сервери та продуктивність системи. Zabbix з урахуванням його безкоштовної версії, пропонує більше функціоналу прямо з коробки, в той час, як Nagios – має дуже обмежений функціонал. Але все це залежить від вимог та цілей, які ставить перед собою проект. Якщо потрібен інструмент моніторингу з більш простим інтерфейсом користувача та більш широким функціоналом враховуючи всі за та проти, та висока ціна не є проблемою, то AppDynamics може бути кращим вибором.

ВИСНОВКИ

Під час написання роботи детально проаналізував, дослідив і описав характеристики, особливості, переваги та недоліки сучасних систем моніторингу, для збору показників сервера. Сучасні складні IT-середовища складаються з переплетених рівнів інфраструктури та вбудованого програмного забезпечення, сервісів і служб, а також програмного забезпечення користувача. Кожен рівень генерує свої власні показники, тому дуже важливо мати хороші інструменти моніторингу системи. Моніторинг продуктивності сервера має вирішальне значення для виявлення ризиків і оптимізації продуктивності сервера. Завдяки використанню сучасних систем моніторингу, адміністратор зможе швидко й ефективно налагоджувати та усувати проблеми продуктивності системи. Це в свою чергу, дозволяє клієнтам, співробітникам та партнерам отримувати задоволення від високопродуктивних і завжди доступних програм; DevOps зможуть швидко й ефективно налагоджувати та усувати проблеми продуктивності системи.

Дані, зібрані системами IT-моніторингу, дають організації глибоке уявлення про IT-середовище. Це допомагає запобігти можливим простоям і стає все більш корисним у міру розвитку IT-середовища. З цього виходить, що:

– необхідно обирати систему моніторингу, яка дозволяє налаштовувати політки, докладні сповіщання та візуальні дешборди, на вимогу користувача. Тобто максимально гнучку систему моніторингу. Оскільки не всі системи моніторингу дозволяють тонко підлаштовувати певні параметри під конкретні потреби, а працюю за шаблонами;

– обираючи систему моніторингу, необхідно враховувати всі фактори, пов’язані з інтеграцією та сумісністю компонентів, служб та поточною інфраструктурою;

– в залежності від потреб, майже всі системи моніторингу, пропонують різні пакети своїх сервісів, від безкоштовних з відкритим кодом, до передплачуваних, з місячною або річною оплатою, і відрізняються насиченістю цих пакетів.

ПЕРЕЛІК ПОСИЛАНЬ

1. IT-monitoring [Електронний ресурс] – Режим доступу до ресурсу: https://www.techtarget.com/searchitoperations/definition/IT-monitoring
2. Introduction to IT Monitoring [Електронний ресурс] – Режим доступу до ресурсу: https://www.bmc.com/blogs/it-monitoring/
3. What is IT-Monitoring [Електронний ресурс] – Режим доступу до ресурсу: https://www.splunk.com/en_us/blog/learn/it-monitoring.html?301=/en_us/data-insider/what-is-it-monitoring.html
4. What is IT-Monitoring [Електронний ресурс] – Режим доступу до ресурсу: https://alertops.com/it-monitoring/
5. System monitor [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/System_monitor
6. Essential features of a remote monitoring system [Електронний ресурс] – Режим доступу до ресурсу: https://www.hexnode.com/blogs/essential-features-of-a-remote-monitoring-system/
7. What Is Internet Control Message Protocol (ICMP) [Електронний ресурс] – Режим доступу до ресурсу: https://www.fortinet.com/resources/cyberglossary/internet-control-message-protocol-icmp
8. Secure Shell [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Secure_Shell
9. WMI [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Windows_Management_Instrumentation
10. System Monitoring Tools [Електронний ресурс] – Режим доступу до ресурсу: https://sematext.com/blog/system-monitoring-tools/
11. Server Monitoring Tools [Електронний ресурс] – Режим доступу до ресурсу: https://sematext.com/blog/server-monitoring-tools/
12. Infrastructure Monitoring Tools [Електронний ресурс] – Режим доступу до ресурсу: https://sematext.com/blog/infrastructure-monitoring-tools/
13. Zabbix [Електронний ресурс] – Режим доступу до ресурсу: https://www.zabbix.com/
14. Nagios [Електронний ресурс] – Режим доступу до ресурсу: https://www.nagios.com/products/nagios-xi/
15. AppDynamics [Електронний ресурс] – Режим доступу до ресурсу: https://www.appdynamics.com/
16. LibreNMS [Електронний ресурс] – Режим доступу до ресурсу: https://www.librenms.org/

17. Checkmk [Електронний ресурс] – Режим доступу до ресурсу: https://checkmk.com/
18. Incinga [Електронний ресурс] – Режим доступу до ресурсу: https://icinga.com/
19. The Elastic Stack [Електронний ресурс] – Режим доступу до ресурсу: https://www.elastic.co/elastic-stack
20. New Relic [Електронний ресурс] – Режим доступу до ресурсу: https://newrelic.com/
21. Sematext Monitoring [Електронний ресурс] – Режим доступу до ресурсу: https://sematext.com/
22. SolarWinds Server & Application Monitor [Електронний ресурс] – Режим доступу до ресурсу: https://www.solarwinds.com/server-application-monitor
23. Datadog [Електронний ресурс] – Режим доступу до ресурсу: https://www.datadoghq.com/
24. Prometheus [Електронний ресурс] – Режим доступу до ресурсу: https://prometheus.io/
25. Grafana [Електронний ресурс] – Режим доступу до ресурсу: https://grafana.com/
26. ManageEngine OpManager [Електронний ресурс] – Режим доступу до ресурсу: https://www.manageengine.com/
27. Better Stack [Електронний ресурс] – Режим доступу до ресурсу: https://betterstack.com/
28. Site 24x7 Infrastructure Mobitoring [Електронний ресурс] – Режим доступу до ресурсу: https://www.site24x7.com/infrastructure-monitoring.html
29. Server Performance Monitoring Guide [Електронний ресурс] – Режим доступу до ресурсу: https://www.serverwatch.com/guides/server-performance-monitoring-guide/
30. Best Practices For Server Performance [Електронний ресурс] – Режим доступу до ресурсу: https://www.nexcess.net/blog/server-performance-monitoring/#best-practices-for-server-performance

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)

Державний університет інформаційно-комунікаційних технологій

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Аналіз систем моніторингу для збору показників сервера»

на здобуття освітнього ступеня магістра
зі спеціальності 126 Інформаційні системи та технології
освітньо-професійної програми Інформаційні системи та технології

Виконав: здобувач вищої освіти гр. ІСДМ-62
Валентин ЮХИМЕНКО
Керівник: Доцент кафедри ПУАД
Власенко В.О.

Київ 2024

Об’єкт, предмет, мета дослідження

- **Об’єкт дослідження –** системи моніторингу.
- **Предмет дослідження –** методи, засоби моніторингу серверного обладнання.
- **Мета дослідження –** аналіз існуючих систем моніторингу, які збирають показники серверів.

Актуальність дослідження

У зв’язку з постійний розвитком серверного обладнання, та ростом, розширенням його інфраструктури, необхідність якісного відстеження та моніторингу процесів та компонентів, під час його роботи є одним з ключових моментів за якими слідкують IT-інженери. Сучасні системи моніторингу, дуже допомагають інженерам у виконанні їх роботи, та відповідно забезпеченні високого рівня продуктивності, та відмовостійкості обладнання. Оскільки якісно налаштована система моніторингу, буде охоплювати всі системи та додатки, які потребують уваги, та відповідно своєчасно сповіщатиме про аварії, або будь-які інші події.

Визначення моніторингу

Це важливий процес, який допомагає організаціям, користувачам, відстежувати зміни в IT-інфраструктурі. Мета моніторингу – збір та аналіз даних про IT-сервіси і компоненти інформаційної інфраструктури та використання цих даних для контролю всіх її елементів, а також запобігання збоїв і поломок.

Системи моніторингу – це програмне забезпечення, яке дозволяють відслідковувати ресурси сервера, такі як використання процесора, обсяг пам’яті, потужність сховища, продуктивність введення-виведення, час роботи мережі та інші.

Основні види моніторингу

Моніторинг на основі агентів зазвичай розроблений спеціально для конкретної платформи. Як наслідок, він здатний збирати та аналізувати більше даних для системи, для взаємодії з якою він був запрограмований.
Безагентний моніторинг є популярним вибором, який покладається на різноманітні протоколи, такі як SNMP, WMI, SSH, NetFlow або інші, для передачі системних даних і статистики програмному забезпеченню моніторингу. Ці вбудовані функції відстежують і керують інформацією про інфраструктуру без додаткових агентів.
SNMP (англ. Simple Network Management Protocol – простий протокол керування мережею) – це протокол керування мережами зв'язку на основі архітектури TCP/IP. SNMP – це технологія, покликана забезпечити керування й контроль за пристроями й застосунками в мережі зв'язку шляхом обміну керівною інформацією між агентами, що розташовуються на мережних пристроях, і менеджерами, розташованими на станціях керування.

Моніторинг продуктивності сервера

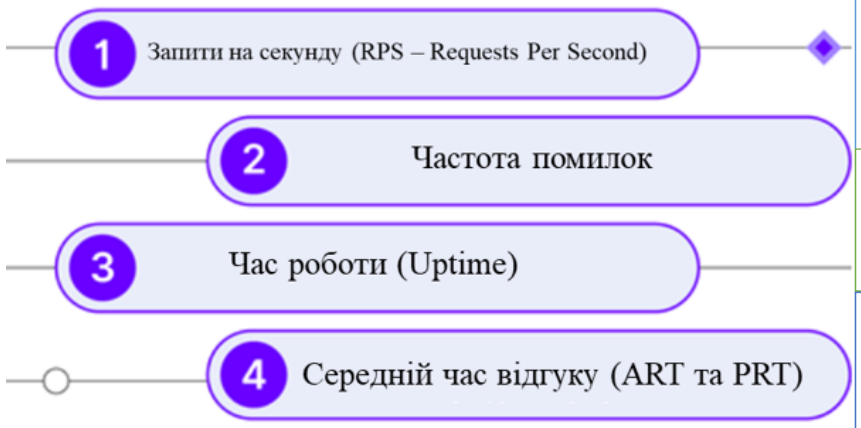
– це детальний процес спостереження, вимірювання та аналізу ключових показників продуктивності (KPI) сервера. Відстеження часу відгуку, використання центрального процесора (ЦП), споживання пам'яті та пропускної здатності мережі допомагає забезпечити функціональну взаємодію з користувачем.

У рамках моніторингу адміністратор:

- збирає та аналізує дані про використання ресурсів сервера;
- визнає вузькі місця;
- вирішує проблеми продуктивності.

Це дає змогу завчасно вирішувати потенційні проблеми до їх загострення. Таким чином, сайт або сервіси/додатки стабільно працюють, не маючи показника downtime.

Основні показники продуктивності в моніторингу



1. Основною функцією сервера є отримання та обробка запитів. Продуктивність сервера може погіршитися, коли кількість запитів збільшується або є нестабільною.

2. Небажані проблеми, які можуть погіршити продуктивність сервера. Зазвичай вони виникають, коли сервер сильно навантажений.

3. Найбільш критичною проблемою для будь-якої операції є доступність сервера. Час безвідмовної роботи означає, як довго сервер працював за певний період без значних збоїв.

4. ART – Average Response Time і PRT – Peak Response Time – ART обчислює загальний час циклу запитів/відповідей, взятий для всіх запитів, поділений на кількість запитів. PRT обчислює тривалість циклів часу запиту/відповіді, щоб відстежити найдовший цикл протягом періоду моніторингу. Оцінка показників ART і PRT є найефективнішою технікою для отримання точного розуміння часу відповіді.

Основні поради, щодо моніторингу продуктивності сервера

Моніторинг продуктивності сервера дозволяє адміністраторам відстежувати поглиблену інформацію про статус і справність сервера.

1. **Сумісність з поточною інфраструктурою** – інструмент моніторингу, має бездоганно інтегруватися з вашою серверною інфраструктурою.
2. **Легкий для розуміння Дешборд** – інструмент моніторингу, має забезпечувати настроювані дешборди, для відстеження продуктивності сервера в реальному часі.
3. **Інтеграція з іншими інструментами** – крім того, необхідно обирати інструмент моніторингу, який інтегрується з іншими сервісами та додатками.
4. **Ціна та якість** – незважаючи на те, що вартість завжди є одним з важливих факторів, дуже важливо збалансувати між ціною та якістю
5. **Технічна підтримка** – надійна та швидка команда технічної підтримки є важливою для будь-якого інструменту моніторингу серверів
6. **Настроювані сповіщення** – хороший інструмент моніторингу має пропонувати настроювані сповіщення, щоб інформувати адміністратора про важливі події та потенційні проблеми.
7. **Безпека та відповідність** – інструмент моніторингу сервера має надавати пріоритет безпеці та відповідати галузевим стандартам і правилам. Необхідно обирати рішення, яке пропонує надійні функції безпеки, такі як шифрування, безпечне зберігання даних і контроль доступу.

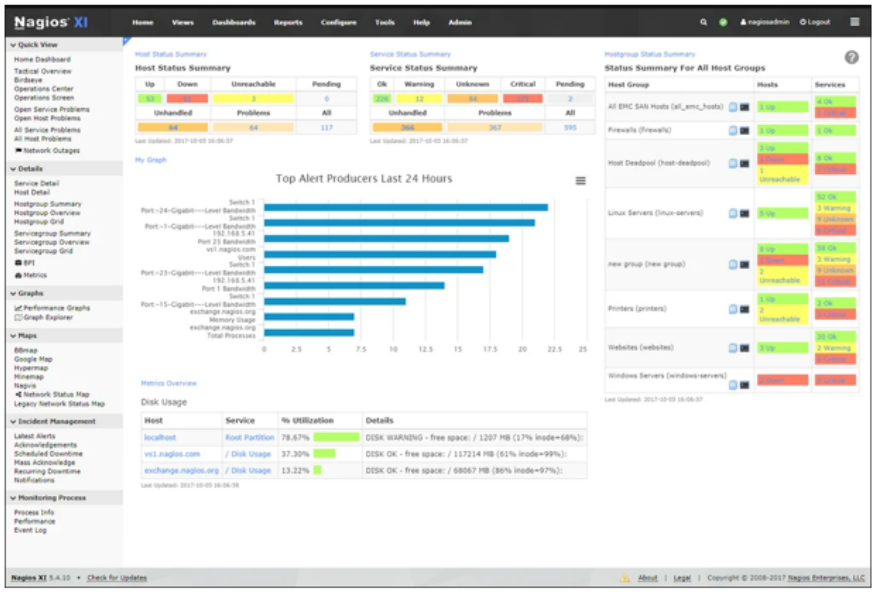
Системи моніторингу для збору показників сервера

Відштовхуючись від порад, щодо моніторингу серверного обладнання, можна виділити основні критерії:

- система моніторингу повинна мати можливість тонкого налаштування дешборду, детального налаштування сповіщань та метрик, та давати можливість регулярно перевіряти показники роботи сервера (програмні та апаратні);
- система моніторингу повинна бути сумісна з інфраструктурою сервера, його операційною системо, додатками та сервісами, які крутяться на сервері;
- обираючи систему моніторингу, необхідно звертати увагу не лише на її ціну, але і на якість, сервісну підтримку та безпеку з відповідністю галузевих стандартів, та відповідно забезпечення надійного зберігання даних та контроль доступу.

Серед усіх систем моніторингу, представлених на ринку, я б хотів виділити лише декілька з них, оскільки вони відповідають всім вище переліченим вимогам: Nagios XI, Zabbix та AppDynamics.

Nagios XI



Nagios XI безкоштовний для невеликих середовищ, але після семи вузлів моніторингу, потрібна річна ліцензія на підтримку та обслуговування сервера Nagios. Базова версія коштує 1995$, в той час, як Enterprise Edition – 3495$.

Nagios XI – комплексне програмне забезпечення для моніторингу корпоративних серверів і мереж. Бізнес-версія Nagios XI, була створена на основі версії з відкритим кодом і має більше функціональних можливостей, що онзанача вимагає менше часу на адміністрування. Nagios зосереджується в основному на показниках сервера, продуктивності додатків і мережевому трафіку. Він збирає дані за допомогою агентів, розміщених як на елементах мережі, так і на компонентах, які він контролює.

Рішення легко настроюється та масштабується, що робить його ідеальним для багатьох підприємств. Однак ця висока можливість налаштування супроводжується додатковою складністю та витратами на обслуговування.

До плюсів, можна віднести:

- підтримка мережевих компонентів, таких як маршрутизатори, комутатори та інше фізичне обладнання;
- можливість налаштування; підтримує спеціальні показники;
- підтримує моніторинг серверів Windows і Linux.

До мінусів:

- обмежений набір інформаційних панелей за замовчуванням;
- накладні витрати на технічне обслуговування та експлуатацію.

Zabbix

Zabbix це монітонинг з відкритим кодом і є безкоштовним, але якщо є необхідність, є можливість доповнити його платними послугами, такими як технічна підтримка, консультації та підтримка оновлення/створення шаблонів.

Zabbix – це універсальний і популярний інструмент моніторингу системи з відкритим вихідним кодом, який збирає, агрегує та аналізує показники з мереж, серверів/віртуальних машин, хмар, програм, служб, баз даних тощо.

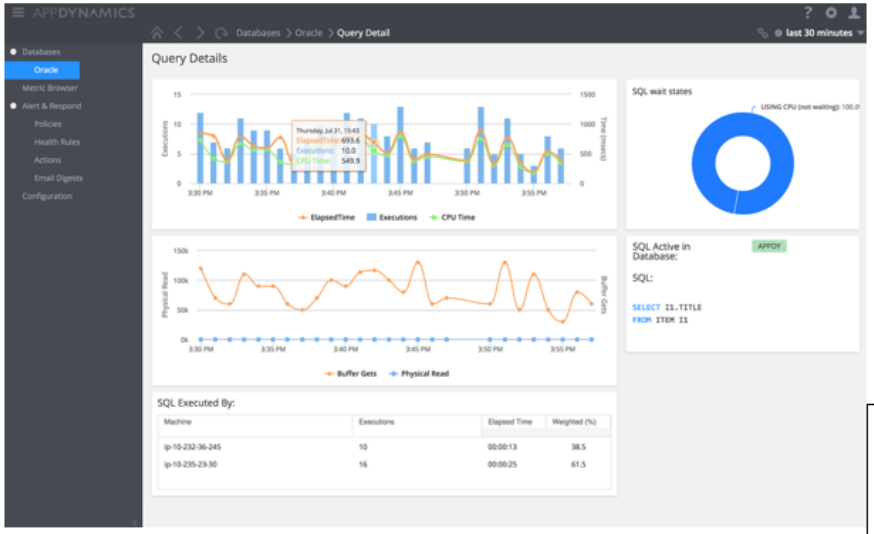
Для зберігання даних використовують MySQL, PostgreSQL, SQLite або Oracle Database, вебінтерфейс написаний на PHP. Підтримує кілька видів моніторингу:

- Simple checks – може перевіряти доступність і реакцію стандартних сервісів, таких як SMTP або HTTP, без встановлення будь-якого програмного забезпечення на хості, що спостерігається;
- Zabbix agent – може бути встановлений на UNIX-подібних або Windows-хостах для отримання даних про навантаження процесора, використання мережі, дисковому просторі тощо;
- External check – виконання зовнішніх програм, також підтримується моніторинг через SNMP.

До плюсів можна віднести:

– проста інтеграція на основі API з існуючими програмами;
– автоматично виявляє аномалії та прогнозує тенденції за допомогою розумних гнучких порогів;
– класифікує виявлені проблеми для ефективнішого сповіщення та прискореного аналізу першопричини;
– видимість на одній панелі з настроюваними інформаційними панелями, графіками та звітами.

AppDynamics



AppDynamics – надає комплексне рішення для моніторингу інфраструктури, яке охоплює компоненти сервера, сховища та мережі як у хмарних, так і в гібридних середовищах. Ви можете розгорнути його локально або використовувати як службу SaaS.

Можливості повного стека моніторингу цього інструменту допомагають співвідносити проблеми продуктивності програми з вузькими місцями інфраструктури низького рівня, прискорюючи тим самим аналіз основних причин і усунення.

Завдяки повному набору інформаційних панелей і показників AppDynamics підтримує детальні сповіщення, які можна інтегрувати зі сторонніми інструментами оповіщення та керування інцидентами, такими як ServiceNow, PagerDuty та Jira.

AppDynamics стягує плату за ядро ЦП. Доступна 15-денна безкоштовна пробна версія. Від стандартної версії за 6$ за одне ядро ЦП, на місяць, до Enterprise Edition – 90$ за одне ядро ЦП, на місяць.

До плюсів можна віднести:

- кореляція показників продуктивності програми з показниками продуктивності сервера та мережі;
- виявлення аномалій і сповіщення;
- перша в бізнесі платформа спостереження з рекомендаціями щодо планування потужностей.

До мінусів:

- для розширених функцій необхідне навчання.

Висновки

Під час написання роботи детально проаналізував, дослідив і описав характеристики, особливості, переваги та недоліки сучасних систем моніторингу, для збору показників сервера. Сучасні складні IT-середовища складаються з переплетених рівнів інфраструктури та вбудованого програмного забезпечення, сервісів і служб, а також програмного забезпечення користувача. Кожен рівень генерує свої власні показники, тому дуже важливо мати хороші інструменти моніторингу системи. Моніторинг продуктивності сервера має вирішальне значення для виявлення ризиків і оптимізації продуктивності сервера. Завдяки вискористанню сучасних систем моніторингу, адміністратор зможе швидко й ефективно налагоджувати та усувати проблеми продуктивності системи.

Дані, зібрані системами IT-моніторингу, дають організації глибоке уявлення про IT-середовище. Це допомагає запобігти можливим простоям і стає все більш корисним у міру розвитку IT-середовища. З цього виходить, що:

- необхідно обирати систему моніторингу, яка дозволяє налаштовувати політки, докладні сповіщання та візуальні дешборди, на вимогу користувача. Тобто максимально гнучку систему моніторингу. Оскільки не всі системи моніторингу дозволяють тонко підлаштовувати певні параметри під конкретні потреби, а працюю за шаблонами;
- в залежності від потреб, майже всі системи моніторингу, пропонують різні пакети своїх сервісів, від безкоштовних з відкритим кодом, до передплачуваних, з місячною або річною оплатою, і відрізняються насиченістю цих пакетів.

Апробація

1. Юхименко В.М. «СТАНДАРТ БЕЗДРОТОВОГО ЗВ’ЯЗКУ WI-FI 7». Тези доповіді на ВСЕУКРАЇНСЬКА НАУКОВО-ТЕХНІЧНА КОНФЕРЕНЦІЯ «ТЕХНОЛОГІЧНІ ГОРИЗОНТИ: ДОСЛІДЖЕННЯ ТА ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ ТЕХНОЛОГІЧНОГО ПРОГРЕСУ УКРАЇНИ І СВІТУ» - Київ, 28 листопада 2023 р. ст.345

2. Юхименко В.М. «МЕРЕЖА СТІЛЬНИКОВОГО ЗВ’ЯЗКУ 5G». Тези доповіді на ВСЕУКРАЇНСЬКА НАУКОВО-ТЕХНІЧНА КОНФЕРЕНЦІЯ «ТЕХНОЛОГІЧНІ ГОРИЗОНТИ: ДОСЛІДЖЕННЯ ТА ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ ТЕХНОЛОГІЧНОГО ПРОГРЕСУ УКРАЇНИ І СВІТУ» - Київ, 28 листопада 2023 р. ст.342

3. Юхименко В.М. «ХМАРНІ ТЕХНОЛОГІЇ В ПОВСЯКДЕННОМУ ЖИТТІ». Тези доповіді на ВСЕУКРАЇНСЬКА НАУКОВО-ТЕХНІЧНА КОНФЕРЕНЦІЯ «ТЕХНОЛОГІЧНІ ГОРИЗОНТИ: ДОСЛІДЖЕННЯ ТА ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ ТЕХНОЛОГІЧНОГО ПРОГРЕСУ УКРАЇНИ І СВІТУ» - Київ, 28 листопада 2023 р. ст. 336-342

Дякую за увагу!